

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()

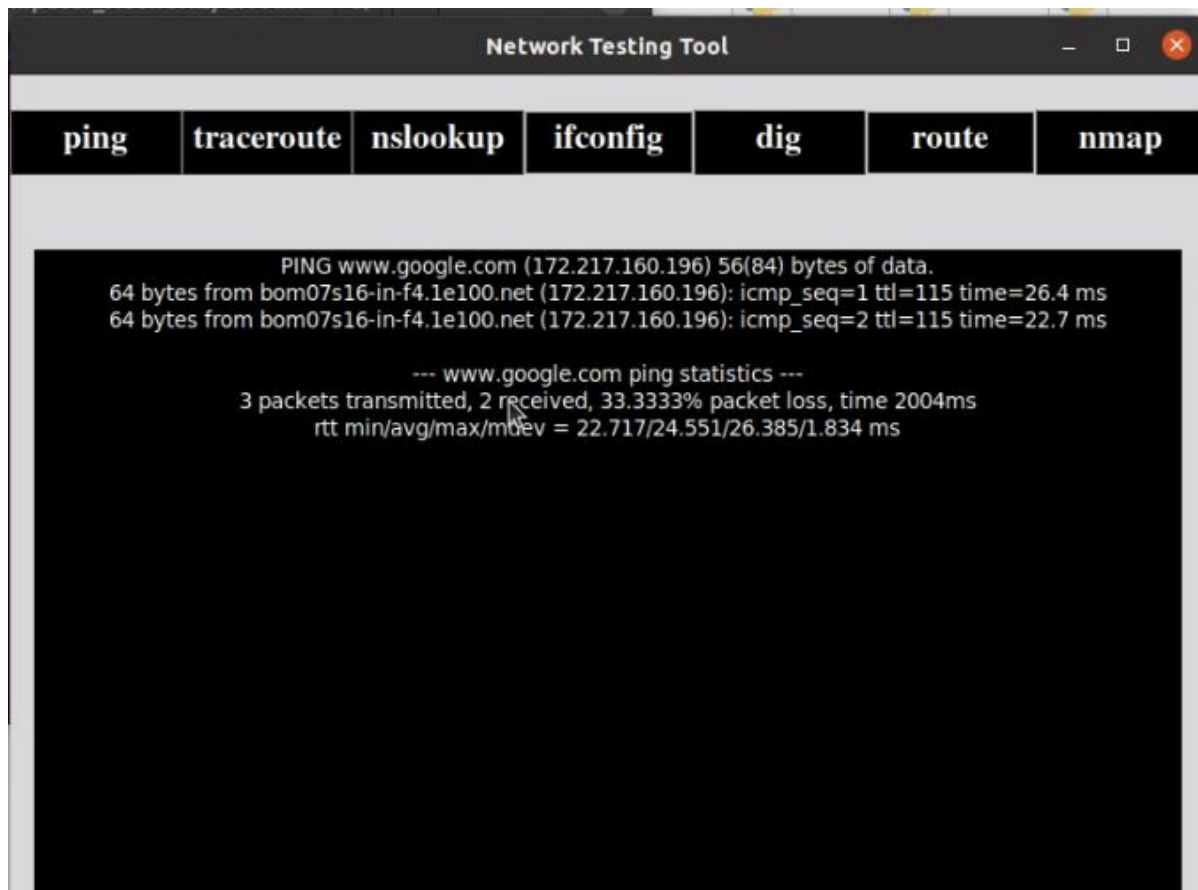
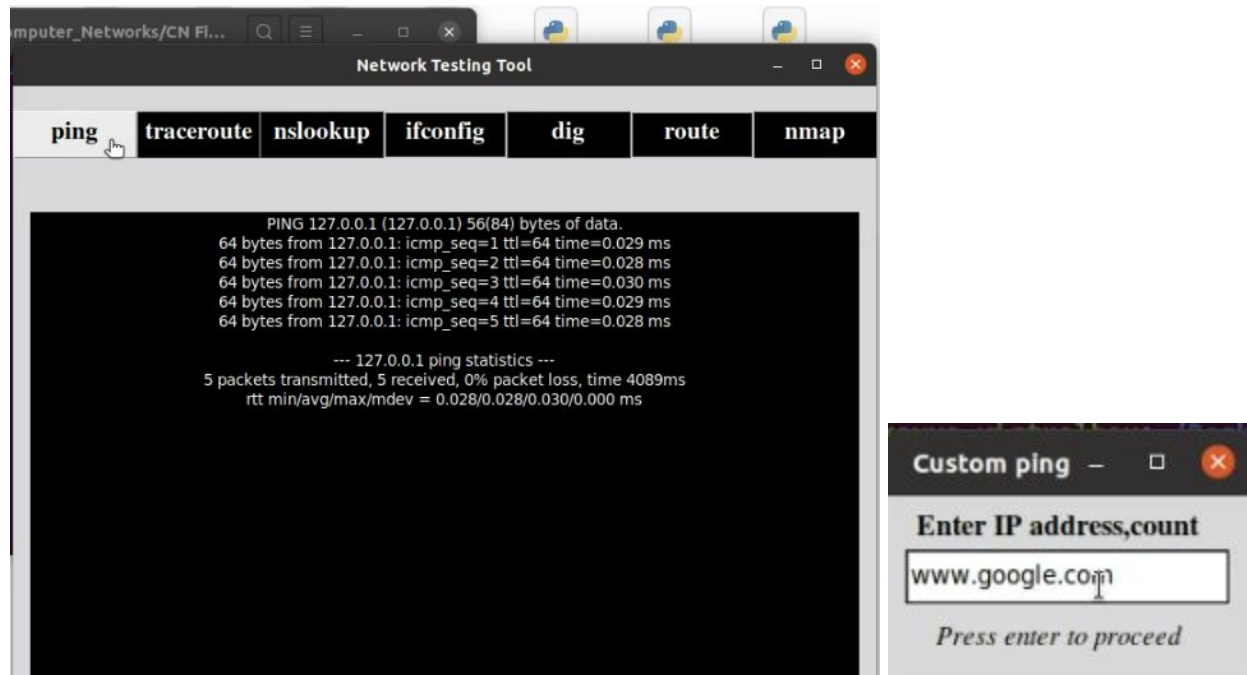
```

OUTPUT:

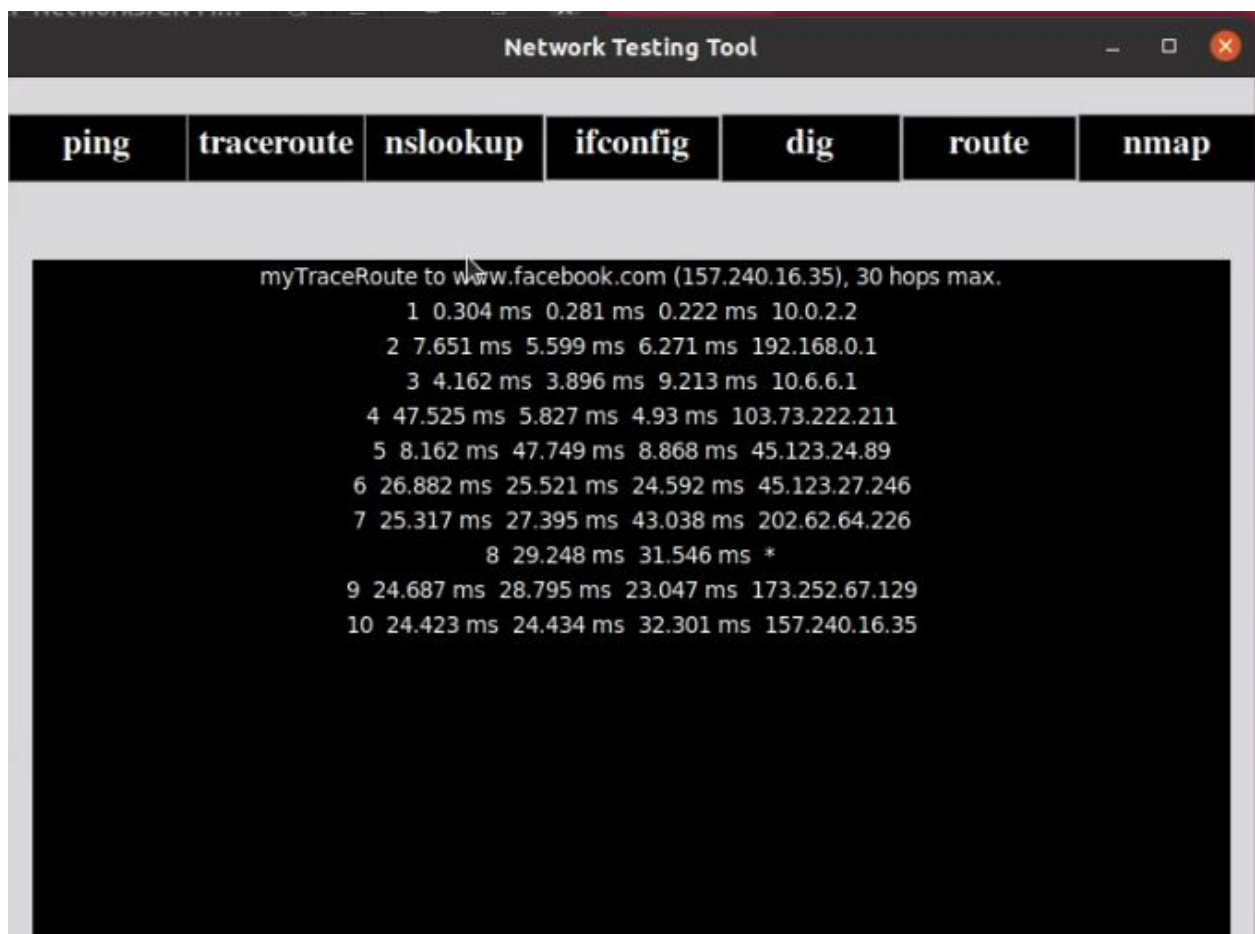
LOOPBACK IP:

CLIENT:

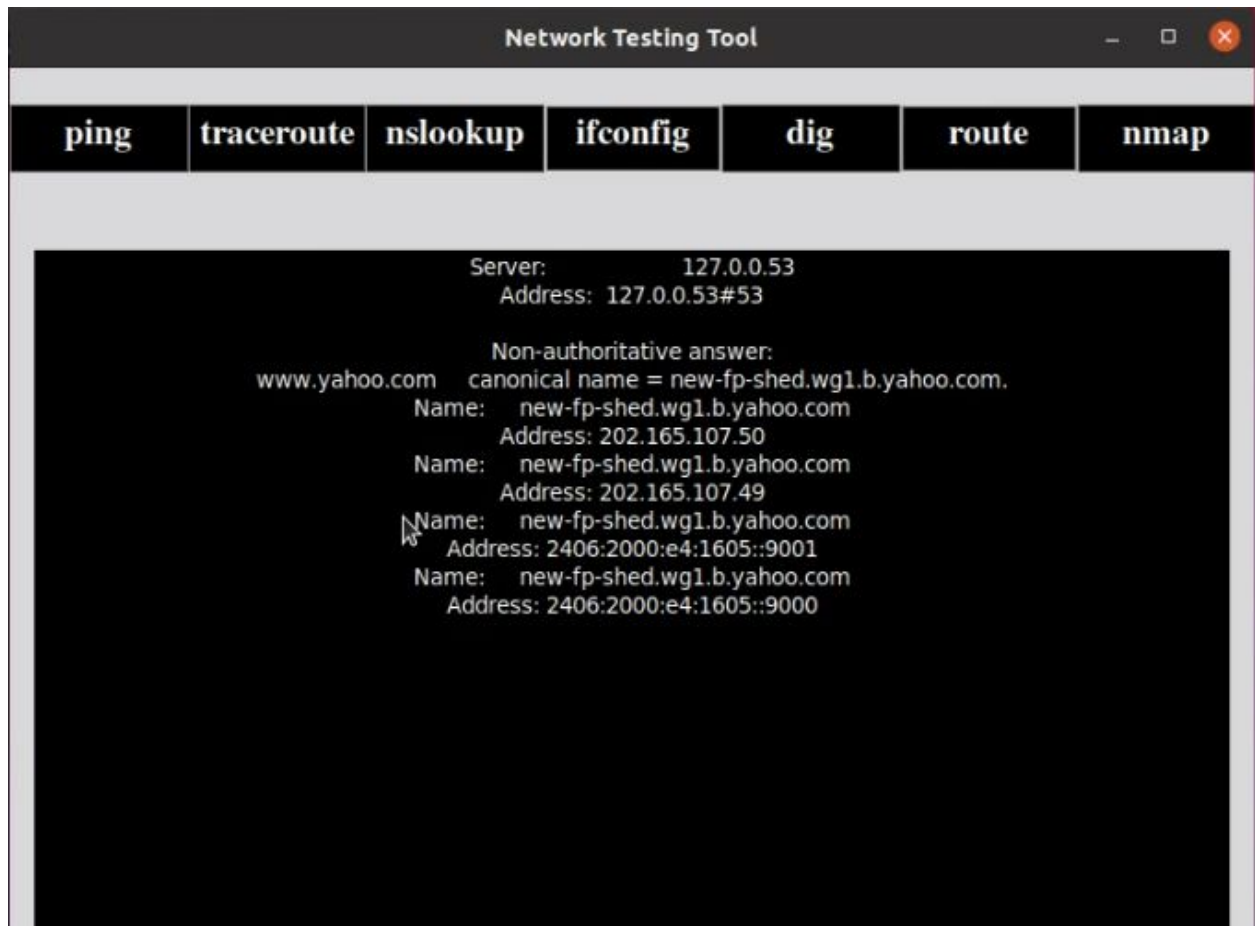
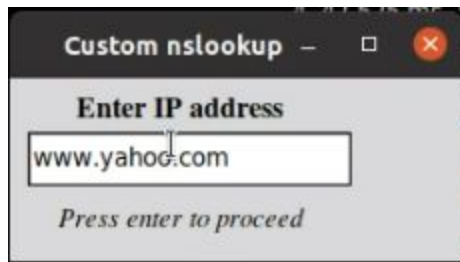
Ping:



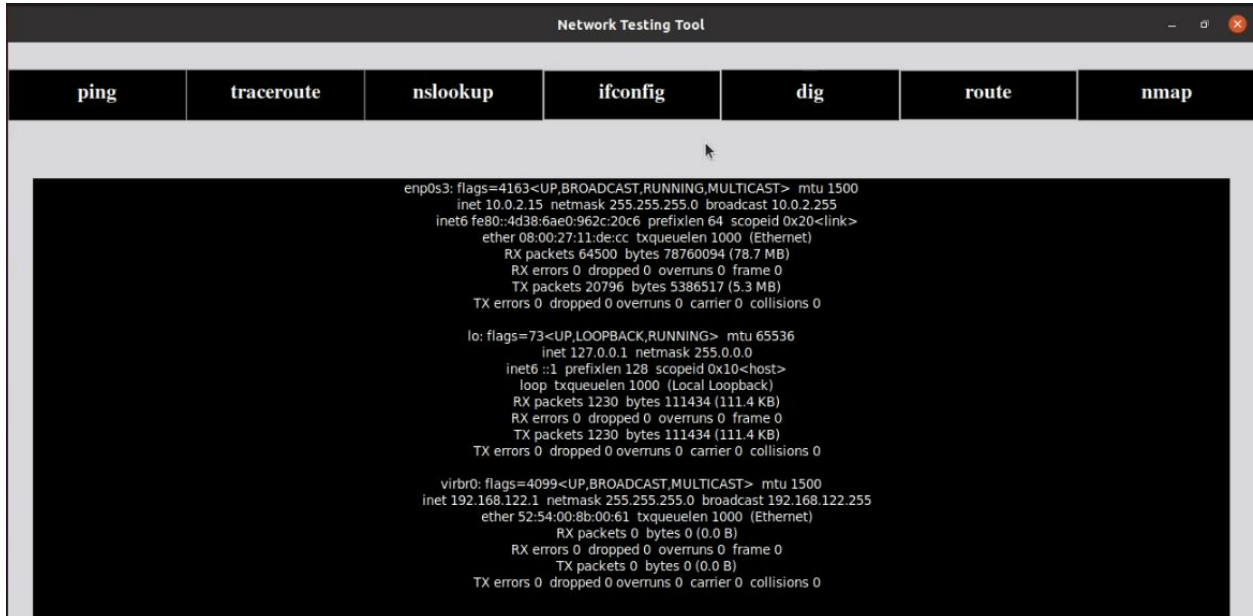
Traceroute:



nslookup:



ifconfig:



