



Switch Access: Using Authentication, Authorization, and Accounting

This chapter describes how to configure authentication, authorization, and accounting (AAA) to monitor and control access to the command-line interface (CLI) on the Catalyst enterprise LAN switches.



Note

For complete syntax and usage information for the commands used in this chapter, refer to the *Command Reference* publication for your switch.

This chapter consists of these sections:

- Understanding How Authentication Works, page 30-1
- Configuring Authentication, page 30-7
- Authentication Example, page 30-34
- Understanding How Authorization Works, page 30-35
- Configuring Authorization, page 30-37
- Authorization Example, page 30-41
- Understanding How Accounting Works, page 30-41
- Configuring Accounting, page 30-44
- Accounting Example, page 30-48

Understanding How Authentication Works

These sections describe how the different authentication methods work:

- Authentication Overview, page 30-2
- Understanding How Local Authentication Works, page 30-2
- Understanding How TACACS+ Authentication Works, page 30-2
- Understanding How RADIUS Authentication Works, page 30-3
- Understanding How Kerberos Authentication Works, page 30-4

Authentication Overview

You can configure any combination of these authentication methods to control access to the switch:

- Local authentication
- TACACS+ authentication
- RADIUS authentication
- Kerberos authentication

**Note**

Kerberos authentication does not work if TACACS+ is used as the authentication method.

When multiple authentication methods are enabled, if local authentication is enabled, it is always attempted last. In supervisor engine software release 4.4 and later, you can specify the authentication method to use for console and Telnet connections independently. For example, you might use local authentication for console connections and RADIUS authentication for Telnet connections.

Understanding How Local Authentication Works

Local authentication uses locally configured login and enable passwords to authenticate login attempts. The login and enable passwords are local to each switch and are not mapped to individual user names.

Local authentication is enabled by default, but can be disabled if one of the other authentication methods is enabled. If local authentication is disabled and you then disable all other authentication methods, local authentication is reenabled automatically.

You can enable local authentication and one or more of the other authentication methods at the same time. Local authentication is only attempted if the other authentication methods fail.

Understanding How TACACS+ Authentication Works

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

TACACS+ works with many authentication types, including fixed password, one-time password, and challenge-response authentication. TACACS+ authentication usually occurs in these instances:

- When you first log onto a machine
- When you send a service request that requires privileged access

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

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

When the TACACS+ server receives the packet, it does the following:

- Authenticates the user information and notifies the client that authentication has either passed or failed.
- Notifies the client that authentication will continue and that the client must provide additional information. This challenge-response process can continue through multiple iterations until authentication either passes or fails.

You can configure a TACACS+ key on the client and server. If you configure a key on the switch, it must be the same as the one configured on the TACACS+ servers. The TACACS+ clients and servers use the key to encrypt all TACACS+ packets transmitted. If you do not configure a TACACS+ key, packets are not encrypted.

You can configure the following TACACS+ parameters on the switch:

- Enable or disable TACACS+ authentication to determine if a user has permission to access the switch
- Enable or disable TACACS+ authentication to determine if a user has permission to enter privileged mode
- Specify a key used to encrypt the protocol packets
- Specify the server on which the TACACS+ server daemon resides
- Set the number of login attempts allowed
- Set the timeout interval for server daemon response
- Enable or disable the directed-request option

TACACS+ authentication is disabled by default. You can enable TACACS+ authentication and local authentication at the same time.

If local authentication is disabled and you then disable all other authentication methods, local authentication is reenabled automatically.

Understanding How RADIUS Authentication Works

RADIUS is a client-server authentication and authorization access protocol used by the NAS to authenticate users attempting to connect to a network device. The NAS functions as a client, passing user information to one or more RADIUS servers. The NAS permits or denies network access to a user based on the response it receives from one or more RADIUS servers. RADIUS uses UDP for transport between the RADIUS client and server.

You can configure a RADIUS key on the client and server. If you configure a key on the client, it must be the same as the one configured on the RADIUS servers. The RADIUS clients and servers use the key to encrypt all RADIUS packets transmitted. If you do not configure a RADIUS key, packets are not encrypted. The key itself is never transmitted over the network.



Note

For more information about how the RADIUS protocol operates, see RFC 2138, “Remote Authentication Dial In User Service (RADIUS).”

You can configure the following RADIUS parameters on the switch:

- Enable or disable RADIUS authentication to control login access
- Enable or disable RADIUS authentication to control enable access
- Specify the IP addresses and UDP ports of the RADIUS servers

- Specify the RADIUS key used to encrypt RADIUS packets
- Specify the RADIUS server timeout interval
- Specify the RADIUS retransmit count
- Specify the RADIUS server deadtime interval

RADIUS authentication is disabled by default. You can enable RADIUS authentication and other authentication methods at the same time. You can specify which method to use first using the **primary** keyword.

If local authentication is disabled and you then disable all other authentication methods, local authentication is reenabled automatically.

Understanding How Kerberos Authentication Works

Kerberos is a client-server based secret-key network authentication method that uses a trusted Kerberos server to verify secure access to both services and users. In Kerberos, this trusted server is called the key distribution center (KDC). The KDC issues tickets to validate users and services. A ticket is a temporary set of electronic credentials that verify the identity of a client for a particular service.

These tickets have a limited life span and can be used in place of the standard user password authentication mechanism if a service trusts the Kerberos server from which the ticket was issued. If the standard user password method is used, Kerberos encrypts user passwords into the tickets, ensuring that passwords are not sent on the network in clear text. When you use Kerberos, passwords are not stored on any machine, except for the Kerberos server, for more than a few seconds. Kerberos also guards against intruders who might pick up the encrypted tickets from the network.

Table 30-1 defines the terms used in Kerberos.

Table 30-1 Kerberos Terminology

Term	Definition
Kerberos Credential	General term referring to authentication tickets, such as ticket granting tickets (TGTs) and service credentials. Kerberos credentials verify the ticket of a user or service. If a network service decides to trust the Kerberos server that issued the ticket, it can be used in place of retyping in a username and password. Credentials have a default life span of eight hours.
Kerberos Principal	Also known as a Kerberos identity. The Kerberos principal is who you are or what a service is according to the Kerberos server.
Kerberos Realm	A domain consisting of users, hosts, and network services that are registered to a Kerberos server. The Kerberos server is trusted to verify the identity of a user or network service to another user or network service. Kerberos realms must always be in uppercase characters.
Kerberos Server	A daemon running on a network host. Users and network services register their identity with the Kerberos server. Network services query the Kerberos server to authenticate to other network services.
Key Distribution Center (KDC)	A Kerberos server and database program running on a network host that allocates the Kerberos Credentials to different users or network services.

Table 30-1 Kerberos Terminology (continued)

Term	Definition
Ticket Granting Ticket (TGT)	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.
Service Credential	A credential for a network service. When issued from the KDC, this credential is encrypted with the password shared by the network service and the KDC and with the user's TGT.
SRVTAB	A password that a network service shares with the KDC. The network service authenticates an encrypted service credential by using the SRVTAB (also known as a KEYTAB) to decrypt it.
Kerberized	Applications and services that have been modified to support the Kerberos credential infrastructure.

In the Catalyst 5000 family switches, Telnet clients and servers through both the console and in-band management port can be Kerberized.

**Note**

Kerberos authentication does not work if TACACS+ is used as the authentication mechanism.

**Note**

If you are logged in to the console through a modem or a terminal server, a Kerberized login procedure cannot be used.

Using Kerberized Login Procedure

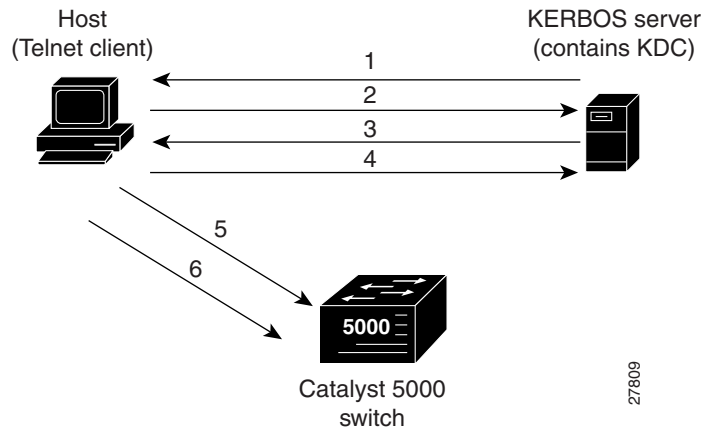
You can use a Kerberized Telnet session if you are logging in through the in-band management port. After the Telnet client and services have been Kerberized, the following process takes place when a user attempts to Telnet to the switch:

1. The Telnet client asks the user for the username and issues a request for a TGT to the KDC on the Kerberos server.
2. The KDC creates the TGT, which contains the user's identity, the KDC's identity, and the TGT's expiration time. The KDC then encrypts the TGT with the user's password and sends the TGT to the client.
3. When the Telnet client receives the encrypted TGT, it prompts the user for the password. If the Telnet client can decrypt the TGT with the entered password, the user is successfully authenticated to the KDC. The client then builds a service credential request and sends this to the KDC. This request contains the user's identity and a message saying that it wants to Telnet to the switch. This request is encrypted using the TGT.
4. When the KDC successfully decrypts the service credential request with the TGT that it issued to the client, it builds a service to the switch. The service credential has the client's identity and the identity of the desired Telnet server. The KDC then encrypts the credential with the password that it shares with the switch's Telnet server and encrypts the resulting packet with the Telnet client's TGT and sends this packet to the client.

5. The Telnet client decrypts the packet first with its TGT. If encryption is successful, the client then sends the resulting packet to the switch's Telnet server. At this point, the packet is still encrypted with the password that the switch's Telnet server and the KDC share.
6. If the Telnet client has been instructed to do so, it forwards the TGT to the switch. This ensures that the user does not need to get another TGT in order to use another network service from the switch.

Figure 30-1 illustrates the Kerberos Telnet connection process.

Figure 30-1 Kerberized Telnet Connection



Using a Non-Kerberized Login Procedure

If a non-Kerberized login procedure is used to log in to the switch, the switch takes care of authentication to the KDC on behalf of the login client. However, the user password is now transferred in clear text from the login client to the switch.



