

Configuring Network Access Security

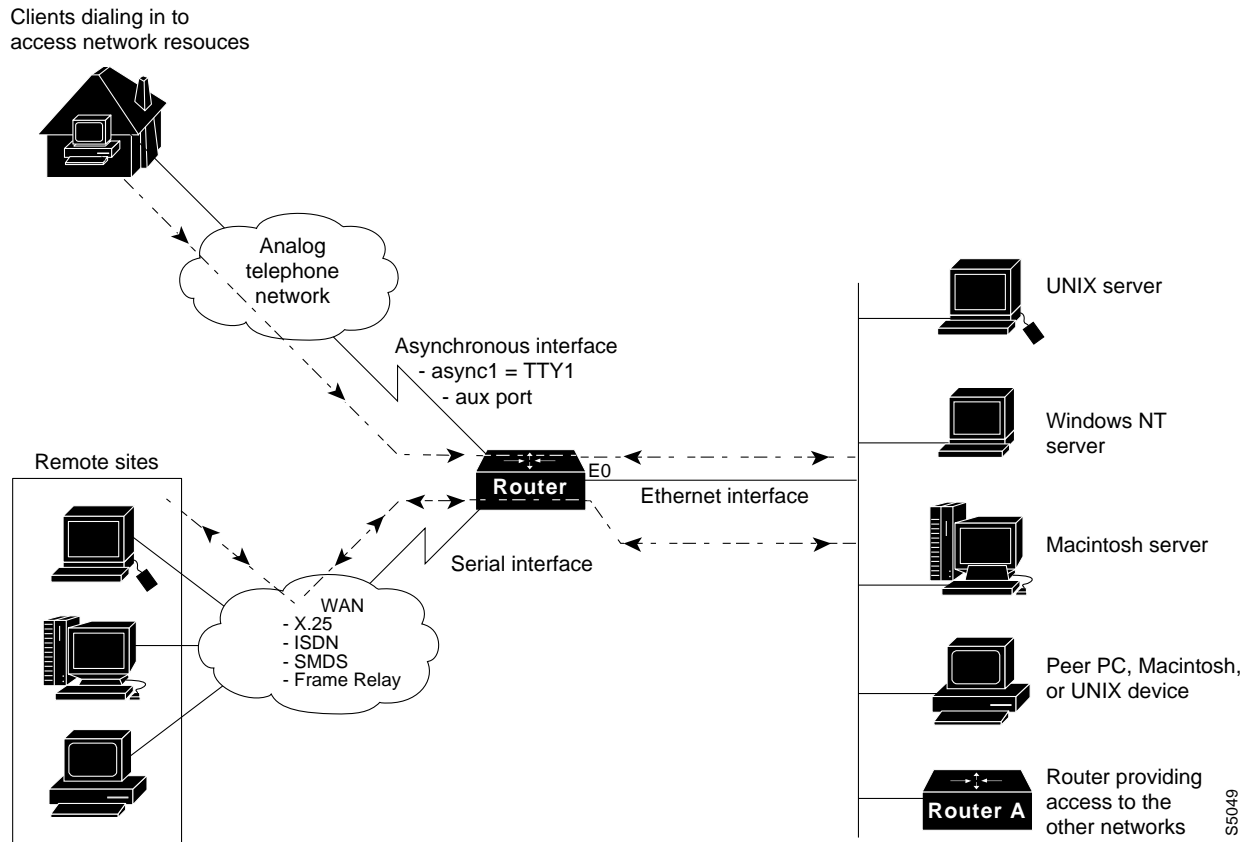
To ensure the confidentiality of sensitive information contained in and moving through your network, you can implement one or more security systems at network access points (see Figure 6). This chapter describes how to control access to your network using several collaborative security systems and includes the following sections:

- Establish RADIUS Access Server Authentication and Accounting
- Establish Kerberos-Authenticated Server-Client System
- Monitor and Maintain Kerberos
- Establish Terminal Access Control Using TACACS
- Additional AAA Authorization Features
- Establish Username Authentication
- Enable CHAP
- Enable PAP
- Configuration Examples

Each section provides a general overview of the technology and offers guidelines for appropriate use. The “Configuration Examples” section at the end of this chapter provides sample configurations for each security system. For a complete description of commands in this chapter, refer to the *Security Command Reference*.

Other chapters in the Cisco internetwork operating system (Cisco IOS) software configuration guides and command references provide information on protocol-specific security features. The “Configuring Interfaces” chapter in the *Configuration Fundamentals Configuration Guide*, for example, provides information on CHAP, an additional authentication feature. Another example is the IP Security Option (IPSO) feature described in the “Configuring IP” chapter in the *Network Protocols Configuration Guide, Part 1*. Finally, refer to the specific protocol chapters for information about how to create access lists.

Figure 6 Network Access Security



Establish RADIUS Access Server Authentication and Accounting

This section describes the Remote Authentication Dial-In User Service (RADIUS) security system, defines its operation, and identifies appropriate and inappropriate network environments for using RADIUS technology. The “Configure RADIUS” section describes how to configure RADIUS with the Authentication, Authorization, and Accounting (AAA) command set. The “RADIUS Authentication and Authorization Examples” section at the end of this chapter offers two possible implementation scenarios.

This section includes the following topics:

- RADIUS Overview
- RADIUS Operation
- Configure RADIUS

RADIUS Overview

Remote Authentication Dial-In User Service (RADIUS) is a distributed client/server system that secures networks against unauthorized access. In the Cisco implementation, RADIUS clients run on Cisco routers and send authentication requests to a central RADIUS server, which contains all user authentication and network service access information.

RADIUS is a fully open protocol, distributed in source code format, that can be modified to work with any security system currently available on the market.

Cisco supports RADIUS under its Authentication, Authorization, and Accounting (AAA) security paradigm. RADIUS can be used with other AAA security protocols, such as TACACS+, Kerberos, or local username lookup. RADIUS is supported on the Cisco 2500, Cisco 4000, and Cisco 7000 series routers.

RADIUS has been implemented in a variety of network environments that require high levels of security while maintaining network access for remote users.

Use RADIUS in the following network environments that require access security:

- Networks with multiple-vendor access servers, each supporting RADIUS—For example, access servers from several vendors use a single RADIUS server-based security database. In an IP-based network with multiple vendor's access servers, dialin users are authenticated through a RADIUS server that has been customized to work with the Kerberos security system.
- Turn-key network security environments in which applications support the RADIUS protocol—For example, in an access environment that uses a “smart card” access control system. In one case, RADIUS has been used with Enigma's security cards to validate users and grant access to network resources.
- Networks already using RADIUS—You can add a Cisco router with RADIUS to the network. This might be the first step when you transition to a Terminal Access Controller Access Control System (TACACS+).
- Networks in which a user must only access a single service—Using RADIUS, you can control user access to a single host, to a single utility such as Telnet, or to a single protocol such as Point-to-Point Protocol (PPP). For example, when a user logs in, RADIUS identifies this user as having authorization to run PPP using IP address 1.2.3.4 and access-list N is started.
- Networks that require resource accounting—You can use RADIUS accounting independently of RADIUS authentication or authorization. The RADIUS accounting functions allow data to be sent at the start and end of services, indicating the amount of resources (such as time, packets, bytes, and so on) used during the session. An Internet service provider (ISP) might use a freeware-based version of RADIUS access control and accounting software to meet special security and billing needs.

RADIUS is not suitable in the following network security situations:

- Multiprotocol access environments—RADIUS does not support the following protocols:
 - AppleTalk Remote Access Protocol (ARAP)
 - NetBIOS Frame Protocol Control Protocol (NBFCP)
 - NetWare Asynchronous Services Interface (NASI)
 - X.25 PAD connections
- Router-to-router situations—RADIUS does not provide two-way authentication. RADIUS can be used to authenticate from one router to a non-Cisco router if the non-Cisco router requires RADIUS authentication.
- Networks using a variety of services—RADIUS generally binds a user to one service model.

RADIUS Operation

When a user attempts to log in and authenticate to an access server using RADIUS, the following steps occur:

- 1 The user is prompted for and enters a username and password.
- 2 The username and encrypted password are sent over the network to the RADIUS server.

- 3 The user receives one of the following responses from the RADIUS server:
 - ACCEPT—The user is authenticated.
 - REJECT—The user is not authenticated and is prompted to re-enter the username and password, or access is denied.
 - CHALLENGE—A challenge is issued by the RADIUS server. The challenge collects additional data from the user.

The ACCEPT or REJECT response is bundled with additional data that is used for EXEC sessions or network authorization. You must first complete RADIUS authentication before using RADIUS authorization. The additional data included with the ACCEPT or REJECT packets consists of the following:

- Services that the user can access, including Telnet, rlogin, or Local Area Transport (LAT) connections, and PPP, Serial Line Internet Protocol (SLIP), or EXEC services.
- Connection parameters, including the host or client IP address, access list, and user timeouts.

Configure RADIUS

To configure RADIUS configuration, you must perform the following three tasks:

- Configure communication between the router and the RADIUS server.
- Use the **aaa authentication** global configuration command to define method lists for RADIUS authentication and authorization. You can also set up AAA accounting for RADIUS connections.
- Use **line** and **interface** commands to enable the defined method lists to be used.

This section describes how to set up RADIUS for authentication, authorization, and accounting on your network, and includes the following sections:

- Configure Router to RADIUS Server Communication.
- RADIUS Authentication Commands.
- RADIUS Authorization Command.
- RADIUS Accounting.
- Configure Router to Display Network Access Server Port Type.

Configure Router to RADIUS Server Communication

The RADIUS host is normally a multiuser system running RADIUS server software from Livingston, Merit, Microsoft, or another software provider. A RADIUS server and a Cisco router use a shared secret text string to encrypt passwords and exchange responses.

To configure RADIUS to use the AAA security commands, you must specify the host running the RADIUS server daemon and a secret text string that it shares with the router.

Use the **radius-server** commands to specify the RADIUS server host and a secret text string.

To specify a RADIUS server host and shared secret text string, perform the following tasks in global configuration mode:

Task	Command
Specify the IP address or host name of the remote RADIUS server host and assign authentication and accounting destination port numbers.	radius-server host { <i>hostname</i> <i>ip-address</i> } [auth-port <i>port-number</i>] [acct-port <i>port-number</i>]
Specify the shared secret text string used between the router and the RADIUS server.	radius-server key { <i>shared-secret_text_string</i> }

To customize communication between the router and the RADIUS server, use the following optional **radius-server** global configuration commands:

Task	Command
Specify the number of times the router transmits each RADIUS request to the server before giving up (default is three).	radius-server retransmit <i>retries</i>
Specify the number of seconds a router waits for a reply to a RADIUS request before retransmitting the request.	radius-server timeout <i>seconds</i>
Specify the number of minutes a RADIUS server, which is not responding to authentication requests, is passed over by requests for RADIUS authentication.	radius-server dead-time <i>minutes</i>

RADIUS Authentication Commands

You configure RADIUS authentication with the **aaa authentication** commands in conjunction with the following service keywords:

Keyword	Description
login	Set RADIUS authentication at user login.
ppp	Define RADIUS authentication method for serial line PPP connections.

To specify RADIUS as the method of authentication on the router, perform the following task in global configuration mode:

Task	Command
Configure the router to use RADIUS for authentication at the login prompt.	aaa authentication login default radius

To specify RADIUS as the method of authentication on a serial interface running PPP, perform the following task in global configuration mode:

Task	Command
Authenticate with RADIUS on PPP connections for users not already authenticated on a TTY line.	aaa authentication ppp ppp-list if-needed radius

In this command, the **ppp-list** keyword identifies a list of methods to be used for authentication when a user logs in via PPP using Password Authentication Protocol (PAP), or Challenge Handshake Authentication Protocol (CHAP). If the user has not already authenticated by some other means (for example, login), RADIUS is used.

Note RADIUS combines the AAA authentication and authorization steps into a single exchange of packets.

For an example of how to configure RADIUS authentication, see the “RADIUS Authentication and Authorization Examples” section at the end of this chapter.

RADIUS Authorization Command

You use the **aaa authorization** command with the **radius** keyword to set parameters that restrict a user’s network access. The Cisco implementation of RADIUS converts the information into the appropriate RADIUS autocommand.

To specify RADIUS authorization for EXEC access and network services, perform the following tasks in global configuration mode:

Task	Command
User RADIUS authorization for all network-related service requests, including SLIP, PPP NCPs, and ARA protocol.	aaa authorization network radius
User RADIUS authorization to determine if the user is allowed to run an EXEC shell. This keyword might return user profile information (such as autocommand information).	aaa authorization exec radius

Note Authorization is bypassed for authenticated users who log in using the console line, even if authorization has been configured.

The **aaa authorization exec radius local** command sets the following authorization parameters:

- Use RADIUS for EXEC level authorization if authentication was done using RADIUS.
- Use the local database if authentication was not done using RADIUS.

For an example of how to specify RADIUS authorization, see the “RADIUS Authentication and Authorization Examples” section at the end of this chapter.

RADIUS Accounting

You use the **aaa accounting** command with the **radius** keyword to turn on RADIUS accounting for each Cisco IOS privilege level, and network services.

To use RADIUS accounting to send a start record accounting notice at the beginning of an EXEC process and a stop record at the end, perform the following task in global configuration mode:

Task	Command
Turn on RADIUS accounting for the EXEC session.	aaa accounting exec start-stop radius

The RADIUS accounting records contain information about EXEC usage time per user.

To use RADIUS to account for all network-related service requests, including SLIP, PPP, and PPP NCPs, perform the following task in global configuration mode:

Task	Command
Use RADIUS accounting for network-related service requests.	aaa accounting network start-stop radius¹

1. This command is documented in the “Accounting and Billing Commands” chapter of the *Security Command Reference*.

This command provides packet and byte counts for connections.

Note No RADIUS-specific **show** commands exist. You can use the **show accounting** command to display accounting information.

For an example of a general configuration using RADIUS accounting, see the “RADIUS Configuration Examples” section at the end of this chapter.

