Upon completion of this chapter, you will be able to perform the following tasks:

- Describe the Features and Architecture of Cisco Secure ACS 3.0 for Windows 2000/ NT Servers (Cisco Secure ACS for Windows)
- Configure Cisco Secure ACS for Windows to Perform AAA Functions
- Describe the Features and Architecture of Cisco Secure ACS 2.3 for UNIX
- Configure the Perimeter Router to Enable AAA Processes to Use a TACACS Remote Service

CHAPTER 3

Advanced AAA Security for Cisco Router Networks

This chapter covers Cisco Secure ACS 3.0 for Windows 2000/NT Servers (Cisco Secure ACS for Windows) and Cisco Secure ACS for UNIX (Solaris). The Windows 2000 version has the most coverage in this chapter. The configuration of the Windows 2000 product is covered as a high-level overview. This chapter also covers the security services of TACACS+, RADIUS, and Kerberos.

This chapter includes the following topics:

- Introduction to Cisco Secure ACS for Windows
- Product overview: Cisco Secure ACS for Windows
- Product overview: Cisco Secure ACS for UNIX (Solaris)
- Installing Cisco Secure ACS for Windows
- Administering and troubleshooting Cisco Secure ACS for Windows
- TACACS+ overview and configuration
- Verifying TACACS+
- RADIUS configuration overview
- Kerberos overview

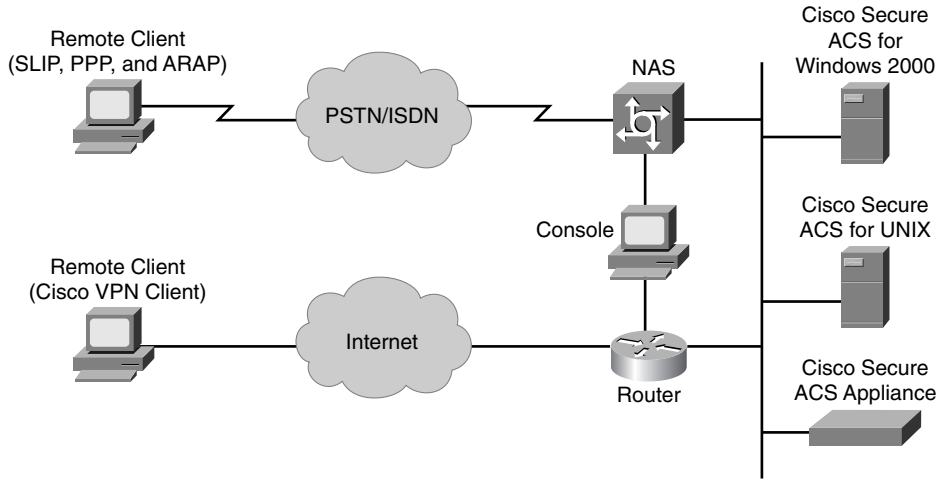
Cisco Secure ACS Introduction

This section presents an introduction to the Cisco Secure ACS offerings shown in Figure 3-1, including the following products:

- Cisco Secure ACS for Windows
- Cisco Secure ACS for UNIX

The next three sections discuss each of the Cisco Secure ACS product offerings.

Figure 3-1 *Cisco Secure ACS Servers*



Cisco Secure ACS for Windows

Cisco Secure ACS for Windows is a network security software application that helps you control access to the campus network, dial-in access, and the Internet. Cisco Secure ACS for Windows operates as Windows NT or Windows 2000 services and controls authentication, authorization, and accounting (AAA) of users accessing the network.

This section presents an overview of the product and prepares you to install and configure Cisco Secure ACS for Windows.

Cisco Secure ACS for Windows provides AAA services to network devices that function as AAA clients, such as routers, network access servers, PIX Firewalls, and VPN 3000 Concentrators. An AAA client is any device that provides AAA client functionality and uses one of the AAA protocols supported by Cisco Secure ACS for Windows. It also supports third-party devices that can be configured to use TACACS+ or RADIUS protocols. Cisco Secure ACS for Windows treats all such devices as AAA clients. Cisco Secure ACS for Windows uses the TACACS+ and RADIUS protocols to provide AAA services that ensure a secure environment.

Cisco Secure ACS for Windows helps to centralize access control and accounting, in addition to router and switch access management. With Cisco Secure ACS for Windows, network administrators can quickly administer accounts and globally change levels of service offerings for entire groups of users. Although the use of an external user database is optional, support for many popular user repository implementations enables companies to use the working knowledge gained from and the investment already made in building the corporate user repositories.

Cisco Secure ACS for Windows is an easy-to-use ACS that is simple to install and administer. It runs on the popular Windows NT Server 4.0 (SP5 or 6) or 2000 Server (SP 1 or 2) Microsoft

operating systems. The Cisco Secure ACS for Windows administration interface is viewed using supported web browsers, making it easy to administer.

Cisco Secure ACS for Windows authenticates usernames and passwords against the Windows NT or Windows 2000 user database, the Cisco Secure ACS for Windows database, a token server database, or Novell NetWare Directory Service (NDS).

Different levels of security can be used with Cisco Secure ACS for Windows for different requirements. The basic user-to-network security level is Password Authentication Protocol (PAP). Although it does not represent the highest form of encrypted security, PAP does offer convenience and simplicity for the client. PAP allows authentication against the Windows NT or Windows 2000 database. With this configuration, users need to log in only a single time. Challenge Handshake Authentication Protocol (CHAP) allows a higher level of security for encrypting passwords when communicating from a client to the network access server. You can use CHAP with the Cisco Secure ACS for Windows user database. Microsoft CHAP (MS-CHAP) is a version of CHAP that was developed by Microsoft to work more closely with the Microsoft Windows operating system.

PAP, CHAP, and MS-CHAP are authentication protocols that are used to encrypt passwords. However, each protocol provides a different level of security:

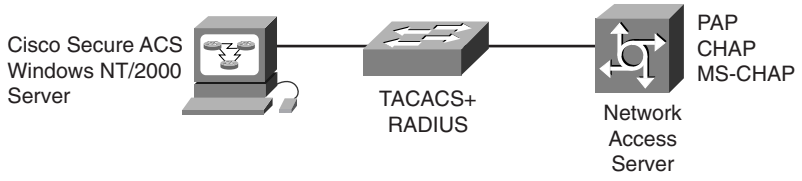
- **PAP**—Uses clear-text passwords and is the least sophisticated authentication protocol. If you are using the Windows NT or Windows 2000 user database to authenticate users, you must use PAP password encryption.
- **CHAP**—Uses a challenge-response mechanism with one-way encryption on the response. CHAP lets Cisco Secure ACS for Windows negotiate downward from the most secure to the least secure encryption mechanism, and it protects passwords transmitted in the process. CHAP passwords are reusable. If you are using the Cisco Secure ACS for Windows user database for authentication, you can use either PAP or CHAP.
- **MS-CHAP**—Cisco Secure ACS for Windows supports MS-CHAP for user authentication. The differences between MS-CHAP and standard CHAP follow:
  - The MS-CHAP response packet is in a format that is compatible with Microsoft Windows and LAN Manager 2.x. The MS-CHAP format does not require the authenticator to store a clear-text or reversibly encrypted password.
  - MS-CHAP provides an authenticator-controlled authentication retry mechanism.
  - MS-CHAP version 2 provides additional failure codes in the Failure Packet Message field.

General Features

Cisco Secure ACS for Windows, depicted in Figure 3-2, has the following general features:

- Simultaneous TACACS+ and RADIUS support between Cisco Secure ACS for Windows and the NAS or perimeter router

Figure 3-2 *General Features*



- Windows NT or Windows 2000 user database support:
  - Leverages and consolidates Windows NT or Windows 2000 username and password management
  - Enables single login to network and Windows NT or Windows 2000 domains
  - Runs on Windows NT or Windows 2000 standalone, primary domain controller (PDC), and backup domain controller (BDC) server configurations

NOTE Although Cisco Secure ACS for Windows can function on a BDC or PDC, Cisco SAFE practices recommend placing the application on a standalone server to separate the services of one authentication server from another. Doing so will improve the security posture by making it potentially more difficult for an attacker to penetrate multiple devices.

- Supports the use of external user database:
  - External token card servers
  - NDS
  - ACS databases
  - Others
- Supports the following, leading authentication protocols:
  - ASCII/PAP
  - CHAP
  - MS-CHAP
  - LEAP
  - EAP-CHAP
  - EAP-TLS
  - ARAP
- Network access server callback feature supported for increased security

AAA Services

Cisco Secure ACS for Windows supports the following AAA features:

- TACACS+ support for:
  - Access lists, named or numbered
  - Time-of-day and day-of-week access restrictions
  - AppleTalk Remote Access (ARA) support
  - Enable-privilege support levels
  - Authentication to an LDAP server
  - One-time password (OTP) for enable passwords
- RADIUS versions:
  - IETF RADIUS
  - Cisco RADIUS Attribute-Value pairs
  - Proprietary RADIUS extensions (Lucent)
- Single TACACS+/RADIUS database for simultaneous support

Other AAA product features are as follows:

- VPN and Virtual Private Dialup Network (VPDN) support is available at the origination and termination of VPN (L2F) tunnels
- User restrictions can be based on remote address Calling Line Identification (CLID)
- Can disable an account on a specific date or after "n" failed attempts

Administration Features

Cisco Secure ACS for Windows has many user-friendly administration features, such as:

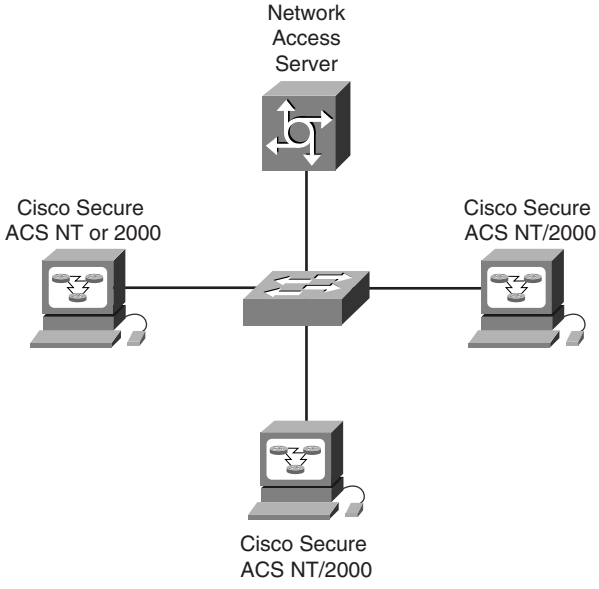
- Browser-based GUI allows management from a web browser via a LAN or by dialing in. Simplifies and distributes configuration for ACS, user profiles, and group profiles:
  - Help and online documentation is included for quick problem solving and access from a web browser (The browser does not use SSL; it uses CSAdmin running as a Windows service to provide the website for ACS)
  - Permits group administration of users for maximum flexibility and to facilitate enforcement and changes of security policies
  - Remote administration can be permitted/denied by using a unique administration username/password
  - Remote administrator session has a timeout value
  - Can view a logged-in user list for a quick view of who is connected
- Creates separate TACACS+ and RADIUS files stored in comma-separated value (CSV) spreadsheet format for easy import into databases and spreadsheet applications

- Has import utility to rapidly import a large number of users
- Hash-indexed flat-file database support for high-speed transaction processing (Cisco Secure ACS for Windows user database)

Distributed System Features

As shown in Figure 3-3, Cisco Secure ACS for Windows can be used in a distributed system. Multiple Cisco Secure ACS for Windows servers and AAA servers can be configured to communicate with one another as masters, clients, or peers. Cisco Secure ACS for Windows also recognizes network access restrictions of other Cisco Secure ACS for Windows servers on the distributed network.

