CHAPTER 3

Accessing the GSS CLI

You can access the GSS CLI in one of the following ways:

- Making a direct connection to the GSS device using a dedicated terminal.
- Establishing a remote connection using the Secure Shell (SSH), Telnet, or FTP protocols from a PC.

This chapter contains the following sections:

- Accessing the CLI Using a Direct Serial Connection
- Logging in to the CLI and Enabling Privileged EXEC Mode
- Remotely Accessing a GSS Device
- Where to Go Next

Accessing the CLI Using a Direct Serial Connection

To access the GSS CLI by using a serial connection, establish a direct serial connection between your terminal and the GSS device. For information on how to establish a serial connection with your device, see the *Cisco Global Site Selector Hardware Installation Guide.*

Once connected, use any terminal communications application to access the GSS CLI. The following procedure uses HyperTerminal for Windows.

To access the GSS CLI using a direct serial connection, perform the following steps:

1. Launch HyperTerminal. The Connection Description window appears.
2. Enter a name for your session in the Name field.
3. Click **OK**. The Connect To window appears.
4. From the drop-down list, choose the COM port to which the device is connected.
5. Click **OK**. The Port Properties window appears.
6. Set the port properties:
   - Baud Rate = 9600
   - Data Bits = 8
   - Flow Control = none
   - Parity = none
   - Stop Bits = 1
7. Click **OK** to connect.

8. Press **Enter** to display the CLI prompt.

Once a session is created, choose **Save As** from the File menu to save the connection description. Saving the connection description has the following two advantages:

- The next time you launch HyperTerminal, the session is listed as an option under **Start > Programs > Accessories > HyperTerminal > Name_of_session**. This option lets you reach the CLI prompt directly without going through the configuration steps.
- You can connect your cable to a different device without configuring a new HyperTerminal session. If you use this option, make sure that you connect to the same port on the new device that you configured in the saved HyperTerminal session. Otherwise, a blank screen appears without a prompt.

Logging in to the CLI and Enabling Privileged EXEC Mode

After you make a direct connection to the GSS device using a dedicated terminal, log in to a GSS device and enable privileged EXEC mode at the CLI. To do so, perform the following steps:

1. Press the power control button on the GSS. After the GSS boot process completes, the software prompts you to log in to the device.
2. Specify your GSS administrative username and password to log in to the GSS device. If this is your first time logging in to the GSS, use the default account name (admin) and password (default) to access the CLI.

   The CLI prompt appears.

   ```
   localhost.localdomain>
   ```

3. At the CLI prompt, enable privileged EXEC mode.

   ```
   localhost.localdomain> enable
   localhost.localdomain#
   ```

   The prompt changes from the user-level EXEC right angle bracket (>) prompt to the privileged-level EXEC pound sign (#).

Remotely Accessing a GSS Device

To monitor the performance of your GSS devices and administer those devices once deployed, you require remote access to those devices. Once you have basic network connectivity on the GSS device, you may use the CLI to enable remote access to each device using the Secure Shell (SSH), Telnet, or FTP protocols.

We recommend using an SSH connection because SSH provides secure communication over insecure channels and provides strong authentication. The GSS supports remote login to the GSS over an SSH session that uses private and public key pairs for authentication.

You must have physical access to the GSS device to set up remote access by Telnet or SSH connection. See the *Cisco Global Site Selector Hardware Installation Guide* for instructions on connecting a console cable to your Cisco Global Site Selector series hardware.

This section contains the following topics:

- Enabling Remote Access on a GSS Device
- Configuring the enable Command Password
- Accessing the CLI Using a Remote Connection

- Accessing the CLI Over SSH Using a Private and Public Key Pair

Enabling Remote Access on a GSS Device

To enable SSH, Telnet, or FTP on your GSS device, perform the following steps:

1. Log on to the GSS and enable privileged EXEC mode as described in the "Logging in to the CLI and Enabling Privileged EXEC Mode" section.
2. Enable global configuration mode on the device.

   ```
   localhost.localdomain# config
   localhost.localdomain(config)#
   ```

3. To enable Secure Shell (SSH) connections to the GSS device, use the **ssh enable** command. SSH on the GSS supports the SSH v2 and v1 protocols. For SSH v2, the software provides 128-bit AES, Blowfish, 3DES, CAST128, Arcfour, 192-bit AES, or 256-bit AES. For SSH v1, the software provides encrypted communication using ciphers such as 3DES or Blowfish.

