CHAPTER 12

User Databases

The Cisco Secure Access Control Server Release 4.2, hereafter referred to as ACS, authenticates users against one of several possible databases, including its internal database. You can configure ACS to authenticate users with more than one type of database. With this flexibility you can use user account data that is collected in different locations without having to explicitly import the users from each external user database into the ACS internal database. You can also apply different databases to different types of users, depending on the security requirements that are associated with user authorizations on your network. For example, a common configuration is to use a Windows user database for standard network users and a token server for network administrators. For information about authentication protocols and the external database types that support them, see Authentication Protocol-Database Compatibility, page 1-8.

Note For information about the Unknown User Policy and group mapping features, see Chapter 15, "Unknown User Policy" and Chapter 16, "User Group Mapping and Specification."

This chapter contains:

ACS Internal Database

The ACS internal database is crucial for the authorization process. Regardless of whether a user is authenticated by the internal user database or by an external user database, ACS authorizes network services for users based on group membership and specific user settings in the ACS internal database. For information about the types of authentication that the ACS internal database supports, see Authentication Protocol-Database Compatibility, page 1-8.

About the ACS Internal Database

For users who are authenticated by using the ACS internal database, ACS stores user passwords in a database which is protected by an administration password and encrypted by using the AES 128 algorithm. For users who are authenticated with external user databases, ACS does not store passwords in the ACS internal database.

Unless you have configured ACS to authenticate users with an external user database, ACS uses usernames and passwords in the ACS internal database during authentication. For more information about specifying an external user database for authentication of a user, see Adding a Basic User Account, page 6-3.

User Import and Creation

The following facilities can import or create user accounts:

- **RDBMS Synchronization**—You can use RDBMS Synchronization to create large numbers of user accounts and configure many settings for user accounts. RDBMS also supports import of user accounts from external sources. We recommend that you use this feature whenever you need to import users by bulk; however, setting up RDBMS Synchronization for the first time requires several important decisions and time to implement them. For more information, see RDBMS Synchronization, page 8-17.
- **CSUtil.exe** (ACS for Windows)—The **CSUtil.exe** command-line utility provides a simple means of creating basic user accounts. **CSUtil.exe** also supports import of user accounts from external sources. When compared to RDBMS Synchronization, its functionality is limited; however, it is simple to prepare for importing basic user accounts and assigning users to groups. For more information, see Appendix C, "CSUtil Database Utility."

The following facilities can create user accounts:

- **ACS web interface**—The web interface provides the ability to create user accounts manually, one user at a time. Regardless of how a user account was created, you can edit a user account by using the web interface. For detailed steps, see Adding a Basic User Account, page 6-3.
- **Unknown User Policy**—The Unknown User Policy enables ACS to add users automatically when it finds a user without an account in an external user database. The creation of a user account in ACS occurs only when the user attempts to access the network and is successfully authenticated by an external user database. For more information, see Chapter 15, "Unknown User Policy."

  If you use the Unknown User Policy, you can also configure group mappings so that each time a user who was added to ACS by the Unknown User Policy is authenticated, the user group assignment is made dynamically. For some external user database types, user group assignment is based on group membership in the external user database. For other database types, all users who were authenticated by a given database are assigned to a single ACS user group. For more information about group mapping, see Chapter 16, "User Group Mapping and Specification."
- **Database Replication**—Database Replication creates user accounts on a secondary ACS by overwriting all existing user accounts on a secondary ACS with the user accounts from the primary ACS. Any user accounts that are unique to a secondary ACS are lost in the replication. For more information, see ACS Internal Database Replication, page 8-1.

About External User Databases

You can configure ACS to forward authentication of users to one or more external user databases. Support for external user databases means that ACS does not require that you create duplicate user entries in the user database. In organizations in which a substantial user database already exists, ACS can leverage the work already invested in building the database without any additional input.

In addition to performing authentication for network access, ACS can perform authentication for TACACS+ enabling privileges by using external user databases. For more information about TACACS+ enable passwords, see Setting TACACS+ Enable Password Options for a User, page 6-23.

Note You can only use external user databases to authenticate users and to determine the group to which ACS assigns a user. The ACS internal database provides all authorization services. With few exceptions, ACS cannot retrieve authorization data from external user databases. Exceptions are noted where applicable in the discussions of specific databases in this chapter. For more information about group mapping for unknown users, see Chapter 16, "User Group Mapping and Specification."

Users can be authenticated when using the following databases:

- Windows User Database
- Generic LDAP Open Database Connectivity (ODBC)-compliant relational databases (ACS for Windows)
- LEAP Proxy Remote Authentication Dial-In User Service (RADIUS) servers
- RADIUS Token server
- RSA SecurID Token Server
- RSA Authentication with LDAP Group Mapping

For ACS to interact with an external user database, ACS requires an API for the third-party authentication source. Then ACS communicates with the external user database by using the API.

ACS for Windows

For RSA token servers, you can install the software components that RSA provides or you can use the RADIUS interface. For token servers by other vendors, the standard RADIUS interface serves as the third-party API.

For Open Database Connectivity (ODBC) authentication sources, in addition to the Windows ODBC interface, you must install the third-party ODBC driver on the ACS Windows server.

ACS SE

For RSA token servers, you must use the RADIUS interface.

For Windows user databases, you must install and configure the ACS Remote Agent for Windows. The Remote Agent interacts with the Windows operating system to provide authentication. See the *Installation and Configuration Guide for Cisco Secure ACS Remote Agents Release 4.2*.

Authenticating with External User Databases

Authenticating users with an external user database requires more than configuring ACS to communicate with an external user database. Performing one of the configuration procedures in this chapter for an external database does not, on its own, instruct ACS to authenticate any users with that database.

After you have configured ACS to communicate with an external user database, you can configure ACS to authenticate users with the external user database by:

- **Specific User Assignment**—You can configure ACS to authenticate specific users with an external user database. To do this, the user must exist in the ACS internal database and you must set the Password Authentication list in User Setup to the external user database that ACS should use to authenticate the user.

  While setting the Password Authentication for every user account is time-consuming, this method of determining which users are authenticated with an external user database is secure because it requires explicit definition of who should authenticate by using the external user database. In addition, the users may be placed in the desired ACS group and thereby receive the applicable access profile.

- **Unknown User Policy**—You can configure ACS to attempt authentication of users who are not in the ACS internal database by using an external user database. You do not need to define new users in the ACS internal database for this method. For more information about the Unknown User Policy, see About Unknown User Authentication, page 15-3.

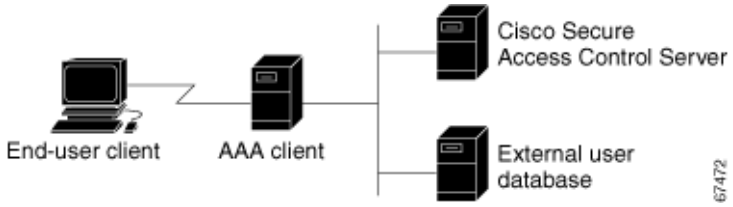
- **Network access profiles (NAPs)—**You can configure NAPs to define which external databases are used to validate the credentials of the user for authentication. For more information about configuring authentication in NAPs, see Authentication Policy Configuration for NAPs, page 14-27.

You can configure ACS with any or all of the previous methods; these methods are not mutually exclusive.

External User Database Authentication Process

When ACS attempts user authentication with an external user database, it forwards the user credentials to the external user database. The external user database passes or fails the authentication request from ACS. On receiving the response from the external user database, ACS instructs the requesting AAA client to grant or deny the user access, depending on the response from the external user database. Figure 12-1 shows a AAA configuration with an external user database.

Figure 12-1 A Simple AAA Scenario



For more information, see the section regarding the database type in which you are interested.

Windows User Database

You can configure ACS to use a Windows user database to authenticate users.

This section contains:

Windows User Database Support

ACS supports the use of Windows external user databases for:

- **User Authentication**—For information about the types of authentication that ACS supports with Windows Security Accounts Manager (SAM) database or a Windows Active Directory database, see Authentication Protocol-Database Compatibility, page 1-8.
- **Machine Authentication**—ACS supports machine authentication with EAP-TLS and PEAP (EAP-MS-CHAPv2). For more information, see EAP and Windows Authentication, page 12-10.
- **Group Mapping for Unknown Users**— ACS supports group mapping for unknown users by requesting group membership information from Windows user databases. For more information about group mapping for users authenticated with a Windows user database, see Group Mapping by Group Set Membership, page 16-3.
- **Password-Aging**— ACS supports password aging for users who are authenticated by a Windows user database. For more information, see User-Changeable Passwords with Windows User Databases, page 12-16.
- **Dial-in Permissions**—ACS supports use of dial-in permissions from Windows user databases. For more information, see Preparing Users for Authenticating with Windows, page 12-17.
- **Callback Settings**—ACS supports use of callback settings from Windows user databases. For information about configuring ACS to use Windows callback settings, see Setting the User Callback Option, page 6-6.

Authentication with Windows User Databases

ACS forwards user credentials to a Windows database by passing the user credentials to the Windows operating system of the computer that is running ACS for Windows or the Solution Engine remote agent. The Windows database passes or fails the authentication request from ACS.

ACS for Windows only: When receiving the response from the Windows database agent ACS instructs the requesting AAA client to grant or deny the user access, depending on the response from the Windows database.

Solution Engine only: When receiving the response from the Windows database, the remote agent forwards the response to ACS, and ACS instructs the requesting AAA client to grant or deny the user access, depending on the response from the Windows database.

ACS grants authorization based on the ACS group to which the user is assigned. While you can determine the group to which a user is assigned information from the Windows database, it is ACS that grants authorization privileges.

To further control access by a user, you can configure ACS to also check the setting for granting dial-in permission to the user. This setting is labeled Grant dialin permission to user in Windows NT and **Allow access** in the Remote Access Permission area in Windows 2000 and Windows 2003 R2. If this feature is disabled for the user, access is denied; even if the username and password are typed correctly.

Trust Relationships

ACS can take advantage of trust relationships established between Windows domains. If the domain that contains ACS for Windows or the computer running the Windows remote agent (ACS SE) trusts another domain, ACS can authenticate users whose accounts reside in the other domain. ACS can also reference the **Grant dialin permission to user** setting across trusted domains.

Note If ACS for Windows is running on a member server, rather than a domain controller, taking advantage of trust relationships depends on proper configuration of ACS for Windows at installation. For more information, see the *Installation Guide for Cisco Secure ACS for Windows Release 4.2*.

If the ACS SE remote agent is running on a member server, rather than a domain controller, taking advantage of trust relationships depends on proper configuration of the remote agent at installation. For more information, see "Configuring for Member Server Authentication" in the *Installation Guide for Cisco Secure ACS Solution Engine Release 4.2*.

ACS can take advantage of indirect trusts for Windows authentication. Consider the example of Windows domains A, B, and C, where ACS for Windows or the remote agent resides on a server in domain A. Domain A trusts domain B, but no trust relationship is established between domain A and domain C. If domain B trusts domain C, ACS for Windows or the remote agent in domain A can authenticate users whose accounts reside in domain C, making use of the indirect trust of domain C.

For more information on trust relationships, refer to your Microsoft Windows documentation.

Windows Dial-Up Networking Clients

The dial-up networking clients for Windows NT/2000/2003 R2/XP Professional and Windows 95/98/Millennium Edition (ME)/XP Home enable users to connect to your network remotely; but the fields that are provided differ:

Windows Dial-Up Networking Clients with a Domain Field

If users dial in to your network by using the dial-up networking client that is provided with Windows NT, Windows 2000, Windows 2003 R2, or Windows XP Professional, three fields appear:

- **username**—Type your username.
- **password**—Type your password.
- **domain**—Type your valid domain name.

Note For more information about the implications of completing or leaving the domain box blank, see Nondomain-Qualified Usernames, page 12-8.

Windows Dial-Up Networking Clients without a Domain Field

If users access your network by using the dial-up networking client that is provided with Windows 95, Windows 98, Windows ME, or Windows XP Home, two fields appear:

- **username**—Type your username.

Note You can also prefix your username with the name of the domain in to which you want to log. For more information about the implications of prefixing or not prefixing the domain name before the username, see Nondomain-Qualified Usernames, page 12-8.

- **password**—Type your password.

Usernames and Windows Authentication

This section contains:

Username Formats and Windows Authentication

ACS supports Windows authentication for usernames in a variety of formats. When ACS attempts Windows authentication, it first determines the username format and submits the username to Windows in the applicable manner. To implement reliable Windows authentication with ACS, you must understand how ACS determines a username format, how it supports each of these formats, and how the types of support are related.

To determine the format of a username that is submitted for Windows authentication, ACS searches the username for the:

- At symbol (@)
- Backslash (\)

Based on the presence and position of these two characters in the username, ACS determines username format by using the following logic:

1. If the username does not contain a backslash (\) *and* does not contain an at symbol (@), ACS considers the username to be nondomain qualified. For example, the username *cyril.yang* is nondomain qualified. For more information, see Nondomain-Qualified Usernames, page 12-8.
2. If the username contains a backslash (\) that precedes any at characters, ACS considers the username to be domain qualified. For example, ACS considers the following usernames to be domain qualified:
   - *MAIN\cyril.yang*
   - *MAIN\cyril.yang@central-office*

   For more information, see Domain-Qualified Usernames, page 12-9.
3. If the username contains an at symbol (@) that does not follow a backslash (\), ACS considers the username to be in User Principal Name (UPN) format. For example, ACS considers the following usernames to be UPN usernames:
   - *cyril.yang@example.com*
   - *cyril.yang@main.example.com*
   - *cyril.yang@main*
   - *cyril.yang@central-office@example.com*
   - *cyril.yang@main\example.com*

   For more information, see UPN Usernames, page 12-9.

Nondomain-Qualified Usernames

ACS supports Windows authentication of usernames that are not domain qualified, provided the username does not contain an at symbol (@). Users with at symbols (@) in their usernames must submit the username in UPN format or in a domain-qualified format. Examples of nondomain-qualified usernames are *cyril.yang* and *msmith*.

In Windows environments with multiple domains, authentication results with nondomain-qualified usernames can vary. This variance occurs because Windows, not ACS, determines which domains are used to authenticate a nondomain-qualified username. If Windows does not find the username in its local domain database, it then checks all trusted domains. If ACS for Windows or the remote agent runs on a member server and the username is not found in trusted domains, Windows also checks its local accounts database. Windows attempts to authenticate a user with the first occurrence of the username that it finds.

When Windows authentication for a nondomain-qualified username succeeds, the privileges that are assigned during authentication will be those that are associated with the Windows user account in the first domain with a matching username and password. This condition also illustrates the importance of removing usernames from a domain when the user account is no longer needed.

Note If the credentials that the user submits do not match the credentials that are associated with the first matching username that Windows finds, authentication fails. Thus, if different users in different domains share the same exact username, logging in with a nondomain-qualified username can result in inadvertent authentication failure.

