CCNA Exam Prep: Back to Networking Basics with Hank Preston and Patrick Gargano

Securing Network Access: From Telnet to SSH

> Discover why SSH is the preferred choice over Telnet for secure network management in this insightful event. Through hands-on activities, master the ins and outs to configuring SSH on Cisco switches and routers, ensuring encrypted and authenticated remote access. Understand the security implications of using SSH, and elevate your skills in managing network devices securely and efficiently. Join us as you prepare for your CCNA exam.

No doubt you've heard that SSH is better than telnet because, well "security". But what does that mean? Is it really a problem? And how exactly do you ensure secure device adminsistration? Well, we will tackle all this in the following exercices:

In this lab you will explore device adminstration with some hands on exercises focused on:

- Enabling Telnet device administration, and seeing why it might not be the most secure
- Creating local administrator accounts on network devices
- Migrating to SSH from Telnet for device administration

> This lab touches on several topics from the CCNA v1.1 Topics List 2.8 Describe network device management access (Telnet, SSH, HTTP, HTTPS, console, TACACS+/RADIUS, and cloud managed) 4.8 Configure network devices for remote access using SSH 5.3 Configure and verify device access control using local passwords

Setup and Scenario

This setup of lab based demonstrations includes a small network made up of an IOS based switch and router, that provides internet access to a `netadmin` host. There is also an `outside-host` that will be used for some testing. The network is preconfigured to provide connectivity to the Internet for the `netadmin` host with DHCP, DNS, and NAT services provided by `rtr01`. `sw01` is configured with a management IP address, and there are DNS entries for both network devices configured to allow `netadmin` to reach the devices by name.

```
netadmin:~$ ping -c 2 sw01

PING sw01 (192.168.0.2): 56 data bytes
64 bytes from 192.168.0.2: seq=0 ttl=42 time=0.975 ms
64 bytes from 192.168.0.2: seq=1 ttl=42 time=1.418 ms

netadmin:~$ ping -c 2 rtr01

PING rtr01 (10.0.0.1): 56 data bytes
64 bytes from 10.0.0.1: seq=0 ttl=42 time=1.447 ms
64 bytes from 10.0.0.1: seq=1 ttl=42 time=1.125 ms
```

Go ahead and open the console for the `netadmin` host and try out the above pings yourself. "Pinging" is fun :-)!!

> Note: The credentials for `netadmin` and `outside-host` are `cisco / cisco`.

*Be sure to **START** the lab before continuing to the demo labs.*

Setting up initial IP based device management with Telnet

Using serial (console) connectivity to configure and operate networking equipment requires the engineer be in physical proximity to the device so that a cable can be plugged from the devices port into their laptop. This is fine during initial setup or emergency troubleshooting, but it isn't very convenient when you want to add a VLAN to a switch, update a router's static routes, or run some show commands to understand the current state of the network. The ability to log into the network devices over "the network" is a far more convenient option.

> There are dedicated "terminal server" or "console server" appliances that can be provide the direct connection to serial ports on devices and offer network administrators access over the network. Many organizations do deploy these to aid in remote adminstration of devices, but even when they are available, console based administration is typically reserved for times when devices are unreachable over the network. Upgrades, troubleshooting, etc.

While it is highly recommended that network devices only be administered using a secure protocol like SSH, we'll begin this lab by configuring telnet to fully explore how (and why) secure network access can be configured.

1. We'll start by verifying that we can NOT log into the network devices with telnet at the start of the lab. Open the console for `netadmin`. The username/password to log in is `cisco/cisco`.

2. Attempt to use telnet to log into the `rtr01`. DNS is configured so you can use the hostname.

   ```
   telnet rtr01

   # Output
   telnet: can't connect to remote host (10.0.0.1): Connection refused
   ```

3. Repeat for `sw01`.
4. Now that we've seen that telent does NOT work, we'll learn how to enable it.
5. Open the console for `rtr01` and enter enable mode.
6. To access the CLI for a network device over the network, you connect to a "VTY line", or "virtual teletype line". Different network devices have different numbers of VTY lines. The IOS router and switch in this lab each have 5 lines, numbered 0 -> 4. Another common number of lines is 16, numbered 0 -> 15.
7. Add the following configuration to `rtr01`. To enable `login` over the VTY lines and to allow communication on the line with `telnet`.

   ```
   line vty 0 4
    login
    transport input telnet
   ```

8. You should get a console message like the below. This is an exmaple of a very useful message to the administrator. While we've turned on logging in via telnet on the lines, the router will NOT allow that to happen until a password is set. Because once telnet administration is enabled, anyone connected to the network could begin administering the device. Security is important!

