CN PROJECT

Create a Network Testing tool: It should analyze the network devices connected with gateways (loopback Ip it uses for that), Ping, Traceroute, and other features which you have experimented with in your LAB sessions.

GROUP MEMBERS:

NAMES	ROLL NUMBER	NETWORK	PUBLIC IP
D Tejashree	COE18B016	AIRTEL	**103.232.131.194**
G.Jaya Sathwika	COE18B019	ACT Fibernet	**49.204.9.179**
Sanjana Dara	CED18I015	JIO	**157.47.124.14**
V.Harini	CED18I023	AIRTEL	**106.208.49.99**
K.Dheekshitha	CED18I026	AIRTEL	**223.228.93.135**
Kavya Vemuri	CED18I027	AIRTEL	**103.73.222.213**

Ping:

Ping (latency is the technically more correct term) **means** the time it takes for a small data set to be transmitted from your device to a server on the Internet and back to your device again. The **ping** time is measured in milliseconds (ms).

Traceroute:

Traceroute is a network diagnostic tool **used to** track in real-time the pathway taken by a packet on an IP network from source to destination, reporting the IP addresses of all the routers it pinged in between. **Traceroute** also records the time taken for each hop the packet makes during its route to the destination.

Nslookup:

To **use** in interactive mode type **nslookup** at the command line and hit return. You should get an **nslookup** command prompt. To **use** in non-interactive mode type **nslookup** options at the command prompt.

Using Nslookup

1. Find the IP address of a host.
2. Find the domain name of an IP address.
3. Find mail servers for a domain.

Ifconfig:

ifconfig is a system administration utility in Unix-like operating systems for network interface configuration. The utility is a command-line interface tool and is also used in the system startup scripts of many operating systems.

Route:

Routing is the process of selecting a **path** for traffic in a **network** or between or across multiple **networks**. ... Packet forwarding is the transit of **network** packets from one **network** interface to another. Intermediate nodes are typically **network** hardware devices such as routers, gateways, firewalls, or switches

Dig:

Dig (**Domain Information Groper**) is a command line utility that performs DNS lookup by querying name servers and displaying the result to you. In this tutorial, you'll find all the basic uses of the command you should know in the Linux operating system.

Nmap:

Nmap is a **network** mapper that has emerged as one of the most popular, free **network** discovery tools on the market. ... The program can be used to find live hosts on a **network**, perform port scanning, ping sweeps, OS detection, and version detection

CODE:

SERVER SIDE:

```
import socket
from Tkinter import *
import Tkinter as tk
import tkMessageBox
import subprocess, platform
from subprocess import Popen, PIPE
from thread import *
import threading
clientlist = []
print_lock = threading.Lock()

# thread function
def threaded(c,addr,i):
    while True:
        # data received from client
        data = c.recv(1024)
        if not data:
            print('Disconnected from {} and port {}'. format(addr[0],
str(addr[1])) )
            del clientlist[i]
            #print_lock.release()
            break
    # connection closed
    c.close()

def Main():
    serverip="127.0.0.1"
    host = serverip
    port = 9009
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.bind((host, port))
    print("socket binded to port", port)

    # put the socket into listening mode
    s.listen(10)
```

```
print("socket is listening")

for i in range(6s):

    # establish connection with client
    c, addr = s.accept()
    l = {}
    #print_lock.acquire()
    print('Connected to :'+addr[0]+'  '+str(addr[1]))
    l["ip"] = addr[0]
    l["port"] = addr[1]
    clientlist.append(l)
    start_new_thread(threaded, (c,addr,i))


top = tk.Tk()
top.title("Server Network Testing Tool")
canvas = tk.Canvas(top, bg = "#99d6ff", height=550, width=750)
canvas.pack(fill=BOTH, expand = YES)

main_frame = tk.Frame(top, bg="black")
main_frame.place(relx=0.02, rely=0.2, relwidth=0.96, relheight=0.8)


scroll_bar = Scrollbar(main_frame)
scroll_bar.pack( side = RIGHT, fill = Y )
mylist = Listbox(main_frame,  yscrollcommand = scroll_bar.set )
scroll_bar.config( command = mylist.yview )


def clearFrame():
    for widget in main_frame.winfo_children():
       widget.destroy()

def pingbox(host, count):
    clearFrame()
    command = "python3 ping.py "+host+" "+str(count)
    stdout = Popen(command, shell=True, stdout=PIPE).stdout
```

```
        output = stdout.read()

        pinglabel = Label(main_frame, text=output, bg="black", fg="white")
        pinglabel.pack()

    def traceroutebox(host=serverip):
        clearFrame()
        command = "sudo python3 tracefinal.py "+host
        stdout = Popen(command, shell=True, stdout=PIPE).stdout
        output = stdout.read()
        for i in output.split("\n"):
            pinglabel = Label(main_frame, text=i, bg="black", fg="white")
            pinglabel.pack()


    def nslookupbox(host=serverip):
        clearFrame()
        command = "python3 nslookup.py "+host
        stdout = Popen(command, shell=True, stdout=PIPE).stdout
        output = stdout.read()
        pinglabel = Label(main_frame, text=output, bg="black", fg="white")
        pinglabel.pack()

    def ifconfigbox():
        clearFrame()
        stdout = Popen('python3 ifconfig.py', shell=True,
stdout=PIPE).stdout

        output = stdout.read()
        pinglabel = Label(main_frame, text=output, bg="black", fg="white")
        pinglabel.pack()
          #tkMessageBox.showinfo("IFCONFIG",output)

    def digbox(host=serverip):
        clearFrame()
        #result=ping(serverip)
        command = "python3 dig.py "+host
```

```
    stdout = Popen(command, shell=True, stdout=PIPE).stdout
    output = stdout.read()
    for i in output.split("\n"):
        pinglabel = Label(main_frame, text=i, bg="black", fg="white")
        pinglabel.pack()

def routebox(host=serverip):
   clearFrame()
   command = "python3 route.py "+host
   stdout = Popen(command, shell=True, stdout=PIPE).stdout
   output = stdout.read()
   routelabel = Label(main_frame, text=output, bg="black", fg="white")
   routelabel.pack()

def nmapbox(host=serverip):
   clearFrame()
   command = "python3 nmap.py "+host
   stdout = Popen(command, shell=True, stdout=PIPE).stdout
   output = stdout.read()
   nmaplabel = Label(main_frame, text=output, bg="black", fg="white")
   nmaplabel.pack()

def getdata(text_input,top1,funcname):
        data = text_input.get();
        top1.destroy()
        funcname(clientlist[int(data)-1]['ip'])

def pingcustom():
    top1 = tk.Tk()
    top1.title("Custom ping")
    canvas1 = tk.Canvas(top1, height=260, width=300)
    canvas1.pack()
    data=[]
    f = open("clientdata.txt","w")
    f.write("Client\t  ip\tport\n")
    for i in range(len(clientlist)):
          #print(clientlist[i])
        f.write(str(i+1)+"         "+clientlist[i]['ip']+"
```

```
"+str(clientlist[i]['port'])+"\n")
            data.append(clientlist[i]['ip']+"   "+str(clientlist[i]['port'])+"\n")
        f.close()

        def getdataping(text_input,top1,funcname):
            data = text_input.get();
            num,count = data.split(",")
            top1.destroy()
            funcname(clientlist[int(num)-1]['ip'],count)

        f = open("clientdata.txt","r")
        cdata = f.read()
        P = Label(top1, text = cdata, font = ('times', 12,'bold'))
        f.close()
        P.place(x=2, y=10, width= 280, height=150)
        L = Label(top1, text = "Enter Client Number, number of pings", font =
('times', 12,'bold'))
        L.place(x=5, y=140, width= 280, height=40)
        text_input = Entry(top1, width=30)
        text_input.bind("<Return>", lambda i :
getdataping(text_input,top1,pingbox))
        text_input.place(x=10, y=180,width=180, height=30)
        S = Label(top1, text = "Press enter to proceed", font = ('times',
11,'italic'))
        S.place(x=5, y=220, width= 180, height=20)
        top1.mainloop()

    def traceroutecustom():
        top1 = tk.Tk()
        top1.title("Custom traceroute")
        canvas1 = tk.Canvas(top1, height=260, width=300)
        canvas1.pack()
        data=[]
        f = open("clientdata.txt","w")
        f.write("Client\t  ip\tport\n")
        for i in range(len(clientlist)):
            f.write(str(i+1)+"          "+clientlist[i]['ip']+"
"+str(clientlist[i]['port'])+"\n")
```

```
        f.close()
        f = open("clientdata.txt","r")
        cdata = f.read()
        P = Label(top1, text = cdata, font = ('times', 12,'bold'))
        f.close()
        P.place(x=2, y=10, width= 280, height=150)
        L = Label(top1, text = "Enter Client Number:", font = ('times',
12,'bold'))
        L.place(x=5, y=140, width= 280, height=40)
        text_input = Entry(top1, width=30)
        text_input.bind("<Return>", lambda i :
getdata(text_input,top1,traceroutebox))
        text_input.place(x=10, y=180,width=180, height=30)
        S = Label(top1, text = "Press enter to proceed", font = ('times',
11,'italic'))
        S.place(x=5, y=220, width= 180, height=20)
        top1.mainloop()

    def nslookupcustom():
        top1 = tk.Tk()
        top1.title("Custom nslookup")
        canvas1 = tk.Canvas(top1, height=260, width=300)
        canvas1.pack()
        data=[]
        f = open("clientdata.txt","w")
        f.write("Client\t  ip\tport\n")
        for i in range(len(clientlist)):
            f.write(str(i+1)+"          "+clientlist[i]['ip']+"
"+str(clientlist[i]['port'])+"\n")
        f.close()
        f = open("clientdata.txt","r")
        cdata = f.read()
        P = Label(top1, text = cdata, font = ('times', 12,'bold'))
        f.close()
        P.place(x=2, y=10, width= 280, height=150)
        L = Label(top1, text = "Enter Client Number:", font = ('times',
12,'bold'))
        L.place(x=5, y=140, width= 280, height=40)
```

```
        text_input = Entry(top1, width=30)
        text_input.bind("<Return>", lambda i :
getdata(text_input,top1,nslookupbox))
        text_input.place(x=10, y=180,width=180, height=30)
        S = Label(top1, text = "Press enter to proceed", font = ('times',
11,'italic'))
        S.place(x=5, y=220, width= 180, height=20)
        top1.mainloop()

    def digcustom():
        top1 = tk.Tk()
        top1.title("Custom dig")
        canvas1 = tk.Canvas(top1, height=260, width=300)
        canvas1.pack()
        data=[]
        f = open("clientdata.txt","w")
        f.write("Client\t  ip\tport\n")
        for i in range(len(clientlist)):
            f.write(str(i+1)+"          "+clientlist[i]['ip']+"
"+str(clientlist[i]['port'])+"\n")
        f.close()
        f = open("clientdata.txt","r")
        cdata = f.read()
        P = Label(top1, text = cdata, font = ('times', 12,'bold'))
        f.close()
        P.place(x=2, y=10, width= 280, height=150)
        L = Label(top1, text = "Enter Client Number :", font = ('times',
12,'bold'))
        L.place(x=5, y=140, width= 280, height=40)
        text_input = Entry(top1, width=30)
        text_input.bind("<Return>", lambda i :
getdata(text_input,top1,digbox))
        text_input.place(x=10, y=180,width=180, height=30)
        S = Label(top1, text = "Press enter to proceed", font = ('times',
11,'italic'))
        S.place(x=5, y=220, width= 180, height=20)

    def nmapcustom():
```

```
        top1 = tk.Tk()
        top1.title("Custom nmap")
        canvas1 = tk.Canvas(top1, height=260, width=300)
        canvas1.pack()
        data=[]
        f = open("clientdata.txt","w")
        f.write("Client\t  ip\tport\n")
        for i in range(len(clientlist)):
            f.write(str(i+1)+"          "+clientlist[i]['ip']+"
"+str(clientlist[i]['port'])+"\n")
        f.close()
        f = open("clientdata.txt","r")
        cdata = f.read()
        P = Label(top1, text = cdata, font = ('times', 12,'bold'))
        f.close()
        P.place(x=2, y=10, width= 280, height=150)
        L = Label(top1, text = "Enter Client Number :", font = ('times',
12,'bold'))
        L.place(x=5, y=140, width= 280, height=40)
        text_input = Entry(top1, width=30)
        text_input.bind("<Return>", lambda i :
getdata(text_input,top1,nmapbox))
        text_input.place(x=10, y=180,width=180, height=30)
        S = Label(top1, text = "Press enter to proceed", font = ('times',
11,'italic'))
        S.place(x=5, y=220, width= 180, height=20)
        top1.mainloop()

    B=  tk.Button( top, text="ping",fg = "white", bg="black", font=('times', 16,
'bold'), cursor="hand2", command=pingcustom)
    B.place(relx=0, rely=0.04, relwidth=0.143, relheight=0.075)



    C=  tk.Button ( top, text="traceroute",  fg = "white", bg="black",
font=('times', 16, 'bold'), cursor="hand2", command=traceroutecustom)
    C.place(relx=0.143, rely=0.04, relwidth=0.143, relheight=0.075)

    D=  tk.Button ( top, text="nslookup" , fg = "white", bg="black",
```

```
font=('times', 16, 'bold'), cursor="hand2", command=nslookupcustom)
    D.place(relx=0.286, rely=0.04, relwidth=0.143, relheight=0.075)

    E = tk.Button(top, text ="ifconfig", fg = "white", bg="black",font=('times',
16, 'bold'),cursor = "hand2", command = ifconfigbox)
    E.place(relx=0.429, rely=0.04, relwidth=0.143, relheight=0.075)


    F=  tk.Button ( top, text="dig", relief=RAISED , fg = "white", bg="black",
font=('times', 16, 'bold'), cursor="hand2", command=digcustom)
    F.place(relx=0.572, rely=0.04, relwidth=0.143, relheight=0.075,)

    G = tk.Button(top, text ="route", fg = "white", bg="black",font=('times',
16, 'bold'),command = routebox, cursor="hand2")
    G.place(relx=0.715, rely=0.04, relwidth=0.143, relheight=0.075,)

    H=  tk.Button(top, text ="nmap", fg = "white", bg="black",font=('times',
16, 'bold'),command = nmapcustom, cursor="hand2")
    H.place(relx=0.858, rely=0.04, relwidth=0.143, relheight=0.075,)

    top.mainloop()
    s.close()
if __name__ == '__main__':
    Main()
```