Configure Router to Display Network Access Server Port Type

There are some situations when PPP or login authentication occurs on an interface different from the interface on which the call itself comes in. For example, in a V.120 ISDN call, login or PPP authentication occurs on a virtual asynchronous interface “tt” but the call itself occurs on one of the channels of the ISDN interface. The **radius-server extended-portnames** command configures RADIUS to expand the size of the NAS-Port attribute field to 32 bits. The upper 16 bits of the NAS-Port attribute display the type and number of the controlling interface; the lower 16 bits indicate the interface undergoing authentication.

To display expanded interface information in the NAS-Port attribute field, perform the following task in global configuration mode:

Task	Command
Expand the size of the NAS-Port attribute from 16 to 32 bits to display extended interface information.	radius-server extended-portnames

Note For a complete list of RADIUS attributes, refer to the RADIUS attributes appendix.

RADIUS Attributes

Table 1 lists the supported RADIUS (IETF) attributes. In cases where the attribute has a security server-specific format, the format is specified.

Table 1 Supported RADIUS (IETF) Attributes

Number	Attribute	Description	Cisco IOS Release 11.1	Cisco IOS Release 11.2	Cisco IOS Release 11.3
1	User-Name	Indicates the name of the user being authenticated.	yes	yes	yes
2	User-Password	Indicates the user's password or the user's input following an Access-Challenge. Passwords longer than 16 characters are encrypted using the IETF Draft #2 (or later) specifications.	yes	yes	yes
3	CHAP-Password	Indicates the response value provided by a PPP Challenge-Handshake Authentication Protocol (CHAP) user in response to an Access-Challenge.	yes	yes	yes
4	NAS-IP Address	Specifies the IP address of the network access server that is requesting authentication.	yes	yes	yes
5	NAS-Port	<p>Indicates the physical port number of the network access server that is authenticating the user. The NAS-Port value (32 bits) consists of one or two 16-bit values (depending on the setting of the radius-server extended-portnames command.) Each 16-bit number should be viewed as a 5-digit decimal integer for interpretation as follows:</p> <p>For asynchronous terminal lines, async network interfaces, and virtual async interfaces, the value is 00ttt, where ttt is the line number or async interface unit number.</p> <p>For ordinary synchronous network interface, the value is 10xxx.</p> <p>For channels on a primary rate ISDN interface, the value is 2ppcc.</p> <p>For channels on a basic rate ISDN interface, the value is 3bb0c.</p> <p>For other types of interfaces, the value is 6nnss.</p>	yes	yes	yes

Table 1 Supported RADIUS (IETF) Attributes (Continued)

Number	Attribute	Description	Cisco IOS Release 11.1	Cisco IOS Release 11.2	Cisco IOS Release 11.3
6	Service-Type	<p>Indicates the type of service requested or the type of service to be provided.</p> <ul style="list-style-type: none"> In a request: <ul style="list-style-type: none"> Framed for known PPP or SLIP connection. Administrative-user for enable command. In response: <ul style="list-style-type: none"> Login—Make a connection. Framed—Start SLIP or PPP. Administrative User—Start an EXEC or enable ok. Exec User—Start an EXEC session. 	yes	yes	yes
7	Framed-Protocol	Indicates the framing to be used for framed access.	yes	yes	yes
8	Framed-IP-Address	Indicates the address to be configured for the user.	yes	yes	yes
9	Framed-IP-Netmask	Indicates the IP netmask to be configured for the user when the user is a router to a network. This attribute value results in a static route being added for Framed-IP-Address with the mask specified.	yes	yes	yes
10	Framed-Routing	Indicates the routing method for the user when the user is a router to a network. Only “None” and “Send and Listen” values are supported for this attribute.	yes	yes	yes
11	Filter-Id	Indicates the name of the filter list for the user and is formatted as follows: %d, %d.in, or %d.out. This attribute is associated with the most recent service-type command. For login and EXEC, use %d or %d.out as the line access list value from 0 to 199. For Framed service, use %d or %d.out as interface output access list, and %d.in for input access list. The numbers are self-encoding to the protocol to which they refer.	yes	yes	yes
13	Framed-Compression	Indicates a compression protocol used for the link. This attribute results in a “/compress” being added to the PPP or SLIP autocommand generated during EXEC authorization. Not currently implemented for non-EXEC authorization.	yes	yes	yes

Table 1 Supported RADIUS (IETF) Attributes (Continued)

Number	Attribute	Description	Cisco IOS Release 11.1	Cisco IOS Release 11.2	Cisco IOS Release 11.3
14	Login-IP-Host	Indicates the host to which the user will connect when the Login-Service attribute is included.	yes	yes	yes
15	Login-Service	Indicates the service that should be used to connect the user to the login host.	yes	yes	yes
16	Login-Port	Defines the TCP port with which the user is to be connected when the Login-Service attribute is also present.	yes	yes	yes
18	Reply-Message	Indicates text that might be displayed to the user.	yes	yes	yes
22	Framed-Route	Provides routing information to be configured for the user on this network access server. The RADIUS RFC format (net/bits [router [metric]]) and the old style dotted mask (net mask [router [metric]]) are supported. If the router field is omitted or 0, the peer IP address is used. Metrics are currently ignored.	yes	yes	yes
24	State	Allows state information to be maintained between the network access server and the RADIUS server. This attribute is applicable only to CHAP challenges.	yes	yes	yes

Table 1 Supported RADIUS (IETF) Attributes (Continued)

Number	Attribute	Description	Cisco IOS Release 11.1	Cisco IOS Release 11.2	Cisco IOS Release 11.3
26	Vendor-Specific	<p>Allows vendors to support their own extended attributes not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. Cisco's vendor-ID is 9, and the supported option has vendor-type 1, which is named "cisco-avpair." The value is a string of the format:</p> <pre>protocol : attribute sep value</pre> <p>"Protocol" is a value of the Cisco "protocol" attribute for a particular type of authorization. "Attribute" and "value" are an appropriate AVpair defined in the Cisco TACACS+ specification, and "sep" is "=" for mandatory attributes and "*" for optional attributes. This allows the full set of features available for TACACS+ authorization to also be used for RADIUS. For example:</p> <pre>cisco-avpair= "ip:addr-pool=first" cisco-avpair= "shell:priv-lvl=15"</pre> <p>The first example causes Cisco's "multiple named ip address pools" feature to be activated during IP authorization (during PPP's IPCP address assignment). The second example causes a "NAS Prompt" user to have immediate access to EXEC commands.</p>	yes	yes	yes
27	Session-Timeout	<p>Sets the maximum number of seconds of service to be provided to the user before the session terminates. This attribute value becomes the per-user "absolute timeout." This attribute is not valid for PPP sessions.</p>	yes	yes	yes
28	Idle-Timeout	<p>Sets the maximum number of consecutive seconds of idle connection allowed to the user before the session terminates. This attribute value becomes the per-user "session-timeout." This attribute is not valid for PPP sessions.</p>	yes	yes	yes

Table 1 Supported RADIUS (IETF) Attributes (Continued)

Number	Attribute	Description	Cisco IOS Release 11.1	Cisco IOS Release 11.2	Cisco IOS Release 11.3
34	Login-LAT-Service	Indicates the system with which the user is to be connected by LAT. This attribute is only available in the EXEC mode.	yes	yes	yes
35	Login-LAT-Node	Indicates the node with which the user is to be automatically connected by LAT.	no	no	no
36	Login-LAT-Group	Identifies the LAT group codes that this user is authorized to use.	no	no	no
49	Terminate-Cause	Reports details on why the connection was terminated.	no	no	no

Table 2 lists the supported RADIUS (IETF) accounting attributes. In cases where the attribute has a security server-specific format, the format is specified.

Table 2 Supported RADIUS (IETF) Accounting Attributes

Number	Attribute	Description	Cisco IOS Release 11.1	Cisco IOS Release 11.2	Cisco IOS Release 11.3
25	Class	Arbitrary value that the network access server includes in all accounting packets for this user if supplied by the RADIUS server.	yes	yes	yes
30	Called-Station-Id	Allows the network access server to send the telephone number the user called as part of the Access-Request packet (using Dialed Number Identification [DNIS] or similar technology). This attribute is only supported on ISDN, and modem calls on the Cisco AS5200 if used with PRI.	yes	yes	yes
31	Calling-Station-Id	Allows the network access server to send the telephone number the call came from as part of the Access-Request packet (using Automatic Number Identification or similar technology). This attribute has the same value as “remote-addr” from TACACS+. This attribute is only supported on ISDN, and modem calls on the Cisco AS5200 if used with PRI.	yes	yes	yes
40	Acct-Status-Type	Indicates whether this Accounting-Request marks the beginning of the user service (start) or the end (stop).	yes	yes	yes

Table 2 Supported RADIUS (IETF) Accounting Attributes (Continued)

Number	Attribute	Description	Cisco IOS Release 11.1	Cisco IOS Release 11.2	Cisco IOS Release 11.3
41	Acct-Delay-Time	Indicates how many seconds the client has been trying to send a particular record.	yes	yes	yes
42	Acct-Input-Octets	Indicates how many octets have been received from the port over the course of this service being provided.	yes	yes	yes
43	Acct-Output-Octets	Indicates how many octets have been sent to the port in the course of delivering this service.	yes	yes	yes
44	Acct-Session-Id	A unique accounting identifier that makes it easy to match start and stop records in a log file. Acct-Session ID numbers restart at 1 each time the router is power cycled or the software is reloaded.	yes	yes	yes
45	Acct-Authentic	Indicates how the user was authenticated, whether by RADIUS, the network access server itself, or another remote authentication protocol. This attribute is set to “radius” for users authenticated by RADIUS; “remote” for TACACS+ and Kerberos; or “local” for local, enable, line, and if-needed methods. For all other methods, the attribute is omitted.	yes	yes	yes
46	Acct-Session-Time	Indicates how long (in seconds) the user has received service.	yes	yes	yes
47	Acct-Input-Packets	Indicates how many packets have been received from the port over the course of this service being provided to a framed user.	yes	yes	yes
48	Acct-Output-Packets	Indicates how many packets have been sent to the port in the course of delivering this service to a framed user.	yes	yes	yes
50	Acct-Multi-Session-Id ¹	A unique accounting identifier used to link multiple related sessions in a log file. Each linked session in a multilink session has a unique Acct-Session-Id value, but shares the same Acct-Multi-Session-Id.	no	no	yes
51	Acct-Link-Count ²	Indicates the number of links known in a given multilink session at the time an accounting record is generated. The network access server can include this attribute in any accounting request that might have multiple links.	no	no	yes

Table 2 Supported RADIUS (IETF) Accounting Attributes (Continued)

Number	Attribute	Description	Cisco IOS Release 11.1	Cisco IOS Release 11.2	Cisco IOS Release 11.3
61	NAS-Port-Type	Indicates the type of physical port the network access server is using to authenticate the user.	yes	yes	yes

1. Only stop records contain multi-session IDs. This is because start records are issued before any multilink processing takes place.
2. Only stop records contain link counts. This is because start records are issued before any multilink processing takes place.

Establish Kerberos-Authenticated Server-Client System

This section describes the Kerberos security system and includes the following topics:

- Kerberos Overview
- Kerberos Client Support Operation
- Configure Kerberos

Kerberos Overview

Kerberos is a secret-key network authentication protocol, developed at 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).

The primary use of Kerberos is to verify that users and the network services they use are really who and what they claim to be. To accomplish this, a trusted Kerberos server issues tickets to users. These tickets, which have a limited lifespan, are stored in a user's credential cache and can be used in place of the standard username-and-password authentication mechanism.

The Kerberos credential scheme embodies a concept called "single logon." This process requires authenticating a user once, and then allows secure authentication (without encrypting another password) wherever that user's credential is accepted.

Cisco IOS Release 11.2 includes Kerberos 5 support, which allows organizations already deploying Kerberos 5 to use the same Kerberos authentication database on their routers that they are already using on their other network hosts (such as UNIX servers and PCs).

The following network services are supported by the Kerberos authentication capabilities in Cisco IOS software:

- Telnet
- rlogin
- rsh
- rcp

Note Cisco’s implementation of Kerberos client support is based on code developed by CyberSafe which was derived from the MIT code. As a result, the Cisco Kerberos implementation has successfully undergone full compatibility testing with the CyberSafe Challenger commercial Kerberos server and MIT’s server code, which is freely distributed.

Table 3 lists common Kerberos-related terms and their definitions.

Table 3 Kerberos Terminology

Term	Definition
Authentication	A process by which a user or service identifies itself to another service. For example, a client can authenticate to a router, or a router can authenticate to another router.
Authorization	A means by which the router determines what privileges you have in a network or on the router, and what actions you can perform.
Credential	A general term that refers to authentication tickets, such as ticket granting tickets and service credentials. Kerberos credentials verify the identity of a user or service. If a network service decides to trust the Kerberos server that issued this ticket, it can be used in place of retyping in a username and password. Credentials have a default limited life-span of 8 hours.
Instance	An authorization level label for Kerberos principals. Most Kerberos principals are of the form user@REALM (for example, smith@BOO.COM). A Kerberos principal with a Kerberos instance has the form user/instance@REALM (for example, smith/admin@BOO.COM). The Kerberos instance can be used to specify the authorization level for the user if authentication is successful. It is up to the server of each network service to implement and enforce the authorization mappings of Kerberos instances.
Kerberized	Applications and services that have been modified to support the Kerberos credential infrastructure.
Kerberos realm	A domain consisting of users, hosts, and network services that are registered to a Kerberos server. The Kerberos server is trusted to verify the identity of a user or network service to another user or network service. Kerberos realms must always be in uppercase characters.
Kerberos server	A daemon running on a network host. Users and network services register their identity with the Kerberos server. Network services query the Kerberos server to authenticate to other network services.
Key distribution center (KDC)	A Kerberos server and database program running on a network host.
Principal	Also known as a Kerberos identity, this is who you are or what a service is according to the Kerberos server.
Service credential	A credential for a network service. When issued from the KDC, this credential is encrypted with the password shared by the network service and the KDC, and with the user’s TGT.

