Configuring Switch-Based Authentication

This chapter describes how to configure switch-based authentication on the Cisco Industrial Ethernet 2000U Series Switches (IE 2000U) and includes the following sections:

Information About Switch-Based Authentication

This section includes the following topics:

Preventing Unauthorized Access to Your Switch

You can prevent unauthorized users from reconfiguring your switch and viewing configuration information. Typically, you want network administrators to have access to your switch while you restrict access to users who dial in from outside the network through an asynchronous port, connect from outside the network through a serial port, or connect through a terminal or workstation from within the local network.

To prevent unauthorized access into your switch, you should configure one or more of these security features:

- At a minimum, you should configure passwords and privileges at each switch port. The software stores these passwords locally on the switch. When users attempt to access the switch through a port or line, they must enter the password specified for the port or line before they can access the switch. For more information, see the "Protecting Access to Privileged EXEC Commands" section on page 1-19.

- For an additional layer of security, you can also configure username and password pairs, which are locally stored on the switch. These pairs are assigned to lines or ports and authenticate each user before that user can access the switch. If you have defined privilege levels, you can also assign a specific privilege level (with associated rights and privileges) to each username and password pair. For more information, see the "Configuring Username and Password Pairs" section on page 1-24.

- If you want to use username and password pairs but you want to store them centrally on a server instead of locally, you can store them in a database on a security server. Multiple networking devices can then use the same database to obtain user authentication (and, if necessary, authorization) information. For more information, see the "TACACS+" section on page 1-2.

TACACS+

TACACS+ is a security application that provides centralized validation of users attempting to gain access to your switch. TACACS+ services are maintained in a database on a TACACS+ daemon typically running on a UNIX or Windows NT workstation. You should have access to and should configure a TACACS+ server before configuring TACACS+ features on your switch.

TACACS+ provides for separate and modular authentication, authorization, and accounting facilities. TACACS+ allows for a single access control server (the TACACS+ daemon) 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, depending on the capabilities of the daemon.

The goal of TACACS+ is to provide a method for managing multiple network access points from a single management service. Your switch can be a network access server along with other Cisco routers and access servers. A network access server provides connections to a single user, to a network or subnetwork, and to interconnected networks as shown in Figure 1-1.

Figure 1-1   Typical TACACS+ Network Configuration
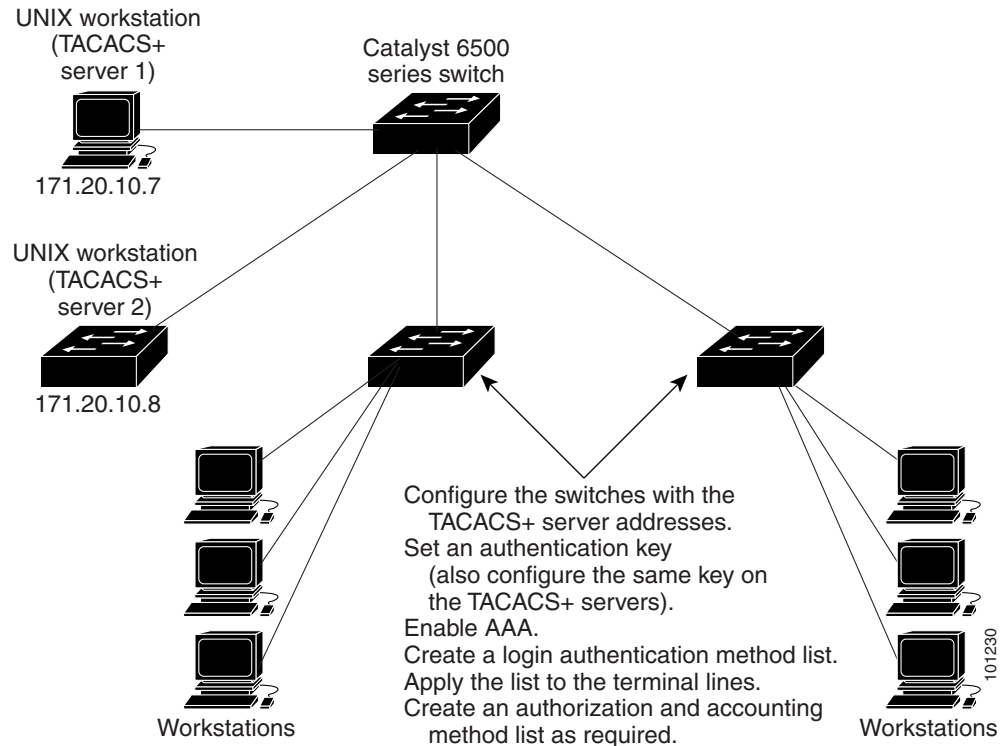
TACACS+, administered through the AAA security services, can provide these services:

- Authentication—Provides complete control of authentication through login and password dialog, challenge and response, and messaging support.

  The authentication facility can conduct a dialog with the user (for example, after a username and password are provided, to challenge a user with several questions, such as home address, mother's maiden name, service type, and social security number). The TACACS+ authentication service can also send messages to user screens. For example, a message could notify users that their passwords must be changed because of the company's password aging policy.

- Authorization—Provides fine-grained control over user capabilities for the duration of the user's session, including but not limited to setting autocommands, access control, session duration, or protocol support. You can also enforce restrictions on what commands a user can execute with the TACACS+ authorization feature.

- Accounting—Collects and sends information used for billing, auditing, and reporting to the TACACS+ daemon. Network managers can use the accounting facility to track user activity for a security audit or to provide information for user billing. Accounting records include 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 switch and the TACACS+ daemon, and it ensures confidentiality because all protocol exchanges between the switch and the TACACS+ daemon are encrypted.

You need a system running the TACACS+ daemon software to use TACACS+ on your switch.

TACACS+ Operation

When a user attempts a simple ASCII login by authenticating to a switch using TACACS+, this process occurs:

1. When the connection is established, the switch contacts the TACACS+ daemon to obtain a username prompt to show to the user. The user enters a username, and the switch then contacts the TACACS+ daemon to obtain a password prompt. The switch displays the password prompt to the user, the user enters a password, and the password is then sent to the TACACS+ daemon.

   TACACS+ allows a dialog between the daemon and the user until the daemon receives enough information to authenticate the user. The daemon prompts for a username and password combination, but can include other items, such as the user's mother's maiden name.

2. The switch eventually receives one of these responses from the TACACS+ daemon:

   • ACCEPT—The user is authenticated and service can begin. If the switch is configured to require authorization, authorization begins at this time.

   • REJECT—The user is not authenticated. The user can be denied access or is prompted to retry the login sequence, depending on the TACACS+ daemon.

   • ERROR—An error occurred at some time during authentication with the daemon or in the network connection between the daemon and the switch. If an ERROR response is received, the switch typically tries to use an alternative method for authenticating the user.

   • CONTINUE—The user is prompted for additional authentication information.

   After authentication, the user undergoes an additional authorization phase if authorization has been enabled on the switch. Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.

3. If TACACS+ authorization is required, the TACACS+ daemon is again contacted, and it returns an ACCEPT or REJECT authorization response. If an ACCEPT response is returned, the response contains data in the form of attributes that direct the EXEC or NETWORK session for that user and the services that the user can access:

   • Telnet, Secure Shell (SSH), rlogin, or privileged EXEC services

   • Connection parameters, including the host or client IP address, access list, and user timeouts

RADIUS

RADIUS is a distributed client/server system that secures networks against unauthorized access. RADIUS clients run on supported Cisco routers and switches. Clients send authentication requests to a central RADIUS server, which contains all user authentication and network service access information. The RADIUS host is normally a multiuser system running RADIUS server software from Cisco (Cisco Secure Access Control Server Version 3.0), Livingston, Merit, Microsoft, or another software provider. For more information, see the RADIUS server documentation.

Use RADIUS in these 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 vendors' access servers, dial-in users are authenticated through a RADIUS server that has been customized to work with the Kerberos security system.

• Turnkey network security environments in which applications support the RADIUS protocol, such as 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 to grant access to network resources.

- Networks already using RADIUS. You can add a Cisco switch containing a RADIUS client to the network. This might be the first step when you make a transition to a TACACS+ server. See Figure 1-2 on page 1-5.

- Network in which the 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 the network, through a protocol such as IEEE 802.1x.

- 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, showing the amount of resources (such as time, packets, bytes, and so forth) used during the session. An Internet service provider 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 these network security situations:

- Multiprotocol access environments. RADIUS does not support AppleTalk Remote Access (ARA), NetBIOS Frame Control Protocol (NBFCP), NetWare Asynchronous Services Interface (NASI), or X.25 PAD connections.

- Switch-to-switch or router-to-router situations. RADIUS does not provide two-way authentication. RADIUS can be used to authenticate from one device to a non-Cisco device if the non-Cisco device requires authentication.

- Networks using a variety of services. RADIUS generally binds a user to one service model.

Figure 1-2   Transitioning from RADIUS to TACACS+ Services
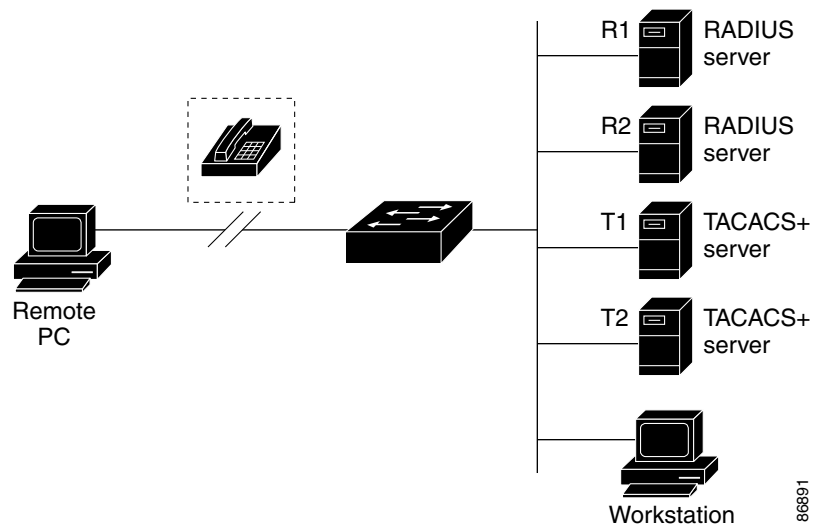
RADIUS Operation

When a user attempts to log in and authenticate to a switch that is access-controlled by a RADIUS server:

1. The user is prompted to enter 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 these responses from the RADIUS server:

   - ACCEPT—The user is authenticated.

- REJECT—The user is either not authenticated and is prompted to reenter the username and password, or access is denied.
- CHALLENGE—A challenge requires additional data from the user.
- CHALLENGE PASSWORD—A response requests the user to select a new password.

The ACCEPT or REJECT response is bundled with additional data that is used for privileged EXEC or network authorization. Users must first successfully complete RADIUS authentication before proceeding to RADIUS authorization, if it is enabled. The additional data included with the ACCEPT or REJECT packets includes these items:

- Telnet, SSH, rlogin, or privileged EXEC services
- Connection parameters, including the host or client IP address, access list, and user timeouts

RADIUS Change of Authorization

This section provides an overview of the RADIUS interface including available primitives and how they are used during a Change of Authorization (CoA).

- Overview, page 1-6
- Change-of-Authorization Requests, page 1-7
- CoA Request Response Code, page 1-8
- CoA Request Commands, page 1-9

Overview

A standard RADIUS interface is typically used in a pulled model, where the request originates from a network attached device and the response come from the queried servers. The switch supports the RADIUS Change of Authorization (CoA) extensions defined in RFC 5176 that are typically used in a pushed model and allow for the dynamic reconfiguring of sessions from external authentication, authorization, and accounting (AAA) or policy servers.

The switch supports these per-session CoA requests:

- Session reauthentication
- Session termination
- Session termination with port shutdown
- Session termination with port bounce

The RADIUS interface is enabled by default on the switch. However, some basic configuration is required for the following attributes:

- Security and Password—see the "Protecting Access to Privileged EXEC Commands" section on page 1-19.
- Accounting—see the "Starting RADIUS Accounting" section on page 1-42.

Change-of-Authorization Requests

Change of Authorization (CoA) requests, as described in RFC 5176, are used in a push model to allow for session identification, host reauthentication, and session termination. The model is comprised of one request (CoA-Request) and two possible response codes:

- CoA acknowledgement (ACK) [CoA-ACK]
- CoA non-acknowledgement (NAK) [CoA-NAK]

The request is initiated from a CoA client (typically a RADIUS or policy server) and directed to the switch that acts as a listener.