Figure 3-3 *Distributed System Features*



Cisco Secure ACS for Windows allows you to use powerful features, such as:

- **Authentication forwarding**—Authentication forwarding allows the Cisco Secure ACS for Windows to automatically forward an authentication request from a network access server to another Cisco Secure ACS for Windows. After authentication, authorization privileges are applied to the network access server for that user authentication.
- **Fallback on failed connection**—You can configure the order in which Cisco Secure ACS for Windows checks the remote Cisco Secure ACS for Windows servers if the network connection to the primary Cisco Secure ACS for Windows server fails. If an authentication request cannot be sent to the first listed server, the next listed server is checked, in order down the list, until a Cisco Secure ACS for Windows server handles the authentication. If Cisco Secure ACS for Windows cannot connect to any of the servers on the list, authentication fails.

- **Remote and centralized accounting**—Cisco Secure ACS for Windows can be configured to point to a centralized Cisco Secure ACS for Windows that is used as the accounting server. The centralized Cisco Secure ACS for Windows will still have all the capabilities that a Cisco Secure ACS for Windows server has, with the addition of being a central repository for all accounting logs that are sent.

External Database Support

You can configure Cisco Secure ACS for Windows to forward authentication of users to one or more external user databases. Support for external user databases means that Cisco Secure ACS for Windows does not require that you create duplicate user entries in the Cisco Secure user database. Users can be authenticated using any of the following:

- Windows NT or Windows 2000 user database
- LDAP
- NDS
- Open Database Connectivity (ODBC)—compliant relational databases
- LEAP Proxy RADIUS servers
- Symantec (AXENT) Defender token servers
- Secure Computing SafeWord token servers
- RSA SecurID token servers
- RADIUS-based token servers, including:
  - ActivCard token servers
  - CRYPTOCard token servers
  - VASCO token servers
  - Generic RADIUS token servers

Regardless of which database is used to authenticate users, the Cisco Secure user database, internal to Cisco Secure ACS for Windows, authorizes requested network services.

Cisco Secure ACS for Windows requires an application program interface (API) for third-party authentication support. Cisco Secure ACS for Windows communicates with the external user database using the API. For Windows NT or Windows 2000, Generic LDAP, and Novell NDS authentication, the API for the external authentication is local to the Cisco Secure ACS for Windows system and is provided by the local operating system. In these cases, no further components are required.

In the case of ODBC authentication sources, in addition to the Windows ODBC interface, the third-party ODBC driver must be installed on the Cisco Secure ACS for Windows server.

To communicate with each traditional token server, you must have software components provided by the OTP vendors installed, in addition to the Cisco Secure ACS for Windows components. You must also specify in User Setup that a token card server be used.

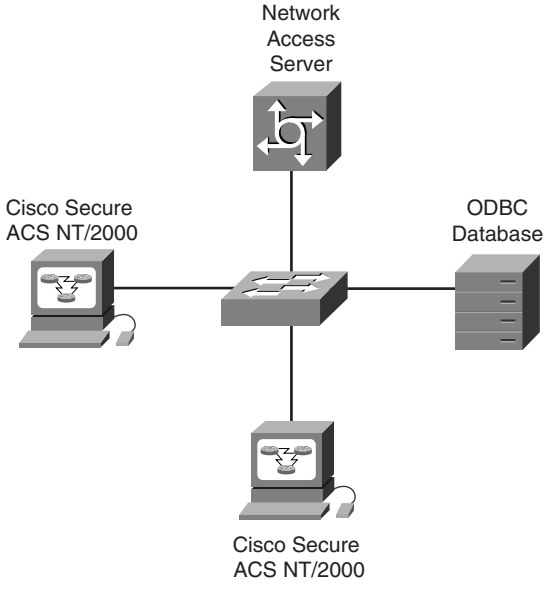
For RADIUS-based token servers, such as those from ActivCard, CRYPTOCard, and VASCO, the standard RADIUS interface serves as the third-party API.

Database Management Features

Two utilities, Database Replication and Relational Database Management System (RDBMS) Synchronization, are provided with Cisco Secure ACS for Windows. These utilities help automate the process of keeping your Cisco Secure ACS for Windows database and network configuration current. A third utility, CSUtil.exe, allows for database backup and restore functionality.

Figure 3-4 shows a typical installation that can support Database Replication, RDBMS Synchronization, and ODBC import. These three topics will be discussed in the following sections.

Figure 3-4 *Database Management Features*



Database Replication

Database Replication is a powerful feature that is designed to simplify the construction of a fault-tolerant AAA service environment based on the Cisco Secure ACS for Windows. The primary purpose of Database Replication is to provide the facility to replicate various parts of the setup on a Cisco Secure ACS for Windows master server to one or more Cisco Secure ACS for Windows client systems, allowing the administrator to automate the creation of mirror systems. These mirror systems can then be used to provide server redundancy as fallback or secondary servers to support fault-tolerant operation if the master or primary system fails.

Do not confuse Database Replication with database/system backup. Database Replication is not a complete replacement for database backup. You should still have a reliable database backup strategy to ensure data integrity.

RDBMS Synchronization

RDBMS Synchronization is an integration feature designed to simplify integration of Cisco Secure ACS for Windows with a third-party RDBMS application. RDBMS Synchronization automates synchronization with an SQL, Oracle, or Sybase RDBMS data source by providing the following functions:

- Specification of an ODBC data source to use for synchronization data that is shared by Cisco Secure ACS for Windows and the other RDBMS application and to provide control of the Cisco Secure ACS for Windows updates to an external application
- Control of the timing of the import/synchronization process, including the creation of schedules
- Control of which systems are to be synchronized

The RDBMS Synchronization feature has two components:

- **CSDBSync**—CSDBSync is a dedicated Windows NT or Windows 2000 service that performs automated user and group account management services for Cisco Secure ACS for Windows.
- **ODBC data store (table)**—This table specifies the record format. Each record holds user or group information that corresponds with the data stored for each user in the Cisco Secure ACS for Windows database. Additionally, each record contains other fields, including an action code for the record. Any application can write to this table, and CSDBSync reads from it and takes actions on each record that it finds in the table (for example, add user, delete user, and so on) as determined by the action code.

ODBC Import Definitions

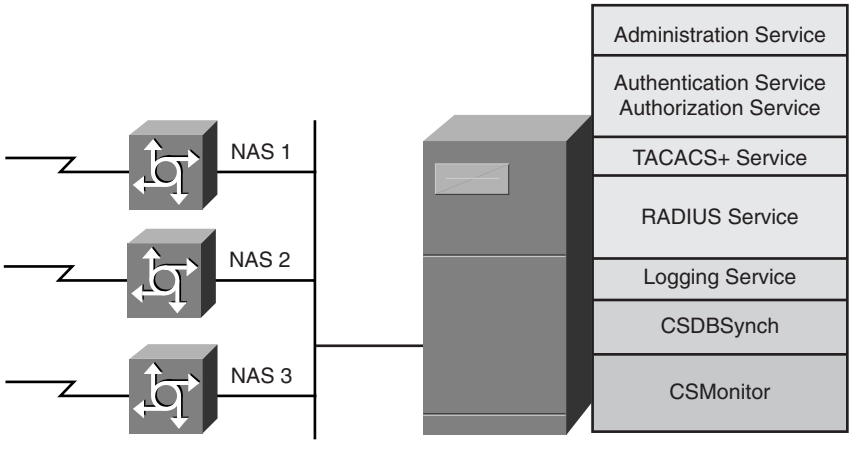
Cisco Secure ACS for Windows supports the import of data from ODBC-compliant databases, such as Microsoft Access or Oracle. Importing is done with a single table to import user/group information into one or more ACS servers.

The CSAccupdate service processes the table and updates local/remote ACS installations according to its configuration.

Windows Architecture

Cisco Secure ACS for Windows provides AAA services to multiple NASs or perimeter routers. It includes seven service modules, as shown in Figure 3-5.

Figure 3-5 *Windows Architecture*



Each module can be started and stopped individually from within the Microsoft Service Control Panel or as a group from within the Cisco Secure ACS for Windows browser interface.

Cisco Secure ACS for Windows installs the following Windows services on your server:

- **Administration service (CSAdmin)**—Cisco Secure ACS for Windows is equipped with its own internal web server. After Cisco Secure ACS for Windows is installed, you must configure it from its HTML/Java interface, which requires CSAdmin to always be enabled.
- **Authentication and authorization service (CSAuth)**—The primary responsibility of Cisco Secure ACS for Windows is the authentication and authorization of requests from devices to permit or deny access to a specified user. CSAuth is the service that is responsible for determining whether access should be granted and for defining the privileges associated with that user. CSAuth is the database manager.
- **TACACS service (CSTacacs) and RADIUS service (CSRadius)**—These services communicate between the CSAuth module and the access device that is requesting the authentication and authorization services. CSTacacs is used to communicate with TACACS+ devices and CSRadius is used to communicate with RADIUS devices. Both services can run simultaneously. When only one security protocol is used, only the respective service needs to be running.
- **Logging service (CSLog)**—CSLog is the service that is used to capture and place logging information. CSLog gathers data from the TACACS+ or RADIUS packet and CSAuth and manipulates the data to be put into the CSV files. The CSV files are created daily starting at midnight.
- **CSDBSync service**—This service performs automated user and group account management services for Cisco Secure ACS for Windows. CSDBSync is the service that is used to synchronize the Cisco Secure ACS for Windows database with third-party RDBMSs and is an alternative to using the ODBC dynamic link library (DLL). Starting

with Version 2.4, CSDBSync synchronizes AAA client, AAA server, network device groups (NDGs), and Proxy Table information with data from an external relational database.

- **CSMon**—CSMon is the Cisco Secure ACS for Windows self-monitoring and self-correcting service. CSMon works for both TACACS+ and RADIUS and automatically detects which protocols are in use. CSMon facilitates minimum downtime in a remote access network environment by performing four basic activities:
  - **Monitoring**—Monitors the overall status of Cisco Secure ACS for Windows and the host server on which it is running. CSMon monitors the generic host system state, application-specific performance, and system resource consumption by Cisco Secure ACS for Windows.
  - **Recording**—Records and reports all exceptions to the CSMon Log or the Windows Event Log.
  - **Notification**—Alerts the administrator to potential problems and real events regarding Cisco Secure ACS for Windows and records the activity. CSMon can be configured to send messages concerning exception events, responses, and the outcomes of response actions.
  - **Response**—Attempts to automatically and intelligently fix detected problems. CSMon can respond to warning events and failure events by taking either predefined actions or customer-definable actions.

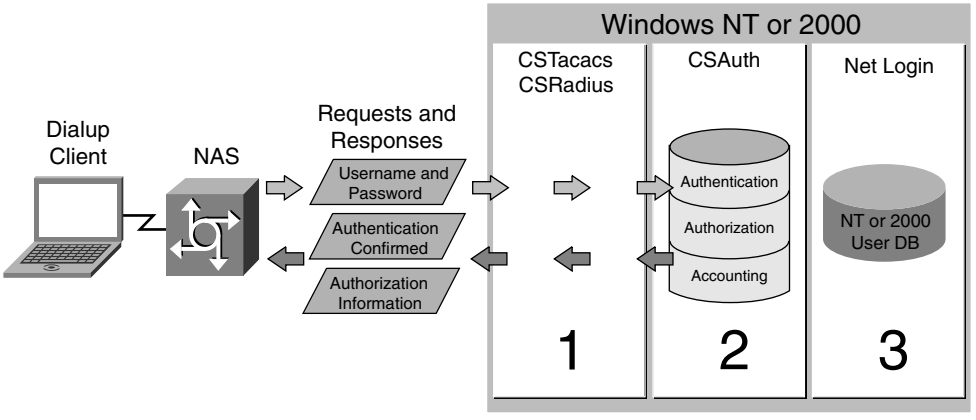
Using the ACS Database

Using either the TACACS+ or the RADIUS protocol, the network access server directs all dial-in user access requests to Cisco Secure ACS for Windows for authentication and authorization of privileges, which verifies the username and password. Cisco Secure ACS for Windows then returns a success or failure response to the network access server, which permits or denies user access. When the user has been authenticated, Cisco Secure ACS for Windows sends a set of authorization attributes to the network access server, and then the accounting functions take effect.