Table 3 Kerberos Terminology (Continued)

Term	Definition
SRVTAB	A password that a network service shares with the KDC. The network service authenticates an encrypted service credential by using the SRVTAB (also known as a KEYTAB) to decrypt it.
Ticket granting ticket (TGT)	A credential that the key distribution center (KDC) issues to authenticated users. When users receive a TGT, they can authenticate to network services within the Kerberos realm represented by the KDC.

Kerberos Client Support Operation

This section describes how the Kerberos security system works with a Cisco router functioning as the security server. Although (for convenience or technical reasons) you can customize Kerberos in a number of ways, remote users attempting to access network services must pass through the following three layers of security before they can access network services:

- Authenticate to the Boundary Router
- Obtain a TGT from a KDC
- Authenticate to Network Services

Authenticate to the Boundary Router

This section describes the first layer of security remote users must pass through when they attempt to access a network. The first step in the Kerberos authentication process is for users to authenticate themselves to the boundary router. The following procedure describes how users authenticate to a boundary router:

- 1 The remote user opens a PPP connection to the corporate site router.
- 2 The router prompts the user for a username and password.
- 3 The router requests a ticket granting ticket (TGT) from the key distribution center (KDC) for this particular user.
- 4 The KDC sends an encrypted TGT to the router that includes (among other things) the user's identity.
- 5 The router attempts to decrypt the TGT using the password the user entered. If the decryption is successful, the remote user is authenticated to the router.

A remote user who successfully initiates a PPP session and authenticates to the boundary router is inside the firewall but still must authenticate to the KDC directly before being allowed to access network services. This is because the TGT issued by the KDC is stored on the router and is not useful for additional authentication unless the user physically logs on to the router.

Obtain a TGT from a KDC

This section describes how remote users who are authenticated to the boundary router authenticate themselves to a key distribution center (KDC).

When a remote user authenticates to a boundary router, that user technically becomes part of the network; that is, the network is extended to include the remote user and the user's machine or network. To gain access to network services, however, the remote user must obtain a ticket granting ticket (TGT) from the KDC. The following procedure describes how remote users authenticate to the KDC:

- 1 The remote user, at a workstation on a remote site, launches the KINIT program (part of the client software provided with the Kerberos protocol).
- 2 The KINIT program finds the user's identity and requests a TGT from the KDC.
- 3 The KDC creates a TGT, which contains the identity of the user, the identity of the KDC, and the TGT's expiration time.
- 4 Using the user's password as a key, the KDC encrypts the TGT and sends the TGT to the workstation.
- 5 When the KINIT program receives the encrypted TGT, it prompts the user for a password (this is the password that is defined for the user in the KDC).
- 6 If the KINIT program can decrypt the TGT with the password the user enters, the user is authenticated to the KDC, and the KINIT program stores the TGT in the user's credential cache.

At this point, the user has a TGT and can communicate securely with the KDC. In turn, the TGT allows the user to authenticate to other network services.

Authenticate to Network Services

The following procedure describes how a remote user with a TGT authenticates to network services within a given Kerberos realm. Assume the user on remote workstation (Host A) wants to log in to Host B.

- 1 The user on Host A initiates a Kerberized application (such as Telnet) to Host B.
- 2 The Kerberized application builds a service credential request and sends it to the KDC. The service credential request includes (among other things) the user's identity and the identity of the desired network service. The TGT is used to encrypt the service credential request.
- 3 The KDC tries to decrypt the service credential request with the TGT it issued to the user on Host A.

If the KDC can decrypt the packet, it is assured that the authenticated user on Host A sent the request.

- 4 The KDC notes the network service identity in the service credential request.
- 5 The KDC builds a service credential for the appropriate network service on Host B on behalf of the user on Host A. The service credential contains the client's identity and the desired network service's identity.
- 6 The KDC then encrypts the service credential twice. It first encrypts the credential with the SRVTAB that it shares with the network service identified in the credential. It then encrypts the resulting packet with the TGT of the user (who, in this case, is on Host A).
- 7 The KDC sends the twice-encrypted credential to Host A.
- 8 Host A attempts to decrypt the service credential with the user's TGT.

If Host A can decrypt the service credential, it is assured the credential came from the real KDC.

- 9 Host A sends the service credential to the desired network service. Note that the credential is still encrypted with the SRVTAB the KDC and the network service share.

10 The network service attempts to decrypt the service credential using its SRVTAB.

If the network service can decrypt the credential, it is assured the credential was in fact issued from the KDC. Note that the network service trusts anything it can decrypt from the KDC, even if it receives it indirectly from a user. This is because the user first authenticated with the KDC.

At this point, the user is authenticated to the network service on Host B. This process is repeated each time a user wants to access a network service in the Kerberos realm.

Configure Kerberos

In order for hosts and the key distribution center (KDC) in your Kerberos realm to communicate and mutually authenticate, you must identify them to one another. To do this, you add entries for the hosts to the Kerberos database on the KDC and add SRVTAB files generated by the KDC to all hosts in the Kerberos realm. You also make entries for users in the KDC database.

This section describes how to set up a Kerberos-authenticated server-client system and contains the following topics:

- Configure the KDC Using Kerberos Commands
- Configure the Router to Use the Kerberos Protocol

This section assumes that you have installed the Kerberos administrative programs on a UNIX host, known as the Key Distribution Center (KDC), initialized the database, and selected a Kerberos realm name and password. For instructions about completing these tasks, refer to documentation that came with your Kerberos software.

Note Write down the host name or IP address of the KDC, the port number you want the KDC to monitor for queries, and the name of the Kerberos realm it will serve. You need this information to configure the router.

Configure the KDC Using Kerberos Commands

After you set up a host to function as the key distribution center (KDC) in your Kerberos realm, you must make entries to the KDC database for all principles in the realm. Principles can be network services on Cisco routers and hosts, or users.

This section describes how to use Kerberos commands to add services to the KDC database (and to modify existing database information) and includes the following procedures:

- Add Users to the KDC Database
- Create SRVTABs on the KDC
- Extract SRVTABs

Note All Kerberos command examples are based on Kerberos 5 Beta 5 of the original MIT implementation. Later versions use a slightly different interface.

Add Users to the KDC Database

To add users to the KDC and create privileged instances of those users, use the **su** command to become root on the host running the KDC and use the `kdb5_edit` program to perform the following tasks:

Task	Command
Use the ank (add new key) command to add a user to the KDC. This command prompts for a password, which the user must enter to authenticate to the router.	ank <i>username@REALM</i>
Use the ank command to add a privileged instance of a user.	ank <i>username/instance@REALM</i>

For example, to add user *loki* of Kerberos realm CISCO.COM, enter the following Kerberos command:

```
ank loki@CISCO.COM
```

You might want to create privileged instances to allow network administrators to connect to the router at the enable level, for example, so that they need not enter a cleartext password (and compromise security) to enter enable mode.

To add an instance of *loki* with additional privileges (in this case, *enable*, although it could be anything) enter the following Kerberos command:

```
ank loki/enable@CISCO.COM
```

In each of these examples, you are prompted to enter a password, which you must give to user *loki* to use at login.

The “Enable Kerberos Instance Mapping” section describes how to map Kerberos instances to various Cisco IOS privilege levels.

Create SRVTABs on the KDC

All routers that you want to authenticate to using the Kerberos protocol must have a SRVTAB. This section and the “Extract SRVTABs” section describe how to create and extract SRVTABs for a router called *router1*. The section “Copy SRVTAB Files” describes how to copy SRVTAB files to the router.

To make SRVTAB entries on the KDC, use the **su** command to become root on the host running the KDC and use the `kdb5_edit` program to perform the following task:

Task	Command
Use the ark (add random key) command to add a network service supported by a host or router to the KDC.	ark <i>SERVICE/HOSTNAME@REALM</i>

For example, to add a Kerberized authentication service for a Cisco router called *router1* to the Kerberos realm CISCO.COM, enter the following Kerberos command:

```
ark host/router1.cisco.com@CISCO.COM
```

Note The Kerberos realm name must be in uppercase characters.

Make entries for all network services on all Kerberized hosts that use this KDC for authentication.

Extract SRVTABs

SRVTABs contain (among other things) the passwords or randomly generated keys for the service principles you entered into the KDC database. Service principle keys must be shared with the host running that service. To do this, you must save the SRVTAB entries to a file, then copy the file to the router and all hosts in the Kerberos realm. Saving SRVTAB entries to a file is called *extracting* SRVTABs. To extract SRVTABs, use the **su** command to become root on the host running the KDC and perform the following task:

Task	Command
Use the <code>kdb5_edit</code> command xst to write a SRVTAB entry to a file.	<code>xst router_name host</code>

For example, to write the `host/router1.cisco.com@CISCO.COM` SRVTAB to a file, enter the following Kerberos command:

```
xst router1.cisco.com@CISCO.COM host
```

Use the **quit** command to exit the `kdb5_edit` program.

Configure the Router to Use the Kerberos Protocol

This section describes how to configure a Cisco router to function as a network security server and authenticate users using the Kerberos protocol.

Topics in this section include the following:

- Define a Kerberos Realm
- Copy SRVTAB Files
- Specify Kerberos Authentication
- Enable Credentials Forwarding
- Telnet to the Router
- Enable Kerberos Instance Mapping
- Enable Mandatory Kerberos Authentication
- Monitor and Maintain Kerberos

Define a Kerberos Realm

For a router to authenticate a user defined in the Kerberos database, it must know the host name or IP address of the host running the KDC, the name of the Kerberos realm and, optionally, be able to map the host name or Domain Naming System (DNS) domain to the Kerberos realm.

To configure the router to authenticate to a specified KDC in a specified Kerberos realm, perform the following tasks in global configuration mode (DNS domain names must begin with a leading dot (.)):

Task	Command
Define the default realm for the router.	<code>kerberos local-realm kerberos-realm</code>

Task	Command
Specify to the router which KDC to use in a given Kerberos realm and, optionally, the port number the KDC is monitoring (default is 88).	kerberos server <i>kerberos-realm</i> { <i>hostname</i> <i>ip-address</i> } [<i>port-number</i>]
Optionally, map a host name or DNS domain to a Kerberos realm.	kerberos realm { <i>dns-domain</i> <i>host</i> } <i>kerberos-realm</i>

Note Because the machine running the KDC and all Kerberized hosts must interact within a 5-minute window or authentication fails, all Kerberized machines, and especially the KDC, should be running the Network Time Protocol (NTP).

The **Kerberos local realm**, **Kerberos realm**, and **Kerberos server** commands are equivalent to the UNIX *krb.conf* file. Table 4 identifies mappings from the Cisco IOS configuration commands to a Kerberos 5 configuration file (*krb5.conf*).

Table 4 Kerberos 5 Configuration File and Commands

krb5.conf file	Cisco IOS Configuration Command
[libdefaults]	(in config mode)
default_realm = <i>MURUGA.COM</i>	kerberos local-realm <i>MURUGA.COM</i>
[domain_realm]	(in config mode)
.muruga.com = <i>MURUGA.COM</i>	kerberos realm <i>.muruga.com MURUGA.COM</i>
muruga.com = <i>MURUGA.COM</i>	kerberos realm <i>muruga.com MURUGA.COM</i>
[realms]	(in config mode)
kdc = <i>MURUGA.PIL.COM:750</i>	kerberos server <i>MURUGA.COM 172.65.44.2</i>
admin_server = <i>MURUGA.PIL.COM</i>	(<i>172.65.44.2</i> is the example IP address for <i>MURUGA.PIL.COM</i>)
default_domain = <i>MURUGA.COM</i>	

For an example of defining a Kerberos realm, see the “Define a Kerberos Realm” section at the end of this chapter.

Copy SRVTAB Files

To make it possible for remote users to authenticate to the router using Kerberos credentials, the router must share a secret key with the KDC. To do this, you must give the router a copy of the SRVTAB you extracted on the KDC.

The most secure method to copy SRVTAB files to the hosts in your Kerberos realm is to copy them onto physical media and go to each host in turn and manually copy the files onto the system. To copy SRVTAB files to the router, which does not have a physical media drive, you must transfer them via the network using the Trivial File Transfer Protocol (TFTP).

To remotely copy SRVTAB files to the router from the KDC, perform the following task in global configuration mode:

Task	Command
Retrieve a SRVTAB file from the KDC.	kerberos srvtab remote {hostname ip-address} {filename}

When you copy the SRVTAB file from the router to the KDC, the **kerberos srvtab remote** command parses the information in this file and stores it in the router's running configuration in the **kerberos srvtab entry** format. To ensure that the SRVTAB is available (does not need to be acquired from the KDC) when you reboot the router, use the **write memory** configuration command to write your running configuration (which contains the parsed SRVTAB file) to NVRAM.

For an example of copying SRVTAB files, see the "Copy SRVTAB Files Example" section at the end of this chapter.

Specify Kerberos Authentication

You have now configured Kerberos on your router. This makes it possible for the router to authenticate using Kerberos. But you have not yet told it to do so. To specify to the router to use Kerberos as the authentication method, perform the following task in global configuration mode:

Task	Command
Set AAA authentication at login using Kerberos.	aaa authentication login {default list-name} method1 [...[method4]]

Remote users logging in to the network are now prompted for a username. If the KDC has an entry for that user, it creates an encrypted ticket granting ticket (TGT) with the password for that user and sends it back to the router. The user is then prompted for a password, and the router attempts to decrypt the TGT with that password. If it succeeds, the user is authenticated and the TGT is stored in the user's credential cache on the router.

A user does not need to run the KINIT program to get a TGT to authenticate to the router. This is because KINIT has been integrated into the login procedure in the Cisco IOS implementation of Kerberos. However, you might want to configure some remote users to authenticate via Kerberos when they log in using PPP. In that case, they would never actually be on the router but use it only as a gateway to their workstations.