CLIENT SIDE:

```
import socket
from Tkinter import *
import Tkinter as tk
import tkMessageBox
import subprocess, platform
from subprocess import Popen, PIPE

def Main():
    host = '127.0.0.1'   #35.246.29.33(hosted server ip address)
    serverip = host
```

```
port = 9009
s = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect((host,port))

top = tk.Tk()
top.title("Network Testing Tool")
canvas = tk.Canvas(top, height=550, width=750)
canvas.pack(fill=BOTH, expand = YES)

main_frame = tk.Frame(top, bg="black")
main_frame.place(relx=0.02, rely=0.2, relwidth=0.96, relheight=0.8)

scroll_bar = Scrollbar(main_frame)
scroll_bar.pack( side = RIGHT, fill = Y )
mylist = Listbox(main_frame,  yscrollcommand = scroll_bar.set )
scroll_bar.config( command = mylist.yview )

def clearFrame():
   for widget in main_frame.winfo_children():
      widget.destroy()

def pingbox(host=serverip, count=5):
   clearFrame()
   command = "python3 ping.py "+host+" "+str(count)
   stdout = Popen(command, shell=True, stdout=PIPE).stdout
   output = stdout.read()
   pinglabel = Label(main_frame, text=output, bg="black", fg="white")
   pinglabel.pack()

def traceroutebox(host=serverip):
   clearFrame()
   command = "sudo python3 tracefinal.py "+host
   stdout = Popen(command, shell=True, stdout=PIPE).stdout
   output = stdout.read()
   for i in output.split("\n"):
      pinglabel = Label(main_frame, text=i, bg="black", fg="white")
      pinglabel.pack()
```

```
def nslookupbox(host=serverip):
   clearFrame()
   command = "python3 nslookup.py "+host
   stdout = Popen(command, shell=True, stdout=PIPE).stdout
   output = stdout.read()
   pinglabel = Label(main_frame, text=output, bg="black", fg="white")
   pinglabel.pack()

def ifconfigbox():
   clearFrame()
   stdout = Popen('python3 ifconfig.py', shell=True, stdout=PIPE).stdout
   output = stdout.read()
   pinglabel = Label(main_frame, text=output, bg="black", fg="white")
   pinglabel.pack()

def digbox(host=serverip):
   clearFrame()
   command = "python3 dig.py "+host
   stdout = Popen(command, shell=True, stdout=PIPE).stdout
   output = stdout.read()
   for i in output.split("\n"):
      pinglabel = Label(main_frame, text=i, bg="black", fg="white")
      pinglabel.pack()

def nmapbox(host=serverip):
   clearFrame()
   command = "python3 nmap.py "+host
   stdout = Popen(command, shell=True, stdout=PIPE).stdout
   output = stdout.read()
   nmaplabel = Label(main_frame, text=output, bg="black", fg="white")
   nmaplabel.pack()

def routebox(host=serverip):
   clearFrame()
   command = "python3 route.py "+host
   stdout = Popen(command, shell=True, stdout=PIPE).stdout
   output = stdout.read()
   routelabel = Label(main_frame, text=output, bg="black", fg="white")
```

```
      routelabel.pack()


   def getdataping(text_input,top1,funcname):
      data = text_input.get();
      host,count = data.split(",")
      top1.destroy()
      funcname(host,count)

   def getdata(text_input,top1,funcname):
      data = text_input.get();
      top1.destroy()
      funcname(data)

   def pingcustom():
      top1 = tk.Tk()
      top1.title("Custom ping")
      canvas1 = tk.Canvas(top1, height=100, width=200)
      canvas1.pack()
      L = Label(top1, text = "Enter IP address,count", font = ('times',
12,'bold'))
      L.place(x=5, y=0.2, width= 180, height=40)
      text_input = Entry(top1, width=30)
      text_input.bind("<Return>", lambda i :
getdataping(text_input,top1,pingbox))
      text_input.place(x=10, y=30,width=180, height=30)
      S = Label(top1, text = "Press enter to proceed", font = ('times',
11,'italic'))
      S.place(x=5, y=60, width= 180, height=40)
      top1.mainloop()

   B=  Menubutton ( top, text="ping", relief=RAISED , fg = "white",
bg="black", font=('times', 16, 'bold'), cursor="hand2", direction=RIGHT)
   B.menu =  Menu ( B, tearoff = 0 )
   B["menu"] =  B.menu
   B.menu.add_checkbutton ( label="ping server" ,command=pingbox)
   B.menu.add_checkbutton ( label="custom" ,command=pingcustom)
   B.place(relx=0, rely=0.04, relwidth=0.143, relheight=0.075)
```

```
    def traceroutecustom():
        top1 = tk.Tk()
        top1.title("Custom Traceroute")
        canvas1 = tk.Canvas(top1, height=100, width=250)
        canvas1.pack()
        L = Label(top1, text = "Enter IP address", font = ('times', 12,'bold'))
        L.place(x=5, y=0.2, width= 180, height=40)
        text_input = Entry(top1, width=30)
        text_input.bind("<Return>", lambda i :
getdata(text_input,top1,traceroutebox))
        text_input.place(x=10, y=30,width=180, height=30)
        S = Label(top1, text = "Press enter to proceed", font = ('times',
11,'italic'))
        S.place(x=5, y=60, width= 180, height=40)
        top1.mainloop()

    C=  Menubutton ( top, text="traceroute", relief=RAISED , fg = "white",
bg="black", font=('times', 16, 'bold'), cursor="hand2")
    C.menu =  Menu ( C, tearoff = 0 )
    C["menu"] =  C.menu
    C.menu.add_checkbutton ( label="server",command = traceroutebox)
    C.menu.add_checkbutton ( label="custom",command =
traceroutecustom)
    C.place(relx=0.143, rely=0.04, relwidth=0.143, relheight=0.075)

    def nslookupcustom():
        top1 = tk.Tk()
        top1.title("Custom nslookup")
        canvas1 = tk.Canvas(top1, height=100, width=250)
        canvas1.pack()
        L = Label(top1, text = "Enter IP address", font = ('times', 12,'bold'))
        L.place(x=5, y=0.2, width= 180, height=40)
        text_input = Entry(top1, width=30)
        text_input.bind("<Return>", lambda i :
getdata(text_input,top1,nslookupbox))
        text_input.place(x=10, y=30,width=180, height=30)
        S = Label(top1, text = "Press enter to proceed", font = ('times',
```

```
11,'italic'))
      S.place(x=5, y=60, width= 180, height=40)
      top1.mainloop()

   D=  Menubutton ( top, text="nslookup", relief=RAISED , fg = "white",
bg="black", font=('times', 16, 'bold'), cursor="hand2")
   D.menu =  Menu ( D, tearoff = 0 )
   D["menu"] =  D.menu
   D.menu.add_checkbutton ( label="server",command = nslookupbox)
   D.menu.add_checkbutton ( label="custom",command =
nslookupcustom)
   D.place(relx=0.286, rely=0.04, relwidth=0.143, relheight=0.075)

   E = tk.Button(top, text ="ifconfig", fg = "white", bg="black",font=('times',
16, 'bold'),cursor = "hand2", command = ifconfigbox)
   E.place(relx=0.429, rely=0.04, relwidth=0.143, relheight=0.075)

   def digcustom():
      top1 = tk.Tk()
      top1.title("Custom dig")
      canvas1 = tk.Canvas(top1, height=100, width=200)
      canvas1.pack()
      L = Label(top1, text = "Enter IP address", font = ('times', 12,'bold'))
      L.place(x=5, y=0.2, width= 180, height=40)
      text_input = Entry(top1, width=30)
      text_input.bind("<Return>", lambda i :
getdata(text_input,top1,digbox))
      text_input.place(x=10, y=30,width=180, height=30)
      S = Label(top1, text = "Press enter to proceed", font = ('times',
11,'italic'))
      S.place(x=5, y=60, width= 180, height=40)
      top1.mainloop()

   F=  Menubutton ( top, text="dig", relief=RAISED , fg = "white",
bg="black",font=('times', 16, 'bold'), cursor="hand2")
   F.menu =  Menu ( F, tearoff = 0 )
   F["menu"] =  F.menu
   F.menu.add_checkbutton ( label="server" ,command = digbox)
```

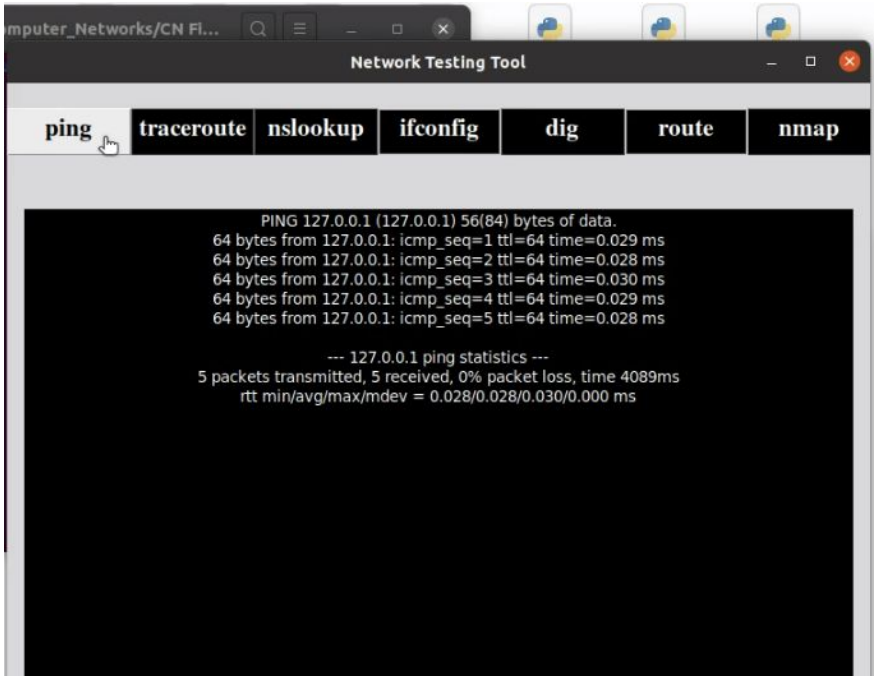
```
    F.menu.add_checkbutton ( label="custom",command = digcustom)
    F.place(relx=0.572, rely=0.04, relwidth=0.143, relheight=0.075,)

    G = tk.Button(top, text ="route", fg = "white", bg="black",font=('times',
16, 'bold'),command = routebox, cursor="hand2")
    G.place(relx=0.715, rely=0.04, relwidth=0.143, relheight=0.075,)

    def nmapcustom():
        top1 = tk.Tk()
        top1.title("Custom nmap")
        canvas1 = tk.Canvas(top1, height=100, width=200)
        canvas1.pack()
        L = Label(top1, text = "Enter IP address", font = ('times', 12,'bold'))
        L.place(x=5, y=0.2, width= 180, height=40)
        text_input = Entry(top1, width=30)
        text_input.bind("<Return>", lambda i :
getdata(text_input,top1,nmapbox))
        text_input.place(x=10, y=30,width=180, height=30)
        S = Label(top1, text = "Press enter to proceed", font = ('times',
11,'italic'))
        S.place(x=5, y=60, width= 180, height=40)
        top1.mainloop()

    H=  Menubutton ( top, text="nmap", relief=RAISED , fg = "white",
bg="black",font=('times', 16, 'bold'), cursor="hand2")
    H.menu =  Menu ( H, tearoff = 0 )
    H["menu"] =  H.menu
    H.menu.add_checkbutton ( label="server" ,command = nmapbox)
    H.menu.add_checkbutton ( label="custom",command = nmapcustom)
    H.place(relx=0.858, rely=0.04, relwidth=0.143, relheight=0.075,)

    top.mainloop()
    s.close()

if __name__ == '__main__':
    Main()
```

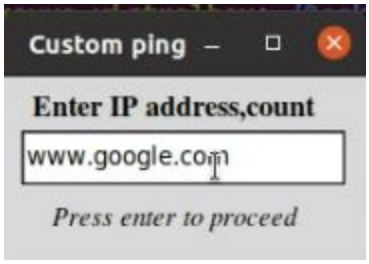
OUTPUT:

