

LAB – TELNET TO A NETWORK DEVICE**OBJECTIVE**

In this lab, you will create a **pexpect** session object connecting via telnet to a network device with error checking for each step and then disconnect, showing the success or failure of the ability to connect via telnet.

PART 1

Copy the required configuration for this lab from flash into the running-configuration of the CSR1000v router.

STEP 1: COPY CONFIGURATION

In the CSR1kv router console, enter **privileged exec** mode and copy the additional configuration located in the **CONFIG3** file to the running-configuration using the commands:

```
CSR1kv> enable
CSR1kv# copy flash:CONFIG3 running-config
```

```
CSR1kv>enable
CSR1kv#copy flash:CONFIG3 running-config
Destination filename [running-config]?
75 bytes copied in 0.009 secs (8333 bytes/sec)
CSR1kv#
```

STEP 2: VERIFY CONFIGURATION

Verify the new running-configuration by using the command:

```
CSR1kv# show run | section line vty
```

NOTE:

To get the pipe (|) command, you may need to use the keys **Shift + #** rather than your local keyboard input.

```
CSR1kv#show run | section line vty
line vty 0 4
  logging synchronous
  login local
  transport input all
line vty 5 15
  logging synchronous
  login local
  transport input all
CSR1kv#
```

PART 2

Open a terminal and switch to the lab directory

STEP 1: OPEN A TERMINAL WINDOW

Double-click the Terminal icon on the desktop to open the terminal window for use in this lab.

STEP 2: CHANGE DIRECTORY

Change to the directory **labs/prne/** in the user home directory, which holds the files for the course labs.

```
~$ cd labs/prne/
```

PART 3

Open **Visual Studio Code**, create a new file and save it with a filename of **telnet-to-a-network-device.py**. Ensuring to save the file in the **~/labs/prne/** directory.

STEP 1: IMPORT PEXPECT

Import the pexpect library.

```
# Import required modules/packages/library
import pexpect
```

STEP 2: DEFINE VARIABLES

Define variables for your IP address, username, and password and assign the following values:

- IP Address: 192.168.56.101
- Username: cisco
- Password: cisco123!

```
# Define variables
ip_address = '192.168.56.101'
username = 'cisco'
password = 'cisco123!'
```

STEP 3: CREATE SESSION

Create a session using the pexpect 'spawn' method, passing in a telnet command.

```
# Create telnet session
session = pexpect.spawn('telnet ' + ip_address, encoding='utf-8',
                        timeout=20)
result = session.expect(['Username:', pexpect.TIMEOUT])
```

STEP 4: CHECK FOR ERRORS

Check for error in connecting session and display any errors in terminal.

```
# Check for error, if exists then display error and exit
if result != 0:
    print('--- FAILURE! creating session for: ', ip_address)
    exit()
```

STEP 5: SEND CREDENTIALS

Send the user credentials to the device and check for any errors and display any in terminal.

```
# Session is expecting username, enter details
session.sendline(username)
result = session.expect(['Password:', pexpect.TIMEOUT])

# Check for error, if exists then display error and exit
if result != 0:
    print('--- FAILURE! entering username: ', username)
    exit()

# Session is expecting password, enter details
session.sendline(password)
result = session.expect(['#', pexpect.TIMEOUT])

# Check for error, if exists then display error and exit
if result != 0:
    print('--- FAILURE! entering password: ', password)
    exit()
```

NOTE:

As the user account being used has a privilege level of 15, after entering the password the expected mode is the privileged exec mode (#), if you were to use an account with default privilege levels the expected mode would be the user exec mode (>).

STEP 6: DISPLAY MESSAGE IF SUCCESSFUL

Display in terminal a success message if the login completes successfully.

```
# Display a success message if it works
print('-----')
print('')
print('--- Success! connecting to: ', ip_address)
print('--- Username: ', username)
print('--- Password: ', password)
print('')
print('-----')
```

STEP 7: CLOSE SESSION

Close the telnet session.

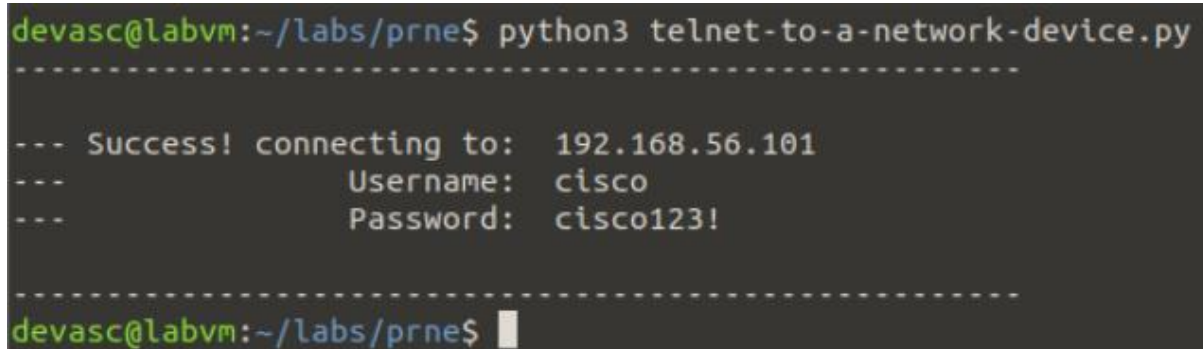
```
# Terminate telnet to device and close session
session.sendline('quit')
session.close()
```

STEP 8: SAVE, RUN AND VERIFY APPLICATION

Save you application and then run it from the terminal rather than from within visual studio code.

```
~/labs/prne$ python3 telnet-to-a-network-device.py
```

Verify that the telnet connection is successful and the output is comparable to below.



```
devasc@labvm:~/labs/prne$ python3 telnet-to-a-network-device.py
-----
--- Success! connecting to: 192.168.56.101
--- Username: cisco
--- Password: cisco123!
-----
devasc@labvm:~/labs/prne$
```

PART 4 (OPTIONAL BUT HIGHLY RECOMMENDED)

As this lab is completed in NETLAB+ and your code files will be erased when the reservation ends, it is advisable to save your files in GitHub under your repository for this course.