Mawlana Bhashani Science and Technology University

Santosh, Tangail-1902.

Lab Report

Department of Information and Communication Technology

Report No: 06
Report Name: Python for Networking.
Course Title: Network Planning and Design Lab.
Course Code: ICT-3208

Submitted By	Submitted To
Name: **S.M. Hazrat Ali** ID: **IT-17016** Session: 2016-17 3rd Year 2nd Semester Dept. of Information & Communication Technology, MBSTU.	**Nazrul Islam** **Assistant Professor** Dept. of Information & Communication Technology, MBSTU.

Theory: Python functions: Functions are reusable pieces of programs. They allow you to give a name to a block of statements, allowing you to run that block using the specified name anywhere in the program and any number of times. This is known as calling the function.

Local Variables: Variables declared inside a function definition are not related in any way to other variables with the same names used outside the function (variable names are local to the function). This is called the scope of the variable. All variables have the scope of the block they are declared in starting from the point of definition of the name.

The global statement: Variables defined at the top level of the program are intended global. Global variables are intended to be used in any functions or classes). Global statement allows defining global variables inside functions as well. Modules: Modules allow reusing a number of functions in other programs.

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)

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. Learn the fundamentals of the language, including objects and variables, strings, loops, and functions. Discover how to use lists, tuples, and dictionaries, and integrate specialized Python libraries and modules such as Netmiko and telnetlib. Skip ahead to concentrate on the topics of relevance to you, or watch the entire course from beginning to end to build your core skills. Instructor David Bombal doesn't make you wait before you can start automating networks. Along the way, he shows how to quickly and easily build basic functional scripts to configure routers and switches using GNS3, Cisco IOS, and Python, so that you can put your new skills to immediate use

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

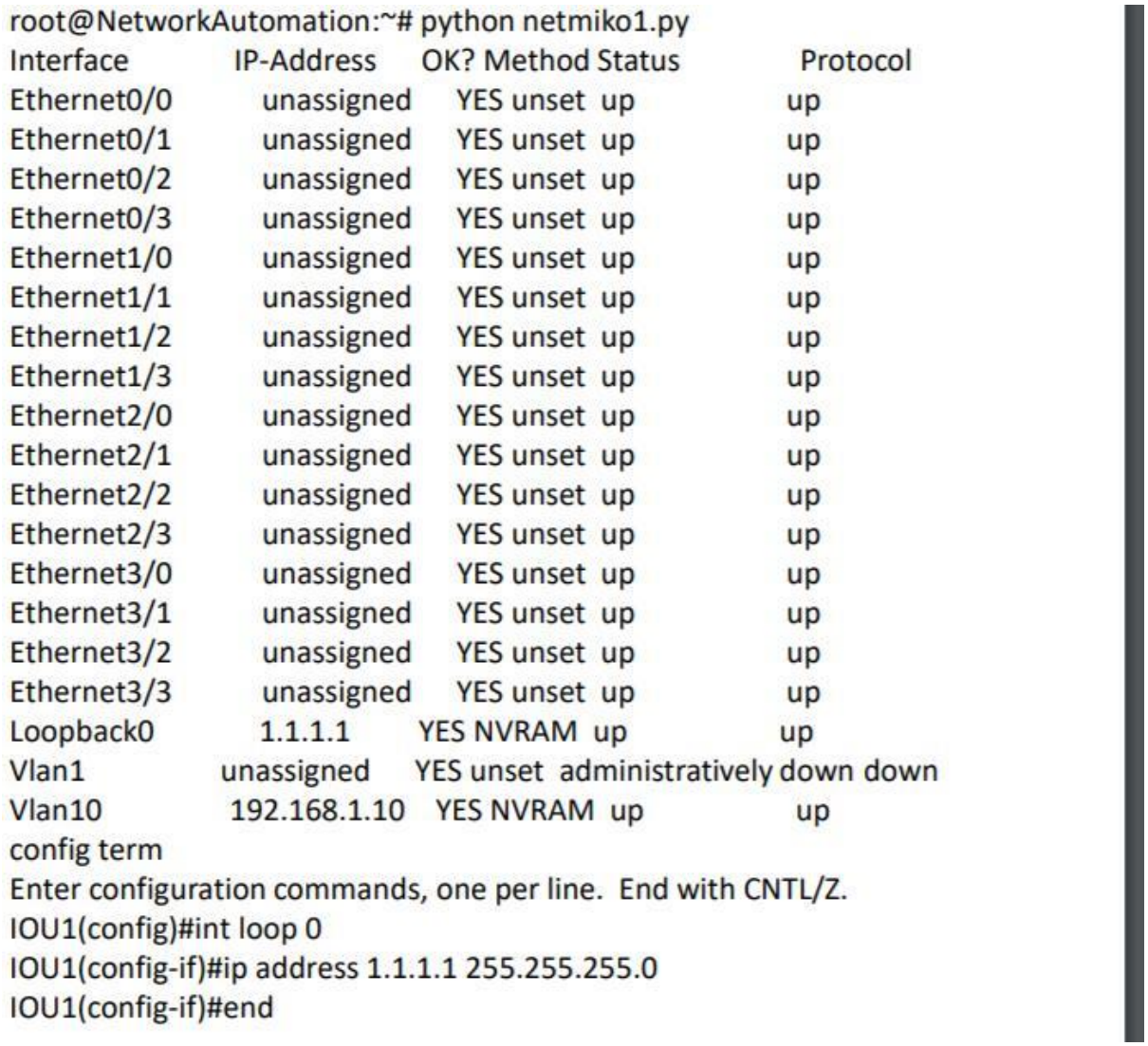
For Creating Vlan :

```
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 is used in large scale, so we can develop & customized our Network tools. Precisely as all the defence team like army use the advanced encrypted system in this case we do rely on Python. The source code of python in network prospective can be changed according to our way. 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.