   ```
   % Login disabled on line 2, until 'password' is set
   ```

   - You will actually see several of these messages, 5 to be exact. Why do you think that is?

     ▶ Answer:

9. So go ahead and add the password to the `line vty 0 4` configuration block.

   ```
   password telnetpass
   ```

   Note: If you left `config-line` mode, you will need to re-enter with `line vty 0 4`.

10. Now return to the console for `netadmin` and try to connect with telnet. When you are prompted for the `Password:`, use the newly configured `telnetpass` that you just set.

    ```
    telnet rtr01

    # Output
    Connected to rtr01

    Entering character mode
    Escape character is '^]'.

    User Access Verification

    Password:
    rtr01>
    ```

11. Now enter enable mode. Did it work? Why not?

    ▶ Answer

12. Return to the console for `rtr01` and set an enable secret.

    ```
    enable secret enablepass
    ```

13. And now go back to the console for `netadmin` and try to enter enable mode again.
14. Ta-da... you can now administer `rtr01` remotely from your network administration workstation, from anywhere on the network.
15. Disconnect by typing `exit`.
16. Repeat the configuration to enable telnet administration on `sw01` and verify that you can login and access enable mode from `netadmin`.
17. Make sure to `exit` from the connection to `sw01` when you are done.

Understanding why clear text protocols like Telnet are a bad idea

Great work... maybe. I'm sure you've heard by now that "telnet bad, ssh good" but why? "Security" you may say. You might even know that the issue is because "telnet isn't encrypted". Both true, but let's see if we can experience this with our own eyes.

> ***Pro Tip!:*** CML allows you to "split" the Panes display into multiple sections to allow the viewing of two or more different interfaces (ie Console, VNC, or Packet Capture) at the same time. This feature is particularly handy when doing a packet capture, so you can see the captured packets while performing actions on a node.
>
> Click the + button on the right side of the "Panes" to split the window. Then "click" within the section before opening starting the capture. You can also "drag" a window from one pane to another.

1. Right-click the link between `netadmin` and `sw01` and choose "Packet Capture".
2. Click the gear icon to open the settings for the capture. Add the following BPF filter and "Apply" the settings change.

   ```
   tcp port 23
   ```

   Why tcp port 23?

   ▶ Answer
3. Start the packet capture.
4. Now return to the console for `netadmin` and connect to `rtr01` with telnet. Go ahead and enter the password to login.
5. Now look at the packet capture and click on Frame 6 that shows "Telnet Data..." in the Info column. Click on the frame and expand the `Telnet` section in the packet details. What do you see?
   ▶ Answer:
6. Change to page 2 and look at Frame 12, 14, 16, and the other "Telnet Data..." frames sourced by `netadmin`. What do you see?
   ▶ Answer:

   Note: The frame numbers above should match for you if you created the filter, started the capture, and then used `telnet rtr01` from `netadmin` and logged in with the password immediately. If you aren't seeing the expected data within the frame numbers indicated, click and check the "Telnet Data..." frames until you find them.

So now you see why "telnet bad". Anyone who has access to the network packets that have a telnet session running can see everything that is sent in both directions. And you don't need to do it frame by frame. There are many programs that can be used to analyze network streams, find telnet applications, capture the output and search for credentials.

> So why is "SSH good"? Because as an encrypted protocol, an attack like this isn't possible. You'll be able to try this out yourself and see it in action later!

Configuring local user accounts for managing network devices

Now the network team can all log into the network devices remotely from the convenience of anywhere on the network. However, everyone uses the same telnet password. Providing individual accounts to each user is a standard procedure for organizations and applications. There are many reasons to go down this path including:

- Allowing for accurate auditing of who is connected to the network and making changes
- Providing different levels of access to different people. Often called "Role Based Access Control" or "RBAC"
- Being able to disable access to individuals without effecting an entire team

There is another reason we are going to enable individual user accounts in this lab, SSH access requires it to be setup. And that's our ultimate destination :-)

1. Go ahead and log into `rtr1`. Either with the direct console connection, or through telnet from `netadmin`.
2. Enter configuration mode, and create a new admin user on the router.

   ```
   username admin secret adminpass
   ```

   Note: Be careful to ***NOT*** add a space after the password. If you do, the space becomes PART of the password. This is a common mistake that can lead to you being unable to log in and access your router. The author of this lab guide has made this mistake many many times.
3. Now you need to update the VTY configuration to use the "local" username/passwords rather than the telnet password that was configured.