Note

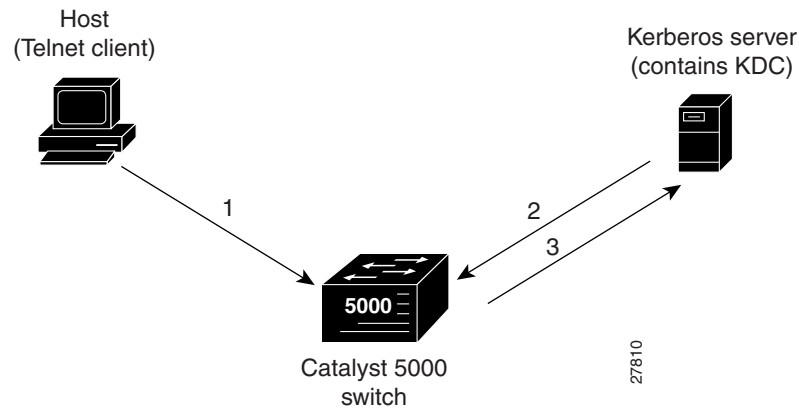
A non-Kerberized login can be performed through a modem or terminal server through the in-band management port. Telnet does not support non-Kerberized login.

If a non-Kerberized login is launched, the following process takes place:

1. The switch prompts you for a username and password.
2. The switch requests a TGT from the KDC so that you can be authenticated to the switch.
3. The KDC sends an encrypted TGT to the switch, which contains your identity, KDC's identity, and TGT's expiration time.
4. The switch tries to decrypt the TGT with the password that you entered. If the decryption is successful, you are authenticated to the switch.
5. If you want to access other network services, the KDC must be contacted directly for authentication. To obtain the TGT, you can run the program "kinit," the client software provided with the Kerberos package.

Figure 30-2 illustrates the non-Kerberized login process.

Figure 30-2 Non-Kerberized Telnet Connection



Configuring Authentication

These sections describe how to configure the different authentication methods:

- Authentication Default Configuration, page 30-7
- Authentication Configuration Guidelines, page 30-8
- Configuring Local Authentication, page 30-9
- Configuring TACACS+ Authentication, page 30-12
- Configuring RADIUS Authentication, page 30-18
- Configuring Kerberos Authentication, page 30-25

Authentication Default Configuration

Table 30-2 shows the default authentication configuration.

Table 30-2 Authentication Default Configuration

Feature	Default Value
Local login authentication (console and Telnet)	Enabled
Local enable authentication (console and Telnet)	Enabled
Kerberos login authentication (console and Telnet)	Disabled
Kerberos enable authentication (console and Telnet)	Disabled
Kerberos server IP address	None specified
Kerberos DES key	None specified

Table 30-2 Authentication Default Configuration (continued)

Feature	Default Value
Kerberos server auth-port	Port 750
Kerberos local-realm name	NULL string
Kerberos credentials forwarding	Disabled
Kerberos clients mandatory	Not mandatory
Kerberos preauthentication	Disabled
RADIUS login authentication (console and Telnet)	Disabled
RADIUS enable authentication (console and Telnet)	Disabled
RADIUS server IP address	None specified
RADIUS server UDP auth-port	Port 1812
RADIUS key	None specified
RADIUS server timeout	5 seconds
RADIUS server deadtime	0 (servers not marked dead)
RADIUS retransmit attempts	2 times
TACACS+ login authentication (console and Telnet)	Disabled
TACACS+ enable authentication (console and Telnet)	Disabled
TACACS+ key	None specified
TACACS+ login attempts	3
TACACS+ server timeout	5 seconds
TACACS+ directed request	Disabled

Authentication Configuration Guidelines

These guidelines apply when configuring authentication on the switch:

- Authentication configuration applies both to console and Telnet connection attempts unless you use the **console** and **telnet** keywords to specify the authentication methods to use for each connection type individually.
- If you configure a RADIUS or TACACS+ key on the switch, make sure you configure an identical key on the RADIUS or TACACS+ server.
- You must specify a RADIUS or TACACS+ server before enabling RADIUS or TACACS+ on the switch.
- If you configure multiple RADIUS or TACACS+ servers, the first server configured is the primary and authentication requests are sent to this server first. You can specify a particular server as primary by using the **primary** keyword.
- RADIUS and TACACS+ supports one privileged mode only (level 1).
- Kerberos authentication does not work if TACACS+ is also used as an authentication mechanism.

Configuring Local Authentication

These sections describe how to configure local authentication on the switch:

- Enabling Local Authentication, page 30-9
- Setting the Login Password, page 30-10
- Setting the Enable Password, page 30-10
- Disabling Local Authentication, page 30-11
- Recovering a Lost Password, page 30-11

Enabling Local Authentication



Note

Local login and enable authentication are enabled for both console and Telnet connections by default. You do not need to perform this task unless you want to modify the default configuration or you have disabled local authentication.

To enable local authentication on the switch, perform this task in privileged mode:

	Task	Command
Step 1	Enable local login authentication. Use the console or telnet keywords to enable local authentication only for console or Telnet connection attempts.	set authentication login local enable [all console http telnet]
Step 2	Enable local enable authentication. Use the console or telnet keywords to enable local authentication only for console or Telnet connection attempts.	set authentication enable local enable [all console http telnet]
Step 3	Verify the local authentication configuration.	show authentication

This example shows how to enable local login and enable authentication for both console and Telnet connections and how to verify the configuration:

```
Console> (enable) set authentication login local enable
local login authentication set to enable for console and telnet session.
Console> (enable) set authentication enable local enable
local enable authentication set to enable for console and telnet session.
Console> (enable) show authentication
```

```
Login Authentication: Console Session Telnet Session
-----
tacacs                disabled          disabled
radius                disabled          disabled
kerberos              disabled          disabled
local                 enabled(primary) enabled(primary)
```

```
Enable Authentication: Console Session Telnet Session
-----
tacacs                disabled          disabled
radius                disabled          disabled
kerberos              disabled          disabled
local                 enabled(primary) enabled(primary)
Console> (enable)
```

Setting the Login Password

The login password controls access to the user mode CLI. Passwords are case sensitive, can contain up to 30 characters, and use any printable character, including a space.


Note

Passwords set in releases prior to software release 5.4 remain non-case sensitive. You must reset the password after installing release 5.4 to activate case sensitivity.

To set the login password for local authentication, perform this task in privileged mode:

Task	Command
Set the login password for access. Enter your old password (press Return on a switch with no password configured), enter your new password, and reenter your new password.	set password

This example shows how to set the login password on the switch:

```
Console> (enable) set password
Enter old password:old_password
Enter new password:new_password
Retype new password:new_password
Password changed.
Console> (enable)
```

Setting the Enable Password

The enable password controls access to the privileged mode CLI. Passwords are case sensitive, can contain up to 30 characters, and can use any printable character, including a space.


Note

Passwords set in releases prior to software release 5.4 remain non-case sensitive. You must reset the password after installing Version 5.4 to activate case sensitivity.

To set the enable password for local authentication, perform this task in privileged mode:

Task	Command
Set the password for privileged mode. Enter your old password (press Return on a switch with no password configured), enter your new password, and reenter your new password.	set enablepass

This example shows how to set the enable password on the switch:

```
Console> (enable) set enablepass
Enter old password:<old_password>
Enter new password:<new_password>
Retype new password:<new_password>
Password changed.
Console> (enable)
```

Disabling Local Authentication



Caution

Make sure that RADIUS or TACACS+ authentication is configured and operating correctly before disabling local login or enable authentication. If you disable local authentication and RADIUS or TACACS+ is not configured correctly, or if the RADIUS or TACACS+ server is not online, you may be unable to log in to the switch.

To disable local authentication on the switch, perform this task in privileged mode:

	Task	Command
Step 1	Disable local login authentication. Use the console or telnet keywords to disable local authentication only for console or Telnet connection attempts.	set authentication login local disable [all console http telnet]
Step 2	Disable local enable authentication. Use the console or telnet keywords to disable local authentication only for console or Telnet connection attempts.	set authentication enable local disable [all console http telnet]
Step 3	Verify the local authentication configuration.	show authentication

This example shows how to disable local login and enable authentication for both console and Telnet connections, and how to verify the configuration (you must have RADIUS or TACACS+ authentication enabled before you disable local authentication):

```
Console> (enable) set authentication login local disable
local login authentication set to disable for console and telnet session.
Console> (enable) set authentication enable local disable
local enable authentication set to disable for console and telnet session.
Console> (enable) show authentication
```

```
Login Authentication: Console Session Telnet Session
-----
tacacs                disabled          disabled
radius                enabled(primary) enabled(primary)
kerberos              disabled          disabled
local                 disabled          disabled
```

```
Enable Authentication: Console Session Telnet Session
-----
tacacs                disabled          disabled
radius                enabled(primary) enabled(primary)
kerberos              disabled          disabled
local                 disabled          disabled
Console> (enable)
```

Recovering a Lost Password

To recover a lost local authentication password, perform this procedure. You must complete Steps 3 through 7 within 30 seconds of a power cycle or the recovery will fail. If you lost both the login and enable passwords, repeat the process for each password.

-
- Step 1** Connect to the switch through the supervisor engine console port (you cannot recover the password if you are connected through a Telnet connection).
- Step 2** Enter the **reset system** command to reboot the switch.
- Step 3** At the “Enter Password” prompt, press **Return** (the login password is null for 30 seconds when you are connected to the console port).
- Step 4** Enter privileged mode using the **enable** command.
- Step 5** At the “Enter Password” prompt, press **Return** (the enable password is null for 30 seconds when you are connected to the console port).
- Step 6** Enter the **set password** or **set enablepass** command, as appropriate.
- Step 7** When prompted for your old password, press **Return**.
- Step 8** Enter and confirm your new password.
-

Configuring TACACS+ Authentication

These sections describe how to configure TACACS+ authentication on the switch.

- Specifying TACACS+ Servers, page 30-12
- Enabling TACACS+ Authentication, page 30-13
- Specifying the TACACS+ Key, page 30-14
- Setting the TACACS+ Timeout Interval, page 30-15
- Setting the TACACS+ Login Attempts, page 30-15
- Enabling TACACS+ Directed Request, page 30-16
- Disabling TACACS+ Directed Request, page 30-16
- Clearing TACACS+ Servers, page 30-17
- Clearing the TACACS+ Key, page 30-17
- Disabling TACACS+ Authentication, page 30-17

Specifying TACACS+ Servers

Specify one or more TACACS+ servers before you enable TACACS+ authentication on the switch. The first server you specify is the primary server, unless you explicitly make one server the primary server by using the **primary** keyword.