Use of the Domain List is not required to support Windows authentication, but it can alleviate authentication failures that nondomain-qualified usernames cause. If you have configured the Domain List in the Windows User Database Configuration page of the External User Databases section, ACS submits the username and password to each domain in the list in a domain-qualified format until it successfully authenticates the user. If ACS has tried each domain in the Domain List or if no trusted domains have been configured in the Domain List, ACS stops attempting to authenticate the user and does not grant that user access.

Note If your Domain List contains domains and your Windows Security Account Manager (SAM) or Active Directory user databases are configured to lock out users after a number of failed attempts, users can be inadvertently locked out because ACS tries each domain in the Domain List explicitly, resulting in failed attempts for identical usernames that reside in different domains.

Domain-Qualified Usernames

The most reliable method of authenticating users against a specific domain is to require users to submit the domains that they should be authenticated against along with their usernames. Authentication of a domain-qualified username is directed to a specific domain; rather than depending on Windows to attempt authentication with the correct domain or on using the Domain List to direct ACS to submit the username repeatedly in a domain-qualified format.

Domain-qualified usernames have the following format:

DOMAIN\user

For example, the domain-qualified username for user Mary Smith (*msmith*) in Domain10 would be *Domain10\msmith*.

For usernames containing an at symbol (@), such as *cyril.yang@central-office*, using a domain-qualified username format is required. For example, *MAIN\cyril.yang@central-office*. If a username containing an at symbol (@) is received in a nondomain-qualified format, ACS perceives it as a username in UPN format. For more information, see UPN Usernames, page 12-9.

UPN Usernames

ACS supports authentication of usernames in UPN format, such as *cyril.yang@example.com* or *cyril.yang@central-office@example.com*.

If the authentication protocol is EAP-TLS, by default, ACS submits the username to Windows in UPN format. For all other authentication protocols that it can support with Windows databases, ACS submits the username to Windows that is stripped of all characters after and including the last at symbol (@). This behavior allows for usernames that contain an at symbol (@). For example:

- If the username received is *cyril.yang@example.com*, ACS submits to Windows an authentication request containing the username *cyril.yang*.
- If the username received is *cyril.yang@central-office@example.com*, ACS submits to Windows an authentication request containing the username *cyril.yang@central-office*.

Note ACS cannot tell the difference between a nondomain-qualified username that contains an at symbol (@) and a UPN username; all usernames containing an at symbol (@) that do not follow a backslash (\) are submitted to Windows with the final at symbol (@) and the characters that follow it removed. Users with at symbols (@) in their usernames must submit the username in UPN format or in a domain-qualified format.

EAP and Windows Authentication

This section contains information about Windows-specific EAP features that you can configure on the Windows User Database Configuration page.

This section contains:

Machine Authentication

ACS supports the authentication of computers that are running the Microsoft Windows operating systems that support EAP computer authentication, such as Windows XP with Service Pack 1. Machine authentication, also called computer authentication, allows networks services only for computers known to Active Directory. This feature is especially useful for wireless networks, where unauthorized users outside the physical premises of your workplace can access your wireless access points.

When machine authentication is enabled, there are three different types of authentications. When starting a computer, the authentications occur in this order:

- **Machine authentication**—ACS authenticates the computer prior to user authentication. ACS checks the credentials that the computer provides against the Windows user database. If you use Active Directory and the matching computer account in Active Directory has the same credentials, the computer gains access to Windows domain services.
- **User domain authentication**—If machine authentication succeeded, the Windows domain authenticates the user. If machine authentication failed, the computer does not have access to Windows domain services and the user credentials are authenticated by using cached credentials that the local operating system retains. In this case, the user can log in to only the local system. When a user is authenticated by cached credentials instead of the domain, the computer does not enforce domain policies, such as running login scripts that the domain dictates.

Tip If a computer fails machine authentication and the user has not successfully logged in to the domain by using the computer since the most recent user password change, the cached credentials on the computer will not match the new password. Instead, the cached credentials will match an older password of the user, provided that the user once logged in to the domain successfully from this computer.

- **User network authentication**—ACS authenticates the user, allowing the user to have network connectivity. If the user profile exists, the user database that is specified is used to authenticate the user. While the user database is not required to be the Windows user database, most Microsoft clients can be configured to automatically perform network authentication by using the same credentials used for user domain authentication. This method allows for a single sign-on.

Note Microsoft PEAP clients may also initiate machine authentication whenever a user logs off. This feature prepares the network connection for the next user login. Microsoft PEAP clients may also initiate machine authentication when a user has selected to shutdown or restart the computer; rather than just logging off.

ACS supports EAP-TLS, PEAP (EAP-MS-CHAPv2), and PEAP (EAP-TLS) for machine authentication. You can enable each separately on the Windows User Database Configuration page, which allows a mix of computers that authenticate with EAP-TLS or PEAP (EAP-MS-CHAPv2). Microsoft operating systems that perform machine authentication might limit the user authentication protocol to the same protocol that is used for machine authentication. For more information about Microsoft operating systems and machine authentication, see Microsoft Windows and Machine Authentication, page 12-13.

The Unknown User Policy supports machine authentication. Computers that were previously unknown to ACS are handled similarly to users. If the Unknown User Policy is enabled and an Active Directory external user database is included on the Selected Databases list on the Configure Unknown User Policy page, machine authentication succeeds; provided that the machine credentials presented to Active Directory are valid.

On a computer that is configured to perform machine authentication, machine authentication occurs when the computer started. Provided that the AAA client sends RADIUS accounting data to ACS, when a computer is started and before a user logs in on that computer, the computer appears on the Logged-In Users List in the Reports and Activity section. Once user authentication begins, the computer no longer appears on the Logged-In Users List.

PEAP-based machine authentication uses PEAP (EAP-MS-CHAPv2) and the password for the computer established automatically when it was added to the Microsoft Windows domain. The computer sends its name as the username and the format is:

`host/`*computer*`.`*domain*

where *computer* is the name of the computer and *domain* is the domain to which the computer belongs. The domain segment might also include subdomains, if they are used; so that the format may be:

`host/`*computer*`.`*subdomain*`.`*domain*

The usernames of computers that are authenticated must appear in the ACS internal database. If you enable unknown user processing, ACS adds them automatically once they authenticate successfully. During authentication, the domain name is not used.

EAP-TLS-based machine authentication uses EAP-TLS to authenticate the computer that is using a client certificate. The certificate that the computer uses can be one installed automatically when the computer was added to the domain or one that was added to the local machine storage later. As with PEAP-based machine authentication, the computer name must appear in the ACS internal database in the format contained in the computer client certificate and the user profile corresponding to the computer name must be configured to authenticate by using the Windows external user database. If you enable unknown user processing, ACS adds the computer names to the ACS internal database automatically; once they authenticate successfully. It also automatically configures the user profiles that are created to use the external user database in which the user was found. For machine authentication, this will always be the Windows external user database.

Machine Access Restrictions

You can use the machine access restrictions (MAR) feature as an additional means of controlling authorization for Windows-authenticated EAP-TLS, EAP-FASTv1a, and Microsoft PEAP users, based on machine authentication of the computer used to access the network.

When you enable the feature:

- For every successful machine authentication, ACS caches the value that was received in the Internet Engineering Task Force (IETF) RADIUS `Calling-Station-Id attribute (31)` as evidence of the successful machine authentication. ACS stores each `Calling-Station-Id` attribute value for the number of hours that is specified on the Windows User Database Configuration page before deleting it from the cache.
- When a user authenticates with an EAP-TLS, EAP-FASTv1a, or Microsoft PEAP end-user client, ACS searches the cache of `Calling-Station-Id` values from successful machine authentications for the `Calling-Station-Id` value received in the user authentication request. Whether ACS finds the user-authentication `Calling-Station-Id` value in the cache affects how ACS assigns the user requesting authentication to a user group.
  - **Calling-Station-Id value found in the cache**—ACS assigns the user to a user group by normal methods, which include manual specification of a group in the user profile, group mapping, or RADIUS-based group specification. For example, if a user logs in with a computer that was successfully authenticated and the user profile indicates that the user is a member of group 137, ACS applies to the user session the authorization that were settings specified in group 137.
  - **Calling-Station-Id value not found in the cache**—ACS assigns the user to the user group specified by **Group map for successful user authentication without machine authentication** list. This can include the **<No Access>** group.

Note User profile settings always override group profile settings. If a user profile grants an authorization that is denied by the group specified in the **Group map for successful user authentication without machine authentication** list, ACS grants the authorization.

The MAR feature supports full EAP-TLS, EAP-FASTv1a, and Microsoft PEAP authentication, as well as resumed sessions for EAP-TLS, EAP-FASTv1a, and Microsoft PEAP and fast reconnections for Microsoft PEAP.

The MAR feature has the following limitations and requirements:

- Machine authentication must be enabled.
- Users must authenticate with EAP-TLS, EAP-FASTv1a, or a Microsoft PEAP client. MAR does not apply to users who are authenticated by other protocols, such as LEAP, or MS-CHAP.
- The AAA client must send a value in the IETF RADIUS `Calling-Station-Id attribute (31).`
- ACS does not replicate the cache of `Calling-Station-Id` attribute values from successful machine authentications.
- Users that are authenticated through dial-up will always be treated according to the MAR configuration, since there is no machine authentication when by using dial up. A user will be mapped to a specific group, as defined in the External User Databases > Database Group Mappings > Windows Database settings, when machine authentication occurs. If group mapping is not configured, the user will be mapped to the default group.

Setting Up a MAR Exception List

You might need to set up a MAR exception list if you need to set up specific users (for example managers and administrators) to have access to the network; regardless of whether they pass machine authentication. This feature allows you to select user groups that would be exempt from the MAR.

Before You Begin

So that users can immediately authenticate as part of the MAR exception list, you should set up the required number of groups and permissions before changing your Windows database settings. To manage your group settings, see Group TACACS+ Settings, page 5-2 and Listing Users in a User Group, page 5-40.

To set up a MAR exception list for selected user groups:

Step 1 From the navigation bar, choose **External User Databases > Database Configuration > Windows Database**.

Step 2 Click **Configure**.

Step 3 In the Windows User Database Configuration page, enable the correct machine authentication settings and move the user groups that you want to include in the MAR exemption list to the Selected Groups list.

Step 4 Click **Submit**.

The exception list is based on ACS user groups to which the relevant NT groups would map. You can create exceptions for several user groups, and map different authorization permission to each group.

Microsoft Windows and Machine Authentication

ACS supports machine authentication with Active Directory in Windows 2000 and 2003 R2. To enable machine authentication support in Windows Active Directory you must:

1. Apply Service Pack 4 to the computer that is running Active Directory.
2. Complete the steps in Microsoft Knowledge Base Article 306260: Cannot Modify Dial-In Permissions for Computers That Use Wireless Networking.

Client operating systems that support machine authentication are:

- Microsoft Windows XP with Service Pack 1 applied.
- Microsoft Windows 2000 with:
  - Service Pack 4 applied.
  - Patch Q313664 applied (available from Microsoft.com).
- Microsoft Windows 2003 R2

The following list describes the essential details of enabling machine authentication on a client computer with a Cisco Aironet 350 wireless adapter. For more information about enabling machine authentication in Microsoft Windows operating systems, please refer to Microsoft documentation.

1. Ensure that the wireless network adapter is installed correctly. For more information, see the documentation that is provided with the wireless network adapter.

2. Ensure that the certification authority (CA) certificate of the CA that issued the ACS server certificate is stored in machine storage on client computers. User storage is not available during machine authentication; therefore, if the CA certificate is in user storage, machine authentication fails.
3. Select the wireless network:
   - In Windows XP, you can choose **Windows Network Connection > Properties > Network Connection Properties**.
   - In Windows 2000, you can manually enter the Service Set Identifier (SSID) of the wireless network. Use the Advanced tab of the properties dialog box for the wireless network adapter.
4. To enable PEAP machine authentication, configure the Authentication tab. In Windows XP, the Authentication tab is available from the properties of the wireless network. In Windows 2000, it is available from the properties of the wireless network connection. To configure the Authentication tab:
   a. Check the **Enable network access control using IEEE 802.1X** check box.
   b. Check the **Authenticate as computer when computer information is available** check box.
   c. From the EAP type list, select **Protected EAP (PEAP)**.
   d. On the **Protected EAP Properties** dialog box, you can enforce that ACS has a valid server certificate by checking the **Validate server certificate** check box. If you do check this check box, you must also select the applicable **Trusted Root Certification Authorities**.
   e. Also open the PEAP properties dialog box, from the **Select Authentication Method** list, select **Secured password (EAP-MS-CHAP v2)**.
5. To enable EAP-TLS machine authentication, configure the Authentication tab. In Windows XP, the Authentication tab is available from the properties of the wireless network. In Windows 2000, it is available from the properties of the wireless network connection.
   a. Check the **Enable network access control using IEEE 802.1X** check box.
   b. Check the **Authenticate as computer when computer information is available** check box.
   c. From the **EAP type** list, select **Smart Card or other Certificate**.
   d. On the **Smart Card or other Certificate Properties** dialog box, select the **Use a certificate on this computer** option.
   e. Also on the **Smart Card or other Certificate Properties** dialog box, you can enforce that ACS has a valid server certificate by checking the **Validate server certificate** check box. If you check this check box, you must also select the applicable Trusted Root Certification Authorities.

If you have a Microsoft certification authority server that is configured on the domain controller, you can configure a policy in Active Directory to produce a client certificate automatically when a computer is added to the domain. For more information, see the Microsoft Knowledge Base Article 313407, HOW TO: Create Automatic Certificate Requests with Group Policy in Windows.

Enabling Machine Authentication

This procedure contains an overview of the detailed procedures required to configure ACS to support machine authentication.

Note You must configure end-user client computers and the applicable Active Directory to support machine authentication. This procedure is specific to configuration of ACS only. For information about configuring Microsoft Windows operating systems to support machine authentication, see Microsoft Windows and Machine Authentication, page 12-13.

Note Solution Engine only: Windows authentication requires that you install at least one ACS Remote Agent for Windows and complete the steps in Adding a Remote Agent, page 3-21. For information about installing the ACS Remote Agent for Windows, see the *Installation and Configuration Guide for Cisco Secure ACS Remote Agents Release 4.2*.

To enable ACS to perform machine authentication:

Step 1 Install a server certificate in ACS. PEAP (EAP-MS-CHAPv2) and EAP-TLS require a server certificate. ACS uses a single certificate to support both protocols. For detailed steps, see Installing an ACS Server Certificate, page 9-22.

Note If you have installed a certificate to support EAP-TLS or PEAP user authentication or to support HTTPS protection of remote ACS administration, you do not need to perform this step. A single server certificate will support all certificate-based ACS services and remote administration.