Referring to the numbers shown in Figure 3-6, when the Cisco Secure ACS for Windows user database is selected, the following service and database interaction occurs:

1. TACACS+ or RADIUS service directs the request to the Cisco Secure ACS Authentication and Authorization Windows NT or Windows 2000 service.
2. The request is authenticated against the Cisco Secure ACS for Windows user database, associated authorizations are assigned, and accounting information is logged to the Cisco Secure ACS Logging service.
3. The Windows NT or Windows 2000 user database does not authenticate the user to permit dial. The user must log in to Windows NT or Windows 2000 once the dialup AAA process is complete.

Figure 3-6 *Using the ACS Database*



Cisco Secure ACS for Windows uses a built-in user database that is a hash-indexed flat file. This type of file is not searched from the top of a text file as typically associated with the term flat file, but instead is indexed like a database. The hash-indexed flat file builds an index and tree structure so that searches can occur exponentially, which enables the Cisco Secure ACS for Windows user database to rapidly authenticate users.

Using the Cisco Secure ACS for Windows user database requires you to manually enter the usernames. However, after the usernames exist in the Cisco Secure ACS for Windows user database, administration is easier than using the Windows NT or Windows 2000 user database. The Cisco Secure ACS for Windows user database supports authentication for PAP, CHAP, and MS-CHAP.

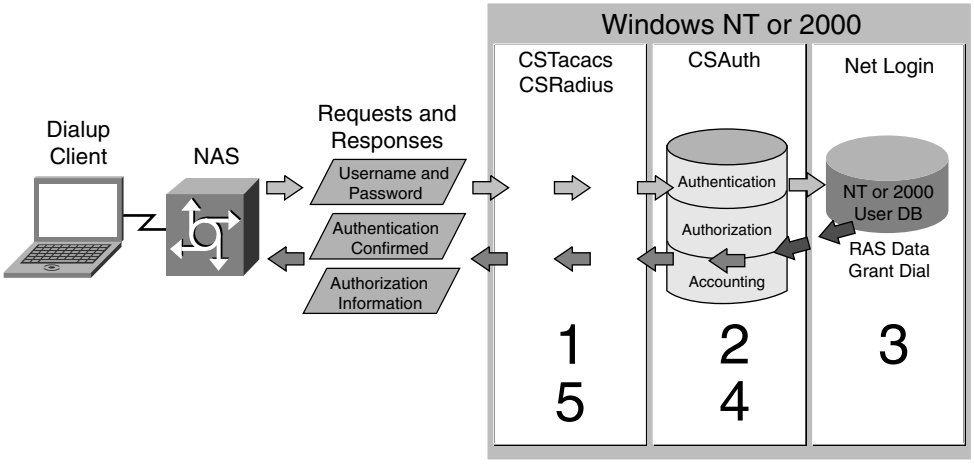
Using Windows User Database

Figure 3-7 shows the flow of the steps used when you elect to use the Windows NT or Windows 2000 user database for authentication and authorization.

Following the numbers shown in Figure 3-7, when Cisco Secure ACS for Windows uses the Windows NT or Windows 2000 user database for AAA, the following service and database interaction occurs:

1. TACACS+ or RADIUS service directs the request to the Cisco Secure ACS Authentication and Authorization service.
2. The username and password are sent to the Windows NT or Windows 2000 user database for authentication.
3. If approved, Windows NT or Windows 2000 grants dial permission as a local user.
4. A response is returned to Cisco Secure ACS for Windows and authorizations are assigned.
5. Confirmation and associated authorizations assigned in Cisco Secure ACS for Windows for that user are sent to the network access server. Accounting information is logged.

Figure 3-7 *Using Windows User Database*

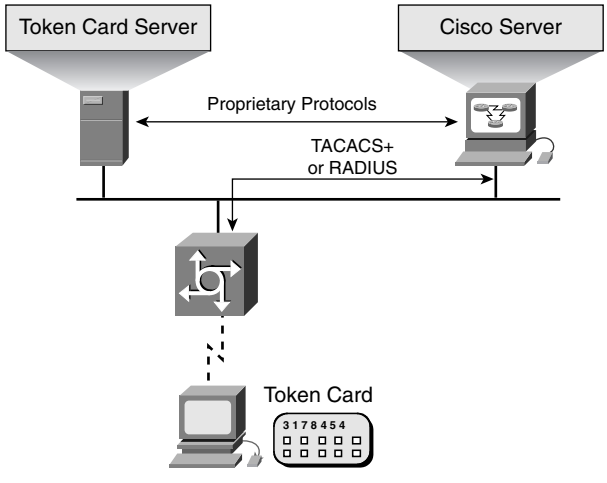


An added benefit of using the Windows NT or Windows 2000 user database is that the username and password that are used for authentication are the same that are used for network login. As such, you can require users to enter their username and password once, for the convenience of a simple, single login.

Token Card Support

Cisco Secure ACS for Windows supports several third-party token servers, such as RSA SecurID, Secure Computing SafeWord, Symantec (AXENT) Defender, and any hexadecimal X.909 token card such as CRYPTOCard. As shown in Figure 3-8, for some token servers, Cisco Secure ACS for Windows acts as a client to the token server.

Figure 3-8 *Token Card Support*



For others, it uses the token server's RADIUS interface for authentication requests. As with the Windows NT or Windows 2000 database, after the username is located in the Cisco Secure user database, CSAuth can check the selected token server to verify the username and token-card password. The token server then provides a response, approving or denying validation. If the response is approved, CSAuth knows that authentication should be granted for the user.

Cisco Secure ACS for Windows can support token servers using the RADIUS server that is built into the token server. Rather than using the vendor's proprietary API, Cisco Secure ACS for Windows sends standard RADIUS authentication requests to the RADIUS authentication port on the token server. The token servers that are supported through their RADIUS servers are those from ActivCard, CRYPTOCard, VASCO, PassGo Technologies, RSA Security, and Secure Computing.

NOTE Before Cisco Secure ACS 3.0.1, support for CRYPTOCard token servers used the vendor-proprietary interface provided with the CRYPTOCard token server.

Beginning with Cisco Secure ACS 3.0.1, Cisco supports CRYPTOCard token servers using a standard RADIUS interface.

Cisco Secure ACS for Windows also supports any token server that is a RADIUS server compliant with IETF RFC 2865. So, in addition to the RADIUS-enabled token server vendors that are explicitly supported, this enables you to use any token server that supports RADIUS-based authentication.

You can create multiple instances of each of these token server types in Cisco Secure ACS for Windows.

Versions 3.1 and 3.2 Enhancements

Cisco is constantly upgrading and enhancing hardware and software products, and Cisco Secure ACS for Windows is no exception. You can always find the latest version information at Cisco's website. This section looks at some of the important new features that have been added to Cisco Secure ACS for Windows by versions 3.1 and 3.2.

The following are the Cisco Secure ACS for Windows version 3.1 product enhancements:

- **Protected Extensible Authentication Protocol (PEAP) support**—Nonproprietary PEAP for wireless user authentication provides stronger security, greater extensibility, and support for one-time token authentication and password aging.
- **SSL support for administrative access**—SSL can be used to secure administrative access to the Cisco Secure ACS for Windows HTML interface.

- **Change Password (CHPASS) improvements**—Cisco Secure ACS for Windows allows you to control whether network administrators can change passwords during Telnet sessions that are hosted by TACACS+ AAA clients.
- **Improved IP pool addressing**—To reduce the possibility of allocating an IP address that is already in use, Cisco Secure ACS for Windows uses the IETF RADIUS Class attribute as an additional index for user sessions.
- **Network device search**—New search capabilities let you search for a configured network device based on the device name, IP address, type (AAA client or AAA server), and network device group.
- **Improved Public Key Infrastructure (PKI) support**—During Extensible Authentication Protocol Transport Layer Security (EAP-TLS) authentication, Cisco Secure ACS for Windows can perform binary comparison of the certificate received from an end-user client to user certificates stored in Lightweight Directory Access Protocol (LDAP) directories.
- **Extensible Authentication Protocol (EAP) proxy enhancements**—Cisco Secure ACS for Windows supports Light Extensible Authentication Protocol (LEAP) and EAP-TLS proxy to other RADIUS or external databases using EAP over standard RADIUS.
- **CiscoWorks Management Center application support**—Cisco Secure ACS for Windows provides a consolidated administrative TACACS+ control framework for many Cisco security management tools, such as CiscoWorks VPN/Security Management Solution (VMS) and the suite of CiscoWorks Management Centers.

The following are the Cisco Secure ACS for Windows version 3.2 product enhancements:

- **PEAP support for Microsoft Windows clients**—Support for Microsoft PEAP supplicants that are available for Windows 98, NT, 2000, and XP was added in this update.
- **LDAP multithreading**—To improve performance in task-intensive environments such as wireless deployments, Cisco Secure ACS for Windows Server Version 3.2 is now capable of processing multiple LDAP authentication requests in parallel.
- **EAP-TLS enhancements**—New EAP-TLS enhancements have been brought in Cisco Secure ACS for Windows Server Version 3.2 that further extend Cisco Secure ACS PKI capabilities.
- **Machine authentication support**—Machine authentication allows pulling down machine group policies from Windows Active Directory independently of a subsequent interactive user authentication session.
- **EAP mixed configurations**—Cisco Secure ACS for Windows Server Version 3.2 supports the following EAP types:
  - PEAP (EAP-GTC)
  - PEAP (EAP-MSCHAPv2)
  - EAP-TLS
  - EAP-MD5
  - Cisco EAP wireless

- **Flexible EAP settings allowed**—One or several EAP types can be selected concurrently.
- **Accounting support for Aironet**—Cisco Secure ACS for Windows Server Version 3.2 supports user-based accounting from Cisco Aironet wireless access.
- **Downloadable access control lists for VPN users**—Cisco Secure ACS for Windows Server Version 3.2 extends per-user ACL support to Cisco VPN solutions.

Cisco Secure ACS for UNIX (Solaris)

Cisco Secure ACS for UNIX is used to authenticate users and determine which internal networks and services they may access. By authenticating users against a database of user and group profiles, Cisco Secure ACS for UNIX effectively secures private enterprise and service provider networks from unauthorized access.

Cisco Secure ACS for UNIX incorporates a multiuser, web-based Java configuration and management tool that simplifies server administration and enables multiple system administrators to simultaneously manage security services from multiple locations. The GUI supports Microsoft and Netscape web browsers, providing multiplatform compatibility and offering secure administration via the industry-standard SSL communication mechanism.

Token cards from CRYPTOCard, Secure Computing Corporation, and RSA Security are supported. Token cards are the strongest available method to authenticate users dialing in and to prevent unauthorized users from accessing proprietary information. Cisco Secure ACS for UNIX now supports industry-leading relational database technologies from Sybase, Inc. and Oracle Corporation. Traditional scalability, redundancy, and nondistributed architecture limitations are removed with the integration of relational database technologies, such as Sybase's SQLAnywhere. Storage and management of user and group profile information is greatly simplified.

General Features

Security is an increasingly important aspect of the growth and proliferation of LANs and WANs. You want to provide easy access to information on your network, but you also want to prevent access by unauthorized personnel. Cisco Secure ACS for UNIX is designed to help ensure the security of your network and track the activity of people who successfully connect to your network. Cisco Secure ACS for UNIX uses the TACACS+ protocol to provide this network security and tracking.

TACACS+ uses AAA to provide network access security and enable you to control access to your network from a central location. Each facet of AAA significantly contributes to the overall security of your network, as follows:

- Authentication determines the identity of users and whether they should be allowed access to the network.

- Authorization determines the level of network services available to authenticated users once they are connected.
- Accounting keeps track of each user's network activity.
- AAA within a client or server architecture (in which transaction responsibilities are divided into two parts: client [front end], and server [back end]) allows you to store all ecurity information in a single, centralized database instead of distributing the information around the network in many different devices.