You can use Kerberos to authenticate PPP sessions using the Password Authentication Protocol (PAP). After configuring PPP for PAP authentication, perform the following task in global configuration mode to specify Kerberos as the method of PAP authentication:

Task	Command
Specify Kerberos as the method of authentication.	aaa authentication ppp [default list-name] method1 [...[method4]]

For an example of specifying Kerberos authentication, see the "Specify Kerberos Authentication Examples" section at the end of this chapter.

Enable Credentials Forwarding

With Kerberos configured thus far, a user authenticated to a Kerberized router has a TGT and can use it to authenticate to a host on the network. However, if the user tries to list credentials after authenticating to a host, the output will show no Kerberos credentials present.

You can optionally configure the router to forward users' TGTs with them as they authenticate from the router to Kerberized remote hosts on the network when using Kerberized Telnet, rcp, rsh, and rlogin (with the appropriate flags).

To force all clients to forward users' credentials as they connect to other hosts in the Kerberos realm, perform the following task in global configuration mode:

Task	Command
Force all clients to forward user credential upon successful Kerberos authentication.	kerberos credential forward

With credentials forwarding enabled, users' TGTs are automatically forwarded to the next host they authenticate to. In this way, users can connect to multiple hosts in the Kerberos realm without running the KINIT program each time to get a new TGT.

Telnet to the Router

You might want a way for users to open a secure Telnet session to the router to configure it. To use Kerberos to authenticate users opening Telnet session to the router from within the network, perform the following task in global configuration mode:

Task	Command
Set login authentication to use the Kerberos 5 Telnet authentication protocol when using Telnet to connect to the router.	aaa authentication login { default <i>list-name</i> } <i>method1</i> [... <i>method4</i>]

Although Telnet sessions to the router are authenticated, users must still enter a cleartext password if they want to enter enable mode. The **kerberos instance map** command, discussed in the next section, allows them to authenticate to the router at a predefined privilege level.

For an example of using Kerberized Telnet to open a secure session to the router, see the "Kerberized Telnet Example" section at the end of this chapter.

Enable Kerberos Instance Mapping

As mentioned in the section "Create SRVTABs on the KDC," you can create administrative instances of users in the KDC database. The **kerberos instance map** command allows you to map those instances to Cisco IOS privilege levels so that users can open secure Telnet sessions to the router at a predefined privilege level, obviating the need to enter a cleartext password to enter enable mode.

To map a Kerberos instance to Cisco a IOS privilege level, perform the following task in global configuration mode:

Task	Command
Map a Kerberos instance to a Cisco IOS privilege level.	kerberos instance map <i>instance privilege-level</i>

If there is a Kerberos instance for user *loki* in the KDC database (for example, *loki/admin*), user *loki* can now open a Telnet session to the router as *loki/admin* and authenticate automatically at privilege level 15. (See the section “Add Users to the KDC Database” earlier in this chapter.)

Cisco IOS commands can be set to various privilege levels using the **privilege level** command.

After you map a Kerberos instance to a Cisco IOS privilege level, you must configure the router to check for Kerberos instances each time a user logs in.

To configure the router to use a Kerberos instance as a method of authorization, perform the following task in global configuration mode:

Task	Command
Run authorization to determine if a user is allowed to run an EXEC shell.	aaa authorization {exec} method

Note Authorization is bypassed for authenticated users who log in using the console line, even if authorization has been configured.

For an example of how to enable Kerberos instance mapping, see the “Enable Kerberos Instance Mapping Examples” section at the end of this chapter.

Enable Mandatory Kerberos Authentication

As an added layer of security, you can optionally configure the router so that, after remote users authenticate to it, these users can authenticate to other services on the network only with Kerberized Telnet, rlogin, rsh, and rcp. If you do not make Kerberos authentication mandatory and Kerberos authentication fails, the application attempts to authenticate users using the default method of authentication for that network service; for example, Telnet and rlogin prompt for a password, rsh attempts to authenticate using the local rhost file.

To make Kerberos authentication mandatory, perform the following task in global configuration mode:

Task	Command
Set Telnet, rlogin, rsh, and rcp to fail if they cannot negotiate the Kerberos protocol with the remote server.	kerberos clients mandatory

Monitor and Maintain Kerberos

To display or remove a current user’s credentials, perform the following tasks in EXEC mode:

Task	Command
List the credentials in a current user’s credentials cache.	show kerberos creds
Destroy all credentials in a current user’s credentials cache.	clear kerberos creds

For an example of Kerberos configuration, see the “Kerberos Configuration Example” section at the end of this chapter.

Establish Terminal Access Control Using TACACS

The Terminal Access Controller Access Control System (TACACS) provides a way to centrally validate users attempting to gain access to a router or access server. Basic Cisco TACACS support is modeled after the original Defense Data Network (DDN) application. TACACS services are maintained in a database on a TACACS server running, typically, on a UNIX workstation. You must have access to and must configure a TACACS server before configuring the TACACS features on your Cisco router.

Cisco implements TACACS in the Cisco IOS software to allow centralized control over access to routers and access servers. Authentication can also be provided for Cisco IOS administration tasks on the router and access server user interfaces. With TACACS enabled, the router or access server prompts for a username and password, then verifies the password with a TACACS server.

This section describes the different Cisco IOS TACACS protocols and the various ways you can use TACACS to secure access to your network. Topics include the following:

- Comparative Analysis of TACACS Protocols
- Configure AAA/TACACS+
- Configure TACACS and Extended TACACS

Comparative Analysis of TACACS Protocols

The Cisco IOS software supports the following versions of the Terminal Access Controller/Access Control System (TACACS):

- AAA/TACACS+—Provides detailed accounting information and flexible administrative control over authentication and authorization processes.
- Extended TACACS—Provides information about protocol translator and router use. This information is used in UNIX auditing trails and accounting files.
- TACACS—Provides password checking and authentication, and notification of user actions for security and accounting purposes.

Table 5 identifies Cisco IOS commands available to the different versions of TACACS.

Table 5 TACACS Command Comparison

Cisco IOS Command	TACACS VERSIONS		
	TACACS	Extended TACACS	TACACS+
aaa accounting	–	–	Yes
aaa authentication arap	–	–	Yes
aaa authentication enable default	–	–	Yes
aaa authentication login	–	–	Yes
aaa authentication local override	–	–	Yes
aaa authentication ppp	–	–	Yes
aaa authorization	–	–	Yes
aaa new-model	–	–	Yes
arap authentication	–	–	Yes
arap use-tacacs	Yes	Yes	–

Table 5 TACACS Command Comparison (Continued)

Cisco IOS Command	TACACS VERSIONS		
	TACACS	Extended TACACS	TACACS+
enable last-resort	Yes	Yes	–
enable use-tacacs	Yes	Yes	–
ip tacacs source-interface	Yes	Yes	Yes
login authentication	–	–	Yes
login tacacs	Yes	Yes	–
ppp authentication	Yes	Yes	Yes
ppp use-tacacs	Yes	Yes	Yes
tacacs-server attempts	Yes	Yes	Yes
tacacs-server authenticate	Yes	Yes	–
tacacs-server directed-request	Yes	Yes	Yes
tacacs-server extended	–	Yes	–
tacacs-server host	Yes	Yes	Yes
tacacs-server key	–	–	Yes
tacacs-server last-resort	Yes	Yes	–
tacacs-server notify	Yes	Yes	–
tacacs-server optional-passwords	Yes	Yes	–
tacacs-server retransmit	Yes	Yes	Yes
tacacs-server timeout	Yes	Yes	Yes

Configure AAA/TACACS+

AAA/TACACS+ provides for separate and modular authentication, authorization, and accounting (AAA) facilities. TACACS+ allows for a single access control server (the TACACS+ server) to provide each service—authentication, authorization, and accounting—independently. Each service can be tied into its own database to take advantage of other services available on that server or on the network.

The goal of TACACS+ is to provide a methodology for managing dissimilar network access points from a single set of management services. The Cisco family of access servers and routers and the Cisco IOS user interface (for both routers and access servers) can be network access points.

Network access points enable traditional “dumb” terminals, terminal emulators, workstations, personal computers (PCs), and routers, to communicate using a serial line framing protocol such as Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), Compressed SLIP (CSLIP), or AppleTalk Remote Access Protocol (ARAP). In other words, a network access point provides connections to a single user, to a network or subnetwork, and to interconnected networks. The entities connected to the network through network access points are called *network access clients*; for example, a PC running PPP over a voice-grade circuit is a network access client. The benefits are more accurate accounting information and improved remote access functionality.

The AAA network security services perform the following functions:

- **Authentication**—Provides complete control of authentication through login and password dialog, challenge and response, messaging support, and encryption using the Message Digest 5 (MD5) encryption algorithm.

The authentication facility provides the ability, after a login and password are provided, to challenge a user with a number of questions; for example, home address, mother's maiden name, service request (SLIP, CSLIP, PPP, XRemote, TTY, ARAP, or TN3270), and city of birth. The TACACS+ administrator can improve the integrity of the authentication dialog by routinely changing the challenge questions. The TACACS+ authentication service is flexible enough to send messages to user screens. For example, a message might instruct users that their passwords must be changed because of the company's password aging policy.

- **Authorization**—Provides remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, IPX, ARA, and Telnet. Additionally, you can create access or command permissions and restrictions.

The authorization component in TACACS+ allows for greater levels of control over user actions after users have been authenticated and can be used to create separate administrative groups based on user functionality. With the TACACS+ authorization facility, a network manager can restrict a user to a subset of functions on the router user interface by using autocommands. Autocommands are user interface commands tied to user profiles. For example, when certain users are authenticated, their profiles can automatically create a Telnet session to a specific host.

- **Accounting**—Collects and sends the TACACS server information used for billing, auditing, and reporting.

Network managers can use the accounting facility to track user activity for a security audit or to provide information for user billing. A report can be structured to provide user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes.

The TACACS+ protocol provides authentication between the network access points and the TACACS+ server, and it ensures the confidentiality of packets. This is accomplished by never sending sensitive user information (such as passwords) over the network in cleartext. Cisco provides a general-purpose TACACS+ protocol specification for integrating TACACS+ servers with third-party or proprietary authentication, authorization, and accounting services. The intention of this protocol specification is to provide access to future services, (such as Kerberos authentication, token password cards, and so on) that can be provided by a third party. TACACS+ is not constrained by a single mode of access; it is supported over SLIP, CSLIP, XRemote, PPP, ARAP, TN3270, X.25, and dumb terminals (TTYs).

You need a server running TACACS software to use the AAA/TACACS+ functionality on your router. To find out how to specify a TACACS server host, see the section "Establish the TACACS Server Host" later in this chapter. You can obtain AAA/TACACS+ free of charge from Cisco, or purchase software from a third-party vendor.

Note Many original TACACS and extended TACACS commands cannot be used after you initialize AAA/TACACS+. To identify the commands that can be used with the three versions of TACACS, refer to Table 5 earlier in this chapter.

The following sections describe the features available in AAA/TACACS+:

- Enable AAA/TACACS+ and Set Authentication Key.
- Enable Authentication for ARA.

- Enable TACACS+ Password Protection at the Privileged Level.
- Enable Authentication for Login.
- Enable an Authentication Override.
- Enable Authentication for PPP.
- Restrict Network Access.
- Specify TACACS+ Authorization for EXEC Access and Network Services.
- Start TACACS+ Accounting.

In addition, the following features from the previous versions of TACACS are also available in TACACS+. (Refer to the section “Configure TACACS and Extended TACACS” later in this chapter.)

- Establish the TACACS Server Host
- Set Limits on Login Attempts
- Specify the Amount of Time for Login Input

Enable AAA/TACACS+ and Set Authentication Key

To enable AAA/TACACS+, perform the following tasks in global configuration mode:

Task	Command
Enable AAA/TACACS+.	aaa new-model
Set the authentication and encryption key to the same key used on the TACACS+ daemon.	tacacs-server key <i>key</i>

Enable Authentication for ARA

With the **aaa authentication arap** command, you create one or more lists of authentication methods that are tried when ARA users attempt to log in to the router. These lists are used with the **arap authentication** line configuration command.

The *list-name* is any character string used to name the list you are creating. The *method* refers to the actual list of methods the authentication algorithm tries, in the sequence entered. You can enter up to four methods.

To create a default list that is used if no list is specified in the **arap authentication** command, use the **default** argument followed by the methods you want to be used in default situations.

The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

Note By default, guest logins through ARAP are disabled when you initialize AAA/TACACS+. To allow guest logins, you must use either the **guest** or **auth-guest** methods described in the “System Management Commands” chapter in the *Configuration Fundamentals Command Reference* publication.

Perform at least the first of the following tasks starting in global configuration mode:

Task	Command
Enable authentication for ARA users.	aaa authentication arap { default <i>list-name</i> } <i>method1</i> [... <i>method4</i>]
(Optional) Change to line configuration mode.	line <i>number</i>
(Optional) Enable autoselection of ARA.	autoselect arap ¹
(Optional) Start the ARA session automatically at user login.	autoselect during-login ¹
(Optional—not needed if default is used in the aaa authentication arap command) Enable TACACS+ authentication for ARA on a line.	arap authentication <i>list-name</i>

1. This command is documented in the “Terminal Lines and Modem Support Commands” chapter of the *Access Services Command Reference*.

By using the optional **during-login** argument with the **autoselect** command, you can display the username or password prompt without pressing the Return key. When the username or password name is displayed, you can choose to answer these prompts, or to start sending packets from an autoselected protocol.

For an example of enabling authentication for ARA, see the “TACACS+ Authentication with ARA Examples” section at the end of this chapter.

The following table lists the supported login authentication methods.

Keyword	Description
guest	Allows guest logins.
auth-guest	Allows guest logins only if the user has already logged into EXEC.
line	Uses the line password for authentication.
local	Uses the local username database for authentication.
tacacs+	Uses TACACS+ authentication.
radius	Uses RADIUS authentication.

