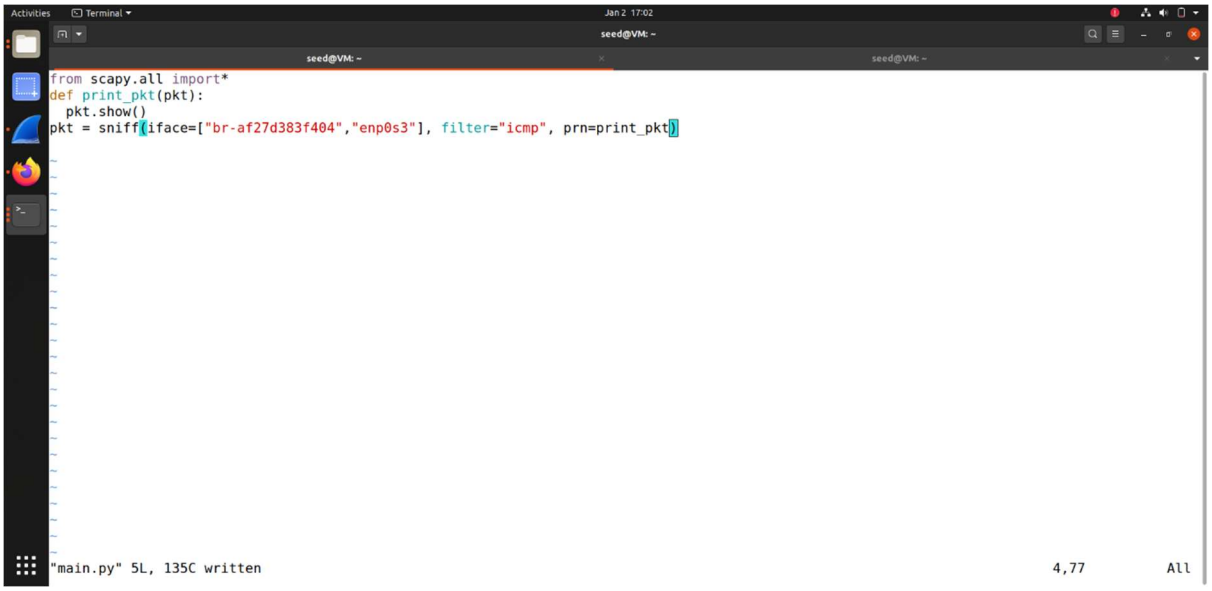
First Part:

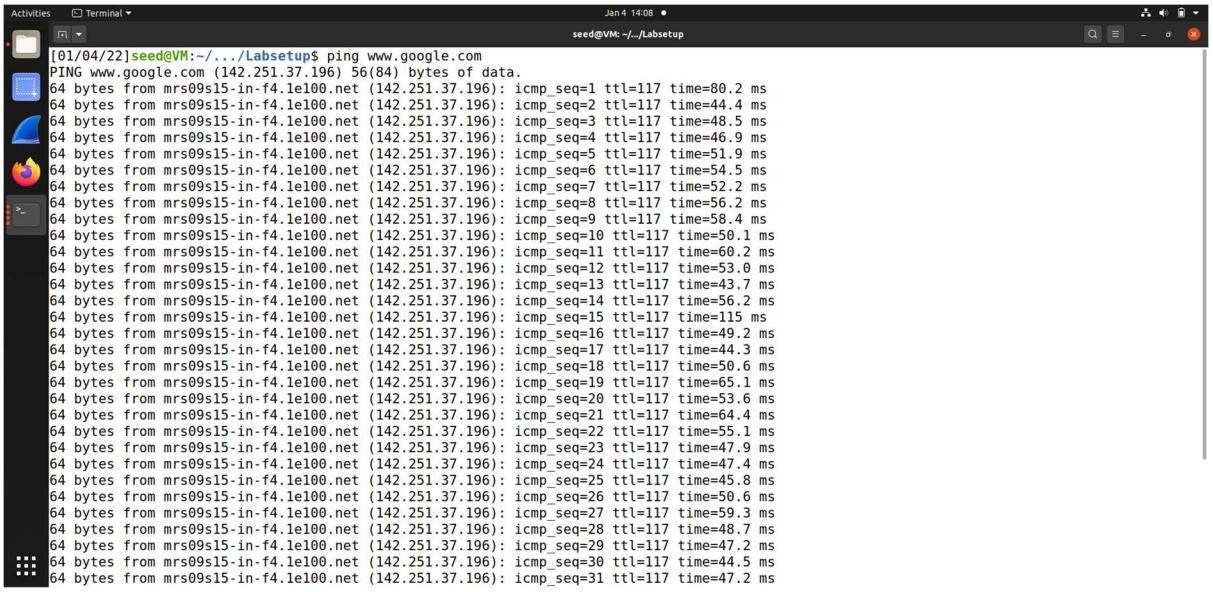
All the codes in the first part are written in python



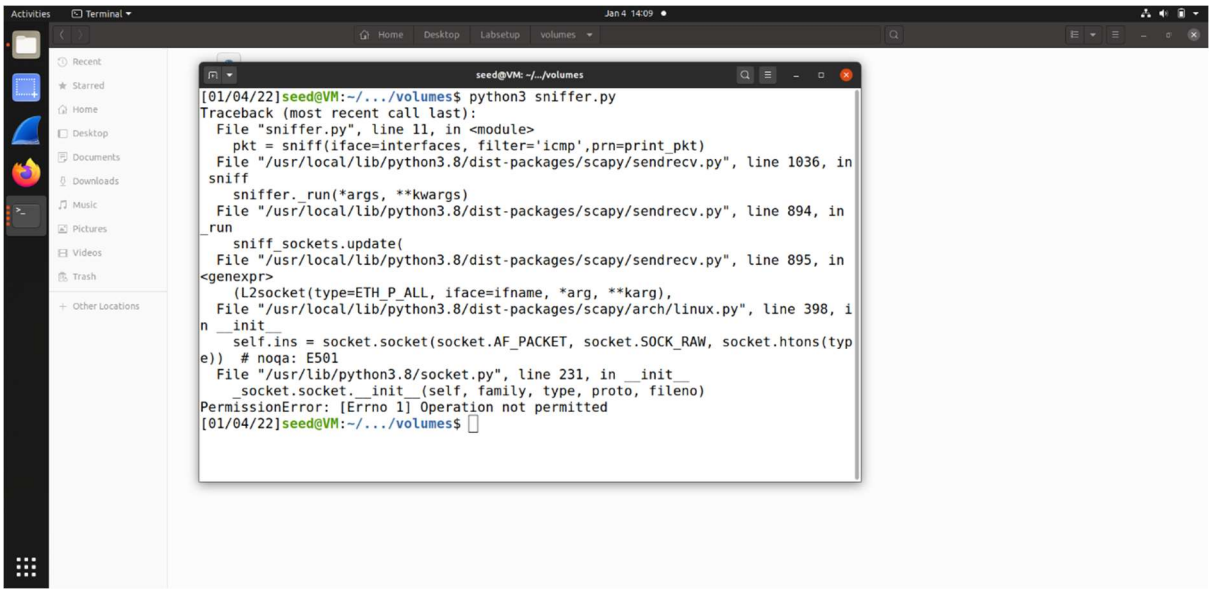
The code above will sniff the packets on the br-af27d383f404 interface