For further information on AAA, see the section titled "Introduction to AAA for Cisco Routers" in Chapter 2, "Basic Cisco Router Security."

You can use Cisco Secure ACS for UNIX to make changes to the database that administers security on your network on a few security servers instead of making changes to every NAS in your network.

Using Cisco Secure ACS for UNIX, you can expand your network to accommodate more users and provide more services without overburdening system administrators with security issues. As new users are added, system administrators can make a small number of changes in a few places and still ensure network security.

Cisco Secure ACS for UNIX can be used with the TACACS+ protocol, the RADIUS protocol, or both. Some features are common to both protocols, while other features are protocol-dependent.

Cisco Secure ACS for UNIX has the following features when used with either the TACACS+ or RADIUS protocol:

- Support for use of common token card servers, including those from CRYPTOCard, Secure Computing (formerly Enigma Logic), and RSA Security
- Relational database support for Oracle Enterprise, Sybase Enterprise, and Sybase SQLAnywhere (supplied with Cisco Secure ACS for UNIX)
- Encrypted protocol transactions so that passwords are never subject to unauthorized monitoring
- Supported on SPARC Solaris version 2.51 or greater
- Support for group membership
- Support for accounting
- Support for S/Key authentication
- Ability to specify the maximum number of sessions per user
- Ability to disable an account after *n* failed attempts
- Web-based interface for easy administration of network security

Customers can upgrade to any 2.x version of Cisco Secure ACS for UNIX from existing versions, gaining access to the many user-friendly features of the latest version of Cisco Secure ACS for UNIX.

Cisco Secure ACS for UNIX 2.3 adds the Distributed Systems Manager (DSM), which enables system administrators to

- Limit the number of concurrent sessions that are available to a specific user, group, or VPDN (DSM enabled)
- Set per-user session limits for individual users or groups of users (limited support without DSM enabled)

Installing Cisco Secure ACS 3.0 for Windows 2000/NT Servers

Cisco Secure ACS 3.0 for Windows 2000/NT Servers is easy to install and configure. This section presents a brief overview of the essential installation steps. The following discussion is based on a Point-to-Point Protocol (PPP) dialup user being authenticated against Cisco Secure ACS for Windows using the Windows NT or Windows 2000 user database, via the TACACS+ protocol.

The Cisco Secure ACS for Windows installation can be condensed to the following steps:

Step 1 Configure the Windows NT or Windows 2000 server to work with Cisco Secure ACS for Windows.

Step 2 Verify a basic network connection from the Windows NT or Windows 2000 server to the network access server using ping and Telnet.

Step 3 Install Cisco Secure ACS for Windows on the Windows NT or Windows 2000 server following the Windows NT or Windows 2000 installation shield.

Step 4 Configure Cisco Secure ACS for Windows via the web browser interface.

Step 5 Configure the network access server for AAA.

Step 6 Verify correct installation and operation.

Configuring the Server

The first step to follow when installing Cisco Secure ACS for Windows is to configure Windows NT or Windows 2000 for Cisco Secure ACS for Windows by performing the following steps:

Step 1 Determine whether the host server is a domain controller or a member server. This decision must be made based on the design of the Windows NT or Windows 2000 server architecture of your company.

Step 2 Configure Windows NT or Windows 2000 User Manager.

Step 3 Use Windows NT or Windows 2000 services to control ACS.

Cisco does not recommend that you install Cisco Secure ACS for Windows on PDCs or BDCs. These Windows authentication devices can become very busy and are frequent targets of

network attacks. Placing Cisco Secure ACS for Windows on one of these devices exposes it to potential compromise and possible service delays.

Verifying Connections Between Windows Server and Other Network Devices

Verify that the NAS or router can ping the Windows NT or Windows 2000 server that will host Cisco Secure ACS for Windows. This verification will simplify installation and eliminate problems when configuring Cisco Secure ACS for Windows and devices that interface with it.

Cisco Secure ACS for Windows is easy to install from a CD-ROM. It installs like any other Windows application, using an InstallShield template. Before you begin the installation, ensure that you have the network access server information, such as host name, IP address, and TACACS+ key. Be sure that the version of Java that is identified in the installation manual is installed on the server before you begin the installation process.

NOTE Beginning with Cisco Secure ACS for Windows version 3.1, Cisco no longer supports running Cisco Secure ACS for Windows on a Windows NT 4.0 server.

Installing Cisco Secure ACS for Windows on the Server

Follow the InstallShield template instructions as listed below:

Step 1 Select and configure the database.

Step 2 Configure Cisco Secure ACS for Windows for the NAS or router using the web browser.

Step 3 Configure the NAS or router for Cisco Secure ACS for Windows.

Configuring Cisco Secure ACS for Windows Using the Web Browser

After you successfully install Cisco Secure ACS for Windows, an ACS Admin icon appears on the Windows NT or 2000 desktop. You configure and manage Cisco Secure ACS for Windows through the web-based GUI. The GUI is designed using frames, so you must view it with a supported web browser.

Cisco Secure ACS for Windows supports only HTML; a web browser is the only way to configure it. Cisco Secure ACS for Windows supports the following browsers:

- Microsoft Internet Explorer version 5.0 and above for Microsoft Windows
- Netscape Communicator version 4.76 and above for Microsoft Windows

Continue the initial configuration of Cisco Secure ACS for Windows as follows:

Step 1 Select the icon to launch the browser with the address http://127.0.0.1:2002/.

- http://*ip address*:2002/ and http://*host name*:2002/ also work.

Step 2 Perform required tasks to establish users and groups, and to configure network and system settings as outlined in the section of this chapter titled, "Administering and Troubleshooting Cisco Secure ACS for Windows."

Configuring Remaining Devices for AAA

You must configure the NAS, routers, and switches to work with Cisco Secure ACS for Windows. Router configuration is described in Chapter 6, "Cisco IOS Firewall Authentication Proxy."

You may also need to configure a token card server to work with Cisco Secure ACS for Windows to perform AAA.

The following are some of the possible configuration combinations in which Cisco Secure ACS for Windows is used to perform AAA. In each configuration, each of the devices must be configured to work with Cisco Secure ACS for Windows.

- Dialup using the Windows NT or Windows 2000 user database with TACACS+
- Dialup using the Cisco Secure ACS for Windows user database with TACACS+
- Dialup using a token card server with TACACS+
- Dialup using the Cisco Secure ACS for Windows user database with RADIUS (Cisco)
- Dialup for an AppleTalk Remote Access Protocol (ARAP) client using the Cisco Secure ACS for Windows user database with TACACS+
- Router management using the Cisco Secure ACS for Windows user database with TACACS+
- PIX Firewall authentication/authorization using the Windows NT or Windows 2000 user database with TACACS+

Verify Correct Installation and Operation

Verification of correct installation begins by checking to see whether Cisco Secure ACS services are running or stopped by accessing the Service Control page. You can do that by following these steps:

Step 1 In the navigation bar, click **System Configuration**.

Step 2 Click **Service Control** to display the status of the Cisco Secure ACS for Windows services.

Next, you need to test authentication and authorization from one of your devices that has been configured to use the server. A good test is to use a Telnet connection to a router that has been configured for AAA on its VTY lines.

Step 1 Connect to the router though Telnet.

Step 2 Enter your username and password when prompted by the system.

Step 3 Verify that you are granted the level of access control you expected to receive based on the username you used.

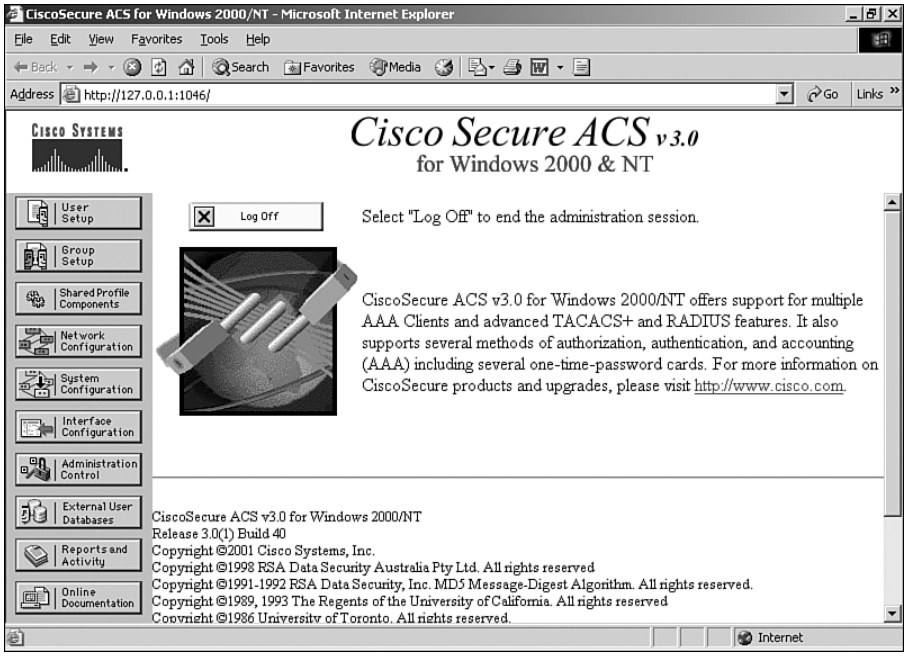
Step 4 If there are any problems, verify the configuration on the router and double-check the Cisco Secure ACS for Windows settings that you established for the test user.

Administering and Troubleshooting Cisco Secure ACS for Windows

The Cisco Secure ACS for Windows web browser interface makes administration of AAA features easy. Each of the buttons on the navigational bar (listed next in top-to-bottom order as shown in Figure 3-9) represents a particular area or function that you can configure. Depending on your configuration, you may not need to configure all the areas.

- **User Setup**—Add, edit, and delete user accounts, and list users in databases
- **Group Setup**—Create, edit, and rename groups, and list all users in a group

Figure 3-9 *Main Window*



- **Shared Profile Components**—Develop and name reusable, shared sets of authorization components that may be applied to one or more users or groups of users and referenced by name within individual profiles. Components include network access restrictions (NARs), command authorization sets, and downloadable PIX ACLs.
- **Network Configuration**—Configure and edit network access server parameters, add and delete network access servers, and configure AAA server distribution parameters
- **System Configuration**—Start and stop Cisco Secure ACS for Windows services, configure logging, control Database Replication, and control RDBMS Synchronization
- **Interface Configuration**—Configure user-defined fields that will be recorded in accounting logs, configure TACACS+ and RADIUS options, and control display of options in the user interface
- **Administration Control**—Control administration of Cisco Secure ACS for Windows from any workstation on the network
- **External User Databases**—Configure the unknown user policy, configure authorization privileges for unknown users, and configure external database types
- **Reports and Activity**—View the following information, which is a partial list of the types of reports available to you when you select this button. You can import these files into most database and spreadsheet applications.
  - **TACACS+ Accounting Report**—Lists when sessions stop and start, records network access server messages with username, provides CLID information, and records the duration of each session
  - **RADIUS Accounting Report**—Lists when sessions stop and start, records network access server messages with username, provides CLID information, and records the duration of each session
  - **Failed Attempts Report**—Lists authentication and authorization failures with an indication of the cause
  - **Logged in Users**—Lists all users who are currently receiving services for a single network access server or all network access servers with access to Cisco Secure ACS for Windows
  - **Disabled Accounts**—Lists all user accounts that are currently disabled
  - **Admin Accounting Report**—Lists configuration commands entered on a TACACS+ (Cisco) network access server
- **Online Documentation**—Provides more detailed information about the configuration, operation, and concepts of Cisco Secure ACS for Windows

As previously stated, the preceding list represents the order in which the buttons appear on the navigational bar, not the order that you want to follow for configuration. The order to follow for configuration depends on your preferences and needs. One typical order of configuration is as follows:

Step 1 **Administration Control**—Configure access for remote administrators.