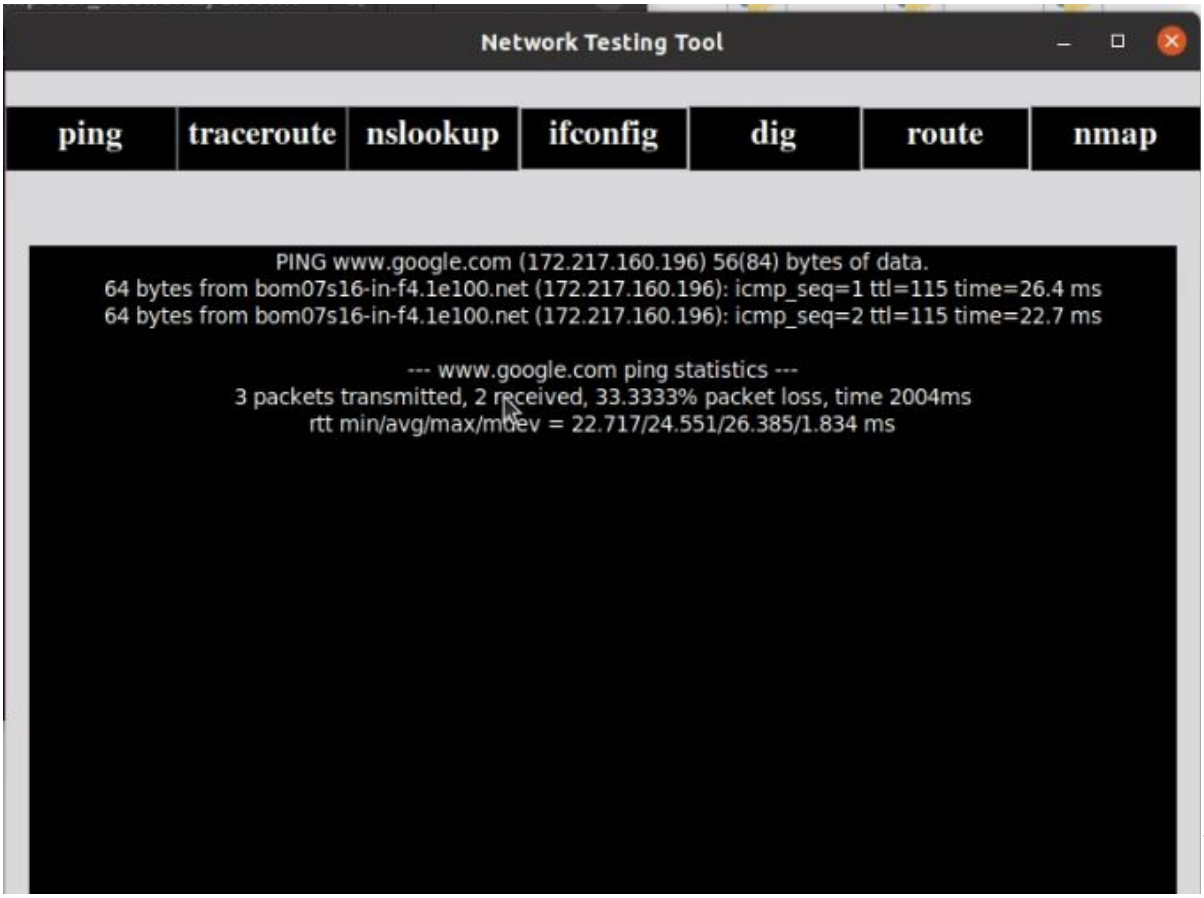
LOOPBACK IP:

CLIENT:

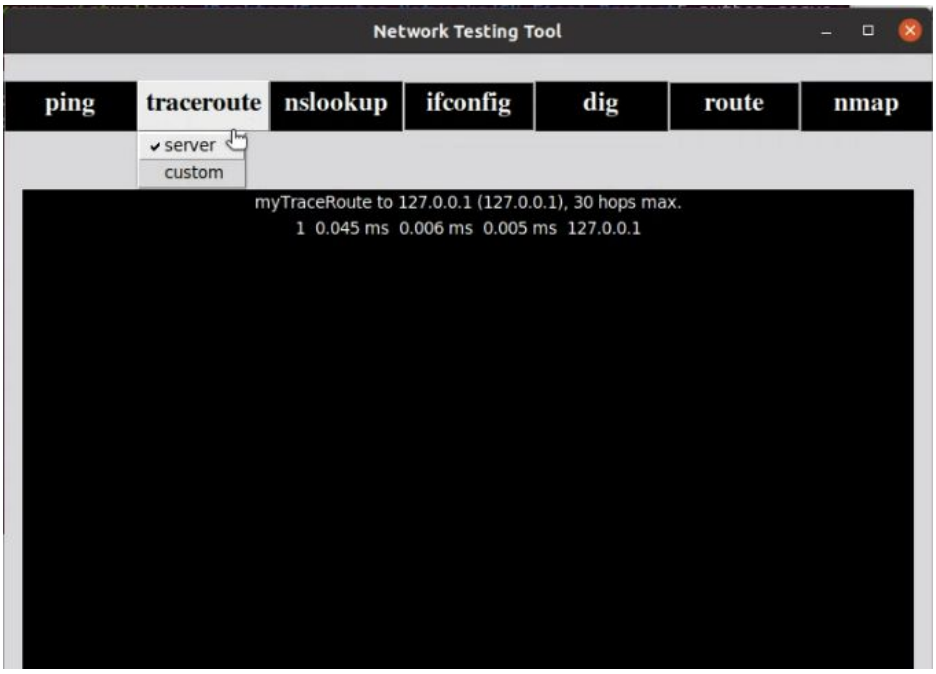
Ping:



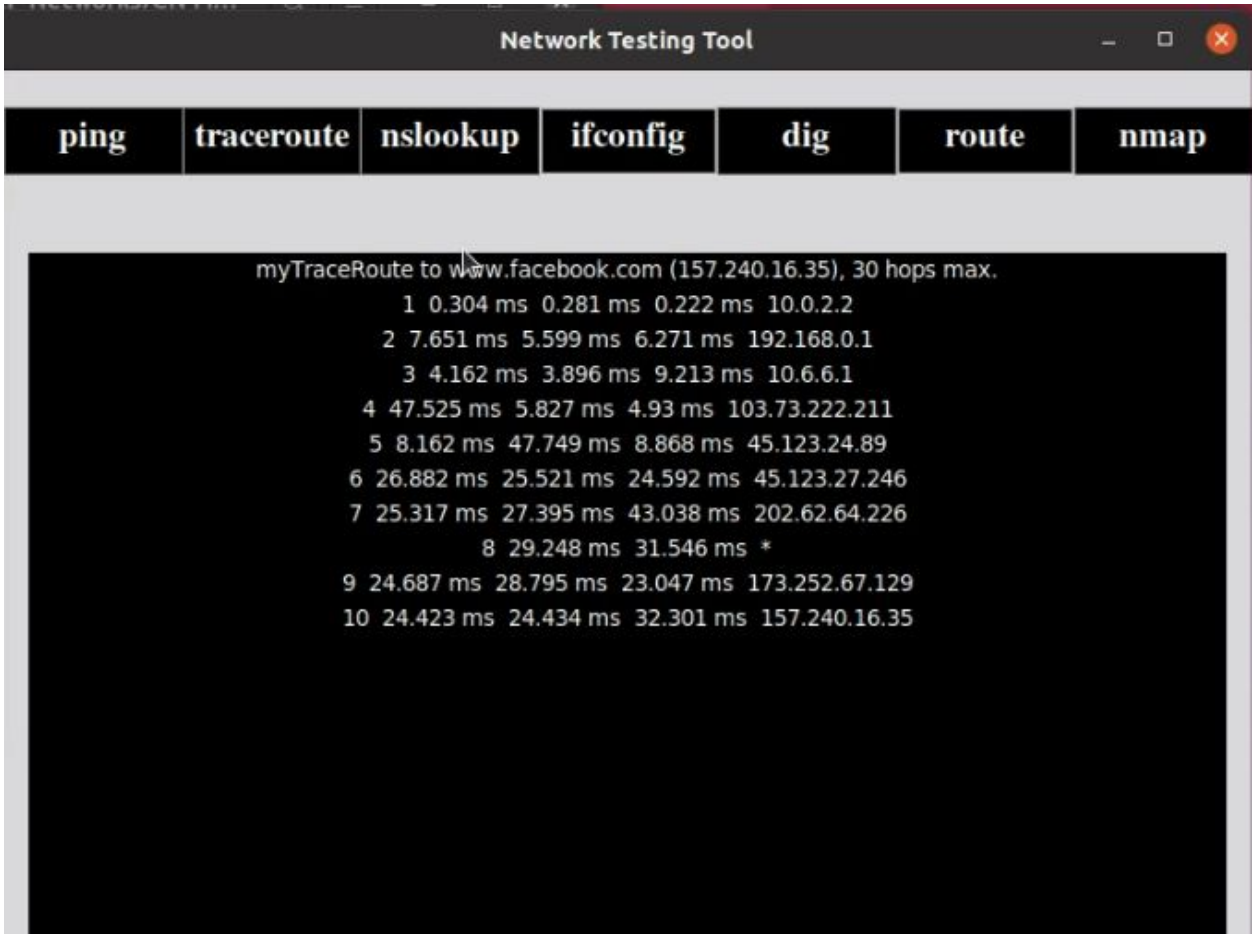




Traceroute:

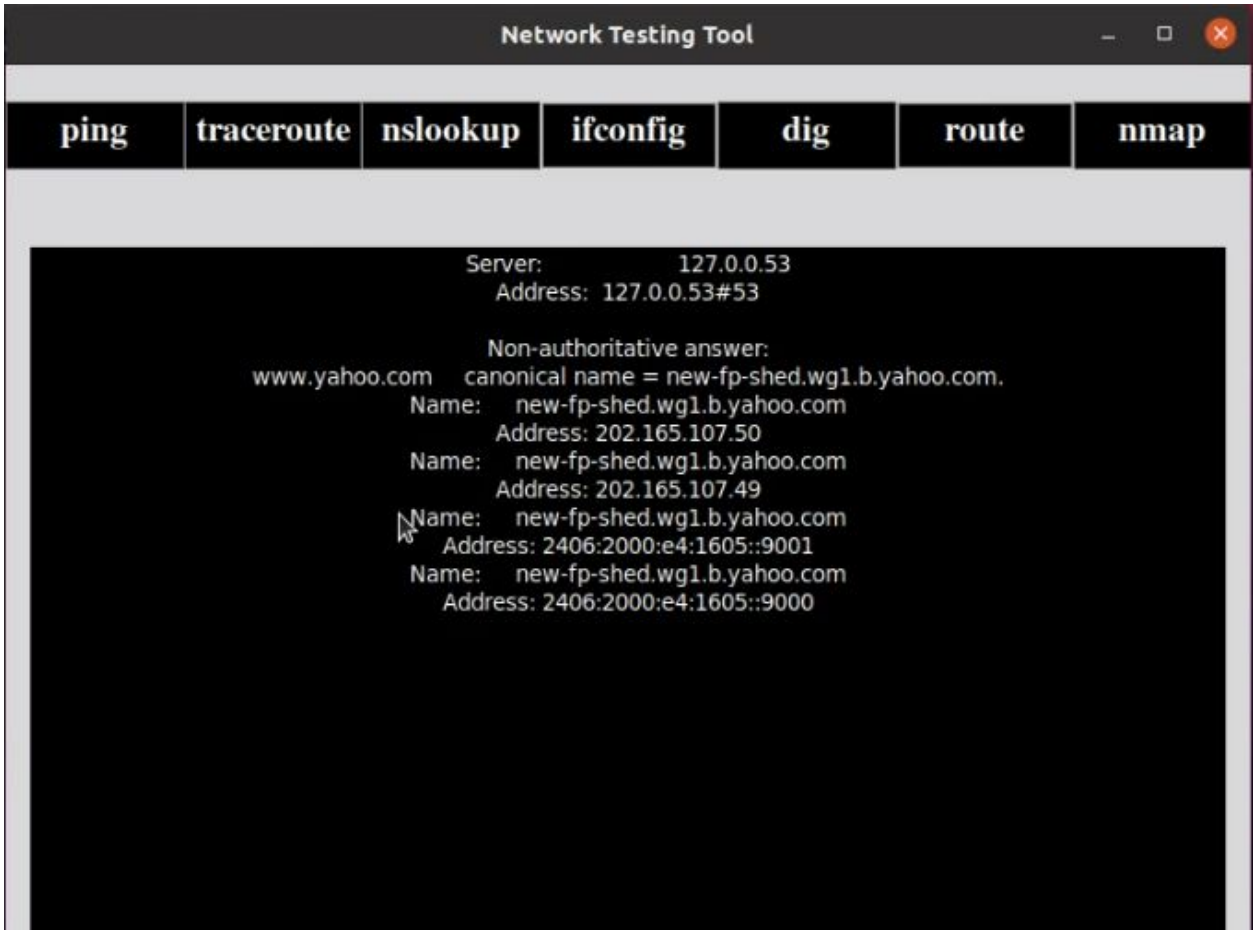

Network Testing Tool
ping
traceroute
nslookup
ifconfig
dig
route
nmap
server
custom
myTraceRoute to 127.0.0.1 (127.0.0.1), 30 hops max.
1 0.045 ms 0.006 ms 0.005 ms 127.0.0.1


