



Cisco Secure Telnet Configuration

This chapter provides a description and overview of Cisco Secure Telnet and contains the following topics:

- [Cisco Secure Telnet Components, page 26-2](#)
- [Cisco Secure Telnet Applications, page 26-3](#)
- [Cisco Secure Telnet Usage Scenario, page 26-7](#)

Cisco Secure Telnet offers Cisco Service Engineers (CSE) transparent firewall access to Cisco CallManager servers on your site.

Cisco Secure Telnet works by enabling a Telnet client inside the Cisco Systems firewall to connect to a Telnet daemon behind your firewall. This secure connection allows remote monitoring and maintenance of your Cisco CallManager servers without requiring firewall modifications.



Note

Cisco accesses your network only with your permission. You must provide a network administrator at your site to help initiate the process.

Cisco Secure Telnet Components

The following sections describe the Cisco Secure Telnet components.

The Relay Server

The Cisco relay server runs on the Windows NT platform. Because it is a node on the public Internet that is configured as a multiuser system, any Cisco customer using the Cisco Secure Telnet service can access it freely.

You can make the relay server a dedicated device, or during a pilot period, you may attach it on an as-needed basis.

The relay server resides outside the Cisco Systems firewall, but it comprises a secure and controlled system that Cisco Systems owns, manages, and operates.

**Note**

If easy access to the Internet is not available to access the relay server, use a dial-in connection to an ISP that allows direct Internet access to its devices.

Telnet Client

The Telnet client runs at the Cisco site on a UNIX host or Windows NT system. It provides terminal emulation over TCP/IP, providing a remote shell allowing entry into the server at your site.

The Telnet client provides command line access to the Cisco CallManager host through the Cisco relay server.

Telnet Server

The Telnet server resides on your network and runs on the Cisco CallManager server. The Windows 2000 operating system runs on the Cisco CallManager server and supports execution of the text-based commands used on the local system.

The Telnet proxy, **tndconnect**, runs on the Cisco CallManager server and links your Telnet server to the Cisco relay server.

**Note**

The **tndconnect** window running on the Windows 2000 system will display the commands and responses being issued between the Cisco TAC and your Cisco CallManager system. However, depending on its terminal type support, the Windows 2000 Telnet daemon may eliminate spaces between the elements.

Cisco Secure Telnet Applications

The Cisco Secure Telnet system comprises four software components.

1. The Telnet Daemon Connect (**tndconnect**) program runs on a Cisco CallManager server at your site.
2. The relay application (**relayapp**) program runs on the Cisco relay or “connect” server.
3. The Windows 2000 Telnet Daemon comprises Microsoft software that runs on the Cisco CallManager server at your site.
4. The standard Telnet client runs behind the firewall at the Cisco TAC.

Cisco Secure Telnet Executables

Invoke the **tndconnect** program command-line executable from a Windows 2000 command prompt window.

Remote serviceability users invoke the **tndconnect** program from a command window on the Cisco CallManager server. This program contacts the **relayapp** application on the relay server, which resides on the Internet.

After you invoke **tndconnect** at your site, the CSE uses Telnet to reach **relayapp** to gain access to your system. Logging in to that system maps the Telnet sessions together, allowing technical support access to your site.

Each executable includes command line parameters, such as passwords and TCP ports, that control the operating characteristics of each program.

Telnet Proxy

The **tndconnect** program acts as a proxy that provides the connection from your site to an external application residing on the relay server.

If you use this connector program, you must specify certain command line parameters; some of these parameters are optional.

Command Line Syntax for tndconnect

The **tndconnect** program resides on your Cisco CallManager server at C:\Program Files\Cisco\Bin. Invoke it on a command line; for example:

```
tndconnect -host relayservername -password cisco -file connect.log -port 80 -verbose -noecho
```



Note

You can terminate **tndconnect** by entering Control-C in the window.

The following list defines each of these parameters:

-host *<relay hostname>*

You must use the host argument because it defines the Domain Name System (DNS) name of target relay server. The CSE supplies you with that host name during the initial call.

-password *<any 4+ character string>*

You must use the password argument because it allows access to the relay server. CSE gives the password during the initial call.

-file *<logfile name>*

All Telnet exchanges generate logs, allowing for later review. The **-file** parameter provides an opportunity to name the log files that provide an audit trail of startup and console activity. The system logs all activity to the command window.

If the **-file** argument is not used, the filename defaults to **tndconnect.log**.

-port *<optional port number>*

The port argument allows selection of a port on the relay server other than Telnet port 23, which is the default. You need this option in case the firewall at your site has blocked the Telnet port. For example, some servers might allow only HTTP transmissions through the firewall, in which case **-port 80** could be used.

-verbose

Use the verbose option if connectivity problems are anticipated. This parameter logs debug messages and program trace details to the log file as well as to the console window.

