Netmiko and Python

“The fool doth think he is wise, but the wise man knows himself to be a fool.”
— Shakespeare, As You Like It

$ whoami

Kirk Byers
Network Engineer:
CCIE #6243 (emeritus)

Programmer:
Netmiko
NAPALM

Teach Python and Ansible Courses
SF Network Automation Meetup

What is Netmiko?

Paramiko is the standard Python SSH library.

Netmiko is a multi-vendor networking library based on Paramiko.

Netmiko Vendors

Regularly tested
Arista vEOS
Cisco ASA
Cisco IOS
Cisco IOS-XE
Cisco IOS-XR
Cisco NX-OS
Cisco SG300
HP Comware7
HP ProCurve
Juniper Junos
Linux

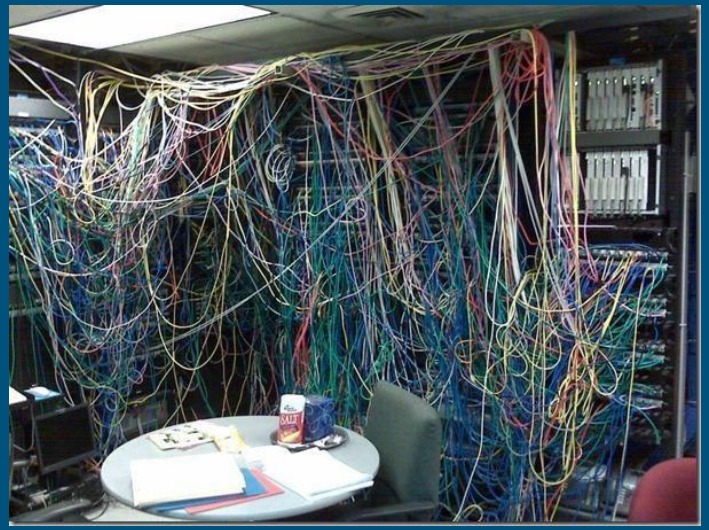
Limited testing
Alcatel AOS6/AOS8
Avaya ERS
Avaya VSP
Brocade VDX
Brocade MLX/NetIron
Calix B6
Cisco WLC
Dell-Force10
Dell PowerConnect

Limited testing
Huawei
Mellanox
NetApp cDOT
Palo Alto PAN-OS
Pluribus
Ruckus ICX/FastIron
Ubiquity EdgeSwitch
Vyatta VyOS

Experimental
A10
Accedian
Aruba
Ciena SAOS
Cisco Telepresence
CheckPoint GAiA
Coriant
Eltex
Enterasys
Extreme EXOS
Extreme Wing
F5 LTM
Fortinet
MRV OptiSwitch
Nokia SR-OS
QuantaMesh

General Notes for Tonight

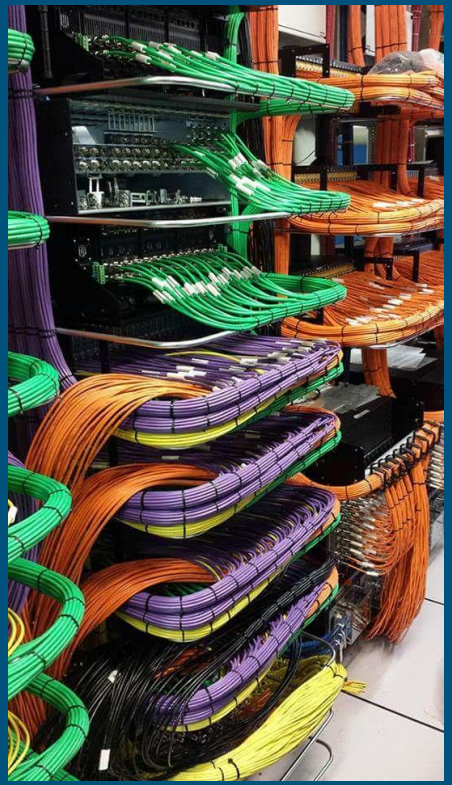
- Lots of examples, reference code.
- Example code is posted here: https://github.com/ktbyers/pynet/tree/master/presentations/dfwcug/examples
- Code is running on a Linux box (AWS), running Netmiko 2.1.1 connecting to either physical or virtual devices.



Before Netmiko.

General Notes for Tonight

- Using Python3.6, but should be very similar in Python2.7.
- I will just assume you know some amount of Python. I will provide some Python resources at the end of the presentation.
- Coordination for questions.



After Netmiko.