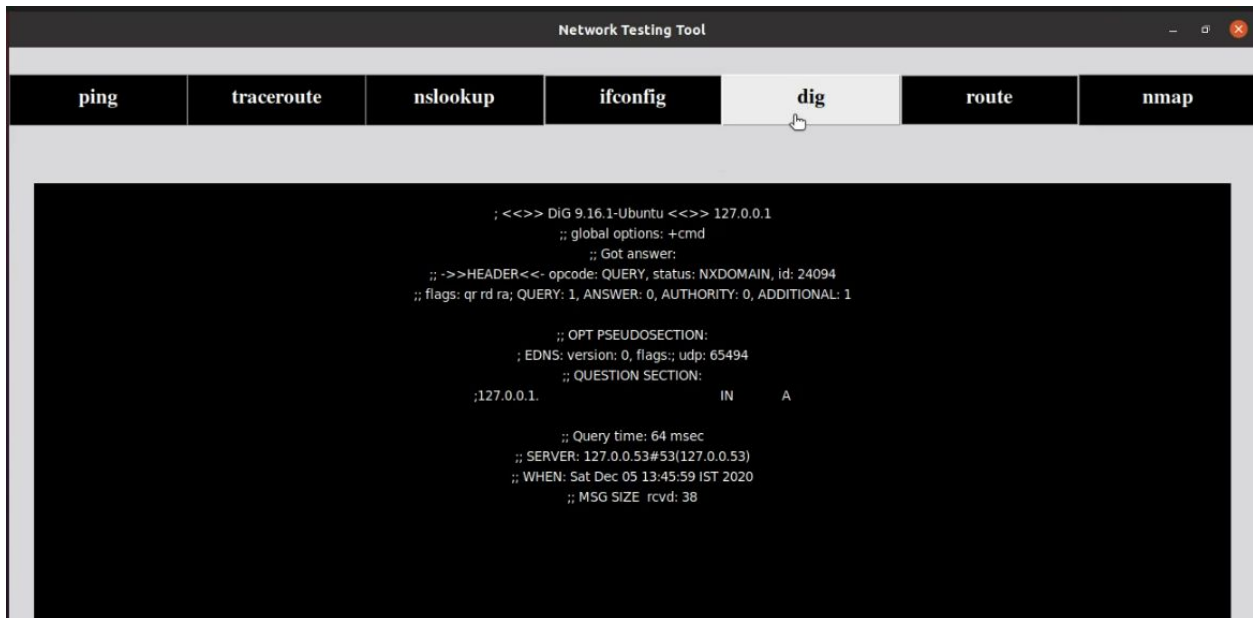
The screenshot shows the 'Network Testing Tool' window with the 'ifconfig' tab selected. The terminal displays the configuration for three network interfaces: enp0s3, lo, and virbr0. Each interface shows its flags, MTU, IP address, netmask, broadcast address, and various statistics like RX and TX packets, bytes, errors, and collisions.