RFC 5176 Compliance

The Disconnect Request message, which is also referred to as Packet of Disconnect (POD), is supported by the switch for session termination. Table 1-1 shows the IETF attributes that are supported for the RADIUS CoA feature. Table 1-2 shows the possible values for the Error-Cause attribute.

Table 1-1   Supported IETF Attributes

Attribute Number	Attribute Name
24	State
31	Calling-Station-ID
44	Acct-Session-ID
80	Message-Authenticator
101	Error-Cause

Table 1-2   Error-Cause Values

Value	Explanation
201	Residual Session Context Removed
202	Invalid EAP Packet (Ignored)
401	Unsupported Attribute
402	Missing Attribute
403	NAS Identification Mismatch
404	Invalid Request
405	Unsupported Service
406	Unsupported Extension
407	Invalid Attribute Value
501	Administratively Prohibited
502	Request Not Routable (Proxy)
503	Session Context Not Found
504	Session Context Not Removable
505	Other Proxy Processing Error
506	Resources Unavailable

Table 1-2   Error-Cause Values (continued)

Value	Explanation
507	Request Initiated
508	Multiple Session Selection Unsupported

Preconditions

To use the CoA interface, a session must already exist on the switch. CoA can be used to identify a session and enforce a disconnect request. The update affects only the specified session.

CoA Request Response Code

The CoA Request response code can be used to convey a command to the switch. The supported commands are listed in .

Session Identification

For disconnect and CoA requests targeted at a particular session, the switch locates the session based on one or more of the following attributes:

- Calling-Station-Id (IETF attribute 31 which contains the host MAC address)
- Audit-Session-Id (Cisco VSA)
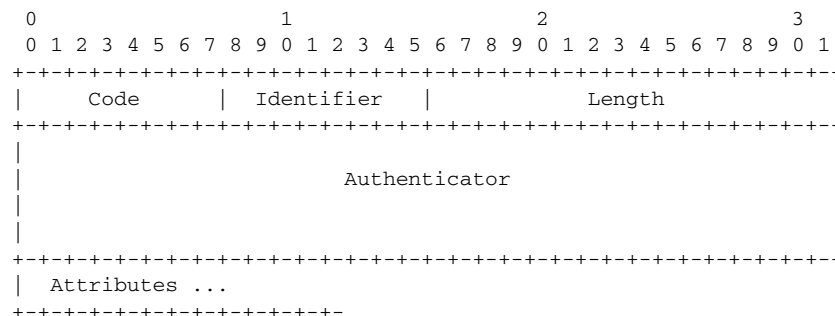- Acct-Session-Id (IETF attribute 44)

Unless all session identification attributes included in the CoA message match the session, the switch returns a Disconnect-NAK or CoA-NAK with the "Invalid Attribute Value" error-code attribute.

For disconnect and CoA requests targeted to a particular session, any one of the following session identifiers can be used:

- Calling-Station-ID (IETF attribute 31, which should contain the MAC address)
- Audit-Session-ID (Cisco vendor-specific attribute)
- Accounting-Session-ID (IETF attribute 44)

If more than one session identification attribute is included in the message, all the attributes must match the session or the switch returns a Disconnect- negative acknowledgement (NAK) or CoA-NAK with the error code "Invalid Attribute Value."

The packet format for a CoA Request code as defined in RFC 5176 consists of the fields Code, Identifier, Length, Authenticator, and Attributes in Type:Length:Value (TLV) format.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         Authenticator                         |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Attributes ...
+-+-+-+-+-+-+-+-+-+-+-+-
```

The attributes field is used to carry Cisco VSAs.

CoA ACK Response Code

If the authorization state is changed successfully, a positive acknowledgement (ACK) is sent. The attributes returned within CoA ACK vary based on the CoA Request and are discussed in the individual CoA command descriptions.

CoA NAK Response Code

A negative acknowledgement (NAK) means that the authorization state did not change. The message can include attributes that show the reason for the failure. Use **show** commands to verify a successful CoA.

CoA Request Commands

- Session Reauthentication
- Session Termination
- CoA Disconnect-Request
- CoA Request: Disable Host Port
- CoA Request: Bounce-Port

The switch supports the commands shown in Table 1-3.

Table 1-3   CoA Commands Supported on the Switch

Command[1]	Cisco VSA
Reauthenticate host	Cisco:Avpair="subscriber:command=reauthenticate"
Terminate session	This is a standard disconnect request that does not require a VSA.
Bounce host port	Cisco:Avpair="subscriber:command=bounce-host-port"
Disable host port	Cisco:Avpair="subscriber:command=disable-host-port"

1. All CoA commands must include the session identifier between the switch and the CoA client.

Session Reauthentication

The AAA server generates a session reauthentication request when a host with an unknown identity or posture joins the network and is associated with a restricted access authorization profile (such as a guest VLAN). A reauthentication request allows the host to be placed in the appropriate authorization group when its credentials are known.

To initiate session authentication, the AAA server sends a standard CoA-Request message that contains a Cisco vendor-specific attribute (VSA) in this form:
Cisco:Avpair="subscriber:command=reauthenticate" and one or more session identification attributes.

The current session state determines the switch response to the message. If the session is currently authenticated by 802.1x, the switch responds by sending an Extensible Authentication Protocol over LAN (EAPoL)-RequestId message to the server.

If the session is currently authenticated by MAC authentication bypass (MAB), the switch sends an access-request to the server, passing the same identity attributes used for the initial successful authentication.

If session authentication is in progress when the switch receives the command, the switch terminates the process, and restarts the authentication sequence, starting with the method configured to be attempted first.

If the session is not yet authorized or is authorized via guest VLAN, critical VLAN, or similar policies, the reauthentication message restarts the access control methods, beginning with the method configured to be attempted first. The current authorization of the session is maintained until the reauthentication leads to a different authorization result.

Session Termination

Three CoA types of requests can trigger session termination. A CoA Disconnect-Request terminates the session, without disabling the host port. This command causes re-initialization of the authenticator state machine for the specified host, but does not restrict its access to the network.

To restrict access to the network, use a CoA Request with the Cisco:Avpair="subscriber:command=disable-host-port" VSA. This command is useful when a host is known to be causing problems on the network and you need to immediately block network access. When you want to restore network access on the port, reenable it using a non-RADIUS mechanism.

When a device with no supplicant, such as a printer, needs a new IP address (for example, after a VLAN change), end the session on the host port with port-bounce (temporarily disable and then reenable the port).

CoA Disconnect-Request

Because this Disconnect-Request command is session-oriented, it must be accompanied by one or more of the session identification attributes described in the "Session Identification" section on page 1-8. If the session cannot be located, the switch returns a Disconnect-NAK message with the "Session Context Not Found" error-code attribute. If the session *is* located, the switch terminates the session. After the session has been completely removed, the switch returns a Disconnect-ACK.

If the switch fails-over to a standby switch before returning a Disconnect-ACK to the client, the process is repeated on the new active switch when the request is re-sent from the client. If the session is still not found, a Disconnect-ACK is sent with the "Session Context Not Found" error-code attribute.

CoA Request: Disable Host Port

This command is carried in a standard CoA-Request message that has this new VSA:

Cisco:Avpair="subscriber:command=disable-host-port"

Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes described in the "Session Identification" section on page 1-8. If the session cannot be located, the switch returns a CoA-NAK message with the "Session Context Not Found" error-code attribute. If the session is located, the switch disables the hosting port and returns a CoA-ACK message.

If the switch fails before returning a CoA-ACK to the client, the process is repeated on the new active switch when the request is re-sent from the client. If the switch fails after returning a CoA-ACK message to the client but before the operation has completed, the operation is restarted on the new active switch.

Note A Disconnect-Request failure following command resending could be the result of either a successful session termination before change-over (if the Disconnect-ACK was not sent) or a session termination by other means (for example, a link failure) that occurred after the original command was issued and before the standby switch became active.

CoA Request: Bounce-Port

This command is carried in a standard CoA-Request message that contains the following new VSA:

Cisco:Avpair="subscriber:command=bounce-host-port"

Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes described in the "Session Identification" section on page 1-8. If the session cannot be located, the switch returns a CoA-NAK message with the "Session Context Not Found" error-code attribute. If the session is located, the switch disables the hosting port for a period of 10 seconds, reenables it (port-bounce), and returns a CoA-ACK.

If the switch fails before returning a CoA-ACK to the client, the process is repeated on the new active switch when the request is re-sent from the client. If the switch fails after returning a CoA-ACK message to the client but before the operation has completed, the operation is re-started on the new active switch.

Kerberos

Kerberos is a secret-key network authentication protocol, which was developed at the Massachusetts Institute of Technology (MIT). It uses the Data Encryption Standard (DES) cryptographic algorithm for encryption and authentication and authenticates requests for network resources. Kerberos uses the concept of a trusted third party to perform secure verification of users and services. This trusted third party is called the *key distribution center* (KDC).

Kerberos verifies that users are who they claim to be and the network services that they use are what the services claim to be. To do this, a KDC or trusted Kerberos server issues tickets to users. These tickets, which have a limited lifespan, are stored in user credential caches. The Kerberos server uses the tickets instead of usernames and passwords to authenticate users and network services.

Note A Kerberos server can be an IE 2000U switch that is configured as a network security server and that can authenticate users by using the Kerberos protocol.

The Kerberos credential scheme uses a process called *single logon*. This process authenticates a user once and then allows secure authentication (without encrypting another password) wherever that user credential is accepted.

This software release supports Kerberos 5, which allows organizations that are already using Kerberos 5 to use the same Kerberos authentication database on the KDC that they are already using on their other network hosts (such as UNIX servers and PCs).

In this software release, Kerberos supports these network services:

- Telnet
- rlogin
- rsh (Remote Shell Protocol)

Table 1-4 lists the common Kerberos-related terms and definitions.

Table 1-4   Kerberos Terms

Term	Definition
Authentication	A process by which a user or service identifies itself to another service. For example, a client can authenticate to a switch or a switch can authenticate to another switch.
Authorization	A means by which the switch identifies what privileges the user has in a network or on the switch and what actions the user can perform.
Credential	A general term that refers to authentication tickets, such as TGTs[1] 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 a ticket, it can be used in place of reentering a username and password. Credentials have a default lifespan of eight hours.
Instance	An authorization-level label for Kerberos principals. Most Kerberos principals are of the form *user@REALM* (for example, smith@EXAMPLE.COM). A Kerberos principal with a Kerberos instance has the form *user/instance@REALM* (for example, smith/admin@EXAMPLE.COM). The Kerberos instance can be used to specify the authorization level for the user if authentication is successful. The server of each network service might implement and enforce the authorization mappings of Kerberos instances but is not required to do so. **Note** The Kerberos principal and instance names *must* be in all lowercase characters. **Note** The Kerberos realm name *must* be in all uppercase characters.
KDC[2]	Key distribution center that consists of a Kerberos server and database program that is running on a network host.
Kerberized	A term that describes 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. **Note** The Kerberos realm name *must* be in all uppercase characters.
Kerberos server	A daemon that is 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.
KEYTAB[3]	A password that a network service shares with the KDC. In Kerberos 5 and later Kerberos versions, the network service authenticates an encrypted service credential by using the KEYTAB to decrypt it. In Kerberos versions earlier than Kerberos 5, KEYTAB is referred to as SRVTAB[4].
Principal	Also known as a Kerberos identity, this is who you are or what a service is according to the Kerberos server. **Note** The Kerberos principal name *must* be in all lowercase characters.

Table 1-4   Kerberos Terms (continued)

Term	Definition
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. The password is also shared with the user TGT.
SRVTAB	A password that a network service shares with the KDC. In Kerberos 5 or later Kerberos versions, SRVTAB is referred to as KEYTAB.
TGT	Ticket granting ticket that is a credential that the KDC issues to authenticated users. When users receive a TGT, they can authenticate to network services within the Kerberos realm represented by the KDC.

1. TGT = ticket granting ticket

2. KDC = key distribution center

3. KEYTAB = key table

4. SRVTAB = server table

Kerberos Operation

A Kerberos server can be a Cisco IE 2000U switch that is configured as a network security server and that can authenticate remote users by using the Kerberos protocol. Although you can customize Kerberos in a number of ways, remote users attempting to access network services must pass through three layers of security before they can access network services.

To authenticate to network services by using a IE 2000U switch as a Kerberos server, remote users must follow these steps:

1. Authenticating to a Boundary Switch, page 1-13

2. Obtaining a TGT from a KDC, page 1-14

3. Authenticating to Network Services, page 1-14

Authenticating to a Boundary Switch

This section describes the first layer of security through which a remote user must pass. The user must first authenticate to the boundary switch. This process then occurs:

1. The user opens an un-Kerberized Telnet connection to the boundary switch.

2. The switch prompts the user for a username and password.

3. The switch requests a TGT from the KDC for this user.

4. The KDC sends an encrypted TGT that includes the user identity to the switch.

5. The switch attempts to decrypt the TGT by using the password that the user entered.

    • If the decryption is successful, the user is authenticated to the switch.

    • If the decryption is not successful, the user repeats Step 2 either by reentering the username and password (noting if Caps Lock or Num Lock is on or off) or by entering a different username and password.

A remote user who initiates a un-Kerberized Telnet session and authenticates to a boundary switch is inside the firewall, but the user must still authenticate directly to the KDC before getting access to the network services. The user must authenticate to the KDC because the TGT that the KDC issues is stored on the switch and cannot be used for additional authentication until the user logs on to the switch.

Obtaining a TGT from a KDC

This section describes the second layer of security through which a remote user must pass. The user must now authenticate to a KDC and obtain a TGT from the KDC to access network services.

For instructions about how to authenticate to a KDC, see the "Obtaining a TGT from a KDC" section in the "Configuring Kerberos" chapter of the User Security Configuration Guide, Cisco IOS Release 15MT.

Authenticating to Network Services

This section describes the third layer of security through which a remote user must pass. The user with a TGT must now authenticate to the network services in a Kerberos realm.

For instructions about how to authenticate to a network service, see the "Authenticating to Network Services" section in the "Configuring Kerberos" chapter of the User Security Configuration Guide, Cisco IOS Release 15MT.

SSH

SSH is a protocol that provides a secure, remote connection to a device. SSH provides more security for remote connections than Telnet does by providing strong encryption when a device is authenticated. This software release supports SSH Version 1 (SSHv1) and SSH Version 2 (SSHv2).

SSH Servers, Integrated Clients, and Supported Versions

The SSH feature has an SSH server and an SSH integrated client, which are applications that run on the switch. You can use an SSH client to connect to a switch running the SSH server. The SSH server works with the SSH client supported in this release and with non-Cisco SSH clients. The SSH client also works with the SSH server supported in this release and with non-Cisco SSH servers.

The switch supports an SSHv1 or an SSHv2 server.

The switch supports an SSHv1 client.

SSH supports the Data Encryption Standard (DES) encryption algorithm, the Triple DES (3DES) encryption algorithm, and password-based user authentication.

SSH also supports these user authentication methods:

- TACACS+ (for more information, see the "Configuring TACACS+" section on page 1-28)
- RADIUS (for more information, see the "Configuring RADIUS" section on page 1-34)
- Local authentication and authorization (for more information, see the "Configuring the Switch for Local Authentication and Authorization" section on page 1-49)

Note   This software release does not support IP Security (IPsec).

Secure HTTP Servers and Clients

On a secure HTTP connection, data to and from an HTTP server is encrypted before being sent over the Internet. HTTP with SSL encryption provides a secure connection to allow such functions as configuring a switch from a web browser. Cisco uses an implementation of SSL Version 3.0 with application-layer encryption to run the secure HTTP server and secure HTTP client. HTTP over SSL is abbreviated as HTTPS; the URL of a secure connection begins with https:// instead of http://.

The primary role of the HTTP secure server (the switch) is to listen for HTTPS requests on a designated port (the default HTTPS port is 443) and pass the request to the HTTP 1.1 web server. The HTTP 1.1 server processes requests and passes responses (pages) back to the HTTP secure server, which, in turn, responds to the original request.

The primary role of the HTTP secure client (the web browser) is to respond to Cisco IOS application requests for HTTPS User Agent services, perform HTTPS User Agent services for the application, and pass the response back to the application.

Certificate Authority Trustpoints

Certificate authorities (CAs) manage certificate requests and issue certificates to participating network devices. These services provide centralized security key and certificate management for the participating devices. Specific CA servers are referred to as *trustpoints*.

When a connection attempt is made, the HTTPS server provides a secure connection by issuing a certified X.509v3 certificate, obtained from a specified CA trustpoint, to the client. The client (usually a web browser), in turn, has a public key that allows it to authenticate the certificate.

For secure HTTP connections, we highly recommend that you configure a CA trustpoint. If a CA trustpoint is not configured for the device running the HTTPS server, the server certifies itself and generates the needed RSA key pair. Because a self-certified (self-signed) certificate does not provide adequate security, the connecting client generates a notification that the certificate is self-certified, and the user has the opportunity to accept or reject the connection. This option is useful for internal network topologies (such as testing).

If you do not configure a CA trustpoint, when you enable a secure HTTP connection, either a temporary or a persistent self-signed certificate for the secure HTTP server (or client) is automatically generated.

- If the switch is not configured with a hostname and a domain name, a temporary self-signed certificate is generated. If the switch reboots, any temporary self-signed certificate is lost, and a new temporary new self-signed certificate is assigned.

- If the switch has been configured with a host and domain name, a persistent self-signed certificate is generated. This certificate remains active if you reboot the switch or if you disable the secure HTTP server so that it will be there the next time you reenable a secure HTTP connection.

> **Note** The certificate authorities and trustpoints must be configured on each device individually. Copying them from other devices makes them invalid on the switch.

If a self-signed certificate has been generated, this information is included in the output of the **show running-config** privileged EXEC command. This is a partial sample output from that command displaying a self-signed certificate.

