// code of KnowIP.py

\_\_author\_\_ = "sanjib sinha"

import socket

ip = socket.gethostbyname("sanjibsinha.wordpress.com")

print(ip)

---------

//code of TestingSocket.py

\_\_author\_\_ = "sanjib sinha"

# reviewing the socket family

import socket

mySocket = socket.socket()

print(mySocket)

----------

//running Python code on the linux terminal

ss@ss-H81M-S1:~$ python3

Python 3.4.3 (default, Nov 28 2017, 16:41:13)

[GCC 4.8.4] on linux

Type "help", "copyright", "credits" or "license" for more information.

>>> from ipwhois import IPWhois

>>> w = IPWhois('192.0.78.13')

/usr/local/lib/python3.4/dist-packages/ipwhois/net.py:138: UserWarning:

---------

>>> w.lookup\_rdap(asn\_methods=["whois"])

--------

sudo apt-get install mercurial

----------

//Python code using pywhois

ss@ss-H81M-S1:~$ hg clone https://bitbucket.org/richardpenman/pywhois

------------

// installing pywhois

ss@ss-H81M-S1:~$ sudo pip install python-whois

----------

//Python 2.7 whois search is more powerful

ss@ss-H81M-S1:~$ python2

Python 2.7.6 (default, Nov 23 2017, 15:49:48)

[GCC 4.8.4] on linux2

Type "help", "copyright", "credits" or "license" for more information.

>>> import whois

>>> w = whois.whois('scanme.nmap.org')

>>> print w

{

"updated\_date": [

"2017-12-04 19:16:56",

"2018-02-02 14:44:01"

],

"status": [

"clientTransferProhibited https://icann.org/epp#clientTransferProhibited",

"clientTransferProhibited (https://www.icann.org/epp#clientTransferProhibited)"

],

"name": "Domain Hostmaster",

"dnssec": "unsigned",

"city": "Seattle",

"expiration\_date": [

"2024-01-18 05:00:00",

"2024-01-17 23:00:00"

],

"zipcode": "98104-2205",

"domain\_name": "NMAP.ORG",

"country": "US",

"whois\_server": "whois.fabulous.com",

"state": "WA",

"registrar": "Sea Wasp, LLC",

"referral\_url": null,

"address": "113 Cherry St #1337",

"name\_servers": [

"NS1.LINODE.COM",

"NS2.LINODE.COM",

"NS3.LINODE.COM",

"NS4.LINODE.COM",

"NS5.LINODE.COM",

"ns5.linode.com",

"ns1.linode.com",

"ns2.linode.com",

"ns3.linode.com",

"ns4.linode.com"

],

"org": "Insecure.Com LLC",

"creation\_date": [

"1999-01-18 05:00:00",

"1999-01-17 23:00:00"

],

"emails": [

"support@fabulous.com",

"hostmaster@insecure.com",

"abuse@fabulous.com"

]

}

----------------

// Python code to write a robot file

#!/usr/bin/python3

import urllib.request

import io

def GetRobots(url):

if url.endswith("/"):

path = url

else:

path = url + "/"

requestingData = urllib.request.urlopen(path + "robots.txt", data=None)

data = io.TextIOWrapper(requestingData, encoding="utf 8")

return data.read()

print(GetRobots("https://sanjibsinha.wordpress.com/"))

-----------

print(GetRobots("https://www.reddit.com"))

--------

#!/usr/bin/python3

import os

def create\_dir(directory):

if not os.path.exists(directory):

os.makedirs(directory)

def write\_file(path, data):

f = open(path, 'w')

f.write(data)

f.close()

-----------

sudo apt-get install python3-pip

-----------

hagudu@hagudu-H81M-S1:~$ sudo pip3 install tld

[sudo] password for hagudu:

Requirement already satisfied (use --upgrade to upgrade): tld in /usr/local/lib/python3.4/dist-packages

Cleaning up...

----------

#!/usr/bin/python3

from tld import get\_tld

def GetDomainName(url):

DomainName = get\_tld(url)

return DomainName

print(GetDomainName('https://google.com'))

------------

/usr/bin/python3.4 /home/hagudu/PycharmProjects/FirstPythonProject/Nmap/domain.py

google.com

Process finished with exit code 0

-----------

hagudu@hagudu-H81M-S1:~$ host google.com

google.com has address 216.58.203.174

google.com has IPv6 address 2404:6800:4009:807::200e

google.com mail is handled by 30 alt2.aspmx.l.google.com.

google.com mail is handled by 20 alt1.aspmx.l.google.com.

google.com mail is handled by 50 alt4.aspmx.l.google.com.

google.com mail is handled by 40 alt3.aspmx.l.google.com.

google.com mail is handled by 10 aspmx.l.google.com.

