//code for knowing virtualenv version

virtualenv –version

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

//code for installing virtualenv

$ sudo apt-get install python-virtualenv

$ sudo easy\_install virtualenv

$ sudo pip install virtualenv

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

//code to create a directory for virtualenv

mkdir ~/virtual

-----------

//now we are ready to go with virtualenv

root@kali:~# virtualenv ~/virtual/myServer

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

// now python is ready to work in virtual directory

root@kali:~# cd ~/virtual/myServer/

root@kali:~/virtual/myServer# ls

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

//code of installing python-whois in the virtual directory

root@kali:~/virtual/myServer# cd bin/

root@kali:~/virtual/myServer/bin# pip install python-whois

----------

//testing the whois package in virtual directory

>>> 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", ...

...

//testing the whois package in virtual directory

>>> 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", ...

...

// the output is a shortened version for the sake of brevity.

-----------

//code for an echo server

#!/usr/bin/python2.7

# silence, listen to mother earth

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

# importing socket module

import socket

#creating a socket object

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

# let us bind this socket object to an IP address and port

HOST = ""

PORT = 2323

mySock.bind((HOST, PORT))

# we have kept the IP address argument empty string so that it could take

# any value

mySock.listen(2)

print("I am waiting for a client...")

(client, (ip, sock)) = mySock.accept()

print("Recieved connection from ", ip)

print ("Starting our ECHO Server object that will echo to the client")

# let us build some false data

data = "false"

while len(data):

data = client.recv(2048)

print ("Client sent this data : ", data)

client.send(data)

print("Closing our connection after sending data to the client....")

client.close()

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

//code for reusing a port

#creating a socket object

mySock = socket.socket(socket.SOL\_SOCKET, socket.SO\_REUSEADDR)

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

// the while loop is open until the length of the data

while len(data):

data = client.recv(2048)

print ("Client sent this data : ", data)

client.send(data)

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

// Let’s see how it works in virtual Kali Linux.

sanjib@kali:~$ nslookup scanme.nmap.org

Server: 208.67.222.222

Address: 208.67.222.222#53

Non-authoritative answer:

Name: scanme.nmap.org

Address: 45.33.32.156

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

// Let’s see how it works in Host Ubuntu.

hagudu@hagudu-H81M-S1:~$ nslookup scanme.nmap.org

Server: 127.0.1.1

Address: 127.0.1.1#53

Non-authoritative answer:

Name: scanme.nmap.org

Address: 45.33.32.156

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

// trying nslookup in the host machine

hagudu@hagudu-H81M-S1:~$ nslookup 45.33.32.156

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

// trying nslookup in the virtual Kali Linux

sanjib@kali:~$ nslookup 45.33.32.156

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

sanjib@kali:~$ nmap scanme.nmap.org -vv

-----------

sanjib@kali:~$ curl ipinfo.io/45.33.32.156

{

"ip": "45.33.32.156",

"hostname": "No Hostname",

"city": "Fremont",

"region": "California",

"country": "US",

"loc": "37.5670,-121.9829",

"org": "AS63949 Linode, LLC",

"postal": "94536"

}

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

//code of using DMitry

dmitry google.com

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

//output from running DMitry

HostIP:216.58.203.206

HostName:google.com

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

//keeping DMitry output in a text file

Dmitry -wise -o dmitry.txt google.com

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