Lab Report No:7

Lab Report Name:Python for Networking

Name : Binodon

ID : IT-17046

Theory: If they begin working with Python for one job or career, they can easily jump to another, even if it's in an unrelated industry. The language is used for system operations, web development, server and administrative tools, deployment, scientific modeling and much more. But, surprisingly, many developers don't pick up Python as their primary language. Because it's so easy to use and learn, they choose it as a second or third language. This may be another reason why it's so popular among developers.

Plus, it just so happens that one of the biggest tech companies in the world Google uses the language for a number of their applications. They even have a developer portal devoted to Python, with free classes offered including exercises, lecture videos and more.

In addition, the rise in the use of the Django framework for web development and a decline in popularity of PHP has also contributed to Python's success, but, ultimately, it's the perfect storm just the right amount of developer and official support, as well as demand.

Python is a very simple programming language so even if you are new to programming, you can learn python without facing any issues. Python is free to download and use. This means you can download it for free and use it in your application.

Python is a general-purpose language sometimes referred to as utilitarian which is designed to be simple to read and write. The point that it's not a complex language is important. The designers placed less of an emphasis on conventional syntax, which makes it easier to work with, even for non-programmers or developers.

Furthermore, because it's considered truly universal and used to meet various development needs, it's a language that offers a lot of options to programmers in general.

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
```

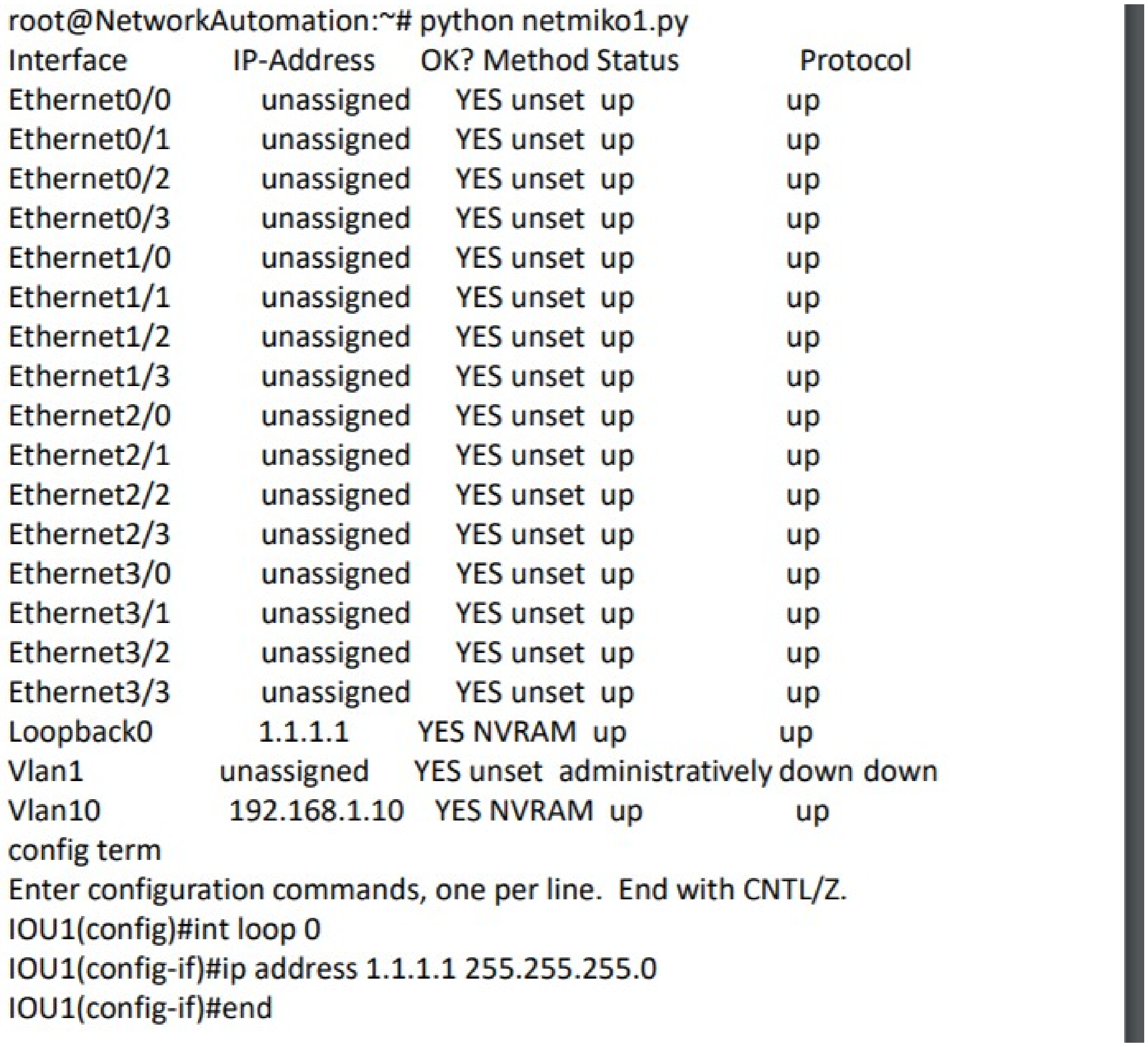
For Creating Vlan :

```
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: Network Security Team is looking for a developer to provide the building blocks for an automation system encompassing both Open Source and third party vendor solutions that will encompass all aspects of monitoring, alarming, and network security configuration for several hundred proxy servers, firewalls, caches, and load balancers. In this case 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.