Custom Traceroute
Enter IP address
www.facebook.com
Press enter to proceed

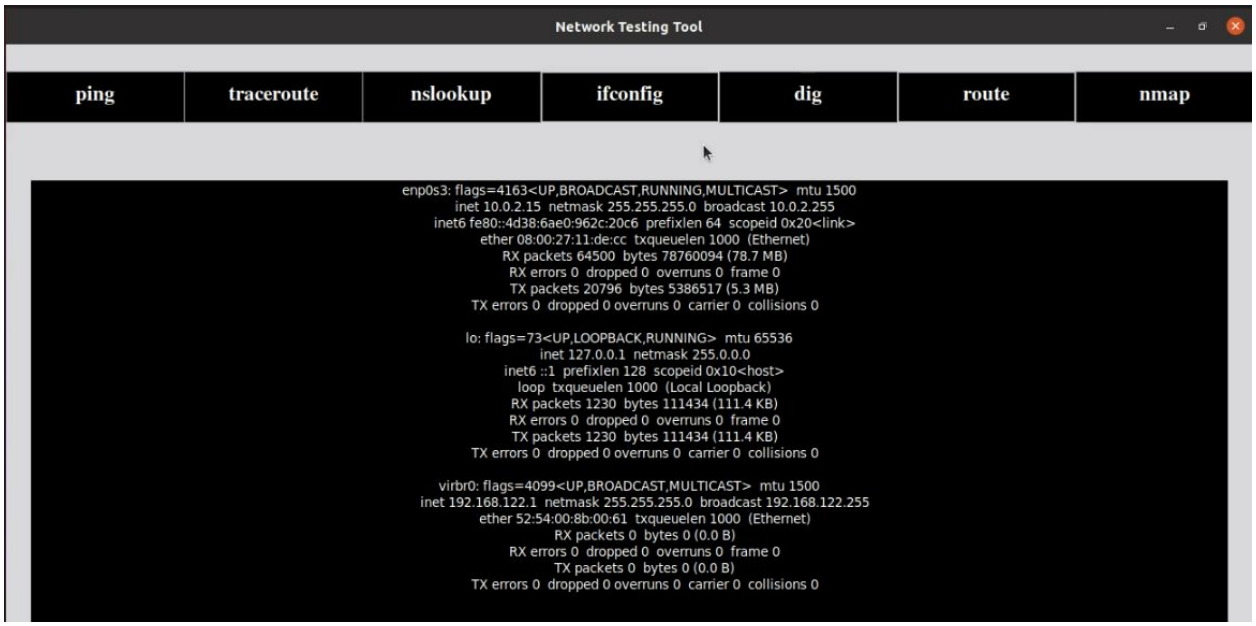

Network Testing Tool
ping
traceroute
nslookup
ifconfig
dig
route
nmap
myTraceRoute to www.facebook.com (157.240.16.35), 30 hops max.
1 0.304 ms 0.281 ms 0.222 ms 10.0.2.2
2 7.651 ms 5.599 ms 6.271 ms 192.168.0.1
3 4.162 ms 3.896 ms 9.213 ms 10.6.6.1
4 47.525 ms 5.827 ms 4.93 ms 103.73.222.211
5 8.162 ms 47.749 ms 8.868 ms 45.123.24.89
6 26.882 ms 25.521 ms 24.592 ms 45.123.27.246
7 25.317 ms 27.395 ms 43.038 ms 202.62.64.226
8 29.248 ms 31.546 ms *
9 24.687 ms 28.795 ms 23.047 ms 173.252.67.129
10 24.423 ms 24.434 ms 32.301 ms 157.240.16.35

nslookup:

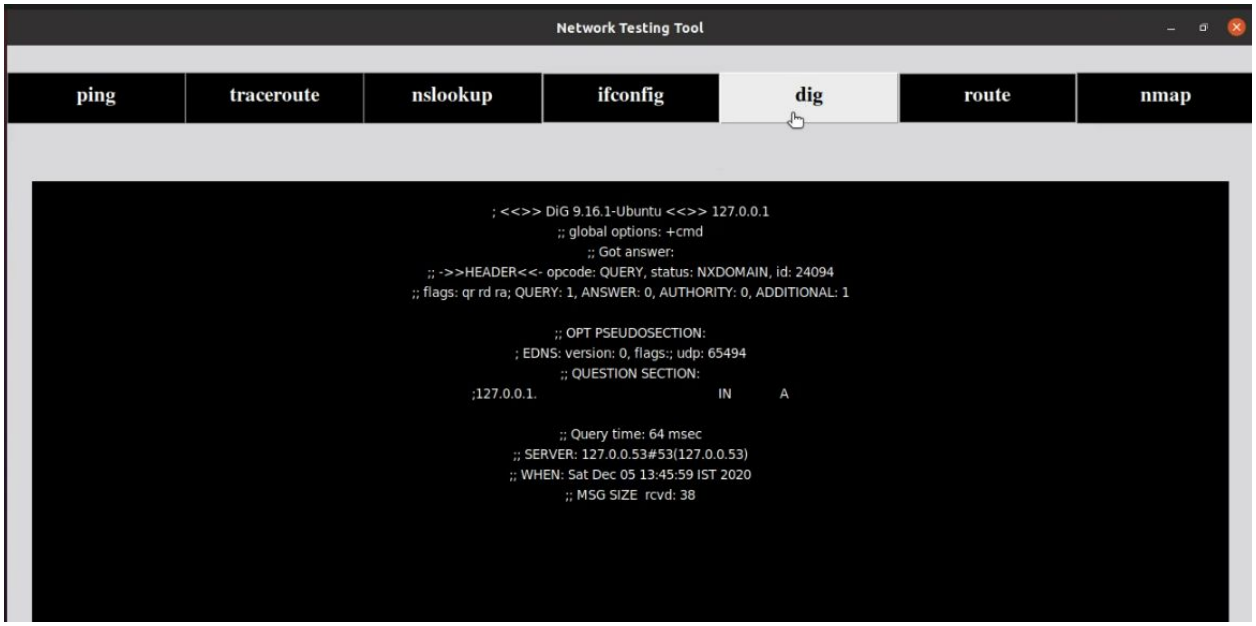

Custom nslookup
Enter IP address
www.yahoo.com
Press enter to proceed

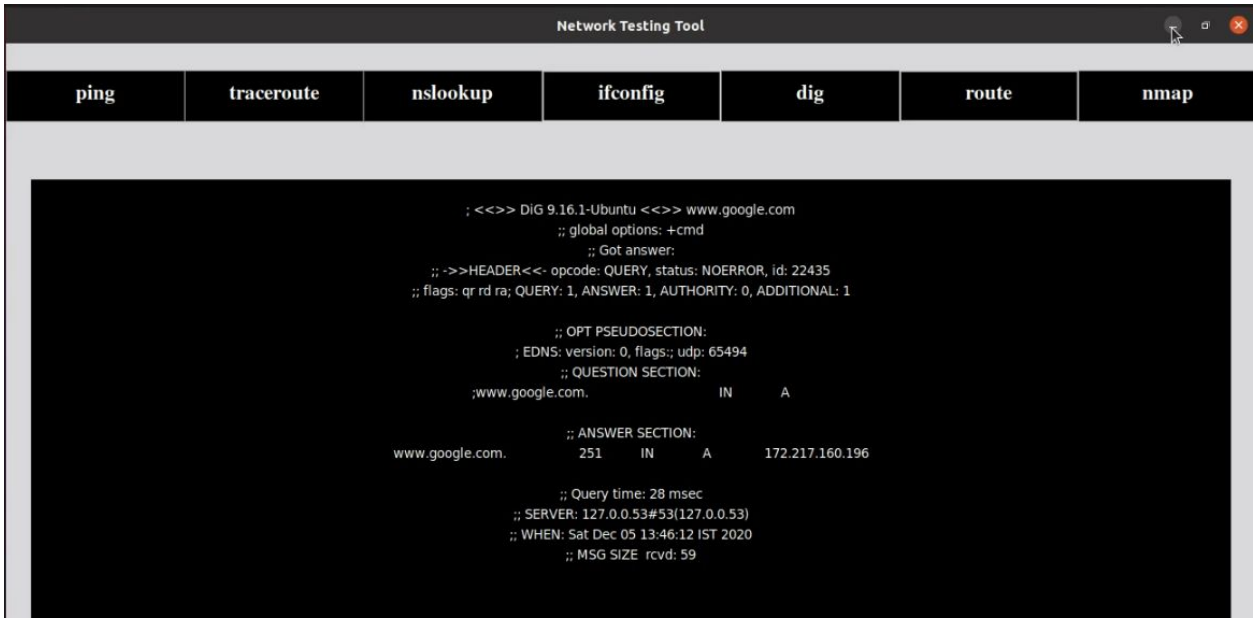

Network Testing Tool
ping
traceroute
nslookup
ifconfig
dig
route
nmap
Server: 127.0.0.53
Address: 127.0.0.53#53
Non-authoritative answer:
www.yahoo.com canonical name = new-fp-shed.wg1.b.yahoo.com.
Name: new-fp-shed.wg1.b.yahoo.com
Address: 202.165.107.50
Name: new-fp-shed.wg1.b.yahoo.com
Address: 202.165.107.49
Name: new-fp-shed.wg1.b.yahoo.com
Address: 2406:2000:e4:1605::9001
Name: new-fp-shed.wg1.b.yahoo.com
Address: 2406:2000:e4:1605::9000

ifconfig:

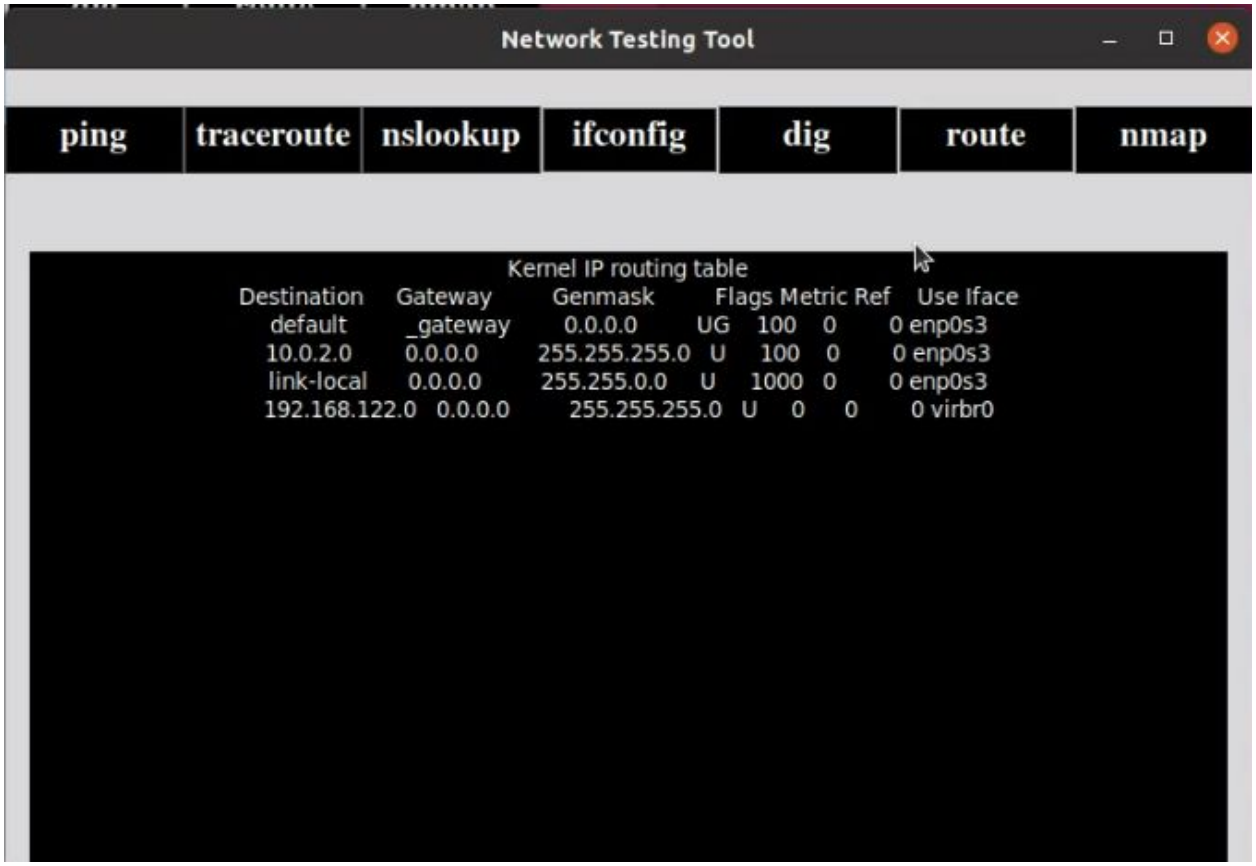


dig:

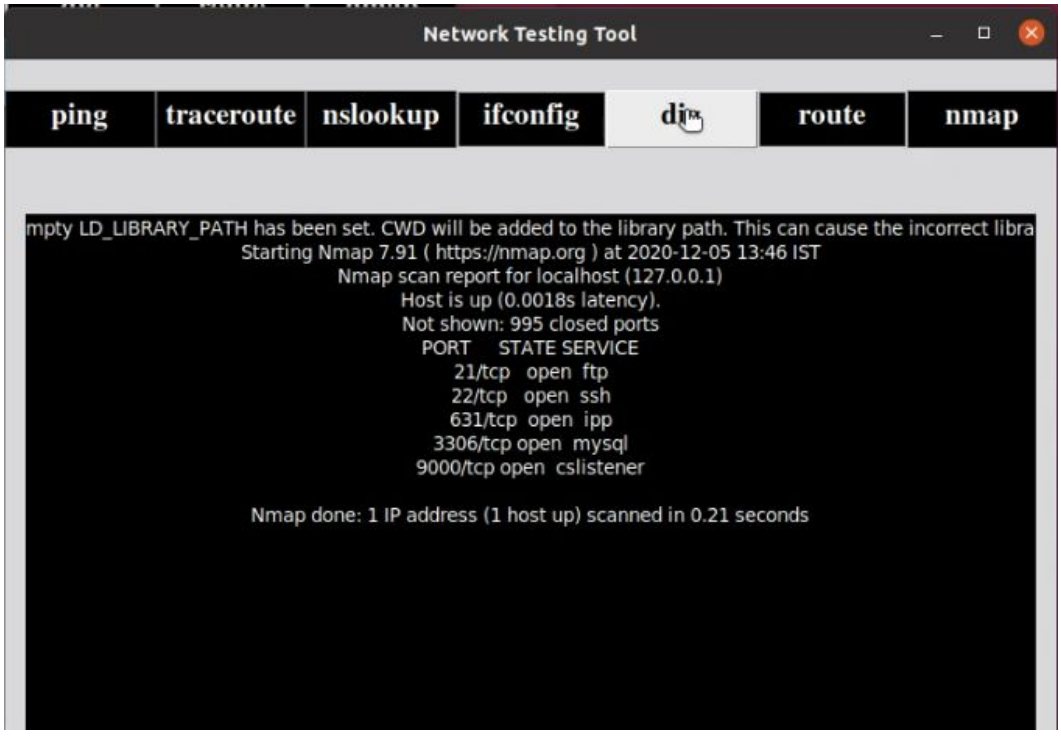



Network Testing Tool
ping
traceroute
nslookup
ifconfig
dig
route
nmap
; <<>> DiG 9.16.1-Ubuntu <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22435
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;www.google.com. IN A
;; ANSWER SECTION:
www.google.com. 251 IN A 172.217.160.196
;; Query time: 28 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sat Dec 05 13:46:12 IST 2020
;; MSG SIZE rcvd: 59