To specify one or more TACACS+ servers, perform this task in privileged mode:

	Task	Command
Step 1	Specify the IP address of one or more TACACS+ servers.	set tacacs server <i>ip_addr</i> [primary]
Step 2	Verify the TACACS+ configuration.	show tacacs

This example shows how to specify TACACS+ servers and verify the configuration:

```

Console> (enable) set tacacs server 172.20.52.3
172.20.52.3 added to TACACS server table as primary server.
Console> (enable) set tacacs server 172.20.52.2 primary
172.20.52.2 added to TACACS server table as primary server.
Console> (enable) set tacacs server 172.20.52.10
172.20.52.10 added to TACACS server table as backup server.
Console> (enable) show tacacs

Login Authentication: Console Session Telnet Session
-----
tacacs                disabled                disabled
radius                disabled                disabled
local                 enabled(primary)       enabled(primary)

Enable Authentication: Console Session Telnet Session
-----
tacacs                disabled                disabled
radius                disabled                disabled
local                 enabled(primary)       enabled(primary)

Tacacs key:
Tacacs login attempts: 3
Tacacs timeout: 5 seconds
Tacacs direct request: disabled

Tacacs-Server                Status
-----
172.20.52.3
172.20.52.2                primary
172.20.52.10
Console> (enable)

```

Enabling TACACS+ Authentication



Note

Specify at least one TACACS+ server before enabling TACACS+ authentication on the switch. For information on specifying a TACACS+ server, see the “Specifying TACACS+ Servers” section on page 30-12.

You can enable TACACS+ authentication for login and enable access to the switch. If desired, you can use the **console** and **telnet** keywords to specify that TACACS+ authentication be used only on console or Telnet connections. If you are using both RADIUS and TACACS+, you can use the **primary** keyword to force the switch to try TACACS+ authentication first.

To enable TACACS+ authentication, perform this task in privileged mode:

	Task	Command
Step 1	Enable TACACS+ authentication for normal login mode. Use the console or telnet keywords if you want to enable TACACS+ only for console port or Telnet connection attempts.	set authentication login tacacs enable [all console http telnet] [primary]
Step 2	Enable TACACS+ authentication for enable mode. Use the console or telnet keywords if you want to enable TACACS+ only for console port or Telnet connection attempts.	set authentication enable tacacs enable [all console http telnet] [primary]
Step 3	Verify the TACACS+ configuration.	show authentication

This example shows how to enable TACACS+ authentication for console and Telnet connections and how to verify the configuration:

```
Console> (enable) set authentication login tacacs enable
tacacs login authentication set to enable for console and telnet session.
Console> (enable) set authentication enable tacacs enable
tacacs enable authentication set to enable for console and telnet session.
Console> (enable) show authentication
```

```
Login Authentication:  Console Session  Telnet Session
-----
tacacs                enabled(primary)  enabled(primary)
radius                disabled           disabled
local                 enabled            enabled
Enable Authentication: Console Session  Telnet Session
-----
tacacs                enabled(primary)  enabled(primary)
radius                disabled           disabled
local                 enabled            enabled
Console> (enable)
```

Specifying the TACACS+ Key



Note

If you configure a TACACS+ key on the client, make sure you configure an identical key on the TACACS+ server.

To specify the TACACS+ key, perform this task in privileged mode:

	Task	Command
Step 1	Configure the key used to encrypt packets.	set tacacs key <i>key</i>
Step 2	Verify the TACACS+ configuration.	show tacacs

This example shows how to specify the TACACS+ key and verify the configuration:

```

Console> (enable) set tacacs key Secret_TACACS_key
The tacacs key has been set to Secret_TACACS_key.
Console> (enable) show tacacs
Tacacs key: Secret_TACACS_key
Tacacs login attempts: 3
Tacacs timeout: 5 seconds
Tacacs direct request: disabled

Tacacs-Server                               Status
-----
172.20.52.3
172.20.52.2                                 primary
172.20.52.10
Console> (enable)

```

Setting the TACACS+ Timeout Interval

You can specify the timeout interval between retransmissions to the TACACS+ server. The default timeout is 5 seconds.

To specify the TACACS+ timeout interval, perform this task in privileged mode:

	Task	Command
Step 1	Configure the TACACS+ timeout interval.	set tacacs timeout <i>seconds</i>
Step 2	Verify the TACACS+ configuration.	show tacacs

This example shows how to set the server timeout interval and verify the configuration:

```

Console> (enable) set tacacs timeout 30
Tacacs timeout set to 30 seconds.
Console> (enable) show tacacs
Tacacs key: Secret_TACACS_key
Tacacs login attempts: 3
Tacacs timeout: 30 seconds
Tacacs direct request: disabled

Tacacs-Server                               Status
-----
172.20.52.3
172.20.52.2                                 primary
172.20.52.10
Console> (enable)

```

Setting the TACACS+ Login Attempts

You can specify the number of failed login attempts allowed.

To specify the number of login attempts allowed, perform this task in privileged mode:

	Task	Command
Step 1	Configure the number of allowed login attempts.	set tacacs attempts <i>number</i>
Step 2	Verify the TACACS+ configuration.	show tacacs

This example shows how to set the number of login attempts and verify the configuration:

```

Console> (enable) set tacacs attempts 5
Tacacs number of attempts set to 5.
Console> (enable) show tacacs
Tacacs key: Secret_TACACS_key
Tacacs login attempts: 5
Tacacs timeout: 30 seconds
Tacacs direct request: disabled

Tacacs-Server                               Status
-----
172.20.52.3
172.20.52.2                                primary
172.20.52.10
Console> (enable)

```

Enabling TACACS+ Directed Request

When TACACS+ directed request is enabled, users must specify the host name of a configured TACACS+ server (in the form *username@server_hostname*) or the authentication request will fail.

To enable TACACS+ directed request, perform this task in privileged mode:

	Task	Command
Step 1	Enable TACACS+ directed request on the switch.	set tacacs directedrequest enable
Step 2	Verify the TACACS+ configuration.	show tacacs

This example shows how to enable TACACS+ directed request and verify the configuration:

```

Console> (enable) set tacacs directedrequest enable
Tacacs direct request has been enabled.

Console> (enable) show tacacs
Tacacs key: Secret_TACACS_key
Tacacs login attempts: 5
Tacacs timeout: 30 seconds
Tacacs direct request: enabled

Tacacs-Server                               Status
-----
172.20.52.3
172.20.52.2                                primary
172.20.52.10
Console> (enable)

```

Disabling TACACS+ Directed Request

To disable TACACS+ directed request, perform this task in privileged mode:

	Task	Command
Step 1	Disable TACACS+ directed request on the switch.	set tacacs directedrequest disable
Step 2	Verify the TACACS+ configuration.	show tacacs

This example shows how to disable TACACS+ directed request:

```
Console> (enable) set tacacs directedrequest disable
Tacacs direct request has been disabled.
Console> (enable)
```

Clearing TACACS+ Servers

To clear one or more TACACS+ servers, perform this task in privileged mode:

	Task	Command
Step 1	Specify the IP address of the TACACS+ server to clear from the configuration. Use the all keyword to clear all of the servers from the configuration.	clear tacacs server [<i>ip_addr</i> all]
Step 2	Verify the TACACS+ server configuration.	show tacacs

This example shows how to clear a specific TACACS+ server from the configuration:

```
Console> (enable) clear tacacs server 172.20.52.3
172.20.52.3 cleared from TACACS table
Console> (enable)
```

This example shows how to clear all TACACS+ servers from the configuration:

```
Console> (enable) clear tacacs server all
All TACACS servers cleared
Console> (enable)
```

Clearing the TACACS+ Key

To clear the TACACS+ key, perform this task in privileged mode:

	Task	Command
Step 1	Clear the TACACS+ key.	clear tacacs key
Step 2	Verify the TACACS+ configuration.	show tacacs

This example shows how to clear the TACACS+ key:

```
Console> (enable) clear tacacs key
TACACS server key cleared.
Console> (enable)
```

Disabling TACACS+ Authentication

If you disable TACACS+ authentication with both RADIUS and local authentication disabled, local authentication is reenabled automatically.

To disable TACACS+ authentication, perform this task in privileged mode:

	Task	Command
Step 1	Disable TACACS+ authentication for normal login mode. Use the console or telnet keywords if you want to disable TACACS+ only for console port or Telnet connection attempts.	set authentication login tacacs disable [all console http telnet]
Step 2	Disable TACACS+ authentication for enable mode. Use the console or telnet keywords if you want to disable TACACS+ only for console port or Telnet connection attempts.	set authentication enable tacacs disable [all console http telnet]
Step 3	Verify the TACACS+ configuration.	show authentication

This example shows how to disable TACACS+ authentication for console and Telnet connections and how to verify the configuration:

```
Console> (enable) set authentication login tacacs disable
tacacs login authentication set to disable for console and telnet session.
Console> (enable) set authentication enable tacacs disable
tacacs enable authentication set to disable for console and telnet session.
Console> (enable) show authentication
```

```
Login Authentication: Console Session Telnet Session
-----
tacacs                disabled          disabled
radius                disabled          disabled
local                 enabled(primary) enabled(primary)

Enable Authentication: Console Session Telnet Session
-----
tacacs                disabled          disabled
radius                disabled          disabled
local                 enabled(primary) enabled(primary)
Console> (enable)
```

Configuring RADIUS Authentication

These sections describe how to configure RADIUS authentication on the switch.

