Lab 16.4.7 - Configure Network Devices with SSH

This lab has been updated for use on NETLAB+.
www.netdevgroup.com

Topology

Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/0/1	192.168.1.1	255.255.255.0	N/A
S1	VLAN 1	192.168.1.11	255.255.255.0	192.168.1.1
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1

Objectives

Part 1: Configure Basic Device Settings

Part 2: Configure the Router for SSH Access

Part 3: Configure the Switch for SSH Access

Part 4: SSH from the CLI on the Switch

Background / Scenario

In the past, Telnet was the most common network protocol used to remotely configure network devices. Telnet does not encrypt the information between the client and server. This allows a network sniffer to intercept passwords and configuration information.

Secure Shell (SSH) is a network protocol that establishes a secure terminal emulation connection to a router or other networking device. SSH encrypts all information that passes over the network link and provides authentication of the remote computer. SSH is rapidly replacing Telnet as the remote login tool of choice for network professionals. SSH is most often used to log in to a remote device and execute commands. However, it can also transfer files using the associated Secure FTP (SFTP) or Secure Copy (SCP) protocols.

The network devices that are communicating must be configured to support SSH in order for SSH to function. In this lab, you will enable the SSH server on a router and then connect to that router using a PC with an SSH client installed. On a local network, the connection is normally made using Ethernet and IP.

Note: The routers used with CCNA hands-on labs are Cisco 4221 with Cisco IOS XE Release 16.9.4 (universalk9 image). The switches used in the labs are Cisco Catalyst 2960s with Cisco IOS Release 15.2(2) (lanbasek9 image). Other routers, switches, and Cisco IOS versions can be used. Depending on the model

and Cisco IOS version, the commands available and the output produced might vary from what is shown in the labs. Refer to the Router Interface Summary Table at the end of the lab for the correct interface identifiers.

Instructions

Part 1: Configure Basic Device Settings

In Part 1, you will configure basic settings, such as the interface IP addresses, device access, and passwords on the router.

Step 1: Configure the router.

a. Click on the router to access the console port and enable privileged EXEC mode.

b. Enter configuration mode.

c. Disable DNS lookup to prevent the router from attempting to translate incorrectly entered commands as though they were host names.

d. Assign **class** as the privileged EXEC encrypted password.

e. Assign **cisco** as the console password and enable login.

f. Assign **cisco** as the VTY password and enable login.

g. Encrypt the plaintext passwords.

h. Create a banner that will warn anyone accessing the device that unauthorized access is prohibited.

i. Configure and activate the G0/0/1 interface on the router using the information contained in the Addressing Table.

j. Save the running configuration to the startup configuration file.

Step 2: Configure PC-A.

a. Configure PC-A with an IP address and subnet mask.

b. Configure a default gateway for PC-A.

Step 3: Verify network connectivity.

Ping R1 from PC-A. If the ping fails, troubleshoot the connection.

Part 2: Configure the Router for SSH Access

Using Telnet to connect to a network device is a security risk because all the information is transmitted in a clear text format. SSH encrypts the session data and provides device authentication, which is why SSH is recommended for remote connections. In Part 2, you will configure the router to accept SSH connections over the VTY lines.

Step 1: Configure device authentication.

The device name and domain are used as part of the crypto key when it is generated. Therefore, these names must be entered prior to issuing the **crypto key** command.

a. Configure device name.

b. Configure the domain for the device.

Step 2: Configure the encryption key method.

Use the **crypto key** command and select a modulus key size of 1024 bits.

Step 3: Configure a local database username.

Configure a username using **admin** as the username and **Adm1nP@55** as the password.

Step 4: Enable SSH on the VTY lines.

a. Enable Telnet and SSH on the inbound VTY lines using the **transport input** command.

b. Change the login method to use the local database for user verification.

c. Configure the encryption method to establish SSH connectivity between network devices.

```
R1(config)# IP ssh server algorithm encryption aes256-cbc aes128-cbc
R1(config)# end
```

Step 5: Save the running configuration to the startup configuration file.

Step 6: Establish an SSH connection to the router.

a. Start Tera Term from PC-A.

b. Establish an SSH session to R1. Use the username **admin** and password **Adm1nP@55**. You should be able to establish an SSH session with R1.

