Cisco config lab

Cisco Config Elements

Introduction

Goals

- Learn the basic set of IOS commands required to enable SSH on your Cisco Switch or Router

Notes

- Commands preceded with “$” imply that you should be working as a regular user.
- Commands preceded with “#” imply that you should be working as root.
- Commands with more specific command lines (e.g. “rtrX>” or “mysql>”) imply that you are executing commands on remote equipment, or within another program.
- If a command line ends with “\” this indicates that the command continues on the next line and you should treat this as a single line.
- References to “N” represent your group number.

Exercises Part I

Work as a group

Each group has 4 network devices:

- core1.campusY.ws.nsrc.org
- bdr1.campusY.ws.nsrc.org
- dist1-b1.campusY.ws.nsrc.org
- dist1-b2.campusY.ws.nsrc.org

Each of these devices has the user *nmmlab* configured with both a log in password and and enable password. At this time telnet is enabled on these devices and ssh is not yet configured.

As a group you need to update all four of these devices so that when you finish this lab you can log in as the *nmmlab* only using ssh and with the password given in class.

Connect to your router or switch

First, log in to your host (hostX.campusY.ws.nsrc.org).

Next, connect to the network device on which you will be working. Note, if you try to do this directly from your laptop it will not work for the two building switches, dist1-b1 nor dist1-b2 (why?).

```
$ telnet <DEVICE-NAME>.campusY.ws.nsrc.org
```

```
username: nmmlab
password: \<GIVEN IN CLASS\>
```

Display information about your network device

```
<device-name>.campusy> enable
Password:                                (password given in
class)
<device-name>.campusy# show run          (space to continue -
see note below)
<device-name>.campusy# show ip int brief
<device-name>.campusy# show int gi0/0    (or any other
interface that is up)
<device-name>.campusy# show ?            (lists all options)
```

Note: Press “q” to exit from information screen before reaching the end if you wish, otherwise press the <SPACE BAR> to move scroll through the information until the end.

Configure your router or switch to only use SSH

These steps will do the following:

- Create an ssh key for your router
- Create an encrypted password for the user nmmlab
- Encrypt the enable password
- Turn off telnet (unencrypted) access to your router
- Turn on SSH (version 2) access to your router

You should be connected to your router or switch and at the enable prompt. The prompt will look something like:

```
<device-name>.campusy#
```

At this prompt do the following:

```
<device-name>.campusy# configure terminal
(or "conf t" for short)
<device-name>.campusy(config)# aaa new-model
<device-name>.campusy(config)# ip domain-name ws.nsrc.org
<device-name>.campusy(config)# crypto key generate rsa
```

```
    How many bits in the modulus [512]: 2048
```

Wait for the key to generate. This is the “host key”, which the router needs to participate in SSH, and will be remembered by your SSH client when it connects.

Now we’ll tell our router to only allow SSH connections on the 5 defined consoles (vty 0 through 4):

```
<device-name>.campusy(config)# line vty 0 4
<device-name>.campusy(config-line)# transport input ssh
<device-name>.campusy(config-line)# exit
```

This drops us out of the “line” configuration mode and back in to the general configuration mode. Now we’ll tell the router to log SSH-related events and to only allow SSH version 2 connections:

```
<device-name>.campusy(config)# ip ssh logging events
<device-name>.campusy(config)# ip ssh version 2
```

Now exit from configuration mode:

```
<device-name>.campusy(config)# exit
```

And, write these changes to the routers permament configuration:

```
<device-name>.campusy# write memory                    (or "wr
mem" for short)
Overwrite the previous NVRAM configuration?[confirm]   (just hit
<ENTER>)
```

Ok. That’s it. You can no longer use telnet to connect to your router. You must connect using SSH.

Naturally in a real-world situation you would use much more secure passwords than those we’ve used in the class - or better, a centralized authentication service like TACACS or RADIUS. But we’re not going to deploy that here.

Before you exit your Telnet session be sure to test ssh connectivity from another PC in your group (or, open another terminal window). Do this in case you made a mistake to avoid locking yourself out of your router.