For example, to create a default AAA authentication method list used with the ARA protocol, enter:

```
aaa authentication arap default if-needed none
```

To create the same authentication method list for the ARA protocol but name the list *MIS-access*, enter:

```
aaa authentication arap MIS-access if-needed none
```

Enable TACACS+ Password Protection at the Privileged Level

Use the **aaa authentication enable default** command to create a series of authentication methods that are used to determine whether a user can access privileged EXEC command level. You can specify up to four authentication methods. The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication succeed even if all methods return an error, specify **none** as the final method in the command line.

Perform the following task in global configuration mode:

Task	Command
Enable TACACS+ user ID and password checking for users requesting privileged EXEC level.	aaa authentication enable default <i>method1</i> [... <i>method4</i>]

Enable Authentication for Login

With the **aaa authentication login** command, you create one or more lists of authentication methods that are tried at login. These lists are used with the **login authentication** line configuration command.

The keyword *list-name* is any character string used to name the list you are creating. The *method* keyword refers to the actual method the authentication algorithm tries, in the sequence entered. You can enter up to four methods.

To create a default list that is used if no list is specified in the **login authentication** command, use the **default** argument followed by the methods you want used in default situations.

The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication succeed even if all methods return an error, specify **none** as the final method in the command line.

Perform the following task in global configuration mode:

Task	Command
Enable AAA authentication at login.	aaa authentication login { default <i>list-name</i> } <i>method1</i> [... <i>method4</i>]

Enable an Authentication Override

To configure the Cisco IOS software to check the local user database for authentication before attempting another form of authentication, use the **aaa authentication local-override** command. This command is useful when you want to configure an override to the normal authentication process for certain personnel (such as system administrators).

Perform the following task in global configuration mode:

Task	Command
Create an override for authentication.	aaa authentication local-override

Enable Authentication for PPP

With the **aaa authentication ppp** command, you create one or more lists of authentication methods that are tried during PPP sessions. These lists are used with the **ppp authentication** line configuration command.

The keyword *list-name* is any character string used to name the list you create. The *method* keyword refers to the actual list of methods the authentication algorithm tries, in the sequence entered. You can enter up to four methods that this list tries in sequence.

To create a default list that is used if no list is specified in the **ppp authentication** command, use the **default** argument followed by the methods you want used in default situations.

The additional methods of authentication are used only if the previous method returns an error, not if it fails. To specify that the authentication should succeed even if all methods return an error, specify **none** as the final method in the command line.

Perform the following task in global configuration mode:

Task	Command
Enable AAA authentication for PPP.	aaa authentication ppp { default <i>list-name</i> } <i>method1</i> [... <i>method4</i>]

Restrict Network Access

With the **aaa authorization** command, you create a list of one and up to four authorization methods that are used when a user accesses the specified function.

The additional methods of authorization are used only if the previous method returns an error, not if it fails. To specify that the authorization succeed even if all methods return an error, specify **none** as the final method in the command line.

Perform the following task in global configuration mode:

Task	Command
Restrict network access using AAA.	aaa authorization { network exec command <i>level</i> } <i>method...</i>

Note Authorization is bypassed for authenticated users who log in using the console line, even if authorization has been configured.

For an example of the **aaa authorization** command, and for an example of how to use address pooling with this command, see the section “Restrict Network Access Examples” at the end of this chapter.

Specify TACACS+ Authorization for EXEC Access and Network Services

You use the **aaa authorization** command with the **tacacs+** keyword to set parameters that restrict a user’s network access.

To specify TACACS+ authorization for EXEC access and network services, perform the following tasks in global configuration mode:

Task	Command
User TACACS+ authorization for all network-related service requests, including SLIP, PPP NCPs, and ARA protocol.	aaa authorization exec tacacs+
User TACACS+ authorization to determine if the user is allowed to run an EXEC shell. This keyword might return user profile information (such as autocommand information).	aaa authorization network tacacs+

Note Authorization is bypassed for authenticated users who log in using the console line, even if authorization has been configured.

The **aaa authorization exec tacacs+ local** command sets the following authorization parameters:

- Use TACACS+ for EXEC level authorization if authentication was done using TACACS+.
- Use the local database if authentication was not done using TACACS+.

For an example of how to specify TACACS+ authorization, see the “TACACS+ Authorization Example” section at the end of this chapter.

Start TACACS+ Accounting

You use the **aaa accounting** command with the **tacacs+** keyword to turn on TACACS+ accounting for each Cisco IOS privilege level, and network services.

To use TACACS+ accounting to send a start record accounting notice at the beginning of an EXEC process and a stop record at the end, perform the following task in global configuration mode:

Task	Command
Turn on TACACS+ accounting for the EXEC session.	aaa accounting exec start-stop tacacs+

To use TACACS+ to account for all network-related service requests, including SLIP, PPP, and PPP NCPs, perform the following task in global configuration mode:

Task	Command
Use TACACS+ accounting for network-related service requests.	aaa accounting network start-stop tacacs+¹

1. This command is documented in the “Accounting and Billing Commands” chapter of the *Security Command Reference*.

Note No TACACS+-specific **show** commands exist. You can use the **show accounting** command to display accounting information.

TACACS+ AV Pairs

Table 6 lists the supported TACACS+ AV pairs.

Table 6 Supported TACACS+ AV Pairs

Attribute	Description	Cisco IOS Release 11.0	Cisco IOS Release 11.1	Cisco IOS Release 11.2	Cisco IOS Release 11.3
service=x	The primary service. Specifying a service attribute indicates that this is a request for authorization or accounting of that service. Current values are slip , ppp , arap , shell , tty-daemon , connection , and system . This attribute must always be included.	yes	yes	yes	yes
protocol=x	A protocol that is a subset of a service. An example would be any PPP NCP. Currently known values are lcp , ip , ipx , atalk , vines , lat , xremote , tn3270 , telnet , rlogin , pad , vpdn , osicp , deccp , ccp , cdp , bridging , xns , nbf , bap , multilink , and unknown .	yes	yes	yes	yes
cmd=x	A shell (EXEC) command. This indicates the command name for a shell command that is to be run. This attribute must be specified if service equals "shell." A NULL value indicates that the shell itself is being referred to.	yes	yes	yes	yes
cmd-arg=x	An argument to a shell (EXEC) command. This indicates an argument for the shell command that is to be run. Multiple cmd-arg attributes may be specified, and they are order dependent.	yes	yes	yes	yes
acl=x	ASCII number representing a connection access list. Used only when service=shell.	yes	yes	yes	yes
inacl=x	ASCII identifier for an interface input access list. Used with service=ppp and protocol=ip. Per-user access lists do not currently work with ISDN interfaces.	yes	yes	yes	yes

Table 6 Supported TACACS+ AV Pairs (Continued)

Attribute	Description	Cisco IOS Release 11.0	Cisco IOS Release 11.1	Cisco IOS Release 11.2	Cisco IOS Release 11.3
inac1#<n>	ASCII access list identifier for an input access list to be installed and applied to an interface for the duration of the current connection. Used with service=ppp and protocol=ip, and service service=ppp and protocol =ipx. Per-user access lists do not currently work with ISDN interfaces.	no	no	no	yes
outac1=x	ASCII identifier for an interface output access list. Used with service=ppp and protocol=ip, and service service=ppp and protocol=ipx. Contains an IP output access list for SLIP or PPP/IP (for example, outac1=4). The access list itself must be preconfigured on the router. Per-user access lists do not currently work with ISDN interfaces.	yes (PPP/IP only)	yes	yes	yes
outac1#<n>	ASCII access list identifier for an interface output access list to be installed and applied to an interface for the duration of the current condition. Used with service=ppp and protocol=ip, and service service=ppp and protocol=ipx. Per-user access lists do not currently work with ISDN interfaces.	no	no	no	yes
zonelist=x	A numeric zonelist value. Used with service=arap. Specifies an AppleTalk zonelist for ARA (for example, zonelist=5).	yes	yes	yes	yes
addr=x	A network address. Used with service=slip, service=ppp, and protocol=ip. Contains the IP address that the remote host should use when connecting via SLIP or PPP/IP. For example, addr=10.2.3.4.	yes	yes	yes	yes

Table 6 Supported TACACS+ AV Pairs (Continued)

Attribute	Description	Cisco IOS Release 11.0	Cisco IOS Release 11.1	Cisco IOS Release 11.2	Cisco IOS Release 11.3
addr-pool=x	<p>Specifies the name of a local pool from which to get the address of the remote host. Used with <code>service=ppp</code> and <code>protocol=ip</code>.</p> <p>Note that addr-pool works in conjunction with local pooling. It specifies the name of a local pool (which must be preconfigured on the network access server). Use the ip-local pool command to declare local pools. For example:</p> <pre>ip address-pool local ip local pool boo 10.0.0.1 10.0.0.10 ip local pool moo 10.0.0.1 10.0.0.20</pre> <p>You can then use TACACS+ to return <code>addr-pool=boo</code> or <code>addr-pool=moo</code> to indicate the address pool from which you want to get this remote node's address.</p>	yes	yes	yes	yes
routing=x	<p>Specifies whether routing information is to be propagated to and accepted from this interface. Used with <code>service=slip</code>, <code>service=ppp</code>, and <code>protocol=ip</code>. Equivalent in function to the <code>/routing</code> flag in SLIP and PPP commands. Can either be true or false (for example, <code>routing=true</code>).</p>	yes	yes	yes	yes
route	<p>Specifies a route to be applied to an interface. Used with <code>service=slip</code>, <code>service=ppp</code>, and <code>protocol=ip</code>.</p> <p>During network authorization, the route attribute can be used to specify a per-user static route, to be installed by TACACS+ as follows:</p> <pre>route="dst_address mask [gateway]"</pre> <p>This indicates a temporary static route that is to be applied. The <i>dst_address</i>, <i>mask</i>, and <i>gateway</i> are expected to be in the usual dotted-decimal notation, with the same meanings as in the familiar ip route configuration command on a network access server.</p> <p>If <i>gateway</i> is omitted, the peer's address is the gateway. The route is expunged when the connection terminates.</p>	no	yes	yes	yes

Table 6 Supported TACACS+ AV Pairs (Continued)

Attribute	Description	Cisco IOS Release 11.0	Cisco IOS Release 11.1	Cisco IOS Release 11.2	Cisco IOS Release 11.3
route#<n>	Like the route AV pair, this specifies a route to be applied to an interface, but these routes are numbered, allowing multiple routes to be applied. Used with service=ppp and protocol=ip, and service=ppp and protocol=ipx.	no	no	no	yes
timeout=x	The number of minutes before an EXEC or ARA session disconnects (for example, timeout=60). A value of zero indicates no timeout. Used with service=arap.	yes	yes	yes	yes
idletime=x	Sets a value, in minutes, after which an idle session is terminated. Does not work for PPP. A value of zero indicates no timeout.	no	yes	yes	yes
autocmd=x	Specifies an autocommand to be executed at EXEC startup (for example, autocmd=telnet muruga.com). Used only with service=shell.	yes	yes	yes	yes
noescape=x	Prevents user from using an escape character. Used with service=shell. Can be either true or false (for example, noescape=true).	yes	yes	yes	yes
nohangup=x	Used with service=shell. Specifies the nohangup option, which means that after an EXEC shell is terminated, the user is presented with another login (username) prompt. Can be either true or false (for example, nohangup=false).	yes	yes	yes	yes
priv-lvl=x	Privilege level to be assigned for the EXEC. Used with service=shell. Privilege levels range from 0 to 15, with 15 being the highest.	yes	yes	yes	yes
callback-dialstring	Sets the telephone number for a callback (for example: callback-dialstring=408-555-1212). Value is NULL, or a dial-string. A NULL value indicates that the service may choose to get the dialstring through other means. Used with service=arap, service=slip, service=ppp, service=shell. Not valid for ISDN.	no	yes	yes	yes
callback-line	The number of a TTY line to use for callback (for example: callback-line=4). Used with service=arap, service=slip, service=ppp, service=shell. Not valid for ISDN.	no	yes	yes	yes

Table 6 Supported TACACS+ AV Pairs (Continued)

Attribute	Description	Cisco IOS Release 11.0	Cisco IOS Release 11.1	Cisco IOS Release 11.2	Cisco IOS Release 11.3
callback-rotary	The number of a rotary group (between 0 and 100 inclusive) to use for callback (for example: callback-rotary=34). Used with service=arap, service=slip, service=ppp, service=shell. Not valid for ISDN.	no	yes	yes	yes
nocallback-verify	Indicates that no callback verification is required. The only valid value for this parameter is 1 (for example, nocallback-verify=1). Used with service=arap, service=slip, service=ppp, service=shell. There is no authentication on callback. Not valid for ISDN.	no	yes	yes	yes
tunnel-id	Specifies the username that will be used to authenticate the tunnel over which the individual user MID will be projected. This is analogous to the <i>remote name</i> in the vpdn outgoing command. Used with service=ppp and protocol=vpdn.	no	no	yes	yes
ip-addresses	Space-separated list of possible IP addresses that can be used for the end-point of a tunnel. Used with service=ppp and protocol=vpdn.	no	no	yes	yes
nas-password	Specifies the password for the network access server during the L2F tunnel authentication. Used with service=ppp and protocol=vpdn.	no	no	yes	yes
gw-password	Specifies the password for the home gateway during the L2F tunnel authentication. Used with service=ppp and protocol=vpdn.	no	no	yes	yes
rte-ftp-in#<n>	Specifies an input access list definition to be installed and applied to routing updates on the current interface for the duration of the current connection. Used with service=ppp and protocol=ip, and with service=ppp and protocol=ipx.	no	no	no	yes
rte-ftp-out#<n>	Specifies an output access list definition to be installed and applied to routing updates on the current interface for the duration of the current connection. Used with service=ppp and protocol=ip, and with service=ppp and protocol=ipx.	no	no	no	yes
sap#<n>	Specifies static Service Advertising Protocol (SAP) entries to be installed for the duration of a connection. Used with service=ppp and protocol=ipx.	no	no	no	yes

Table 6 Supported TACACS+ AV Pairs (Continued)

Attribute	Description	Cisco IOS Release 11.0	Cisco IOS Release 11.1	Cisco IOS Release 11.2	Cisco IOS Release 11.3
sap-fltr-in#<n>	Specifies an input SAP filter access list definition to be installed and applied on the current interface for the duration of the current connection. Used with service=ppp and protocol=ipx.	no	no	no	yes
sap-fltr-out#<n>	Specifies an output SAP filter access list definition to be installed and applied on the current interface for the duration of the current connection. Used with service=ppp and protocol=ipx.	no	no	no	yes
pool-def#<n>	Used to define IP address pools on the network access server. Used with service=ppp and protocol=ip.	no	no	no	yes
source-ip=x	Used as the source IP address of all VPDN packets generated as part of a VPDN tunnel. This is equivalent to the Cisco vpdn outgoing global configuration command.	no	no	yes	yes
max-links=<n>	Restricts the number of links that a user can have in a multilink bundle. Used with service=ppp and protocol=multilink. The range for <n> is from 1 to 255.	no	no	no	yes
load-threshold=<n>	Sets the load threshold at which additional links are either added to or deleted from the multilink bundle. If the load goes above the specified value, additional links are added. If the load goes below the specified value, links are deleted. Used with service=ppp and protocol=multilink. The range for <n> is from 1 to 255.	no	no	no	yes

Table 7 lists the supported TACACS+ accounting AV pairs.

