

Topic 4 – Netmiko

Author: Jake Palczewski, CCIE #63320

Last Updated: December 6, 2020

Table of Contents

Introduction.....	2
Installing Netmiko	2
Using Netmiko	2
Conclusion	5

Introduction

As a network engineer, I find myself needing to SSH into routers, switches, load balancers, and firewalls to run show commands and push config changes. Instead of using a program like PuTTY or SecureCRT, we can use Python to SSH into most network equipment, allowing us to run commands and push config! Python has a few external modules that allow this capability, including Netmiko. This lesson will impart how to use Netmiko to run a show command on a Nexus switch.

Installing Netmiko

At the end of Lesson 4, we used pip to pull and install Netmiko into our Python library. We also called the Netmiko module into our script. Here is a recap:

Using elevated mode on Powershell:

```
PS H:\>  
PS H:\> pip install netmiko
```

In the Python script:

```
2 from netmiko import ConnectHandler
```

Using Netmiko

In Schwab's lab, we have switch rlf14lab. Using PuTTY to log into this switch, we are immediately prompted for our username and password. Since we will be using Python to SSH into rlf14lab, we need to feed Python our credentials.

One way in which to do this is to hard-code your credentials into Python as variables, like so:

```
6 from netmiko import ConnectHandler  
7  
8 username = "jake.palczewski"  
9 password = REDACTED
```

However, to avoid storing our username and password in plain text, we can use the `input()` function to prompt us for a username, and use the `getpass` module to prompt us for a password. We can interact with our Python script via the terminal!:

```
6  from netmiko import ConnectHandler
7  from getpass import *
8
9  username = input("What is your username?: ")
10 password = getpass("What is your password?: ")
11
12 print("Thanks!")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
What is your username?: jake.palczewski
What is your password?:
Thanks!
```

Let's break this down:

- We use a **from/import** statement to install getpass. Since it is part of Python's Standard Library, we do not need to use git to import it
- The **username** variable uses the **input** function to ask for our username. The string, "What is your username?: ", is presented with the prompt.
- Once our username is typed in via the terminal and we press Enter, the **username** variable becomes equal to the value we inputted.
- The **password** variable uses the **getpass** function from the getpass module to discreetly ask for and store our password. As we type in our password for this prompt, we'll notice our password does not show in the terminal.
- Once our password is typed in via the terminal and we press Enter, the **password** variable becomes equal to the value we inputted.

Now, let's configure the Netmiko **ConnectHandler**:

```

6  from netmiko import ConnectHandler
7  from getpass import *
8
9  username = input("What is your username?: ")
10 password = getpass("What is your password?: ")
11
12 device_handler = ConnectHandler(
13     device_type = 'cisco_nxos',
14     username = username,
15     password = password,
16     ip = "NXswitch01"
17 )
18
19 output = device_handler.send_command("show run | include hostname")
20 print(output)

```

Let's break this down:

- The **device_handler** variable encapsulates the **ConnectHandler** function, which was **imported** from Netmiko.
- The **ConnectHandler** function allows us to define the **device_type** into which we are SSHing, the username we will use, the password we will use, and the destination IP of the device.
 - **device_type**: Netmiko supports most device types. In our case, rlf14lab is a Nexus switch, so we want to use the "cisco_nxos" device_type. The device_type value is assigned to device_type as a **string**.
 - A link pointing to a list of supported device types can be found at the bottom of this document
 - **username**: we can simply re-use the **username** variable defined earlier in the script, since it contains our inputted username
 - **password**: we can simply re-use the **password** variable defined earlier in the script, since it contains our inputted password
 - **ip**: since rlf14lab resolves to an IP address via DNS, we can use either the IP address of the switch or the hostname, in **string** format.
- The **output** variable encapsulates the **send_command** method, which leverages the **device_handler** variable. This method allows us to send a command to rlf14lab. In this case, we are sending the command "show run | include hostname". This variable is multi-functional: it both sends the command to the switch, and stores the output of the command!
- Lastly, we are using the **print** function to check the value of the **output** variable, which should be the result of the "show run | include hostname" command.

Let's run the script and see what happens!

```
6 from netmiko import ConnectHandler
7 from getpass import *
8
9 username = input("What is your username?: ")
10 password = getpass("What is your password?: ")
11
12 device_handler = ConnectHandler(
13     device_type = 'cisco_nxos',
14     username = username,
15     password = password,
16     ip = "NXswitch01"
17 )
18
19 output = device_handler.send_command("show run | include hostname")
20 print(output)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
What is your username?: jake.palczewski
What is your password?:
hostname NXswitch01
```

Looks like "hostname NXswitch01" was **printed** to the terminal.

Conclusion

As shown in this lesson, Python and Netmiko can be a powerful ally at a network engineer. Netmiko is a robust module that will be explored throughout future lessons. For now, here's some helpful links:

- [Netmiko example, including commonly-used methods](#)
- [Netmiko's supported platforms](#)