- Specifying RADIUS Servers, page 30-19
- Enabling RADIUS Authentication, page 30-19
- Specifying the RADIUS Key, page 30-20
- Setting the RADIUS Timeout Interval, page 30-21
- Setting the RADIUS Retransmit Count, page 30-22
- Setting the RADIUS Deadtime, page 30-22
- Clearing RADIUS Servers, page 30-23
- Clearing the RADIUS Key, page 30-24
- Disabling RADIUS Authentication, page 30-24

Specifying RADIUS Servers

To specify one or more RADIUS servers, perform this task in privileged mode:

	Task	Command
Step 1	Specify the IP address of up to three RADIUS servers. Specify the primary server using the primary keyword. Optionally, specify the destination UDP port to use on the server.	set radius server <i>ip_addr</i> [auth-port <i>port_number</i>] [primary]
Step 2	Verify the RADIUS server configuration.	show radius

This example shows how to specify a RADIUS server and verify the configuration:

```
Console> (enable) set radius server 172.20.52.3
172.20.52.3 with auth-port 1812 added to radius server table as primary server.
Console> (enable) show radius
```

```
Login Authentication: Console Session Telnet Session
-----
tacacs                disabled          disabled
radius                disabled          disabled
local                 enabled(primary) enabled(primary)

Enable Authentication: Console Session Telnet Session
-----
tacacs                disabled          disabled
radius                disabled          disabled
local                 enabled(primary) enabled(primary)

Radius Deadtime:      0 minutes
Radius Key:
Radius Retransmit:   2
Radius Timeout:      5 seconds

Radius-Server        Status Auth-port
-----
172.20.52.3          primary 1812
Console> (enable)
```

Enabling RADIUS Authentication



Note

Specify at least one RADIUS server before enabling RADIUS authentication on the switch. For information on specifying a RADIUS server, see the “Specifying RADIUS Servers” section on page 30-19.

You can enable RADIUS authentication for login and enable access to the switch. If desired, you can use the **console** and **telnet** keywords to specify that RADIUS authentication be used only on console or Telnet connections. If you are using both RADIUS and TACACS+, you can use the **primary** keyword to force the switch to try RADIUS authentication first.

To configure RADIUS authentication, perform this task in privileged mode:

	Task	Command
Step 1	Enable RADIUS authentication for normal login mode.	set authentication login radius enable [all console http telnet] [primary]
Step 2	Enable RADIUS authentication for enable mode.	set authentication enable radius enable [all console http telnet] [primary]
Step 3	Verify the RADIUS configuration.	show authentication

This example shows how to enable RADIUS authentication and verify the configuration:

```
Console> (enable) set authentication login radius enable
radius login authentication set to enable for console and telnet session.
Console> (enable) set authentication enable radius enable
radius enable authentication set to enable for console and telnet session.
Console> (enable) show authentication
```

```
Login Authentication: Console Session Telnet Session
-----
tacacs                disabled          disabled
radius                enabled(primary) enabled(primary)
local                 enabled           enabled
```

```
Enable Authentication: Console Session Telnet Session
-----
tacacs                disabled          disabled
radius                enabled(primary) enabled(primary)
local                 enabled           enabled
Console> (enable)
```

Specifying the RADIUS Key



Note

If you configure a RADIUS key on the client, make sure you configure an identical key on the RADIUS server.

The RADIUS key is used to encrypt and authenticate all communication between the RADIUS client and server. You must configure the same key on the client and the RADIUS server.

The length of the key is limited to 65 characters. It can include any printable ASCII characters except tabs.

To specify the RADIUS key, perform this task in privileged mode:

	Task	Command
Step 1	Configure the RADIUS key used to encrypt packets sent to the RADIUS server.	set radius key <i>key</i>
Step 2	Verify the RADIUS configuration.	show radius

This example shows how to specify the RADIUS key and verify the configuration (in normal mode, the RADIUS key value is hidden):

```

Console> (enable) set radius key Secret_RADIUS_key
Radius key set to Secret_RADIUS_key

Console> (enable) show radius

Login Authentication: Console Session  Telnet Session
-----
tacacs                disabled          disabled
radius                enabled(primary) enabled(primary)
local                 enabled           enabled

Enable Authentication: Console Session  Telnet Session
-----
tacacs                disabled          disabled
radius                enabled(primary) enabled(primary)
local                 enabled           enabled

Radius Deadtime:      0 minutes
Radius Key:           Secret_RADIUS_key
Radius Retransmit:   2
Radius Timeout:      5 seconds

Radius-Server        Status  Auth-port
-----
172.20.52.3          primary 1812
Console> (enable)

```

Setting the RADIUS Timeout Interval

You can specify the timeout interval between retransmissions to the RADIUS server. The default timeout is 5 seconds.

To specify the RADIUS timeout interval, perform this task in privileged mode:

	Task	Command
Step 1	Configure the RADIUS timeout interval.	set radius timeout <i>seconds</i>
Step 2	Verify the RADIUS configuration.	show radius

This example shows how to specify the RADIUS timeout interval and verify the configuration:

```

Console> (enable) set radius timeout 10
Radius timeout set to 10 seconds.
Console> (enable) show radius

Login Authentication: Console Session  Telnet Session
-----
tacacs                disabled          disabled
radius                enabled(primary) enabled(primary)
local                 enabled           enabled

Enable Authentication: Console Session  Telnet Session
-----
tacacs                disabled          disabled
radius                enabled(primary) enabled(primary)
local                 enabled           enabled
Radius Deadtime:      0 minutes

```

```

Radius Key:                Secret_RADIUS_key
Radius Retransmit:        2
Radius Timeout:           10 seconds

Radius-Server              Status   Auth-port
-----
172.20.52.3                primary  1812
Console> (enable)

```

Setting the RADIUS Retransmit Count

You can specify the number of times the switch will attempt to contact a RADIUS server before the next configured server is tried. By default, each RADIUS server will be tried two times.

To specify the RADIUS retransmit count, perform this task in privileged mode:

	Task	Command
Step 1	Configure the RADIUS server retransmit count.	set radius retransmit <i>count</i>
Step 2	Verify the RADIUS configuration.	show radius

This example shows how to specify the RADIUS retransmit count and verify the configuration:

```

Console> (enable) set radius retransmit 4
Radius retransmit count set to 4.
Console> (enable) show radius

Login Authentication: Console Session   Telnet Session
-----
tacacs                disabled                disabled
radius                enabled(primary)        enabled(primary)
local                 enabled                  enabled

Enable Authentication: Console Session   Telnet Session
-----
tacacs                disabled                disabled
radius                enabled(primary)        enabled(primary)
local                 enabled                  enabled

Radius Deadtime:      0 minutes
Radius Key:           Secret_RADIUS_key
Radius Retransmit:    4
Radius Timeout:       10 seconds

Radius-Server         Status   Auth-port
-----
172.20.52.3           primary  1812
Console> (enable)

```

Setting the RADIUS Deadtime

You can configure the switch so that when a RADIUS server does not respond to an authentication request, the switch marks that server as dead for the length of time specified by the deadtime. Any authentication requests received during the deadtime interval (such as other users attempting to log in to the switch) are not sent to a RADIUS server marked dead. Configuring a deadtime speeds up the authentication process by eliminating timeouts and retransmissions to the dead RADIUS server.

If you configure only one RADIUS server, or if all of the configured servers are marked dead, the deadtime is ignored because there are no alternate servers available.

To set the RADIUS dead time, perform this task in privileged mode:

	Task	Command
Step 1	Configure the RADIUS server deadtime interval.	set radius deadtime <i>minutes</i>
Step 2	Verify the RADIUS configuration.	show radius

This example shows how to set the RADIUS deadtime interval and verify the configuration:

```

Console> (enable) set radius deadtime 5
Radius deadtime set to 5 minute(s).
Console> (enable) show radius

Login Authentication: Console Session Telnet Session
-----
tacacs                disabled          disabled
radius               enabled(primary) enabled(primary)
local                enabled           enabled

Enable Authentication: Console Session Telnet Session
-----
tacacs                disabled          disabled
radius               enabled(primary) enabled(primary)
local                enabled           enabled

Radius Deadtime:      5 minutes
Radius Key:           Secret_RADIUS_key
Radius Retransmit:    4
Radius Timeout:       10 seconds

Radius-Server
-----
172.20.52.3          primary 1812
172.20.52.2          1812
Console> (enable)

```

Clearing RADIUS Servers

To clear one or more RADIUS servers, perform this task in privileged mode:

	Task	Command
Step 1	Specify the IP address of the RADIUS server to clear from the configuration. Use the all keyword to clear all of the servers from the configuration.	clear radius server [<i>ip_addr</i> all]
Step 2	Verify the RADIUS server configuration.	show radius

This example shows how to clear a single RADIUS server from the configuration:

```

Console> (enable) clear radius server 172.20.52.3
172.20.52.3 cleared from radius server table.
Console> (enable)

```

This example shows how to clear all RADIUS servers from the configuration:

```
Console> (enable) clear radius server all
All radius servers cleared from radius server table.
Console> (enable)
```

Clearing the RADIUS Key

To clear the RADIUS key, perform this task in privileged mode:

	Task	Command
Step 1	Clear the RADIUS key.	clear radius key
Step 2	Verify the RADIUS configuration.	show radius

This example shows how to clear the RADIUS key and verify the configuration:

```
Console> (enable) clear radius key
Radius key cleared.
Console> (enable)
```

Disabling RADIUS Authentication

If you disable RADIUS authentication with both TACACS+ and local authentication disabled, local authentication is reenabled automatically.

To disable RADIUS authentication, perform this task in privileged mode:

	Task	Command
Step 1	Disable RADIUS authentication for login mode.	set authentication login radius disable [all console http telnet]
Step 2	Disable RADIUS authentication for enable mode.	set authentication enable radius disable [all console http telnet]
Step 3	Verify the RADIUS configuration.	show radius show authentication

This example shows how to disable RADIUS authentication:

```
Console> (enable) set authentication login radius disable
radius login authentication set to disable for console and telnet session.
Console> (enable) set authentication enable radius disable
radius enable authentication set to disable for console and telnet session.
Console> (enable) show authentication
```

```
Login Authentication:  Console Session  Telnet Session
-----
tacacs                 disabled          disabled
radius                 disabled          disabled
local                  enabled(primary) enabled(primary)
```

```
Enable Authentication: Console Session  Telnet Session
-----
tacacs                 disabled          disabled
```

```
radius          disabled          disabled
local          enabled(primary)  enabled(primary)
Console> (enable)
```

Configuring Kerberos Authentication

Before you can use Kerberos as an authentication method on the switch, you need to configure the Kerberos server. You will need to create a database for the KDC and add the switch to the database.



Note

Kerberos authentication requires that NTP is enabled. Additionally, we recommend that you enable DNS.