   ```
   line vty 0 4
    login local
   ```

   - It is also a good idea to remove the telnet password that we won't be using anymore.

     ```
     line vty 0 4
      no password
     ```
4. Now return to the `netadmin` host and try to log into `rtr01` with telnet once again. Use the `admin / adminpass` credentials you just configured.

   ```
   telnet rtr01

       # Output
       User Access Verification

       Username: admin
       Password:
       rtr01>
   ```
5. Go ahead and enter "enable" mode. You should be prompted for the enable password, this will be the same `enablepass` that you configured before.

Adding privilege to user accounts

Now we'll enhance the login process by adding a "privilege level" to the user accounts. This isn't required, but it is a common configuration that will place "administrators" automatically into "enable" mode. We can also use it to provide "read only" access to "operator" users.

Cisco IOS uses "privilege levels" to track access levels. `priv 15` is the same as "Privileged EXEC mode" (ie "enable mode"), and `priv 1` is the same "User EXEC mode".

1. From the CLI of `rtr01` (either console or using the telnet access), add `priv 15` to the `admin` user.

   ```
   username admin priv 15
   ```

2. Now try to reconnect with telnet to the router.

   > If you configured the above from the telnet connection, `exit` out to disconnect and reconnect.

3. Did you see any difference?

   ▶ Answer:

4. Handy right? Create a new account with `priv 15` for yourself with whatever username and password you want to use. You can do all this with a single line of configuration.

   ```
   username carl priv 15 secret carlpass
   ```

5. Create one more user, `oper` with the secret password `operpass` that has `priv 1`.

   ```
   username oper priv 1 secret operpass
   ```

6. Take a look at the configuration for the user accounts.

   ```
   show run | section username

   # Output
   username admin privilege 15 secret 9 $9$JTuPJ0yjNpNuNk$hCbbeDhhX8DStDn5BKbOlJ7LWjvKkZP.wnFmZdNhwHA
   username carl privilege 15 secret 9 $9$O1J7C4XQ8aE/rU$TcDZQXJiv3h1c7gkqdgUqKJEhCydXfY2yz6hZzxWJbI
   username oper secret 9 $9$cjBNZ2V.CJT.fU$BDC0biwbJ8Wn.Lw1uau53B3jQhlFqYoEG5VF2BjpitY
   ```

   - What do you notice about them?

     ▶ Answer:

Configuring `sw01`

Before moving onto SSH, go ahead and configure `sw01`.

1. Create user accounts for `admin` and `oper` with appropriate `priv` levels
2. Update the VTY lines to use the newly created user accounts.
3. Remove the telnet password

Migrating from Telnet to SSH for managing network devices

Excellent work so far. Just one more thing to do, enable SSH and make our device administration secure. Well, there are a couple of steps needed to enable SSH, but let's get to it!

1. Log into `rtr01` from the console or with the `admin` account and telnet.

2. From "Privileged EXEC mode" (enable mode) and NOT "Configuration Mode", create an RSA key-pair.

   > Note: If you enter this command from `config` mode, you'll get an error message about deprecation of the command. Keys should now be generated from "enable mode".

   ```
   crypto key generate rsa general-keys modulus 2048

   # Output
   The name for the keys will be: rtr01.example.com

   % The key modulus size is 2048 bits

   % Generating crypto RSA keys in background ...
   ```

   - `rsa` is the type of key we are generating `ec` is an alternative key generation mechanism, but it won't work for SSH.
   - `general-keys` means we are creating a single key that can be used by the router for both signing and encryption.
   - `modulus 2048` indicates we are creating a `2048` bit key. The more bits, the more secure. `2048` is generally considered secure today, however some organizations opt for larger bit sizes for more security.

1. If you are on the console, or watching the console output, you'll also see this output.

```
*Mar  9 14:08:48.049: %CRYPTO_ENGINE-5-KEY_ADDITION: A key named rtr01.example.com has been generated or impor
*Mar  9 14:08:48.050: %SSH-5-ENABLED: SSH 2.0 has been enabled
*Mar  9 14:08:48.732: %CRYPTO_ENGINE-5-KEY_ADDITION: A key named rtr01.example.com.server has been generated o
```

   Note: The "name" of the RSA key is in the format of `<HOSTNAME>.<DOMAIN NAME>`. This means that if you have NOT configured either of these on your network device, you will get an error message. The initial configuration for this lab included both `hostname <HOSTNAME>` and `ip domain name <DOMAIN NAME>` configuration.