Step 2 For EAP-TLS machine authentication, if certificates on end-user clients are issued by a different CA than the CA that issued the server certificate on ACS, you must edit the certification trust list so that CAs that issue end-user client certificates are trusted. If you do not perform this step and the CA of the server certificate is not the same as the CA of an end-user client certificate CA, EAP-TLS will operate normally; but reject the EAP-TLS machine authentication because it does not trust the correct CA. For detailed steps, see Editing the Certificate Trust List, page 9-28.

Step 3 Enable the applicable protocols on the Global Authentication Setup page:

- To support machine authentication with PEAP, enable the PEAP (EAP-MS-CHAPv2) protocol.
- To support machine authentication with EAP-TLS, enable the EAP-TLS protocol.

Note Solution Engine only: If you are using a Network Access Profile (NAP), the same protocols must be enabled in the NAP configuration.

You can use ACS to complete this step only after you have successfully completed Step 1. For detailed steps, see Configuring Authentication Options, page 9-21.

Step 4 Configure a Windows external user database and enable the applicable types of machine authentication on the Windows User Database Configuration page:

- To support machine authentication with PEAP, check the **Enable PEAP machine authentication** check box.

- To support machine authentication with EAP-TLS, check the **Enable EAP-TLS machine authentication** check box.
- To require machine authentication in addition to user authentication, check the **Enable machine access restrictions** check box.

Note If you already have a Windows external user database configured, modify its configuration to enable the applicable machine authentication types.

For detailed steps, see Configuring a Windows External User Database, page 12-21.

Note Solution Engine only: Windows authentication requires an ACS Remote Agent for Windows.

ACS is ready to perform machine authentication for computers whose names exist in the ACS internal database.

Step 5 If you have not already enabled the Unknown User Policy and added the Windows external user database to the Selected Databases list, consider doing so to allow computers that are not known to ACS to authenticate. For detailed steps, see Configuring the Unknown User Policy, page 15-8.

Note Enabling the Unknown User Policy to support machine authentication also enables the Unknown User Policy for user authentication. ACS makes no distinction in unknown user support between computers and users.

Step 6 If you have users to whom you want to allow access to the network, even if they do not pass machine authentication, you might want to set up a list of user groups that are exempt from the MAR. For detailed steps, see Machine Access Restrictions, page 12-12.

ACS is ready to perform machine authentication for computers; regardless of whether the computer names exist in ACS internal database.

User-Changeable Passwords with Windows User Databases

For network users who are authenticated by a Windows user database, ACS supports user-changeable passwords on password expiration. You can enable this feature in the MS-CHAP Settings and Windows EAP Settings tables on the Windows User Database Configuration page in the External User Databases section. The use of this feature in your network requires that:

- Users must be present in the Windows Active Directory or SAM user database.
- User accounts in ACS must specify the Windows user database for authentication.
- End-user clients must be compatible with MS-CHAP, PEAP (EAP-GTC), PEAP (EAP-MS-CHAPv2), or EAP-FAST.
- The AAA client that the end-user clients connect to must support the applicable protocols:
  - For MS-CHAP password aging, the AAA client must support RADIUS-based MS-CHAP authentication.
  - For PEAP (EAP-MS-CHAPv2), PEAP (EAP-GTC), and EAP-FAST password aging, the AAA client must support EAP.

When the previous conditions are met and this feature is enabled, users receive a dialog box prompting them to change their passwords on their first successful authentication after their passwords have expired. The dialog box is the same as Windows presents to users when a user with an expired password accesses a network via a remote-access server.

For more information about password aging support in ACS, see Enabling Password Aging for the ACS Internal Database, page 5-15.

Preparing Users for Authenticating with Windows

Before using the Windows user database for authentication:

Step 1 Ensure that the username exists in the Windows user database.

Step 2 In Windows, for each user account, clear the following **User Properties** check boxes:

- User must change password at next logon
- Account disabled

Step 3 If you want to control dial-in access from within Windows NT, click **Dial-in** and select **Grant dialin permission to user**. In Windows 2000 and Windows 2003 R2, access the **User Properties** dialog box, select the **Dial-In** tab, and in the Remote Access area, click **Allow access**. You must also configure the option to reference this feature under Database Group Mappings in the External User Databases section of ACS.

Selecting Remote Agents for Windows Authentication (Solution Engine Only)

Before you can configure ACS to authenticate users with a Windows external user database, you must select a primary remote agent that is to deliver authentication requests to the Windows operating system. You may also select a secondary remote agent for ACS to use if the primary remote agent is unavailable.

Before You Begin

To complete this procedure, you must have installed at least one ACS Remote Agent for Windows and completed the steps in Adding a Remote Agent, page 3-21.

To select remote agents for Windows authentication:

Step 1 In the navigation bar, click **External User Databases**.

Step 2 Click **Database Configuration**.

ACS displays a list of all possible external user database types.

Step 3 Click **Windows Database**.

ACS displays the External User Database Configuration page.

Step 4 Click **Configure**.

ACS displays the Windows Remote Agent Selection lists.

Step 5 From the **Primary** list, select the remote agent that ACS should always use to authenticate users, provided that the remote agent is available.

Step 6 From the **Secondary** list, select the remote agent that ACS should use to authenticate users when the remote agent selected in the Primary list is unavailable.

Note If you do not want to use a secondary remote agent, from the **Secondary** list, choose **None**.

Step 7 Click **Submit**.

ACS saves the remote agent selections that you made. The Windows User Database Configuration page appears.

Windows User Database Configuration Options

The Windows User Database Configuration page contains:

- **Dialin Permission**—You can restrict network access to users whose Windows accounts have Windows dial-in permission. The Grant dialin permission to user check box controls this feature.

  Note This feature applies to all users when ACS authenticates with a Windows external user database; despite the name of the feature, it is not limited to users who access the network with a dial-up client but is applied regardless of client type. For example, if you have configured a PIX Firewall to authenticate Telnet sessions by using ACS as a RADIUS server, a user authenticated by a Windows external user database would be denied Telnet access to the PIX Firewall if the Dialin Permission feature is enabled and the Windows user account does not have dial-in permission.

Tip Windows dial-in permission is enabled in the Dialin section of user properties in Windows NT and on the Dial-In tab of the user properties in Windows 2000 and Windows 2003 R2.

- **Windows Callback**—You should enable this setting if you have Windows users that require dial-up access with callback and the User Setup or Group Setup callback setting is configured for Windows Database Callback. If dial-in access with callback is not required or is not configured for Windows Database Callback, then do not enable this setting.

  Note If you disable the Windows Callback option, be certain to disable the callback options in the User Setup or Group Setup callback settings. If the settings contain inconsistencies, the client will not receive the callback number.

- **Unknown User Policy**—If the unknown user policy contains additional external databases and the Windows database is not the last database on the Selected Databases list, you might enable this option. For example, If a user does not exist in the Windows database, or has typed an incorrect password, the error `1326(bad username or password)` is returned. ACS treats this error as a `wrong password` error and does not default to another external database. You should enable this option when additional external databases appear after the Windows database in the Selected Databases list. When enabled, ACS searches for the unknown user in the other external databases.

- **Configure Domain List**—ACS tries to authenticate to any domain listed in Available Domains. If your Windows users do not specify their domain when dialing up, ACS relies on Windows to try to locate the appropriate user account. However, Windows may not be able to authenticate a user properly if the same username exists in more than one trusted domain. We recommend that you ask users to enter their domains when dialing in. If this is not practical, you can define a Domain List. If ACS fails to authenticate a user because the account exists in more than one domain and a Domain List exists, ACS will then retry authentication for each domain in the list. The list order is significant: domains that appear earlier in the list will be tried first. Because of the delay (typically two seconds) for each domain that fails authentication, you should set your AAA client timeout accordingly.

  The Domain List is only required if you have multiple user accounts with the same SAM username in more than one domain AND you are not using EAP/PEAP (for example, PAP/MSCHAP) and you cannot supply a domain prefix in the username.

  Use this option only when *all* the following occur:

  - You are using PAP or MSCHAP.
  - Usernames are not domain-prefixed.
  - There are duplicate usernames, for example, administrator.

  When usernames are not domain-prefixed, and there are multiple occurrences of the same username across the entire network, if no domains are in the **Domain List**, Windows can try to authenticate the wrong credentials. When Windows rejects the initial user authentication request, ACS stops attempting to authenticate the user. For more information, see Nondomain-Qualified Usernames, page 12-8. If domains are in the **Domain List**, ACS qualifies the username with a domain from the list and submits the domain-qualified username to Windows, once for each domain in the Domain List, until each domain has rejected the user or until one of the domains authenticates the user.

  In all other cases, the **Domain List** should be left empty as it might cause performance problems if you use it when you do not need to. Also, each time ACS uses the Domain List and checks the wrong account, Windows counts that as a failed login for that user, which can cause an account lockout.

- **MS-CHAP Settings**—You can control whether ACS supports MS-CHAP-based password changes for Windows user accounts. You can use the Permit password changes by checking the MS-CHAP version *N* check boxes to specify which versions of MS-CHAP ACS supports.

  Note The check boxes under MS-CHAP Settings do no affect password aging for Microsoft PEAP, EAP-FAST, or machine authentication.

  For more information about Windows password changes, see Enabling Password Aging for the ACS Internal Database, page 5-15.

- **Windows EAP Settings**:
  - **Enable password change inside PEAP or EAP-FAST**—The **Permit password change inside PEAP or EAP-FAST** check box controls whether ACS supports PEAP-based or EAP-FAST-based password changes for Windows user accounts. PEAP password changes are supported only when the end-user client uses PEAP (EAP-MS-CHAPv2) for user authentication. For EAP-FAST, ACS supports password changes in phase zero and phase two.
  - **Enable PEAP machine authentication**—This check box controls whether ACS performs machine authentication by using a machine name and password with PEAP (EAP-MS-CHAPv2). For more information about machine authentication, see Machine Authentication, page 12-10.

- **Enable EAP-TLS machine authentication**—This check box controls whether ACS performs machine authentication by using a machine name and password with EAP-TLS. For more information about machine authentication, see Machine Authentication, page 12-10.
- **Enable machine access restrictions**— This box determines whether ACS uses machine authentication as a condition for user authorization. The following protocols are supported for machine authentication: Microsoft PEAP EAP-TLS, EAP-FAST v1a, Cisco PEAP-TLS. If one of these protocols is used for machine authentication, the settings for MAR do not effect the user authentication. If a user tries to access the network with a computer that failed machine authentication, or with another protocol that does not support machine authentication, authorizations are implemented according to the machine access restriction configuration. For more information about the MAR feature, see Machine Access Restrictions, page 12-12.

Note Users that are authenticated through dial-up will always be treated according to the MAR configuration, since there is no machine authentication when using dial up. A user will be mapped to a specific group, as defined in the **External User Databases > Database Group Mappings > Windows Database** settings, when machine authentication occurs. If group mapping is not configured, the user will be mapped to the default group.

Tip To enable machine access restrictions, you must specify a number greater than zero (0) in the **Aging time (hours)** box.

- **Aging time (hours)**—This box specifies the number of hours that ACS caches IETF RADIUS `Calling-Station-Id` attribute values from successful machine authentications, for use with the MAR feature. The default value is 12 hours, which means that ACS does not cache `Calling-Station-Id` values.

Note If you do not change the value of the **Aging time (hours)** box to something other than zero (0), all EAP-TLS and Microsoft PEAP users whose computers perform machine authentication are assigned to the group that is specified in the **Group map for successful user authentication without machine authentication** list.

Tip To clear the cache of `Calling-Station-Id` values, type zero (0) in the **Aging time (hours)** box and click **Submit**.

- **Group map for successful user authentication without machine authentication**—This list specifies the group profile that ACS applies to a user who accesses the network from a computer that has not passed machine authentication for longer than the number of hours specified in the Aging time (hours) box. To deny such users any access to the network, select **<No Access>** (which is the default setting).

Note User profile settings always override group profile settings. If a user profile grants an authorization that is denied by the group that is specified in the Group map for successful user authentication without machine authentication list, ACS grants the authorization.

– **User Groups that are exempt from passing machine authentication**— The Selected User Group list controls what ACS does when machine authentication is not successfully completed. If the Selected User Group list contains no groups and Windows rejects the initial machine authentication, ACS stops attempting to authenticate the user. If the Selected User Group list contains groups, ACS will provide authentication and the privileges within that group to the user; even though their computers are unknown to Active Directory.

Note Configuring the User Group list is optional. For more information about the user group management, see Chapter 5, "User Group Management."

Caution If your User Group list contains groups and you configure your Windows SAM or Active Directory user databases to lock out users after a number of failed attempts, users can be inadvertently locked out. You should ensure that your group settings are configured properly.

- **Available User Groups**—This list represents the user groups for which ACS *requires* machine authentication.
- **Selected User Groups**—This list represents the user groups for which ACS *do not require* machine authentication to gain entry into the network.

- **Windows Authentication Configuration—**This option provides the capability of configuring a workstation name for ACS authentications to the Windows Active Directory.
  - **Default: "CISCO"**—Choose this option if you want CISCO as the workstation name.
  - **Local: machine-name**—Choose this option if you want to use your local machine name as the workstation name.
  - **User defined workstation name**—Choose this option to define your own workstation name. You can use alpha numeric characters, the hyphen(-), and the period(.).
  - **Enable nested group evaluation during group mapping** — When this option is enabled, you can create group mappings that use Active Directory groups that a user is an indirect member of. However, this option will not evaluate Domain Local groups in Active Directory so any group mapping that references Domain Local groups will not operate correctly. When this option is disabled, group mapping will not work if it references an Active Directory group of which a user is an indirect member. Only groups that users are direct members of should be used. However, this option will evaluate Domain Local groups in Activate Directory and they can be used in group mappings.

Configuring a Windows External User Database

Note Solution Engine only: You must have completed the steps in Selecting Remote Agents for Windows Authentication (Solution Engine Only), page 12-17.

For information about the options that are available on the Windows User Database Configuration page, see Windows User Database Configuration Options, page 12-18.

To configure ACS to authenticate users against the Windows user database in the trusted domains of your network:

Step 1 In the navigation bar, click **External User Databases**.

Step 2 Click **Database Configuration**.

ACS displays a list of all possible external user database types.

Step 3 Click **Windows Database**.

If no Windows database configuration exists, the Database Configuration Creation table appears. Otherwise, the External User Database Configuration page appears.

Step 4 Click **Configure**.

The Windows User Database Configuration page appears.

Step 5 As needed, configure the options in:

- Dialin Permission
- Windows Callback
- Unknown User Policy
- Domain List
- MS-CHAP Settings
- Windows EAP Settings
- Windows Authentication Configuration

For information about the options on the Windows User Database Configuration page, see Windows User Database Configuration Options, page 12-18.