Part 3: Configure the Switch for SSH Access

In Part 3, you will configure the switch to accept SSH connections. After the switch has been configured, establish an SSH session using Tera Term.

Step 1: Configure the basic settings on the switch.

a. Click on the switch to access the console port and enable privileged EXEC mode.

b. Enter configuration mode.

c. Disable DNS lookup to prevent the switch from attempting to translate incorrectly entered commands as though they were host names.

d. Assign **class** as the privileged EXEC encrypted password.

e. Assign **cisco** as the console password and enable login.

f. Assign **cisco** as the VTY password and enable login.

g. Encrypt the plain text passwords.

h. Create a banner that will warn anyone accessing the device that unauthorized access is prohibited.

i. Configure and activate the VLAN 1 interface on the switch according to the Addressing Table.

j. Save the running configuration to the startup configuration file.

Step 2: Configure the switch for SSH connectivity.

Use the same commands that you used to configure SSH on the router in Part 2 to configure SSH for the switch.

a. Configure the device name as listed in the Addressing Table.

b. Configure the domain for the device.

c. Configure the encryption key method.

d. Configure a local database username.

e. Enable Telnet and SSH on the VTY lines.

f. Change the login method to use the local database for user verification.

Step 3: Establish an SSH connection to the switch.

Start Tera Term from PC-A, and then SSH to the SVI interface on S1.

Are you able to establish an SSH session with the switch?

Type your answers here.

Yes

Part 4: SSH From the CLI on the Switch

The SSH client is built into the Cisco IOS and can be run from the CLI. In Part 4, you will SSH to the router from the CLI on the switch.

Step 1: View the parameters available for the Cisco IOS SSH client.

Use the question mark (**?**) to display the parameter options available with the **ssh** command.

```
S1# ssh ?
  -c    Select encryption algorithm
  -l    Log in using this user name
  -m    Select HMAC algorithm
  -o    Specify options
  -p    Connect to this port
  -v    Specify SSH Protocol Version
  -vrf  Specify vrf name
  WORD  IP address or hostname of a remote system
```

Step 2: SSH to R1 from S1.

a. You must use the **–l admin** option when you SSH to R1. This allows you to log in as user **admin**. When prompted, enter **Adm1nP@55** for the password.

```
S1# ssh -l admin 192.168.1.1
Password:
Authorized Users Only!
R1>
```

b. You can return to S1 without closing the SSH session to R1 by pressing **Ctrl**+**Shift**+**6**. Release the **Ctrl**+**Shift**+**6** keys and press **x**. The switch privileged EXEC prompt displays.

```
R1>
S1#
```

c. To return to the SSH session on R1, press Enter on a blank CLI line. You may need to press Enter a second time to see the router CLI prompt.

```
S1#
[Resuming connection 1 to 192.168.1.1 ... ]

R1>
```

d. To end the SSH session on R1, type **exit** at the router prompt.

```
R1# exit
```

```
[Connection to 192.168.1.1 closed by foreign host]
S1#
```

What versions of SSH are supported from the CLI?

Using the "ssh -v ?" command, the switch lists the protocol versions. The CLI supports version 1 and 2.

Reflection Question

How would you provide multiple users, each with their own username, access to a network device?

Using the same command to create the "admin" users, someone could use that command to add multiple users to the local database. In a larger scenario, a user database server such as RADIUS could be set up.

Router and Switch Interface Summary Table

Router / Switch Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1800	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
1900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2801	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
2811	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2960	Fast Ethernet 0/1 (F0/1)	Fast Ethernet 0/2 (F0/2)	n/a	n/a
3560	Fast Ethernet 0/1 (F0/1)	Fast Ethernet 0/2 (F0/2)	n/a	n/a
3650	Gigabit Ethernet 1/0/1 (G1/0/1)	Gigabit Ethernet 1/0/2 (G1/0/2)	n/a	n/a
4221	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
4300	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)

Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. There is no way to effectively list all the combinations of configurations for each router class. This table includes identifiers for the possible combinations of Ethernet and Serial interfaces in the device. The table does not include any other type of interface, even though a specific router may contain one. An example of this might be an ISDN BRI interface. The string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.