```
Switch# show running-config
Building configuration...

<output truncated>
```

```
crypto pki trustpoint TP-self-signed-3080755072
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-3080755072
 revocation-check none
 rsakeypair TP-self-signed-3080755072
!
!
crypto ca certificate chain TP-self-signed-3080755072
 certificate self-signed 01
  3082029F 30820208 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  59312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 33303830 37353530 37323126 30240609 2A864886 F70D0109

<output truncated>
```

You can remove this self-signed certificate by disabling the secure HTTP server and entering the **no crypto pki trustpoint TP-self-signed-30890755072** global configuration command. If you later reenable a secure HTTP server, a new self-signed certificate is generated.

Note   The values that follow *TP self-signed* depend on the serial number of the device.

You can use an optional command (**ip http secure-client-auth**) to allow the HTTPS server to request an X.509v3 certificate from the client. Authenticating the client provides more security than server authentication by itself.

For additional information on Certificate Authorities, see the "Configuring Certification Authority Interoperability" chapter in the *Public Key Infrastructure Configuration Guide, Cisco IOS Release 15MT*.

CipherSuites

A CipherSuite specifies the encryption algorithm and the digest algorithm to use on a SSL connection. When connecting to the HTTPS server, the client web browser offers a list of supported CipherSuites, and the client and server negotiate the best encryption algorithm.

Note   To ensure the best results with encryption, please use the recommended web browsers listed in the release notes for this product.

The more secure and more complex CipherSuites require slightly more processing time. This list defines the CipherSuites supported by the switch and ranks them from fastest to slowest in terms of router processing load (speed):

1. SSL_RSA_WITH_DES_CBC_SHA—RSA key exchange (RSA Public Key Cryptography) with DES-CBC for message encryption and SHA for message digest

2. SSL_RSA_WITH_RC4_128_MD5—RSA key exchange with RC4 128-bit encryption and MD5 for message digest

3. SSL_RSA_WITH_RC4_128_SHA—RSA key exchange with RC4 128-bit encryption and SHA for message digest

4. SSL_RSA_WITH_3DES_EDE_CBC_SHA—RSA key exchange with 3DES and DES-EDE3-CBC for message encryption and SHA for message digest

RSA (with the specified encryption and digest algorithm combinations) is used for both key generation and authentication on SSL connections. This usage is independent of whether or not a CA trustpoint is configured.

Secure Copy

The Secure Copy Protocol (SCP) feature provides a secure and authenticated method for copying switch configurations or switch image files. SCP relies on Secure Shell (SSH), an application and a protocol that provides a secure replacement for the Berkeley r-tools.

For SSH to work, the switch needs an RSA public/private key pair. This is the same with SCP, which relies on SSH for its secure transport.

Because SSH also relies on AAA authentication, and SCP relies further on AAA authorization, correct configuration is necessary.

- Before enabling SCP, you must correctly configure SSH, authentication, and authorization on the switch.
- Because SCP relies on SSH for its secure transport, the router must have an Rivest, Shamir, and Adelman (RSA) key pair.

Note When using SCP, you cannot enter the password into the copy command. You must enter the password when prompted.

The behavior of SCP is similar to that of remote copy (rcp), which comes from the Berkeley r-tools suite, except that SCP relies on SSH for security. SCP also requires that authentication, authorization, and accounting (AAA) authorization be configured so the router can determine whether the user has the correct privilege level.

A user who has appropriate authorization can use SCP to copy any file in the Cisco IOS File System (IFS) to and from a switch by using the **copy** command. An authorized administrator can also do this from a workstation.

For more information on how to configure and verify SCP, see the "Secure Copy Protocol" chapter of the *Secure Shell Configuration Guide, Cisco IOS Release 15M&T.*

Prerequisites

- You should have access to and should configure a TACACS+ server before configuring TACACS+ features on your switch.
- You should have access to and should configure a RADIUS server before configuring RADIUS features on your switch.
- To use Kerberos, SSH, and SSL, you must install the cryptographic (encrypted) software image on your switch. You must obtain authorization to use this feature and to download the cryptographic software files from Cisco.com. For more information, see the release notes for this release.

Guidelines and Limitations

Setting Password Encryption

If you enable password encryption by using the **enable secret** command, it applies to all passwords including username passwords, authentication key passwords, the privileged command password, and console and virtual terminal line passwords. (See "Protecting Enable and Enable Secret Passwords with Encryption" section on page 1-21.)

- We recommend that you use the **enable secret** command because it uses an improved encryption algorithm.

- If you configure the **enable secret** command, it takes precedence over the **enable password** (unencrypted) command; the two commands cannot be in effect simultaneously.

TACACS+

To prevent a lapse in security, you cannot configure TACACS+ through a network management application. When enabled, TACACS+ can authenticate users accessing the switch through the CLI.

> **Note**   Although TACACS+ configuration is performed through the CLI, the TACACS+ server authenticates HTTP connections that have been configured with a privilege level of 15.

RADIUS

To prevent a lapse in security, you cannot configure RADIUS through a network management application. When enabled, RADIUS can authenticate users accessing the switch through the CLI.

SSH

These limitations apply to SSH:

- The switch supports RSA authentication.

- SSH supports only the execution-shell application.

- The SSH server and the SSH client are supported only on DES (56-bit) and 3DES (168-bit) data encryption software.

- The switch supports the Advanced Encryption Standard (AES) encryption algorithm with a 128-bit key, 192-bit key, or 256-bit key. However, symmetric cipher AES to encrypt the keys is not supported.

SSL

- When SSL is used in a switch cluster, the SSL session terminates at the cluster commander. Cluster member switches must run standard HTTP.

- Before you configure a CA trustpoint, ensure that the system clock is set. If the clock is not set, the certificate is rejected due to an incorrect date.

Default Settings

Feature	Default Setting
Default Password and Privilege Level	
Enable password and privilege level	No password is defined. The default is level 15 (privileged EXEC level). The password is not encrypted in the configuration file.
Enable secret password and privilege level	No password is defined. The default is level 15 (privileged EXEC level). The password is encrypted before it is written to the configuration file.
Line password	No password is defined.
Security Application	
TACACS+ and AAA	Disabled.
RADIUS and AAA	Disabled.
SSL	The standard HTTP server is disabled. (Use the **ip http server** global configuration command to enable the standard HTTP server.)
	SSL is enabled.
	No CA trustpoints are configured.
	No self-signed certificates are generated.

Configuring Switch-Based Authentication

This section includes the following topics:

Protecting Access to Privileged EXEC Commands

A simple way of providing terminal access control in your network is to use passwords and assign privilege levels. Password protection restricts access to a network or network device. Privilege levels define what commands users can enter after they have logged into a network device.

Setting or Changing a Static Enable Password

The **enable password** command controls access to the privileged EXEC mode.

BEFORE YOU BEGIN

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**enable password** *password*	Define a new password or change an existing password for access to privileged EXEC mode.
		By default, no password is defined.
		For *password*, specify a string from 1 to 25 alphanumeric characters. The string cannot start with a number, is case sensitive, and allows spaces but ignores leading spaces. It can contain the question mark (?) character if you precede the question mark with the key combination Crtl-v when you create the password; for example, to create the password abc?123, do this:
		Enter **abc**.
		Enter **Crtl-v**.
		Enter **?123**.
		When the system prompts you to enter the enable password, you need not precede the question mark with the Ctrl-v; you can simply enter abc?123 at the password prompt.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show running-config**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.
		The enable password is not encrypted and can be read in the switch configuration file.

To remove the password, use the **no enable password** global configuration command.

EXAMPLE

This example shows how to change the enable password to *l1u2c3k4y5*. The password is not encrypted and provides access to level 15 (traditional privileged EXEC mode access):