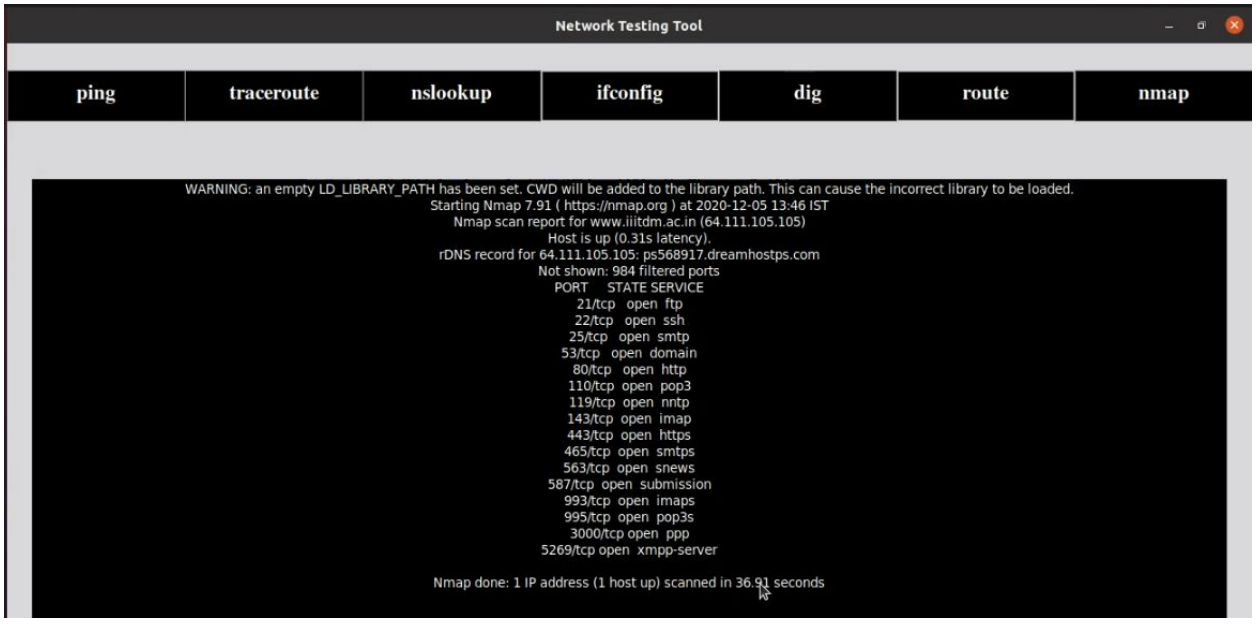
route:


Network Testing Tool
ping
traceroute
nslookup
ifconfig
dig
route
nmap
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default _gateway 0.0.0.0 UG 100 0 0 enp0s3
10.0.2.0 0.0.0.0 255.255.255.0 U 100 0 0 enp0s3
link-local 0.0.0.0 255.255.0.0 U 1000 0 0 enp0s3
192.168.122.0 0.0.0.0 255.255.255.0 U 0 0 0 virbr0

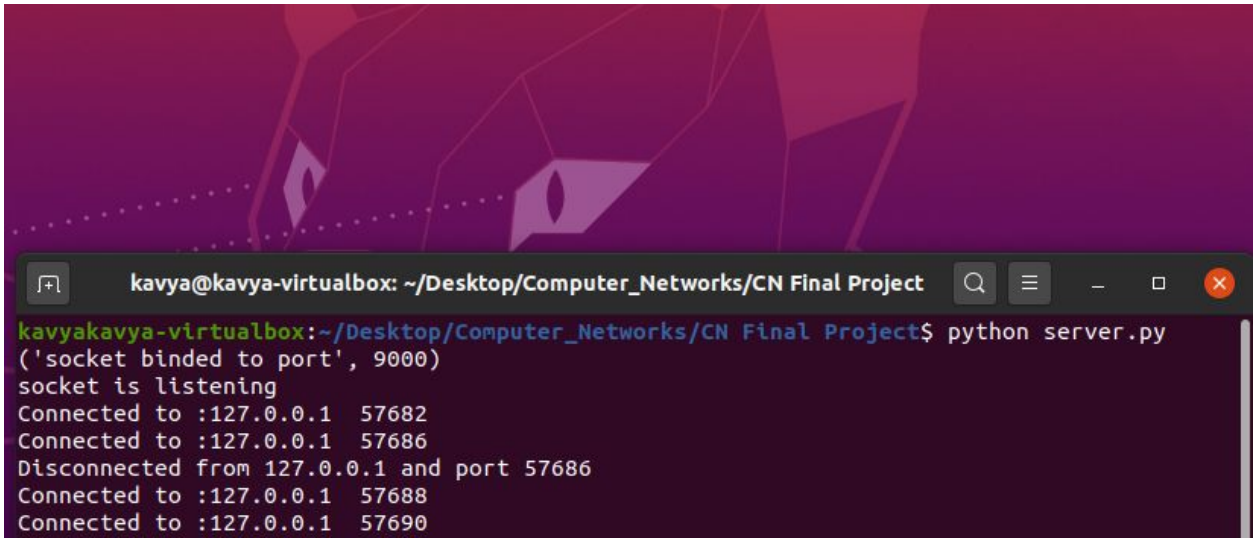
Nmap:


Network Testing Tool
ping
traceroute
nslookup
ifconfig
dig
route
nmap
mpty LD_LIBRARY_PATH has been set. CWD will be added to the library path. This can cause the incorrect libra
Starting Nmap 7.91 (https://nmap.org) at 2020-12-05 13:46 IST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0018s latency).
Not shown: 995 closed ports
PORT STATE SERVICE
21/tcp open ftp
22/tcp open ssh
631/tcp open ipp
3306/tcp open mysql
9000/tcp open cslistener
Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds

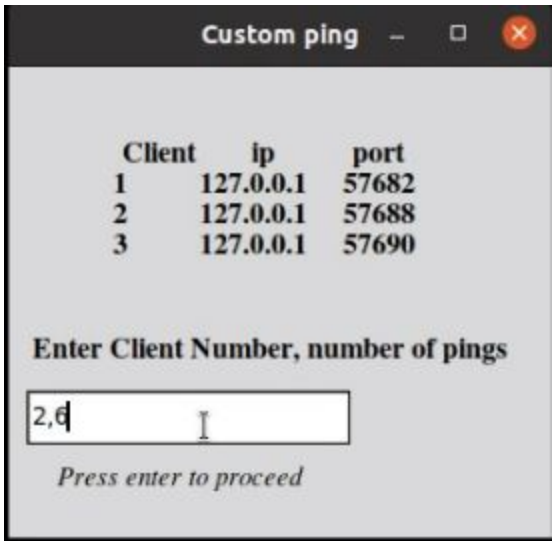

Custom nmap
Enter IP address
www.iiitdm.ac.in
Press enter to proceed

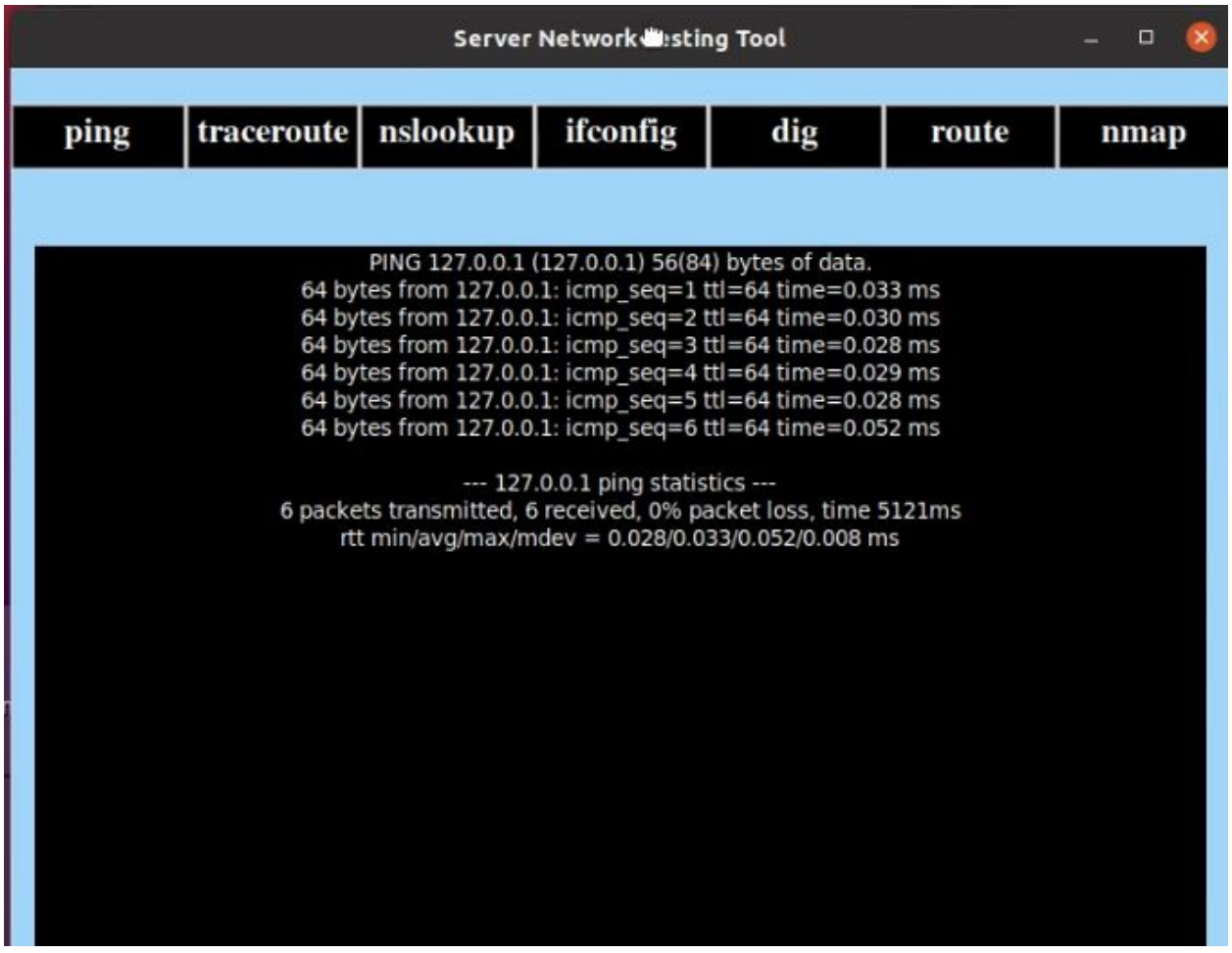

Network Testing Tool
ping
traceroute
nslookup
ifconfig
dig
route
nmap
WARNING: an empty LD_LIBRARY_PATH has been set. CWD will be added to the library path. This can cause the incorrect library to be loaded.
Starting Nmap 7.91 (https://nmap.org) at 2020-12-05 13:46 IST
Nmap scan report for www.iiitdm.ac.in (64.111.105.105)
Host is up (0.31s latency).
rDNS record for 64.111.105.105: ps568917.dreamhostps.com
Not shown: 984 filtered ports
PORT STATE SERVICE
21/tcp open ftp
22/tcp open ssh
25/tcp open smtp
53/tcp open domain
80/tcp open http
110/tcp open pop3
119/tcp open nntp
143/tcp open imap
443/tcp open https
465/tcp open smtps
563/tcp open snews
587/tcp open submission
993/tcp open imaps
995/tcp open pop3s
3000/tcp open ppp
5269/tcp open xmpp-server
Nmap done: 1 IP address (1 host up) scanned in 36.91 seconds

SERVER:

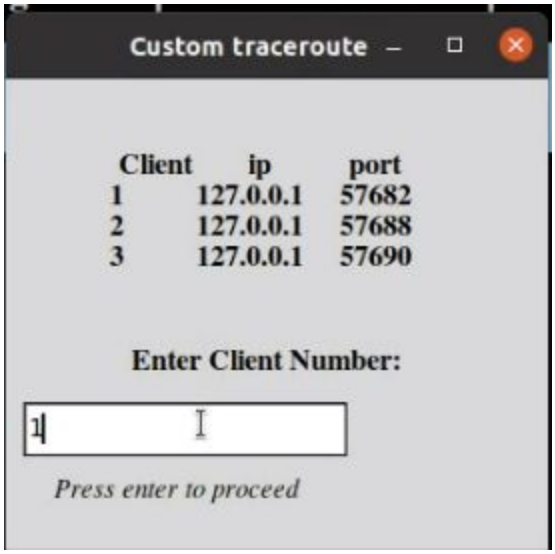

kavya@kavya-virtualbox: ~/Desktop/Computer_Networks/CN Final Project
kavyakavya-virtualbox:~/Desktop/Computer_Networks/CN Final Project$ python server.py
('socket binded to port', 9000)
socket is listening
Connected to :127.0.0.1 57682
Connected to :127.0.0.1 57686
Disconnected from 127.0.0.1 and port 57686
Connected to :127.0.0.1 57688
Connected to :127.0.0.1 57690