-target *<optional host name of Cisco CallManager system>*

Use this parameter only if **tndconnect** resides on a system other than the Cisco CallManager server; for example, a Telnet server at your site.

In this case, you would have to specify the host name of the Cisco CallManager system. If you do not use this argument, its value defaults to localhost.

-noecho

Disable the echo of the relayed data to the console.

Display All Options

Invoke the **tndconnect** program with the **/?** argument on a command line to give information on the entire system:

tndconnect /?

This command returns information on all the command line options.

Telnet Connector Program Structure

The **tndconnect** program performs the tasks that provide the CSE with the ability to log in to your Cisco CallManager system.

**Note**

If the Telnet daemon is not running when the program executes, **tndconnect** prints an error message on your console, and the command terminates.

When the **tndconnect** program starts, the command parser sets the values specified in the command line. If no specific parameters are set in the command line, the program uses default values, except for the **-host** and **-password** parameters. You must enter those values each time that you start **tndconnect**.

When the CSE connects to the relay application, a signal instructs the **tndconnect** program to create TCP/IP structures and to connect to the local Telnet daemon.

If the **-target** option is not used, **tndconnect** defaults to your local Telnet server.

**Note**

You may have to add the name of the Cisco relay application server to the hosts file of the local system if DNS cannot locate it. If so, add a line with *IP address* and *hostname* to the file C:\WINNT\system32\drivers\etc\hosts to provide that hostname.

The **tndconnect** program sends the information to the Cisco relay server, where the CSE logs in. The CSE then provides the IP address and password of the Cisco CallManager and enters a return, which prompts the appearance of a login screen. The CSE then logs in to the Cisco CallManager system using the password provided by the customer.

Terminate a Cisco Secure Telnet Session

Once a connection is established, a timer starts and eventually terminates the connection if no traffic is observed. The tunnel to your site established by **tndconnect** automatically terminates after 30 minutes of inactivity. If you deliberately disconnect, you must manually restart the Telnet Connector program each time that you use it.

Any socket failure can also terminate the session as does closing the connection from either Cisco or the customer site. A close can occur when the CSE disconnects, the relay program ends, or if the Telnet daemon terminates. Termination may also occur if TCP detects connection failure, if either of the Telnet sessions closes, or if you terminate the **tndconnect** program by pressing *Control-C*.

When the connection terminates, the program terminates, and the program takes down the tunnel constructed between the Cisco CallManager Telnet server and the relay server.

Connect with the Telnet Daemon

Under the Cisco Secure Telnet system, the Telnet client and daemon comprise standard, unmodified components and exchange data just as they would if they connected directly.

Any system running Cisco CallManager must run the standard Windows 2000 Telnet daemon service. Because this program expects clients to connect into it, you need an additional software component, **tndconnect**, to both connect to the Telnet daemon and to establish a connection out of the corporate network and into the Telnet relay program. The Telnet daemon and **tndconnect** work together to provide an end-to-end connection through which Telnet session traffic may flow.

The Telnet server component can run as a service in the background on the Cisco CallManager Windows 2000 system. Responding to a connection request provides the primary function of the daemon. Once a Telnet session is established, you can execute command line programs.

All these considerations affect the choice of Telnet daemon software.

Plan for a Windows NT Telnet Daemon

Because both Telnet daemon and File Transfer Protocol (FTP) servers make up part of Windows 2000 standard features, you can logically use these connectivity methods for remote serviceability.

Cisco Secure Telnet Usage Scenario

You may want to activate Cisco Secure Telnet to call in problems that require diagnosis of a remote Cisco CallManager server by a CSE. The following sequence of events represents a common scenario.

-
- Step 1** Configure the Windows 2000 Telnet daemon to allow Telnet access from UNIX hosts. You may use the Windows 2000 command **tndadmn** to set the operating characteristics of the Telnet daemon.



Note Make sure that the local Telnet session works and that a suitable userid and password are chosen for the Cisco TAC to log in.

- Step 2** The CSE gives you a one-time password and the DNS name of the Cisco relay server to use to open communications.

- Step 3** If the Telnet daemon is not already running, you should start it by using the Windows 2000 System Control Panel, Services option.
- Step 4** You run **tndconnect** to initiate a tunnel session between the Telnet Daemon and the Cisco relay server. The command-line syntax allows transmission of a one-time password, which acts as a correlator to the one supplied by the CSE upon connection to the relay server.



Note Tunnels allow secure transmission of data streams between networks, making the routing appear transparent. Create them by using software that communicates through the firewall protecting the destination network.

- Step 5** After bit manipulation encrypts the one-time password for transmission, **tndconnect** transmits it to **relayapp** for identification.
- When transmission is complete, your system is set up for Cisco TAC diagnostics. (You may observe the support engineer commands and responses in the **tndconnect** window.)
- When the Cisco TAC engineer disconnects, the **tndconnect** program terminates.
-