```
Switch(config)# enable password l1u2c3k4y5
```

Protecting Enable and Enable Secret Passwords with Encryption

To provide an additional layer of security, particularly for passwords that cross the network or that are stored on a Trivial File Transfer Protocol (TFTP) server, you can use either the **enable password** or **enable secret** global configuration commands. Both commands accomplish the same thing; that is, you can establish an encrypted password that users must enter to access privileged EXEC mode (the default) or any privilege level you specify.

We recommend that you use the **enable secret** command because it uses an improved encryption algorithm.

If you configure the **enable secret** command, it takes precedence over the **enable password** command; the two commands cannot be in effect simultaneously.

BEFORE YOU BEGIN

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**enable password** [**level** *level*] {*password*	*encryption-type encrypted-password*}	Define a new password or change an existing password for access to privileged EXEC mode.
	or	or
	enable secret [**level** *level*] {*password*	*encryption-type encrypted-password*}	Define a secret password, which is saved using a nonreversible encryption method.
		• (Optional) For *level*, the range is from 0 to 15. Level 1 is normal user EXEC mode privileges. The default level is 15 (privileged EXEC mode privileges).
		• For *password*, specify a string from 1 to 25 alphanumeric characters. The string cannot start with a number, is case sensitive, and allows spaces but ignores leading spaces. By default, no password is defined.
		• (Optional) For *encryption-type*, only type 5, a Cisco proprietary encryption algorithm, is available. If you specify an encryption type, you must provide an encrypted password—an encrypted password that you copy from another switch configuration.
		Note If you specify an encryption type and then enter a clear text password, you can not reenter privileged EXEC mode. You cannot recover a lost encrypted password by any method.

	Command	Purpose
Step 3	**service password-encryption**	(Optional) Encrypt the password when the password is defined or when the configuration is written.
		Encryption prevents the password from being readable in the configuration file.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

If both the enable and enable secret passwords are defined, users must enter the enable secret password.

Use the **level** keyword to define a password for a specific privilege level. After you specify the level and set a password, give the password only to users who need to have access at this level. Use the **privilege level** global configuration command to specify commands accessible at various levels. For more information, see the "Configuring Multiple Privilege Levels" section on page 1-25.

If you enable password encryption, it applies to all passwords including username passwords, authentication key passwords, the privileged command password, and console and virtual terminal line passwords.

To remove a password and level, use the **no enable password** [**level** *level*] or **no enable secret** [**level** *level*] global configuration command. To disable password encryption, use the **no service password-encryption** global configuration command.

EXAMPLE

This example shows how to configure the encrypted password *1FaD0$Xyti5Rkls3LoyxzS8* for privilege level 2:

```
Switch(config)# enable secret level 2 5 $1$FaD0$Xyti5Rkls3LoyxzS8
```

Disabling Password Recovery

By default, any end user with physical access to the switch can recover from a lost password by interrupting the boot process while the switch is powering on and then entering a new password.

The password-recovery disable feature protects access to the switch password by disabling part of this functionality. When this feature is enabled, the end user can interrupt the boot process only by agreeing to set the system back to the default configuration. With password recovery disabled, you can still interrupt the boot process and change the password, but the configuration file (config.text) and the VLAN database file (vlan.dat) are deleted.

Note   If you disable password recovery, we recommend that you keep a backup copy of the configuration file on a secure server in case the end user interrupts the boot process and sets the system back to default values. Do not keep a backup copy of the configuration file on the switch. We recommend that you also keep a backup copy of the VLAN database file on a secure server. When the switch is returned to the default system configuration, you can download the saved files to the switch by using the XMODEM protocol.

BEFORE YOU BEGIN

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**no service password-recovery**	Disable password recovery.
		This setting is saved in an area of the flash memory that is accessible by the boot loader and the Cisco IOS image, but it is not part of the file system and is not accessible by any user.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show version**	Verify the configuration by checking the last few lines of the command output.

To reenable password recovery, use the **service password-recovery** global configuration command.

✎
Note   Disabling password recovery will not work if you have set the switch to boot manually by using the **boot manual** global configuration command. This command produces the boot loader prompt (*switch:*) after the switch is power cycled.

EXAMPLE

```
Switch(config)# no service password-recovery
Switch(config)# end
```

Setting a Telnet Password for a Terminal Line

When you power-up your switch for the first time, an automatic setup program runs to assign IP information and to create a default configuration for continued use. The setup program also prompts you to configure your switch for Telnet access through a password. If you did not configure this password during the setup program, you can configure it now through the command-line interface (CLI).

BEFORE YOU BEGIN

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose
Step 1		Attach a PC or workstation with emulation software to the switch console port.
		The default data characteristics of the console port are 9600, 8, 1, no parity. You might need to press the Return key several times to see the command-line prompt.
Step 2	**enable password** *password*	Enter privileged EXEC mode.
Step 3	**configure terminal**	Enter global configuration mode.

	Command	Purpose
Step 4	**line vty 0 15**	Configure the number of Telnet sessions (lines), and enter line configuration mode.
		There are 16 possible sessions on a command-capable switch. The 0 and 15 mean that you are configuring all 16 possible Telnet sessions.
Step 5	**password** *password*	Enter a Telnet password for the line or lines.
		For *password*, specify a string from 1 to 25 alphanumeric characters. The string cannot start with a number, is case sensitive, and allows spaces but ignores leading spaces. By default, no password is defined.
Step 6	**end**	Return to privileged EXEC mode.
Step 7	**show running-config**	Verify your entries.
		The password is listed under the command **line vty 0 15**.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To remove the password, use the **no password** global configuration command.

EXAMPLE

This example shows how to set the Telnet password to *let45me67in89*:

```
Switch(config)# line vty 10
Switch(config-line)# password let45me67in89
```

Configuring Username and Password Pairs

You can configure username and password pairs, which are locally stored on the switch. These pairs are assigned to lines or ports and authenticate each user before that user can access the switch. If you have defined privilege levels, you can also assign a specific privilege level (with associated rights and privileges) to each username and password pair.

BEFORE YOU BEGIN

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**username** *name* [**privilege** *level*] {**password** *encryption-type password*}	Enter the username, privilege level, and password for each user. • For *name*, specify the user ID as one word. Spaces and quotation marks are not allowed. • (Optional) For *level*, specify the privilege level the user has after gaining access. The range is 0 to 15. Level 15 gives privileged EXEC mode access. Level 1 gives user EXEC mode access. • For *encryption-type*, enter 0 to specify that an unencrypted password will follow. Enter 7 to specify that a hidden password will follow. • For *password*, specify the password the user must enter to gain access to the switch. The password must be from 1 to 25 characters, can contain embedded spaces, and must be the last option specified in the **username** command.
Step 3	**line console 0** or **line vty 0 15**	Enter line configuration mode, and configure the console port (line 0) or the VTY lines (line 0 to 15).
Step 4	**login local**	Enable local password checking at login time. Authentication is based on the username specified in Step 2.
Step 5	**end**	Return to privileged EXEC mode.
Step 6	**show running-config**	Verify your entries.
Step 7	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To disable username authentication for a specific user, use the **no username** *name* global configuration command. To disable password checking and allow connections without a password, use the **no login** line configuration command.

EXAMPLE

```
Switch(config)# username user privilege 0 password 0 cisco
Switch(config)# line console 0
Switch(config-line)# login local
Switch(config-line)# end
```

Configuring Multiple Privilege Levels

By default, the Cisco IOS software has two modes of password security: user EXEC and privileged EXEC. You can configure up to 16 hierarchical levels of commands for each mode. By configuring multiple passwords, you can allow different sets of users to have access to specified commands.

For example, if you want many users to have access to the **clear line** command, you can assign it level 2 security and distribute the level 2 password fairly widely. But if you want more restricted access to the **configure** command, you can assign it level 3 security and distribute that password to a more restricted group of users.

This section includes the following topics:

Setting the Privilege Level for a Command

BEFORE YOU BEGIN

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**privilege** *mode* **level** *level command*	Set the privilege level for a command.
		- For *mode*, enter **configure** for global configuration mode, **exec** for EXEC mode, **interface** for interface configuration mode, or **line** for line configuration mode.
		- For *level*, the range is from 0 to 15. Level 1 is for normal user EXEC mode privileges. Level 15 is the level of access permitted by the **enable** password.
		- For *command*, specify the command to which you want to restrict access.
Step 3	**enable password level** *level password*	Specify the enable password for the privilege level.
		- For *level*, the range is from 0 to 15. Level 1 is for normal user EXEC mode privileges.
		- For *password*, specify a string from 1 to 25 alphanumeric characters. The string cannot start with a number, is case sensitive, and allows spaces but ignores leading spaces. By default, no password is defined.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show running-config** or **show privilege**	Verify your entries. The first command shows the password and access level configuration. The second command shows the privilege level configuration.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

When you set a command to a privilege level, all commands whose syntax is a subset of that command are also set to that level. For example, if you set the **show ip traffic** command to level 15, the **show** commands and **show ip** commands are automatically set to privilege level 15 unless you set them individually to different levels.

To return to the default privilege for a given command, use the **no privilege** *mode* **level** *level command* global configuration command.

EXAMPLE

This example shows how to set the **configure** command to privilege level 14 and define *SecretPswd14* as the password users must enter to use level 14 commands:

```
Switch(config)# privilege exec level 14 configure
Switch(config)# enable password level 14 SecretPswd14
```

Changing the Default Privilege Level for Lines

BEFORE YOU BEGIN

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**line vty** *line*	Select the virtual terminal line on which to restrict access.
Step 3	**privilege level** *level*	Change the default privilege level for the line.
		For *level*, the range is from 0 to 15. Level 1 is for normal user EXEC mode privileges. Level 15 is the level of access permitted by the **enable** password.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show running-config**	Verify your entries.
	or	The first command shows the password and access level configuration. The second command shows the privilege level configuration.
	show privilege	
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Users can override the privilege level you set using the **privilege level** line configuration command by logging in to the line and enabling a different privilege level. They can lower the privilege level by using the **disable** command. If users know the password to a higher privilege level, they can use that password to enable the higher privilege level. You might specify a high level or privilege level for your console line to restrict line usage.

To return to the default line privilege level, use the **no privilege level** line configuration command.

EXAMPLE

```
Switch(config)# line vty 1
Switch(config-line)# privilege level 1
Switch(config-line)# end
```

Logging into and Exiting a Privilege Level

BEFORE YOU BEGIN

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose
Step 1	**enable** *level*	Log in to a specified privilege level.
		For *level*, the range is 0 to 15.
Step 2	**disable** *level*	Exit to a specified privilege level.
		For *level*, the range is 0 to 15.

EXAMPLE

In the following example, the user enters privileged EXEC mode (changes to privilege level 15) by using the **enable** command without a privilege-level argument. The system prompts the user for a password before allowing access to the privileged EXEC mode. The password is not printed to the screen. The user then exits back to user EXEC mode using the disable command. Note that the prompt for user EXEC mode is the greater than symbol (>), and the prompt for privileged EXEC mode is the number sign (#).

