Name : Shuvo Biswas
ID : IT-16014
Lab Report No : 04
Lab Report on : Python for Networking

Objectives :

The objective of the lab 04 is to:
 Install python and use third-party libraries
 Interact with network interfaces using python
 Getting information from internet using Python

Theory :

Third-party libraries: Although the Python's standard library provides a great set of awesome functionalities, there will be times that you will eventually run into the need of making use of third party libraries. Can you imagine building a webserver from scratch? Or making a port to a database driver? Or, maybe, comming up with an image manipulation tool?. Third party libraries are welcome in a way that they prevent you from reinventing the something that exit. They save you time to focus on finishing and delivering your application.

Networking Glossary: Before we begin discussing networking with any depth, we must define some common terms that you will see throughout this guide, and in other guides and documentation regarding networking.

Methodology :

Installing Python Third-party includes:
 Python Third-party includes a setup.py file, it is usually distributed as a tarball (.tar.gz or .tar.bz2 file). The instructions for installing these generally look like:
 Download the file from website.
 Extract the tarball.
 Change into the new directory that has been newly extracted.
 Run sudo python setup.py build
 Run sudo python setup.py install

We use python in networking because:

1. Easy to understand and readable language.
2. Dominating language at this point of time in Network Automation space.
3. A high level language, don't have to write a lot of blue codes to get things done.
4. Powerful enough to be used as a convenient tool for daily parsing tasks, performance management, and configuration.
5. Interpreted : we run the program straight from the source code.
6. Python program Bytecode a platforms native language we can just copy over your code to another system and it will auto-magically work! with python platform
7. Object-Oriented
8. Simple and additionally supports procedural programming
9. Extensible – easily import other code
10. Embeddable –easily place your code in non-python programs
11. Extensive libraries (i.e. reg. expressions, doc generation, CGI, ftp, web browsers, ZIP, WAV, cryptography, etc...) (wxPython, Twisted, Python Imaging library)

Code for telnet:

```
import telnetlib

import time

 password = ("s")

 tn = telnetlib.Telnet("192.168.1.10")

tn.read_until("Password: ") tn.write(password + "\n")

 tn.write("enable \n")

 tn.read_until("Password: ")

 tn.write(password + "\n")

 tn.write("conf t \n") time.sleep(1)

tn.write("interface loopback10 \n")

time.sleep(1)
```

```
tn.write("ip address 10.1.1.1 255.255.255.0 \n")
time.sleep(1) tn.write("end \n") time.sleep(1)
tn.write("exit \n") print tn.read_very_eager()
 print("\nThank You")
```

On router terminal we put the commands below:

```
SW1#conf t
 Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)#interface loopback100

SW1(config-if)#ip address 100.1.1.1 255.255.255.0
SW1(config-if)#do wr
SW1(config-if)#end
```

For Creating Vlan :