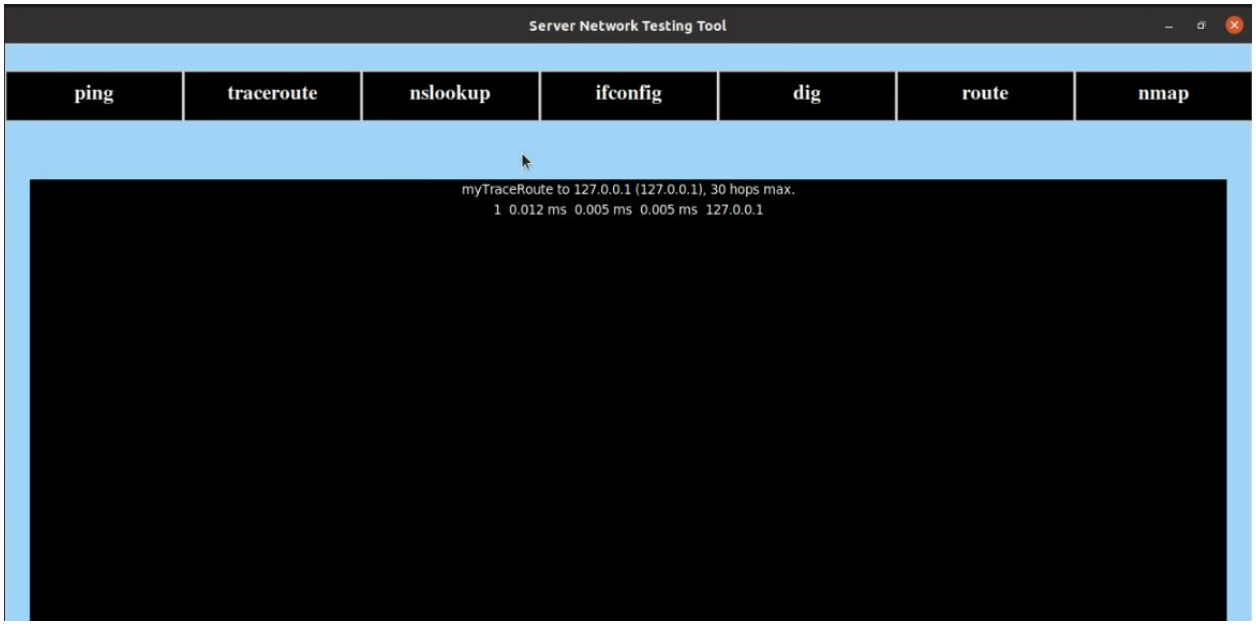
ping:


Custom ping
Client ip port
1 127.0.0.1 57682
2 127.0.0.1 57688
3 127.0.0.1 57690
Enter Client Number, number of pings
2,6
Press enter to proceed


Server Network sting Tool
ping
traceroute
nslookup
ifconfig
dig
route
nmap
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.033 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.030 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.028 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.029 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.028 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.052 ms
--- 127.0.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5121ms
rtt min/avg/max/mdev = 0.028/0.033/0.052/0.008 ms

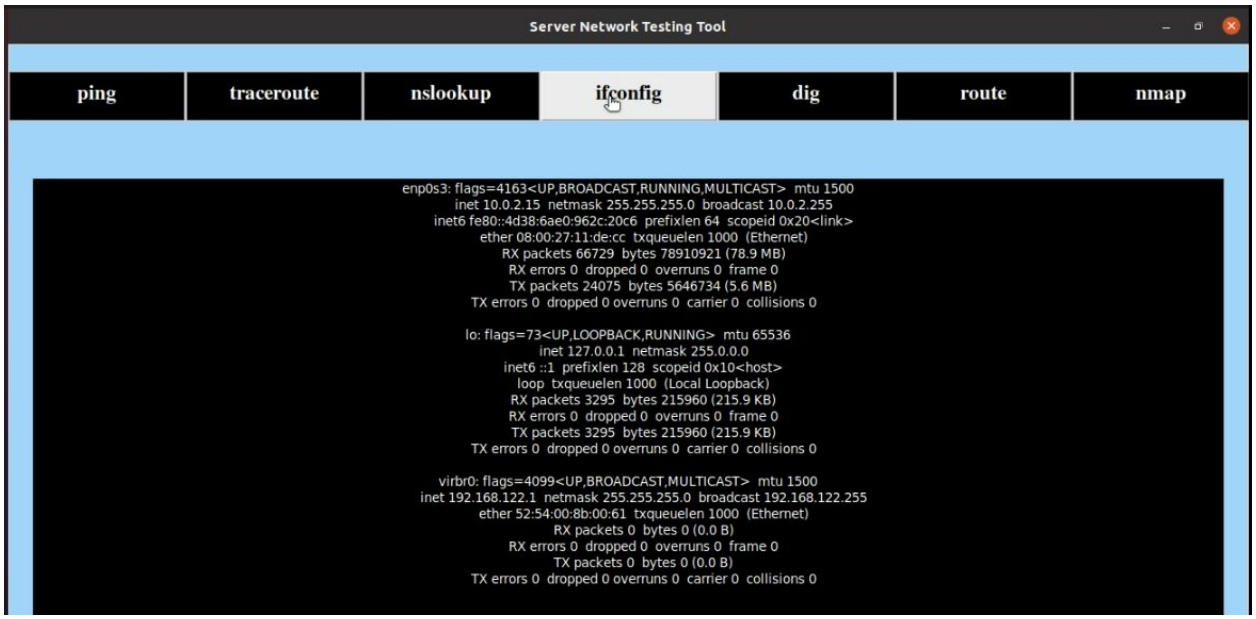
traceroute:


Custom traceroute
Client ip port
1 127.0.0.1 57682
2 127.0.0.1 57688
3 127.0.0.1 57690
Enter Client Number:
1
Press enter to proceed

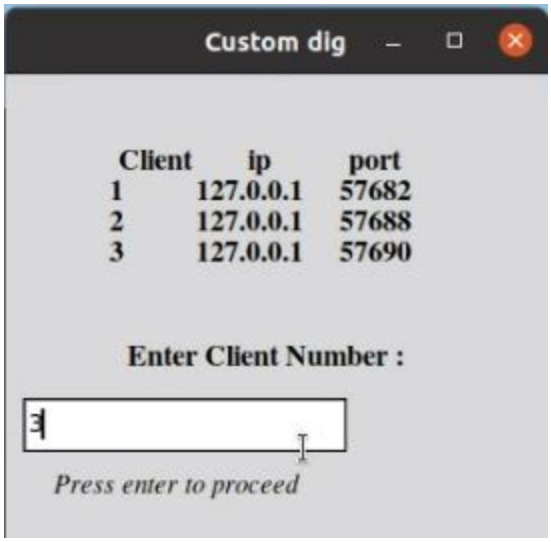

Server Network Testing Tool
ping
traceroute
nslookup
ifconfig
dig
route
nmap
myTraceRoute to 127.0.0.1 (127.0.0.1), 30 hops max.
1 0.012 ms 0.005 ms 0.005 ms 127.0.0.1

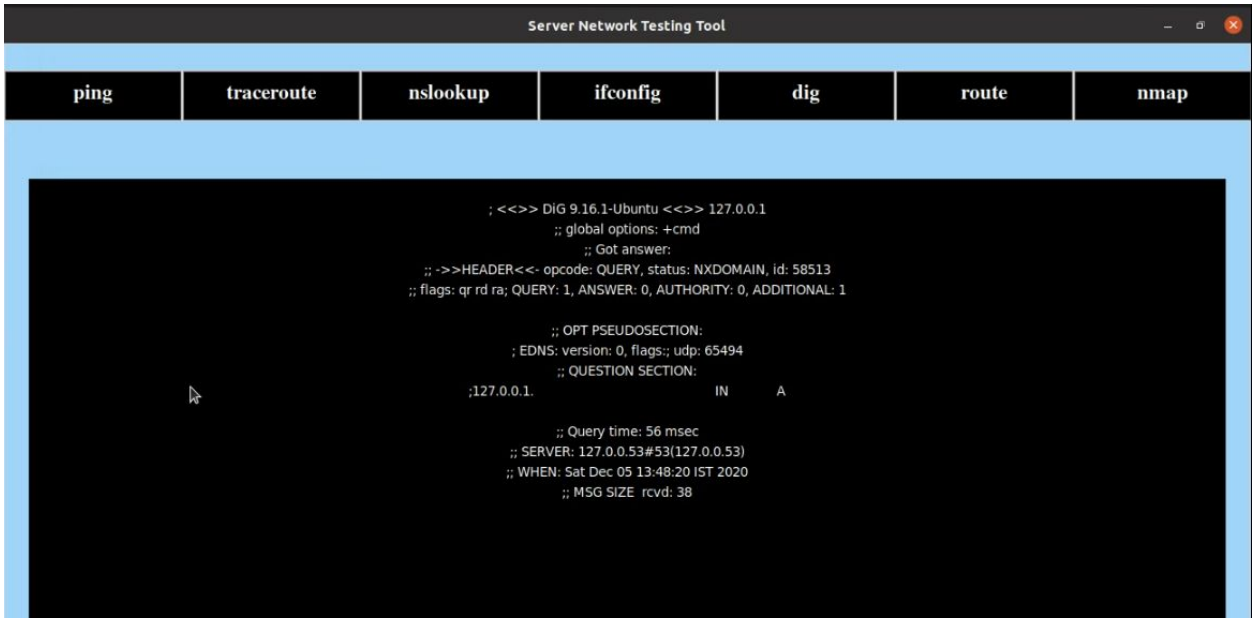
nslookup:

ifconfig:

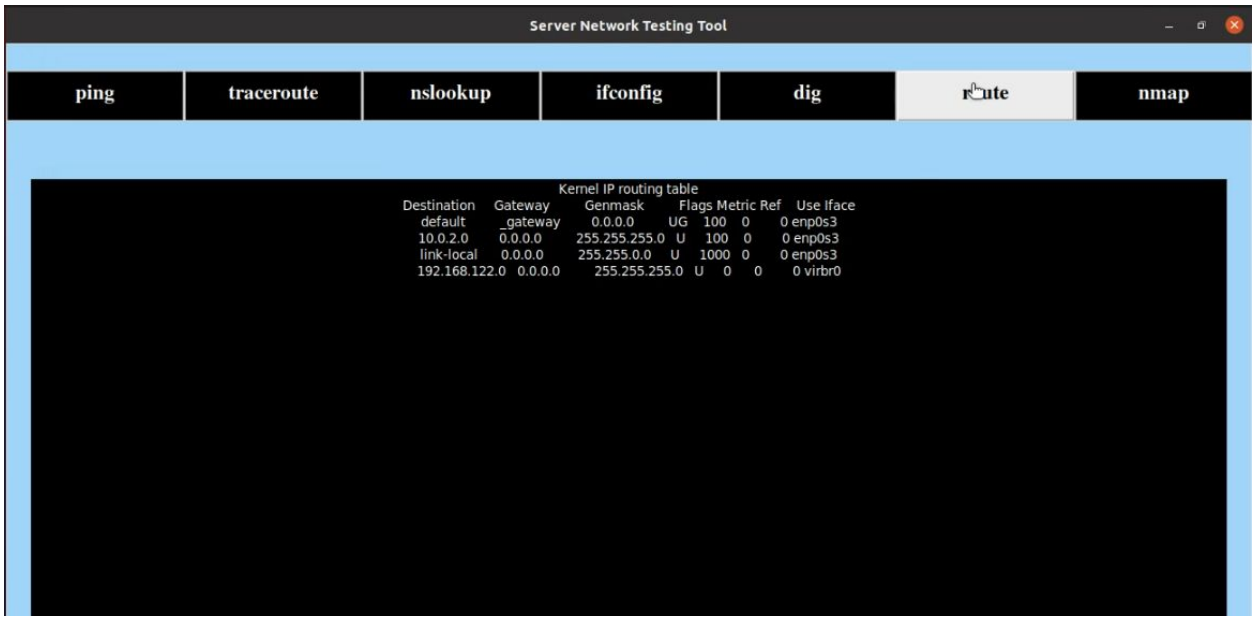


dig:

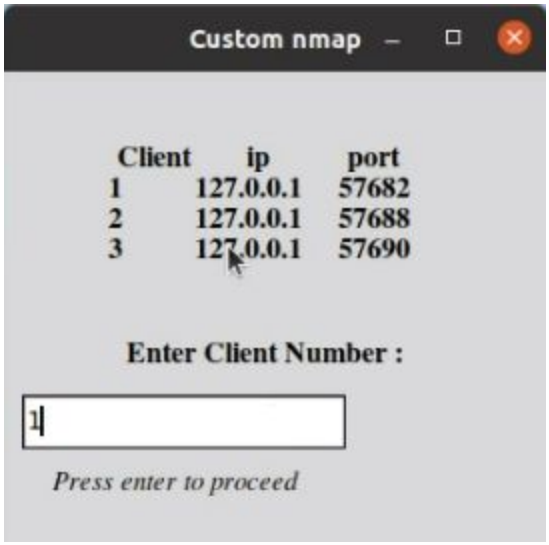

Custom dig
Client ip port
1 127.0.0.1 57682
2 127.0.0.1 57688
3 127.0.0.1 57690
Enter Client Number :
3
Press enter to proceed