Table 7 Supported TACACS+ Accounting AV Pairs

Attribute	Description	Cisco IOS Release 11.0	Cisco IOS Release 11.1	Cisco IOS Release 11.2
service	The service the user used.	yes	yes	yes
port	The port the user was logged in to.	yes	yes	yes
task_id	Start and stop records for the same event must have matching (unique) task_id's.	yes	yes	yes

Table 7 Supported TACACS+ Accounting AV Pairs

Attribute	Description	Cisco IOS Release 11.0	Cisco IOS Release 11.1	Cisco IOS Release 11.2
start_time	The time the action started (in seconds since the epoch, 12:00 a.m. Jan 1 1970). The clock must be configured to receive this information.	yes	yes	yes
stop_time	The time the action stopped (in seconds since the epoch.) The clock must be configured to receive this information.	yes	yes	yes
elapsed_time	The elapsed time in seconds for the action. Useful when the device does not keep real time.	yes	yes	yes
timezone	The timezone abbreviation for all timestamps included in this packet.	yes	yes	yes
priv_level	The privilege level associated with the action.	yes	yes	yes
cmd	The command the user executed.	yes	yes	yes
protocol	The protocol associated with the action.	yes	yes	yes
bytes_in	The number of input bytes transferred during this connection.	yes	yes	yes
bytes_out	The number of output bytes transferred during this connection.	yes	yes	yes
paks_in	The number of input packets transferred during this connection.	yes	yes	yes
paks_out	The number of output packets transferred during this connection.	yes	yes	yes
event	Information included in the accounting packet that describes a state change in the router. Events described are accounting starting and accounting stopping.	yes	yes	yes
reason	Information included in the accounting packet that describes the event that caused a system change. Events described are system reload, system shutdown, or when accounting is reconfigured (turned on or off).	yes	yes	yes

Configure TACACS and Extended TACACS

You can establish TACACS-style password protection on both user and privileged levels of the system EXEC.

The following sections describe the features available with TACACS and extended TACACS. The extended TACACS software is available using the File Transfer Protocol (FTP)—see the README file in the *ftp.cisco.com* directory.

Note Many original TACACS and extended TACACS commands cannot be used once you have initialized AAA/TACACS+. To identify which commands can be used with the three versions, refer to Table 5 earlier in this chapter.

- Set TACACS Password Protection at the User Level.
- Disable Password Checking at the User Level.
- Set Optional Password Verification.
- Set TACACS Password Protection at the Privileged Level.
- Disable Password Checking at the Privileged Level.
- Set Notification of User Actions.
- Set Authentication of User Actions.
- Establish the TACACS Server Host.
- Set Limits on Login Attempts.
- Enable the Extended TACACS Mode.
- Enable TACACS for PPP Authentication.
- Enable Standard TACACS for ARA Authentication.
- Enable Extended TACACS for ARA Authentication.
- Enable TACACS to Use a Specific IP Address.

Note If you require additional security using TCP/IP access lists, see the “Configuring IP” chapter in the *Network Protocols Configuration Guide, Part 1* for more information.

Set TACACS Password Protection at the User Level

To enable password checking at login, perform the following task in line configuration mode:

Task	Command
Set the TACACS-style user ID and password-checking mechanism.	login tacacs¹

1. This command is documented in the “Terminal Line and Modem Support Commands” chapter of the *Access Services Command Reference*.

Disable Password Checking at the User Level

If a TACACS server does not respond to a login request, the Cisco IOS software denies the request by default. However, you can prevent that login failure in one of the following two ways.

- You can allow a user to access privileged EXEC mode if that user enters the password set by the **enable** command.
- Or you can ensure a successful login by allowing the user to access the privileged EXEC mode without further question.

To specify one of these features, perform either of the following tasks in global configuration mode:

Task	Command
Allow a user to access privileged EXEC mode.	tacacs-server last-resort password
Set last resort options for logins.	tacacs-server last-resort succeed

Set Optional Password Verification

You can specify that the first TACACS request to a TACACS server is made without password verification. To do so, perform the following task in global configuration mode:

Task	Command
Set TACACS password as optional.	tacacs-server optional-passwords

When the user enters in the login name, the login request is transmitted with the name and a zero-length password. If accepted, the login procedure is completed. If the TACACS server refuses this request, the terminal server prompts for a password and tries again when the user supplies a password. The TACACS server must support authentication for users without passwords to make use of this feature. This feature supports all TACACS requests such as login, SLIP, and enable.

Set TACACS Password Protection at the Privileged Level

You can set the TACACS protocol to determine whether a user can access the privileged EXEC level. To do so, perform the following task in global configuration mode:

Task	Command
Set the TACACS-style user ID and password-checking mechanism at the privileged EXEC level.	enable use-tacacs

When you set TACACS password protection at the privileged EXEC level, the EXEC **enable** command will ask for both a new username and a password. This information is then passed to the TACACS server for authentication. If you are using the extended TACACS, it also passes any existing UNIX user identification code to the server.



Caution If you use the **enable use-tacacs** command, you must also specify **tacacs-server authenticate enable**; otherwise, you will be locked out.

Note When used without extended TACACS, this task allows anyone with a valid username and password to access the privileged command level, creating a potential security problem. This is because the TACACS query resulting from entering the **enable** command is indistinguishable from an attempt to log in without extended TACACS.

Disable Password Checking at the Privileged Level

You can specify a last resort if the TACACS servers used by the **enable** command do not respond. To invoke this “last resort” login feature, perform either of the following tasks in global configuration mode:

Task	Command
Allow user to enable by asking for the privileged EXEC-level password.	enable last-resort password
Allow user to enable without further questions.	enable last-resort succeed

Set Notification of User Actions

The **tacacs-server notify** command allows you to configure the TACACS server to send a message when a user does the following:

- Makes a TCP connection
- Enters the **enable** command
- Logs out

To specify that the TACACS server send notification, perform the following task in global configuration mode:

Task	Command
Set server notification of user actions.	tacacs-server notify { connection [always] enable logout [always] slip [always] }

The retransmission of the message is performed by a background process for up to 5 minutes. The terminal user, however, receives an immediate response, allowing access to the terminal.

The **tacacs-server notify** command is available only if you have set up an extended TACACS server using the latest Cisco extended TACACS server software, available via FTP. (See the README file in the *ftp.cisco.com* directory.)

Set Authentication of User Actions

For a SLIP or PPP session, you can specify that if a user tries to start a session, the TACACS software requires a response (either from the TACACS server host or the router) indicating whether the user can start the session. You can specify that the TACACS software perform authentication even when a user is not logged in; you can also request that the TACACS software install access lists.

If a user issues the **enable** command, the TACACS software must respond indicating whether the user can give the command. You can also specify authentication when a user issues the **enable** command.

To configure any of these scenarios, perform the following task in global configuration mode:

Task	Command
Set server authentication of user actions.	tacacs-server authenticate { connection[always] enable slip [always] [access-lists] }

The **tacacs-server authenticate** command is available only when you have set up an extended TACACS server using the latest Cisco extended TACACS server software, which is available via FTP. (See the README file in the *ftp.cisco.com* directory).

Establish the TACACS Server Host

The **tacacs-server host** command allows you to specify the names of the IP host or hosts maintaining a AAA/TACACS+ server. Because the TACACS software searches for the hosts in the order specified, this feature can be useful for setting up a list of preferred servers.

On AAA/TACACS+ servers, you can configure the following additional options:

- Specify **single-connection** (CiscoSecure Release 1.0.1 or later). Rather than have the router open and close a TCP connection to the server each time it must communicate, the single-connection option maintains a single open connection between the router and the daemon. This is more efficient, allowing the daemon to handle a higher number of TACACS operations.
- Specify the port number on the server.
- Specify the period of time (in seconds) the router attempts to contact the server before it times out.
- Specify an encryption key to encrypt and decrypt all traffic between the router and the daemon.

With TACACS and extended TACACS, the **tacacs-server retransmit** command allows you to modify the number of times the system software searches the list of TACACS servers (from the default of two times) and the interval it waits for a reply (from the default of 5 seconds).

To define the number of times the Cisco IOS software searches the list of servers, and how long the server waits for a reply, perform the following tasks as needed for your system configuration in global configuration mode:

Task	Command
Specify a AAA/TACACS+ host.	tacacs-server host <i>name</i> [single-connection] [port <i>integer</i>] [timeout <i>integer</i>] [key <i>string</i>]
Specify the number of times the server will search the list of TACACS and extended TACACS server hosts before giving up.	tacacs-server retransmit <i>retries</i>
Set the interval the server waits for a TACACS and extended TACACS server host to reply.	tacacs-server timeout <i>seconds</i>

Set Limits on Login Attempts

The **tacacs-server attempts** command allows you to specify the number of login attempts that can be made on a line set up for TACACS. Perform the following task in global configuration mode to limit login attempts:

Task	Command
Control the number of login attempts that can be made on a line set for TACACS verification.	tacacs-server attempts <i>count</i>

Specify the Amount of Time for Login Input

The **tacacs-server login-timeout** command allows you to specify how long the system will wait for login input (such as username and password) before timing out. The default login value is 30 seconds; with the **tacacs-server login-timeout** command, you can specify a timeout value from 1 to 300 seconds. Perform the following task in global configuration mode to change the login timeout value from the default of 30 seconds:

Task	Command
Specify how long the system will wait for login information before timing out.	tacacs-server login-timeout <i>seconds</i>

Enable the Extended TACACS Mode

While standard TACACS provides only username and password information, extended TACACS mode provides information about the terminal requests to help set up UNIX auditing trails and accounting files for tracking the use of protocol translators, access servers, and routers. The information includes responses from these network devices and validation of user requests.

An unsupported, extended TACACS server is available via FTP for UNIX users who want to create the auditing programs (see the README file in the *ftp.cisco.com* directory).

To enable extended TACACS mode, perform the following task in global configuration mode:

Task	Command
Enable an extended TACACS mode.	tacacs-server extended

Enable TACACS for PPP Authentication

You can use extended TACACS for authentication within PPP sessions. To do so, perform the following steps in interface configuration mode:

Task	Command
Step 1 Enable CHAP or PAP.	ppp authentication {chap chap pap pap chap pap} [if-needed] [list-name / default] [callin]
Step 2 Enable TACACS under PPP.	ppp use-tacacs [single-line]

For more information on PPP, refer to the “Configuring Interfaces” chapter in the *Configuration Fundamentals Configuration Guide*. For an example of enabling TACACS for PPP protocol authentication, see the “TACACS Authentication Examples” section at the end of this chapter.

Enable Standard TACACS for ARA Authentication

You can use the Standard TACACS protocol for authentication within AppleTalk Remote Access (ARA) protocol sessions. To do so, perform the following tasks starting in line configuration mode:

Task	Command
Enable standard TACACS under the ARA protocol.	arap use-tacacs single-line¹
Enable autoselection of ARA.	autoselect arap²
(Optional) Have the ARA session start automatically at user login.	autoselect during-login²

1. This command is documented in the “AppleTalk Commands” chapter of the *Network Protocols Command Reference, Part 2*.

2. This command is documented in the “Terminal Lines and Modem Support Commands” chapter of the *Access Services Command Reference*.

By using the optional **during-login** argument with the **autoselect** command, you can display the username or password prompt without pressing the Return key. While the username or password name is displayed, you can choose to answer these prompts or to start sending packets from an autoselected protocol.

The remote user logs in through ARA as follows:

Step 1 When prompted for a username by the ARA application, the remote user enters *username*password* and presses Return.

Step 2 When prompted for password by the ARA application, the remote user enters **arap** and presses Return.

For more information on the ARA protocol, refer to the “Configuring an AppleTalk Remote Access Server” chapter in the *Access Services Configuration Guide*. For examples of enabling TACACS for ARA protocol authentication, see the “TACACS Authentication Examples” section at the end of this chapter.

Enable Extended TACACS for ARA Authentication

You can use extended TACACS for authentication within AppleTalk Remote Access (ARA) protocol sessions. The extended TACACS server software is available via FTP (see the README file in the *ftp.cisco.com* directory).

Note Before issuing the commands listed in the following task table, you must edit the file called “Makefile” in the extended TACACS server software to use ARA. To do this, you must uncomment the lines that enable ARA support and recompile the file.

After installing an extended TACACS server with ARA support, perform the following tasks in line configuration mode on each line:

Task	Command
Enable extended TACACS under the ARA protocol on each line.	arap use-tacacs ¹
(Optional) Enable autoselection of ARA.	autoselect arap ²
(Optional) Have the ARA session start automatically at user login.	autoselect during-login ²

1. This command is documented in the “AppleTalk Remote Access Commands” chapter of the *Access Services Command Reference*.