```
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
inet6 fe80::4d38:6ae0:962c:20c6 prefixlen 64 scopeid 0x20<link>
ether 08:00:27:11:d5:cc txqueuelen 1000 (Ethernet)
RX packets 64500 bytes 78760094 (78.7 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 20796 bytes 5386517 (5.3 MB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 1230 bytes 111434 (111.4 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 1230 bytes 111434 (111.4 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255
ether 52:54:00:8b:00:61 txqueuelen 1000 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

dig:



The screenshot shows the 'Network Testing Tool' window with the 'dig' tab selected. The terminal displays the output of a dig command for the IP address 127.0.0.1. The output includes global options, the query type (QUERY), the status (NXDOMAIN), the id (24094), the flags (qr rd ra), the query (1), the answer (0), the authority (0), the additional (1), the question section (127.0.0.1 IN A), the query time (64 msec), the server (127.0.0.53), the when (Sat Dec 05 13:45:59 IST 2020), and the message size (38).

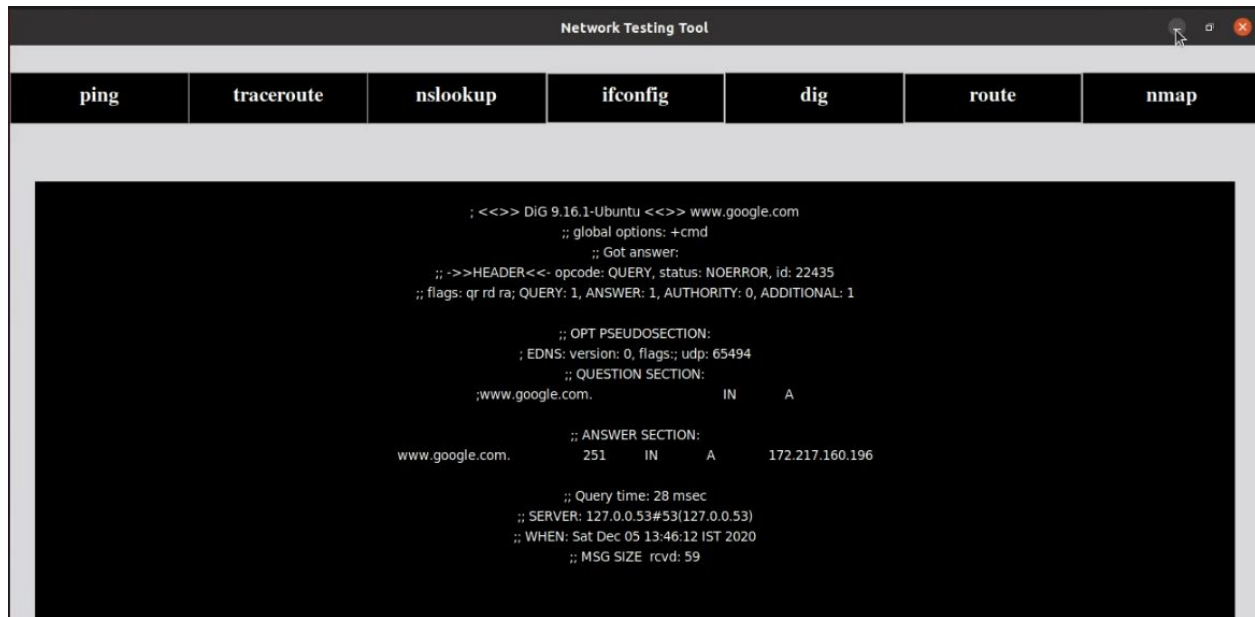
```
; <<<>> DiG 9.16.1-Ubuntu <<<>> 127.0.0.1
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 24094
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: udp: 65494
;; QUESTION SECTION:
;127.0.0.1. IN A

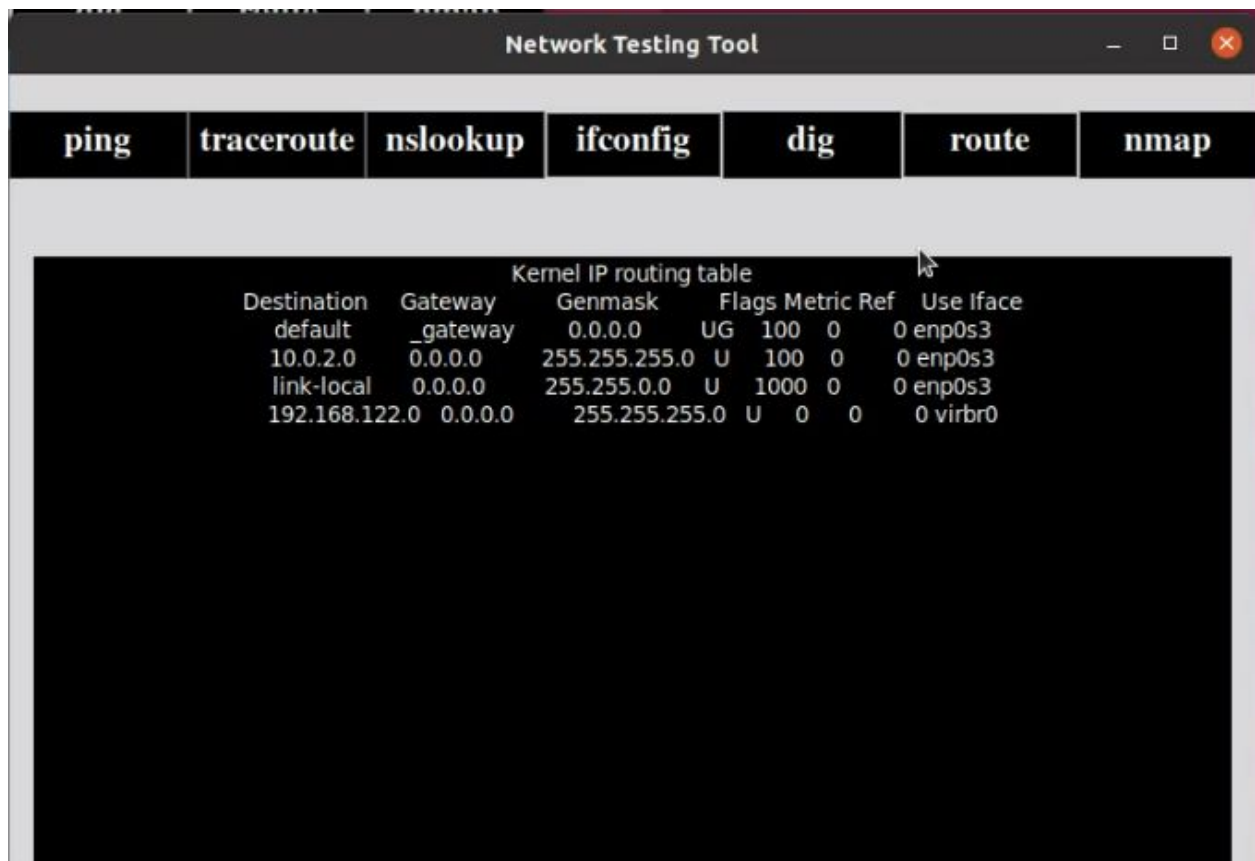
;; Query time: 64 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sat Dec 05 13:45:59 IST 2020
;; MSG SIZE rcvd: 38
```