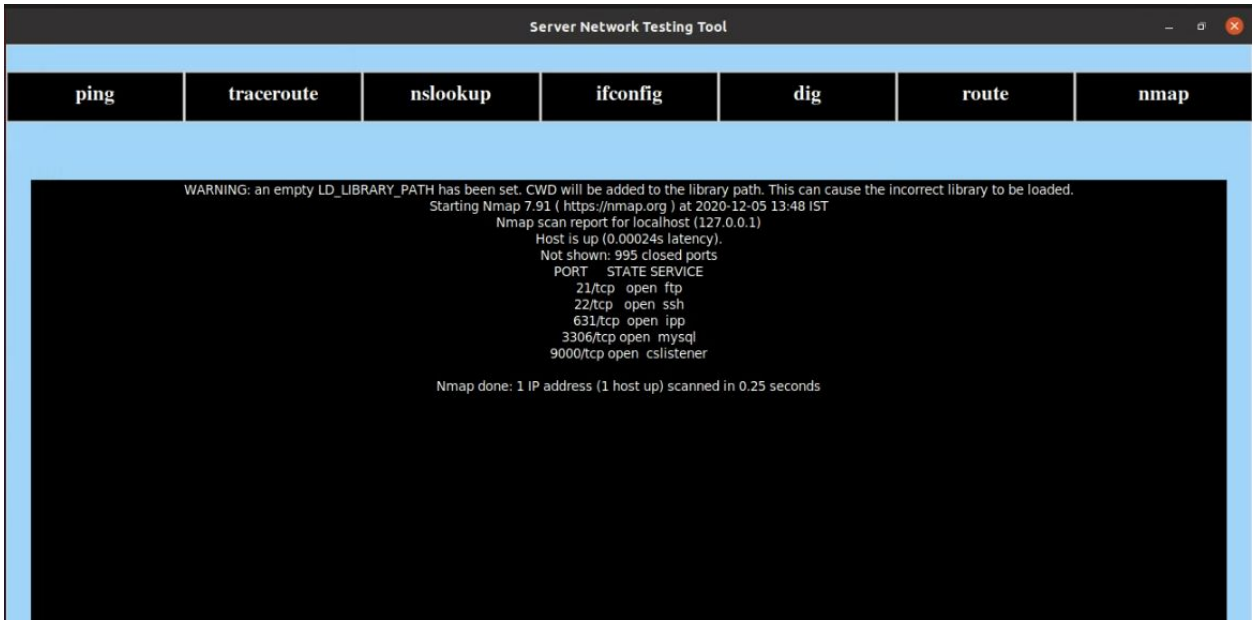

Server Network Testing Tool
ping
traceroute
nslookup
ifconfig
dig
route
nmap
; <<>> DiG 9.16.1-Ubuntu <<>> 127.0.0.1
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 58513
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;127.0.0.1. IN A
;; Query time: 56 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sat Dec 05 13:48:20 IST 2020
;; MSG SIZE rcvd: 38

route:


Server Network Testing Tool
ping
traceroute
nslookup
ifconfig
dig
route
nmap
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default _gateway 0.0.0.0 UG 100 0 0 enp0s3
10.0.2.0 0.0.0.0 255.255.255.0 U 100 0 0 enp0s3
link-local 0.0.0.0 255.255.0.0 U 1000 0 0 enp0s3
192.168.122.0 0.0.0.0 255.255.255.0 U 0 0 0 virbr0

nmap:


Custom nmap
Client ip port
1 127.0.0.1 57682
2 127.0.0.1 57688
3 127.0.0.1 57690
Enter Client Number :
1
Press enter to proceed

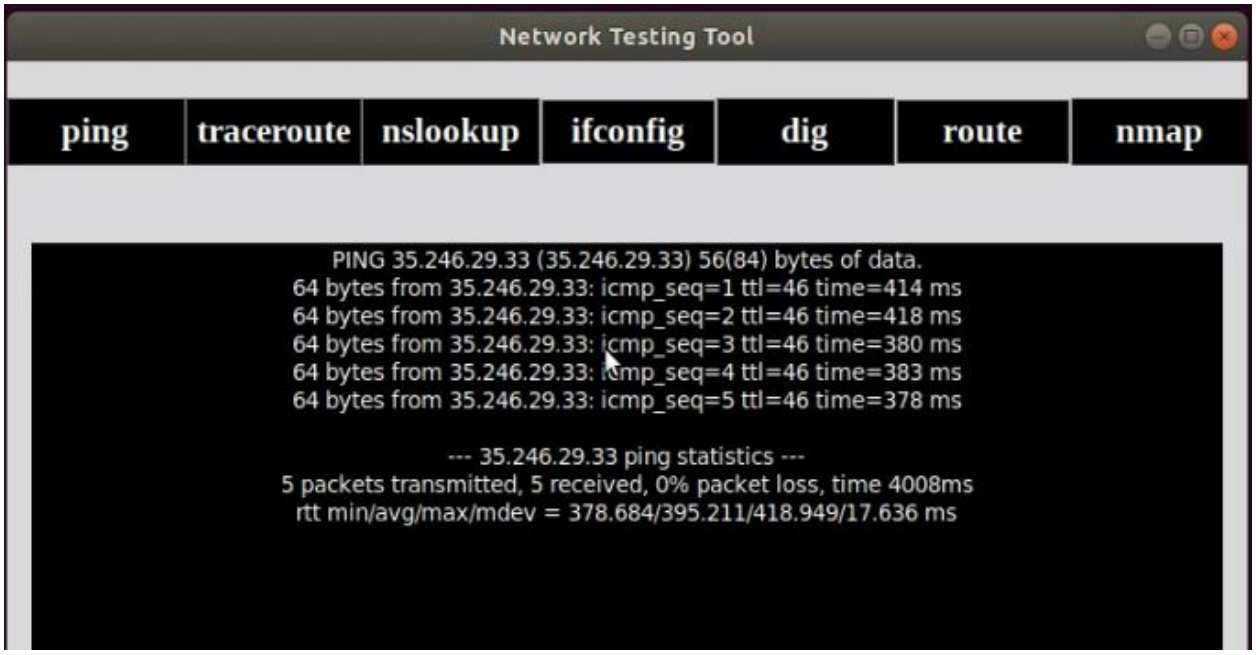

Server Network Testing Tool
ping
traceroute
nslookup
ifconfig
dig
route
nmap
WARNING: an empty LD_LIBRARY_PATH has been set. CWD will be added to the library path. This can cause the incorrect library to be loaded.
Starting Nmap 7.91 (https://nmap.org) at 2020-12-05 13:48 IST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00024s latency).
Not shown: 995 closed ports
PORT STATE SERVICE
21/tcp open ftp
22/tcp open ssh
631/tcp open ipp
3306/tcp open mysql
9000/tcp open cslistener
Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds

PUBLIC HOSTING:

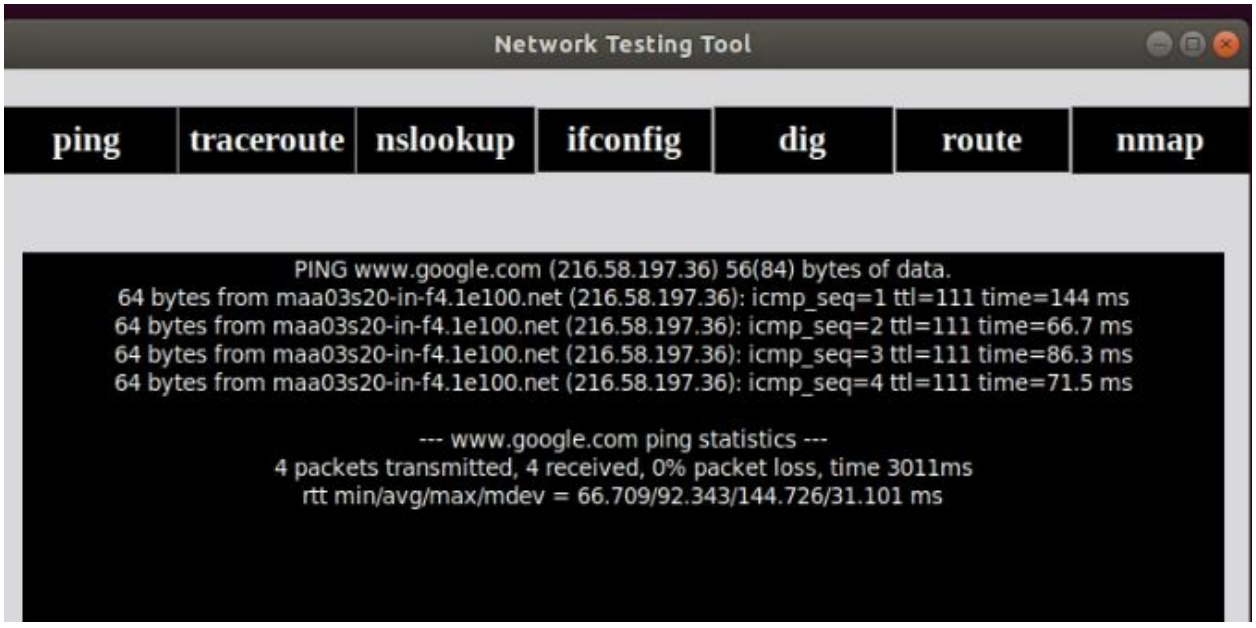
SERVER:

CLIENT:

Ping:

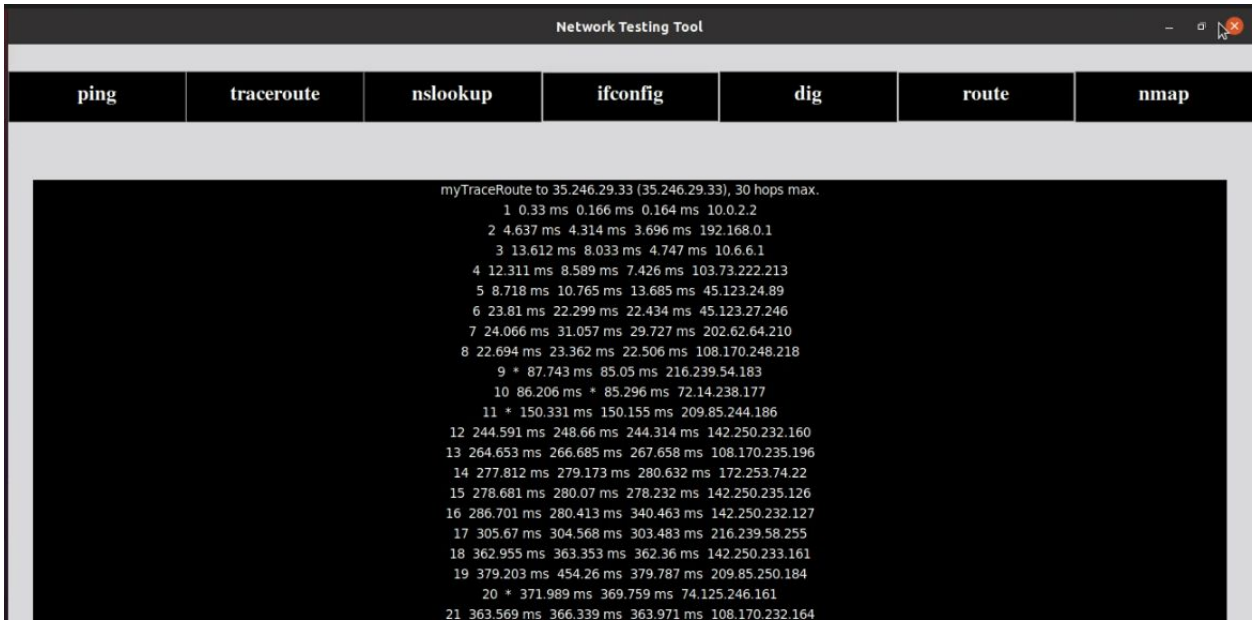


www.google.com

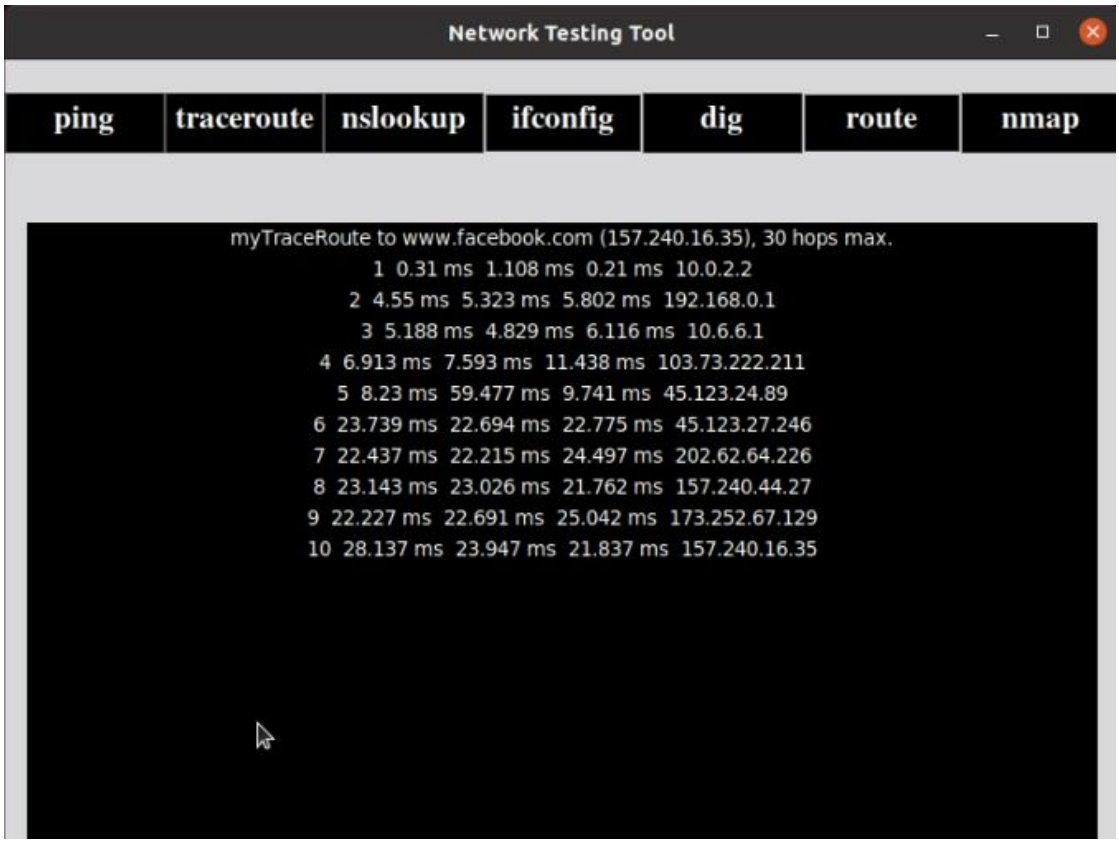


Traceroute:

server

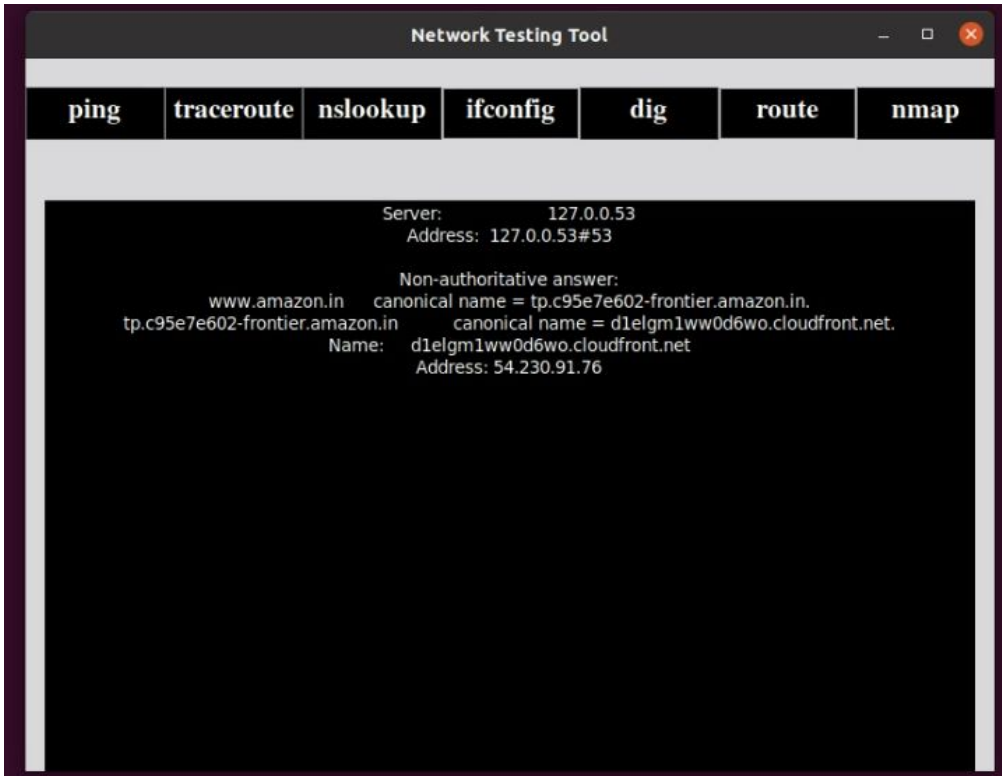


www.facebook.com

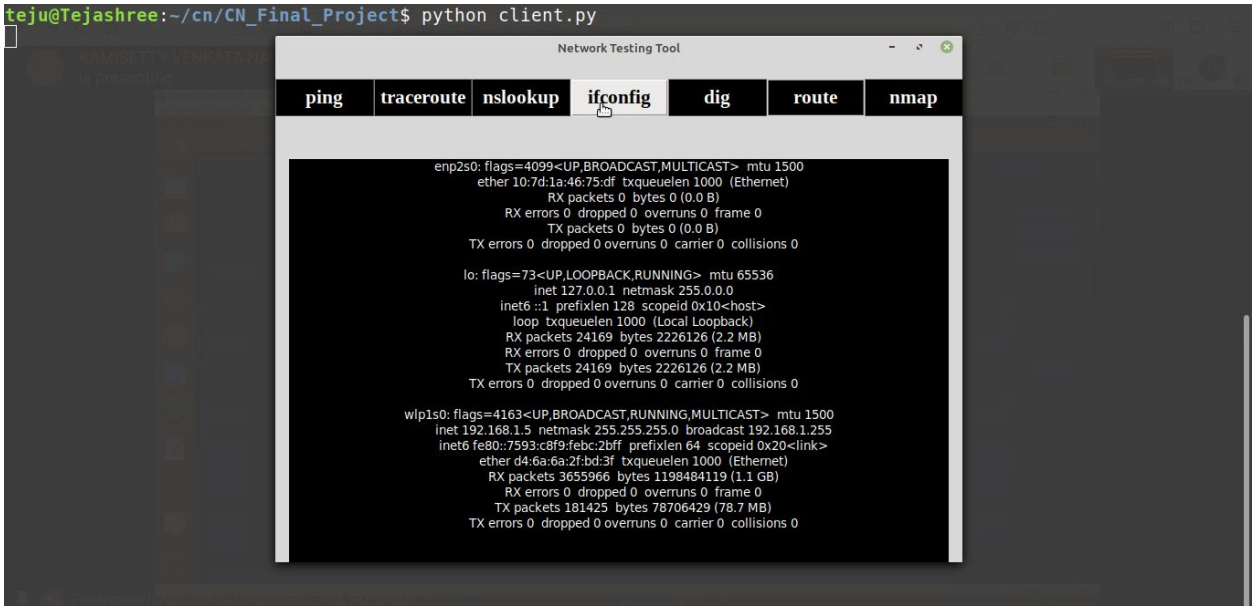


nslookup:

www.amazon.in

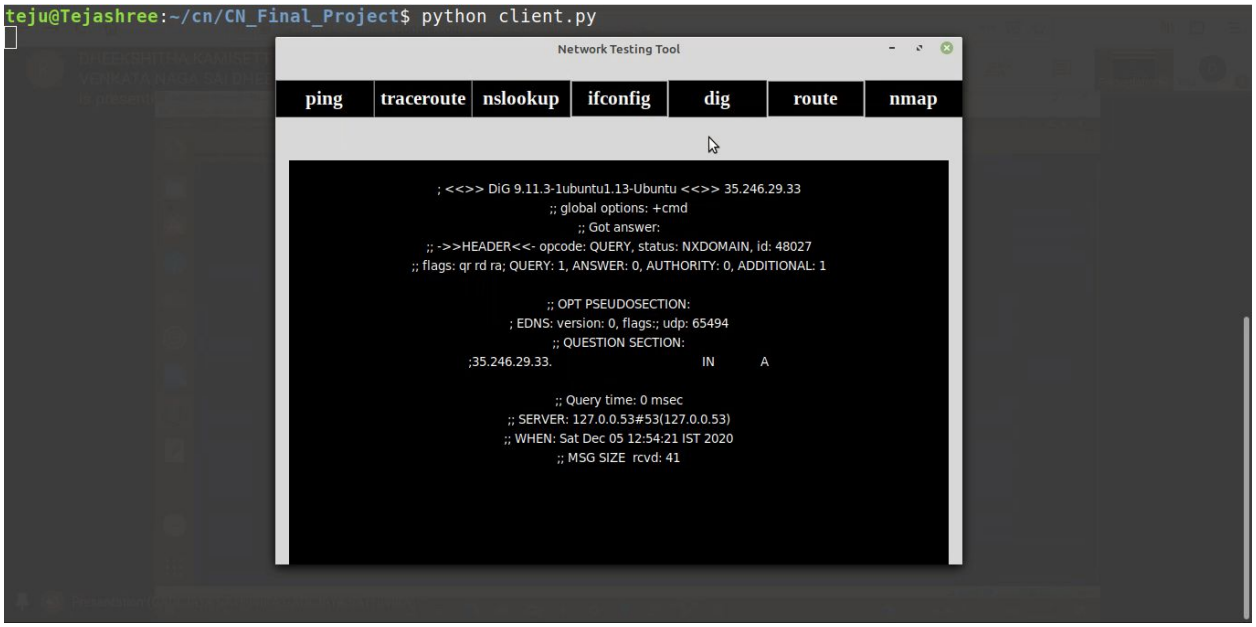


Ifconfig:

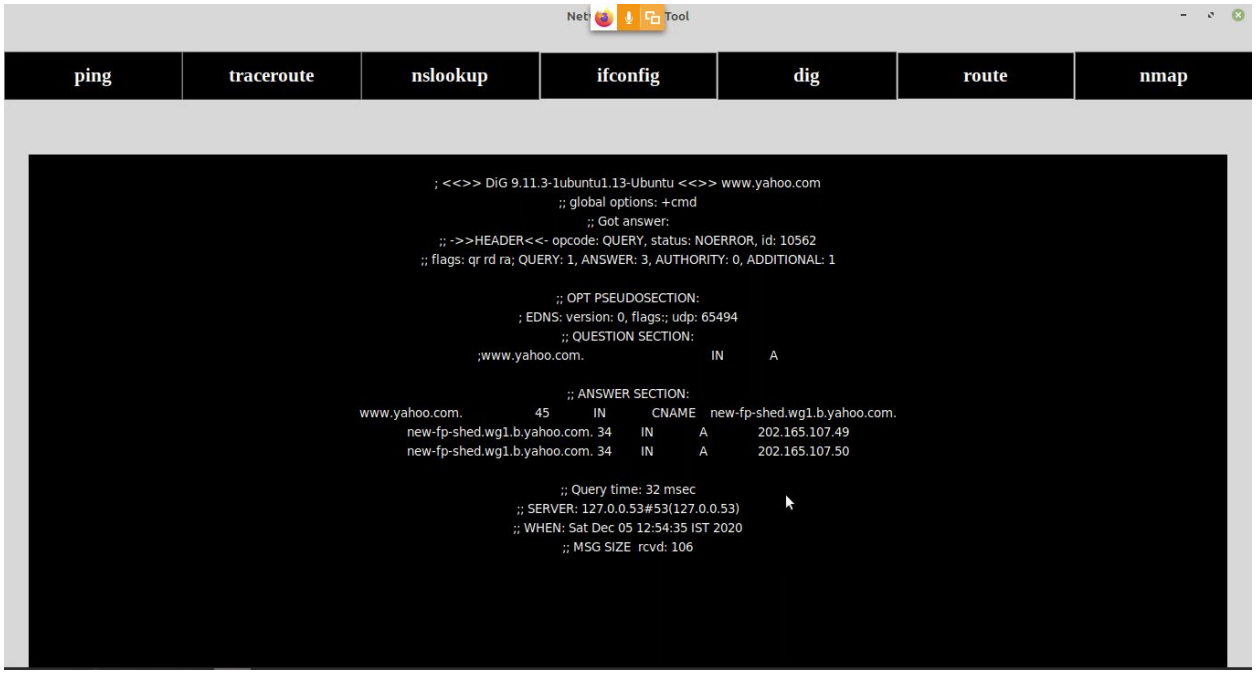


dig:

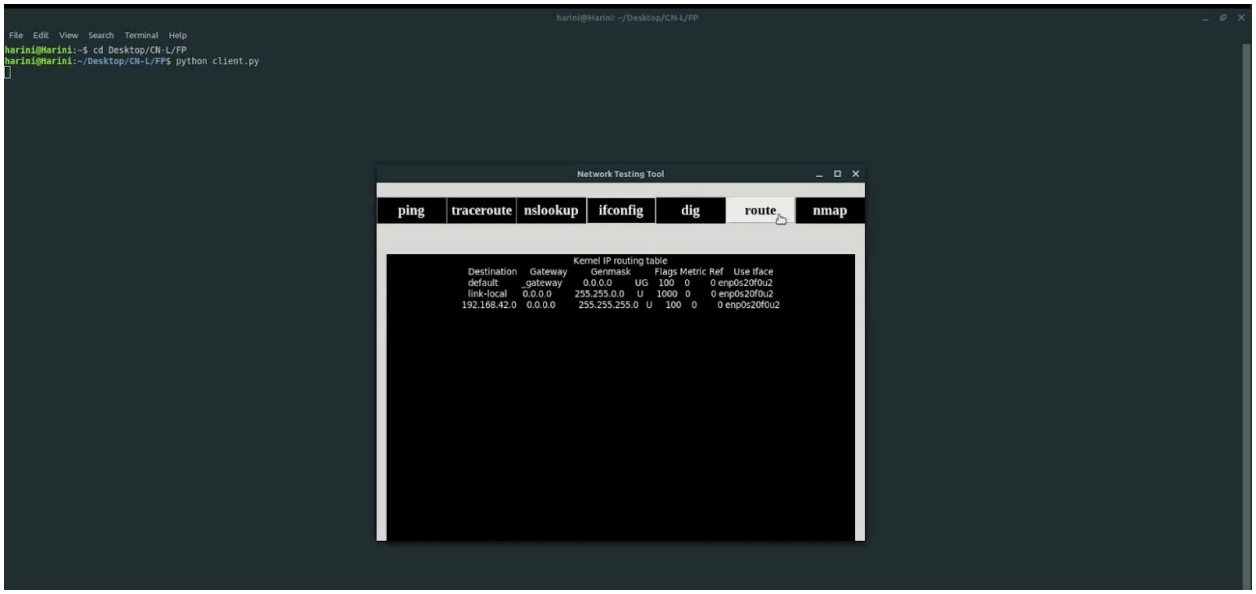
server



www.yahoo.com



route:


harini@Harini: ~/Desktop/CN-L/FP
File Edit View Search Terminal Help
harini@Harini:~$ cd Desktop/CN-L/FP
harini@Harini:~/Desktop/CN-L/FP$ python client.py
Network Testing Tool
ping
traceroute
nslookup
ifconfig
dig
route
nmap
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default _gateway 0.0.0.0 UG 100 0 0 enp0s20f0u2
link-local 0.0.0.0 255.255.0.0 U 1000 0 0 enp0s20f0u2
192.168.42.0 0.0.0.0 255.255.255.0 U 100 0 0 enp0s20f0u2

Nmap:

server

www.google.com