Step 2 **Network Configuration**—Configure and verify connectivity to a network access server.

Step 3 **Group Setup**—Configure available options and parameters for specific groups. All users must belong to a group.

Step 4 **User Setup**—Add users to a group that is configured.

Step 5 **Additional configuration**—Verify or configure settings in all other necessary areas.

Start troubleshooting Cisco Secure ACS for Windows–related AAA problems by examining the Failed Attempts Report under Reports and Activity, as shown in Figure 3-10. The report shows several types of failures.

Figure 3-10 *Troubleshooting Grid*

Date ↓	Time	Message-Type	User Name	Group Name	Caller-ID	Authen-Failure-Code	Authen-Failure-Code	Author-Data	NAS-Port	NAS-IP-Address
12/06/2002	12:59:46	Author Failed	aaauser	Default Group	10.1.2.12	..	Service denied	service=auth-proxy cmd*	Ethernet0/0	10.0.2.2
12/06/2002	12:58:31	Author Failed	aaauser	Default Group	10.1.2.12	..	Service denied	service=auth-proxy cmd*	Ethernet0/0	10.0.2.2
12/06/2002	12:38:10	Authen Failed	andy	is-in	async	CS password invalid	..	..	tty0	10.0.2.2

Cisco Secure ACS for Windows has debug capabilities that uses a combination of logging files to record debug information. You can view these logging files as reports in order to check for system problems. You can also run CSTacacs, CSRadius, and CSAuth from a DOS command line to see debug information for those services. See the Cisco Tech Notes article at http://www.cisco.com/en/US/products/sw/secursw/ps2086/products_tech_note09186a00800afec1.shtml for more information on setting debug log levels and using the debug command-line capabilities of the Cisco Secure ACS for Windows service modules.

Authentication Failure

Assuming that Cisco Secure ACS for Windows and the router are communicating and that you are authenticating against the Windows NT or Windows 2000 user database, check the following items if you encounter an authentication failure:

- Are the username and password being entered correctly? (The password is case sensitive.)
- Do the username and password exist in the Windows NT or Windows 2000 user database? (Use the Windows 2000 User Manager administration tool to verify the user entry and to reset passwords, if necessary.)
- Is the dial-in interface on the network access server configured with **ppp authentication pap**?

- Is the User Must Change Password at Next Login check box checked in Windows NT or Windows 2000? (Uncheck the check box it if it is checked.)
- Does the username have the rights to log on locally in the Windows NT or Windows 2000 Server window (Trust Relationship/Domain)?
- Is Cisco Secure ACS for Windows configured to authenticate against the Windows NT or Windows 2000 user database?
- Is Cisco Secure ACS for Windows configured to grant dial-in permission to the user?
- If the username was able to authenticate before and is unable to now, is the account disabled on Windows NT or Windows 2000 or Cisco Secure ACS for Windows?
- Has the password expired on Windows NT or Windows 2000?
- Does the username contain an illegal character?

Windows NT or Windows 2000 will send the domain name and username for authentication when a user attempts to access the network through Dial-Up Networking (DUN).

Authorization Failure

If the dial-in user is authenticating, but authorization is failing, check the following:

- Are the proper network services checked in the Group Settings?
- If IP is checked, how is the dial-in user obtaining an IP address?
- Is there an IP pool configured on the network access server?
- Is the name of the IP pool entered in the Group Settings? (Leave this blank if a default IP pool has been configured.)
- If authorizing commands, has the **aaa authorization** *commands* **1 tacacs+** command been entered into the Cisco IOS configuration? (You can substitute any privilege level from 0 to 15 for the **1** in this command.)
- Has the radio button for the command been selected?
- Has the radio button for the argument been selected?

Accounting Failure

If AAA is not working, yet there is no entry in the report, there is an invalid setup between Cisco Secure ACS for Windows and the router. Check the following items to troubleshoot this problem:

- Can the router ping the Windows NT or Windows 2000 server?
- Can the Windows NT or Windows 2000 server ping the router?
- Is the TACACS+ host IP address correctly configured in the router?
- Is the identical TACACS+ host key entered on both the router and Cisco Secure ACS for Windows?
- Is TACACS+ accounting configured on the router?

Troubleshooting Dial-In Client PC Problems

If the dial-in user is a Windows 95 or Windows 98 PC using DUN, here are some things to check:

- Is the proper version of DUN installed? It should be DUN version 1.3.
- Are connection properties configured to use Require Encrypted Password under Server Type?
- Is the connection configured to use the correct protocol?
- Is the selected Dial-Up Server type PPP: Windows 95/98, Windows NT 3.5, Internet?
- Is the user authorized to use a specific command?

Other problems may be encountered with remote administration. Check the following:

- Ensure that the web browser is correctly configured—enough cache is allocated and Java is enabled.
- Ensure that Remote Administration is configured to allow remote web browser access (IP address and username/password).

Troubleshooting Using Cisco IOS Commands

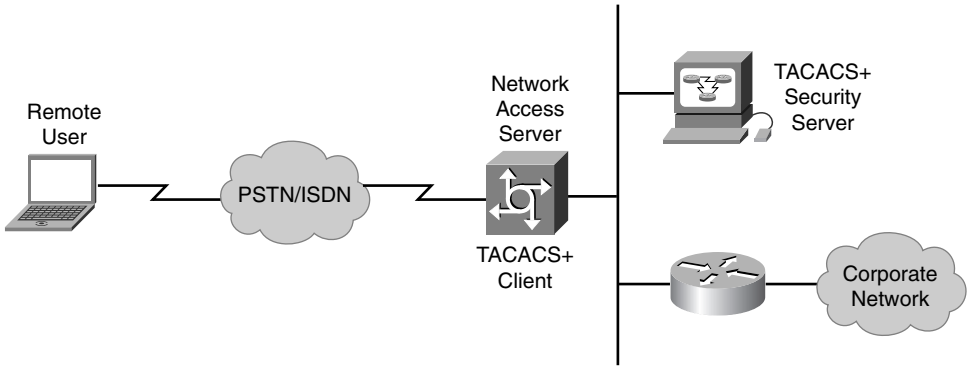
The following Cisco IOS debug commands are useful for troubleshooting:

```
debug aaa authentication
debug aaa authorization
debug tacacs
debug radius
```

TACACS+ Overview

TACACS+ is an improved version of TACACS. TACACS+ forwards username and password information to a centralized security server. Figure 3-11 shows a typical TACACS+ topology.

Figure 3-11 *General Features*



General Features

TACACS+ has the following features:

- **TCP packets for reliable data transport**—TACACS+ uses TCP as the communication protocol between the remote client and security server.
  - Supports the AAA architecture
- **Link is encrypted**—The data payload of IP packets (TCP packets) is encrypted for security and is stored in encrypted form in the remote security database.
  - Supports PAP, CHAP, and MS-CHAP authentication
  - Useful for both LAN and WAN security
- **Serial Line Internet Protocol (SLIP), PPP, and ARA supported for dialup security**—SLIP is TCP/IP over direct connections and modems, which allows one computer to connect to another or to a whole network. PPP is more robust than SLIP, supporting multiple protocols with built-in security. ARA provides to Macintosh users direct access to information and resources at a remote AppleTalk site. TN3270 and X.121 addresses used with X.25 are also supported.
  - Auto-command supported
  - Callback supported
  - Per-user access lists can be assigned in authorization phase

There are at least three versions of TACACS:

- **TACACS**—An industry-standard protocol specification (RFC 1492) that forwards username and password information to a centralized server. The centralized server can be either a TACACS database or a database such as the UNIX password file with TACACS protocol support. For example, the UNIX server with TACACS passes requests to the UNIX database and sends an accept or reject message back to the access server.
- **XTACACS**—Defines the extensions that Cisco added to the TACACS protocol to support new and advanced features. XTACACS is multiprotocol and can authorize connections with SLIP, PPP (IP or Internet Packet Exchange [IPX]), ARA, EXEC, and Telnet. XTACACS supports multiple TACACS servers, syslog for sending accounting information to a UNIX host, connects where the user is authenticated into the access server "shell," and can Telnet or initiate SLIP, PPP, or ARA after initial authentication. XTACACS is essentially obsolete concerning Cisco AAA features and products.
- **TACACS+**—Enhanced and continually improved version of TACACS that allows a TACACS+ server to provide the services of AAA independently. Each service can be tied into its own database or can use the other services available on that server or on the network. TACACS+ was introduced in Cisco IOS Release 10.3. This protocol is a completely new version of the TACACS protocol referenced by RFC 1492 and developed by Cisco. It is not compatible with XTACACS. TACACS+ has been submitted to the IETF as a draft proposal.

The rich feature set of the TACACS+ client/server security protocol is fully supported in Cisco Secure ACS for Windows software.

The first steps in configuring the router are as follows:

Step 1 Enable TACACS+.

Step 2 Specify the list of Cisco Secure ACS for Windows servers that will provide AAA services for the router.

Step 3 Configure the encryption key that is used to encrypt the data transfer between the router and the Cisco Secure ACS for Windows server.

The **aaa new-model** command forces the router to override every other authentication method previously configured for the router lines. If an administrative Telnet or console session is lost while enabling AAA on a Cisco router, and no enable password is specified, the administrator may be locked out of the router and may need to perform the password-recovery process specific to that router to regain access to the device.

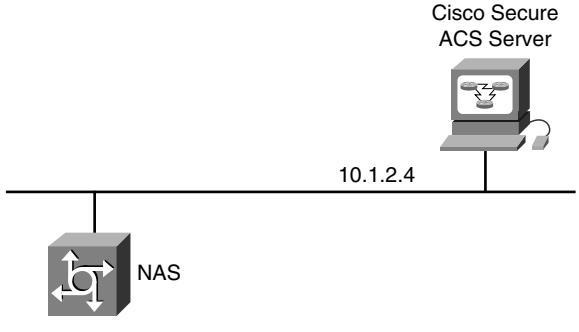
CAUTION When using the Cisco IOS **aaa new-model** command, always provide for an enable password login method. This guards against the risk of being locked out of the router if the administrative session fails while you are in the process of enabling AAA, or if the TACACS+ server becomes unavailable.

Configuring TACACS+

When configuring a NAS to support AAA, as shown in Figure 3-12, at a minimum the following commands should be entered, in the order shown:

```
Router(config)# aaa new-model
Router(config)# aaa authentication login default tacacs+ enable
```

Figure 3-12 *Globally Enabling AAA*



Specifying the "enable" authentication method enables you to reestablish your Telnet or console session and use the enable password to access the router once more. If you fail to do this, and you become locked out of the router, physical access to the router is required (console session), with a minimum of having to perform a password-recovery sequence. At worst, the entire configuration saved in nonvolatile random-access memory (NVRAM) can be lost.

To begin global configuration, enter the following commands, using the correct IP address of the Cisco Secure ACS for Windows servers and your own encryption key:

```
router(config)# tacacs-server key 2bor!2b@?
router(config)# tacacs-server host 10.1.2.4
```

Or use the following:

```
router(config)# tacacs-server host 10.1.2.4 key 2bor!2b@?
```

In this example, the 2bor!2b@? key is the encryption key that is shared between the router and the Cisco Secure ACS for Windows server. The encryption key you choose for your environment should be kept secret to protect the privacy of passwords that are sent between the Cisco Secure ACS for Windows server and the router during the authentication process.

When configuring a TACACS+ key for use on multiple TACACS+ servers, remember that the key must be the same for all TACACS+ servers listed for a given router.

You can specify multiple Cisco Secure ACS for Windows servers by repeating the **tacacs-server** host command.

After you enable AAA globally on the access server, define the authentication method lists, and then apply them to lines and interfaces. These authentication method lists are security profiles that indicate the protocol (ARAP or PPP) or login and authentication method (TACACS+, RADIUS, or local authentication).