Task 1.1 A:

Capturing packets using root privileges:

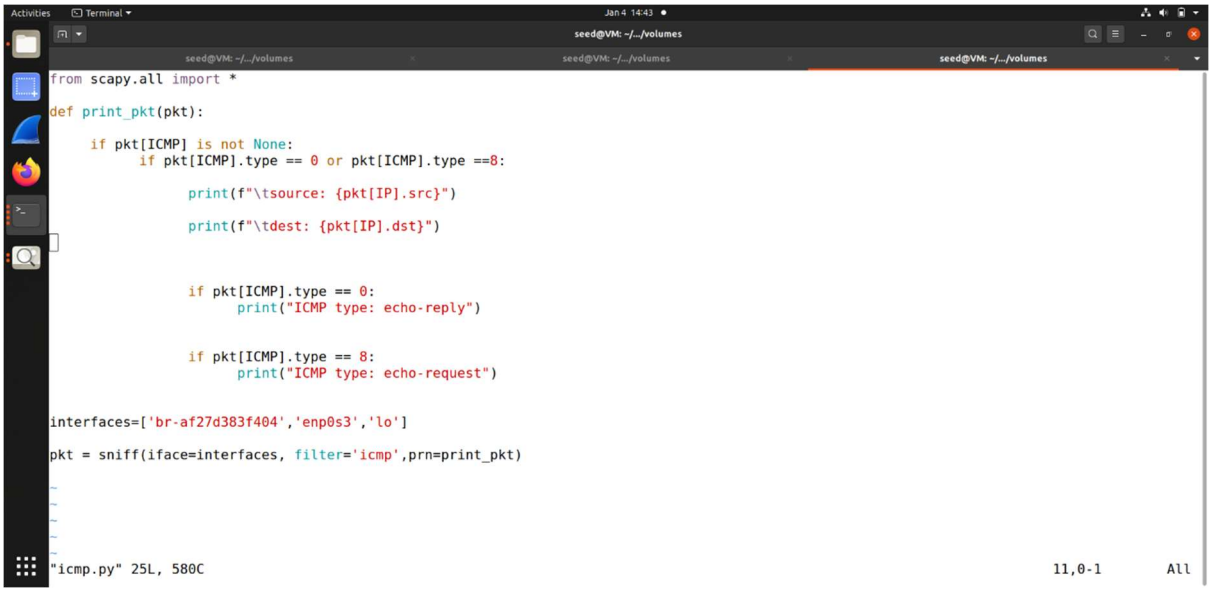


Running the code with out root privilege :



Task 1.1 B:

Code is written to capture ICMP packets:

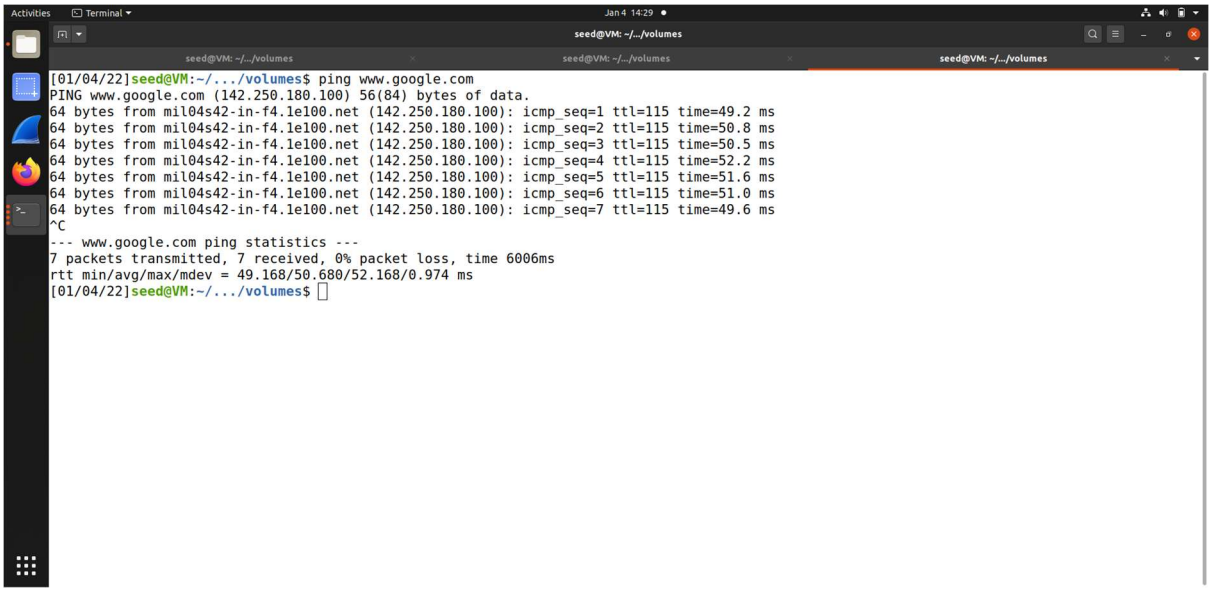


```
from scapy.all import *

def print_pkt(pkt):

    if pkt[ICMP] is not None:
        if pkt[ICMP].type == 0 or pkt[ICMP].type ==8:

            print(f"\tsource: {pkt[IP].src}")

            print(f"\tdest: {pkt[IP].dst}")


            if pkt[ICMP].type == 0:
                print("ICMP type: echo-reply")


            if pkt[ICMP].type == 8:
                print("ICMP type: echo-request")


interfaces=['br-af27d383f404','enp0s3','lo']

pkt = sniff(iface=interfaces, filter='icmp',prn=print_pkt)

~
~
~
~
"icmp.py" 25L, 580C                                                    11,0-1        All
```