```
Switch> enable
Password: <letmein>
Switch# disable
Switch>
```

Configuring TACACS+

This section describes how to enable and configure Terminal Access Controller Access Control System Plus (TACACS+), which provides detailed accounting information and flexible administrative control over authentication and authorization processes. TACACS+ is facilitated through authentication, authorization, accounting (AAA) and can be enabled only through AAA commands.

To configure your switch to support TACACS+, you must identify the host or hosts maintaining the TACACS+ daemon and define the method lists for TACACS+ authentication. You can optionally define method lists for TACACS+ authorization and accounting. A method list defines the sequence and methods to be used to authenticate, to authorize, or to keep accounts on a user. You can use method lists to designate one or more security protocols to be used, thus ensuring a backup system if the initial method fails. The software uses the first method listed to authenticate, to authorize, or to keep accounts on users; if that method does not respond, the software selects the next method in the list. This process continues until there is successful communication with a listed method or the method list is exhausted.

The **aaa authorization console** global configuration command allows you to enable AAA and TACACS+ to work on the console port.

This section includes the following topics:

- Configuring TACACS+ Authorization for Privileged EXEC Access and Network Services, page 1-32
- Starting TACACS+ Accounting, page 1-33

Identifying the TACACS+ Server Host and Setting the Authentication Key

You can configure the switch to use a single server or AAA server groups to group existing server hosts for authentication. You can group servers to select a subset of the configured server hosts and use them for a particular service. The server group is used with a global server-host list and contains the list of IP addresses of the selected server hosts.

BEFORE YOU BEGIN

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**tacacs-server host** *hostname* [**port** *integer*] [**timeout** *integer*] [**key** *string*]	Identify the IP host or hosts maintaining a TACACS+ server. Enter this command multiple times to create a list of preferred hosts. The software searches for hosts in the order in which you specify them.
		• For *hostname*, specify the name or IP address of the host.
		• (Optional) For **port** *integer*, specify a server port number. The default is port 49. The range is 1 to 65535.
		• (Optional) For **timeout** *integer*, specify a time, in seconds, that the switch waits for a response from the daemon before it times out and declares an error. The default is 5 seconds. The range is 1 to 1000 seconds.
		• (Optional) For **key** *string*, specify the encryption key for encrypting and decrypting all traffic between the switch and the TACACS+ daemon. You must configure the same key on the TACACS+ daemon for encryption to be successful.
Step 3	**aaa new-model**	Enable AAA.
Step 4	**aaa group server tacacs+** *group-name*	(Optional) Define the AAA server-group with a group name. This command puts the switch in a server group subconfiguration mode.
Step 5	**server** *ip-address*	(Optional) Associate a particular TACACS+ server with the defined server group. Repeat this step for each TACACS+ server in the AAA server group. Each server in the group must be previously defined in Step 2.

	Command	Purpose
Step 6	**end**	Return to privileged EXEC mode.
Step 7	**show tacacs**	Verify your entries.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To remove the specified TACACS+ server name or address, use the **no tacacs-server host** *hostname* global configuration command. To remove a server group from the configuration list, use the **no aaa group server tacacs+** *group-name* global configuration command. To remove the IP address of a TACACS+ server, use the **no server ip-address** server group subconfiguration command.

EXAMPLE

The following example specifies that, for authentication, authorization, and accounting (AAA) confirmation, the router consults the TACACS+ server host named Sea_Cure on port number 51. The timeout value for requests on this connection is three seconds; the encryption key is a_secret.

```
Switch(config)# tacacs-server host Sea_Cure port 51 timeout 3 key a_secret
Switch(config)# aaa new-model
Switch(config)# aaa group server tacacs+
Switch(config)# tacgroup1
Switch(config)# server 0.1.1.1
```

Configuring TACACS+ Login Authentication

To configure AAA authentication, you define a named list of authentication methods and then apply that list to various ports. The method list defines the types of authentication to be performed and the sequence in which they are performed; it must be applied to a specific port before any of the defined authentication methods are performed. The only exception is the default method list (which, by coincidence, is named *default*). The default method list is automatically applied to all ports except those that have a named method list explicitly defined. A defined method list overrides the default method list.

A method list describes the sequence and authentication methods to be queried to authenticate a user. You can designate one or more security protocols to be used for authentication, thus ensuring a backup system for authentication in case the initial method fails. The software uses the first method listed to authenticate users; if that method fails to respond, the software selects the next authentication method in the method list. This process continues until there is successful communication with a listed authentication method or until all defined methods are exhausted. If authentication fails at any point in this cycle—meaning that the security server or local username database responds by denying the user access—the authentication process stops, and no other authentication methods are attempted.

BEFORE YOU BEGIN

Complete the .

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**aaa new-model**	Enable AAA.

	Command	Purpose
Step 3	**aaa authentication login** {**default** | *list-name*} *method1* [*method2*...]	Create a login authentication method list. • To create a default list that is used when a named list is *not* specified in the **login authentication** command, use the **default** keyword followed by the methods that are to be used in default situations. The default method list is automatically applied to all ports. • For *list-name*, specify a character string to name the list you are creating. • For *method1*..., specify the actual method the authentication algorithm tries. The additional methods of authentication are used only if the previous method returns an error, not if it fails. Select one of these methods: • **enable**—Use the enable password for authentication. Before you can use this authentication method, you must define an enable password by using the **enable** *password* global configuration command. • **group tacacs+**—Use TACACS+ authentication. Before you can use this authentication method, you must configure the TACACS+ server. For more information, see the "Identifying the TACACS+ Server Host and Setting the Authentication Key" section on page 1-29. • **line**—Use the line password for authentication. Before you can use this authentication method, you must define a line password. Use the **password** *password* line configuration command. • **local**—Use the local username database for authentication. You must enter username information in the database. Use the **username** *password* global configuration command. • **local-case**—Use a case-sensitive local username database for authentication. You must enter username information in the database by using the **username** *name* **password** global configuration command. • **none**—Do not use any authentication for login.
Step 4	**line** [**console** | **tty** | **vty**] *line-number* [*ending-line-number*]	Enter line configuration mode, and configure the lines to which you want to apply the authentication list.
Step 5	**login authentication** {**default** | *list-name*}	Apply the authentication list to a line or set of lines. • If you specify **default**, use the default list created with the **aaa authentication login** command. • For *list-name*, specify the list created with the **aaa authentication login** command.
Step 6	**end**	Return to privileged EXEC mode.

	Command	Purpose
Step 7	show running-config	Verify your entries.
Step 8	copy running-config startup-config	(Optional) Save your entries in the configuration file.

To disable AAA, use the **no aaa new-model** global configuration command. To disable AAA authentication, use the **no aaa authentication login** {**default** | *list-name*} *method1* [*method2*...] global configuration command. To either disable TACACS+ authentication for logins or to return to the default value, use the **no login authentication** {**default** | *list-name*} line configuration command.

EXAMPLE

The following example shows how to create an AAA authentication list called MIS-access. This authentication first tries to contact a TACACS+ server. If no server is found, TACACS+ returns an error and AAA tries to use the enable password. If this attempt also returns an error (because no enable password is configured on the server), the user is allowed access with no authentication.

```
Switch(config)# aaa new-model
Switch(config)# aaa authentication login MIS-access group tacacs+ enable none
Switch(config)# line vty 0 4
Switch(config-line)# login authentication MIS-access
Switch(config-line)# end
```

Configuring TACACS+ Authorization for Privileged EXEC Access and Network Services

AAA authorization limits the services available to a user. When AAA authorization is enabled, the switch uses information retrieved from the user's profile, which is located either in the local user database or on the security server, to configure the user's session. The user is granted access to a requested service only if the information in the user profile allows it.

You can use the **aaa authorization** global configuration command with the **tacacs+** keyword to set parameters that restrict a user's network access to privileged EXEC mode.

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

- Use TACACS+ for privileged EXEC access authorization if authentication was performed by using TACACS+.
- Use the local database if authentication was not performed by using TACACS+.

Note Authorization is bypassed for authenticated users who log in through the CLI even if authorization has been configured.

BEFORE YOU BEGIN

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	aaa authorization network tacacs+	Configure the switch for user TACACS+ authorization for all network-related service requests.
Step 3	aaa authorization exec tacacs+	Configure the switch for user TACACS+ authorization if the user has privileged EXEC access.
		The **exec** keyword might return user profile information (such as **autocommand** information).
Step 4	end	Return to privileged EXEC mode.
Step 5	show running-config	Verify your entries.
Step 6	copy running-config startup-config	(Optional) Save your entries in the configuration file.

To disable authorization, use the **no aaa authorization** {**network** | **exec**} *method1* global configuration command.

EXAMPLE

```
Switch(config)# aaa authorization network tacacs+
Switch(config)# aaa authorization exec tacacs+
Switch(config-line)# end
```

Starting TACACS+ Accounting

The AAA accounting feature tracks the services that users are accessing and the amount of network resources that they are consuming. When AAA accounting is enabled, the switch reports user activity to the TACACS+ security server in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server. This data can then be analyzed for network management, client billing, or auditing.

Follow this procedure to enable TACACS+ accounting for each Cisco IOS privilege level and for network services.

BEFORE YOU BEGIN

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	aaa accounting network start-stop tacacs+	Enable TACACS+ accounting for all network-related service requests.
Step 3	aaa accounting exec start-stop tacacs+	Enable TACACS+ accounting to send a start-record accounting notice at the beginning of a privileged EXEC process and a stop-record at the end.

	Command	Purpose
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show running-config**	Verify your entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To disable accounting, use the **no aaa accounting** {**network** | **exec**} {**start-stop**} *method1...* global configuration command.

EXAMPLE

```
Switch(config)# aaa accounting network start-stop tacacs+
Switch(config)# aaa accounting exec start-stop tacacs+
Switch(config-line)# end
```

Configuring RADIUS

This section describes how to enable and configure RADIUS, which provides detailed accounting information and flexible administrative control over authentication and authorization processes. RADIUS is facilitated through AAA and can be enabled only through AAA commands.

To configure your switch to support RADIUS, you must identify the host or hosts that run the RADIUS server software and define the method lists for RADIUS authentication. You can optionally define method lists for RADIUS authorization and accounting.

A method list defines the sequence and methods to be used to authenticate, to authorize, or to keep accounts on a user. You can use method lists to designate one or more security protocols to be used (such as TACACS+ or local username lookup), thus ensuring a backup system if the initial method fails. The software uses the first method listed to authenticate, to authorize, or to keep accounts on users; if that method does not respond, the software selects the next method in the list. This process continues until there is successful communication with a listed method or the method list is exhausted.

You should have access to and should configure a RADIUS server before configuring RADIUS features on your switch.

This section includes the following topics:

Identifying the RADIUS Server Host

Switch-to-RADIUS-server communication involves several components:

- Hostname or IP address
- Authentication destination port
- Accounting destination port
- Key string
- Timeout period
- Retransmission value

You identify RADIUS security servers by their hostname or IP address, hostname and specific UDP port numbers, or their IP address and specific UDP port numbers. The combination of the IP address and the UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. This unique identifier enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address.

If two different host entries on the same RADIUS server are configured for the same service—for example, accounting—the second host entry configured acts as a fail-over backup to the first one. Using this example, if the first host entry fails to provide accounting services, the switch tries the second host entry configured on the same device for accounting services. (The RADIUS host entries are tried in the order that they are configured.)

A RADIUS server and the switch 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 (key) string that it shares with the switch.

The timeout, retransmission, and encryption key values can be configured globally for all RADIUS servers, on a per-server basis, or in some combination of global and per-server settings. To apply these settings globally to all RADIUS servers communicating with the switch, use the three unique global configuration commands: **radius-server timeout**, **radius-server retransmit**, and **radius-server key**. To apply these values on a specific RADIUS server, use the **radius-server host** global configuration command.

You can configure the switch to use AAA server groups to group existing server hosts for authentication. For more information, see the "Defining AAA Server Groups" section on page 1-39.

Follow this procedure to configure per-server RADIUS server communication. This procedure is required.

BEFORE YOU BEGIN

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**aaa new-model**	Enable AAA authentication.

	Command	Purpose
Step 3	**radius-server host** {*hostname* | *ip-address*} [**auth-port** *port-number*] [**acct-port** *port-number*] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*]	Specify the IP address or hostname of the remote RADIUS server host. • (Optional) For **auth-port** *port-number*, specify the UDP destination port for authentication requests. • (Optional) For **acct-port** *port-number*, specify the UDP destination port for accounting requests. • (Optional) For **timeout** *seconds*, specify the time interval that the switch waits for the RADIUS server to reply before resending. The range is 1 to 1000. This setting overrides the **radius-server timeout** global configuration command setting. If no timeout is set with the **radius-server host** command, the setting of the **radius-server timeout** command is used. • (Optional) For **retransmit** *retries*, specify the number of times a RADIUS request is resent to a server if that server is not responding or responding slowly. The range is 1 to 1000. If no retransmit value is set with the **radius-server host** command, the setting of the **radius-server retransmit** global configuration command is used. • (Optional) For **key** *string*, specify the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server. **Note** The key is a text string that must match the encryption key used on the RADIUS server. Always configure the key as the last item in the **radius-server host** command. Leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks are part of the key. To configure the switch to recognize more than one host entry associated with a single IP address, enter this command as many times as necessary, making sure that each UDP port number is different. The switch software searches for hosts in the order in which you specify them. Set the timeout, retransmit, and encryption key values to use with the specific RADIUS host.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show running-config**	Verify your entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To remove the specified RADIUS server, use the **no radius-server host** *hostname* \| *ip-address* global configuration command.

EXAMPLE

This example shows how to configure one RADIUS server to be used for authentication and another to be used for accounting:

```
Switch(config)# radius-server host 172.29.36.49 auth-port 1612 key rad1
Switch(config)# radius-server host 172.20.36.50 acct-port 1618 key rad2
```

This example shows how to configure *host1* as the RADIUS server and to use the default ports for both authentication and accounting:

```
Switch(config)# radius-server host host1
```

Note   You also need to configure some settings on the RADIUS server. These settings include the IP address of the switch and the key string to be shared by both the server and the switch. For more information, see the RADIUS server documentation.

Configuring RADIUS Login Authentication

To configure AAA authentication, you define a named list of authentication methods and then apply that list to various ports. The method list defines the types of authentication to be performed and the sequence in which they are performed; it must be applied to a specific port before any of the defined authentication methods are performed. The only exception is the default method list (which, by coincidence, is named *default*). The default method list is automatically applied to all ports except those that have a named method list explicitly defined.

A method list describes the sequence and authentication methods to be queried to authenticate a user. You can designate one or more security protocols to be used for authentication, thus ensuring a backup system for authentication in case the initial method fails. The software uses the first method listed to authenticate users; if that method fails to respond, the software selects the next authentication method in the method list. This process continues until there is successful communication with a listed authentication method or until all defined methods are exhausted. If authentication fails at any point in this cycle—meaning that the security server or local username database responds by denying the user access—the authentication process stops, and no other authentication methods are attempted.

Follow this procedure to configure login authentication. This procedure is required.

BEFORE YOU BEGIN

Before defining a method list that uses any of the following authentication methods, perform the required configuration for that method:

- Enable password—Define an enable password by using the **enable** *password* global configuration command.

- RADIUS authentication —Configure the RADIUS server. For more information, see the .

- Line password—Define a line password by using the **password** *password* line configuration command.

- Local username database—Enter username information in the database using the **username** *name* **password** global configuration command.

- Case-sensitive local username database—Enter username information in the database by using the **username** *password* global configuration command.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**aaa new-model**	Enable AAA.
Step 3	**aaa authentication login** {**default** | *list-name*} *method1* [*method2*...]	Create a login authentication method list. • To create a default list that is used when a named list is *not* specified in the **login authentication** command, use the **default** keyword followed by the methods that are to be used in default situations. The default method list is automatically applied to all ports. • For *list-name*, specify a character string to name the list you are creating. • For *method1*..., specify the actual method the authentication algorithm tries. The additional methods of authentication are used only if the previous method returns an error, not if it fails. Select one of these methods: – **enable**—Use the enable password for authentication. – **group radius**—Use RADIUS authentication. – **line**—Use the line password for authentication. – **local**—Use the local username database for authentication. – **local-case**—Use a case-sensitive local username database for authentication. – **none**—Do not use any authentication for login.
Step 4	**line** [**console** | **tty** | **vty**] *line-number* [*ending-line-number*]	Enter line configuration mode, and configure the lines to which you want to apply the authentication list.

	Command	Purpose
Step 5	**login authentication** {**default** | *list-name*}	Apply the authentication list to a line or set of lines. • If you specify **default**, use the default list created with the **aaa authentication login** command. • For *list-name*, specify the list created with the **aaa authentication login** command.
Step 6	**end**	Return to privileged EXEC mode.
Step 7	**show running-config**	Verify your entries.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To disable AAA, use the **no aaa new-model** global configuration command. To disable AAA authentication, use the **no aaa authentication login** {**default** \| *list-name*} *method1* [*method2*...] global configuration command. To either disable RADIUS authentication for logins or to return to the default value, use the **no login authentication** {**default** \| *list-name*} line configuration command.

EXAMPLE

The following example shows how to create a default AAA authentication list. This authentication first tries to contact a RADIUS server. If no server is found, RADIUS returns an error and AAA tries to use the password in the local username database.

```
Switch(config)# aaa new-model
Switch(config)# aaa authentication login default group radius local
Switch(config)# login authentication default
Switch(config)# end
```

Defining AAA Server Groups

You can configure the switch to use AAA server groups to group existing server hosts for authentication. You select a subset of the configured server hosts and use them for a particular service. The server group is used with a global server-host list, which lists the IP addresses of the selected server hosts.

Server groups also can include multiple host entries for the same server if each entry has a unique identifier (the combination of the IP address and UDP port number), allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. If you configure two different host entries on the same RADIUS server for the same service, (for example, accounting), the second configured host entry acts as a fail-over backup to the first one.

You use the **server** group server configuration command to associate a particular server with a defined group server. You can either identify the server by its IP address or identify multiple host instances or entries by using the optional **auth-port** and **acct-port** keywords.

BEFORE YOU BEGIN

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**radius-server host** {*hostname* | *ip-address*} [**auth-port** *port-number*] [**acct-port** *port-number*] [**timeout** *seconds*] [**retransmit** *retries*] [**key** *string*]	Specify the IP address or hostname of the remote RADIUS server host. • (Optional) For **auth-port** *port-number*, specify the UDP destination port for authentication requests. • (Optional) For **acct-port** *port-number*, specify the UDP destination port for accounting requests. • (Optional) For **timeout** *seconds*, specify the time interval that the switch waits for the RADIUS server to reply before resending. The range is 1 to 1000. This setting overrides the **radius-server timeout** global configuration command setting. If no timeout is set with the **radius-server host** command, the setting of the **radius-server timeout** command is used. • (Optional) For **retransmit** *retries*, specify the number of times a RADIUS request is resent to a server if that server is not responding or responding slowly. The range is 1 to 1000. If no retransmit value is set with the **radius-server host** command, the setting of the **radius-server retransmit** global configuration command is used. • (Optional) For **key** *string*, specify the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server. **Note** The key is a text string that must match the encryption key used on the RADIUS server. Always configure the key as the last item in the **radius-server host** command. Leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks are part of the key. To configure the switch to recognize more than one host entry associated with a single IP address, enter this command as many times as necessary, making sure that each UDP port number is different. The switch software searches for hosts in the order in which you specify them. Set the timeout, retransmit, and encryption key values to use with the specific RADIUS host.
Step 3	**aaa new-model**	Enable AAA.
Step 4	**aaa group server radius** *group-name*	Define the AAA server-group with a group name. This command puts the switch in a server group configuration mode.

	Command	Purpose
Step 5	**server** *ip-address*	Associate a particular RADIUS server with the defined server group. Repeat this step for each RADIUS server in the AAA server group.
		Each server in the group must be previously defined in Step 2.
Step 6	**end**	Return to privileged EXEC mode.
Step 7	**show running-config**	Verify your entries.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.
Step 9		Enable RADIUS login authentication. See the "Configuring RADIUS Login Authentication" section on page 1-37.

To remove the specified RADIUS server, use the **no radius-server host** *hostname | ip-address* global configuration command. To remove a server group from the configuration list, use the **no aaa group server radius** *group-name* global configuration command. To remove the IP address of a RADIUS server, use the **no server** *ip-address* server group configuration command.

EXAMPLE

In this example, the switch is configured to recognize two different RADIUS group servers (*group1* and *group2*). Group1 has two different host entries on the same RADIUS server configured for the same services. The second host entry acts as a fail-over backup to the first entry.

```
Switch(config)# radius-server host 172.20.0.1 auth-port 1000 acct-port 1001
Switch(config)# radius-server host 172.10.0.1 auth-port 1645 acct-port 1646
Switch(config)# aaa new-model
Switch(config)# aaa group server radius group1
Switch(config-sg-radius)# server 172.20.0.1 auth-port 1000 acct-port 1001
Switch(config-sg-radius)# exit
Switch(config)# aaa group server radius group2
Switch(config-sg-radius)# server 172.20.0.1 auth-port 2000 acct-port 2001
Switch(config-sg-radius)# exit
```

Configuring RADIUS Authorization for User Privileged Access and Network Services

AAA authorization limits the services available to a user. When AAA authorization is enabled, the switch uses information retrieved from the user's profile, which is in the local user database or on the security server, to configure the user's session. The user is granted access to a requested service only if the information in the user profile allows it.

You can use the **aaa authorization** global configuration command with the **radius** keyword to set parameters that restrict a user's network access to privileged EXEC mode.

The **aaa authorization exec radius local** command sets these authorization parameters:

- Use RADIUS for privileged EXEC access authorization if authentication was performed by using RADIUS.

- Use the local database if authentication was not performed by using RADIUS.

Note   Authorization is bypassed for authenticated users who log in through the CLI even if authorization has been configured.

BEFORE YOU BEGIN

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**aaa authorization network radius**	Configure the switch for user RADIUS authorization for all network-related service requests.
Step 3	**aaa authorization exec radius**	Configure the switch for user RADIUS authorization if the user has privileged EXEC access.
		The **exec** keyword might return user profile information (such as **autocommand** information).
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show running-config**	Verify your entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To disable authorization, use the **no aaa authorization** {**network** | **exec**} *method1* global configuration command.

EXAMPLE

```
Switch(config)# aaa authorization network radius
Switch(config)# aaa authorization exec radius
Switch(config)# end
```

Starting RADIUS Accounting

The AAA accounting feature tracks the services that users are accessing and the amount of network resources that they are consuming. When AAA accounting is enabled, the switch reports user activity to the RADIUS security server in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server. This data can then be analyzed for network management, client billing, or auditing.

Follow this procedure to enable RADIUS accounting for each Cisco IOS privilege level and for network services.

BEFORE YOU BEGIN

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**aaa accounting network start-stop radius**	Enable RADIUS accounting for all network-related service requests.
Step 3	**aaa accounting exec start-stop radius**	Enable RADIUS accounting to send a start-record accounting notice at the beginning of a privileged EXEC process and a stop-record at the end.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show running-config**	Verify your entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To disable accounting, use the **no aaa accounting** {**network** | **exec**} {**start-stop**} *method1*... global configuration command.

EXAMPLE

```
Switch(config)# aaa accounting network start-stop radius
Switch(config)# aaa accounting exec start-stop radius
Switch(config)# end
```

Configuring Settings for All RADIUS Servers

BEFORE YOU BEGIN

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**radius-server key** *string*	Specify the shared secret text string used between the switch and all RADIUS servers.
		Note The key is a text string that must match the encryption key used on the RADIUS server. Leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks are part of the key.
Step 3	**radius-server retransmit** *retries*	Specify the number of times the switch sends each RADIUS request to the server before giving up. The default is 3; the range 1 to 1000.
Step 4	**radius-server timeout** *seconds*	Specify the number of seconds a switch waits for a reply to a RADIUS request before resending the request. The default is 5 seconds; the range is 1 to 1000.

	Command	Purpose
Step 5	**radius-server deadtime** *minutes*	Specify the length of time, in minutes, for which a RADIUS server that is not responding to authentication requests is skipped over by transaction requests. Setting the deadtime avoids the wait for the request to timeout before trying the next configured server.
		The default is 0; the range is 1 to 1440 minutes.
Step 6	**end**	Return to privileged EXEC mode.
Step 7	**show running-config**	Verify your settings.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return to the default setting for the retransmit, timeout, and deadtime, use the **no** forms of these commands.

EXAMPLE

```
Switch(config)# radius-server key key1
Switch(config)# radius-server retransmit 5
Switch(config)# radius-server timeout 10
Switch(config)# radius-server deadtime 5
Switch(config)# end
```

Configuring the Switch to Use Vendor-Specific RADIUS Attributes

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific information between the switch and the RADIUS server by using the vendor-specific attribute (attribute 26). Vendor-specific attributes (VSAs) allow vendors to support their own extended attributes not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option by 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 with this format:

```
protocol : attribute sep value *
```

Protocol is a value of the Cisco protocol attribute for a particular type of authorization. *Attribute* and *value* are an appropriate attribute-value (AV) pair defined in the Cisco TACACS+ specification, and *sep* is = for mandatory attributes and is * for optional attributes. The full set of features available for TACACS+ authorization can then be used for RADIUS.

For example, this AV pair activates Cisco's *multiple named ip address pools* feature during IP authorization (during PPP IPCP address assignment):