   - In the console log message, notice the `SSH-5-ENABLED` message. SSH is typically already setup to be "enabled" on an IOS device, but in order for it to work there needs to be a "key" that can be used to encrypt the traffic. As soon as you create the key, the device will enable SSH.
   - `SSH 2.0` is enabled. The version `2.0` is significant because earlier versions of SSH are insecure today and shouldn't be used. Some network devices may default to an earlier version and should be explicitly configured for `ip ssh version 2`. Newer devices likely only support version 2.

2. Go ahead and try to connect to `rtr01` with SSH from `netadmin`.

```
ssh admin@rtr01

# Output
ssh: connect to host rtr01 port 22: Connection refused
```

   - Why didn't it work? We say that SSH was enabled already?

     ▶ Answer:

3. Update the configuration of the VTY lines to support SSH.

```
line vty 0 4
 transport input ssh
```

   Note: You could enable both telnet and ssh at the same time with `transport input telnet ssh`, and if you are making this change remotely without console access to the router, that is recommended. But once you've verified that SSH is working, don't forget to remove it as a supported input protocol or you are still vulnerable.

4. Now try to log into `rtr01` with SSH again.

```
ssh admin@rtr01

# Output
The authenticity of host 'rtr01 (10.0.0.1)' can't be established.
RSA key fingerprint is SHA256:uKLBCGaK0AWdqR6NOVbvSFjB6Mc0GMiRGH7xcI7+X0w.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'rtr01' (RSA) to the list of known hosts.
(admin@rtr01) Password:

rtr01#
```

   - You will first be asked if you want to trust the fingerprint for the SSH key. The best practice for security would be to check the shown fingerprint against the known fingerprint for the device you are connecting to. Unfortunately, there isn't an easy way to find the fingerprint for an SSH key for an IOS device, and calculating it is out of scope for this lab or the CCNA. So just type "yes" at the prompt.
   - Once you connect the first time to a router, your computer will store the fingerprint and verify it on future connections to the router. If the fingerprint changes, you'll be warned something is different. This could indicate a security problem, *OR* it could indicate that key changed on the device. A key change can happen if a device is replaced or upgraded. The key can also change if an administrator manually changes it.
   - Provide the password for the `admin` user and log in.

5. Disconnect with SSH and try to log in with telnet. This should fail.

Verifying Encryption

Setup a new packet capture between `netadmin` and `sw01`, but use the filter `tcp port 22` for SSH traffic. With it running, log back into `rtr01` with SSH. Then checkout the packets captured. You'll see many packets setting up the secure communications, and then "Encrypted packets" being sent. If you look at those you'll find that you can NOT read the messages being sent between the devices.

Configuring SSH on `sw01`

Now go ahead and configure `sw01` for SSH with the same approach we used on `rtr01`.

1. Create a new RSA key
2. Change the VTY transport method to only support SSH

SSH from anywhere... even the Internet?

So far we've been testing SSH access from our "trusted host", `netadmin`. But can we also access the router from "the Internet"

1. Find the "public IP" for `rtr01`. This will be the IP address on interface 'E0/0'.

   ```
   show ip int brief

   # Output
   Interface              IP-Address      OK? Method Status                Protocol
   Ethernet0/0            192.168.255.196 YES DHCP   up                    up
   Ethernet0/1            192.168.0.1     YES TFTP   up                    up
   Ethernet0/2            unassigned      YES TFTP   administratively down down
   Ethernet0/3            unassigned      YES TFTP   administratively down down
   Loopback0              10.0.0.1        YES TFTP   up                    up
   ```

   > ***Note:*** The IP address assigned to YOUR `Ethernet0/0` interface will mostly likely differ from the above output. This is because the IP address is assigned with DHCP from the CML server itself. Make note of YOUR address for the next command.

2. Open up the console for `outside-host` and try to SSH into the router using the "public IP".

   - ***Remember to use the IP address from YOUR output in the below command.***

     ```
     ssh admin@<YOUR E0/0 IP address>
     ```

3. After answering "yes" to accept the fingerprint, this should work just like from `netadmin`. This is because the VTY lines are "available" from all router Interfaces. Even more reason to use SSH instead of a clear text protocol like telnet.

4. Now that you are connected, checkout the output from a couple of handy `show commands`.

   ```
   show ssh
   show users
   show crypto key mypubkey rsa
   ```

Great Job!

Excellent work on this lab! You've successfully enabled SSH on your network devices and seen why it is a much better choice than Telnet. This is by no means the end to device administration. Here are some other topics to look into.

1. How can you limit management access to a network device? Maybe you do NOT want the Internet to be able to log into your router - even if it is secure?
2. How do TACACS+ and RADIUS fit into device administration? How would they be configured?