To configure the Kerberos server, perform this procedure:

- Step 1** Before you can enter the switch in the Kerberos server's key table, you must create the database the KDC will use. In the following example, a database called CISCO.EDU is created:

```
/usr/local/sbin/kdb5_util create -r CISCO.EDU -s
```

- Step 2** Add the switch to the database. The following example adds a switch called Cat5509 to the CISCO.EDU database.

```
ank host/Cat5509.cisco.edu@CISCO.EDU
```

- Step 3** Add the user name.

```
ank user1@CISCO.EDU
```

- Step 4** Add the Administrative Principals.

```
ank user1/admin@CISCO.EDU
```

- Step 5** Create the entry for the switch in the database, using the **admin.local ktadd** command.

```
ktadd host/Cat5509.cisco.edu@CISCO.EDU
```

- Step 6** Move the keytab file to a place where the switch can reach it.

- Step 7** Start the KDC server.

```
/usr/local/sbin/krb5kdc
/usr/local/sbin/kadmind
```

The following sections describe how to configure Kerberos authentication on the switch.

- Enabling Kerberos, page 30-26
- Defining the Kerberos Local-Realm, page 30-27
- Specifying a Kerberos Server, page 30-27
- Mapping a Kerberos Realm to a Host Name or DNS Domain, page 30-28
- Copying SRVTAB Files, page 30-28
- Enabling Credentials Forwarding, page 30-30
- Defining a Private DES Key, page 30-31

- Encrypting a Telnet Session, page 30-32
- Monitoring and Maintaining Kerberos, page 30-33

Enabling Kerberos

To enable Kerberos authentication, perform this task in privileged mode:

	Task	Command
Step 1	Specify Kerberos as the authentication method.	set authentication login kerberos enable [all console http telnet] [primary]
Step 2	Verify the configuration.	show authentication

This example shows how to enable Kerberos as the login authentication method for Telnet and verify the configuration:

```
kerberos> (enable) set authentication login kerberos enable telnet
kerberos login authentication set to enable for telnet session.
kerberos> (enable) show authentication
```

```
Login Authentication: Console Session  Telnet Session
-----
tacacs                disabled          disabled
radius               disabled          disabled
kerberos             disabled          enabled(primary)
local                 enabled(primary) enabled
```

```
Enable Authentication: Console Session  Telnet Session
-----
tacacs                disabled          disabled
radius               disabled          disabled
kerberos             disabled          enabled(primary)
local                 enabled(primary) enabled
kerberos> (enable)
```

This example shows how to enable Kerberos as the login authentication method for the console and verify the configuration:

```
kerberos> (enable) set authentication login kerberos enable console
kerberos login authentication set to enable for console session.
kerberos> (enable) show authentication
```

```
Login Authentication: Console Session  Telnet Session
-----
tacacs                disabled          disabled
radius               disabled          disabled
kerberos             enabled(primary) enabled(primary)
local                 enabled           enabled
```

```
Enable Authentication: Console Session  Telnet Session
-----
tacacs                disabled          disabled
radius               disabled          disabled
kerberos             enabled(primary) enabled(primary)
local                 enabled           enabled
kerberos> (enable)
```

Defining the Kerberos Local-Realm

The Kerberos realm is a domain consisting of users, hosts, and network services that are registered to a Kerberos server. To authenticate a user defined in the Kerberos database, the switch must know the host name or IP address of the host running the KDC and the name of the Kerberos realm.

To configure the switch to authenticate to the KDC in a specified Kerberos realm, perform this task in privileged mode:

Task	Command
Define the default realm for the switch.	set kerberos local-realm <i>kerberos-realm</i>



Note

Make sure the realm is entered in uppercase letters. Kerberos will not authenticate users if the realm is entered in lowercase letters.

This example shows how to define a local-realm and how to verify the configuration:

```

kerberos> (enable) set kerberos local-realm CISCO.COM
Kerberos local realm for this switch set to CISCO.COM.
kerberos> (enable) show kerberos
Kerberos Local Realm:CISCO.COM
Kerberos server entries:
Realm:CISCO.COM, Server:187.0.2.1, Port:750

Kerberos Domain<->Realm entries:
Domain:cisco.com, Realm:CISCO.COM

Kerberos Clients NOT Mandatory
Kerberos Credentials Forwarding Enabled
Kerberos Pre Authentication Method set to None
Kerberos config key:
Kerberos SRVTAB Entries
Srvtab Entry 1:host/niners.cisco.com@CISCO.COM 0 932423923 1 1 8 01;;8>00>50;0=0=0
kerberos> (enable)

```

Specifying a Kerberos Server

You can specify to the switch which KDC to use in a specific Kerberos realm. Optionally, you can also specify the port number which the KDC is monitoring. The Kerberos server information you enter is maintained in a table with one entry for each Kerberos realm. The maximum number of entries in the table is 100.

To specify the Kerberos server, perform this task in privileged mode:

	Task	Command
Step 1	Specify which KDC to use in a given Kerberos realm. Optionally, enter the port number the KDC is monitoring. (The default port number is 750.)	set kerberos server <i>kerberos-realm</i> { <i>hostname</i> <i>ip-address</i> } [<i>port-number</i>]
Step 2	Clear the Kerberos server entry.	clear kerberos server <i>kerberos-realm</i> { <i>hostname</i> <i>ip-address</i> } [<i>port-number</i>]

This example shows how to define which Kerberos server will serve as the KDC for the specified Kerberos realm and how to clear the entry:

```
kerberos> (enable) set kerberos server CISCO.COM 187.0.2.1 750
Kerberos Realm-Server-Port entry set to:CISCO.COM - 187.0.2.1 - 750
kerberos> (enable)

Console> (enable) clear kerberos server CISCO.COM 187.0.2.1 750
Kerberos Realm-Server-Port entry CISCO.COM-187.0.2.1-750 deleted
Console> (enable)
```

Mapping a Kerberos Realm to a Host Name or DNS Domain

Optionally, you can map a host name or domain name server (DNS) domain to a Kerberos realm.

To map a Kerberos realm to either a host name or DNS domain, perform this task in privileged mode:

	Task	Command
Step 1	Optionally, map a host name or DNS domain to a Kerberos realm.	set kerberos realm { <i>dns-domain</i> <i>host</i> } <i>kerberos-realm</i>
Step 2	Clear the Kerberos realm domain or host mapping entry.	clear kerberos realm { <i>dns-domain</i> <i>host</i> } <i>kerberos-realm</i>

This example shows how to map a Kerberos realm to a DNS domain and how to clear the entry:

```
Console> (enable) set kerberos realm CISCO CISCO.COM
Kerberos DnsDomain-Realm entry set to CISCO - CISCO.COM
Console> (enable)

Console> (enable) clear kerberos realm CISCO CISCO.COM
Kerberos DnsDomain-Realm entry CISCO - CISCO.COM deleted
Console> (enable)
```

Copying SRVTAB Files

To make it possible for remote users to authenticate to the switch using Kerberos credentials, the switch must share a key with the KDC. To allow this configuration, you must give the switch a copy of the file that is stored in the KDC and which contains the key. These files are called SRVTAB files on the switch and KEYTAB files on the servers.

The most secure method to copy SRVTAB files to the hosts in your Kerberos realm is to copy them onto physical media and then manually copy the files onto the system. To copy SRVTAB files to a switch that does not have a physical media drive, you must transfer them through the network by using the Trivial File Transfer Protocol (TFTP).

When you copy the SRVTAB file from the switch to the KDC, the switch parses the information in this file and stores it in the running configuration in the Kerberos SRVTAB entry format. If you enter the SRVTAB directly into the switch, create an entry for each Kerberos principal (service) on the switch. The entries are maintained in the SRVTAB table. The maximum size of the table is 20 entries.

To remotely copy SRVTAB files to the switch from the KDC, perform this task in privileged mode:

	Task	Command
Step 1	Retrieve a specified SRVTAB file from the KDC.	set kerberos srvtab remote {hostname ip-address} filename
Step 2	Optionally, you can enter the SRVTAB directly into the switch.	set kerberos srvtab entry kerberos-principal principal-type timestamp key-version number key-type key-length encrypted-keytab

This example shows how to retrieve a SRVTAB file from the KDC, enter a SRVTAB directly into the switch, and verify the configuration:

```

kerberos> (enable) set kerberos srvtab remote 187.20.32.10 /users/jdoe/krb5/ninerskeytab
kerberos> (enable)

kerberos> (enable) set kerberos srvtab entry host/niners.cisco.com@CISCO.COM 0 932423923 1
1 8 03;;5>00>50;0=0=0
Kerberos SRVTAB entry set to
Principal:host/niners.cisco.com@CISCO.COM
Principal Type:0
Timestamp:932423923
Key version number:1
Key type:1
Key length:8
Encrypted key tab:03;;5>00>50;0=0=0

kerberos> (enable) show kerberos
Kerberos Local Realm:CISCO.COM
Kerberos server entries:
Realm:CISCO.COM, Server:187.0.2.1, Port:750
Realm:CISCO.COM, Server:187.20.2.1, Port:750

Kerberos Domain<->Realm entries:
Domain:cisco.com, Realm:CISCO.COM

Kerberos Clients NOT Mandatory
Kerberos Credentials Forwarding Enabled
Kerberos Pre Authentication Method set to None
Kerberos config key:
Kerberos SRVTAB Entries
Srvtab Entry 1:host/niners.cisco.com@CISCO.COM 0 932423923 1 1 8 03;;5>00>50;0=0=0
Srvtab Entry 2:host/niners.cisco.edu@CISCO.EDU 0 933974942 1 1 8 00?58:127:223=::;9
kerberos> (enable)

```

To delete the SRVTAB entry, perform this task in privileged mode:

Task	Command
Delete the SRVTAB entry for a particular Kerberos principal.	clear kerberos srvtab entry kerberos-principal principal-type

This example shows how to clear a SRVTAB entry:

```

kerberos> (enable) clear kerberos srvtab entry host/niners.cisco.com@CISCO.COM 0
kerberos> (enable)

```

Enabling Credentials Forwarding

A user authenticated to a Kerberized switch has a TGT and can use it to authenticate to a host on the network. However, if forwarding is not enabled and a user tries to list credentials after authenticating to a host, the output will show no Kerberos credentials present.

To enable credentials forwarding, configure the switch to forward user TGTs when they authenticate from the switch to Kerberized remote hosts on the network using Kerberized Telnet.

As an additional layer of security, you can configure the switch so that after users authenticate to it, these users can authenticate only to other services on the network with Kerberized clients. If you do not make Kerberos authentication mandatory and Kerberos authentication fails, the application attempts to authenticate users using the default method of authentication for that network service. For example, Telnet prompts for a password.