Note All the settings on the Windows User Database Configuration page are optional and need not be enabled; unless you want to permit and configure the specific features that they support.

Step 6 Click **Submit**.

ACS saves the Windows user database configuration that you created. You can now add it to your Unknown User Policy or assign specific user accounts to use this database for authentication. For more information about the Unknown User Policy, see About Unknown User Authentication, page 15-3. For more information about configuring user accounts to authenticate by using this database, see Chapter 6, "User Management."

Machine Authentication Support in a Multi-Forest Environment

ACS supports machine authentication in a multi-forest environment. Machine authentications succeed as long as an appropriate trust relationship exists between the primary ACS forest and the requested domain's forest. When a requested user's or machine's domain is part of trusted forest, machine authentication will succeed.

User authentication between multiple forests is supported for EAP-FASTv1a with PEAP, MSPEAP, and for EAP-TLS.

Note The multi-forest feature works only when the username contains the domain information.

Generic LDAP

For information about the types of authentication that ACS supports with generic LDAP databases, such as Netscape Directory Services, see Authentication Protocol-Database Compatibility, page 1-8.

ACS supports group mapping for unknown users by requesting group membership information from LDAP user databases. For more information about group mapping for users who are authenticated with an LDAP user database, see Group Mapping by Group Set Membership, page 16-3.

Configuring ACS to authenticate against an LDAP database has no effect on the configuration of the LDAP database. To manage your LDAP database, see your LDAP database documentation.

This section contains:

ACS Authentication Process with a Generic LDAP User Database

ACS forwards the username and password to an LDAP database by using a Transmission Control Protocol (TCP) connection on a port that you specify. The LDAP database passes or fails the authentication request from ACS. When receiving the response from the LDAP database, ACS instructs the requesting AAA client to grant or deny the user access, depending on the response from the LDAP server.

ACS grants authorization based on the ACS group to which the user is assigned. While the group to which a user is assigned can be determined by information from the LDAP server, ACS grants authorization privileges.

Multiple LDAP Instances

You can create more than one LDAP configuration in ACS. By creating more than one LDAP configuration with different IP address or port settings, you can configure ACS to authenticate by using different LDAP servers or different databases on the same LDAP server. Each primary server IP address and port configuration, along with the secondary server IP address and port configuration, forms an LDAP instance that corresponds to one ACS LDAP configuration instance.

ACS does not require that each LDAP instance corresponds to a unique LDAP database. You can have more than one LDAP configuration set to access the same database. This method is useful when your LDAP database contains more than one subtree for users or groups. Because each LDAP configuration supports only one subtree directory for users and one subtree directory for groups, you must configure separate LDAP instances for each user directory subtree and group directory subtree combination for which ACS should submit authentication requests.

For each LDAP instance, you can add it to or omit it from the Unknown User Policy. For more information, see About Unknown User Authentication, page 15-3.

For each LDAP instance, you can establish unique group mapping. For more information, see Group Mapping by Group Set Membership, page 16-3.

Multiple LDAP instances are also important when you use domain filtering. For more information, see Domain Filtering, page 12-24.

LDAP Organizational Units and Groups

LDAP groups do not need the same name as their corresponding ACS groups. You can map the LDAP group to an ACS group with any name that you want to assign. For more information about how your LDAP database handles group membership, see your LDAP database documentation. For more information on LDAP group mappings and ACS, see Chapter 16, “User Group Mapping and Specification.”

Domain Filtering

Using domain filtering, you can control which LDAP instance is authenticates a user based on domain-qualified usernames. Domain filtering is based on parsing the characters at the beginning or end of a username that is submitted for authentication. Domain filtering provides you with greater control over the LDAP instance to which ACS submits any given user authentication request. You can also control whether usernames are submitted to an LDAP server with their domain qualifiers intact.

For example, when EAP-TLS authentication a Windows XP client initiates, ACS receives the username in `username@domainname` format. When PEAP authentication is initiated by a Cisco Aironet end-user client, ACS receives the username without a domain qualifier. If both clients are to be authenticated with an LDAP database that stores usernames without domain qualifiers, ACS can strip the domain qualifier. If separate user accounts are maintained in the LDAP database—domain-qualified and nondomain-qualified user accounts—ACS can pass usernames to the LDAP database without domain filtering.

If you choose to make use of domain filtering, each LDAP configuration that you create in ACS can perform domain filtering:

- **Limiting users to one domain**—Per each LDAP configuration in ACS, you can require that ACS only attempts to authenticate usernames that are qualified with a specific domain name. This corresponds to the Only process usernames that are domain qualified option on the LDAP Configuration page. For more information about this option, see LDAP Configuration Options, page 12-27.

  With this option, each LDAP configuration is limited to one domain and to one type of domain qualification. You can specify whether ACS strips the domain qualification before submitting the username to an LDAP server. If the LDAP server stores usernames in a domain-qualified format, you should not configure ACS to strip domain qualifiers.

  Limiting users to one domain is useful when the LDAP server stores usernames differently per domain: by user context or how the username is stored in ACS—domain qualified or nondomain qualified. The end-user client or AAA client must submit the username to ACS in a domain-qualified format; otherwise ACS cannot determine the user's domain and does not attempt to authenticate the user with the LDAP configuration that uses this form of domain filtering.

- **Allowing any domain but stripping domain qualifiers**—Per each LDAP configuration in ACS, you can configure ACS to attempt to strip domain qualifiers based on common domain-qualifier delimiting characters. This method corresponds to the Process all usernames after stripping domain name and delimiter option on the LDAP Configuration page. For more information about this option, see LDAP Configuration Options, page 12-27.

  ACS supports prefixed and suffixed domain qualifiers. A single LDAP configuration can attempt to strip both prefixed and suffixed domain qualifiers; however, you can only specify one delimiting character each for prefixed and suffixed domain qualifiers. To support more than one type of domain-qualifier delimiting character, you can create more than one LDAP configuration in ACS.

  Allowing usernames of any domain but stripping domain qualifiers is useful when the LDAP server stores usernames in a nondomain-qualified format but the AAA client or end-user client submits the username to ACS in a domain-qualified format.

Note With this option, ACS submits usernames that are nondomain qualified, too. Usernames are not required to be domain qualified to be submitted to an LDAP server.

LDAP Failover

ACS supports failover between a primary LDAP server and secondary LDAP server. In the context of LDAP authentication with ACS, failover applies when an authentication request fails because ACS could not connect to an LDAP server, such as when the server is down or is otherwise unreachable by ACS. To use this feature, you must define the primary and secondary LDAP servers on the LDAP Database Configuration page. Also, you must check the **On Timeout Use Secondary** check box. For more information about configuring an LDAP external user database, see Configuring a Generic LDAP External User Database, page 12-31.

If you check the **On Timeout Use Secondary** check box, and if the first LDAP server that ACS attempts to contact cannot be reached, ACS always attempts to contact the other LDAP server. The first server ACS attempts to contact might not always be the primary LDAP server. Instead, the first LDAP server that ACS attempts to contact depends on the previous LDAP authentications attempt and on the value that you enter in the **Failback Retry Delay** box.

Successful Previous Authentication with the Primary LDAP Server

If, on the previous LDAP authentication attempt, ACS successfully connected to the primary LDAP server, ACS attempts to connect to the primary LDAP server. If ACS cannot connect to the primary LDAP server, ACS attempts to connect to the secondary LDAP server.

If ACS cannot connect with LDAP server, ACS stops attempting LDAP authentication for the user. If the user is an unknown user, ACS tries the next external user database in the Unknown User Policy list. For more information about the Unknown User Policy list, see About Unknown User Authentication, page 15-3.

Unsuccessful Previous Authentication with the Primary LDAP Server

If, on the previous LDAP authentication attempt, ACS could not connect to the primary LDAP server, whether ACS first attempts to connect to the primary server or secondary LDAP server for the current authentication attempt depends on the value in the Failback Retry Delay box. If the Failback Retry Delay box is set to zero (0), ACS always attempts to connect to the primary LDAP server first. And if ACS cannot connect to the primary LDAP server, ACS then attempts to connect to the secondary LDAP server.

If the Failback Retry Delay box is set to a number other than zero (0), ACS determines how many minutes have passed since the last authentication attempt by using the primary LDAP server. If more minutes have passed than the value in the Failback Retry Delay box, ACS attempts to connect to the primary LDAP server first. And if ACS cannot connect to the primary LDAP server, ACS then attempts to connect to the secondary LDAP server.

If fewer minutes have passed than the value in the Failback Retry Delay box, ACS attempts to connect to the secondary LDAP server first. And if ACS cannot connect to the secondary LDAP server, ACS then attempts to connect to the primary LDAP server.

If ACS cannot connect to either LDAP server, ACS stops attempting LDAP authentication for the user. If the user is an unknown user, ACS tries the next external user database in the Unknown User Policy list. For more information about the Unknown User Policy list, see About Unknown User Authentication, page 15-3.

LDAP Admin Logon Connection Management

When ACS checks authentication and authorization of a user on an LDAP server, it uses a connection with the LDAP administrator account permissions. It uses the connection to search for the user and user groups on the Directory subtree. ACS retains the administrator connections that are open for successive use and additional administrator binds are not required for each authentication request. You can limit the maximum number of concurrent administrator connections per Generic LDAP External DB configuration (primary and secondary).

Distinguished Name Caching

Searching can be an expensive LDAP operation, which introduces an element of unpredictability into the authentication. ACS takes the username that the authentication process supplies, and asks the LDAP server to search a full subtree of unknown depth, over an unknown user population.

After successful authentication ACS caches the Distinguished Name (DN) that the search returns. Reauthentications can then use the cached DN to perform an immediate lookup of the user.

A cached DN cannot appear on a screen. If a bind to a cached DN fails, ACS falls back to a full search of the data base for authenticating a user.

LDAP Configuration Options

The LDAP Database Configuration page contains many options, presented in three tables:

- **Domain Filtering**—This table contains options for domain filtering. The settings in this table affect all LDAP authentication that is performed by using this configuration; regardless of whether the primary or secondary LDAP server handles the authentication. For more information about domain filtering, see Domain Filtering, page 12-24

  This table contains:

  - **Process all usernames**—When you select this option, ACS does not perform domain filtering on usernames before submitting them to the LDAP server for authentication.
  - **Only process usernames that are domain qualified**—When you select this option, ACS only attempts authentication for usernames that are domain-qualified for a single domain. You must specify the type of domain qualifier and the domain in the **Qualified by** and **Domain** options. ACS only submits usernames that are qualified in the method that you specify in the **Qualified by** option and that are qualified with the username that is specified in the Domain Qualifier box. You can also specify whether ACS removes the domain qualifier from usernames before submitting them to an LDAP server.
  - **Qualified by**—When you select **Only process usernames that are domain qualified**, this option specifies the type of domain qualification. If you select **Prefix**, ACS only processes usernames that begin with the characters that you specify in the Domain Qualifier box. If you select **Suffix**, ACS only processes usernames that end in the characters that you specify in the Domain Qualifier box.

Note Regardless of the domain qualifier type that is selected, the domain name must match the domain that is specified in the Domain Qualifier box.

  - **Domain Qualifier**—When Only process usernames that are domain qualified is selected, this option specifies the domain name and delimiting character that must qualify usernames so ACS can submit the username to an LDAP server. The Domain box accepts up to 512 characters; however, only one domain name and its delimiting character are permitted.

    For example, if the domain name is *mydomain*, the delimiting character is the at symbol (@), and Suffix is selected on the Qualified by list, the Domain box should contain *@mydomain*. If the domain name is *yourdomain*, the delimiting character is the backslash (\), and Prefix is selected on the Qualified by list, the Domain Qualifier box should contain *yourdomain*\.
  - **Strip domain before submitting username to LDAP server**—When you select Only process usernames that are domain qualified, this option specifies whether ACS removes the domain qualifier and its delimiting character before submitting a username to an LDAP server. For example, if the username is *jwiedman@domain.com*, the stripped username is *jwiedman*.
  - **Process all usernames after stripping domain name and delimiter**—When this option is selected, ACS submits all usernames to an LDAP server after attempting to strip domain names. Usernames that are not domain qualified are processed, too. Domain name stripping occurs as specified by the following two options:

– **Strip starting characters through the last *X* character**—When you select Process all usernames after stripping domain name and delimiter, this option specifies that ACS attempts to strip a prefixed domain qualifier. If, in the username, ACS finds the delimiter character that is specified in the *X* box, it strips all characters from the beginning of the username through the delimiter character. If the username contains more than one of the characters that are specified in the *X* box, ACS strips characters through the last occurrence of the delimiter character.

  For example, if the delimiter character is the backslash (\) and the username is *DOMAIN\echamberlain*, ACS submits *echamberlain* to an LDAP server.

Note The *X* box cannot contain the following special characters: the pound sign (#), the question mark (?), the quote (“), the asterisk (*), the right angle bracket (>), and the left angle bracket (<). ACS does not allow these characters in usernames. If the *X* box contains any of these characters, stripping fails.

– **Strip ending characters through the first *Y* character**—When you select Process all usernames after stripping domain name and delimiter, this option specifies that ACS attempts to strip a suffixed domain qualifier. If, in the username, ACS finds the delimiter character that is specified in the *Y* box, it strips all characters from the delimiter character through the end of the username. If the username contains more than one of the character specified in the *Y* box, ACS strips characters starting with the first occurrence of the delimiter character.

  For example, if the delimiter character is the at symbol (@) and the username is *jwiedman@domain*, then ACS submits *jwiedman* to an LDAP server.

Note The *Y* box cannot contain the following special characters: the pound sign (#), the question mark (?), the quote ("), the asterisk (*), the right angle bracket (>), and the left angle bracket (<). ACS does not allow these characters in usernames. If the *Y* box contains any of these characters, stripping fails.

- **Common LDAP Configuration**—This table contains options that apply to all LDAP authentication that is performed by using this configuration. ACS uses the settings in this section; regardless of whether the primary or secondary LDAP server handles the authentication.

  This table contains:

  – **User Directory Subtree**—The distinguished name (DN) for the subtree that contains all users. For example:

    `ou=`*organizational unit*`[,ou=`*next organizational unit*`]o=corporation.com`

    If the tree containing users is the base DN, type:

    ```
    o=corporation.com
    ```

    or

    ```
    dc=corporation,dc=com
    ```

    as applicable to your LDAP configuration. For more information, refer to your LDAP database documentation.

  – **Group Directory Subtree**—The DN for the subtree that contains all groups. For example:

    `ou=`*organizational unit*`[,ou=`*next organizational unit*`]o=corporation.com`

    If the tree containing groups is the base DN, type:

`o=corporation.com`

or

`dc=corporation,dc=com`

as applicable to your LDAP configuration. For more information, refer to your LDAP database documentation.