Running the code to capture ICMP packets ,on sending ping to google.com:

```
[01/04/22]seed@VM:~/.../volumes$ vi icmp.py
[01/04/22]seed@VM:~/.../volumes$ sudo chmod a+x icmp.py
[01/04/22]seed@VM:~/.../volumes$ sudo python3 icmp.py
        source: 10.0.2.15
        dest: 216.58.208.164
ICMP type: echo-request
        source: 216.58.208.164
        dest: 10.0.2.15
ICMP type: echo-reply
        source: 10.0.2.15
        dest: 216.58.208.164
ICMP type: echo-request
        source: 216.58.208.164
        dest: 10.0.2.15
ICMP type: echo-reply
        source: 10.0.2.15
        dest: 216.58.208.164
ICMP type: echo-request
        source: 216.58.208.164
        dest: 10.0.2.15
ICMP type: echo-reply
        source: 10.0.2.15
        dest: 216.58.208.164
ICMP type: echo-request
        source: 216.58.208.164
        dest: 10.0.2.15
ICMP type: echo-reply
        source: 10.0.2.15
        dest: 216.58.208.164
ICMP type: echo-request
        source: 216.58.208.164
```

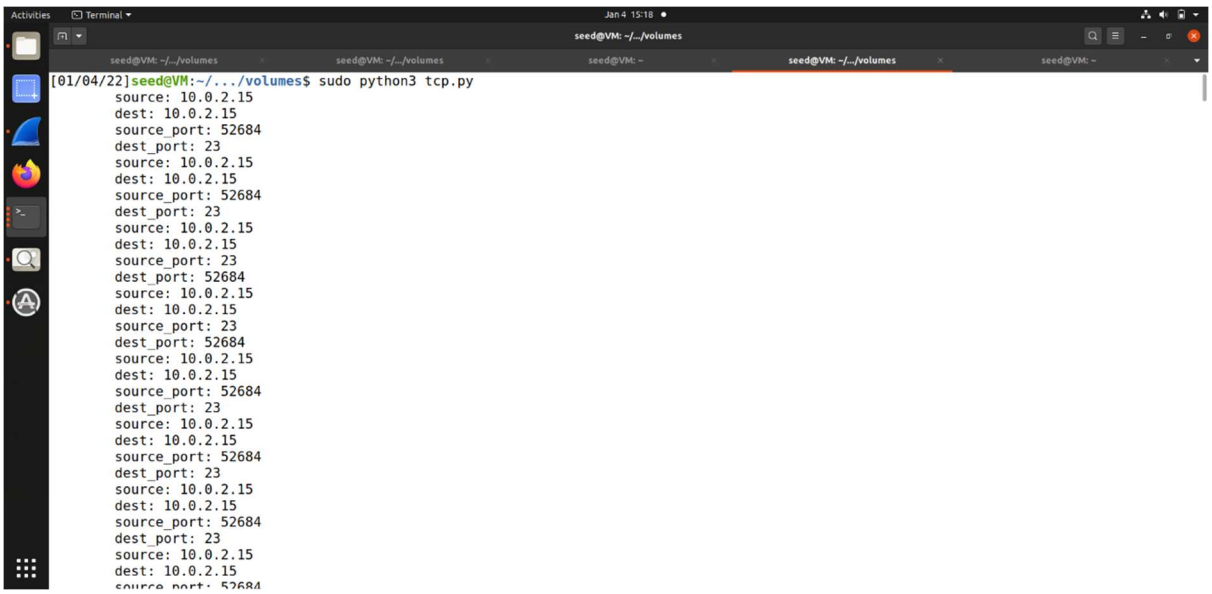
Ping to www.google.com:



Task 1.1 B 2):

Capture Tcp packets:

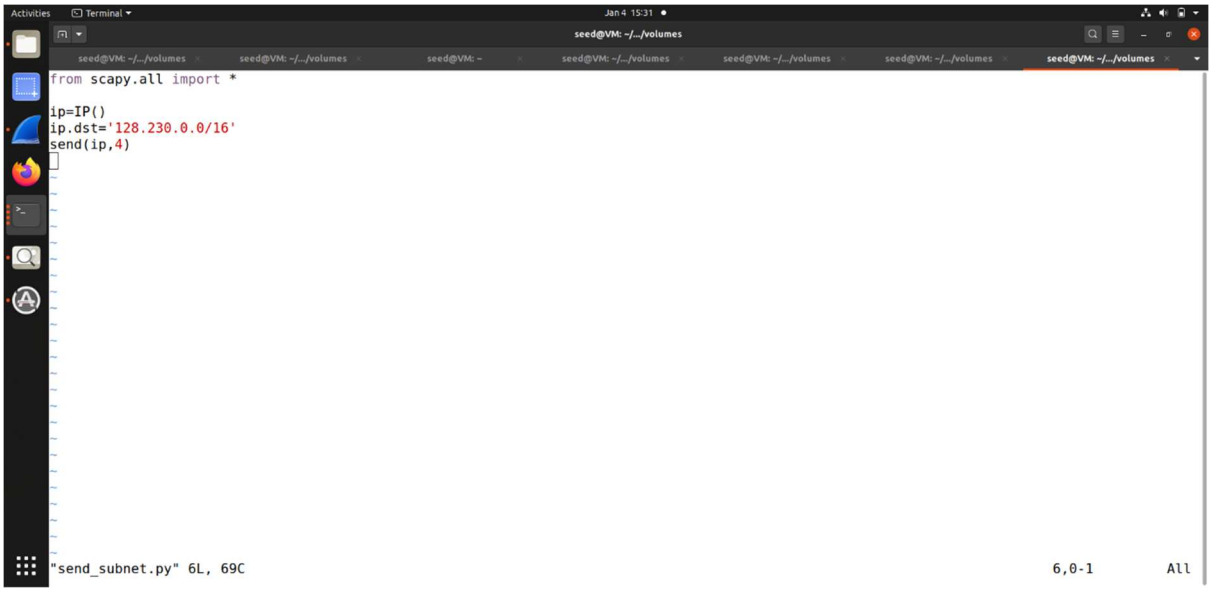
Tcp packets captured on port 23 and host 10.0.2.15