Installing Netmiko

> “What’s in a name? That which we call a package manager. By any other name would smell as sweet.”
>
> — Romeo and Juliet

pip install netmiko

Use a virtual environment

MacOS - Use homebrew and a virtual environment.

Newer versions of Paramiko should be fairly easy to install on Windows (install python, pip install netmiko).

A simple example (case1)

```
#!/usr/bin/env python
from netmiko import Netmiko
from getpass import getpass

net_connect = Netmiko('cisco1.twb-tech.com', username='pyclass',
                      password=getpass(), device_type='cisco_ios')

print(net_connect.find_prompt())
net_connect.disconnect()
```

Give every man thy ear but few thy voice.

— Hamlet

Expanding on simple example (case2)

```
#!/usr/bin/env python
from netmiko import Netmiko
from getpass import getpass

cisco1 = {
    'host': 'cisco1.twb-tech.com',
    'username': 'pyclass',
    'password': getpass(),
    'device_type': 'cisco_ios',
}

net_connect = Netmiko(**cisco1)
print(net_connect.find_prompt())
net_connect.disconnect()
```

And though Python be but little, she is fierce.

- A Midsummer Night's Dream

What if I don’t know the device_type?

Just use an invalid device_type.

How do I get into enable mode?

Add ‘secret’ argument and call
.enable() method.

Ambition should be made of sterner stuff.
- Julius Caesar

Connecting to multiple devices (case3)

```
#!/usr/bin/env python
from netmiko import Netmiko
from getpass import getpass
password = getpass()

cisco1 = { … }
cisco2 = { … }
nxos1 = { … }
srx1 = { … }

for device in (cisco1, cisco2, nxos1, srx1):
    net_connect = Netmiko(**device)
    print(net_connect.find_prompt())
```

For when your firewall changes go wrong.

Executing show commands (case4)

- Send_command
  - Automatically strips command echo and trailing router prompt.
- Adding the expect_string argument
- Increasing the time allocated for send_command to complete.
  - delay_factor=2
  - max_loops=1000

"But if it be a sin to covet honour, I am the most offending soul alive."
— Henry V

TextFSM Integration

- Must have ntc-templates installed (available on GitHub)
  - Needs to be installed in ~/ntc-templates/templates/
  - Or set the NET_TEXTFSM environment variable

```
export NET_TEXTFSM=/path/to/ntc-templates/templates/
```

- Add use_textfsm=True argument to send_command()

Handling additional prompts (case5)

- Some commands ask us for additional confirmation.
- Use send_command_timing() or expect_string argument.

“He speaks an infinite deal of nothing, more than any man in all Venice. His reasons are as two grains of wheat hid in two bushels of chaff.”
- Merchant of Venice

Making config changes (case6)

- Use send_config_set() or send_config_from_file().
- send_config_set() takes a list of commands or a single command string.
- Automatically handles entering/exiting config mode.
- The configuration is not saved, use the save_config() method.

Making config changes and commit (case7)

- With juniper_junos and IOS-XR you can call a commit() method.
- There are extra arguments in this method to handle special cases including platform specific situations (commit confirm, commit comments).

"No legacy is as rich as honesty"
— All's Well that Ends Well

Auto-detecting the device_type (case8)

- SSH auto-detection.

```
-------
guesser = SSHDetect(**device)
best_match = guesser.autodetect()
print(best_match)
print(guesser.potential_matches)
```

"He never went out without a book under his arm, and he often returned with two."
- Victor Hugo, Les Miserables

- SNMP auto-detection.

```
-------
snmp_key = getpass("Enter SNMP community: ")
my_snmp = SNMPDetect("cisco1.twb-tech.com", snmp_version="v3", user='pysnmp',
                     auth_key=snmp_key, encrypt_key=snmp_key, auth_proto="sha",
                     encrypt_proto="aes128")
device_type = my_snmp.autodetect()
print(device_type)
```

Using SSH keys (case9)

```
key_file = "/home/gituser/.ssh/test_rsa"

cisco1 = {
    'device_type': 'cisco_ios',
    'host': 'cisco1.twb-tech.com',
    'username': 'testuser',
    'use_keys': True,
    'key_file': key_file,
}

net_connect = Netmiko(**cisco1)
print(net_connect.find_prompt())
```

"to learn to read is to light a fire; every spelled syllable sparkles."
- Victor Hugo, Les Miserables