- **UserObjectType**—The name of the attribute in the user record that contains the username. You can obtain this attribute name from your Directory Server. For more information, refer to your LDAP database documentation. ACS contains default values that reflect the default configuration of a Netscape Directory Server. Confirm all values for these fields with your LDAP server configuration and documentation.
- **UserObjectClass**—The value of the LDAP `objectType` attribute that identifies the record as a user. Often, user records have several values for the `objectType` attribute, some of which are unique to the user, some of which are shared with other object types. This box should contain a value that is not shared.
- **GroupObjectType**—The name of the attribute in the group record that contains the group name.
- **GroupObjectClass**—A value of the LDAP `objectType` attribute in the group record that identifies the record as a group.
- **Group Attribute Name**—The name of the attribute of the group record that contains the list of user records that are a member of that group.
- **Server Timeout**—The number of seconds that ACS waits for a response from an LDAP server before determining that the connection with that server has failed.
- **On Timeout Use Secondary**—Determines whether ACS performs failover of LDAP authentication attempts. For more information about the LDAP failover feature, see LDAP Failover, page 12-25.
- **Failback Retry Delay**—The number of minutes after the primary LDAP server fails to authenticate a user that ACS resumes sending authentication requests to the primary LDAP server first. A value of zero (0) causes ACS to always use the primary LDAP server first.
- **Max. Admin Connections**—The maximum number of concurrent connections (greater than zero (0)) with LDAP administrator account permissions that can run for a specific LDAP configuration. These connections are used to search the Directory for users and groups under the User Directory Subtree and Group Directory Subtree.

- **Primary and Secondary LDAP Servers**—You can use the Primary LDAP Server table and the Secondary LDAP Server table to identify the LDAP servers and make settings that are unique to each. You do not need to complete the Secondary LDAP Server table if you do not intend to use LDAP failover.

  These tables contain:

  - **Hostname**—The name or IP address of the server that is running the LDAP software. If you are using DNS on your network, you can type the hostname instead of the IP address.
  - **Port**—The TCP/IP port number on which the LDAP server is listening. The default is 389, as stated in the LDAP specification. If you do not know the port number, you can find this information by viewing those properties on the LDAP server. If you want to use secure authentication, port 636 is usually used.

- **LDAP Version**—ACS uses LDAP version 3 or version 2 to communicate with your LDAP database. If you check this check box, ACS uses LDAP version 3. If it is not checked, ACS uses LDAP version 2.
- **Security—**ACS uses SSL to encrypt communication between ACS and the LDAP server. If you do not enable SSL, user credentials are passed to the LDAP server in clear text. If you select this option, then you must select **Trusted Root CA** or **Certificate Database Path**. ACS supports only server-side authentication for SSL communication with the LDAP server. Solution Engine only: You must be sure that the Port box contains the port number used for SSL on the LDAP server.
- **Trusted Root CA—**LDAP over SSL includes the option to authenticate by using the certificate database files other than the Netscape *cert7.db* file. This option uses the same mechanism as other SSL installations in the ACS environment. Select the certification authority that issued the server certificate that is installed on the LDAP server.
- **Certificate DB Path—**Uses the path to the Netscape *cert7.db file*, which contains the certificates for the server to be queried, and the certificates for the trusted CA.

ACS for Windows

- The path to the Netscape *cert7.db* file. This file must contain the certificates for the server to be queried and the trusted CA. You can use a Netscape web browser to generate *cert7.db* files. For information about generating a *cert7.db* file, refer to Netscape documentation.

  To perform secure authentication by using SSL with this option, you must provide a Netscape *cert7.db* certificate database file. ACS requires a certificate database so that it can establish the SSL connection because the certificate database must be local to the ACS Windows server.

ACS SE

- This option provides a link to the Download Certificate Database page. ACS displays information about whether the Netscape *cert7.db* certificate database file has been downloaded to support secure communication to the LDAP server that you specified. For information about the Download Certificate Database page, see Downloading a Certificate Database (Solution Engine Only), page 12-47.

  To perform secure authentication by using SSL with this option, you must provide a Netscape *cert7.db* certificate database file. ACS requires a certificate database so that it can establish the SSL connection. Since the certificate database must be local to the Solution Engine, you must use FTP to transfer the certificate database to ACS.

  ACS requires a *cert7.db* certificate database file for each LDAP server that you configure. For example, to support users distributed in multiple LDAP trees, you might configure two LDAP instances in ACS that can communicate with the same LDAP servers. Each LDAP instance then has a primary and a secondary LDAP server. Even though the two LDAP configurations share the same primary server, each LDAP configuration requires that you download a certificate database file to ACS.

Note The database must be a Netscape *cert7.db* certificate database file. No other filename is supported.

Caution TACACS+ authentications to the back-end LDAP database may not work properly when ACS is operating under a heavy load by using *cert7.db*. TACACS+ services may shut down and authentications cease. The third-party DLL may be unstable and cause exceptions. In addition, Netscape no longer

supports the third-party DLL. The recommended way to work around this problem is to use a secure connection with the OpenSSL infrastructure with a CA root certificate, instead of the *cert7.db*. In this ACS release, Cisco has fully tested and supports this method.

- **Admin DN**—The DN of the administrator; that is, the LDAP account which, if bound to, permits searches for all required users under the User Directory Subtree. It must contain the following information about your LDAP server:

  uid=*user id*,[ou=*organizational unit*,][ou=*next organizational unit*]o=*organization*

  where *user id* is the username, *organizational unit* is the last level of the tree, and *next organizational unit* is the next level up the tree.

  For example:

  ```
  uid=joesmith,ou=members,ou=administrators,o=cisco
  ```

  You can use anonymous credentials for the administrator username if the LDAP server is configured to make the group name attribute visible in searches by anonymous credentials. Otherwise, you must specify an administrator username that permits the group name attribute to be visible to searches.

Note If the administrator username that you specify does not have permission to see the group name attribute in searches, group mapping fails for users that LDAP authenticates.

- **Password**—The password for the administrator account that you specified in the **Admin DN** box. The LDAP server determines case sensitivity.

Configuring a Generic LDAP External User Database

Creating a generic LDAP configuration provides ACS information that enables it to pass authentication requests to an LDAP database. This information reflects the way that you have implemented your LDAP database and does not dictate how your LDAP database is configured or functions. For information about your LDAP database, refer to your LDAP documentation.

Before You Begin

For information about the options on the LDAP Database Configuration page, see LDAP Configuration Options, page 12-27.

To configure ACS to use the LDAP User Database:

Step 1 In the navigation bar, click **External User Databases**.

Step 2 Click **Database Configuration**.

ACS displays a list of all possible external user database types.

Step 3 Click **Generic LDAP**.

Note The user authenticates against only one LDAP database.

If no LDAP database configuration exists, only the Database Configuration Creation table appears. Otherwise, in addition to the Database Configuration Creation table, the External User Database Configuration table appears.

Step 4 If you are creating a configuration:

a. Click **Create New Configuration**.

b. Type a name for the new configuration for generic LDAP in the box provided.

c. Click **Submit**.

ACS lists the new configuration in the External User Database Configuration table.

Step 5 Under External User Database Configuration, select the name of the LDAP database that to configure.

Note If only one LDAP configuration exists, the name of that configuration appears instead of the list. Proceed to Step 6.

Step 6 Click **Configure**.

Caution If you click **Delete**, the configuration of the selected LDAP database is deleted.

Step 7 If you do not want ACS to filter LDAP authentication requests by username, under Domain Filtering, select **Process all usernames**.

Step 8 If you want to limit authentications processed by this LDAP configuration to usernames with a specific domain qualification:

Note For information about domain filtering, see Domain Filtering, page 12-24.

a. Under Domain Filtering, select **Only process usernames that are domain qualified**.

b. From the Qualified by list, select the applicable type of domain qualification, either Suffix or Prefix. Only one type of domain qualification is supported per LDAP configuration.

For example, if you want this LDAP configuration to authenticate usernames that begin with a specific domain name, select **Prefix**. If you want this LDAP configuration to authenticate usernames that end with a specific domain name, select **Suffix**.

c. In the **Domain Qualifier** box, type the name of the domain for which you this LDAP configuration should authenticate usernames. Include the delimiting character that separates the user ID from the domain name. Ensure that the delimiting character appears in the applicable position: at the end of the domain name if **Prefix** is selected on the **Qualified by** list; at the beginning of the domain name if Suffix is selected on the Qualified by list.

Only one domain name is supported per LDAP configuration. You can type up to 512 characters.

d. If you want ACS to remove the domain qualifier before submitting it to the LDAP database, check the **Strip domain before submitting username to LDAP server** check box.

e. If you want ACS to pass the username to the LDAP database *without* removing the domain qualifier, clear the **Strip domain before submitting username to LDAP server** check box.

Step 9 If you want to enable ACS to strip domain qualifiers from usernames before submitting them to an LDAP server:

Note For information about domain filtering, see Domain Filtering, page 12-24.

a. Under Domain Filtering, select **Process all usernames after stripping domain name and delimiter**.

b. If you want ACS to strip prefixed domain qualifiers, check the **Strip starting characters through the last *X* character** check box, and then type the domain-qualifier delimiting character in the *X* box.

Note The *X* box cannot contain the following special characters: the pound sign (#), the question mark (?), the quote ("), the asterisk (*), the right angle bracket (>), and the left angle bracket (<). ACS does not allow these characters in usernames. If the *X* box contains any of these characters, stripping fails.

c. If you want ACS to strip suffixed domain qualifiers, check the **Strip ending characters from the first *X* character** check box, and then type the domain-qualifier delimiting character in the *X* box.

Note The *X* box cannot contain the following special characters: the pound sign (#), the question mark (?), the quote ("), the asterisk (*), the right angle bracket (>), and the left angle bracket (<). ACS does not allow these characters in usernames. If the *X* box contains any of these characters, stripping fails.

Step 10 Under Common LDAP Configuration, in the **User Directory Subtree** box, type the DN of the tree containing all your users.

Step 11 In the **Group Directory Subtree** box, type the DN of the subtree containing all your groups.

Step 12 In the **User Object Type** box, type the name of the attribute in the user record that contains the username. You can obtain this attribute name from your Directory Server. For more information, refer to your LDAP database documentation.

Note The default values in the UserObjectType and following fields reflect the default configuration of the Netscape Directory Server. Confirm all values for these fields with your LDAP server configuration and documentation.

Step 13 In the **User Object Class** box, type the value of the LDAP `objectType` attribute that identifies the record as a user. Often, user records have several values for the `objectType` attribute, some of which are unique to the user, some of which are shared with other object types. Select a value that is not shared.

Step 14 In the **GroupObjectType** box, type the name of the attribute in the group record that contains the group name.

Step 15 In the **GroupObjectClass** box, type a value of the LDAP `objectType` attribute in the group record that identifies the record as a group.

Step 16 In the **GroupAttributeName** box, type the name of the attribute of the group record that contains the list of user records who are a member of that group.

Step 17 In the **Server Timeout** box, type the number of seconds that ACS waits for a response from an LDAP server before determining that the connection with that server has failed.

Step 18 To enable failover of LDAP authentication attempts, check the **On Timeout Use Secondary** check box. For more information about the LDAP failover feature, see LDAP Failover, page 12-25.

Step 19 In the **Failback Retry Delay** box, type the number of minutes after the primary LDAP server fails to authenticate a user that ACS resumes sending authentication requests to the primary LDAP server first.

Note To specify that ACS should always use the primary LDAP server first, type zero (0) in the **Failback Retry Delay** box.

Step 20 In the **Max. Admin Connection** box, enter the number of maximum concurrent connections with LDAP administrator account permissions.

Step 21 For the Primary LDAP Server and Secondary LDAP Server tables:

Note If you did not check the On Timeout Use Secondary check box, you do not need to complete the options in the Secondary LDAP Server table.

a. In the **Hostname** box, type the name or IP address of the server that is running the LDAP software. If you are using DNS on your network, you can type the hostname instead of the IP address.

b. In the **Port** box, type the TCP/IP port number on which the LDAP server is listening. The default is 389, as stated in the LDAP specification. If you do not know the port number, you can find this information by viewing those properties on the LDAP server. If you want to use secure authentication, port 636 is usually used.

c. To specify that ACS should use LDAP version 3 to communicate with your LDAP database, check the **LDAP Version** check box. If the LDAP Version check box is not checked, ACS uses LDAP version 2.

Note If you want ACS to use SSL to connect to the LDAP server, check the **Use secure authentication** check box and complete the next three steps. If you do not use SSL, the username and password credentials are normally passed over the network to the LDAP directory in clear text.

d. Solution Engine only: If you checked the **Use Secure authentication** check box, perform one of the following procedures:

- Check the **Trusted Root CA** check box, and in the adjacent drop-down list, select a **Trusted Root CA**.
- Check the **Certificate Database Path** check box, and download a *cert7.db* file.

Note To download a *cert7.db* certificate database file to ACS now, complete the steps in Downloading a Certificate Database (Solution Engine Only), page 12-47, and then continue with step f. You can download a certificate database later. Until a certificate database is downloaded for the current LDAP server, secure authentication to this LDAP server fails.

e. ACS for Windows only: If you checked the **Use Secure authentication** check box, perform one of the following procedures:

- Click the **Trusted Root CA** button, and in the adjacent drop-down list, select a **Trusted Root CA**.
- Click the **Certificate Database Path** button, and in the adjacent box, type the path to the Netscape *cert7.db* file, which contains the certificates for the server to be queried and the trusted CA.

f. The Admin DN box requires the fully qualified (DN) of the administrator; that is, the LDAP account which, if bound to, permits searches for all required users under the User Directory Subtree.

In the **Admin DN** box, type the following information from your LDAP server:

uid=*user id*, [ou=*organizational unit*,]
[ou=*next organizational unit*]o=*organization*

where *user id* is the username

organizational unit is the last level of the tree

next organizational unit is the next level up the tree.

For example:

uid=joesmith,ou=members,ou=administrators,o=cisco

Tip If you are using Netscape DS as your LDAP software, you can copy this information from the Netscape console.

g. In the **Password** box, type the password for the administrator account that is specified in the Admin DN box. The server determines password case sensitivity.

Step 22 Click **Submit**.

ACS saves the generic LDAP configuration that you created. You can now add it to your Unknown User Policy or assign specific user accounts to use this database for authentication. For more information about the Unknown User Policy, see About Unknown User Authentication, page 15-3. For more information about configuring user accounts to authenticate by using this database, see Chapter 6, "User Management."

ODBC Database (ACS for Windows Only)

As with Windows user database support, you can use ACS ODBC-compliant relational database support to use existing user records in an external ODBC-compliant relational database. Configuring ACS to authenticate against an ODBC-compliant relational database does not affect the configuration of the relational database. To manage your relational database, refer to your relational database documentation.

Note As with all other external databases that ACS supports, the ODBC-compliant relational database is not supplied as part of ACS. For general guidance with setting up your ODBC external user database, see Preparing to Authenticate Users with an ODBC-Compliant Relational Database, page 12-37.