   ```
   localhost.localdomain(config)# ssh enable
   ```

   By default, the GSS turns off SSH protocol v1 because it is considered to be cryptographically insecure. If your remote SSH application cannot support SSH protocol v2 and requires SSH protocol v1, enter the following command to enable SSH protocol version 1 for the GSS:

   ```
   localhost.localdomain(config)# ssh protocol version 1
   ```

   Note If your clients support both SSH protocol v2 and v1, we recommend that you configure the client to use SSH protocol v2 by default.

   To disable SSH, use the **no** form of this command.

   ```
   localhost.localdomain(config)# no ssh enable
   ```

4. To enable Telnet on the selected GSS device and to establish a Telnet connection, use the **telnet** command. The syntax of this command is as follows:

   telnet {**enable** | {*ip_or_host*} | [*port*]

   The keywords and arguments are as follows:

   - *ip_or_host*—Specifies the IP address or hostname of the device with which you want to establish a Telnet connection. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1) or a mnemonic hostname (for example, myhost.mydomain.com).
   - *port*—(Optional) Allows you to change the port number for the Telnet session to a port other than 23 (the Telnet port). Enter a number from 1 to 65535. The default is 23.

   For example, to launch Telnet and establish a connection to the device at IP address 192.168.2.3, enter:

   ```
   localhost.localdomain(config)# telnet enable
   localhost.localdomain# telnet 192.168.2.3
   ```

   To disable Telnet on your GSS device, use the **no** form of this command.

   ```
   localhost.localdomain(config)# no telnet enable
   ```

5. To enable the File Transfer Protocol (FTP) or launch an FTP session on your GSS device, use the **ftp enable** command. The syntax of this command is as follows:

ftp enable | *ip_or_host*

The *ip_or_host* option specifies the IP address or hostname of the FTP server you want to access. Be sure to enter the IP address in dotted-decimal notation (for example, 192.168.11.1) or a mnemonic hostname (for example, myhost.mydomain.com).

For example, to launch the FTP session and access the device at IP address 192.168.0.1, enter:

```
localhost.localdomain(config)# ftp enable
localhost.localdomain(config)# ftp 192.168.0.1
```

To disable FTP on your GSS device or remove the IP address from the FTP server, use the **no** form of this command.

```
localhost.localdomain(config)# no ftp enable
localhost.localdomain(config)# no ftp 192.168.0.1
```

6. To enable access to an FTP client for different types of users, use the **ftp-client enable** command. The syntax of this command is as follows:

   ftp-client enable {**all** | **admin**}

   The **all** keyword enables FTP client access for all users, while the **admin** keyword enables FTP client access for administrative users only. For example:

   ```
   localhost.localdomain(config)# ftp-client enable all
   localhost.localdomain(config)# ftp-client enable admin
   ```

   To remove a specific FTP client configuration and return to the default disabled state, use the **no** form of this command.

   ```
   localhost.localdomain(config)# no ftp-client enable all
   ```

7. Save your configuration changes to memory.

   ```
   localhost.localdomain(config)# copy running-config startup-config
   ```

8. Exit global configuration mode.

   ```
   localhost.localdomain(config)# exit
   localhost.localdomain#
   ```

Note the following SSH, Telnet, and FTP remote access considerations:

- The GSS supports a maximum limit of 40 concurrent Telnet or FTP sessions within a 60-second window. The GSS can receive additional concurrent Telnet and FTP connections that are made outside of a 60-second window.
- The GSS supports a maximum limit of 250 SSH connections. When the GSS reaches this limit, the `Connection terminated on signal 13` message appears at the CLI of the computer where you initiated the SSH session to the GSS.

To view the operating status of the remote access protocol (SSH, Telnet, or FTP) on your GSS device, enter the following commands:

- To view if FTP and the FTP client are enabled for the GSS device, enter:

  ```
  localhost.localdomain# show ftp
  ftp is enabled
  ftp-client is enabled for all users
  ```

- To view if Telnet is enabled for the GSS device, enter:

  ```
  localhost.localdomain# show telnet
  telnet is enabled
  ```

- To view if SSH is enabled for the GSS device, enter:

```
localhost.localdomain# show ssh
ssh is enabled
```

Configuring the enable Command Password

You can control user access to the privileged Exec mode for users that remotely connect to the GSS using Telnet or SSH. When connecting to the GSS remotely, the CLI prompts you for a password when you enter the **enable** command to access the privileged Exec mode. The default password is default. The admin user can change the **enable** command password by using the **enable-passwd** command.