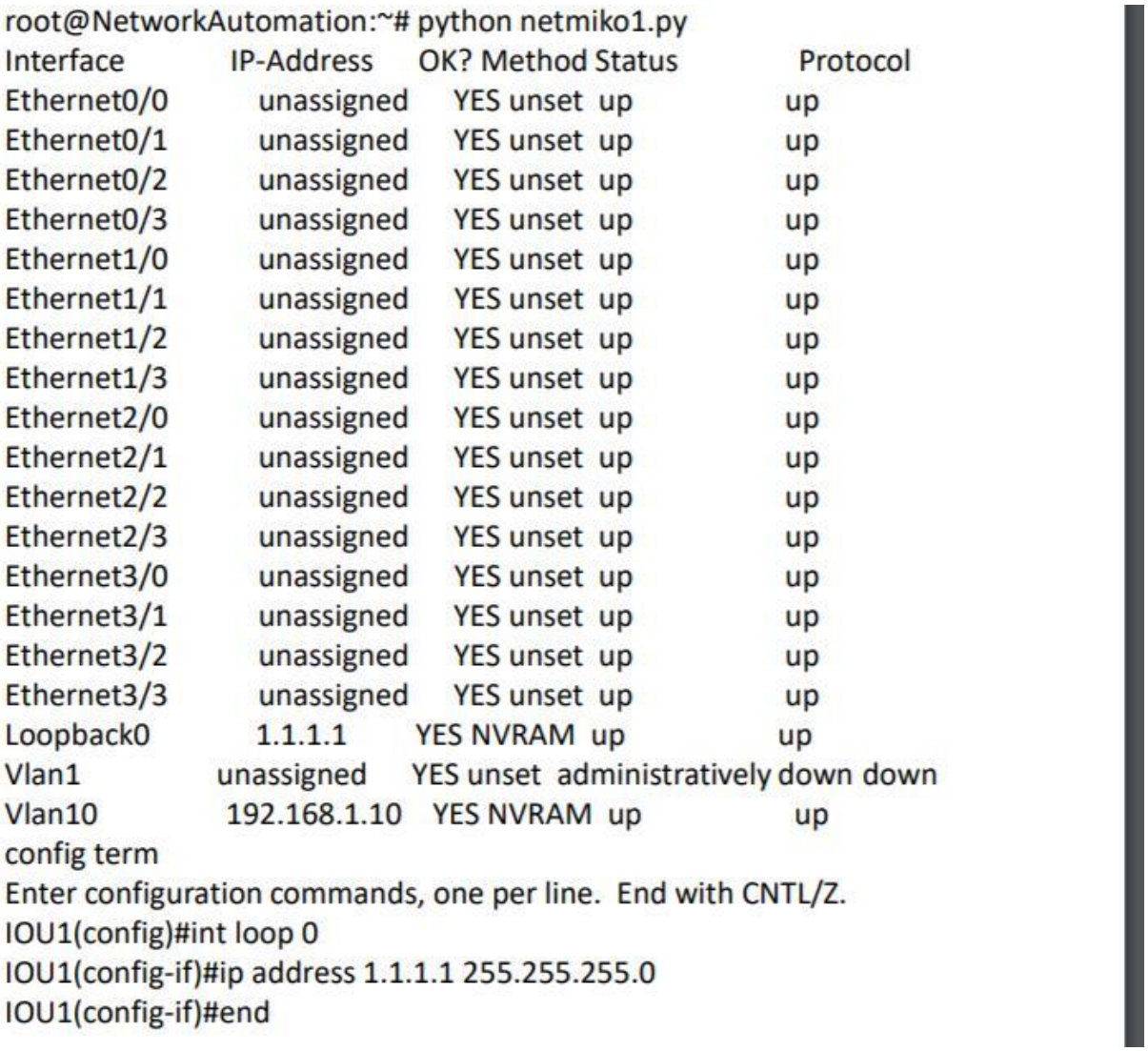
```
import telnetlib
import time
username = ("shanto") password = ("s")
tn = telnetlib.Telnet("192.168.1.10")
 tn.read_until("Username: ")
tn.write(username + " \n")
 tn.read_until("Password: ")
 tn.write(password + " \n")
tn.write("conf t \n")
 time.sleep(1) for x in range (2,10):
  {
       tn.write("vlan " + str(x) + " \n")
       time.sleep(1)
```

```
}
tn.write("name vlan_" + str(x) + " \n")
time.sleep(1) tn.write("end \n")
 time.sleep(1)
 tn.write("exit \n") print tn.read_very_eager()
print(" \nThank You")
```

In Our router we type:

Router(config)#do sh ip int br

Then the given image will appears.



```
root@NetworkAutomation:~# python netmiko1.py
Interface              IP-Address      OK? Method Status                Protocol
Ethernet0/0            unassigned      YES unset  up                    up
Ethernet0/1            unassigned      YES unset  up                    up
Ethernet0/2            unassigned      YES unset  up                    up
Ethernet0/3            unassigned      YES unset  up                    up
Ethernet1/0            unassigned      YES unset  up                    up
Ethernet1/1            unassigned      YES unset  up                    up
Ethernet1/2            unassigned      YES unset  up                    up
Ethernet1/3            unassigned      YES unset  up                    up
Ethernet2/0            unassigned      YES unset  up                    up
Ethernet2/1            unassigned      YES unset  up                    up
Ethernet2/2            unassigned      YES unset  up                    up
Ethernet2/3            unassigned      YES unset  up                    up
Ethernet3/0            unassigned      YES unset  up                    up
Ethernet3/1            unassigned      YES unset  up                    up
Ethernet3/2            unassigned      YES unset  up                    up
Ethernet3/3            unassigned      YES unset  up                    up
Loopback0              1.1.1.1         YES NVRAM  up                    up
Vlan1                  unassigned      YES unset  administratively down down
Vlan10                 192.168.1.10    YES NVRAM  up                    up
config term
Enter configuration commands, one per line.  End with CNTL/Z.
IOU1(config)#int loop 0
IOU1(config-if)#ip address 1.1.1.1 255.255.255.0
IOU1(config-if)#end
```

Conclusion:

Python allows you to build scripts to automate complex network configuration. It is the most widely used programming language for software-defined networking, and is a critical skill for new network engineers. This course teaches the very basics of network programming with Python—the theoretical building blocks that will lead to better scripts.

The successful detail-oriented candidate will have the opportunity to work on the initial builds and implementation of this multi-platform and multi-tiered system, and be involved from requirements generation, to development planning, development, implementation, and feature and bug fix prioritization. Above all Python can be get developed more and as in the programming sector as well as in the security issue like networking.