You can use the Windows ODBC feature to create a data source name (DSN), which specifies the database and other important parameters that are necessary for communicating with the database. Among the parameters that you provide are the username and password that are required for the ODBC driver to gain access to your ODBC-compliant relational database.

This section contains:

What is Supported with ODBC User Databases

ACS supports the use of ODBC external user databases for:

- **Authentication**—For information about the types of authentication that ACS supports by using a relational database via the ODBC authenticator feature, see Authentication Protocol-Database Compatibility, page 1-8.
- **Group Specification**—ACS supports group assignment for users who are authenticated by an ODBC user database. Authentication queries to the ODBC database must contain the group number to which you want to assign a user. For unknown users authenticated by an ODBC user database, group specification overrides group mapping.

  For more information about expected query output, see PAP Procedure Output, page 12-41, CHAP/MS-CHAP/ARAP Procedure Output, page 12-42, and EAP-TLS Procedure Output, page 12-43.
- **Group Mapping for Unknown Users**—ACS supports group mapping for unknown users by requesting group membership information from Windows user databases. For more information about group mapping for users who are authenticated with a Windows user database, see Group Mapping by Group Set Membership, page 16-3.

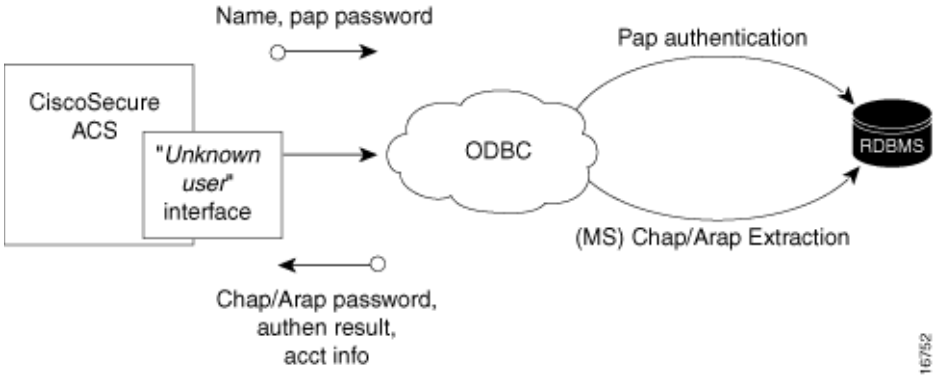
ACS Authentication Process with an ODBC External User Database

ACS forwards user authentication requests to an ODBC database when the user:

- Account in the ACS internal database lists an ODBC database configuration as the authentication method.
- Is unknown to the ACS internal database, and the Unknown User Policy dictates that an ODBC database is the next external user database to try.

In either case, ACS forwards user credentials to the ODBC database via an ODBC connection. The relational database must have a stored procedure that queries the appropriate tables and returns values to ACS. If the returned values indicate that the user credentials that were provided are valid, ACS instructs the requesting AAA client to grant the user access; otherwise, ACS denies the user access (Figure 12-2).

Figure 12-2 Using the ODBC Database for Authentication



ACS grants authorization based on the ACS group to which the user is assigned. While the group to which a user is assigned can be determined by information from the ODBC database by using a process known as "group specification," it is ACS that grants authorization privileges.

Preparing to Authenticate Users with an ODBC-Compliant Relational Database

Authenticating users with an ODBC-compliant relational database requires that you complete several significant steps that are external to ACS before configuring ACS with an ODBC external user database.

To prepare for authenticating with an ODBC-compliant relational database:

Step 1 Install your ODBC-compliant relational database on its server. For more information, refer to the relational database documentation.

Note The relational database that you use is not supplied with ACS.

Step 2 Create the database to hold the usernames and passwords. The database name is irrelevant to ACS, so that you can name the database however you like.

Step 3 Create the table or tables that will hold the usernames and passwords for your users. The table names are irrelevant to ACS, so you can name the tables and columns however you like.

Note For SQL database columns that hold user passwords, we recommend using **varchar** format. If you define password columns as **char**, password comparison might fail if the password does not use the full length of the field. For example, if a password column is 16 characters wide but the password is only ten characters long, the database might append six spaces. The value used for password comparison then grows to 16 characters, causing comparison to the actual password that the user submitted to fail.

Step 4 Write the stored procedures that are intended to return the required authentication information to ACS. For more information about these stored procedures, see Implementation of Stored Procedures for ODBC Authentication, page 12-38.

Step 5 Set up a system DSN on the computer that is running ACS. For steps, see Configuring a System Data Source Name for an ODBC External User Database, page 12-44.

Step 6 Configure ACS to authenticate users with an ODBC database. For steps, see Configuring an ODBC External User Database, page 12-45.

Implementation of Stored Procedures for ODBC Authentication

When you configure ACS to authenticate users against an ODBC-compliant relational database, you must create a stored procedure to perform the necessary query and return the values that ACS expects. The values that are returned and the tasks that are required of the stored procedure vary depending on the authentication protocol used.

Authentication for PAP, or PEAP (EAP-GTC) occurs within the relational database; that is, if the stored procedure finds a record with the username and the password matching the input, the user is considered authenticated.

Authentication for CHAP, MS-CHAP, ARAP, LEAP, or EAP-MD5 occurs within ACS. The stored procedure returns the fields for the record with a matching username, including the password. ACS confirms or denies authentication based on the values that are returned from the procedure.

Authentication for EAP-TLS occurs within ACS. The stored procedure returns the field for the record, indicating whether it found the username in the ODBC database. ACS confirms or denies authentication based on the values that are returned from the procedure and on the validity of the user certificate. For more information about ACS support for the EAP-TLS protocol, see EAP-TLS Authentication, page 9-2.

To support the three sets of protocols, ACS provides different input to, and expects different output from, the ODBC authentication request. This feature requires a separate stored procedure in the relational database to support each of the three sets of protocols.

The ACS product CD contains **stub** routines for creating a procedure in Microsoft SQL Server or an Oracle database. You can modify a copy of these routines to create your stored procedure or write your own. You can see example routines for creating PAP and CHAP/MS-CHAP/ARAP authentication stored procedures in SQL Server in Sample Routine for Generating a PAP Authentication SQL Procedure, page 12-39, and Sample Routine for Generating an SQL CHAP Authentication Procedure, page 12-40.

The following sections provide reference information about ACS data types versus SQL data types, PAP/PEAP (EAP-GTC) authentication procedure input and output, CHAP/MS-CHAP/ARAP authentication procedure input and output, EAP-TLS authentication procedure input and output, and expected result codes. You can use this information while writing your authentication stored procedures in your relational database.

Note Two stored procedures are required when using EAP-FAST; MS-CHAP (used for phase zero provisioning) and PAP (used for phase two authentication).

Type Definitions

The ACS types and their matching SQL types are:

- **Integer**—SQL_INTEGER
- **String**—SQL_CHAR or SQL_VARCHAR

Note For SQL database columns that hold user passwords, we recommend using **varchar** format. If you define password columns as **char**, password comparison might fail if the password does not use the full length of the field. For example, if a password column is 16 characters wide but the password is only ten characters long, the database might append six spaces. The value used for password comparison then grows to 16 characters, causing comparison to the actual password that the user submitted to fail.

Microsoft SQL Server and Case-Sensitive Passwords

If you want your passwords to be case sensitive and are using Microsoft SQL Server as your ODBC-compliant relational database, configure your SQL Server to accommodate this feature. If your users are authenticating by using PPP via PAP or Telnet login, the password might not be case sensitive, depending on how you set the case-sensitivity option on the SQL Server. For example, an Oracle database will default to case sensitive, whereas Microsoft SQL Server defaults to case insensitive. However, in the case of CHAP/ARAP, the password is case sensitive if you configured the CHAP stored procedure.

For example, with Telnet or PAP authentication, the passwords **cisco** or **CISCO** or **CiScO** will all work if you configure the SQL Server to be case insensitive.

For CHAP/ARAP, the passwords **cisco** or **CISCO** or **CiScO** are not the same, regardless of whether the SQL Server is configured for case-sensitive passwords.

Sample Routine for Generating a PAP Authentication SQL Procedure

The following example routine creates a procedure named **CSNTAuthUserPap** in Microsoft SQL Server, the default procedure that ACS uses for PAP authentication. Table and column names that could vary for your database schema appear in variable text. For your convenience, the ACS product CD includes a stub routine for creating a procedure in SQL Server or Oracle. For more information about data type definitions, procedure parameters, and procedure results, see ODBC Database (ACS for Windows Only), page 12-35.

```
if exists (select * from sysobjects where id = object_id (`dbo.CSNTAuthUserPap') and
sysstat & 0xf = 4)drop procedure dbo.CSNTAuthUserPap
GO

CREATE PROCEDURE CSNTAuthUserPap
@username varchar(64), @pass varchar(255)
AS
SET NOCOUNT ON
IF EXISTS( SELECT  username
FROM  users
WHERE  username  = @username
AND  csntpassword  = @pass )
SELECT 0,csntgroup,csntacctinfo,"No Error"
FROM  users
WHERE  username  = @username
```

```
ELSE
SELECT 3,0,"odbc","ODBC Authen Error"
GO

GRANT EXECUTE ON dbo.CSNTAuthUserPap TO ciscosecure
GO
```

Sample Routine for Generating an SQL CHAP Authentication Procedure

The following example routine creates in Microsoft SQL Server a procedure named **CSNTExtractUserClearTextPw**, the default procedure that ACS uses for CHAP/MS-CHAP/ARAP authentication. Table and column names that could vary for your database schema appear in variable text. For more information about data type definitions, procedure parameters, and procedure results, see ODBC Database (ACS for Windows Only), page 12-35.

```
if exists (select * from sysobjects where id = object_id(`dbo.CSNTExtractUserClearTextPw')
and sysstat & 0xf = 4) drop procedure dbo.CSNTExtractUserClearTextPw
GO

CREATE PROCEDURE CSNTExtractUserClearTextPw
@username varchar(64)
AS
SET NOCOUNT ON
IF EXISTS( SELECT  username
FROM  users
WHERE  username  = @username )
SELECT 0,csntgroup,csntacctinfo,"No Error",csntpassword
FROM  users
WHERE  username  = @username
ELSE
SELECT 3,0,"odbc","ODBC Authen Error"
GO

GRANT EXECUTE ON dbo.CSNTExtractUserClearTextPw TO ciscosecure
GO
```

Sample Routine for Generating an EAP-TLS Authentication Procedure

The following example routine creates in Microsoft SQL Server a procedure named **CSNTFindUser**, the default procedure that ACS uses for EAP-TLS authentication. Table and column names that could vary for your database schema appear in variable text. For more information about data type definitions, procedure parameters, and procedure results, see ODBC Database (ACS for Windows Only), page 12-35.

PAP Authentication Procedure Input

Table 12-1 details the input that ACS provides to the stored procedure that supports PAP authentication. The stored procedure should accept the named input values as variables.

Table 12-1 PAP Stored Procedure Input

Field	Type	Explanation
CSNTusername	String	0-64 characters
CSNTpassword	String	0-255 characters

The input names are for guidance only. Procedure variables that are created from them can have different names; however, you must define them in the procedure in the order shown: the username must precede the password variable.

PAP Procedure Output

The stored procedure must return a single row that contains the nonnull fields.

Table 12-2 lists the procedure results that ACS expects as output from stored procedure.

Table 12-2 PAP Stored Procedure Results

Field	Type	Explanation
CSNTresult	Integer	See Table 12-7.
CSNTgroup	Integer	The ACS group number for authorization. You use **0xFFFFFFFF** to assign the default value. Values other than 0-499 are converted to the default. **Note** The group that is specified in the CSNTgroup field overrides group mapping that is configured for the ODBC external user database.
CSNTacctInfo	String	0-16 characters. A customer-defined string that ACS adds to subsequent account log file entries.
CSNTerrorString	String	0-255 characters. A customer-defined string that ACS writes to the **CSAuth** service log file if an error occurs.

The CSNTGroup and CSNTacctInfo fields are processed only after a successful authentication. The *CSNTerrorString* file is logged only after a failure (if the result is greater than or equal to 4).

Note If the ODBC database returns data in **recordset** format rather than in parameters, the procedure must return the result fields in the order previously listed.

CHAP/MS-CHAP/ARAP Authentication Procedure Input

ACS provides a single value for input to the stored procedure that supports CHAP/MS-CHAP/ARAP authentication. The stored procedure should accept the named input value as a variable.

Note Because ACS performs authentication for CHAP/MS-CHAP/ARAP, the user password is not an input (Table 12-3).

Table 12-3 CHAP Stored Procedure Input

Field	Type	Explanation
CSNTusername	String	0-64 characters

The input name is for guidance only. A procedure variable that is created from it can have a different name.

CHAP/MS-CHAP/ARAP Procedure Output

The stored procedure must return a single row that contains the nonnull fields.

Table 12-4 lists the procedure results that ACS expects as output from a stored procedure.

Table 12-4 CHAP/MS-CHAP/ARAP Stored Procedure Results

Field	Type	Explanation
CSNTresult	Integer	See Table 12-7 Result Codes.
CSNTgroup	Integer	The ACS group number for authorization. You use **0xFFFFFFFF** to assign the default value. Values other than 0-499 are converted to the default. **Note** The group that is specified in the CSNTgroup field overrides group mapping that is configured for the ODBC external user database.
CSNTacctInfo	String	0-15 characters. A customer-defined string that ACS adds to subsequent account log file entries.
CSNTerrorString	String	0-255 characters. A customer-defined string that ACS writes to the **CSAuth** service log file if an error occurs.
CSNTpassword	String	0-255 characters. ACS authenticates the password. **Note** If the password field in the database is defined by using a **char** datatype rather than **varchar**, the database might return a string that is 255 characters long; regardless of actual password length. We recommend using the **varchar** datatype for the CHAP password field in your ODBC database.

The CSNTGroup and CSNTacctInfo fields are processed only after a successful authentication. The *CSNTerrorString* file is logged only after a failure (if the result is greater than or equal to 4).

Note If the ODBC database returns data in **recordset** format rather than in parameters, the procedure must return the result fields in the order previously listed.

EAP-TLS Authentication Procedure Input

ACS provides a single value for input to the stored procedure that supports EAP-TLS authentication. The stored procedure should accept the named input value as a variable.

Note Because ACS performs authentication for EAP-TLS, the user password is not an input (Table 12-3).

Table 12-5 EAP-TLS Stored Procedure Input

Field	Type	Explanation
CSNTusername	String	0-64 characters

The input name is for guidance only. A procedure variable that is created from it can have a different name.

EAP-TLS Procedure Output

The stored procedure must return a single row that contains the nonnull fields.

Table 12-4 lists the procedure results that ACS expects as output from stored procedure.