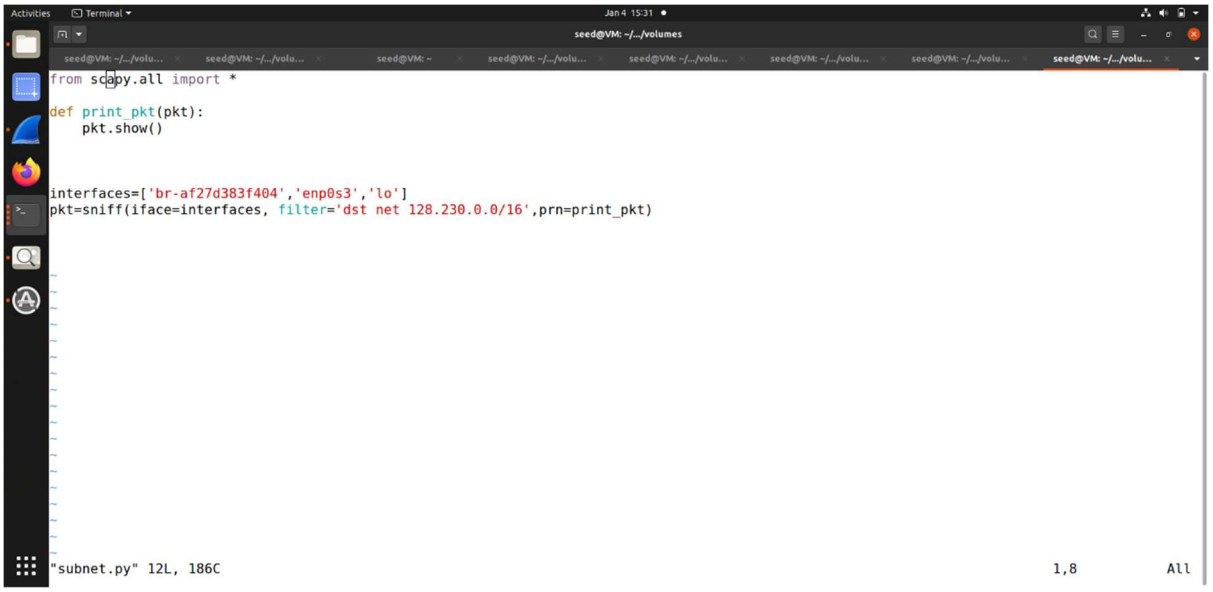
Code used to capture tcp packets:

Task 1.1 B 3):

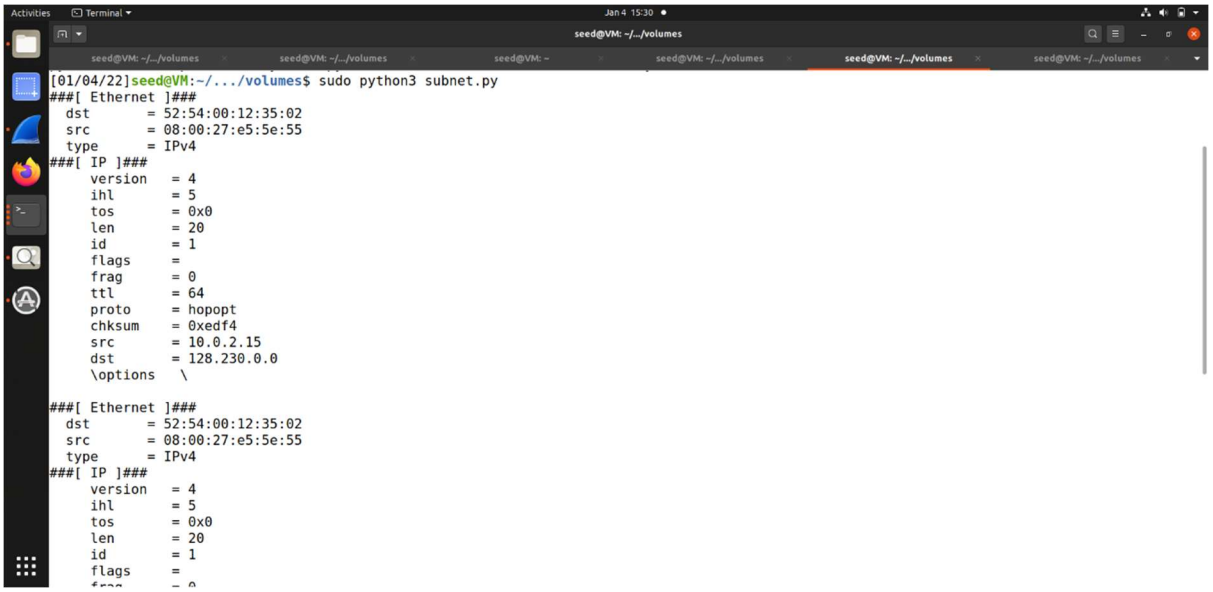
Code used to send packets to specific subnet:



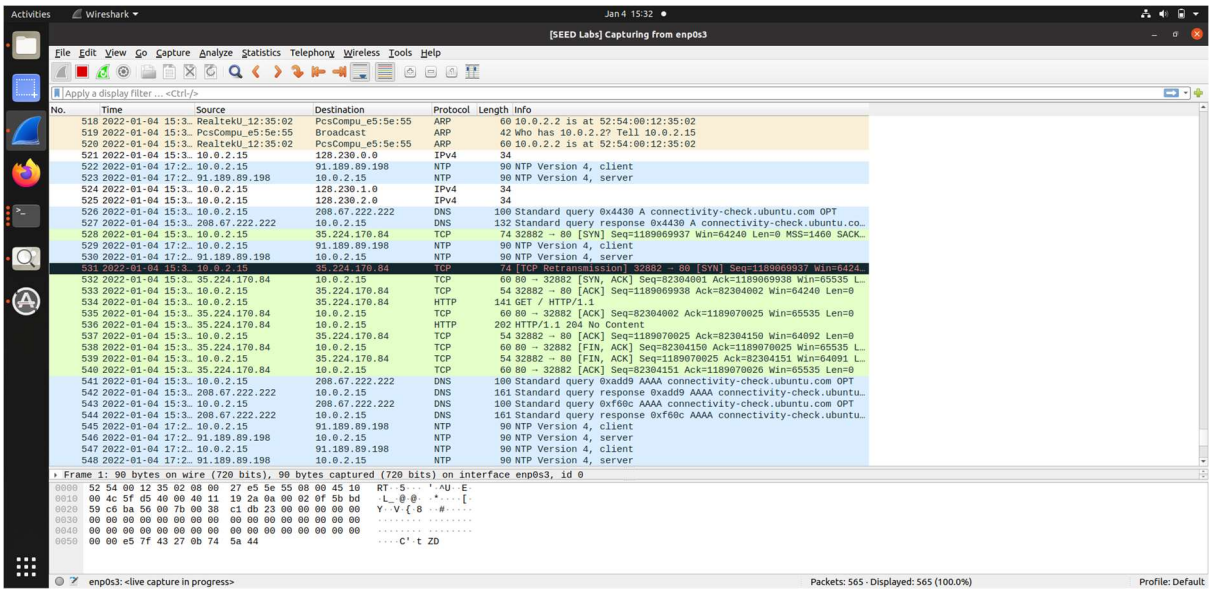
Code used to sniff packets on dest 128.230.0.0/16:



Running the program to sniff packets:



Wireshark photo to show the Tcp packets being sniffed on the 128.230.0.0/16:

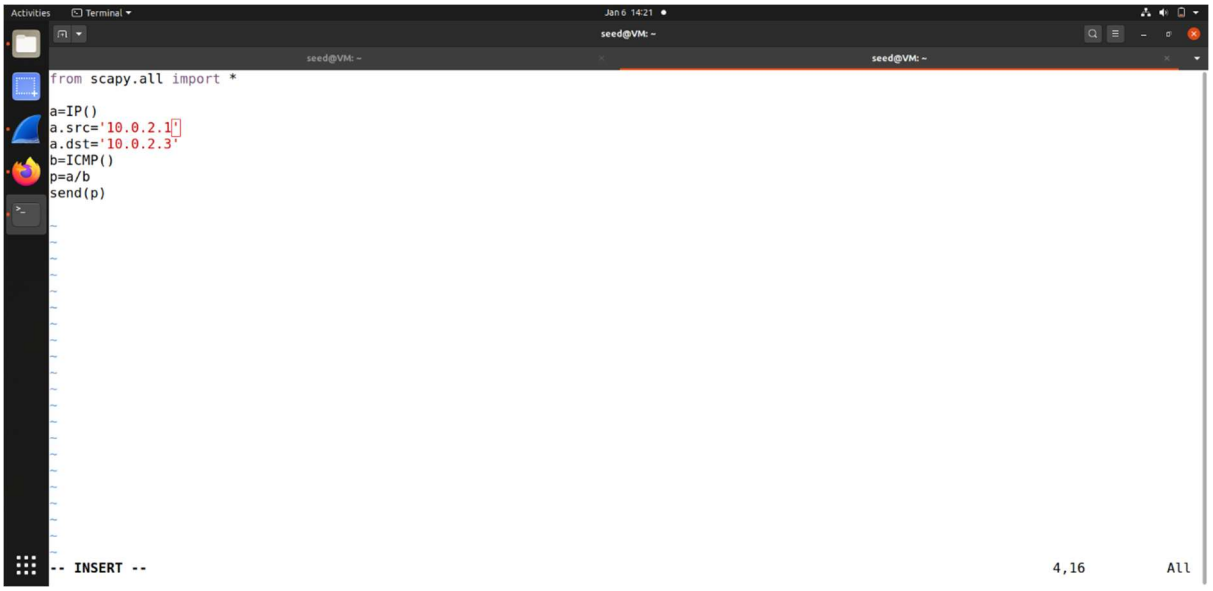


Task 1.2:

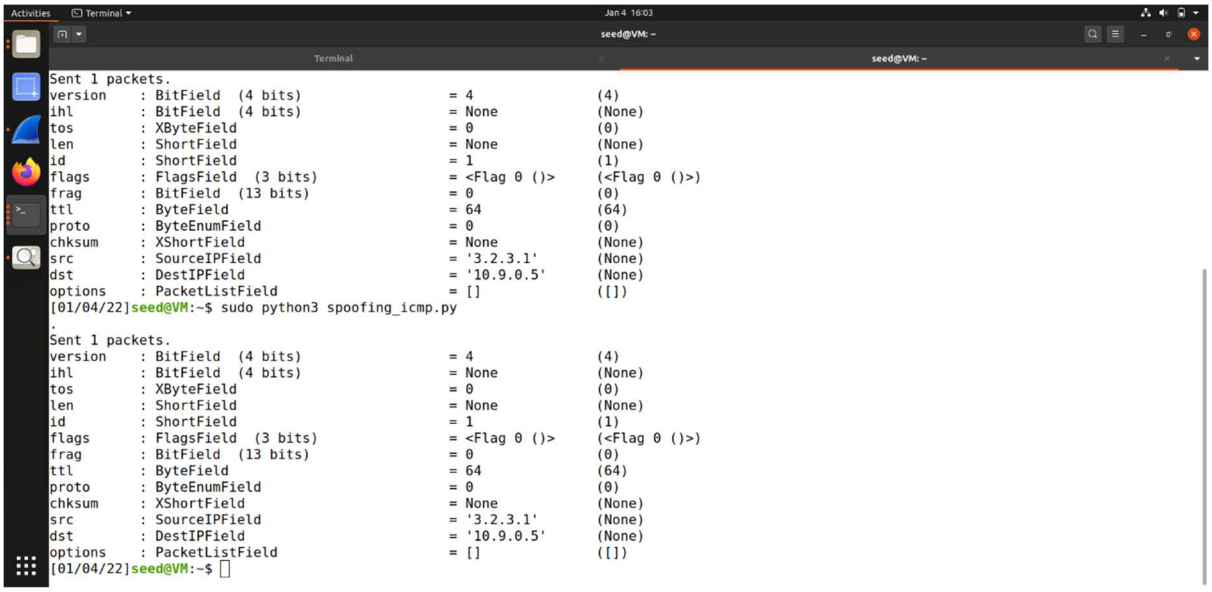
We will spoof ICMP echo request packets, and send them to another VM on the same network. We will use Wireshark to observe whether our request will be accepted by the receiver.