To configure clients to forward user credentials as they connect to other hosts in the Kerberos realm, perform this task in privileged mode:

	Task	Command
Step 1	Set all clients to forward user credentials upon successful Kerberos authentication.	set kerberos credentials forward
Step 2	Optionally, configure Telnet to fail if clients cannot authenticate to the remote server.	set kerberos clients mandatory

This example shows how to configure clients to forward user credentials and verify the configuration:

```

kerberos> (enable) set kerberos credentials forward
Kerberos credentials forwarding enabled
kerberos> (enable) show kerberos
Kerberos Local Realm:CISCO.COM
Kerberos server entries:
Realm:CISCO.COM, Server:187.0.2.1, Port:750
Realm:CISCO.COM, Server:187.20.2.1, Port:750

Kerberos Domain<->Realm entries:
Domain:cisco.com, Realm:CISCO.COM

Kerberos Clients NOT Mandatory
Kerberos Credentials Forwarding Enabled
Kerberos Pre Authentication Method set to None
Kerberos config key:
Kerberos SRVTAB Entries
Srvtab Entry 1:host/aspens-niners.cisco.edu@CISCO.EDU 0 933974942 1 1 8 00?91:107:423=:;9
kerberos> (enable)

```

This example shows how to configure the switch so that Kerberos clients are mandatory for users to authenticate to other network services:

```

Console> (enable) set kerberos clients mandatory
Kerberos clients set to mandatory
Console> (enable)

```

To clear the credentials forwarding configuration, perform this task in privileged mode:

	Task	Command
	Clear the credentials forwarding configuration.	clear kerberos credentials forward

This example shows how to clear the credentials forwarding configuration and verify the change:

```

Console> (enable) clear kerberos credentials forward
Kerberos credentials forwarding disabled
Console> (enable) show kerberos
Kerberos Local Realm not configured
Kerberos server entries:

Kerberos Domain<->Realm entries:

Kerberos Clients NOT Mandatory
Kerberos Credentials Forwarding Disabled
Kerberos Pre Authentication Method set to None
Kerberos config key:
Kerberos SRVTAB Entries
Console> (enable)

```

To clear the Kerberos clients mandatory configuration, perform this task in privileged mode:

Task	Command
Clear the Kerberos clients mandatory configuration.	clear kerberos clients mandatory

This example shows how to clear the clients mandatory configuration and verify the change:

```

Console> (enable) clear kerberos clients mandatory
Kerberos clients mandatory cleared
Console> (enable) show kerberos
Kerberos Local Realm not configured
Kerberos server entries:

Kerberos Domain<->Realm entries:

Kerberos Clients NOT Mandatory
Kerberos Credentials Forwarding Disabled
Kerberos Pre Authentication Method set to None
Kerberos config key:
Kerberos SRVTAB Entries
Console> (enable)
Kerberos server entries:

Kerberos Domain<->Realm entries:

Kerberos Clients Mandatory
Kerberos Credentials Forwarding Disabled
Kerberos Pre Authentication Method set to Encrypted Unix Time Stamp
Kerberos config key:
Kerberos SRVTAB Entries
Console> (enable)

```

Defining a Private DES Key

You can define a private DES key for the switch. The private DES key can be used to encrypt the secret key that the switch shares with the KDC so that when the **show kerberos** command is executed, the secret key is not displayed in clear text. The key length should be eight characters or less.

To define a DES key, perform this task in privileged mode:

Task	Command
Define a DES key for the switch.	set key config-key <i>string</i>

This example shows how to define a DES key and verify the configuration:

```

kerberos> (enable) set key config-key abcd
Kerberos config key set to abcd
kerberos> (enable) show kerberos
Kerberos Local Realm:CISCO.COM
Kerberos server entries:
Realm:CISCO.COM, Server:170.20.2.1, Port:750
Realm:CISCO.COM, Server:172.20.2.1, Port:750

Kerberos Domain<->Realm entries:
Domain:cisco.com, Realm:CISCO.COM

Kerberos Clients Mandatory
Kerberos Credentials Forwarding Disabled
Kerberos Pre Authentication Method set to Encrypted Unix Time Stamp
Kerberos config key:abcd
Kerberos SRVTAB Entries
Srvtab Entry 1:host/aspens-niners.cisco.edu@CISCO.EDU 0 933974942 1 1 8 12151<<88?=>>3>11
kerberos> (enable)

```

To clear the DES key, perform this task in privileged mode:

Task	Command
Clear a DES key from the switch.	clear key config-key <i>string</i>

This example shows how to clear the DES key:

```

Console> (enable) clear key config-key
Kerberos config key cleared
Console> (enable)

```

Encrypting a Telnet Session

After a user authenticates to the switch using Kerberos and wants to Telnet to another switch or host, whether this will be a Kerberized Telnet session depends on the authentication method that the Telnet server uses. If the Telnet server uses Kerberos for authentication, you can have all the application data packets encrypted during the duration of the Telnet session. To encrypt the Telnet session, select the **encrypt kerberos** option in the **telnet** command.

To encrypt a Telnet session, perform this task in privileged mode:

Task	Command
Encrypt a Telnet session.	telnet [encrypt kerberos] <i>host</i>

This example shows how to configure a Telnet session for Kerberos authentication and encryption:

```

Console> (enable) telnet encrypt kerberos 172.20.52.5

```

Monitoring and Maintaining Kerberos

These commands can be used to display and clear Kerberos configurations on the switch:

- **clear kerberos creds**
- **show kerberos**
- **show kerberos creds**

To clear all Kerberos credentials, perform this task in privileged mode:

Task	Command
Clear all credentials.	clear kerberos creds

This example shows how to clear all credentials from the switch:

```
Console> (enable) clear kerberos creds
Console> (enable)
```

To display Kerberos configuration information, perform this task in privileged mode:

Task	Command
Display Kerberos configuration information.	show kerberos

This example shows how to display Kerberos configuration information:

```
kerberos> (enable) show kerberos
Kerberos Local Realm:CISCO.COM
Kerberos server entries:
Realm:CISCO.COM, Server:187.0.2.1, Port:750
Realm:CISCO.COM, Server:187.20.2.1, Port:750

Kerberos Domain<->Realm entries:
Domain:cisco.com, Realm:CISCO.COM

Kerberos Clients NOT Mandatory
Kerberos Credentials Forwarding Enabled
Kerberos Pre Authentication Method set to None
Kerberos config key:
Kerberos SRVTAB Entries
Srvtab Entry 1:host/niners.cisco.com@CISCO.COM 0 932423923 1 1 8 03;;5>00>50;0=0=0
Srvtab Entry 2:host/niners.cisco.edu@CISCO.EDU 0 933974942 1 1 8 00?58:127:223=::;9
kerberos> (enable)
```

To display Kerberos credentials information, perform this task in privileged mode:

Task	Command
Display Kerberos credentials information.	show kerberos creds

This example shows how to display the Kerberos credentials:

```
Console> (enable) show kerberos creds
No Kerberos credentials.
Console> (enable)
```

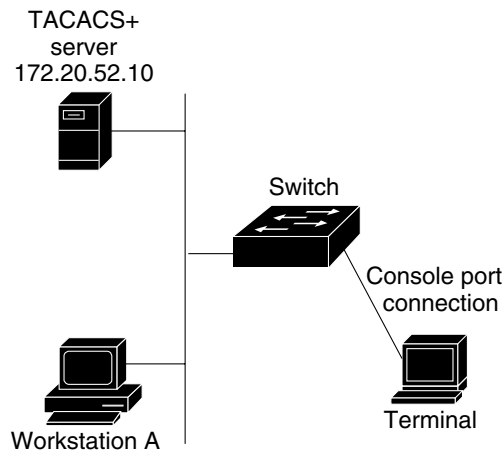
Authentication Example

Figure 30-3 shows a simple network topology using TACACS+.

In this example, TACACS+ authentication is enabled and local authentication is disabled for both login and enable access to the switch for all Telnet connections. When Workstation A attempts to connect to the switch, the user is challenged for a TACACS+ username and password.

However, only local authentication is enabled for both login and enable access on the console port. Any user with access to the directly connected terminal can access the switch using the login and enable passwords.

Figure 30-3 TACACS+ Example Network Topology



18927

This example shows how to configure the switch so that TACACS+ authentication is enabled for Telnet connections and local authentication is enabled for console connections. In addition, a TACACS+ encryption key is specified.

```
Console> (enable) show tacacs
Tacacs key:
Tacacs login attempts: 3
Tacacs timeout: 5 seconds
Tacacs direct request: disabled
```

```
Tacacs-Server                               Status
-----
Console> (enable) set tacacs server 172.20.52.10
172.20.52.10 added to TACACS server table as primary server.
Console> (enable) set tacacs key tintin_et_milou
The tacacs key has been set to tintin_et_milou.
Console> (enable) set authentication login tacacs enable telnet
tacacs login authentication set to enable for telnet session.
Console> (enable) set authentication enable tacacs enable telnet
tacacs enable authentication set to enable for telnet session.
Console> (enable) set authentication login local disable telnet
local login authentication set to disable for telnet session.
Console> (enable) set authentication enable local disable telnet
local enable authentication set to disable for telnet session.
Console> (enable) show tacacs
Tacacs key: tintin_et_milou
Tacacs login attempts: 3
```

```
Tacacs timeout: 5 seconds
Tacacs direct request: disabled

Tacacs-Server                               Status
-----
172.20.52.10                                 primary
Console> (enable)
```

Understanding How Authorization Works

These sections describe how authorization works:

- Authorization Overview, page 30-35
- Authorization Events, page 30-35
- TACACS+ Primary Options and Fallback Options, page 30-36
- TACACS+ Command Authorization, page 30-36
- RADIUS Authorization, page 30-37

Authorization Overview

Your switch supports TACACS+ and RADIUS authorization to control access to the switch. Authorization limits access to specified users using a dynamically applied access list (or user profile) based on the username and password pair. The access list resides on the host running the TACACS+ or RADIUS server. The server responds to the user password information and applies the access list.

Authorization Events

You can enable TACACS+ authorization for the following:

- **Commands**—When the authorization feature is enabled for commands, the user must supply a valid username and password pair to execute certain commands. You can require authorization for all commands or for configuration (enable mode) commands only. When a user enters a command, the authorization server receives the command and user information and compares it against an access list. If the user is authorized to enter that command, the command is executed; otherwise, the command is not executed.
- **Exec mode (normal login)**—When the authorization feature is enabled for exec mode, the user must supply a valid username and password pair to access the exec mode. Authorization is required only if you have enabled the authorization feature.
- **Enable mode (privileged login)**—When the authorization feature is enabled for enable mode, the user must supply a valid username and password pair to access enable mode. Authorization is required only if you have enabled the authorization feature for enable mode.