```
cisco-avpair= "ip:addr-pool=first"
```

Other vendors have unique vendor-IDs, options, and associated VSAs. For more information about vendor-IDs and VSAs, see RFC 2138, "Remote Authentication Dial-In User Service (RADIUS)."

Follow this procedure to configure the switch to recognize and use VSAs.

BEFORE YOU BEGIN

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**radius-server vsa send** [**accounting** | **authentication**]	Enable the switch to recognize and use VSAs as defined by RADIUS IETF attribute 26. • (Optional) Use the **accounting** keyword to limit the set of recognized vendor-specific attributes to only accounting attributes. • (Optional) Use the **authentication** keyword to limit the set of recognized vendor-specific attributes to only authentication attributes. If you enter this command without keywords, both accounting and authentication vendor-specific attributes are used.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show running-config**	Verify your settings.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

For a complete list of RADIUS attributes or more information about vendor-specific attribute 26, see the *RADIUS Attributes Configuration Guide, Cisco IOS Release 15M&T*.

EXAMPLE

This example shows how to provide a user logging in from a switch with immediate access to privileged EXEC commands:

```
cisco-avpair= ”shell:priv-lvl=15“
```

This example shows how to specify an authorized VLAN in the RADIUS server database:

```
cisco-avpair= ”tunnel-type(#64)=VLAN(13)”
cisco-avpair= ”tunnel-medium-type(#65)=802 media(6)”
cisco-avpair= ”tunnel-private-group-ID(#81)=vlanid”
```

This example shows how to apply an input ACL in ASCII format to an interface for the duration of this connection:

```
cisco-avpair= “ip:inacl#1=deny ip 10.10.10.10 0.0.255.255 20.20.20.20 255.255.0.0”
cisco-avpair= “ip:inacl#2=deny ip 10.10.10.10 0.0.255.255 any”
cisco-avpair= “mac:inacl#3=deny any any decnet-iv”
```

This example shows how to apply an output ACL in ASCII format to an interface for the duration of this connection:

```
cisco-avpair= “ip:outacl#2=deny ip 10.10.10.10 0.0.255.255 any”
```

Configuring the Switch for Vendor-Proprietary RADIUS Server Communication

Although an IETF draft standard for RADIUS specifies a method for communicating vendor-proprietary information between the switch and the RADIUS server, some vendors have extended the RADIUS attribute set in a unique way. Cisco IOS software supports a subset of vendor-proprietary RADIUS attributes.

As mentioned earlier, to configure RADIUS (whether vendor-proprietary or IETF draft-compliant), you must specify the host running the RADIUS server daemon and the secret text string it shares with the switch. You specify the RADIUS host and secret text string by using the **radius-server** global configuration commands.

Follow this procedure to specify a vendor-proprietary RADIUS server host and a shared secret text string.

BEFORE YOU BEGIN

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**radius-server host** {*hostname* | *ip-address*} **non-standard**	Specify the IP address or hostname of the remote RADIUS server host and identify that it is using a vendor-proprietary implementation of RADIUS.
Step 3	**radius-server key** *string*	Specify the shared secret text string used between the switch and the vendor-proprietary RADIUS server. The switch and the RADIUS server use this text string to encrypt passwords and exchange responses.
		Note The key is a text string that must match the encryption key used on the RADIUS server. Leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks are part of the key.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show running-config**	Verify your settings.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To delete the vendor-proprietary RADIUS host, use the **no radius-server host** {*hostname* \| *ip-address*} **non-standard** global configuration command. To disable the key, use the **no radius-server key** global configuration command.

EXAMPLE

This example shows how to specify a vendor-proprietary RADIUS host and to use a secret key of *rad124* between the switch and the server:

```
Switch(config)# radius-server host 172.20.30.15 nonstandard
Switch(config)# radius-server key rad124
```

Configuring CoA on the Switch

This procedure is required.

BEFORE YOU BEGIN

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**aaa new-model**	Enable AAA.
Step 3	**aaa server radius dynamic-author**	Configure the switch as an authentication, authorization, and accounting (AAA) server to facilitate interaction with an external policy server.
Step 4	**client** {*ip-address*	*name*} [**vrf** *vrfname*] [**server-key** *string*]	Enter dynamic authorization local server configuration mode, and specify a RADIUS client from which a device accepts CoA and disconnect requests.
Step 5	**server-ke**y [**0**	**7**] *string*	Configure the RADIUS key to be shared between a device and RADIUS clients.
Step 6	**port** *port-number*	Specify the port on which a device listens for RADIUS requests from configured RADIUS clients.
Step 7	**auth-type** {**any**	**all**	**session-key**}	Specify the type of authorization the switch uses for RADIUS clients. The client must match all the configured attributes for authorization.
Step 8	**ignore session-key**	(Optional) Configure the switch to ignore the session-key. For more information about the **ignore** command, see the *Cisco IOS Intelligent Services Gateway Command Reference* on Cisco.com.
Step 9	**ignore server-key**	(Optional) Configure the switch to ignore the server-key. For more information about the **ignore** command, see the *Cisco IOS Intelligent Services Gateway Command Reference* on Cisco.com.
Step 10	**authentication command bounce-port ignore**	(Optional) Configure the switch to ignore a CoA request to temporarily disable the port hosting a session. The purpose of temporarily disabling the port is to trigger a DHCP renegotiation from the host when a VLAN change occurs and there is no supplicant on the endpoint to detect the change.
Step 11	**authentication command disable-port ignore**	(Optional) Configure the switch to ignore a nonstandard command requesting that the port hosting a session be administratively shut down. Shutting down the port terminates the session. Use standard CLI or SNMP commands to reenable the port.

	Command	Purpose
Step 12	**end**	Return to privileged EXEC mode.
Step 13	**show running-config**	Verify your entries.
Step 14	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To disable AAA, use the **no aaa new-model** global configuration command. To disable the AAA server functionality on the switch, use the **no aaa server radius dynamic authorization** global configuration command.

EXAMPLE

```
Switch(config)# aaa new-model
Switch(config)# aaa server radius dynamic-author
Switch(config-locsvr-da-radius)# client 10.0.0.1
Switch(config-locsvr-da-radius)# server-key cisco123
Switch(config-locsvr-da-radius)# port 3799
Switch(config-locsvr-da-radius)# auth-type all
Switch(config-locsvr-da-radius)# ignore session-key
Switch(config-locsvr-da-radius)# ignore server-key
Switch(config-locsvr-da-radius)# exit
```

Monitoring and Troubleshooting CoA Functionality

Use the following commands to monitor and troubleshoot CoA on the switch:

- **debug radius**
- **debug aaa coa**
- **debug aaa pod**
- **debug aaa subsys**
- **debug cmdhd** [**detail** | **error** | **events**]
- **show aaa attributes protocol radius**

Configuring RADIUS Server Load Balancing

This feature allows access and authentication requests to be evenly distributed across all RADIUS servers in a server group. For more information, see the "RADIUS Server Load Balancing" chapter of the *RADIUS Configuration Guide, Cisco IOS Release 15M&T*.

Configuring Kerberos

The Kerberos security system authenticates requests for network resources by using a trusted third party.

For Kerberos configuration examples, see the "Configuring Kerberos" chapter of the *User Security Configuration Guide, Cisco IOS Release 15MT*.

For complete syntax and usage information for the commands used to configure Kerberos, see the *Cisco IOS Security Command Reference*.

Note   In the Kerberos configuration examples and in the *Cisco IOS Security Command Reference,* the trusted third party can be a IE 2000U switch that supports Kerberos, that is configured as a network security server, and that can authenticate users by using the Kerberos protocol.

So that remote users can authenticate to network services, you must configure the hosts and the KDC in the Kerberos realm to communicate and mutually authenticate users and network services. To do this, you must identify them to each other. You add entries for the hosts to the Kerberos database on the KDC and add KEYTAB files generated by the KDC to all hosts in the Kerberos realm. You also create entries for the users in the KDC database.

When you add or create entries for the hosts and users, follow these guidelines:

- The Kerberos principal name *must* be in all lowercase characters.
- The Kerberos instance name *must* be in all lowercase characters.
- The Kerberos realm name *must* be in all uppercase characters.

Note   A Kerberos server can be a IE 2000U switch that is configured as a network security server and that can authenticate users by using the Kerberos protocol.

To set up a Kerberos-authenticated server-client system, follow these steps:

1.   Configure the KDC by using Kerberos commands.

2.   Configure the switch to use the Kerberos protocol.

For instructions, see the "How to Configure Kerberos" section in the *User Security Configuration Guide, Cisco IOS Release 15MT*.

Configuring the Switch for Local Authentication and Authorization

You can configure AAA to operate without a server by setting the switch to implement AAA in local mode. The switch then handles authentication and authorization. No accounting is available in this configuration.

BEFORE YOU BEGIN

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**aaa new-model**	Enable AAA.
Step 3	**aaa authentication login default local**	Set the login authentication to use the local username database. The **default** keyword applies the local user database authentication to all ports.
Step 4	**aaa authorization exec local**	Configure user AAA authorization, check the local database, and allow the user to run an EXEC shell.

	Command	Purpose
Step 5	**aaa authorization network local**	Configure user AAA authorization for all network-related service requests.
Step 6	**username** *name* [**privilege** *level*] {**password** *encryption-type password*}	Enter the local database, and establish a username-based authentication system.
		Repeat this command for each user.
		• For *name*, specify the user ID as one word. Spaces and quotation marks are not allowed.
		• (Optional) For *level*, specify the privilege level the user has after gaining access. The range is 0 to 15. Level 15 gives privileged EXEC mode access. Level 0 gives user EXEC mode access.
		• For *encryption-type*, enter 0 to specify that an unencrypted password follows. Enter 7 to specify that a hidden password follows.
		• For *password*, specify the password the user must enter to gain access to the switch. The password must be from 1 to 25 characters, can contain embedded spaces, and must be the last option specified in the **username** command.
Step 7	**end**	Return to privileged EXEC mode.
Step 8	**show running-config**	Verify your entries.
Step 9	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To disable AAA, use the **no aaa new-model** global configuration command. To disable authorization, use the **no aaa authorization** {**network** | **exec**} *method1* global configuration command.

EXAMPLE

```
Switch(config)# aaa new-model
Switch(config)# aaa authentication login default local
Switch(config)# aaa authorization exec local
Switch(config)# aaa authorization network local
Switch(config)# username user2 privilege 2 password 0 cisco
Switch(config)# end
```

Configuring SSH

This section includes the following topics:

Setting Up the Switch to Run SSH

Follow these steps to set up your switch to run SSH:

1. Download the cryptographic software image from Cisco.com. This step is required. For more information, see the release notes for this release.

2. Configure a hostname and IP domain name for the switch. Follow this procedure only if you are configuring the switch as an SSH server.

3. Generate an RSA key pair for the switch, which automatically enables SSH. Follow this procedure only if you are configuring the switch as an SSH server.

4. Configure user authentication for local or remote access. This step is required. For more information, see the "Configuring the Switch for Local Authentication and Authorization" section on page 1-49.

Follow this procedure to configure a hostname and an IP domain name and to generate an RSA key pair. This procedure is required if you are configuring the switch as an SSH server.

BEFORE YOU BEGIN

Follow these guidelines when configuring the switch as an SSH server or SSH client:

- An RSA key pair generated by a SSHv1 server can be used by an SSHv2 server, and the reverse.

- If you get CLI error messages after entering the **crypto key generate rsa** global configuration command, an RSA key pair has not been generated. Reconfigure the hostname and domain, and then enter the **crypto key generate rsa** command. For more information, see the "Setting Up the Switch to Run SSH" section on page 1-51.

- When generating the RSA key pair, the message `No host name specified` might appear. If it does, you must configure a hostname by using the **hostname** global configuration command.

- When generating the RSA key pair, the message `No domain specified` might appear. If it does, you must configure an IP domain name by using the **ip domain-name** global configuration command.

- When configuring the local authentication and authorization authentication method, make sure that AAA is disabled on the console.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**hostname** *hostname*	Configure a hostname for your switch.
Step 3	**ip domain-name** *domain_name*	Configure a host domain for your switch.
Step 4	**crypto key generate rsa**	Enable the SSH server for local and remote authentication on the switch and generate an RSA key pair. We recommend a minimum modulus size of 1024 bits. When you generate RSA keys, you are prompted to enter a modulus length. A longer modulus length might be more secure, but it takes longer to generate and to use.
Step 5	**end**	Return to privileged EXEC mode.

	Command	Purpose
Step 6	show ip ssh or show ssh	Show the version and configuration information for your SSH server. Show the status of the SSH server on the switch.
Step 7	copy running-config startup-config	(Optional) Save your entries in the configuration file.

To delete the RSA key pair, use the **crypto key zeroize rsa** global configuration command. After the RSA key pair is deleted, the SSH server is automatically disabled.

EXAMPLE