--------------

google.com has address 216.58.203.174

google.com has IPv6 address 2404:6800:4009:807::200e

-----------

#!/usr/bin/python3

import os

def GetIPAddress(url):

LinuxCommand = "host " + url

StartProcess = os.popen(LinuxCommand)

results = str(StartProcess.read())

marking = results.find('has address') + 12

return results[marking:].splitlines()[0]

print(GetIPAddress('google.com'))

--------------

LinuxCommand = "host " + url

--------

/usr/bin/python3.4 /home/hagudu/PycharmProjects/FirstPythonProject/Nmap/ipaddress.py

216.58.199.142

-----------------

//Python server side code

#### let the ambiguity remain ####

\_\_author\_\_ = "ss"

#### let the ambiguity remain ####

# a very simple TCP Client Server in Python that will listen to a certain port

# importing the socket and system library

import socket

import sys

# creating a socket object

mySocket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

# socket object has been created

print("Socket has successfully created")

# We need to pass an empty string so that all host interfaces are available

HOST = ''

# let us reserve a port which we can open

PORT = 8080

# let us bind our socket object that particular port

try:

mySocket.bind((HOST, PORT))

except socket.error as msg:

print('Binding has failed. Error Code is : ' + str(msg[0]) + ' Message ' + msg[1])

sys.exit()

print("Socket object is bound to the port ", PORT)

# putting the socket into listening mode

mySocket.listen(10)

print("Socket is listening")

while True:

c, addr = mySocket.accept()

print("Got connection from ", addr)

----------------

// client tries to connect to the server

telnet localhost 8080

---------

//Python code of storing the data about clients

#### let the ambiguity remain ####

\_\_author\_\_ = "ss"

#### let the ambiguity remain ####

# a very simple TCP Client Server in Python that will listen to a certain port

# importing the socket library

import socket

import sys

# creating a socket object

mySocket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

mySocket.setsockopt(socket.SOL\_SOCKET, socket.SO\_REUSEADDR, 1)

# socket object has been created

print("Socket has successfully created")

# We need to pass an empty string so that all host interfaces are available

HOST = ''

# let us reserve a port which we can open

PORT = 8080

# let us bind our socket object that particular port

try:

mySocket.bind((HOST, PORT))

except socket.error as msg:

print('Binding has failed. Error Code is : ' + str(msg[0]) + ' Message ' + msg[1])

sys.exit()

print("Socket object binding is complete to the port ", PORT)

# putting the socket into listening mode

mySocket.listen(5)

print("Socket is listening")

c, addr = mySocket.accept()

data = c.recv(512)

if data:

file = open("store.dat", "+w")

print("Connection from address : ", addr[0])

file.write(addr[0])

file.write(" : ")

file.write(data.decode("utf-8"))

file.close()

mySocket.close()

----------------

sudo apt-get update

sudo apt-get install python-pcapy

----------

#!/usr/bin/python

import pcapy

devices = pcapy.findalldevs()

print(devices)

# what we are going to do is we will open the device first

# that is our first parameter

# next we will capture the byte per packet and that will be the second

# parameter

# in the third parameter we decline promiscuous mode and the fourth one is a timeout in

# milliseconds

packets = pcapy.open\_live("eth0", 1024, False, 100)

dumper = packets.dump\_open("storage.pcap")

count = 1

while count:

try:

packet = packets.next()

except:

continue

else:

print packet

count = count + 1

if count == 10:

break

----------------

//code of getting the devices using Python

#!/usr/bin/python

import pcapy

devices = pcapy.findalldevs()

print(devices)

------------

//output of running the Python code on terminal

ss@ss-H81M-S1:~$ sudo ./raw.py

-------------

nmap

--------

sudo apt-get install nmap

--------

sudo nano test.py

---------

udo pip install python-nmap

----------

#!/usr/bin/python

import nmap

nm = nmap.PortScannerAsync()

def callback\_result(host, scan\_result):

print ('------------------')

print (host, scan\_result)

nm.scan('127.0.0.1', arguments="-O -v", callback=callback\_result)

while nm.still\_scanning():

print("Waiting >>>")

nm.wait(2)

nm1 = nmap.PortScanner()

a = nm1.nmap\_version()

print (a)

------------

nmap -F 91.198.174.192

-----------

hagudu@hagudu-H81M-S1:~$ nmap -F 45.33.49.119

-----------

#!/usr/bin/python3

import os

def GetNMAP(options, ip):

command = "nmap " + options + " " + ip

process = os.popen(command)

results = str(process.read())

return results

print(GetNMAP('-F', '54.186.250.79'))

---------------

#!/usr/bin/python

import nmap

nm = nmap.PortScanner()

print (nm.nmap\_version())

nm.scan('x.x.xx.xxx', '1-1024', '-v')

print(nm.scaninfo())

print(nm.csv())

------------

hagudu@hagudu-H81M-S1:~$ ./test.py

-----------------

#!/usr/bin/python

import nmap

nm = nmap.PortScanner()

print (nm.nmap\_version())

nm.scan('192.168.146.1', '1-1024', '-v')

print(nm.scaninfo())

print(nm.csv())

--------------

#!/usr/bin/python

import nmap

nm = nmap.PortScanner()

print (nm.nmap\_version())

nm.scan('192.168.146.1', '22-455', '-v --version-all')

print(nm.all\_hosts())

--------------

#!/usr/bin/python

import nmap

nm = nmap.PortScanner()

print (nm.nmap\_version())

nm.scan('x.x.xx.xxx', '22-455', '-v --version-all')

print(nm.all\_hosts())

---------------

#!/usr/bin/python

import nmap

nm = nmap.PortScanner()

print (nm.nmap\_version())

nm.scan('192.168.146.1', '22-1024', '-v --version-all')

print (nm.scanstats())

print (nm['192.168.146.1'].state())

print (nm['192.168.146.1'].all\_protocols())

print (nm['192.168.146.1']['tcp'].keys())

--------------