TACACS+ Primary Options and Fallback Options

You can specify the primary option and fallback option used in the authorization process. The following options and fallback options are available:

- **tacacs+**—If you have been authenticated, and there is no response from the TACACS+ server, authorization succeeds immediately.
- **if-authenticated**—If you have been authenticated, and there is no response from the TACACS+ server, authorization succeeds immediately.
- **none**—Authorization succeeds if the TACACS+ server does not respond.
- **deny**—Deny is a fallback option only. Authorization fails if the TACACS+ server fails to respond. This is the default behavior.

TACACS+ Command Authorization

You can require authorization for all commands or for configuration (enable mode) commands only. Configuration commands include the following:

- **copy**
- **clear**
- **commit**
- **configure**
- **delete**
- **download**
- **format**
- **reload**
- **rollback**
- **session**
- **set**
- **squeeze**
- **switch**
- **undelete**

The following TACACS+ authorization process occurs for every command that you enter:

- If you have disabled the command authorization feature, the TACACS+ server allows you to execute any command on the switch.
- If you have enabled authorization for configuration commands only, the switch verifies that the argument string matches one of the commands listed above. If there is no match, the switch completes the command. If there is a match, the switch forwards the command to the NAS for authorization.
- If you have enabled authorization for all commands, the switch forwards the command to the NAS for authorization.

RADIUS Authorization

RADIUS has limited authorization. The Service-Type attribute in the authentication protocol provides authorization information. This attribute is part of the user-profile.

When you login using RADIUS authentication and you do not have Administrative/Shell (6) Service-Type access, the NAS authenticates you and logs you in to exec mode if authentication succeeds. If you have Administrative/Shell (6) Service-Type access, the NAS authenticates you and logs you in to privileged mode if authentication succeeds.

Configuring Authorization

These sections describe how to configure authorization:

- Authorization Default Configuration, page 30-37
- TACACS+ Authorization Configuration Guidelines, page 30-37
- Configuring TACACS+ Authorization, page 30-38

Authorization Default Configuration

Table 30-3 shows the default authorization configuration.

Table 30-3 Default Authorization Configuration

Feature	Default Value
TACACS+ login authorization (console and Telnet)	Disabled
TACACS+ exec authorization (console and Telnet)	Disabled
TACACS+ enable authorization (console and Telnet)	Disabled
TACACS+ commands authorization (console and Telnet)	Disabled

TACACS+ Authorization Configuration Guidelines

These guidelines apply when configuring authorization on the switch:

- TACACS+ authorization is disabled by default.
- Authorization configuration applies to console connections, Telnet connections, or both types of connections.
- You must specify the mode, primary option, fallback option, and connection type when enabling authorization.
- Configure RADIUS and TACACS+ servers before enabling authorization. See the “Specifying TACACS+ Servers” section on page 30-12, or the “Specifying RADIUS Servers” section on page 30-19, for more information on server setup.
- Configure RADIUS and TACACS+ keys to encrypt protocol packets before enabling authorization. See the “Specifying the TACACS+ Key” section on page 30-14, or the “Specifying the RADIUS Key” section on page 30-20, for more information on the key setup.

Configuring TACACS+ Authorization

These sections describe how to configure TACACS+ authorization on the switch.

- Enabling TACACS+ Authorization, page 30-38
- Disabling TACACS+ Authorization, page 30-39

Enabling TACACS+ Authorization

To enable TACACS+ authorization on the switch, perform this task in privileged mode:

	Task	Command
Step 1	Enable authorization for normal login mode. Use the console or telnet keywords if you want to enable authorization only for console port or Telnet connection attempts. Use the both keyword to enable authorization for both console port and Telnet connection attempts.	set authorization exec enable {option} {fallbackoption} [console telnet both]
Step 2	Enable authorization for enable mode. Use the console or telnet keywords if you want to enable authorization only for console port or Telnet connection attempts. Use the both keyword to enable authorization for both console port and Telnet connection attempts.	set authorization enable enable {option} {fallbackoption} [console telnet both]
Step 3	Enable authorization of configuration commands. Use the console or telnet keywords if you want to enable authorization only for console port or Telnet connection attempts. Use the both keyword to enable authorization for both console port and Telnet connection attempts.	set authorization commands enable {config all } {option} {fallbackoption} [console telnet both]
Step 4	Verify the TACACS+ authorization configuration.	show authorization

This example shows how to enable TACACS+ exec mode authorization for both console and Telnet connections. Authorization is configured with the **tacacs+** option. The fallback option is **deny**:

```
Console> (enable) set authorization exec enable tacacs+ deny both
Successfully enabled enable authorization.
Console>
```

This example shows how to enable TACACS+ enable mode authorization for console and Telnet connections. Authorization is configured with the **tacacs+** option. The fallback option is **deny**:

```
Console> (enable) set authorization enable enable tacacs+ deny both
Successfully enabled enable authorization.
Console>
```

This example shows how to enable TACACS+ command authorization for both console and Telnet connections. Authorization is configured with the **tacacs+** option. The fallback option is **deny**:

```
Console> (enable) set authorization commands enable config tacacs+ deny both
Successfully enabled commands authorization.
Console> (enable)
```

This example shows how to verify the configuration:

```

Console> (enable) show authorization
Telnet:
-----
                Primary   Fallback
                -----   -----
exec:           tacacs+   deny
enable:         tacacs+   deny
commands:
  config:       tacacs+   deny
  all:          -         -

Console:
-----
                Primary   Fallback
                -----   -----
exec:           tacacs+   deny
enable:         tacacs+   deny
commands:
  config:       tacacs+   deny
  all:          -         -
Console> (enable)

```

Disabling TACACS+ Authorization

To disable TACACS+ authorization on the switch, perform this task in privileged mode:

	Task	Command
Step 1	Disable authorization for normal mode. Use the console or telnet keywords if you want to disable authorization only for console port or Telnet connection attempts. Use the both keyword to enable authorization for both console port and Telnet connection attempts.	set authorization exec disable [console telnet both]
Step 2	Disable authorization for enable mode. Use the console or telnet keywords if you want to disable authorization only for console port or Telnet connection attempts. Use the both keyword to enable authorization for both console port and Telnet connection attempts.	set authorization enable disable [console telnet both]
Step 3	Disable authorization of configuration commands. Use the console or telnet keywords if you want to disable authorization only for console port or Telnet connection attempts. Use the both keyword to enable authorization for both console port and Telnet connection attempts.	set authorization commands disable [console telnet both]
Step 4	Verify the TACACS+ authorization configuration.	show authorization

This example shows how to disable TACACS+ exec mode authorization for both console and Telnet connections and how to verify the configuration:

```
Console> (enable) set authorization exec disable both
Successfully disabled enable authorization.
Console> (enable)
```

This example shows how to disable TACACS+ enable mode authorization for both console and Telnet connections and how to verify the configuration:

```
Console> (enable) set authorization enable disable both
Successfully disabled enable authorization.
Console> (enable)
```

This example shows how to disable TACACS+ command authorization for both console and Telnet connections and how to verify the configuration:

```
Console> (enable) set authorization commands disable both
Successfully disabled commands authorization.
Console> (enable)
```

This example shows how to verify the configuration:

```
Console> (enable) show authorization
```

```
Telnet:
-----
           Primary   Fallback
           -
exec:      tacacs+   deny
enable:    tacacs+   deny
commands:
  config:  tacacs+   deny
  all:     tacacs+   deny

Console:
-----
           Primary   Fallback
           -
exec:      tacacs+   deny
enable:    tacacs+   deny
commands:
  config:  tacacs+   deny
  all:     tacacs+   deny
Console> (enable)
```

Authorization Example

Figure 30-3 shows a simple network topology using TACACS+.

In this example, TACACS+ authorization is enabled for enable mode access to the switch for both Telnet and console connections, authorizing configuration commands. When Workstation A initiates a command on the switch, the switch registers a request with the TACACS+ daemon. The TACACS+ daemon determines if the user is authorized to use the feature and sends a response either executing the command or denying access.

```

Console> (enable) set authorization enable enable tacacs+ deny both
Successfully enabled enable authorization.
Console> (enable) set authorization commands enable config tacacs+ deny both
Successfully enabled commands authorization.
Console> (enable) show authorization
Telnet:
-----
                Primary      Fallback
                -----      -
exec:           tacacs+      deny
enable:        tacacs+      deny
commands:
  config:      tacacs+      deny
  all:         tacacs+      deny
Console:
-----
                Primary      Fallback
                -----      -
exec:           tacacs+      deny
enable:        tacacs+      deny
commands:
  config:      tacacs+      deny
  all:         tacacs+      deny
Console> (enable)

```

Understanding How Accounting Works

These sections describe how accounting works:

- Accounting Overview, page 30-41
- Accounting Events, page 30-42
- Specifying When to Create Accounting Records, page 30-42
- Specifying RADIUS Servers, page 30-43
- Updating the Server, page 30-44
- Suppressing Accounting, page 30-44

Accounting Overview

You can configure these accounting methods to monitor access to the switch:

- TACACS+ accounting
- RADIUS accounting

Accounting allows you to track user activity to a specified host, suspicious connection attempts in the network, and unauthorized changes to the NAS configuration. The accounting information is sent to the accounting server where it is saved in the form of a record. Accounting information typically consists of the user's action and the duration for which the action lasted. You can use the accounting feature for security, billing, and resource allocation purposes.

The accounting protocol operates in a client-server model, using TCP for transport. The NAS acts as the client and the accounting server acts as the daemon. The NAS sends accounting information to the server. The server, after successfully processing the information, sends a response to the NAS, acknowledging the request. All transactions between the NAS and server are authenticated using a key.

After accounting has been enabled and an accountable event occurs on the system, the accounting information is gathered dynamically in memory. When the event ends, an accounting record is created and sent to the NAS, and then the system deletes the record from memory. The amount of memory used by the NAS for accounting varies depending on the number of concurrent accountable events.