2. This command is documented in the “Terminal Lines and Modem Support Commands” chapter of the *Access Services Command Reference*.

By using the optional **during-login** argument with the **autoselect** command, you can display the username or password prompt without pressing the Return key. While the Username or Password name is being presented, you can choose to answer these prompts, or to start sending packets from an autoselected protocol.

For more information on the ARA protocol, refer to the “Configuring an AppleTalk Remote Access Server” chapter in the *Access Services Configuration Guide*. For examples of enabling TACACS for ARA protocol authentication, see the “TACACS+ Authentication with ARA Examples” section at the end of this chapter.

Enable TACACS to Use a Specific IP Address

You can designate a fixed source IP address for all outgoing TACACS packets. The feature enables TACACS to use the IP address of a specified interface for all outgoing TACACS packets. This is especially useful if the router has many interfaces, and you want to make sure that all TACACS packets from a particular router have the same IP address.

To enable TACACS to use the address of a specified interface for all outgoing TACACS packets, perform the following task in configuration mode:

Task	Command
Enable TACACS to use the IP address of a specified interface for all outgoing TACACS packets.	ip tacacs source-interface <i>subinterface-name</i>

Additional AAA Authorization Features

Establish Username Authentication

You can create a username-based authentication system, which is useful in the following situations:

- To provide a TACACS-like username and encrypted password-authentication system for networks that cannot support TACACS.
- To provide special-case logins; for example, access list verification, no password verification, autocommand execution at login, and “no escape” situation.

To establish username authentication, perform the following tasks in global configuration mode as needed for your system configuration:

Task	Command
Establish username authentication with encrypted passwords. or Establish username authentication by access list.	username <i>name</i> [no password password <i>encryption-type</i> password] username <i>name</i> [access-class <i>number</i>]
Set the privilege level for the user.	username <i>name</i> privilege <i>level</i>
Specify a command to automatically execute.	username <i>name</i> [autocommand <i>command</i>]
Set a “no escape” login environment.	username <i>name</i> [noescape] [nohangup]

The keyword **noescape** prevents users from using escape characters on the hosts to which they are connected.

Enable CHAP

Access control using Challenge Handshake Authentication Protocol (CHAP), which is available on all serial interfaces that use PPP encapsulation, reduces the risk of security violations on your router.

When CHAP is enabled, a remote device (a PC, workstation, router, access server, or communication server) attempting to connect to the local router is requested (or “challenged”) to respond.

The challenge consists of an ID, a random number, and either the host name of the local device or the name of the user on the remote device. This challenge is transmitted to the remote device.

The required response consists of the following two parts:

- An encrypted version of the ID, a secret password (called the *secret*), and the random number
- Either the host name of the remote device or the name of the user on the remote device

When the local device receives the challenge response, it verifies the secret by looking up the name given in the response and performing the same encryption operation. The secret passwords must be identical on the remote and local devices.

Because the secret is never transmitted, other devices are prevented from stealing it and gaining illegal access to the system. Without the proper response, the remote device cannot connect to the local device.

CHAP transactions occur only when a link is established. The local device does not request a password during the remainder of the call. (The local device can, however, respond to such requests from other devices during a call.)

You can create a pool of dialup routers that all appear to be the same host when authenticating with CHAP. Currently, a router dialing a pool of access routers requires a username entry for each possible router in the pool because each router challenges with its hostname. If a router is added to the dialup rotary pool, all connecting routers must be updated. The **ppp chap hostname** command allows you to specify a common alias for all routers in a rotary group to use so that only one username must be configured on the dialing routers.

To use CHAP authentication, and to create a pool of dialup routers that all appear to be the same host when authentication with CHAP, perform the following tasks in interface configuration mode:

Task	Command
Enable CHAP on the interface.	ppp authentication chap [if-needed] or ppp authentication chap [list-name]
Create a pool of dialup routers that all appear to be the same host when authentication with CHAP.	ppp chap hostname hostname

The optional keyword **if-needed** in the **ppp authentication chap** command can be used only with TACACS or extended TACACS. The optional argument *list-name* can be used only with AAA/TACACS+. CHAP is specified in RFC 1334. It is an additional authentication phase of the PPP Link Control Protocol.

After you enable CHAP, the local device requires a response from the remote devices. If the remote device does not support CHAP, no traffic is passed to that device.

For remote CHAP authentication only, you can configure your router to create a common CHAP secret password to use in response to challenges from an unknown peer; for example, if your router calls a rotary of routers (either from another vendor, or running an older version of the Cisco IOS software) to which a new (that is, unknown) router has been added. The **ppp chap password** command allows you to replace several username and password configuration commands with a single copy of this command on any dialer interface or asynchronous group interface.

To enable a router calling a collection of routers to configure a common CHAP secret password, perform the following task in interface configuration mode:

Task	Command
Enable a router calling a collection of routers to configure a common CHAP secret password.	ppp chap password secret

Enable PAP

Access control using the Password Authentication Protocol (PAP), available on all serial interfaces that use PPP encapsulation, can reduce the risk of security violations on your router.

You can also reenable remote PAP support to respond to a peer request to authenticate with PAP.

To configure your router to use PAP, and to reenable remote PAP support to respond to a peer request to authenticate with PAP, perform the following tasks in interface configuration mode:

Task	Command
Enable PAP on the interface.	ppp authentication pap [if-needed] or ppp authentication pap [list-name]
Reenable remote PAP support to respond to a peer request to authenticate with PAP.	ppp pap sent-username <i>username</i> password <i>password</i>

The optional keyword **if-needed** can be used only with TACACS or extended TACACS. The optional argument *list-name* can be used only with AAA/TACACS+.

Configuration Examples

This section contains example configurations for all security systems discussed in this chapter, and includes the following:

- RADIUS Configuration Examples
- Kerberos Configuration Examples
- AAA/TACACS+ Authentication Examples

RADIUS Configuration Examples

Radius configuration examples in this section include the following:

- RADIUS Authentication and Authorization Examples
- RADIUS Configuration Example

RADIUS Authentication and Authorization Examples

This section provides two sample configurations using RADIUS.

The following example shows how to configure the router to authenticate and authorize using RADIUS:

```
aaa authentication login use-radius RADIUS local
aaa authentication ppp user-radius if-needed radius
aaa authorization exec radius if-authenticated
aaa authorization network radius
```

The lines in this sample RADIUS authentication and authorization configuration are defined as follows:

- The **aaa authentication login use-radius RADIUS local** command configures the router to use RADIUS for authentication at the login prompt by default. If RADIUS returns an error, the user is authenticated using the local database.
- The **aaa authentication ppp use-radius if-needed radius** command configures the Cisco IOS software to use PPP authentication using CHAP or PAP if the user has not already logged in. If the EXEC facility has authenticated the user, RADIUS authentication is not performed.

- The **aaa authorization exec radius if-authenticated** command sets the RADIUS information that is used for EXEC authorization, autocommands, and access lists.
- The **aaa authorization network radius** command sets RADIUS for network authorization, address assignment, and other access lists.

The following example shows how to configure the router to prompt for and verify a username and password, authorize the user's EXEC level, and specify TACACS+ as the method of authorization for privilege level 2. In this example, if a local username is entered at the username prompt, that username is used for authentication.

EXEC authorization using RADIUS will fail because no data is saved from the RADIUS authentication. The method list also uses the local database to find an autocommand. If there is no autocommand, the user becomes the EXEC user. If the user then attempts to issue commands that are set at privilege level 2, TACACS+ is used to attempt to authorize the command.

```
aaa authentication local-override
aaa authentication login default radius local
aaa authorization exec radius local
aaa authorization command 2 tacacs+ if-authenticated
```

The lines in this sample RADIUS authentication and authorization configuration are defined as follows:

- The **aaa authentication local-override** command specifies that the username prompt appear before authentication starts, and that the authentication always use the local database if the user has a local account.
- The **aaa authentication login default radius local** command specifies that the username and password are verified by RADIUS or, if RADIUS returns an error, by the router's local user database.
- The **aaa authorization exec radius local** command specifies that RADIUS authentication information be used to set the user's EXEC level if the user authenticates with RADIUS. If no RADIUS information is used, this command specifies that the local user database be used for EXEC authorization.
- The **aaa authorization command 2 tacacs+ if-authenticated** command specifies TACACS+ authorization for commands set at privilege level 2.

RADIUS Configuration Example

The following sample is a general configuration using RADIUS with the AAA command set:

```
radius-server host 123.45.1.2
radius-server key myRaDiUSpassWoRd
username root password ALongPassword
aaa authentication ppp dialins radius local
aaa authorization network radius local
aaa accounting network start-stop radius
aaa authentication login admins local
aaa authorization exec local
line 1 16
autoselect ppp
autoselect during-login
login authentication admins
modem ri-is-cd
interface group-async 1
encaps ppp
ppp authentication pap dialins
```

The lines in this sample RADIUS authentication, authorization, and accounting configuration are defined as follows:

- The **aaa authentication ppp dialins radius local** command configures the line to perform RADIUS authentication by default on lines that automatically detect incoming PPP packets.
- The **ppp authentication pap dialins** command configures PAP to be used for dialin authentication.
- The **aaa authorization network radius local** command is used to assign an address and other network parameters to the RADIUS user.
- The **aaa accounting network start-stop radius** command tracks PPP usage.
- The **aaa authentication login admins local** command sets the administrator's access to use the internal username and password database and is intended to be used by network administrators when they debug problems with the network.

Kerberos Configuration Examples

Configuration examples in this section include the following:

- Define a Kerberos Realm Examples
- Copy SRVTAB Files Example
- Specify Kerberos Authentication Examples
- Enable Kerberos Instance Mapping Examples
- Kerberos Configuration Example

Define a Kerberos Realm Examples

To define CISCO.COM as the default Kerberos realm, use the following command:

```
kerberos local-realm CISCO.COM
```

To tell the router that the CISCO.COM KDC is running on host 1.2.3.4 at port number 170, use the following Kerberos command:

```
kerberos server CISCO.COM 1.2.3.4 170
```

To map the DNS domain cisco.com to the Kerberos realm CISCO.COM, use the following command:

```
kerberos realm .cisco.com CISCO.COM
```

Copy SRVTAB Files Example

To copy over the SRVTAB file on a host named host123.cisco.com for a router named router1.cisco.com, the command would look like this:

```
kerberos srvtab remote host123.cisco.com router1.cisco.com-new-srvtab
```

Specify Kerberos Authentication Examples

To specify Kerberos as the authentication method, use the following command:

```
aaa authentication login default krb5
```

Use the following command to specify Kerberos authentication for PPP:

```
aaa authentication ppp default krb5
```

Kerberized Telnet Example

Use the following command to authenticate users who open Telnet sessions to the router using Kerberos credentials:

```
aaa authentication login default krb5-telnet krb5
```

Enable Kerberos Instance Mapping Examples

Use the following command to map the Kerberos instance, *admin*, to enable mode:

```
kerberos instance map admin 15
```

Use the following command to configure the router to check users' Kerberos instances and set appropriate privilege levels:

```
aaa authorization exec krb5-instance
```

Kerberos Configuration Example

This section provides a typical non-Kerberos router configuration and shows output for this configuration from the **write term** command, then builds on this configuration by adding optional Kerberos functionality. Output for each configuration is presented for comparison against the previous configuration.

This example shows how to use the `kdb5_edit` program to perform the following configuration tasks:

- Add user `chet` to the Kerberos database
- Add an privileged Kerberos instance of user `chet` (`chet/admin`) to the Kerberos database
- Add a restricted instance of `chet` (`chet/restricted`) to the Kerberos database
- Add workstation `chet-ss20.cisco.com`
- Add router `chet-2500.cisco.com` to the Kerberos database
- Add workstation `chet-ss20.cisco.com` to the Kerberos database
- Extract SRVTABs for the router and workstations
- List the contents of the KDC database (with the **ldb** command)

Note that, in this sample configuration, host `chet-ss20` is also the KDC.

```
chet-ss20# sbin/kdb5_edit
kdb5_edit: ank chet
Enter password:
Re-enter password for verification:
kdb5_edit: ank chet/admin
Enter password:
Re-enter password for verification:
kdb5_edit: ank chet/restricted
Enter password:
Re-enter password for verification:
kdb5_edit: ark host/chet-ss20.cisco.com
kdb5_edit: ark host/chet-2500.cisco.com
```

```
kdb5_edit: ark host/chet-ss20.cisco.com
kdb5_edit: xst chet-ss20.cisco.com host
'host/chet-ss20.cisco.com@CISCO.COM' added to keytab
'WRFIL:chet-ss20.cisco.com-new-srvtab'
kdb5_edit: xst chet-2500.cisco.com host
'host/chet-2500.cisco.com@CISCO.COM' added to keytab
'WRFIL:chet-2500.cisco.com-new-srvtab'
kdb5_edit: xst chet-ss20.cisco.com host
kdb5_edit: ldb
entry: host/chet-2500.cisco.com@CISCO.COM
entry: chet/restricted@CISCO.COM
entry: chet@CISCO.COM
entry: K/M@CISCO.COM
entry: host/chet-ss20.cisco.com@CISCO.COM
entry: krbtgt/CISCO.COM@CISCO.COM
entry: chet/admin@CISCO.COM
kdb5_edit: q
chet-ss20#
```