Table 12-6 EAP-TLS Stored Procedure Results

Field	Type	Explanation
CSNTresult	Integer	See Table 12-7 Result Codes.
CSNTgroup	Integer	The ACS group number for authorization. You use **0xFFFFFFFF** to assign the default value. Values other than 0-499 are converted to the default. **Note** The group that is specified in the CSNTgroup field overrides group mapping that is configured for the ODBC external user database.
CSNTacctInfo	String	0-15 characters. A customer-defined string that ACS adds to subsequent account log file entries.
CSNTerrorString	String	0-255 characters. A customer-defined string that ACS writes to the **CSAuth** service log file if an error occurs.

The CSNTGroup and CSNTacctInfo fields are processed only after a successful authentication. The *CSNTerrorString* file is logged only after a failure (if the result is greater than or equal to 4).

Note If the ODBC database returns data in **recordset** format, rather than in parameters, the procedure must return the result fields in the order previously listed.

Result Codes

You can set the result codes that are listed in Table 12-7.

Table 12-7 Result Codes

Result Code	Meaning
0 (zero)	Authentication successful
1	Unknown username
2	Invalid password
3	Unknown username or invalid password
4+	Internal error—authentication not processed

The SQL procedure can decide among 1, 2, or 3 to indicate a failure, depending on how much information that you want the failed authentication log files to include.

A return code of 4 or higher results in an authentication error event. These errors do not increment per-user failed attempt counters. Additionally, error codes are returned to the AAA client so it can distinguish between errors and failures and, if configured to do so, fall back to a backup AAA server.

Successful or failed authentications are not logged; general ACS logging mechanisms apply. In the event of an error (**CSNTresult** equal to or less than 4), the contents of the **CSNTerrorString** are written to the Windows Event Log under the Application Log.

Configuring a System Data Source Name for an ODBC External User Database

On the computer that is running ACS, you must create a system DSN for ACS to communicate with the relational database.

To create a system DSN for use with an ODBC external user database:

Step 1 Using the local administrator account, log in to the computer that is running ACS.

Step 2 In the Windows Control Panel, double-click the **ODBC Data Sources** icon.

Step 3 Choose **Start > Settings > Control Panel > Administrative Tools > Data Sources (ODBC)**.

Tip If the Control Panel is not expanded on the **Start** menu, choose **Start > Settings > Control Panel**, double-click **Administrative Tools**, and then double-click **Data Sources (ODBC)**.

The ODBC Data Source Administrator window appears.

Step 4 Click the **System DSN** tab.

Step 5 Click **Add**.

Step 6 Select the driver that you use with your new DSN, and then click **Finish**.

A dialog box displays fields that require information that is specific to the ODBC driver that you selected.

Step 7 Type a descriptive name for the DSN in the **Data Source Name** box.

Step 8 Complete the other fields that the ODBC driver that you selected requires. These fields might include information such as the IP address of the server on which the ODBC-compliant database runs.

Step 9 Click **OK**.

The name that you assigned to the DSN appears in the System Data Sources list.

Step 10 Close the ODBC Data Source Administrator window and the Windows Control Panel.

The system DSN that ACS will use for communication with the relational database is created on the computer that is running ACS.

Configuring an ODBC External User Database

Creating an ODBC database configuration provides ACS with information that it uses to pass authentication requests to an ODBC-compliant relational database. This information reflects the way that you have implemented your relational database, and does not dictate how your relational database is configured or functions. For information about your relational database, refer to your relational documentation.

Note Before performing this procedure, you should have completed the steps in Preparing to Authenticate Users with an ODBC-Compliant Relational Database, page 12-37.

To configure ACS for ODBC authentication:

Step 1 In the navigation bar, click **External User Databases**.

Step 2 Click **Database Configuration**.

ACS lists all possible external user database types.

Step 3 Click **External ODBC Database**.

Step 4 If you are creating a configuration:

a. Click **Create New Configuration**.

b. Type a name for the new configuration for ODBC authentication in the box provided, or accept the default name in the box.

c. Click **Submit**.

ACS lists the new configuration in the External User Database Configuration table.

Step 5 Click **Configure**.

Step 6 From the System DSN list, select the DSN that is configured to communicate with the ODBC-compliant relational database that you want to use.

Note If you have not configured the computer that is running ACS with a DSN for the relational database, do so before completing these steps. For more information about creating a DSN for ACS ODBC authentication, see Configuring a System Data Source Name for an ODBC External User Database, page 12-44.

Step 7 In the **DSN Username** box, type the username that is required to perform transactions with your ODBC database.

Step 8 In the **DSN Password** box, type the password that is required to perform transactions with your ODBC database.

Step 9 In the **DSN Connection Retries** box, type the number of times that ACS should try to connect to the ODBC database before timing out. The default is 3.

Note If you have connection problems when Windows starts, increase the default value.

Step 10 To change the ODBC worker thread count, in the **ODBC Worker Threads** box, type the number of ODBC worker threads. The maximum thread count is 10. The default is 1.

Note Increase the ODBC worker thread count only if the ODBC driver that you are using is certified thread safe. For example, the Microsoft Access ODBC driver is not thread safe and can cause ACS to become unstable if multiple threads are used. Where possible, ACS queries the driver to find out if it is thread safe. The thread count to use is a factor of how long the DSN takes to execute the procedure and the rate at which authentications are required.

Step 11 From the DSN Procedure Type list, select the type of output that your relational database provides. Different databases return different output:

- **Returns Recordset**—The database returns a raw record set in response to an ODBC query. Microsoft SQL Server responds in this manner.
- **Returns Parameters**—The database returns a set of named parameters in response to an ODBC query. Oracle databases respond in this manner.

Step 12 To support PAP or PEAP (EAP-GTC) authentication with the ODBC database:

a. Check the **Support PAP authentication** check box.

b. In the **PAP SQL Procedure** box, type the name of the PAP SQL procedure routine that runs on the ODBC server. The default value in this box is **CSNTAuthUserPap**. If you named the PAP SQL procedure something else, change this entry to match the name given to the PAP SQL procedure. For more information and an example routine, see Sample Routine for Generating a PAP Authentication SQL Procedure, page 12-39.

Note If you enabled PAP authentication, the PAP authentication SQL procedure must exist on the ODBC database and must have the exact name specified in the **PAP SQL Procedure** box. If it does not, be certain to create it in the ODBC database before attempting to authenticate users against the ODBC database.

Step 13 To support CHAP, MS-CHAP, ARAP, EAP-MD5, or LEAP authentication with the ODBC database:

a. Check the **Support CHAP/MS-CHAP/ARAP Authentication** check box.

b. In the **CHAP SQL Procedure** box, type the name of the CHAP SQL procedure routine on the ODBC server. The default value in this box is **CSNTExtractUserClearTextPw**. If you named the CHAP SQL procedure something else, change this entry to match the name given to the CHAP SQL procedure. For more information and an example routine, see Sample Routine for Generating an SQL CHAP Authentication Procedure, page 12-40.

Note If you enabled CHAP/MS-CHAP/ARAP authentication, the CHAP authentication SQL procedure must exist on the ODBC database and must have the exact name specified in the **PAP SQL Procedure** box. If it does not, be sure to create it in the ODBC database before attempting to authenticate users against the ODBC database.

Step 14 To support EAP-TLS authentication with the ODBC database:

a. Check the **Support EAP-TLS Authentication** check box.

b. In the **EAP-TLS SQL Procedure** box, type the name of the EAP-TLS SQL procedure routine on the ODBC server. The default value in this box is **CSNTFindUser**. If you named the EAP-TLS SQL procedure something else, change this entry to match the name given to the EAP-TLS SQL procedure. For more information and an example routine, see Sample Routine for Generating an EAP-TLS Authentication Procedure, page 12-40.

Note If you enabled EAP-TLS authentication, the EAP-TLS authentication SQL procedure must exist on the ODBC database and must have the exact name specified in the **EAP-TLS SQL Procedure** box. If it does not, be sure to create it in the ODBC database before attempting to authenticate users against the ODBC database.

Step 15 To configure the RADIUS behavior if the database fails:

a. Choose **Send an access reject** for the devices to retry the same server.

b. Choose **Discard the access request** for the devices to try to access other servers.

Note This option is useful in the event of an external ODBC database failure, whereas ACS can deny the authentication (access-reject), or, not respond at all. Conversely, if ACS discards an access-request, the network access device can fail over to another ACS server. A drawback to this approach is that discards can cause excessive network traffic and load on the network access devices as requests continue to travel from network access devices to the ACS servers.

Step 16 Click **Submit**.

ACS saves the ODBC configuration that you created. You can add it to your Unknown User Policy or assign specific user accounts to use this database for authentication. For more information about the Unknown User Policy, see About Unknown User Authentication, page 15-3. For more information about configuring user accounts to authenticate by using this database, see Chapter 6, "User Management."

Downloading a Certificate Database (Solution Engine Only)

Before You Begin

The database must be a Netscape *cert7.db* certificate database file generated by a Netscape web browser. No other filename is supported. For information about generating a Netscape *cert7.db* file, refer to Netscape documentation.

To download a certificate database for a primary or a secondary LDAP server:

Note Downloading a certificate database is a part of the larger process that configures an LDAP external user database. For more information, see Configuring a Generic LDAP External User Database, page 12-31.

Step 1 To access the Download Certificate Database page:

a. Open the LDAP Database Configuration page that contains the information for the LDAP server whose certificate database file you want to download.

Note If you are already on the applicable LDAP Database Configuration page, go to Step b.

b. For the LDAP server whose certificate database file you want to download, click **Download Certificate Database**.

Note ACS lists a primary and secondary LDAP server for each LDAP database configuration. To support secure authentication to both servers, you must download a certificate database file twice, once for the primary LDAP server and once for the secondary LDAP server.

Step 2 In the **FTP Server** box, enter the IP address or hostname of the FTP server. The FTP Server box accepts a maximum of 512 characters.

Note Providing the hostname requires that DNS is correctly operating on your network.

Step 3 In the **Login** box, enter a valid username to enable ACS to access the FTP server. The Login box accepts a maximum of 512 characters.

Step 4 In the **Password** box, enter the password for the username provided in the Login box. The Password box accepts a maximum of 512 characters.

Step 5 In the **Directory** box, enter the path to the Netscape *cert7.db* file. The path is relative to the starting directory at login to the FTP server.

For example, if the Netscape *cert7.db* file is located in `c:\ACS-files\LDAPcertdb` and the user provided in the Login box starts its FTP sessions in `c:\`, you then type `ACS-files\LDAPcertdb`.

The Directory box accepts a maximum of 512 characters.

Step 6 Click **Download**.

ACS downloads the Netscape *cert7.db* file from the FTP server. ACS displays the LDAP Database Configuration page.

LEAP Proxy RADIUS Server Database (Both Platforms)

For ACS-authenticated users who access your network via Cisco Aironet devices, ACS supports authentication with a proxy RADIUS server. For information about the types of authentication that ACS supports with a a proxy RADIUS server, see Authentication Protocol-Database Compatibility, page 1-8. This feature is useful if your own RADIUS-based user database can support MS-CHAP but not LEAP/EAP-FAST. ACS manages the LEAP/EAP-FAST protocol handling and forwards just the MS-CHAP authentication to your server.

ACS uses MS-CHAP Version 1 for LEAP Proxy RADIUS Server authentication. With LEAP Proxy, only the MS-CHAP authentication is proxied to the remote server in a separate RADIUS access request. To manage your proxy RADIUS database, refer to your RADIUS database documentation.

You can use the LEAP proxy RADIUS server authentication to authenticate users against existing Kerberos databases that support MS-CHAP authentication. You can use the LEAP Proxy RADIUS Server database to authenticate users with any third-party RADIUS server that supports MS-CHAP authentication.

Note The third-party RADIUS server must return Microsoft Point-to-Point Encryption (MPPE) keys in the Microsoft RADIUS vendor-specific attribute (VSA) `MS-CHAP-MPPE-Keys (VSA 12)`. If the third-party RADIUS server does not return the MPPE keys, the authentication fails and is logged in the Failed Attempts log.

ACS supports RADIUS-based group specification for users who are authenticated by LEAP Proxy RADIUS Server databases. The RADIUS-based group specification overrides group mapping. For more information, see RADIUS-Based Group Specification, page 16-8.

ACS supports group mapping for unknown users who are authenticated by LEAP Proxy RADIUS Server databases. Group mapping is only applied to an unknown user if RADIUS-based group specification did not occur. For more information about group mapping of users who are authenticated by a LEAP Proxy RADIUS Server database, see Group Mapping by External User Database, page 16-1.

Configuring a LEAP Proxy RADIUS Server External User Database

You should install and configure your proxy RADIUS server before configuring ACS to authenticate users with it. For information about installing the proxy RADIUS server, refer to the documentation that is included with your RADIUS server.

To configure LEAP proxy RADIUS authentication:

Step 1 In the navigation bar, click **External User Databases**.

Step 2 Click **Database Configuration**.

ACS lists all possible external user database types.

Step 3 Click **LEAP Proxy RADIUS Server**.

If no LEAP Proxy RADIUS Server configuration exists, only the Database Configuration Creation table appears. Otherwise, in addition to the Database Configuration Creation table, the External User Database Configuration table appears.

Step 4 If you are creating a configuration:

a. Click **Create New Configuration**.

b. Type a name for the new configuration for the LEAP Proxy RADIUS Server in the box provided, or accept the default name in the box.

c. Click **Submit**.

ACS lists the new configuration in the External User Database Configuration table.

Step 5 Under External User Database Configuration, select the name of the LEAP Proxy RADIUS Server database that you configure.

Note If only one LEAP Proxy RADIUS Server configuration exists, the name of that configuration appears instead of the list. Proceed to Step 6.

Step 6 Click **Configure**.

Step 7 In the following boxes, type the required information:

- **Primary Server Name/IP**—IP address of the primary proxy RADIUS server.
- **Secondary Server Name/IP**—IP address of the secondary proxy RADIUS server.
- **Shared Secret**—The shared secret of the proxy RADIUS server. This must be identical to the shared secret with which the proxy RADIUS server is configured.

- **Authentication Port**—The UDP port over which the proxy RADIUS server conducts authentication sessions. If the LEAP Proxy RADIUS server is installed on the same Windows server as ACS, this port should not be the same port that ACS uses for RADIUS authentication. For more information about the ports that ACS uses for RADIUS, see RADIUS, page 1-4.
- **Timeout (seconds):**—The number of seconds that ACS waits before sending notification to the user that the authentication attempt has timed out.
- **Retries**—The number of authentication attempts ACS makes before failing over to the secondary proxy RADIUS server.
- **Failback Retry Delay (minutes)**—The number of minutes after which ACS attempts authentications by using a failed primary proxy RADIUS server.

Note If the primary and the secondary servers fail, ACS alternates between the servers until one responds.

Step 8 Click **Submit**.