Accounting Events

You can configure accounting for the following types of events:

- Exec mode accounting—Provides information about user exec sessions (normal login sessions) on the NAS. This information includes the duration of the exec session but does not include traffic statistics.
- Connect accounting—Provides information about all outbound connections from the NAS (such as Telnet, rlogin).



Note If you get a connection immediately upon login and then your connection is terminated, the exec and connect events will overlap and will have almost identical start and stop times.

- System accounting—Provides information on system events not related to users. This information includes system reset, system boot, and user configuration of accounting.
- Command accounting—Sends a record for each command issued by the user. This permits audit trail information to be gathered.

Specifying When to Create Accounting Records

You configure the switch to gather accounting information to create records. When Accounting is configured (using the **set accounting command**), the switch can generate two types of records:

- Start records—Include partial information of the event (when the event started, type of service, and traffic statistics).
- Stop records—Include complete information of the event (when the event started, its duration, type of service, and traffic statistics).

Accounting records are created and sent to the server at two events:

- Start-stop—Accounting records are sent at both the start and stop of an action, if the action has duration. If the NAS fails to send the accounting record at the start of the action, it still allows you to proceed with the action.

- **Stop-only**—Accounting records are sent only at the termination of the event. Commands are assumed to have zero duration, so only stop records are generated for command accounting. No users are associated with system events; therefore, the **start-stop** option in the **set accounting system** command is ignored for system events. The **stop-only** option in the **set accounting** commands provides complete accounting information.

**Note**

Stop records include complete information of the event (when the event started, its duration, and traffic statistics). However, you might want redundancy and also to monitor both start and stop records of events occurring on the NAS.

Specifying RADIUS Servers

To specify one or more RADIUS servers, perform this task in privileged mode:

	Task	Command
Step 1	Specify the IP address of up to three RADIUS servers. Specify the primary server using the primary keyword. Optionally, specify the destination UDP port to use on the server.	set radius server <i>ip_addr</i> [acct-port <i>port_number</i>] [primary]
Step 2	Verify the RADIUS server configuration.	show radius

This example shows how to specify a RADIUS server and verify the configuration:

```
Console> (enable) set radius server 172.20.52.3
172.20.52.3 with auth-port 1812 added to radius server table as primary server.
Console> (enable) show radius
```

```
Login Authentication: Console Session  Telnet Session
-----
tacacs                disabled          disabled
radius                disabled          disabled
local                 enabled(primary) enabled(primary)

Enable Authentication: Console Session  Telnet Session
-----
tacacs                disabled          disabled
radius                disabled          disabled
local                 enabled(primary) enabled(primary)

Radius Deadtime:           0 minutes
Radius Key:
Radius Retransmit:        2
Radius Timeout:           5 seconds

Radius-Server              Status  Auth-port
-----
172.20.52.3                primary 1812
Console> (enable)
```

Updating the Server

You can configure the switch to send accounting information to the TACACS+ server. There are two options:

- **Newinfo**—Sends accounting information to the server only when new accounting information becomes available.
- **Periodic**—Sends accounting update records at regular intervals. This option could be used to keep up-to-date connection and session information even if the NAS restarts and loses the initial start time. You must set a time lapse between periodic updates. Valid intervals are from 1 to 71582 minutes.

Suppressing Accounting

You can configure the system to suppress accounting when an unknown user with no username accesses the switch by using the **set accounting suppress null-username enable** command.



Note

RADIUS and TACACS+ accounting are the same, except that RADIUS does not do command accounting, periodic updates, or allow null-username suppression.

Configuring Accounting

These sections describe how to configure accounting for both TACACS+ and RADIUS:

- Accounting Default Configuration, page 30-44
- Accounting Configuration Guidelines, page 30-45
- Configuring Accounting, page 30-45

Accounting Default Configuration

Table 30-4 shows the default accounting configuration.

Table 30-4 Accounting Default Configuration

Feature	Default Value
Accounting	Disabled
Accounting events (exec, system, commands, and connect)	Disabled
Accounting records	Stop-only

Accounting Configuration Guidelines

These guidelines apply when configuring accounting on the switch:

- Configure RADIUS and TACACS+ servers before enabling accounting. See the “Specifying TACACS+ Servers” section on page 30-12, or the “Specifying RADIUS Servers” section on page 30-19, for more information on server setup.
- Configure RADIUS and TACACS+ keys to encrypt protocol packets before enabling accounting. See the “Specifying the TACACS+ Key” section on page 30-14, or the “Specifying the RADIUS Key” section on page 30-20, for more information on the key setup.



Note

The amount of DRAM allocated for one accounting event is approximately 500 bytes. The total amount of DRAM used by accounting will depend on the number of concurrent accountable events occurring in the system.

Configuring Accounting

These sections describe how to configure RADIUS and TACACS+ accounting on the switch:

- Enabling Accounting, page 30-45
- Disabling Accounting, page 30-47

Enabling Accounting

To enable accounting on the switch, perform this task in privileged mode:

	Task	Command
Step 1	Enable accounting for connection events.	set accounting connect enable {start-stop stop-only} {tacacs+ radius}
Step 2	Enable accounting for exec mode.	set accounting exec enable {start-stop stop-only} {tacacs+ radius}
Step 3	Enable accounting for system events.	set accounting system enable {start-stop stop-only} {tacacs+ radius}
Step 4	Enable accounting of configuration commands.	set accounting commands enable {config all} {stop-only} tacacs+
Step 5	Enable suppression of information for unknown users.	set accounting suppress null-username enable
Step 6	Configure accounting to be updated as new information is available.	set accounting update {new-info {periodic [interval]}}
Step 7	Verify the accounting configuration.	show accounting

This example shows how to enable stop-only TACACS+ accounting events:

```

Console> (enable) set accounting connect enable stop-only tacacs+
Accounting set to enable for connect events in stop-only mode.
Console> (enable)

Console> (enable) set accounting exec enable stop-only tacacs+
Accounting set to enable for exec events in stop-only mode.
Console> (enable)

Console> (enable) set accounting system enable stop-only tacacs+
Accounting set to enable for system events in stop-only mode.
Console> (enable)

Console> (enable) set accounting commands enable all stop-only tacacs+
Accounting set to enable for commands-all events in stop-only mode.
Console> (enable)

```

This example shows how to suppress accounting of unknown users:

```

Console> (enable) set accounting suppress null-username enable
Accounting will be suppressed for user with no username.
Console> (enable)

```

This example shows how to periodically update the server:

```

Console> (enable) set accounting update periodic 120
Accounting updates will be periodic at 120 minute intervals.
Console> (enable)

```

This example shows how to verify the configuration:

```

Console> (enable) show accounting
Event      Method  Mode
-----  -
exec:      tacacs+ stop-only
connect:   tacacs+ stop-only
system:    tacacs+ stop-only
commands:
config:    -      -
all:       tacacs+ stop-only
TACACS+ Suppress for no username: enabled
Update Frequency: periodic, Interval = 120

Accounting information:
-----
Active Accounted actions on tty0, User (null) Priv 0
Active Accounted actions on tty288091924, User (null) Priv 0
Overall Accounting Traffic:

```

	Starts	Stops	Active
	-----	-----	-----
Exec	0	0	0
Connect	0	0	0
Command	0	0	0
System	1	0	0

```

Console> (enable)

```

Disabling Accounting

To disable accounting on the switch, perform this task in privileged mode:

	Task	Command
Step 1	Disable accounting for connection events.	set accounting connect disable
Step 2	Disable accounting for exec mode.	set accounting exec disable
Step 3	Disable accounting for system events.	set accounting system disable
Step 4	Disable accounting of configuration commands.	set accounting commands disable
Step 5	Disable suppression of information for unknown users.	set accounting suppress null-username disable
Step 6	Verify the accounting configuration.	show accounting

This example shows how to disable stop-only accounting:

```
Console> (enable) set accounting connect disable
Accounting set to disable for connect events.
Console> (enable)
```

```
Console> (enable) set accounting exec disable
Accounting set to disable for exec events.
Console> (enable)
```

```
Console> (enable) set accounting system disable
Accounting set to disable for system events.
Console> (enable)
```

```
Console> (enable) set accounting commands disable
Accounting set to disable for commands-all events.
Console> (enable)
```

This example shows how to disable suppression of unknown users:

```
Console> (enable) set accounting suppress null-username disable
Accounting will be not be suppressed for user with no username.
Console> (enable)
```

This example shows how to verify the configuration:

```
Console> (enable) show accounting
Event      Method  Mode
-----  -
exec:      -      -
connect:   -      -
system:    -      -
commands:
config:    -      -
all:       -      -

TACACS+ Suppress for no username: disabled
Update Frequency: new-info

Accounting information:
-----
Active Accounted actions on tty0, User (null) Priv 0
Active Accounted actions on tty288091924, User (null) Priv 0
Overall Accounting Traffic:
      Starts  Stops  Active
```

```

-----
Exec          0      0      0
Connect       0      0      0
Command       0      0      0
System        1      2      0
Console> (enable)

```

Accounting Example

Figure 30-3 shows a simple network topology using TACACS+.

In this example, TACACS+ accounting is enabled for connection, exec, system, and all command accounting. Accounting information is gathered at the conclusion of the event. Accounting is suspended for unknown users and the system is updated periodically every 120 minutes.

When Workstation A initiates an accountable event on the switch, the switch gathers event information and forwards the information to the server at the conclusion of the event.

```

Console> (enable) set accounting connect enable stop-only tacacs+
Accounting set to enable for connect events in stop-only mode.
Console> (enable) set accounting exec enable stop-only tacacs+
Accounting set to enable for exec events in stop-only mode.
Console> (enable) set accounting commands enable all stop-only tacacs+
Accounting set to enable for commands-all events in stop-only mode.
Console> (enable) set accounting update periodic 120
Accounting updates will be periodic at 120 minute intervals.
Console> (enable) show accounting
Event      Method  Mode
-----  -
exec:      tacacs+ stop-only
connect:   tacacs+ stop-only
system:    tacacs+ stop-only
commands:
config:    -      -
all:       tacacs+ stop-only

```

```

TACACS+ Suppress for no username: enabled
Update Frequency: periodic, Interval = 120

```

```

Accounting information:
-----
Active Accounted actions on tty0, User (null) Priv 0
Active Accounted actions on tty288091924, User (null) Priv 0
Overall Accounting Traffic:
          Starts  Stops  Active
          -----
Exec          0      0      0
Connect       0      0      0
Command       0      0      0
System        1      0      0
Console> (enable)

```