To define an authentication method list using the **aaa authentication** command, complete the following steps:

Step 1 Specify the dial-in protocol (ARAP, PPP, or NetWare Access Server Interface [NASI]) or login authentication.

Step 2 Identify a list name or default. A list name is any alphanumeric string you choose. You assign different authentication methods to different named lists. You can specify only one dial-in protocol per authentication method list. However, you can create multiple authentication method lists with each of these options. You must give each list a different name.

Step 3 Specify the authentication method. You can specify up to four methods. For example, you could specify TACACS+, followed by local. This would permit you to log in with a local username and password in case a TACACS+ server is not available on the network.

After defining these authentication method lists, apply them to one of the following:

- **Lines**—tty lines or the console port for login and asynchronous lines (in most cases) for ARA
- **Interfaces**—Interfaces (synchronous or asynchronous) configured for PPP

Use the **aaa authentication** command, as described in Chapter 2, in global configuration mode to enable AAA authentication processes.

The NAS configuration in Example 3-1 shows only commands important to AAA security.

Example 3-1 *Important AAA Security Commands*

```
aaa new-model
aaa authentication login default tacacs+ enable
aaa authentication ppp default tacacs+
aaa authorization exec tacacs+
aaa authorization network tacacs+
aaa accounting exec start-stop tacacs+
aaa accounting network start-stop tacacs+
enable secret 5 $1$x1EE$33AXd2VTVvhbWL0A37tQ3.
enable password 7 15141905172924
!
username admin password 7 094E4F0A1201181D19
!
interface Serial2
 ppp authentication pap
!
tacacs-server host 10.1.1.4
tacacs-server key ciscosecure
!
line con 0
 login authentication no_tacacs
```

Referring to this configuration, the meanings of the configuration entries are as follows:

- **aaa new-model**—Enables the AAA access control model. Use of the **no** form of this command disables this functionality. You can subsequently restore previously configured AAA commands by reissuing the command.

  You could use the **aaa authentication login default tacacs+ enable** command to specify that if your TACACS+ server fails to respond, you can log in to the access server by using your enable password. If you do not have an enable password set on the router, you will not be able to log in to it until you have a functioning TACACS+ UNIX daemon or Windows NT or Windows 2000 server process configured with usernames and passwords. The enable password in this case is a last-resort authentication method. You also can specify **none** as the last-resort method, which means that no authentication is required if all other methods failed.

- **aaa authentication login default tacacs+ enable**—Sets AAA authentication at login using the default list against the TACACS+ server. For this code, the enable password would be used if the TACACS+ server became unavailable.
- **aaa authentication ppp default tacacs+**—Sets AAA authentication for PPP connections using the default list against the TACACS+ database.
- **aaa authorization exec tacacs+**—Sets AAA authorization to determine if the user is allowed to run an EXEC shell on the NAS against the TACACS+ database.
- **aaa authorization network tacacs+**—Sets AAA authorization for all network-related service requests, including SLIP, PPP, PPP NCPs, and ARA protocols, against the TACACS+ database. The TACACS+ database and the NAS must be configured to specify the authorized services.
- **aaa accounting exec start-stop tacacs+**—Sets AAA accounting for EXEC processes on the NAS to record the start and stop time of the session against the TACACS+ database.
- **aaa accounting network start-stop tacacs+**—Sets AAA accounting for all network-related service requests, including SLIP, PPP, PPP NCPs, and ARA protocols, to record the start and stop time of the session against the TACACS+ database.
- **username admin password 7 094E4F0A1201181D19**—Sets a username and password in the local security database for use with the **aaa authentication local-override** command.

NOTE This command shows the password after it has been encrypted, as it would be shown in a display of the router's configuration as a result of the **service password-encryption** command. The password would be entered as clear text and Cisco IOS would take care of the encryption.

- **ppp authentication pap**—Sets PPP authentication to use PAP, CHAP, or both CHAP and PAP. MS-CHAP could also be specified. The **ppp authentication if-needed** command causes the NAS to not perform CHAP or PAP authentication if the user has already provided authentication. This option is available only on asynchronous interfaces.
- **tacacs-server host 10.1.1.4**—Provides the IP address of the TACACS+ server.
- **tacacs-server key ciscosecure**—Provides the shared-secret key that authenticates the router on the TACACS+ server. This password permits AAA communications between the router and the TACACS+ server.

The following are the first steps in configuring the router:

Step 1 Enable TACACS+.

Step 2 Specify the list of Cisco Secure ACS for Windows servers that will provide AAA services for the router.

Step 3 Configure the encryption key that is used to encrypt the data transfer between the router and the Cisco Secure ACS for Windows server.

The **tacacs-server** command is described as follows:

- **tacacs-server host** (*hostname* | *ip-address*)—Specifies the IP address or the host name of the remote TACACS+ server host. This host is typically a UNIX system running TACACS+ software.
- **tacacs-server key** *shared-secret-text-string*—Specifies a shared secret text string used between the access server and the TACACS+ server. The access server and TACACS+ server use this text string to encrypt passwords and exchange responses. The shared key set with the **tacacs-server key** command is a default key to be used if a per-host key was not set. It is a better practice to set specific keys per **tacacs-server host**.

It is possible to configure TACACS+ without a shared key at both the client device (that is, NAS) and the security server (that is, Cisco Secure) if you do not want the connection to be encrypted. This might be useful for a lab or training environment but is strongly discouraged in a production environment.

NOTE A router can have only one **tacacs-server key** command even though it might have multiple **tacacs-server host** commands to configure multiple TACACS+ servers for continuity of service. Therefore, the password that is assigned to the router on each of the TACACS+ servers must be identical and will be the password used in the **tacacs-server key** command on the router.

On the other hand, the TACACS+ server can communicate with multiple host routers, each of which can use a unique key. The TACACS+ server associates the key with the individual router identities in its database.

The following command specifies that the AAA authentication list called no_tacacs is to be used on the console:

```
line con 0
 login authentication no_tacacs
```

Verifying TACACS+

This section explains how to verify AAA TACACS+ operations using the following Cisco IOS debug commands:

```
debug aaa authentication
debug tacacs
debug tacacs events
```

Use the **debug tacacs** command on the router to trace TACACS+ packets and display debugging messages for TACACS+ packet traces.

The output listing below shows part of the **debug aaa authentication** command output for a TACACS login attempt that was successful. The information indicates that TACACS+ is the authentication method used.

```
14:01:17: AAA/AUTHEN (567936829): Method=TACACS+
14:01:17: TAC+: send AUTHEN/CONT packet
14:01:17: TAC+ (567936829): received authen response status = PASS
14:01:17: AAA/AUTHEN (567936829): status = PASS
```

Also, note that the AAA/AUTHEN status indicates that the authentication has passed.

There are three possible results of an AAA session:

- Pass
- Fail
- Error

Pass is the desired output. If you see a Fail or Error result in the **debug aaa authentication** output, it could be the result of a configuration or hardware error. Check the configuration on the router first to make sure the TACACS+ server information is correct, and then check connectivity to the server. If you find no problems, it could indicate a misconfiguration of the TACACS+ server.

Troubleshooting follows a basic tenant—look for what changed last. If you made a configuration change and the AAA process quit working, double-check the configuration, going so far as to restore the configuration that you saved before making the change (You did do that, didn't you?) to see if AAA service is restored. If your router and AAA server have been communicating with no problems, no configuration changes have been made, and no new users have been added, then that points toward a hardware or circuit problem.

The next two sections examine the debug output for successful and failed attempts.

debug tacacs Command Example Output—Failure

Example 3-2 shows part of the **debug tacacs** command output for a TACACS+ login attempt that was unsuccessful as indicated by the status FAIL. The status fields are probably the most useful part of the **debug tacacs** command.

Example 3-2 **debug tacacs** *Command Output for a TACACS Unsuccessful Login Attempt*

```
13:53:35: TAC+: Opening TCP/IP connection to 10.1.1.4/49
13:53:35: TAC+: Sending TCP/IP packet number 416942312-1 to 10.1.1.4/49
(AUTHEN/START)
13:53:35: TAC+: Receiving TCP/IP packet number 416942312-2 from 10.1.1.4/49
13:53:35: TAC+ (416942312): received authen response status = GETUSER
13:53:37: TAC+: send AUTHEN/CONT packet
13:53:37: TAC+: Sending TCP/IP packet number 416942312-3 to 10.1.1.4/49
```

Example 3-2 **debug tacacs** *Command Output for a TACACS Unsuccessful Login Attempt (Continued)*

```
(AUTHEN/CONT)
13:53:37: TAC+: Receiving TCP/IP packet number 416942312-4 from 10.1.1.4/49
13:53:37: TAC+ (416942312): received authen response status = GETPASS
13:53:38: TAC+: send AUTHEN/CONT packet
13:53:38: TAC+: Sending TCP/IP packet number 416942312-5 to 10.1.1.4/49
(AUTHEN/CONT)
13:53:38: TAC+: Receiving TCP/IP packet number 416942312-6 from 10.1.1.4/49
13:53:38: TAC+ (416942312): received authen response status = FAIL
13:53:40: TAC+: Closing TCP/IP connection to 10.1.1.4/49
```

debug tacacs Command Example Output—Pass

Example 3-3 shows part of the **debug tacacs** command output for a TACACS login attempt that was successful, as indicated by the status PASS.

Example 3-3 **debug tacacs** *Command Output for a TACACS Successful Login Attempt*

```
14:00:09: TAC+: Opening TCP/IP connection to 10.1.1.4/49
14:00:09: TAC+: Sending TCP/IP packet number 383258052-1 to 10.1.1.4/49 (AUTHEN/START)
14:00:09: TAC+: Receiving TCP/IP packet number 383258052-2 from 10.1.1.4/49
14:00:09: TAC+ (383258052): received authen response status = GETUSER
14:00:10: TAC+: send AUTHEN/CONT packet
14:00:10: TAC+: Sending TCP/IP packet number 383258052-3 to 10.1.1.4/49 (AUTHEN/CONT)
14:00:10: TAC+: Receiving TCP/IP packet number 383258052-4 from 10.1.1.4/49
14:00:10: TAC+ (383258052): received authen response status = GETPASS
14:00:14: TAC+: send AUTHEN/CONT packet
14:00:14: TAC+: Sending TCP/IP packet number 383258052-5 to 10.1.1.4/49 (AUTHEN/CONT)
14:00:14: TAC+: Receiving TCP/IP packet number 383258052-6 from 10.1.1.4/49
14:00:14: TAC+ (383258052): received authen response status = PASS
14:00:14: TAC+: Closing TCP/IP connection to 10.1.1.4/49
```

debug tacacs events Output

Example 3-4 shows sample **debug tacacs events** command output.