First, try connection again with telnet from your virtual machine:

```
$ telnet <DEVICE-NAME>.campusY.ws.nsrc.org
```

What happens? You should see something like:

```
Trying 100.68.1.2...  (for example only)
telnet: Unable to connect to remote host: Connection refused
```

Now try connecting with SSH:

```
$ ssh nmmlab@<DEVICE-NAME>.campusY.ws.nsrc.org
```

If you receive an error while trying to connect see the *Troubleshooting* Section below.

If successful, you should see something looks similar to this:

```
The authenticity of host 'core1.campus1.ws.nsrc.org
(2001:db8:1:0::2)' can't be established.
RSA key fingerprint is
SHA256:pqcy5VRxckdnf/X3IcO4LY83wzeh231NUXA8Qb+74b8.
This key is not known by any other names
Are you sure you want to continue connecting
(yes/no/[fingerprint])?
```

Enter “yes” (in full) and press ENTER to continue.

Now you’ll see the following:

```
Warning: Permanently added 'dist1-
b2.campus5.ws.nsrc.org,2001:db8:1:0::2' (RSA) to the list of
known hosts.

Network Monitoring and Management Workshop Lab
       Network Startup Resource Center
(nmmlab@dist1-b2.campus5) Password:
```

Enter the <CLASS PASSWORD>

You will end up on a prompt like:

```
<device-name>.campusY>
```

Type “enable” to allow us to execute privileged commands:

```
<device-name>.campusY> enable
Password: <CLASS ENABLE PASSWORD>
<device-name>.campusy#
```

Now let’s view the current router configuration:

```
<device-name>.campusy# show running-config
(sh run)
```

Press the space bar to continue. Note some of the entries like:

```
enable secret 9
$9$XD8eRlyviVdqud$eEnnKdtrI.QAMxW3bOcMEnUVMpvMZb.cXhxCP7oV8eI
.
.
.
username nmmlab secret 9
$9$oQQS.9RWSoFrNt$9FGCukqvA5G2CluUisf6uwQu3I4OGgK5pp55OUELOWc
.
.           (lots of lines down)
.
line vty 0 4
 exec-timeout 0 0
 transport preferred none
 transport input ssh
```

You can see that both the enable password and the password for the user *nmmlab* have been hashed (https://community.cisco.com/t5/networking-knowledge-base/understanding-the-differences-between-the-cisco-password-secret/ta-p/3163238) . (This is what “service password-encryption” does, and is generally a good thing).

Now you should exit the router interface to complete this exercise:

```
<device-name>.campusy# exit
```

And, if you still have your older Telnet session in another window running be sure to exit from that as well.

Troubleshooting

“no matching key exchange method found” or “no matching host key type found”

If you attempted to log in and received a message like this:

> *Unable to negotiate with 100.68.6.1 port 22: no matching key exchange method found. Their offer: diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1*

The version of software on your network device is using older, weaker encryption ciphers, which are available in newer SSH clients but disabled for security reasons.

You can force ssh to accept an older, less secure key exchange algorithm by running the ssh command like this:

```
$ ssh -oKexAlgorithms=+diffie-hellman-group14-sha1
nmmlab@<device-name>.campusY.ws.nsrc.org
```

Now you may get another error:

> *Unable to negotiate with 100.68.6.1 port 22: no matching host key type found. Their offer: ssh-rsa*

So we need to add another option:

```
ssh -oKexAlgorithms=+diffie-hellman-group14-sha1 -
oHostKeyAlgorithms=+ssh-rsa nmmlab@<device-
name>.campusY.ws.nsrc.org
```

But this is very painful!

Let's update your virtual machine's system-wide ssh client configuration to allow this older key exchange.

Create a new file, as root:

```
$ sudo editor /etc/ssh/ssh_config.d/cisco.conf
```

Inside this file, paste the following:

```
Host *.ws.nsrc.org
KexAlgorithms +diffie-hellman-group-exchange-sha1,diffie-
hellman-group14-sha1
HostKeyAlgorithms +ssh-rsa
```

Save the file and exit. This tells ssh to permit the less secure crypto, but only when connecting to hosts whose name ends in `.ws.nsrc.org` .