ACS saves the proxy RADIUS token server database configuration that you created. You can add it to your Unknown User Policy or assign specific user accounts to use this database for authentication. For more information about the Unknown User Policy, see About Unknown User Authentication, page 15-3. For more information about configuring user accounts to authenticate by using this database, see Chapter 6, "User Management."

Token Server User Databases

ACS supports the use of token servers for the increased security that one-time passwords (OTPs) provide.

This section contains:

About Token Servers and ACS

For information about the types of authentication that ACS supports with token servers, see Authentication Protocol-Database Compatibility, page 1-8.

Requests from the AAA client are first sent to ACS. If ACS has been configured to authenticate against a token server and finds the username, it forwards the authentication request to the token server. If it does not find the username, ACS checks the database that is configured to authenticate unknown users. If the request for authentication is passed, the appropriate authorizations are forwarded to the AAA client along with the approved authentication. ACS then maintains the accounting information.

ACS for Windows Only

ACS acts as a client to the token server. For all token servers except RSA SecurID, ACS acts as a client by using the RADIUS interface of the token server. For more information about ACS support of token servers with a RADIUS interface, see RADIUS-Enabled Token Servers, page 12-51.

For RSA SecurID, ACS uses an RSA proprietary API. For more information about ACS support of RSA SecurID token servers, see Using RSA Token-Card Client Software, page 12-54.

Solution Engine Only

ACS acts as a client to the token server. For all token servers, ACS acts as a client by using the RADIUS interface of the token server. For more information about ACS support of token servers with a RADIUS interface, see RADIUS-Enabled Token Servers, page 12-51.

Token Servers and ISDN

ACS supports token caching for ISDN terminal adapters and routers. One inconvenience of using token cards for OTP authentication with ISDN is that each B channel requires its own OTP. Therefore, a user must enter at least 2 OTPs, plus any other login passwords, such as those for Windows networking. If the terminal adapter supports the ability to turn on and off the second B channel, users might have to enter many OTPs each time the second B channel comes into service.

ACS caches the token to help make the OTPs easier for users. Therefore, if a token card is being used to authenticate a user on the first B channel, you can set a specified period during which the second B channel can come into service without requiring the user to enter another OTP. To lessen the risk of unauthorized access to the second B channel, you can limit the time that the second B channel is up. Furthermore, you can configure the second B channel to use the CHAP password that is specified during the first login to further lessen the chance of a security problem. When the first B channel is dropped, the cached token is erased.

RADIUS-Enabled Token Servers

This section describes support for token servers that provide a standard RADIUS interface.

This section contains:

About RADIUS-Enabled Token Servers

ACS supports token servers by using the RADIUS server that is built into the token server. Rather than using a vendor-proprietary API, ACS sends standard RADIUS authentication requests to the RADIUS authentication port on the token server. This feature enables ACS to support any IETF RFC 2865-compliant token server.

You can create multiple instances of RADIUS token servers. For information about configuring ACS to authenticate users with one of these token servers, see Configuring a RADIUS Token Server External User Database, page 12-52.

ACS provides a means for specifying a user group assignment in the RADIUS response from the RADIUS-enabled token server. Group specification always takes precedence over group mapping. For more information, see RADIUS-Based Group Specification, page 16-8.

ACS also supports mapping users who are authenticated by a RADIUS-enabled token server to a single group. Group mapping only occurs if group specification does not occur. For more information, see Group Mapping by External User Database, page 16-1.

Token Server RADIUS Authentication Request and Response Contents

When ACS forwards an authentication request to a RADIUS-enabled token server, the RADIUS authentication request contains the following attributes:

- `User-Name` (RADIUS attribute 1)
- `User-Password` (RADIUS attribute 2)
- `NAS-IP-Address` (RADIUS attribute 4)
- `NAS-Port` (RADIUS attribute 5)
- `NAS-Identifier` (RADIUS attribute 32)

ACS expects to receive one of the following three responses:

- **access-accept**—No attributes are required; however, the response can indicate the ACS group to which the user should be assigned. For more information, see RADIUS-Based Group Specification, page 16-8.
- **access-reject**—No attributes required.
- **access-challenge**—Attributes required, per IETF RFC, are:
  - `State` (RADIUS attribute 24)
  - `Reply-Message` (RADIUS attribute 18)

Configuring a RADIUS Token Server External User Database

Use this procedure to configure RADIUS Token Server external user databases.

Before You Begin

You should install and configure your RADIUS token server before configuring ACS to authenticate users with it. For information about installing the RADIUS token server, refer to the documentation included with your token server.

To configure ACS to authenticate users with a RADIUS Token Sever:

Step 1 In the navigation bar, click **External User Databases**.

Step 2 Click **Database Configuration**.

ACS lists all possible external user database types.

Step 3 Click **RADIUS Token Server**.

The Database Configuration Creation table appears. If at least one RADIUS token server configuration exists, the External User Database Configuration table also appears.

Step 4 If you are creating a configuration:

a. Click **Create New Configuration**.

b. Type a name for the new configuration for the RADIUS-enabled token server in the box provided, or accept the default name in the box.

c. Click **Submit**.

ACS lists the new configuration in the External User Database Configuration table.

Step 5 Under External User Database Configuration, select the name of the RADIUS-enabled token server that you configure.

Note If only one RADIUS-enabled token server configuration exists, the name of that configuration appears instead of the list. Proceed to Step 6.

Step 6 Click **Configure**.

Step 7 In the RADIUS Configuration table, type the required information in the following boxes:

- **Primary Server Name/IP**—The hostname or IP address of the primary RADIUS token server. If you provide the hostname, the hostname must be resolvable by DNS.
- **Secondary Server Name/IP**—The hostname or IP address of the secondary RADIUS token server. If you provide the hostname, the hostname must be resolvable by DNS.
- **Shared Secret**—The shared secret of the RADIUS server. This must be identical to the shared secret with which the RADIUS token server is configured.
- **Authentication Port**—The UDP port over which the RADIUS server conducts authentication sessions. If the RADIUS token server is installed on the same Windows server as ACS, this port should not be the same port that ACS uses for RADIUS authentication. For more information about the ports that ACS uses for RADIUS, see RADIUS, page 1-4.

  Note For ACS to send RADIUS OTP messages to a RADIUS-enabled token server, you must ensure that gateway devices between the RADIUS-enabled token server and ACS allow communication over the UDP port that is specified in the Authentication Port box.

- **Timeout (seconds):**—The number of seconds that ACS waits for a response from the RADIUS token server before retrying the authentication request.
- **Retries**—The number of authentication attempts that ACS makes before failing over to the secondary RADIUS token server.
- **Failback Retry Delay (minutes)**—The number of minutes that ACS sends authentication requests to the secondary server when the primary server has failed. When this duration elapses, ACS reverts to sending authentication requests to the primary server.

  Note If the primary and the secondary servers fail, ACS alternates between the servers until one responds.

Step 8 If you want to support token users who perform a shell login to a TACACS+ AAA client, you must configure the options in the TACACS+ Shell Configuration table. Perform one of the following procedures:

- If you want ACS to present a custom prompt for tokens, select **Static (sync and async tokens)**, and then type in the **Prompt** box the prompt that ACS will present.

  For example, if you type **Enter your PassGo token** in the **Prompt** box, users receive an **Enter your PassGo token** prompt rather than a password prompt.

Note If some tokens that are submitted to this server are synchronous tokens, you must click the **Static (sync and async tokens)** option.

- If you want ACS to send the token server a password to trigger a challenge, select **From Token Server (async tokens only)**, and then, in the **Password** box, type the password that ACS will forward to the token server.

  For example, if the token server requires the string **challengeme** in order to evoke a challenge, you should type **challengeme** in the **Password** box. Users will receive a username prompt and a challenge prompt.

Tip Most token servers accept a blank password as the trigger to send a challenge prompt.

Note You should only click the **From Token Server (async tokens only)** option if all tokens that are submitted to this token server are asynchronous tokens.

Step 9 Click **Submit**.

ACS saves the RADIUS token server database configuration that you created. You can add it to your Unknown User Policy or assign specific user accounts to use this database for authentication. For more information about the Unknown User Policy, see About Unknown User Authentication, page 15-3. For more information about configuring user accounts to authenticate by using this database, see Chapter 6, "User Management."

Using RSA Token-Card Client Software

ACS supports mapping users who are authenticated by a RSA token server to a single group. For more information, see Group Mapping by External User Database, page 16-1.

ACS supports PPP (ISDN and async) and Telnet for RSA SecurID token servers by acting as a token-card client to the RSA SecurID token server. To use this client you must install the RSA token-card client software on the computer that is running ACS. The following procedure includes the steps that you follow to install the RSA client correctly on the computer that is running ACS.

ACS supports the RSA SecurID token server custom interface for authentication of users. You can create only one RSA SecurID configuration within ACS.

ACS for Windows

Before You Begin

You should install and configure your RSA SecurID token server before configuring ACS to authenticate users with it. For information about installing the RSA SecurID server, refer to the documentation for your token server.

Ensure that you have the applicable RSA ACE Client.

To configure ACS to authenticate users with an RSA token server:

Step 1 Install the RSA client on the computer that is running ACS:

a. With a username that has administrative privileges, log in to the computer that is running ACS.

b. Run the Setup program of the ACE Client software, following the setup instructions that RSA provides.

Note Do not restart Windows when installation is complete.

c. Locate the ACE Server data directory, for example, `/sdi/ace/data`.

d. Get the file named `sdconf.rec` and place it in the following Windows directory: `%SystemRoot%\system32`.

For example:

`\winnt\system32`

e. Ensure that the ACE server hostname is in the Windows local host file:

`\`*Windows directory*`\system32\drivers\etc\hosts`

f. Restart the computer that is running ACS.

g. Verify connectivity by running the Test Authentication function of your ACE client application. You can run this from the Control Panel.

Step 2 In the navigation bar, click **External User Databases**.

Step 3 Click **Database Configuration**.

ACS lists all possible external user database types.

Step 4 Click **RSA SecurID Token Server**.

If no RSA SecurID token server configuration exists, the Database Configuration Creation table appears. Otherwise, the External User Database Configuration page appears.

Step 5 If you are creating a configuration:

a. Click **Create New Configuration**.

b. Type a name for the new configuration for the RSA SecurID token server in the box provided, or accept the default name in the box.

c. Click **Submit**.

ACS lists the new configuration in the External User Database Configuration table.

Step 6 Click **Configure**.

ACS displays the name of the token server and the path to the authenticator dynamic link library (DLL). This information confirms that ACS can contact the RSA client. You can add the RSA SecurID external user database to your Unknown User Policy or assign specific user accounts to use this database for authentication. For more information about the Unknown User Policy, see About Unknown User Authentication, page 15-3. For more information about configuring user accounts to authenticate by using this database, see Chapter 6, "User Management."

ACS for Solution Engine

Step 1 In the navigation bar, click **External User Databases**.

Step 2 Click **Database Configuration**.

ACS lists all possible external user database types.

Step 3 Click **RSA SecurID Token Server**.

The CiscoSecure ACS to RSA SecurID Configuration page appears.

Step 4 Select **Upload sdconf.rec** to upload the token server file from the ACE Server data directory.

The FTP Setup Page appears. Enter the:

a. **FTP Server** address.

b. **Login** name.

c. **Password**.

d. **Directory** where the *sdconf.rec* file is located.

e. **Decryption Password**.

Note The decryption password must exactly match the password that you specified in the Encryption Password box for the FTP Server.

f. Click **Submit**.

Step 5 Choose **Purge Node Secret** to delete any existing configuration settings.

ACS lists the new configuration in the External User Database Configuration table.

RSA Authentication with LDAP Group Mapping

You can perform authentication with RSA in native mode and group mapping by using the LDAP group mapping configuration. Authorization is controlled based on the user's LDAP group membership. When RSA native mode authentication succeeds, group mapping is performed with LDAP. The user's group is applied based on the group mapping configuration.

Note Before you configure RSA Authentication with LDAP Group Mapping, be certain that you have the correct installation or configuration of the third-party DLLs required to support this type of external database.

ACS for Windows

To configure RSA authentication with LDAP Group Mapping:

Step 1 Install the RSA client for windows.

Note Cisco recommends that you do not install the RSA client on a local PC.

Step 2 Copy the *sdconf.rec* file into the */system32* directory.

Step 3 Restart **CSAuth** and **CSAdmin**.

Step 4 In the navigation bar, click **External User Databases**.

Step 5 Click **Database Configuration**.

ACS lists all possible external user database types.

Step 6 Click **RSA SecurID Token and LDAP Group Mapping**.

The External Database Configuration page appears.

Step 7 Click **Configure.**

Solution Engine Only

You need to upload the RSA client DLL to the Solution Engine image to configure RSA authentication with LDAP Group Mapping. The *sdconf.rec* file interacts with the RSA ACE Server through the *aceclnt.dll* interface. After you upload the dll file, you can purge the node secret. Purging the node secret is useful when configuration changes are made in the RSA server.

The RSA server client software version. 6.1 [*aceclnt.dll*] is included in appliance image. Ther server client software allows authentication via the RSA Native token card without installing the RSA client software. You can make changes to the *sdconf.rec* file and the node secret through the ACS web interface. RSA client software is unnecessary.

Note ACS assumes the delivery of the node secret is set to automatic in the RSA server.

To configure RSA Authentication with LDAP Group Mapping:

Step 1 In the navigation bar, click **External User Databases**.

Step 2 Click **Database Configuration**.

ACS lists all possible external user database types.

Step 3 Click **RSA SecurID Token and LDAP Group Mapping**.

The External Database Configuration page appears.

Step 4 Click **Configure.**

Step 5 Click **Configure Native RSA.**

Step 6 Choose **Upload sdconf.rec** to upload the token server file from the ACE Server data directory.

The FTP Setup Page appears. Enter the:

a. **FTP Server** address.

b. **Login** name.

c. **Password**.

d. **Directory** where the *sdconf.rec* file is located.

e. **Decryption Password**.

Note The decryption password must exactly match the password that you specified in the Encryption Password box for the FTP Server.

Step 7 Chose **Purge Node Secret** to delete any existing configuration settings.

Step 8 Update the port mapper.

Deleting an External User Database Configuration

If you no longer need a particular external user database configuration, you can delete it from ACS.

To delete an external user database configuration:

Step 1 In the navigation bar, click **External User Databases**.

Step 2 Click **Database Configuration**.

ACS lists all possible external user database types.

Step 3 Click the external user database type for which you want to delete a configuration.

The External User Database Configuration table appears.

Step 4 If a list appears in the External User Database Configuration table, select the configuration to delete. Otherwise, proceed to Step 5.

Step 5 Click **Delete**.

A confirmation dialog box appears.

Step 6 Click **OK** to confirm that you want to delete the selected external user database configuration.

The external user database configuration that you selected is deleted from ACS.