Example 3-4 **debug tacacs events** *Command Output*

```
router# debug tacacs events
%LINK-3-UPDOWN: Interface Async2, changed state to up
00:03:16: TAC+: Opening TCP/IP to 10.1.1.4/49 timeout=15
00:03:16: TAC+: Opened TCP/IP handle 0x48A87C to 10.1.1.4/49
00:03:16: TAC+: periodic timer started
00:03:16: TAC+: 10.1.1.4 req=3BD868 id=-1242409656 ver=193 handle=0x48A87C (ESTAB)
expire=14 AUTHEN/START/SENDAUTH/CHAP queued
00:03:17: TAC+: 10.1.1.4 ESTAB 3BD868 wrote 46 of 46 bytes
00:03:22: TAC+: 10.1.1.4 CLOSEWAIT read=12 wanted=12 alloc=12 got=12
00:03:22: TAC+: 10.1.1.4 CLOSEWAIT read=61 wanted=61 alloc=61 got=49
00:03:22: TAC+: 10.1.1.4 received 61 byte reply for 3BD868
00:03:22: TAC+: req=3BD868 id=-1242409656 ver=193 handle=0x48A87C (CLOSEWAIT) expire=9
AUTHEN/START/SENDAUTH/CHAP processed
00:03:22: TAC+: periodic timer stopped (queue empty)
```

continues

Example 3-4 debug tacacs events *Command Output (Continued)*

```
00:03:22: TAC+: Closing TCP/IP 0x48A87C connection to 10.1.1.4/49
00:03:22: TAC+: Opening TCP/IP to 10.1.1.4/49 timeout=15
00:03:22: TAC+: Opened TCP/IP handle 0x489F08 to 10.1.1.4/49
00:03:22: TAC+: periodic timer started
00:03:22: TAC+: 10.1.1.4 req=3BD868 id=299214410 ver=192 handle=0x489F08 (ESTAB)
expire=14 AUTHEN/START/SENDPASS/CHAP queued
00:03:23: TAC+: 10.1.1.4 ESTAB 3BD868 wrote 41 of 41 bytes
00:03:23: TAC+: 10.1.1.4 CLOSEWAIT read=12 wanted=12 alloc=12 got=12
00:03:23: TAC+: 10.1.1.4 CLOSEWAIT read=21 wanted=21 alloc=21 got=9
00:03:23: TAC+: 10.1.1.4 received 21 byte reply for 3BD868
00:03:23: TAC+: req=3BD868 id=299214410 ver=192 handle=0x489F08 (CLOSEWAIT) expire=13
AUTHEN/START/SENDPASS/CHAP processed
00:03:23: TAC+: periodic timer stopped (queue empty)
```

In this example, the opening and closing of a TCP connection to a TACACS+ server are shown, and also the bytes read and written over the connection and the connection's TCP status.

The TACACS messages are intended to be self-explanatory or for consumption by service personnel only. However, the following two messages that may be shown require a brief explanation:

- **00:03:16: TAC+: Opening TCP/IP to 10.1.1.4/49 timeout=15**—Indicates that a TCP open request to host 10.1.1.4 on port 49 will time out in 15 seconds if it gets no response.
- **00:03:16: TAC+: Opened TCP/IP handle 0x48A87C to 10.1.1.4/49**—Indicates a successful open operation and provides the address of the internal TCP "handle" for this connection.

There is certainly more information provided in the output than there is time or space to address in this book. For more detailed information, refer to the *Debug Command Reference* on the documentation CD-ROM, the Cisco.com website, or in printed form.

You can get more meaningful output from debug commands if you first configure the router using the **service timestamps** *type* [**uptime**] **datetime** [**msec**] [**localtime**] [**show-timezone**] command. The following describes the **service timestamps** command parameters (the parameters in brackets in the preceding sentence are optional):

- *type*—Type of message to time-stamp; debug or log.
- **uptime**—Time-stamp with time since the system was rebooted.
- **datetime**—Time-stamp with the date and time.
- **msec**—Include milliseconds in the date and timestamp.
- **localtime**—Time-stamp relative to the local time zone.
- **show-timezone**—Include the time zone name in the timestamp.

RADIUS Overview

RADIUS is an access server AAA protocol developed by Livingston Enterprises, Inc. (now part of Lucent Technologies). It is a system of distributed security that secures remote access to networks and network services against unauthorized access. RADIUS is composed of three components:

- Protocol with a frame format that uses UDP/IP
- Server
- Client

The server runs on a central computer, typically at the customer's site, while the clients reside in the dial-up access servers and can be distributed throughout the network. Cisco incorporated the RADIUS client into Cisco IOS, starting with Cisco IOS Release 11.1.

Client/Server Model

A router operates as a client of RADIUS. The client is responsible for passing user information to designated RADIUS servers, and then acting on the response that is returned. RADIUS servers are responsible for receiving user connection requests, authenticating the user, and then returning all configuration information that is necessary for the client to deliver service to the user. The RADIUS servers can act as proxy clients to other kinds of authentication servers.

Network Security

Transactions between the client and RADIUS server are authenticated using a shared secret, which is never sent over the network. In addition, any user passwords are sent encrypted between the client and RADIUS server to eliminate the possibility that someone who is snooping on an unsecured network could determine a user's password.

Flexible Authentication Mechanisms

The RADIUS server supports a variety of methods to authenticate a user. When it is provided with the username and original password given by the user, it can support PPP, PAP, CHAP, or MS-CHAP UNIX login, and other authentication mechanisms.

Configuring RADIUS

RADIUS configuration is a three-step process:

Step 1 Configure communication between the router and the RADIUS server.

Step 2 Use the AAA global configuration commands to define authentication and authorization method lists containing RADIUS. Method lists include the keywords as follows:

— **enable**—Uses the enable password for authentication

- **line**—Uses the line password for authentication
- **local**—Uses the local username database for authentication
- **none**—Uses no authentication
- **radius**—Uses RADIUS authentication
- **tacacs+**—Uses TACACS+ authentication

Step 3 Use line and interface commands to cause the defined method lists to be used.

Use the following **radius-server** command to configure router to RADIUS server communication:

```
radius-server key keystring
radius-server host {host-name | ipaddress}
```

You can also accomplish the same thing by combining these two commands into the following single command:

```
radius-server host ipaddress key keystring
```

Examples of using these three commands are shown here:

```
router(config)# radius-server key 2bor!2b@?
router(config)# radius-server host 10.1.2.4
!
router(config)# radius-server host 10.1.2.4 key 2bor!2b@?
```

NOTE The **radius-server** global command is analogous to **tacacs-server** global commands.

RADIUS is a fully open protocol, distributed in source code format that can be modified to work with any security system that is currently available on the market. Cisco supports RADIUS under its AAA security paradigm. RADIUS can be used with other AAA security protocols, such as TACACS+, Kerberos, or local username lookup. Cisco Secure ACS for Windows supports RADIUS.

RADIUS has been implemented in a variety of network environments that require high levels of security while maintaining network access for remote users. RADIUS combines authentication and authorization. The protocol is specified in RFCs 2138 and 2139.

As of this writing, three major versions of RADIUS are available:

- **IETF, with approximately 63 attributes**—Developed and proposed to IETF by Livingston Enterprises, now a division of Lucent Technologies. The RADIUS protocol is specified in RFC 2138, and RADIUS accounting in RFC 2139.
- **Cisco implementation, supporting approximately 58 attributes**—Starting in Cisco IOS Release 11.2, an increasing number of attributes and functionality are included in each release of Cisco IOS software and Cisco Secure ACS for Windows.

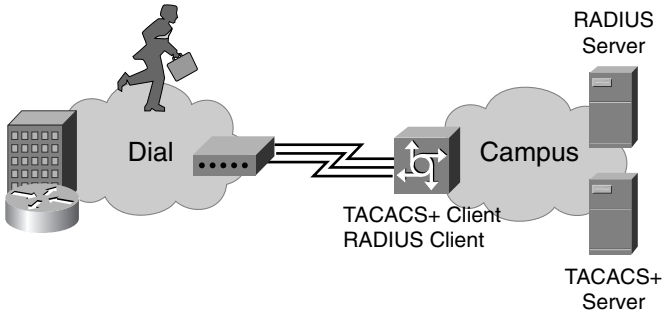
- **Lucent implementation, supporting over 254 attributes**—Lucent is constantly changing and adding vendor-specific attributes, such as token caching and password changing. An API enables rapid development of new extensions, making competing vendors work hard to keep up. Although Livingston Enterprises developed RADIUS originally, it was championed by Ascend.

Vendors have implemented proprietary extensions to RADIUS features. TACACS+ is considered superior because:

- TACACS+ encrypts the entire TACACS+ packet (RADIUS only encrypts the shared-secret password portion).
- TACACS+ separates authentication and authorization, making possible distributed security services.
- RADIUS has limited "name space" for attributes.

Figure 3-13 shows a typical network configuration that uses both TACACS+ and RADIUS.

Figure 3-13 *Comparison of TACACS+ and RADIUS*



There are several differences between TACACS+ and RADIUS:

- **Functionality**—TACACS+ separates AAA functions according to the AAA architecture, allowing modularity of the security server implementation. RADIUS combines authentication and authorization and separates accounting, thus allowing less flexibility in implementation.
- **Transport protocol**—TACACS+ uses TCP. RADIUS uses UDP, which was chosen for simplification of client and server implementation, yet UDP makes the RADIUS protocol less robust and requires the server to implement reliability measures, such as packet retransmission and timeouts, instead having them built into the TCP protocol.
- **Challenge/response**—TACACS+ supports bidirectional challenge and response as used in CHAP between two routers. RADIUS supports unidirectional challenge and response from the RADIUS security server to the RADIUS client.
- **Protocol support**—TACACS+ provides more complete dial-up and WAN protocol support than RADIUS.

- **Data integrity**—TACACS+ encrypts the entire packet body of every packet. RADIUS only encrypts the Password Attribute portion of the Access-Request packet, which makes TACACS+ more secure.
- **Customization**—The flexibility provided in the TACACS+ protocol allows many things to be customized on a per-user basis (that is, customizable username and password prompts). RADIUS lacks flexibility and therefore many features that are possible with TACACS+ are not possible with RADIUS (that is, message catalogs).
- **Authorization process**—With TACACS+, the server accepts or rejects the authentication request based on the contents of the user profile. The client (router) never knows the contents of the user profile. With RADIUS, all reply attributes in the user profile are sent to the router. The router accepts or rejects the authentication request based on the attributes that are received.
- **Accounting**—TACACS+ accounting includes a limited number of information fields. RADIUS accounting can contain more information than TACACS+ accounting records, which is RADIUS's key advantage over TACACS+.

RADIUS Attribute Enhancements

Cisco has introduced enhancements in the latest releases of the Cisco IOS software to support RADIUS attribute capabilities. Some of the more important ones are discussed here.

ACL Default Direction (RADIUS Attribute 11)

The ACL Default Direction feature permits you to configure access lists that dynamically change the RADIUS packet filter direction upon successful RADIUS user authentication. Default filter direction without the application of the RADIUS Attribute 11 capability is outbound, which means that packets must enter the router and be filtered before being sent to the outbound interface.

With RADIUS Attribute 11, you can set the default direction and override that setting as RADIUS authentication occurs. Setting the direction to filter inbound packets causes the router to filter the packets before they are allowed to enter the router.

The following steps are required to implement RADIUS Attribute 11 on a Cisco IOS router:

Step 1 Set the default direction of filters from RADIUS to inbound or outbound using the global **radius-server attribute 11** command. The command format is

```
radius-server attribute 11 direction default [inbound | outbound]
```

Step 2 Attach the **Filter-Id** attribute to the clients on the RADIUS server. Use the format **Filter-Id = "myfilter.out"** to override the default direction to outbound. Use the format **Filter-ID = "myfilter.in"** to override the default direction to inbound. A typical RADIUS user configuration might look like this:

```
Client Password = "mypasswd"
       Service-Type = Framed,
       Framed-Protocol = PPP,
       Filter-Id = "myfilter.out"
```

Accounting Input Gigawords (RADIUS Attribute 52)

Enabled by default, RADIUS Attribute 52 allows the router to maintain a running count of how many times the Acct-Input-Octets counter has wrapped around 2^{32} (4,294,967,296) while providing RADIUS service. The counter resets to 0 at 2^{32} and RADIUS Attribute 52 simply keeps track of how many times the counter refreshed in order to provide an accurate packet count when viewing service statistics. This attribute does not require any configuration.

Accounting Output Gigawords (RADIUS Attribute 53)

Enabled by default, RADIUS Attribute 53 allows the router to maintain a running count of how many times the Acct-Output-Octets counter has wrapped around 2^{32} (4,294,967,296) while providing RADIUS service. The counter resets to 0 at 2^{32} and RADIUS Attribute 52 simply keeps track of how many times the counter refreshed in order to provide an accurate packet count when viewing service statistics. This attribute does not require any configuration.