SSH Proxy Configuration (case10)

```
$ cat ssh_config
---
host jumphost
  IdentityFile ~/.ssh/test_rsa
  user gituser
  hostname 10.10.72.159

host * !jumphost
  ProxyCommand ssh jumphost nc %h %p
```

```
key_file = "/home/gituser/.ssh/test_rsa"
cisco1 = {
    'device_type': 'cisco_ios',
    'host': 'cisco1.twb-tech.com',
    'username': 'testuser',
    'use_keys': True,
    'key_file': key_file,
    'ssh_config_file': './ssh_config',
}

net_connect = Netmiko(**cisco1)
print(net_connect.find_prompt())
```

Troubleshooting/Debugging (case11)

```
Add logging support
-----------
import logging
logging.basicConfig(filename='test.log', level=logging.DEBUG)
logger = logging.getLogger("netmiko")


Manual read/write of channel
------------
net_connect.write_channel("show ip int brief\n")
time.sleep(1)
output = net_connect.read_channel()
```

A man is not idle because he is absorbed in thought. There is visible labor and there is invisible labor.
— Victor Hugo, Les Misérables.

Using telnet (case12)

```
#!/usr/bin/env python
from netmiko import Netmiko
from getpass import getpass

cisco1 = {
    'host': 'cisco1.twb-tech.com',
    'username': 'pyclass',
    'password': getpass(),
    'device_type': 'cisco_ios_telnet',
}

net_connect = Netmiko(**cisco1)
print(net_connect.send_command("show ip arp"))
net_connect.disconnect()
```

Some rise by sin, and some by virtue fall
— Measure for Measure

Using a terminal server and redispatch (case13)

General Process:

1. Connect to the terminal server, use the ‘terminal_server’ device_type.
2. Manually handle terminal server interaction using write_channel and read_channel.
3. Connect to end_device.
4. Manually handle username/password authentication.
5. Post login, call redispatch to reset the netmiko class to proper class.

Using Secure Copy (case14)

```
cisco = { … }
source_file = 'test1.txt'
dest_file = 'test1.txt'
direction = 'put'
file_system = 'flash:'

ssh_conn = ConnectHandler(**cisco)
transfer_dict = file_transfer(ssh_conn,
                              source_file=source_file,
                              dest_file=dest_file,
                              file_system=file_system,
                              direction=direction,
                              overwrite_file=True)
```

Netmiko Tools (case15)

git clone https://github.com/ktbyers/netmiko_tools

In your .bashrc file if you want to retain it
export PATH=~/netmiko_tools/netmiko_tools:$PATH

~/.netmiko.yml

netmiko-grep
netmiko-show
netmiko-cfg

Netmiko Tools (case15)

Automatically uses threading for concurrency to devices.

Creates a directory to store information at ~/.netmiko/tmp

Should have way to pass command-line username and password in a couple of weeks.

netmiko-grep

Pattern search through running-config of devices.

```
$ netmiko-grep --list-devices

# Search for logging string in the cisco group
$ netmiko-grep 'logging' cisco

# Search for Vlan string in the nxos group
netmiko-grep 'Vlan' nxos

$ netmiko-grep 'Vlan' nxos --use-cache
```

netmiko-show

Execute arbitrary show commands on devices

```
# Execute show ip int brief on the cisco group
$ netmiko-show --cmd "show ip int brief" cisco

# Execute show ip arp on the nxos group
$ netmiko-show --cmd "show ip arp" nxos

# Execute wr mem on the cisco group
$ netmiko-show --cmd "wr mem" cisco
```

netmiko-cfg

Execute configuration commands on devices

```
# Configure logging buffer on the cisco group
$ netmiko-cfg --cmd "logging buffered 5000" cisco

# Configure the VLANs specified in vlans.txt on the arista group
$ netmiko-cfg --infile vlans.txt arista

# Configure commands from standard input
$ echo 'logging buffered 10000' | netmiko-cfg --infile - cisco
```

Other Topics

Concurrency:
https://github.com/ktbyers/pynet-ons-oct17/blob/master/threads_procs/

Jinja2 Templating
http://jinja.pocoo.org/docs/2.10/templates/

Learning Python

My free Python course, next session starts May 8.
https://pynet.twb-tech.com/email-signup.html

Automate the Boring Stuff with Python
https://www.amazon.com/gp/product/1593275994/

Treading on Python Volume 1: Foundations of Python by Matt Harrison
https://www.amazon.com/Treading-Python-1-Foundations/dp/1475266413

Network Automation Resources

NAPALM
https://napalm.readthedocs.io/en/latest/

Frameworks: Ansible and Salt

Brigade: New Python Framework
https://github.com/brigade-automation/brigade

Network Programmability and Automation Book
https://www.amazon.com/Network-Programmability-Automation-Next-Generation-Engineer/dp/1491931256

Questions?

ktbyers@twb-tech.com
Twitter: @kirkbyers