The syntax of this command is as follows:

enable-passwd

When you enter this command, the CLI prompts you for an admin password and then to define and confirm the enable password. The password is alphanumeric, can contain spaces and special characters, and can contain a maximum of 32 characters. Leave the password blank to set the password to the default value, which is default.

Note The enable password is not required when you access the GSS using a console or terminal session. If you forget the enable password, use a console or terminal session to configure a new password.

The following example shows how to configure the enable password:

```
localhost.localdomain# enable-passwd
Admin Password: <admin_password>
Set GSS enable Password: <enable_password>
Confirm GSS enable Password: <enable_password>
```

Accessing the CLI Using a Remote Connection

Use either Telnet or SSH from a PC to remotely access the GSS CLI. You cannot connect to more than one device during a single Telnet or SSH session. You can, however, have several Telnet or SSH sessions running in parallel for different devices. Before you attempt to remotely access a GSS device, ensure that you enable Telnet or SSH on that device (see the "Enabling Remote Access on a GSS Device" section).

We recommend using an SSH connection because SSH provides secure communication over insecure channels and provides strong authentication. The GSS supports remote login to the GSS over an SSH session by using a private and public key pair for authentication.

To access the GSS CLI using your preferred SSH or Telnet client, perform the following steps:

1. Enter the hostname or IP address of the GSS or GSSM.
2. Specify your GSS administrative username and password to log in to the GSS device.

Accessing the CLI Over SSH Using a Private and Public Key Pair

The GSS supports remote login to the GSS over an SSH session that uses private and public key pairs for authentication. With this method of remote connection, use a generated private and public key pair to participate in a secure communication by encrypting and decrypting messages. Use of a private and public key pair bypasses the normal username and password authentication process. This remote access method may be useful when running scripts that connect automatically to the GSS.

Generate the private key and the corresponding public key as a key pair on a server separate from the GSS and then use the **scp** command on the GSS to copy the public key to the GSS /home directory. The **scp** command automatically creates an .ssh folder under the GSS /home directory.

By default, the GSS disables SSH key support. As a one-time process, after you initially copy the private and public keys onto the GSS, you must enable global access to those keys to remotely log in to the GSS.

To generate a private and public key pair and copy the keys to the GSS, perform the following steps:

1. Generate the SSH private key and the corresponding SSH public key as a key pair on a server separate from the GSS. See the documentation included with the SSH software for details on generating the private and public key pair.
2. When the SSH private and public key is available, log on to the GSS and enable privileged EXEC mode as described in the "Logging in to the CLI and Enabling Privileged EXEC Mode" section.
3. Use the **scp** command from the GSS to securely copy the generated public key from the server to the GSS /home directory. The **scp** command automatically creates an .ssh folder under the GSS /home directory.

   ```
   localhost.localdomain# scp myusername@1myhost:~/mykey.pub .
   myusername@1myhost password:
   mykey.pub 100% |*****************************| 241 00:00
   ```

   After generating the public key, you may FTP the generated public key to the GSS. In this case, while you are in FTP mode, you must use the **mkdir** command to manually create the .ssh folder on the FTP server.
4. Use the **type** command to append the public key to the /home/.ssh/authorized_keys file, which is a special file that the GSS software looks for when authenticating public/private keys.

   ```
   localhost.localdomain# cd .ssh
   localhost.localdomain# type ../mykey.pub >> authorized_keys
   ```

5. Activate an SSH session from the remote host to the GSS using the private key. For example, on most Unix systems you would enter the following command line:

   ```
   ssh -i private.key gss.cisco.com
   ```

6. Globally enable remote access to the copied private and public keys on the GSS by entering the following command:

   ```
   localhost.localdomain# config
   localhost.localdomain(config)# ssh keys
   ```

You do not need to enter the **ssh keys** command again for subsequent private and public keys that you copy to the GSS.

Where to Go Next

To configure your GSS device from the CLI and connect it to the GSS network, proceed to Chapter 4, Setting Up Your GSS from the CLI. This process also includes information on how to configure the GSS as a primary GSSM, standby GSSM, or as a GSS device.

If you automatically configured the GSS using the setup script (see Chapter 2, Configuring the GSS Using the CLI Setup Script), you may need to configure additional GSS parameters, such as setting the system clock and adjusting Ethernet interface parameters.

If you do not need to configure additional GSS parameters, access the primary GSSM GUI to activate and configure your GSS devices. Proceed to Chapter 5, Activating GSS Devices from the GUI.

You can also activate and configure your GSS devices from the CLI. To do so, proceed to Chapter 6, Activating GSS Devices from the CLI.