Now try connecting to your network device again:

```
$ ssh nmmlab@<DEVICE-NAME>.campusY.ws.nsrc.org
```

(Note: in older versions of Linux the `ssh_config.d` directory may not be available, in which case you have to add your new settings to the end of `/etc/ssh/ssh_config` instead)

"no matching cipher found"

With some other versions of Cisco IOS you may get another error:

Unable to negotiate with 172.26.10.2 port 22: no matching cipher found. Their offer: aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc

Again, this is because the network device is using older ciphers which are not enabled by default in newer versions of OpenSSH.

You can add `-oCiphers=+aes256-cbc` to the ssh command line, or you can add another line to your `/etc/ssh/ssh_config.d/cisco.conf` file:

```
Host *.ws.nsrc.org
...
Ciphers +aes256-cbc
```

NOTES

1. If you are locked out of your router or switch after this exercise let your instructor know and they can reset your network device's configuration back to its original state.
2. Please only do this exercise once. If multiple people do this exercise it's very likely that access to the router or switch will be broken.
3. During the week you will configure items such as SNMP, Netflow and more on your local network devices. From now on you can simply connect to the device directly from your laptop or desktop machine using SSH.

Exercises - Part 2: NTP Configuration

Configure NTP and Timezone

Perhaps you can select another person in your group to execute the following steps to allow them to practice.

Your first step is to connect to your router:

```
$ ssh nmmlab@<DEVICE-NAME>.campusY.ws.nsrc.org
```

Now we will enable the Network Time Protocol so that we can synchronize your router's time with your PCs time so that all devices on our local network will have the same time. To do this follow these steps:

```
<device-name>.campusY> enable                          (en)
Password:
<device-name>.campusY# configure terminal              (conf t)
<device-name>.campusY(config)# ip name-server 100.64.0.1
<device-name>.campusY(config)# ip domain-lookup
<device-name>.campusY(config)# ntp server ntp.ws.nsrc.org
<device-name>.campusY(config)# no clock timezone
```

```
<device-name>.campusY(config)# exit
<device-name>.campusY# write memory                    (wr mem)
```

This uses the classroom NTP time server (if you don't have one, then `pool.ntp.org` is a reasonable alternative). This also indicates that you wish to use UTC time (same as GMT time) for this router.

To verify NTP status, NTP server associations and the reported time on your router:

```
<device-name>.campusY# show ntp status              (sh ntp stat)
```

After some time you will see something like (you may see "unsynchronized" for a while):

```
bdr1.campus6#sh ntp status
Clock is synchronized, stratum 4, reference is 100.64.0.1
nominal freq is 1000.0003 Hz, actual freq is 999.5733 Hz,
precision is 2**19
ntp uptime is 344600 (1/100 of seconds), resolution is 1001
reference time is E6DEFA9B.B49F3848 (16:53:47.705 UTC Wed Sep 28
2022)
clock offset is -44.9735 msec, root delay is 6.90 msec
root dispersion is 94.79 msec, peer dispersion is 1.83 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is
0.000426985 s/s
system poll interval is 128, last update was 41 sec ago.
```

… and to see the NTP server associations:

```
<device-name>.campusY# show ntp associations   (sh ntp assoc)
```

```
  address         ref clock       st   when   poll reach  delay
offset   disp
*~100.64.0.1      10.12.255.11     3     68    128   377  0.925
-44.973  1.836
 * sys.peer, # selected, + candidate, - outlyer, x falseticker,
~ configured
```

… and, finally, to see your router's current time:

```
rtrN# show clock                    (sh clo)
```

You should see something like:

```
*15:59:24.609 UTC Wed Sep 28 2022
```

Now you can exit from your router:

```
<device-name>.campusY# exit
```

Make sure your team finishes working on the other devices in your group. If anyone has problems connecting to a device see the *Troubleshooting* section above.