The following example shows output from a **write term** command, which displays the configuration of router chet-2500. This is a typical configuration with no Kerberos authentication.

```
chet-2500# write term
Building configuration...

Current configuration:
!
! Last configuration
change at 14:03:55 PDT Mon May 13 1996
!
version 11.2
service udp-small-servers
service tcp-small-servers
!
hostname chet-2500
!
clock timezone PST -8
clock summer-time PDT recurring
aaa new-model
aaa authentication login console none
aaa authentication ppp local local
enable password sMudgKin
!
username chet-2500 password 7 sMudgkin
username chet-3000 password 7 sMudgkin
username chetin password 7 sMudgkin
!
interface Ethernet0
 ip address 172.16.0.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
 no fair-queue
!
interface Serial1
 no ip address
 shutdown
 no fair-queue
!
interface Async2
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic routing
```

```

async mode dedicated
no cdp enable
ppp authentication pap local
no tarp propagate
!
interface Async3
ip unnumbered Ethernet0
encapsulation ppp
shutdown
async dynamic address
async dynamic routing
async mode dedicated
no cdp enable
ppp authentication pap local
no tarp propagate
!
router eigrp 109
network 172.17.0.0
no auto-summary
!
ip default-gateway 172.30.55.64
ip domain-name cisco.com
ip name-server 192.168.0.0
ip classless
!
!
line con 0
exec-timeout 0 0
login authentication console
line 1 16
transport input all
line aux 0
transport input all
line vty 0 4
password sMudgKin
!
ntp clock-period 17179703
ntp peer 172.19.10.0
ntp peer 172.19.0.0
end

```

The following example shows the commands you use to perform the following tasks:

- Enter configuration mode
- Define the Kerberos local realm
- Identify the machine hosting the KDC
- Enable credentials forwarding
- Specify Kerberos as the method of authentication for login
- Exit configuration mode (CTL-Z)
- Write the new configuration to the terminal

```

chet-2500# configure term
Enter configuration commands, one per line. End with CNTL/Z.
chet-2500(config)# kerberos local-realm CISCO.COM
chet-2500(config)# kerberos server CISCO.COM chet-ss20
Translating "chet-ss20"...domain server (192.168.0.0) [OK]

chet-2500(config)# kerberos credentials forward
chet-2500(config)# aaa authentication login default krb5
chet-2500(config)#

```

```
chet-2500#
%SYS-5-CONFIG_I: Configured from console by console
chet-2500# write term
```

Compare the following configuration with the previous one. In particular, look at the lines beginning with the the words “aaa,” “username,” and “kerberos” (approximately lines 15 through 25) in this new configuration.

```
Building configuration...

Current configuration:
!
! Last configuration change at 14:05:54 PDT Mon May 13 1996
!
version 11.2
service udp-small-servers
service tcp-small-servers
!
hostname chet-2500
!
clock timezone PST -8
clock summer-time PDT recurring
aaa new-model
aaa authentication login default krb5
aaa authentication login console none
aaa authentication ppp local local
enable password sMudgKin
!
username chet-2500 password 7 sMudgkin
username chet-3000 password 7 sMudgkin
username chetin password 7 sMudgkin
kerberos local-realm CISCO.COM
kerberos server CISCO.COM 172.71.54.14
kerberos credentials forward
!
interface Ethernet0
 ip address 172.16.0.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
 no fair-queue
!
interface Serial1
 no ip address
 shutdown
 no fair-queue
!
interface Async2
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
interface Async3
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic address
 async dynamic routing
```

```

async mode dedicated
no cdp enable
ppp authentication pap local
no tarp propagate
!
router eigrp 109
network 172.17.0.0
no auto-summary
!
ip default-gateway 172.30.55.64
ip domain-name cisco.com
ip name-server 192.168.0.0
ip classless
!
!
line con 0
exec-timeout 0 0
login authentication console
line 1 16
transport input all
line aux 0
transport input all
line vty 0 4
password sMudgKin
!
ntp clock-period 17179703
ntp peer 172.19.10.0
ntp peer 172.19.0.0
end

```

With the router configured thus far, user chet can log in to the router with a username and password and automatically obtain a TGT, as illustrated in the next example. With possession of a credential, user chet successfully authenticates to host chet-ss20 without entering a username/password.

```

chet-ss20% telnet chet-2500
Trying 172.16.0.0 ...
Connected to chet-2500.cisco.com.
Escape character is '^]'.

```

User Access Verification

```

Username: chet
Password:

```

```

chet-2500> show kerberos creds
Default Principal: chet@CISCO.COM
Valid Starting      Expires              Service Principal
13-May-1996 14:05:39  13-May-1996 22:06:40  krbtgt/CISCO.COM@CISCO.COM

```

```

chet-2500> telnet chet-ss20
Trying chet-ss20.cisco.com (172.71.54.14)... Open
Kerberos:      Successfully forwarded credentials

```

SunOS UNIX (chet-ss20) (pts/7)

```

Last login: Mon May 13 13:47:35 from chet-ss20.cisco.c
Sun Microsystems Inc. SunOS 5.4 Generic July 1994
unknown mode: new
chet-ss20%

```

The following example shows the commands you use to perform the following tasks on the router console port:

- Enter configuration mode
- Remotely copy over the SRVTAB file from the KDC
- Set authentication at login to use the Kerberos 5 Telnet authentication protocol when using Telnet to connect to the router
- Write the configuration to the terminal

Note that the new configuration contains a **kerberos srvtab entry** line. This line is created by the **kerberos srvtab remote** command.

```
chet-2500# configure term
Enter configuration commands, one per line.  End with CNTL/Z.
chet-2500(config)#kerberos srvtab remote earth chet/chet-2500.cisco.com-new-srvtab
Translating "earth"...domain server (192.168.0.0) [OK]

Loading chet/chet-2500.cisco.com-new-srvtab from 172.68.1.123 (via Ethernet0): !
[OK - 66/1000 bytes]

chet-2500(config)# aaa authentication login default krb5-telnet krb5
chet-2500(config)#
chet-2500#
%SYS-5-CONFIG_I: Configured from console by console
chet-2500# write term
Building configuration...

Current configuration:
!
! Last configuration change at 14:08:32 PDT Mon May 13 1996
!
version 11.2
service udp-small-servers
service tcp-small-servers
!
hostname chet-2500
!
clock timezone PST -8
clock summer-time PDT recurring
aaa new-model
aaa authentication login default krb5-telnet krb5
aaa authentication login console none
aaa authentication ppp local local
enable password sMudgKin
!
username chet-2500 password 7 sMudgkin
username chet-3000 password 7 sMudgkin
username chetin password 7 sMudgkin
kerberos local-realm CISCO.COM
kerberos srvtab entry host/chet-2500.cisco.com@CISCO.COM 0 832015393 1 1 8 7 sMudgkin
kerberos server CISCO.COM 172.71.54.14
kerberos credentials forward
!
interface Ethernet0
 ip address 172.16.0.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
 no fair-queue
!
```

```

interface Serial1
  no ip address
  shutdown
  no fair-queue
!
interface Async2
  ip unnumbered Ethernet0
  encapsulation ppp
  shutdown
  async dynamic routing
  async mode dedicated
  no cdp enable
  ppp authentication pap local
  no tarp propagate
!
interface Async3
  ip unnumbered Ethernet0
  encapsulation ppp
  shutdown
  async dynamic address
  async dynamic routing
  async mode dedicated
  no cdp enable
  ppp authentication pap local
  no tarp propagate
!
router eigrp 109
  network 172.17.0.0
  no auto-summary
!
ip default-gateway 172.30.55.64
ip domain-name cisco.com
ip name-server 192.168.0.0
ip classless
!
!
line con 0
  exec-timeout 0 0
  login authentication console
line 1 16
  transport input all
line aux 0
  transport input all
line vty 0 4
  password sMudgKin
!
ntp clock-period 17179703
ntp peer 172.19.10.0
ntp peer 172.19.0.0
end

chet-2500#

```

With this configuration, the user can Telnet in to the router using Kerberos credentials, as illustrated in the next example.

```

chet-ss20% bin/telnet -a -F chet-2500
Trying 172.16.0.0...
Connected to chet-2500.cisco.com.
Escape character is '^]'.
[ Kerberos V5 accepts you as "chet@CISCO.COM" ]

```

User Access Verification

```
chet-2500>[ Kerberos V5 accepted forwarded credentials ]

chet-2500> show kerberos creds
Default Principal: chet@CISCO.COM
Valid Starting      Expires      Service Principal
13-May-1996 15:06:25  14-May-1996 00:08:29  krbtgt/CISCO.COM@CISCO.COM

chet-2500>q
Connection closed by foreign host.
chet-ss20%
```

The following example show the commands you use to perform the following tasks:

- Enter configuration mode
- Map the Kerberos instance, admin, to privilege level 15
- Map the Kerberos instance, restricted, to privilege level 3
- Specify that the instance defined by the **Kerberos instance map** command be used for AAA Authorization
- Writes the configuration to the terminal

```
chet-2500# configure term
Enter configuration commands, one per line.  End with CNTL/Z.
chet-2500(config)# kerberos instance map admin 15
chet-2500(config)# kerberos instance map restricted 3
chet-2500(config)# aaa authorization exec krb5-instance
chet-2500(config)#
chet-2500#
%SYS-5-CONFIG_I: Configured from console by console
chet-2500# write term
Building configuration...

Current configuration:
!
! Last configuration change at 14:59:05 PDT Mon May 13 1996
!
version 11.2
service udp-small-servers
service tcp-small-servers
!
hostname chet-2500
!
aaa new-model
aaa authentication login default krb5-telnet krb5
aaa authentication login console none
aaa authentication ppp default krb5 local
aaa authorization exec krb5-instance
enable password sMudgKin
!
username chet-2500 password 7 sMudgkin
username chet-3000 password 7 sMudgkin
username chetin password 7 sMudgkin
ip domain-name cisco.com
ip name-server 192.168.0.0
kerberos local-realm CISCO.COM
kerberos srvtab entry host/chet-2500.cisco.com@CISCO.COM 0 832015393 1 1 8 7 sMudgkin
kerberos server CISCO.COM 172.71.54.14
kerberos instance map admin 15
kerberos instance map restricted 3
kerberos credentials forward
clock timezone PST -8
clock summer-time PDT recurring
```

```

!
interface Ethernet0
 ip address 172.16.0.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
 no fair-queue
!
interface Serial1
 no ip address
 shutdown
 no fair-queue
!
interface Async2
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
interface Async3
 ip unnumbered Ethernet0
 encapsulation ppp
 shutdown
 async dynamic address
 async dynamic routing
 async mode dedicated
 no cdp enable
 ppp authentication pap local
 no tarp propagate
!
router eigrp 109
 network 172.17.0.0
 no auto-summary
!
ip default-gateway 172.30.55.64
ip classless
!
!
line con 0
 exec-timeout 0 0
 login authentication console
line 1 16
 transport input all
line aux 0
 transport input all
line vty 0 4
 password sMudgKin
!
ntp clock-period 17179703
ntp peer 172.19.10.0
ntp peer 172.19.0.0
end

chet-2500#

```

The following example shows output from from the three types of sessions now possible for user chet with Kerberos instances turned on:

```

chet-ss20% telnet chet-2500
Trying 172.16.0.0 ...

```

Configuration Examples

```
Connected to chet-2500.cisco.com.  
Escape character is '^]'.  
  
User Access Verification
```

```
Username: chet  
Password:
```

```
chet-2500> show kerberos creds  
Default Principal: chet@CISCO.COM  
Valid Starting Expires Service Principal  
13-May-1996 14:58:28 13-May-1996 22:59:29 krbtgt/CISCO.COM@CISCO.COM
```

```
chet-2500> show privilege  
Current privilege level is 1  
chet-2500> q  
Connection closed by foreign host.  
chet-ss20% telnet chet-2500  
Trying 172.16.0.0 ...  
Connected to chet-2500.cisco.com.  
Escape character is '^]'.  
  
User Access Verification
```

```
Username: chet/admin  
Password:
```

```
chet-2500# show kerberos creds  
Default Principal: chet/admin@CISCO.COM  
Valid Starting Expires Service Principal  
13-May-1996 14:59:44 13-May-1996 23:00:45 krbtgt/CISCO.COM@CISCO.COM
```

```
chet-2500# show privilege  
Current privilege level is 15  
chet-2500# q  
Connection closed by foreign host.  
chet-ss20% telnet chet-2500  
Trying 172.16.0.0 ...  
Connected to chet-2500.cisco.com.  
Escape character is '^]'.  
  
User Access Verification
```

```
Username: chet/restricted  
Password:
```

```
chet-2500# show kerberos creds  
Default Principal: chet/restricted@CISCO.COM  
Valid Starting Expires Service Principal  
13-May-1996 15:00:32 13-May-1996 23:01:33 krbtgt/CISCO.COM@CISCO.COM
```

```
chet-2500# show privilege  
Current privilege level is 3  
chet-2500# q  
Connection closed by foreign host.  
chet-ss20%
```

AAA/TACACS+ Authentication Examples

TACACS+ configuration examples in this section include the following:

- TACACS+ Authentication with ARA Examples
- Restrict Network Access Examples
- TACACS+ Authentication Example
- TACACS Authentication Examples

TACACS+ Authentication with ARA Examples

The following example creates a default AAA authentication algorithm used with the ARA protocol:

```
aaa authentication arap default if-needed none
```

The following example creates the same authentication algorithm for the ARA protocol but calls the list *MIS-access*:

```
aaa authentication arap MIS-access if-needed none
```

Restrict Network Access Examples

The following example allows network authorization using TACACS+:

```
aaa authorization network tacacs+
```

The following example provides the same authorization, but also creates address pools called *mci* and *att*:

```
aaa authorization network tacacs+
ip address-pool local
ip local-pool mci 172.16.0.1 172.16.0.255
ip local-pool att 172.17.0.1 172.17.0.255
```

These address pools can then be selected by the TACACS daemon. A sample configuration of the daemon follows:

```
user = mci_customer1 {
    login = cleartext "some password"
    service = ppp protocol = ip {
        addr-pool=mci
    }
}

user = att_customer1 {
    login = cleartext "some other password"
    service = ppp protocol = ip {
        addr-pool=att
    }
}
```

TACACS+ Authorization Example

The following example uses a TACACS+ server to authorize the use of network services, including PPP and ARA. If the TACACS+ server is not available or has no information about a user, no authorization is performed, and the user can use all network services:

```
aaa authorization network tacacs+ none
```

TACACS Authentication Examples

The following example shows TACACS enabled for PPP authentication:

```
int async 1
  ppp authentication chap
  ppp use-tacacs
```

The following example shows TACACS enabled for ARAP authentication:

```
line 3
  arap use-tacacs
```