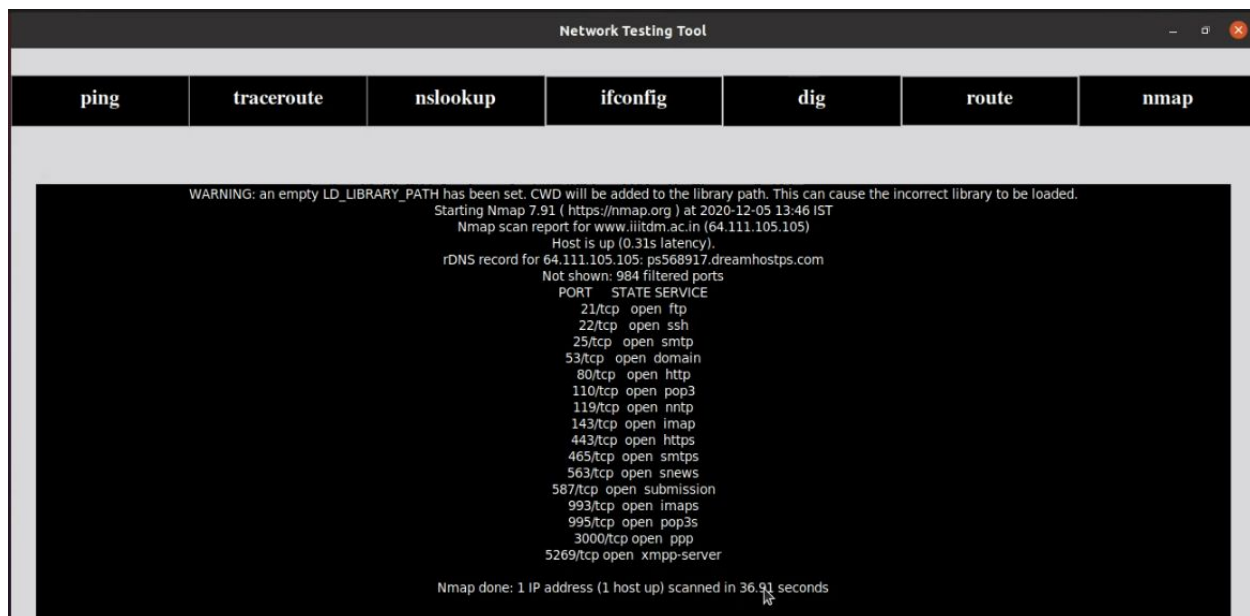
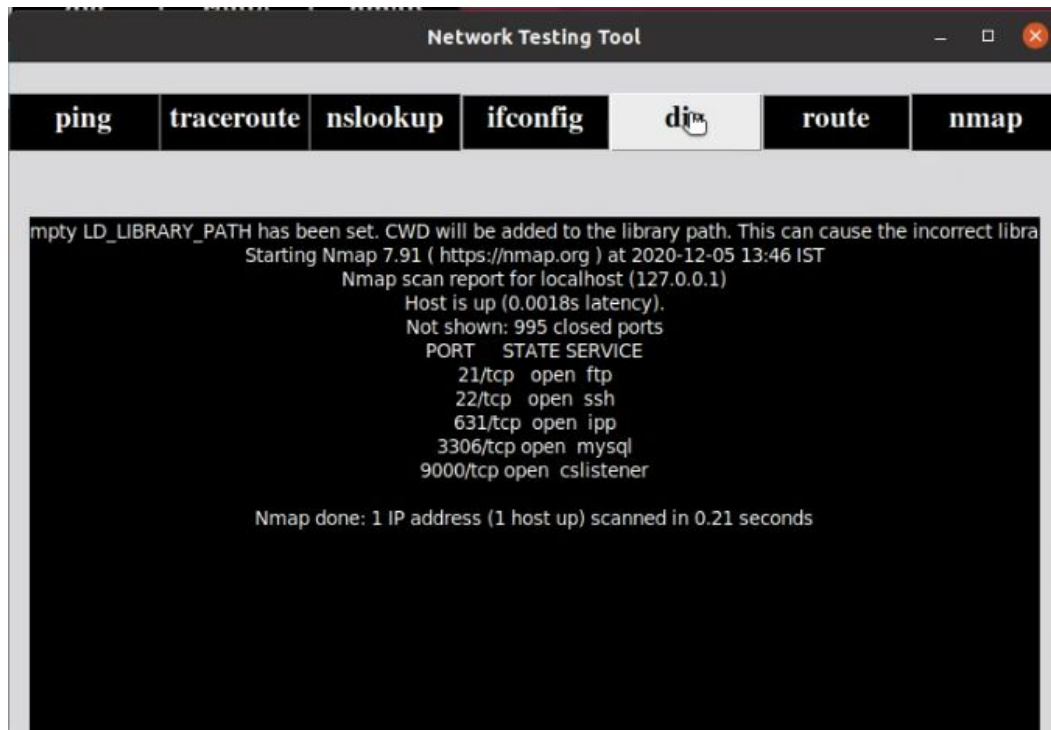
The screenshot shows a small window titled 'Custom dig'. It has a text input field labeled 'Enter IP address' containing the text 'www.google.com'. Below the input field, there is a button labeled 'Press enter to proceed'.



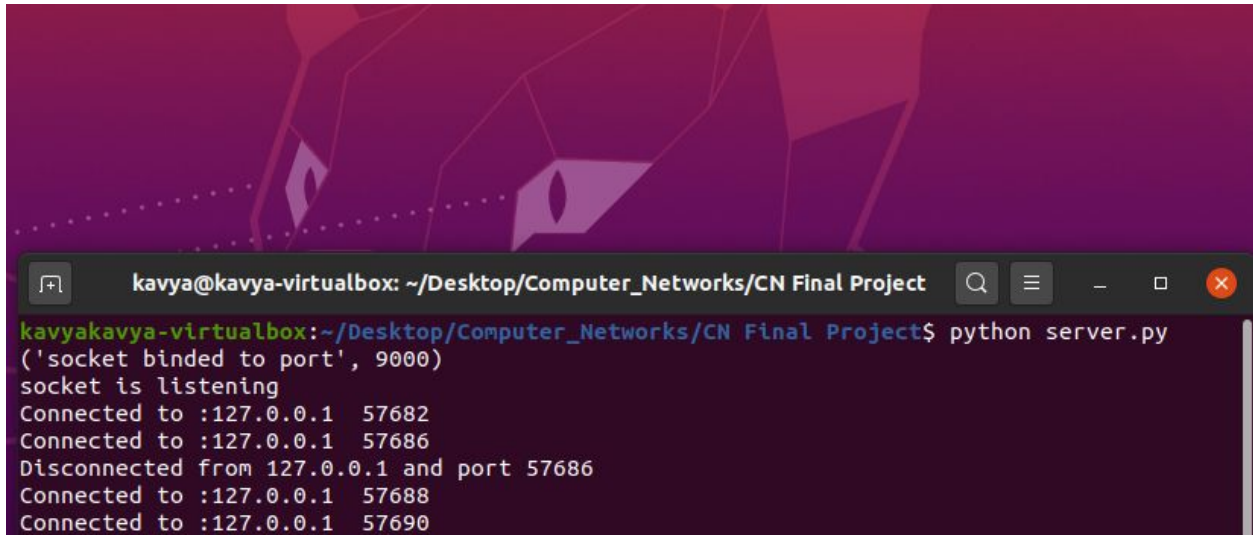
route:



Nmap:



SERVER:

A terminal window with a purple and pink geometric background. The title bar reads 'kavya@kavya-virtualbox: ~/Desktop/Computer_Networks/CN Final Project'. The terminal text shows the execution of 'python server.py', which outputs: ('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, and Connected to :127.0.0.1 57690.

```
kavya@kavya-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

Press enter to proceed

Server Network Testing Tool

ping	tracert	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:

Press enter to proceed

Server Network Testing Tool

ping

traceroute

nslookup

ifconfig

dig

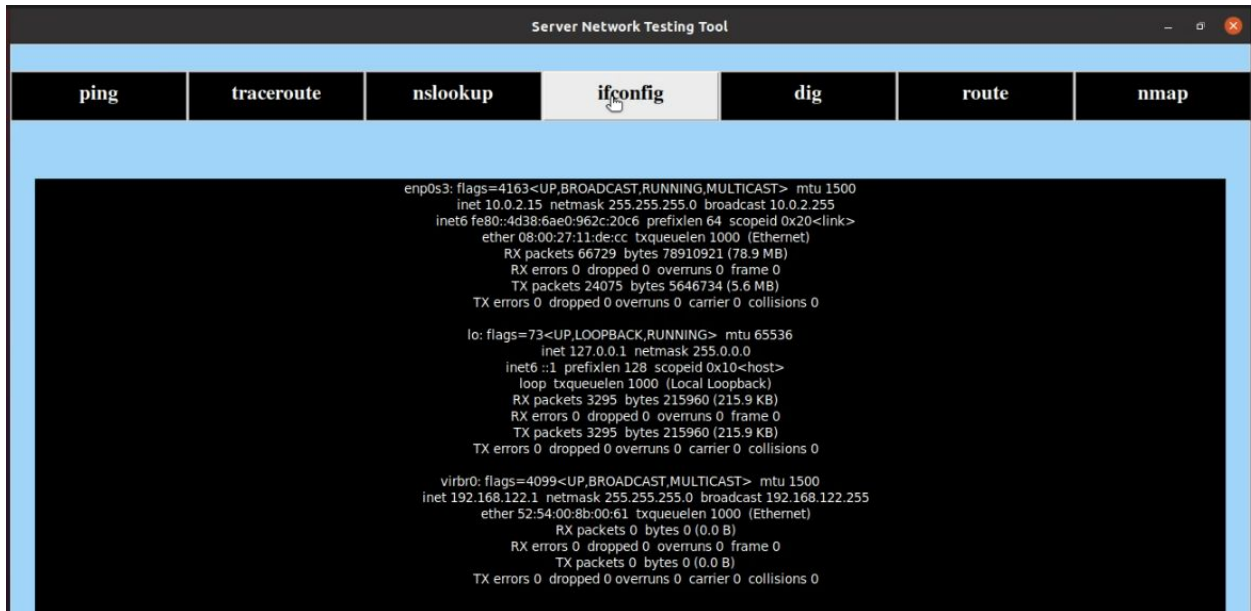
route

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:



The screenshot shows a window titled "Server Network Testing Tool" with a menu bar containing "ping", "traceroute", "nslookup", "ifconfig", "dig", "route", and "nmap". The "ifconfig" tab is selected, and the output of the command is displayed in a black terminal window. The output shows the configuration for three network interfaces: enp0s3, lo, and virbr0.

```
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
inet6 fe80::4d38:6ae0:962c:20c6 prefixlen 64 scopeid 0x20<link>
ether 08:00:27:11:de:cc txqueuelen 1000 (Ethernet)
RX packets 66729 bytes 78910921 (78.9 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 24075 bytes 5646734 (5.6 MB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 3295 bytes 215960 (215.9 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 3295 bytes 215960 (215.9 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255
ether 52:54:00:8b:00:61 txqueuelen 1000 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

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 :

Press enter to proceed

Server Network Testing Tool

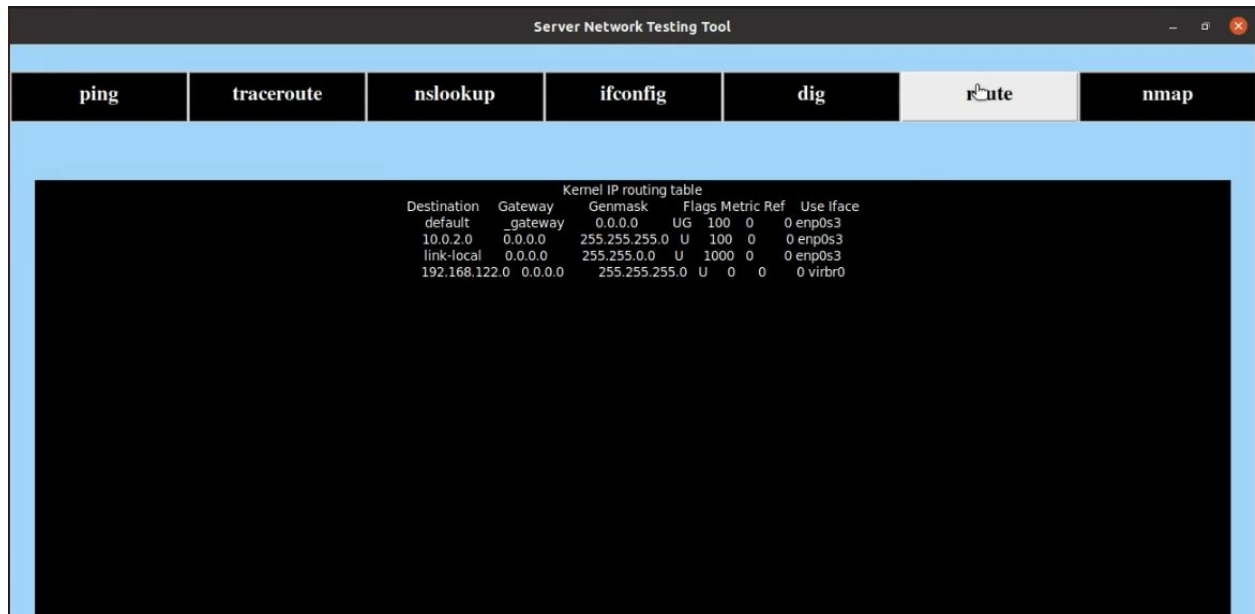
ping traceroute nslookup ifconfig dig route nmap

```
; <<>> 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:



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 :

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
```

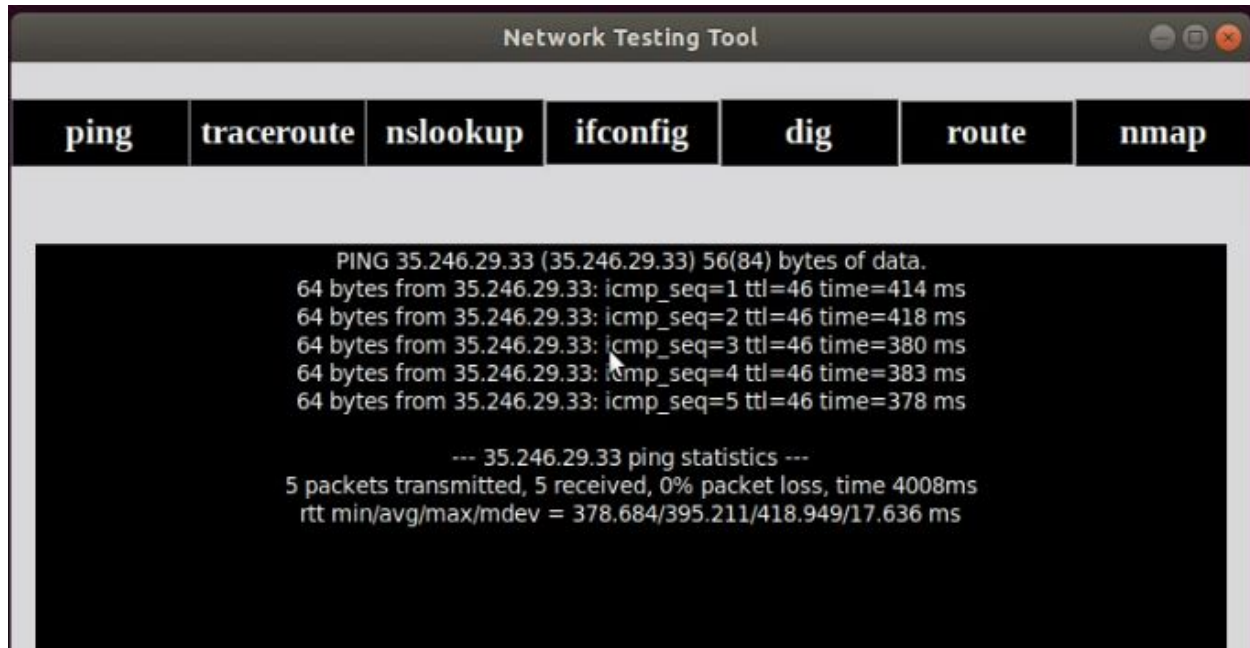

PUBLIC HOSTING:

SERVER:

```
sanjanadara1701@instance-1: ~ - Google Chrome
ssh.cloud.google.com/projects/seraphic-plexus-289709/zones/europe-west2-a/instances/instance-1?useAdminProxy=true&authuser=0&hl=en_US&projectNumber=190327004916
sanjanadara1701@instance-1:~$ python server.py
('socket binded to port', 9000)
socket is listening
Connected to :157.48.207.203 50370
Connected to :106.208.49.99 4334
Connected to :103.73.222.213 55339
Connected to :117.209.171.84 60031
Connected to :103.232.131.194 33036
Disconnected from 103.73.222.213 and port 55339
Disconnected from 157.48.207.203 and port 50370
Disconnected from 117.209.171.84 and port 60031
Disconnected from 103.232.131.194 and port 33036
Connected to :103.73.222.213 55341
Disconnected from 106.208.49.99 and port 4334
]
```

CLIENT:

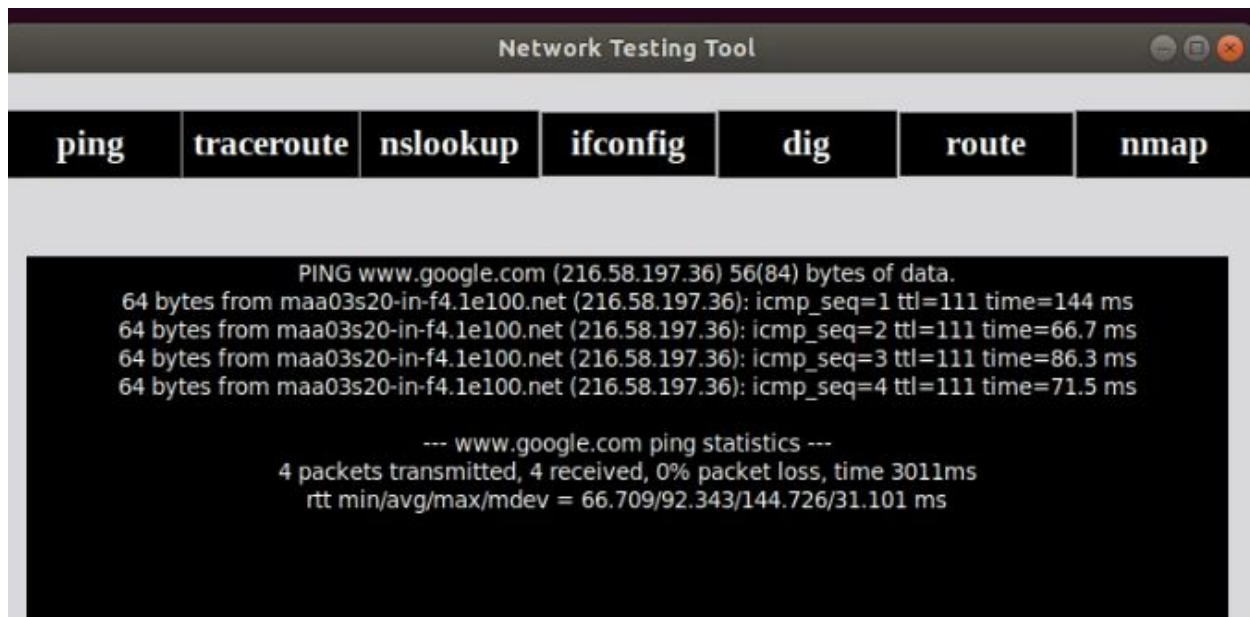
Ping:



The screenshot shows a window titled "Network Testing Tool" with a menu bar containing "ping", "traceroute", "nslookup", "ifconfig", "dig", "route", and "nmap". The "ping" button is selected. The main area displays the following text:

```
PING 35.246.29.33 (35.246.29.33) 56(84) bytes of data.  
64 bytes from 35.246.29.33: icmp_seq=1 ttl=46 time=414 ms  
64 bytes from 35.246.29.33: icmp_seq=2 ttl=46 time=418 ms  
64 bytes from 35.246.29.33: icmp_seq=3 ttl=46 time=380 ms  
64 bytes from 35.246.29.33: icmp_seq=4 ttl=46 time=383 ms  
64 bytes from 35.246.29.33: icmp_seq=5 ttl=46 time=378 ms  
  
--- 35.246.29.33 ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 4008ms  
rtt min/avg/max/mdev = 378.684/395.211/418.949/17.636 ms
```

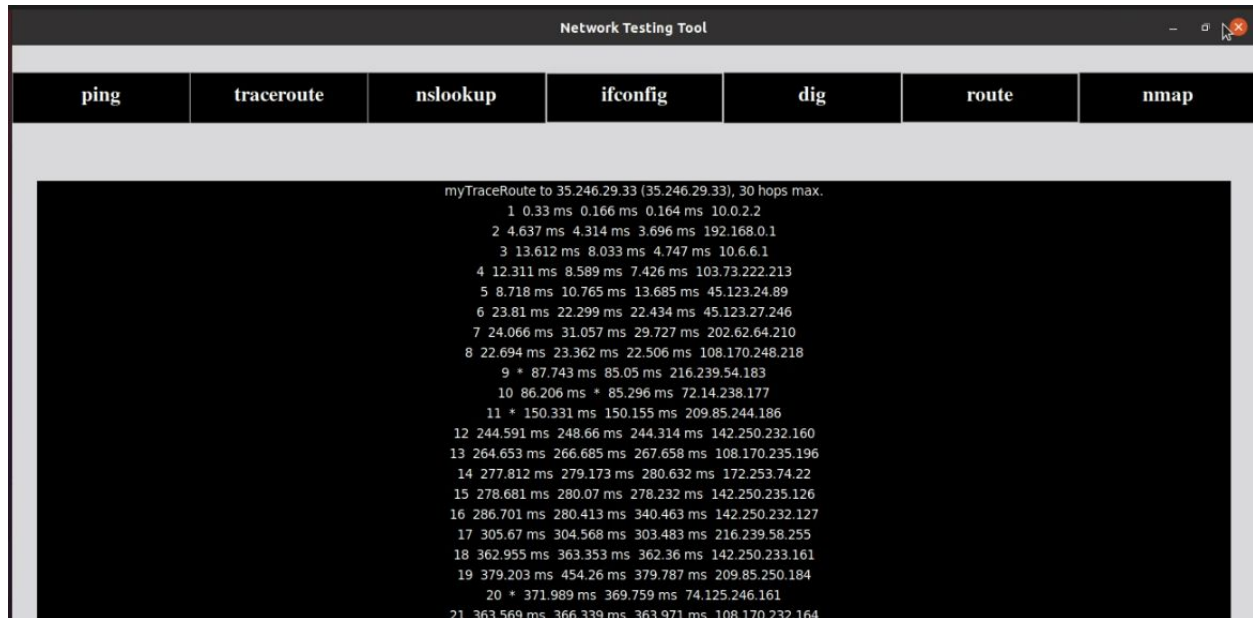
www.google.com



The screenshot shows a window titled "Network Testing Tool" with a menu bar containing "ping", "traceroute", "nslookup", "ifconfig", "dig", "route", and "nmap". The "ping" button is selected. The main area displays the following text:

```
PING www.google.com (216.58.197.36) 56(84) bytes of data.  
64 bytes from maa03s20-in-f4.1e100.net (216.58.197.36): icmp_seq=1 ttl=111 time=144 ms  
64 bytes from maa03s20-in-f4.1e100.net (216.58.197.36): icmp_seq=2 ttl=111 time=66.7 ms  
64 bytes from maa03s20-in-f4.1e100.net (216.58.197.36): icmp_seq=3 ttl=111 time=86.3 ms  
64 bytes from maa03s20-in-f4.1e100.net (216.58.197.36): icmp_seq=4 ttl=111 time=71.5 ms  
  
--- www.google.com ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3011ms  
rtt min/avg/max/mdev = 66.709/92.343/144.726/31.101 ms
```