Tunnel Client Endpoint (Radius Attribute 66)

The Tunnel Client Endpoint capability allows the user to specify the host name of the network access server, rather than having to remember the IP address of the NAS. A typical client configuration might look like the following example:

```
Cisco.com Password = "cisco"
Service-Type = Outbound-User,
Tunnel-Type = :1:L2F,
Tunnel-Medium-Type = :1:IP,
Tunnel-Client-Endpoint = :1:"cisco2"
Tunnel-Server-Endpoint = :1:"172.21.135.4",
Tunnel-Assignment-Id = :1:"nas1",
Tunnel-Password = :1:"cisco"
```

Connection Information (RADIUS Attribute 77)

This attribute is enabled by default and keeps track of the upstream and downstream speeds of connecting clients. You can view these speeds by using the **debug radius** command on the router. You can tell whether the modem connection speed was renegotiated to a lower speed after the connection was made.

Kerberos Overview

Kerberos is a secret-key network authentication protocol, developed at the Massachusetts Institute of Technology (MIT), that uses the Data Encryption Standard (DES) cryptographic algorithm for encryption and authentication. Kerberos was designed to authenticate requests for

network resources. Kerberos, like other secret-key systems, is based on the concept of a trusted third party that performs secure verification of users and services.

In the Kerberos protocol, this trusted third party is called the Key Distribution Center (KDC). It performs the same function as a certification authority (CA), which is discussed in Chapter 9, "Building Advanced IPSec VPNs Using Cisco Routers and Certificate Authorities." The following lists some of the distinguishing characteristics of Kerberos:

- Secret-key authentication protocol
- Authenticates users and network services that they use
- Uses 40- or 56-bit DES for encryption and authentication (weak by today's standards)
- Relies on a trusted third party (KDC) for key distribution
- Embodies "single login" concept
- Expensive to administer—labor intensive

Cisco IOS Release 12.0 includes Kerberos 5 support, which allows organizations that are already deploying Kerberos 5 to use an existing KDC (similar to a CA in IP Security [IPSec]) with their routers and NAS. The following network services are Kerberized in Cisco IOS software:

- **Telnet**—Logs a client (from router to another host) into a server (from another host to router) to permit interactive Telnet sessions
- **rlogin**—Logs a user in to a remote UNIX host for an interactive session similar to Telnet
- **rsh**—Logs a user in to a remote UNIX host and allows execution of one UNIX command
- **rcp**—Logs a user in to a remote UNIX host and allows copying of files from the host

NOTE You can use the **connect** EXEC command with the **/telnet** or **/rlogin** keyword to log in to a host that supports Telnet or rlogin, respectively. You can use the **/encrypt kerberos** keyword to establish an encrypted Telnet session from a router to a remote Kerberos host. Alternatively, you can use the **telnet** EXEC command with the **/encrypt kerberos** keyword to establish an encrypted Telnet session.

NOTE You can use the **rlogin** and **rsh** EXEC commands to initiate rlogin and rsh sessions.

NOTE You can use the **copy rcp** EXEC command or configuration command to enable obtaining configuration or image files from an RCP server.

Chapter Summary

This chapter discussed Cisco Secure ACS for Windows and Cisco Secure ACS for UNIX. The following list identifies important points that were described for each of these management products:

- Cisco Secure ACS for Windows has the following characteristics:
  - Runs as a service on Windows NT or 2000 Server.
  - Authenticates using TACACS+ or RADIUS.
  - Cisco NAS, PIX, VPN 3000 or routers can authenticate against Cisco Secure ACS for Windows.
  - Can use usernames and passwords in the Windows NT or 2000 user database, ACS user database, token server, or NDS.
  - Installation is similar to other Windows applications (InstallShield).
  - Management is done via a web browser.
  - Supports distributed ACS systems.
  - With a remote security server for AAA, the server performs AAA, enabling easier management.
  - TACACS+, RADIUS, and Kerberos are the security server protocols supported by Cisco.
  - Troubleshooting tools include debug commands for TACACS+.
- Cisco Secure ACS for UNIX has the following characteristics:
  - Provides AAA security for enterprise networks.
  - Supports both TACACS+ and RADIUS.
  - Uses the Sybase SQLAnywhere database by default and can interface with Sybase Enterprise SQL and Oracle Enterprise databases.
  - Customers can upgrade any 2.x version of Cisco Secure ACS for UNIX to the most current release.
  - Is easy to install and has a web-based GUI.
  - RADIUS databases can be imported into Cisco Secure ACS for UNIX.

Cisco IOS Commands Presented in This Chapter

Many Cisco IOS version 12.2 commands were discussed or referenced in this chapter. These commands can be found in the *Command Reference* online at http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123mindx/crgindx.htm.

Chapter Review Questions

The following review questions cover some of the key facts and concepts that were introduced in this chapter. Answers to these questions can be found in Appendix A, "Answers to Chapter Review Questions."

1. What authentication protocols are supported by Cisco Secure ACS for Windows?
2. Cisco Secure ACS for Windows can communicate with other ACS servers as masters, clients, or peers to enable what three strong distributed system features?
3. Which Windows NT service module has the primary responsibility for determining whether access should be granted and for defining the privileges associated with each user?
4. List the six steps required to install Cisco Secure ACS for Windows?
5. What protocol must you use to perform the configuration of Cisco Secure ACS for Windows?
6. When you want to configure reusable sets of authorization components to apply to one or more users or groups of users, which option from the Cisco Secure ACS for Windows main menu would you choose?
7. What system administration capabilities does the Cisco Secure ACS for UNIX enable for UNIX 2.3 DSM?
8. What operating system supports Cisco Secure ACS for Unix 2.3?
9. What are the first steps that are required to configure a Cisco IOS router to use TACACS+ with a Cisco Secure ACS for Windows server?
10. You will be configuring your Cisco IOS router for access to three different Cisco Secure ACS for Windows servers using TACACS+. What must you keep in mind as you prepare to configure the router for AAA service?
11. Which Cisco IOS command can you use to get a more meaningful output from debug commands?

Case Study

Continuing with the case study for The Future Corporation, the system administrator now needs to continue the configuration of the DallasR1 router shown in Figure 3-14.

Scenario

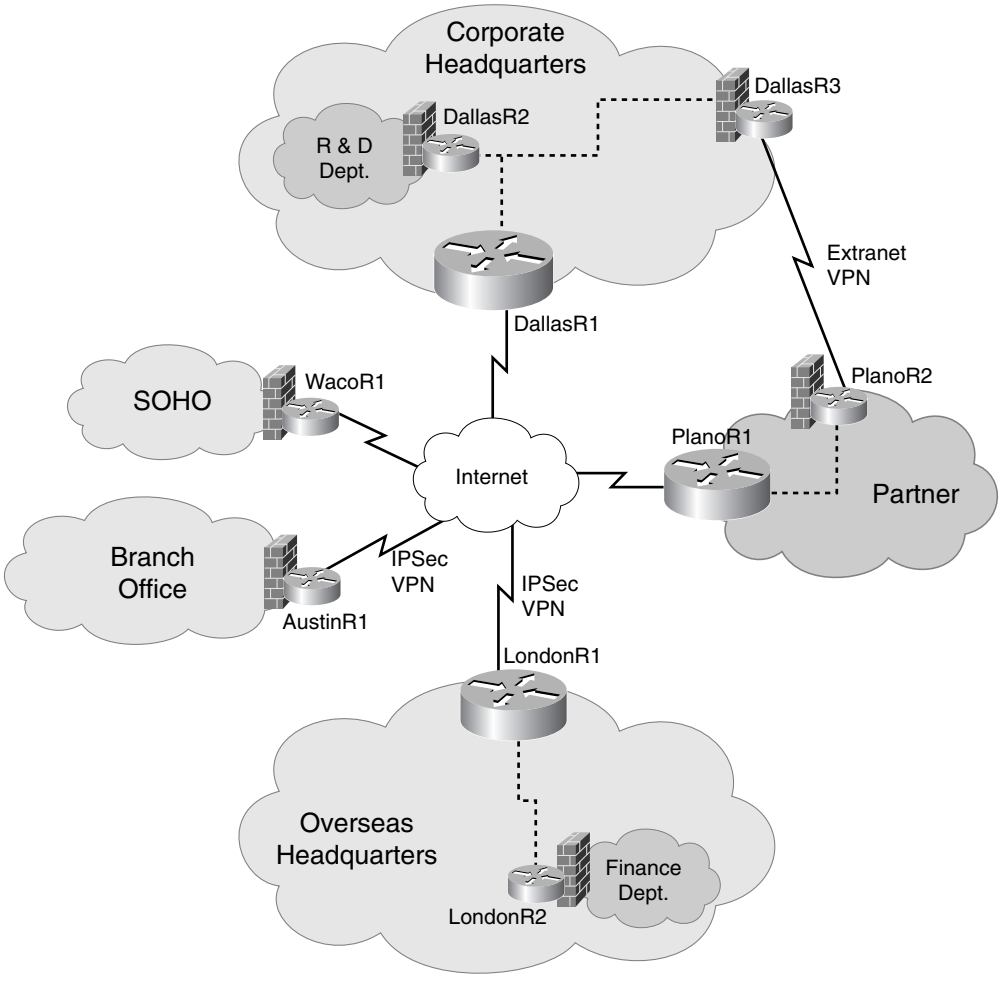
Complete the following configuration tasks on the DallasR1 router:

1. Identify the TACACS+ server whose IP address is 142.16.18.200 using a preshared key of future123key.

2 To make it easier to track logging events and to make debug output more useable, set up the router to time-stamp logging and debug entries using local time. Record debug times to the millisecond.

3 Set up accounting to record all start and stop times for EXEC processes and network processes on the ACS server.

Figure 3-14 *The Future Corporation*



Solutions

The following commands will accomplish the required configuration:

1. Identify the TACACS+ server whose IP address is 142.16.18.200 using a preshared key of future123key:

```
DallasR1(config)# tacacs-server host 142.16.18.200
DallasR1(config)# tacacs-server key future123key
```

2. To make it easier to track logging events and to make debug output more useable, set up the router to time-stamp logging and debug entries using local time. Record debug times to the millisecond.

```
DallasR1(config)# service timestamps debug datetime localtime msec
DallasR1(config)# service timestamps log datetime localtime
```

3. Set up accounting to record all start and stop times for EXEC processes and network processes on the ACS server:

```
DallasR1(config)# aaa accounting exec start-stop tacacs+
DallasR1(config)# aaa accounting network start-stop tacacs+
```

Once these commands have been entered, the configuration for router DallasR1 (excluding interface entries) looks like Example 3-5.

Example 3-5 *DallasR1 Final Configuration*

```
version 12.2
service timestamps debug datetime localtime msec
service timestamps log datetime localtime
service password-encryption
!
hostname DallasR1
!
aaa new-model
aaa authentication login default group tacacs+ local
aaa authentication login console-in group tacacs+ enable
aaa accounting exec start-stop tacacs+
aaa accounting network start-stop tacacs+
enable secret 5 $1$ES4r$tA1rlg0beW/Kvk6jGIj2f.
enable secret level 2 5 $1$mCGe$.1fTlJ.fcR8NHqa0AMR2F/
enable password 7 09611E1C171113171C
!
username aaadmin password 7 1531035C147F3F752B38
!
access-list 88 permit 192.168.44.121
access-list 88 permit 192.168.44.122
access-list 88 permit 192.168.64.123
access-list 88 permit 142.16.18.121
access-list 88 permit 142.16.18.122
access-list 88 permit 142.16.18.123
```

Example 3-5 *DallasR1 Final Configuration (Continued)*

```
snmp-server community ROSNMP ro
snmp-server community RWSNMP rw 88
tacacs-server host 142.16.18.200
tacacs-server key future123key
privilege exec level 2 ping
!
banner motd #
WARNING: You are connected to $(hostname) on The Future Corporation network.
Unauthorized access and use of this network will be vigorously prosecuted. #
!
line con 0
 login authentication console-in
 exec-timeout 4 20
line aux 0
 login
 password 7 112A115507471F5D0721
 exec-timeout 4 20
line vty 0 4
 login
 password 7 05280E5F31195A581A0E
!
end
```