```
Switch(config)# crypto key generate rsa
Switch(config)# end
```

Configuring the SSH Server

BEFORE YOU BEGIN

Complete the "Setting Up the Switch to Run SSH" procedure on page 1-51.

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal	Enter global configuration mode.
Step 2	ip ssh version [1 | 2]	(Optional) Configure the switch to run SSH Version 1 or SSH Version 2. • **1**—Configure the switch to run SSH Version 1. • **2**—Configure the switch to run SSH Version 2. If you do not enter this command or do not specify a keyword, the SSH server selects the latest SSH version supported by the SSH client. For example, if the SSH client supports SSHv1 and SSHv2, the SSH server selects SSHv2.

	Command	Purpose
Step 3	ip ssh {timeout *seconds* | authentication-retries *number*}	Configure the SSH control parameters: • Specify the timeout value in seconds; the default is 120 seconds. The range is 0 to 120 seconds. This parameter applies to the SSH negotiation phase. After the connection is established, the switch uses the default timeout values of the CLI-based sessions. By default, up to five simultaneous, encrypted SSH connections for multiple CLI-based sessions over the network are available (session 0 to session 4). After the execution shell starts, the CLI-based session timeout value returns to the default of 10 minutes. • Specify the number of times that a client can re-authenticate to the server. The default is 3; the range is 0 to 5. Repeat this step when configuring both parameters.
Step 4	end	Return to privileged EXEC mode.
Step 5	show ip ssh or show ssh	Show the version and configuration information for your SSH server. Show the status of the SSH server connections on the switch.
Step 6	copy running-config startup-config	(Optional) Save your entries in the configuration file.

To return to the default SSH control parameters, use the **no ip ssh** {**timeout** | **authentication-retries**} global configuration command.

EXAMPLE

```
Switch(config)# ip ssh timeout 30
Switch(config)# end
```

Using SSH Keyboard Interactive Authentication

The SSH Keyboard Interactive Authentication feature, also known as Generic Message Authentication for SSH, is a method that can be used to implement different types of authentication mechanisms. Basically, any currently supported authentication method that requires only user input can be performed with this feature. The feature is automatically deployed.

Supported methods:

- Password
- SecurID and hardware tokens printing a number or a string in response to a challenge sent by the server
- Pluggable Authentication Module (PAM)
- S/KEY (and other One-Time-Pads)

For SSH keyboard interactive authentication to work, ensure that the **Apply password change rule** checkbox is checked on the Authentication Server Group Setup page on the RADIUS or TACACS server. The keyboard interactive authentication method works only with SSH V2 and the blank password mechanism is supported only with TACACS authentication.

Configuring Secure HTTP Servers and Clients

Secure Socket Layer (SSL) version 3.0 support for the HTTP 1.1 server and client provides server authentication, encryption, and message integrity, as well as HTTP client authentication, to allow secure HTTP communications.

This section includes the following topics:

- Configuring a CA Trustpoint, page 1-54
- Configuring the Secure HTTP Server, page 1-55
- Configuring the Secure HTTP Client, page 1-57

Configuring a CA Trustpoint

For secure HTTP connections, we recommend that you configure an official CA trustpoint. A CA trustpoint is more secure than a self-signed certificate.

BEFORE YOU BEGIN

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**hostname** *hostname*	Specify the hostname of the switch (required only if you have not previously configured a hostname). The hostname is required for security keys and certificates.
Step 3	**ip domain-name** *domain-name*	Specify the IP domain name of the switch (required only if you have not previously configured an IP domain name). The domain name is required for security keys and certificates.
Step 4	**crypto key generate rsa**	(Optional) Generate an RSA key pair. RSA key pairs are required before you can obtain a certificate for the switch. RSA key pairs are generated automatically. You can use this command to regenerate the keys, if needed.
Step 5	**crypto ca trustpoint** *name*	Specify a local configuration name for the CA trustpoint and enter CA trustpoint configuration mode.
Step 6	**enrollment url** *url*	Specify the URL to which the switch should send certificate requests.
Step 7	**enrollment http-proxy** *host-name* *port-number*	(Optional) Configure the switch to obtain certificates from the CA through an HTTP proxy server.

	Command	Purpose
Step 8	**crl query** *url*	Configure the switch to request a certificate revocation list (CRL) to ensure that the certificate of the peer has not been revoked.
Step 9	**primary**	(Optional) Specify that the trustpoint is the primary (default) trustpoint for CA requests.
Step 10	**exit**	Exit CA trustpoint configuration mode, and return to global configuration mode.
Step 11	**crypto ca authentication** *name*	Authenticate the CA by getting the public key of the CA. Use the same name used in Step 5.
Step 12	**crypto ca enroll** *name*	Obtain the certificate from the specified CA trustpoint. This command requests a signed certificate for each RSA key pair.
Step 13	**end**	Return to privileged EXEC mode.
Step 14	**show crypto ca trustpoints**	Verify the configuration.
Step 15	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no crypto ca trustpoint** *name* global configuration command to delete all identity information and certificates associated with the CA.

EXAMPLE

```
Switch(config)# crypto key generate rsa
Switch(config)# crypto ca trustpoint your_trustpoint
Switch(ca-trustpoint)# enrollment url http://your_server:80
Switch(ca-trustpoint)# enrollment http-proxy your_host 49
Switch(ca-trustpoint)# crl query ldap://your_host:49
Switch(ca-trustpoint)# primary your_trustpoint
Switch(ca-trustpoint)# exit
Switch(config)# crypto ca authentication your_trustpoint
Switch(config)# crypto ca enroll your_trustpoint
Switch(config)# end
```

Configuring the Secure HTTP Server

If you are using a certificate authority for certification, you should use the previous procedure to configure the CA trustpoint on the switch before enabling the HTTP server. If you have not configured a CA trustpoint, a self-signed certificate is generated the first time that you enable the secure HTTP server. After you have configured the server, you can configure options (path, access list to apply, maximum number of connections, or timeout policy) that apply to both standard and secure HTTP servers.

BEFORE YOU BEGIN

Complete the "Configuring a CA Trustpoint" procedure on page 1-54.

Review the Guidelines and Limitations for this feature.

DETAILED STEPS

	Command	Purpose

Step 1	**show ip http server status**	(Optional) Display the status of the HTTP server to determine if the secure HTTP server feature is supported in the software. You should see one of these lines in the output:
		
		`HTTP secure server capability: Present`
		`or`
		`HTTP secure server capability: Not present`

Step 2	**configure terminal**	Enter global configuration mode.
Step 3	**ip http secure-server**	Enable the HTTPS server if it has been disabled. The HTTPS server is enabled by default.
Step 4	**ip http secure-port** *port-number*	(Optional) Specify the port number to be used for the HTTPS server. The default port number is 443. Valid options are 443 or any number in the range 1025 to 65535.
Step 5	**ip http secure-ciphersuite** {[**3des-ede-cbc-sha**] [**rc4-128-md5**] [**rc4-128-sha**] [**des-cbc-sha**]}	(Optional) Specify the CipherSuites (encryption algorithms) to be used for encryption over the HTTPS connection. If you do not have a reason to specify a particular CipherSuite, allow the server and client to negotiate a CipherSuite that they both support. This is the default.
Step 6	**ip http secure-client-auth**	(Optional) Configure the HTTP server to request an X.509v3 certificate from the client for authentication during the connection process. The default is for the client to request a certificate from the server, but the server does not attempt to authenticate the client.
Step 7	**ip http secure-trustpoint** *name*	Specify the CA trustpoint to use to get an X.509v3 security certificate and to authenticate the client certificate connection. **Note** Use of this command assumes you have already configured a CA trustpoint according to the previous procedure.
Step 8	**ip http path** *path-name*	(Optional) Set a base HTTP path for HTML files. The path specifies the location of the HTTP server files on the local system (usually located in system flash memory).
Step 9	**ip http access-class** *access-list-number*	(Optional) Specify an access list to use to allow access to the HTTP server.
Step 10	**ip http max-connections** *value*	(Optional) Set the maximum number of concurrent connections that are allowed to the HTTP server. The range is 1 to 16; the default value is 5.

	Command	Purpose
Step 11	**ip http timeout-policy idle** *seconds* **life** *seconds* **requests** *value*	(Optional) Specify how long a connection to the HTTP server can remain open under the defined circumstances: • **idle**—the maximum time period when no data is received or response data cannot be sent. The range is 1 to 600 seconds. The default is 180 seconds (3 minutes). • **life**—the maximum time period from the time that the connection is established. The range is 1 to 86400 seconds (24 hours). The default is 180 seconds. • **requests**—the maximum number of requests processed on a persistent connection. The maximum value is 86400. The default is 1.
Step 12	**end**	Return to privileged EXEC mode.
Step 13	**show ip http server secure status**	Display the status of the HTTP secure server to verify the configuration.
Step 14	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no ip http server** global configuration command to disable the standard HTTP server. Use the **no ip http secure-server** global configuration command to disable the secure HTTP server. Use the **no ip http secure-port** and the **no ip http secure-ciphersuite** global configuration commands to return to the default settings. Use the **no ip http secure-client-auth** global configuration command to remove the requirement for client authentication.

To verify the secure HTTP connection by using a web browser, enter https://*URL*, where the URL is the IP address or hostname of the server switch. If you configure a port other than the default port, you must also specify the port number after the URL. For example:

https://209.165.129:1026
or

https://host.domain.com:1026

EXAMPLE

```
Switch# show ip http server status
Switch# configure terminal
Switch(config)# ip http secure-server
Switch(config)# ip http secure-port 443
Switch(config)# ip http secure-ciphersuite rc4-128-md5
Switch(config)# ip http secure-client-auth
Switch(config)# ip http secure-trustpoint your_trustpoint
Switch(config)# ip http path /your_server:80
Switch(config)# ip http access-class 2
Switch(config)# ip http timeout-policy idle 120 life 240 requests 1
Switch(config)# end
```

Configuring the Secure HTTP Client

The standard HTTP client and secure HTTP client are always enabled. A certificate authority is required for secure HTTP client certification.

BEFORE YOU BEGIN

This procedure assumes that you have previously configured a CA trustpoint on the switch (see "Configuring a CA Trustpoint" procedure on page 1-54). If a CA trustpoint is not configured and the remote HTTPS server requires client authentication, connections to the secure HTTP client fail.

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**ip http client secure-trustpoint** *name*	(Optional) Specify the CA trustpoint to be used if the remote HTTP server requests client authentication. Using this command assumes that you have already configured a CA trustpoint as described in the "Configuring a CA Trustpoint" procedure on page 1-54. The command is optional if client authentication is not needed or if a primary trustpoint has been configured.
Step 3	**ip http client secure-ciphersuite** {[**3des-ede-cbc-sha**] [**rc4-128-md5**] [**rc4-128-sha**] [**des-cbc-sha**]}	(Optional) Specify the CipherSuites (encryption algorithms) to be used for encryption over the HTTPS connection. If you do not have a reason to specify a particular CipherSuite, you should allow the server and client to negotiate a CipherSuite that they both support. This is the default.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show ip http client secure status**	Display the status of the HTTP secure server to verify the configuration.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

Use the **no ip http client secure-trustpoint** *name* to remove a client trustpoint configuration. Use the **no ip http client secure-ciphersuite** to remove a previously configured CipherSuite specification for the client.

EXAMPLE

```
Switch (config)# ip http client secure-trustpoint your_trustpoint
Switch(config)# ip http client secure-ciphersuite rc4-128-md5
Switch(config)# end
```

Verifying Configuration

Command	Purpose
show tacacs	Display TACACS+ server statistics.
show running-config	Display RADIUS configuration.
show ip ssh	Shows the version and configuration information for the SSH server.

Command	Purpose
show ssh	Shows the status of the SSH server.
show ip http client secure status	Shows the HTTP secure client configuration.
show ip http server secure status	Shows the HTTP secure server configuration.
show running-config	Shows the generated self-signed certificate for secure HTTP connections.

Related Documents

- Cisco IOS Master Command List, All Releases

- User Security Configuration Guide, Cisco IOS Release 15M&T

- Cisco IOS Security Command Reference

- RADIUS Attributes Configuration Guide, Cisco IOS Release 15M&T

- RADIUS Configuration Guide, Cisco IOS Release 15M&T

- Secure Shell Configuration Guide, Cisco IOS Release 15M&T

- Public Key Infrastructure Configuration Guide, Cisco IOS Release 15MT