Traceroute: server

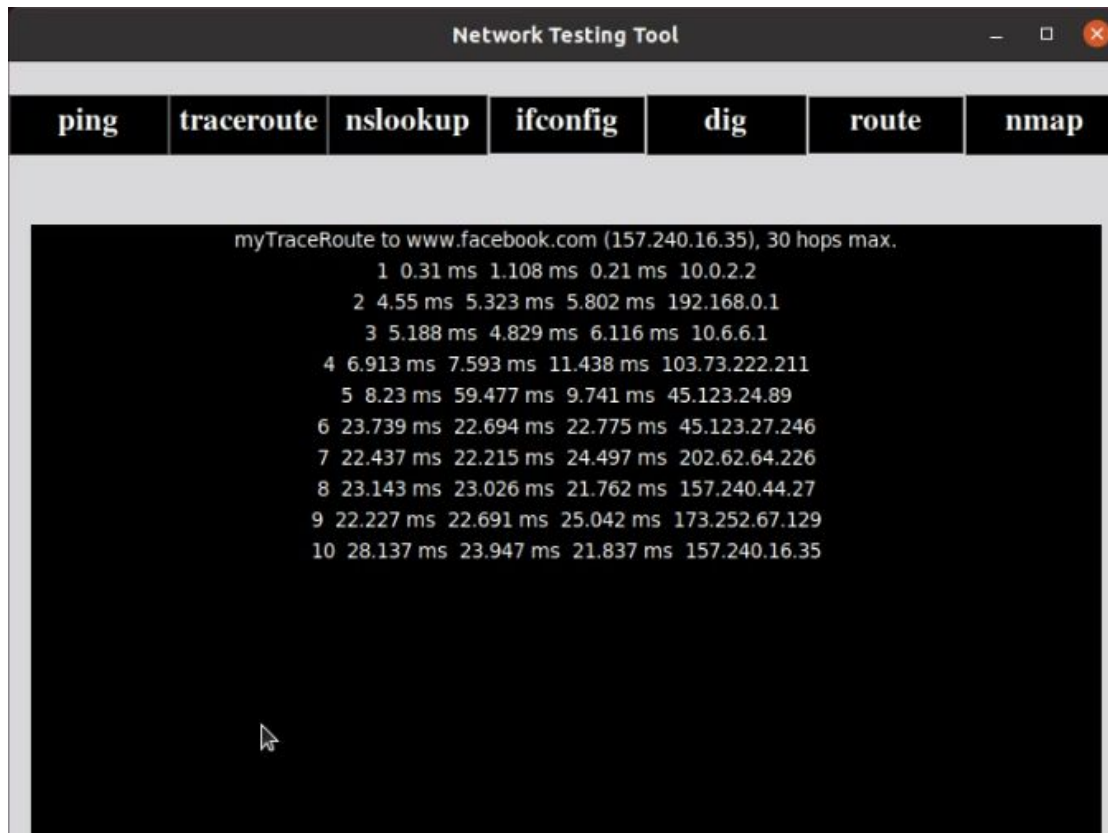


Network Testing Tool

ping | **traceroute** | nslookup | ifconfig | dig | route | nmap

```
myTraceRoute to 35.246.29.33 (35.246.29.33), 30 hops max.  
 1 0.33 ms 0.166 ms 0.164 ms 10.0.2.2  
 2 4.637 ms 4.314 ms 3.696 ms 192.168.0.1  
 3 13.612 ms 8.033 ms 4.747 ms 10.6.6.1  
 4 12.311 ms 8.589 ms 7.426 ms 103.73.222.213  
 5 8.718 ms 10.765 ms 13.685 ms 45.123.24.89  
 6 23.81 ms 22.299 ms 22.434 ms 45.123.27.246  
 7 24.066 ms 31.057 ms 29.727 ms 202.62.64.210  
 8 22.694 ms 23.362 ms 22.506 ms 108.170.248.218  
 9 * 87.743 ms 85.05 ms 216.239.54.183  
10 86.206 ms * 85.296 ms 72.14.238.177  
11 * 150.331 ms 150.155 ms 209.85.244.186  
12 244.591 ms 248.66 ms 244.314 ms 142.250.232.160  
13 264.653 ms 266.685 ms 267.658 ms 108.170.235.196  
14 277.812 ms 279.173 ms 280.632 ms 172.253.74.22  
15 278.681 ms 280.07 ms 278.232 ms 142.250.235.126  
16 286.701 ms 280.413 ms 340.463 ms 142.250.232.127  
17 305.67 ms 304.568 ms 303.483 ms 216.239.58.255  
18 362.955 ms 363.353 ms 362.36 ms 142.250.233.161  
19 379.203 ms 454.26 ms 379.787 ms 209.85.250.184  
20 * 371.989 ms 369.759 ms 74.125.246.161  
21 363.569 ms 366.339 ms 363.971 ms 108.170.232.164
```

www.facebook.com



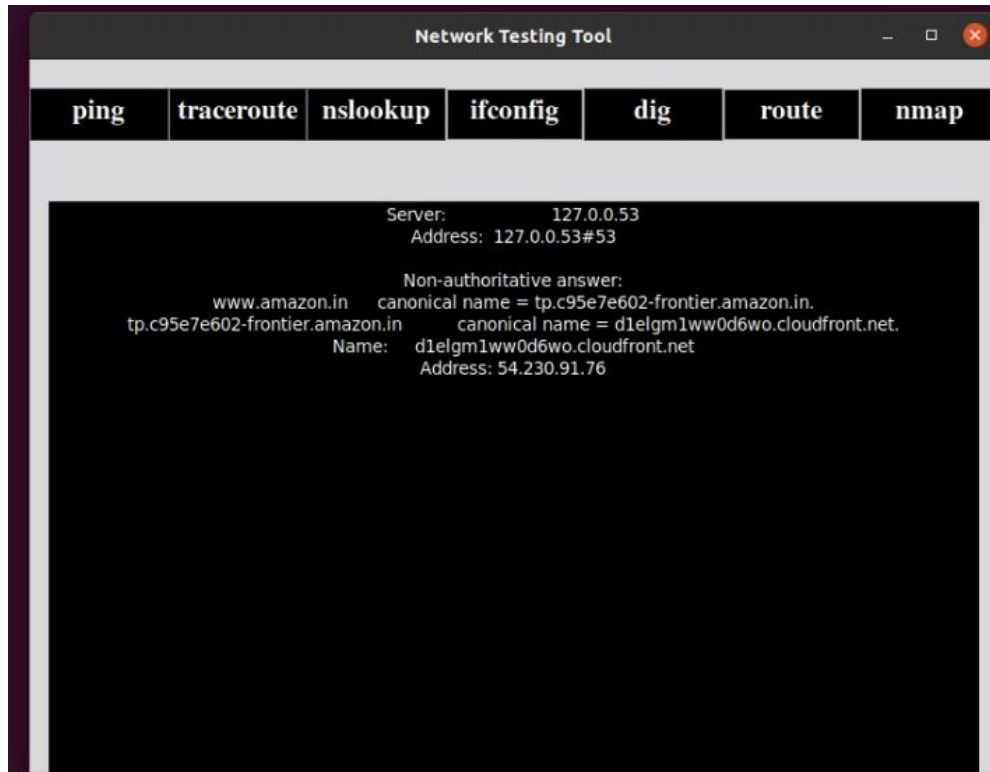
Network Testing Tool

ping | **traceroute** | nslookup | ifconfig | dig | route | nmap

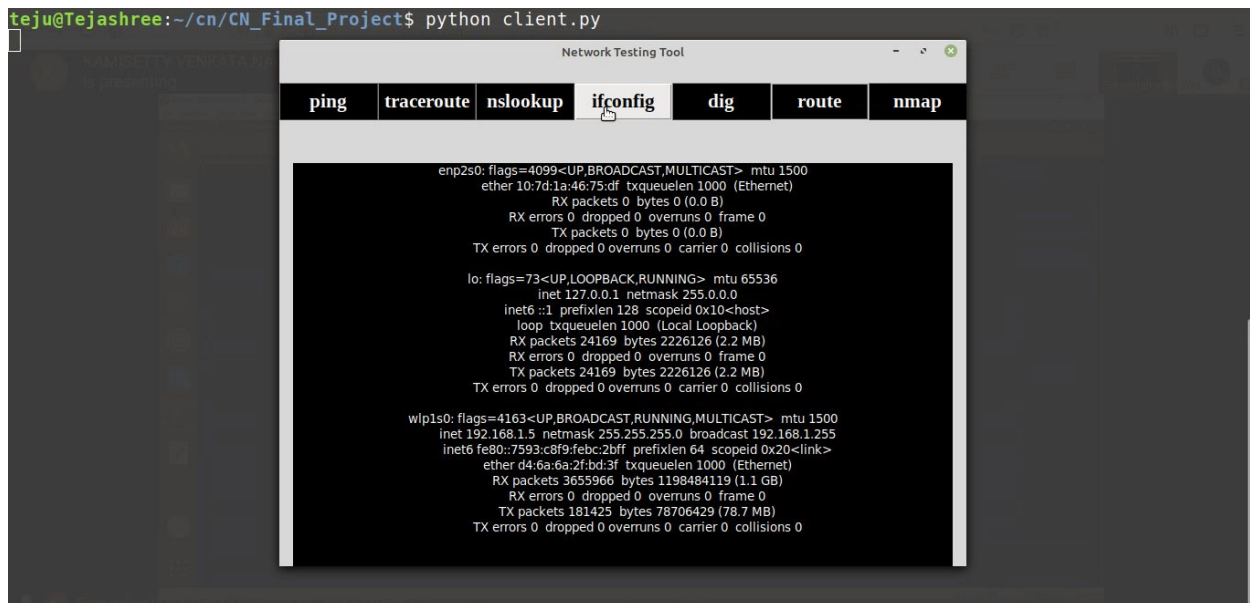
```
myTraceRoute to www.facebook.com (157.240.16.35), 30 hops max.  
 1 0.31 ms 1.108 ms 0.21 ms 10.0.2.2  
 2 4.55 ms 5.323 ms 5.802 ms 192.168.0.1  
 3 5.188 ms 4.829 ms 6.116 ms 10.6.6.1  
 4 6.913 ms 7.593 ms 11.438 ms 103.73.222.211  
 5 8.23 ms 59.477 ms 9.741 ms 45.123.24.89  
 6 23.739 ms 22.694 ms 22.775 ms 45.123.27.246  
 7 22.437 ms 22.215 ms 24.497 ms 202.62.64.226  
 8 23.143 ms 23.026 ms 21.762 ms 157.240.44.27  
 9 22.227 ms 22.691 ms 25.042 ms 173.252.67.129  
10 28.137 ms 23.947 ms 21.837 ms 157.240.16.35
```

nslookup:

www.amazon.in




Ifconfig:



dig:

server

```
teju@Tejashree:~/cn/CN_Final_Project$ python client.py
```



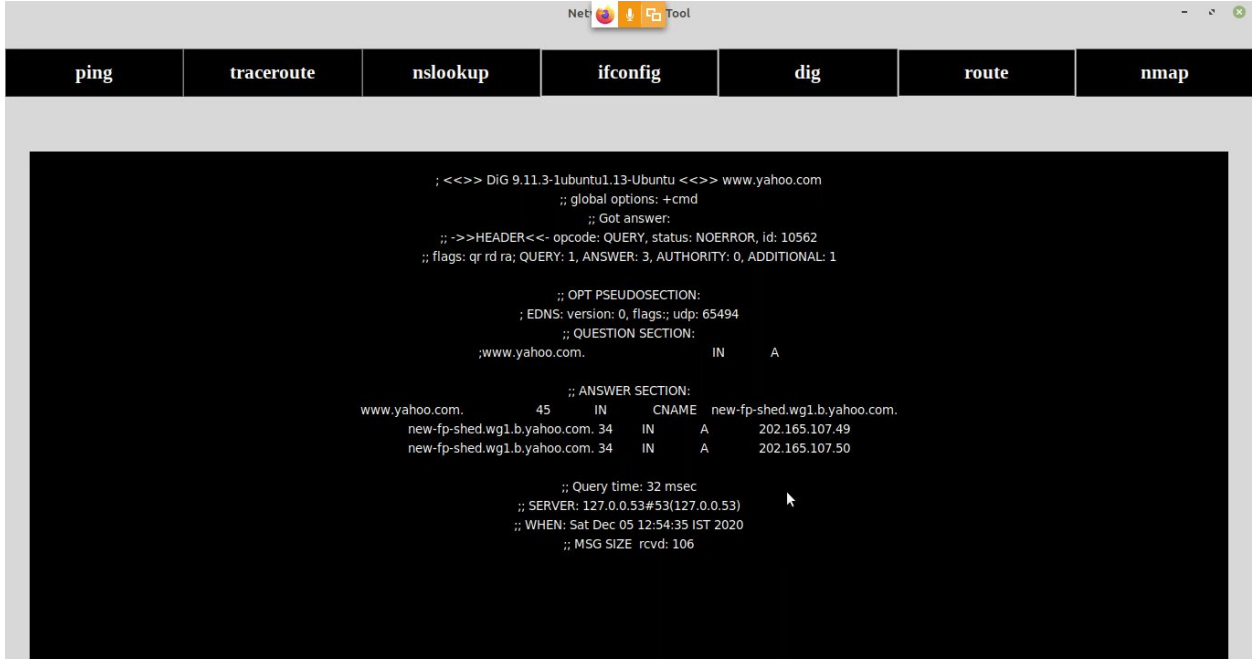
The screenshot shows a terminal window with the command `python client.py` executed. A secondary window titled "Network Testing Tool" is open, displaying the output of a `dig` command. The output shows a successful query for the IP address 35.246.29.33, returning an A record. The tool's interface includes a menu bar with options: ping, traceroute, nslookup, ifconfig, dig, route, and nmap.

```
; <<>> DiG 9.11.3-1ubuntu1.13-Ubuntu <<>> 35.246.29.33
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 48027
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: udp: 65494
;; QUESTION SECTION:
;35.246.29.33.                IN      A

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sat Dec 05 12:54:21 IST 2020
;; MSG SIZE rcvd: 41
```

www.yahoo.com



The screenshot shows the "Network Testing Tool" interface with the `dig` command selected. The output displays a successful query for the domain `www.yahoo.com`, returning multiple A records. The tool's interface includes a menu bar with options: ping, traceroute, nslookup, ifconfig, dig, route, and nmap.

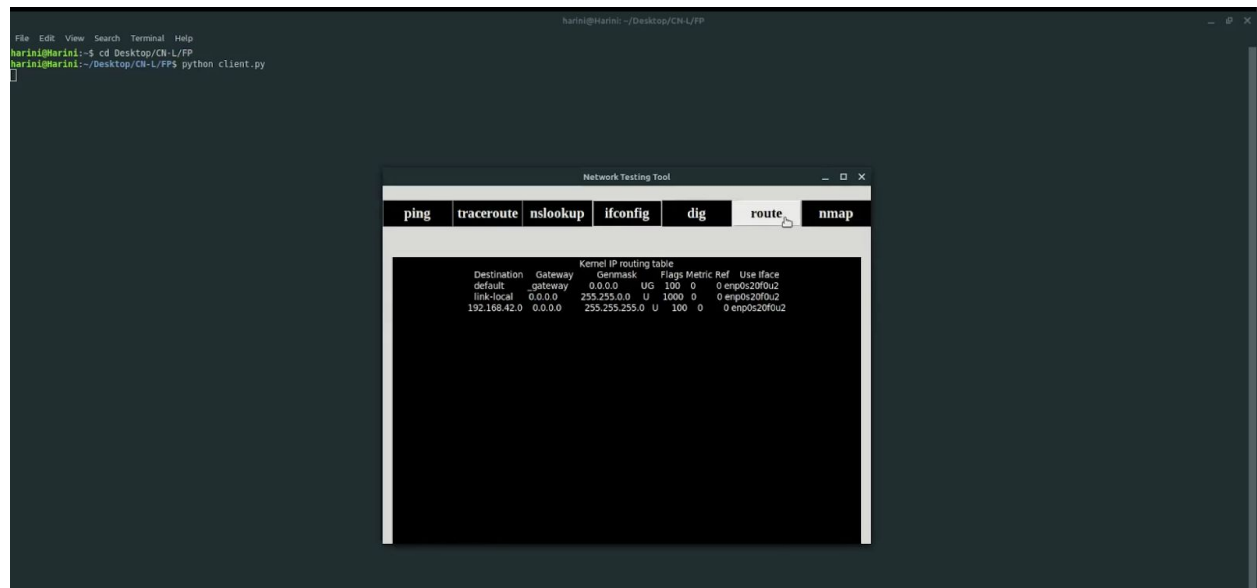
```
; <<>> DiG 9.11.3-1ubuntu1.13-Ubuntu <<>> www.yahoo.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10562
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: udp: 65494
;; QUESTION SECTION:
;www.yahoo.com.              IN      A

;; ANSWER SECTION:
www.yahoo.com.               45      IN      CNAME  new-fp-shed.wg1.b.yahoo.com.
new-fp-shed.wg1.b.yahoo.com. 34      IN      A      202.165.107.49
new-fp-shed.wg1.b.yahoo.com. 34      IN      A      202.165.107.50

;; Query time: 32 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Sat Dec 05 12:54:35 IST 2020
;; MSG SIZE rcvd: 106
```

route:



Nmap: server www.google.com