a.src is an arbitrary ip used for sending packet,

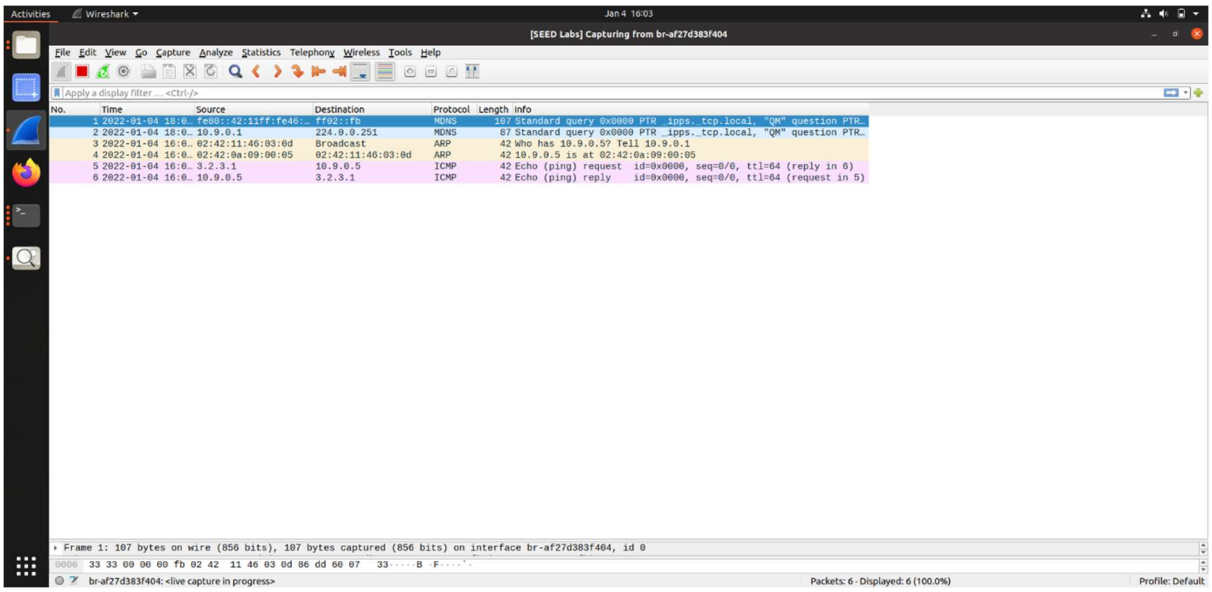
this code used to send packet from src to dest:



Spoofing the sent packet:



As it appears on wireshark image below there is an echo request and an echo reply to spoofed packed that was sent:



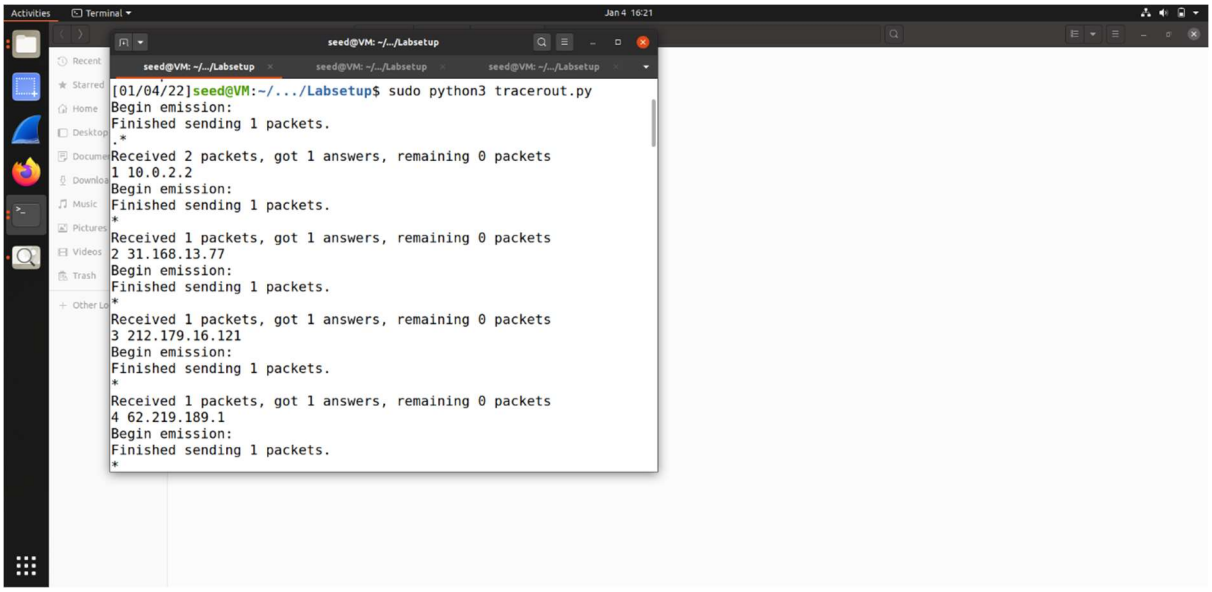
Task1.3:

In this task we estimate the distance of a data being sent from src to dest,

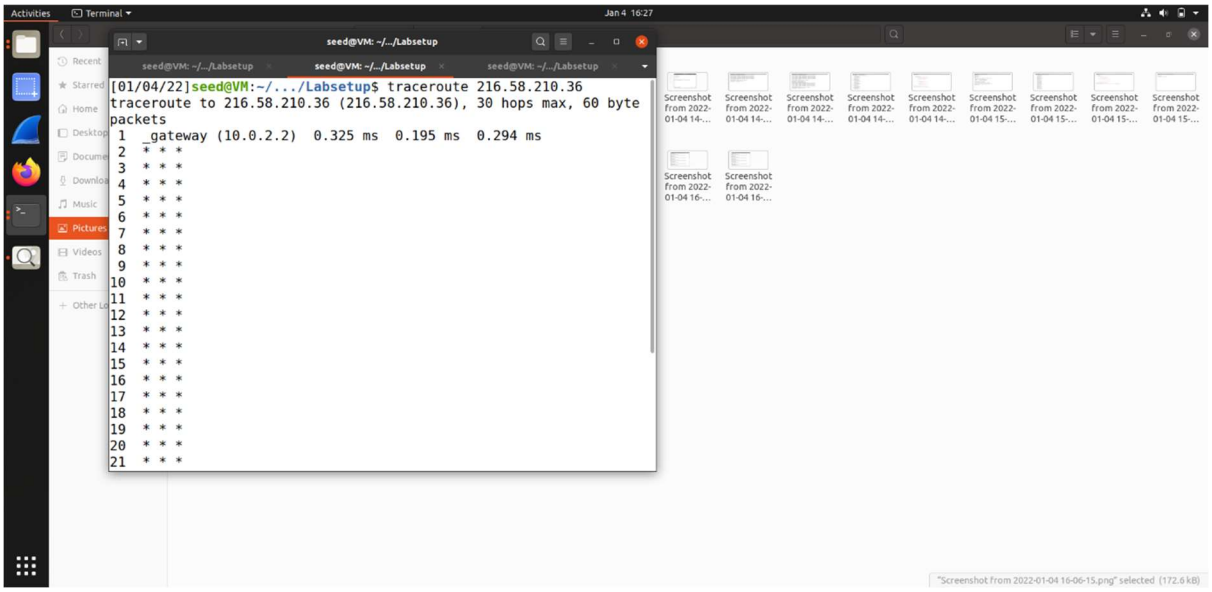
By using TCP/IP protocols

Tracerout program used to send packets over network from src to dest,

The path of the packet:



Estimated distance time at takes to arrive at dest on the network:

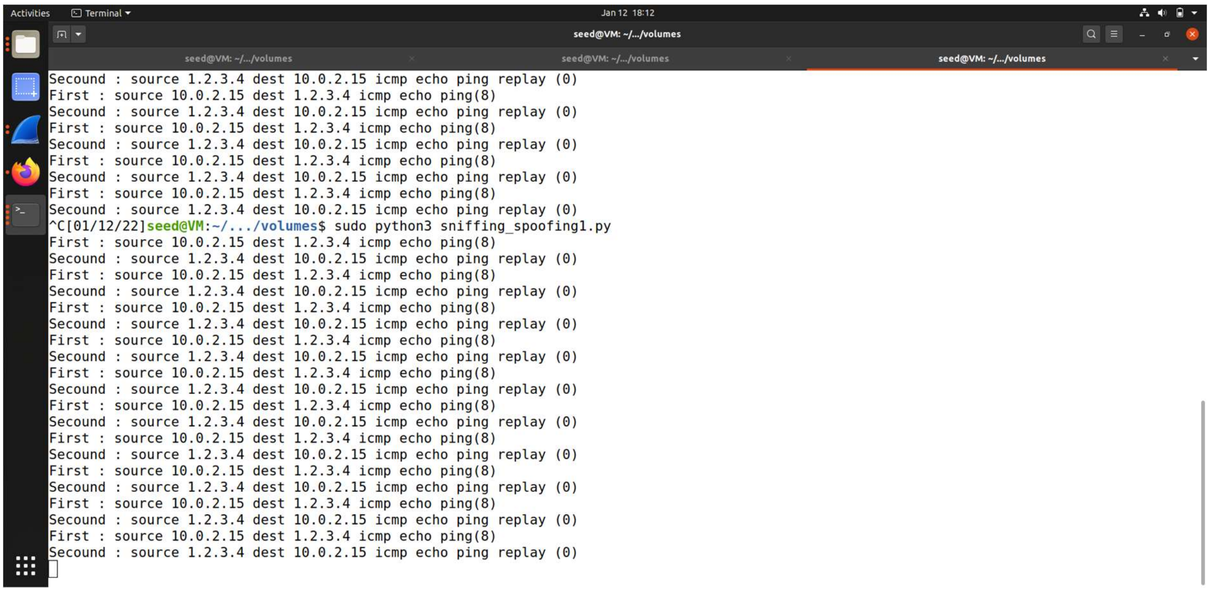


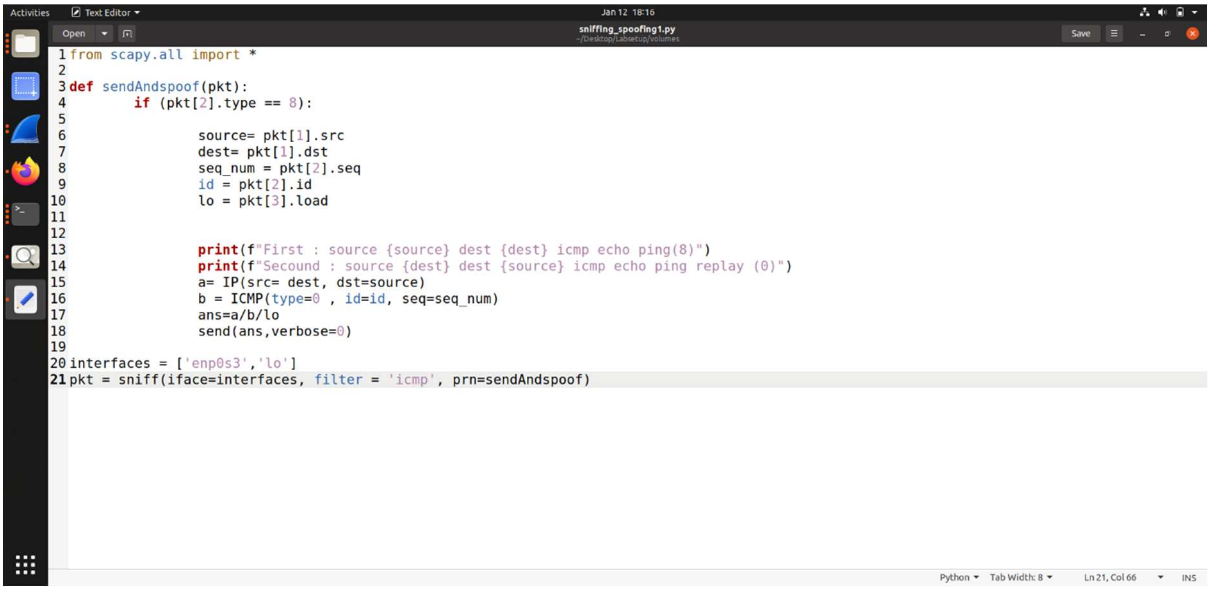
Task1.4:

In this task we have implemented sniff and then spoof program,

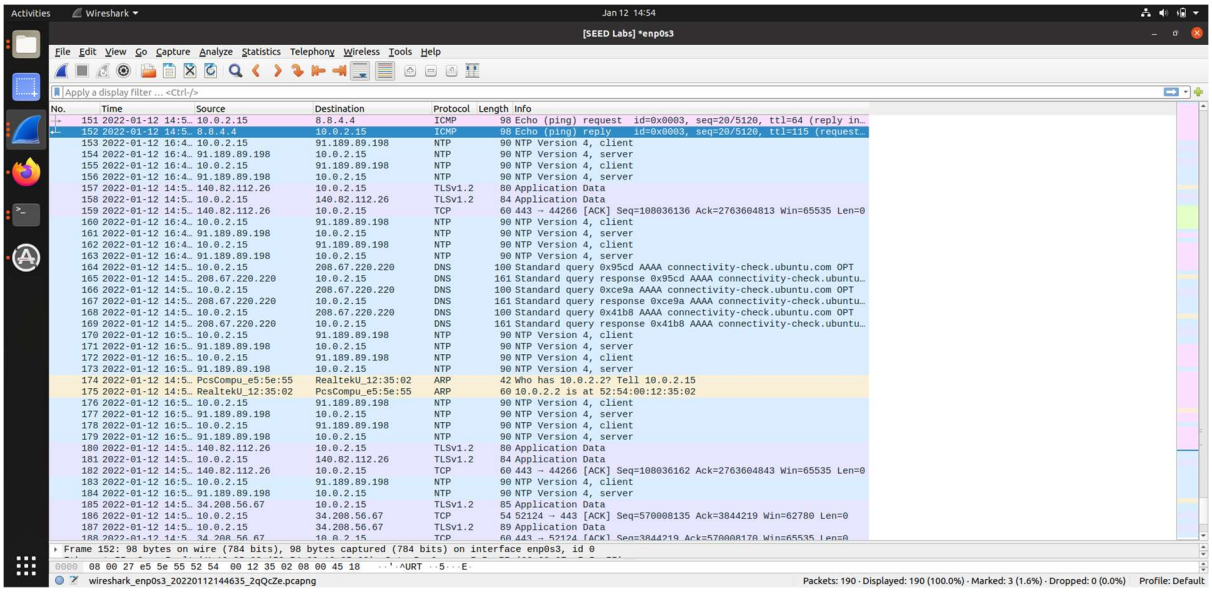
We have sent ping to 1.2.3.4, we have used the attacker container to response to the ARP protocol asking for a specific ip belongs to who.

<u>Runnning the program :</u>





as you can see on the wireshark image below the ARP asks who has this ip and the attacker gives a response to hime that he acknowledges this ip.

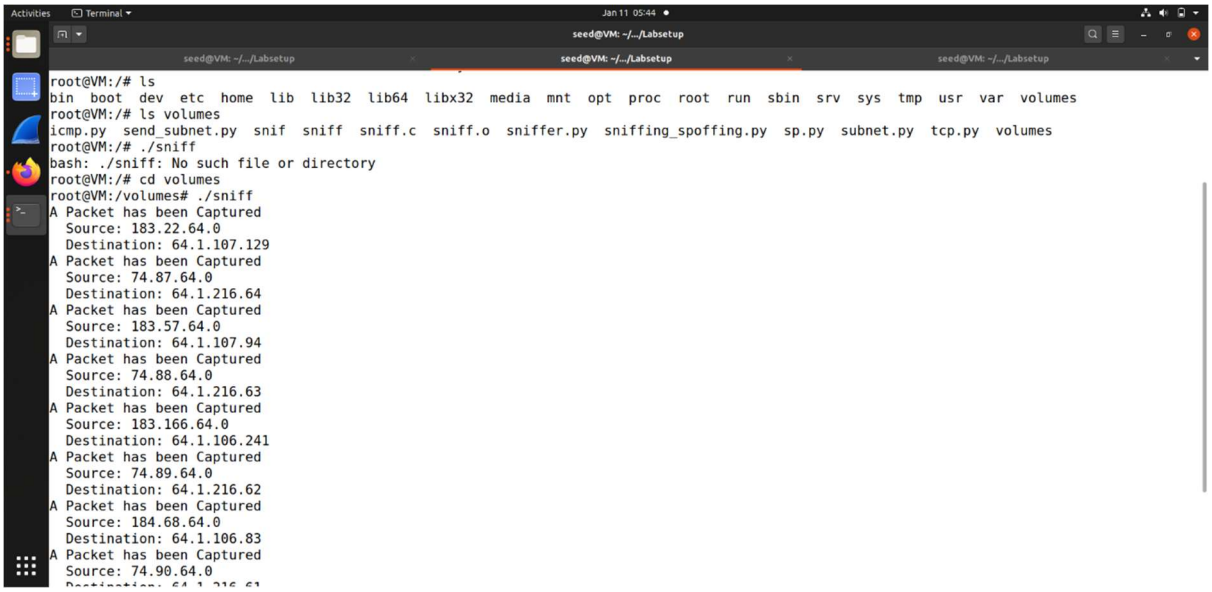


Second part:

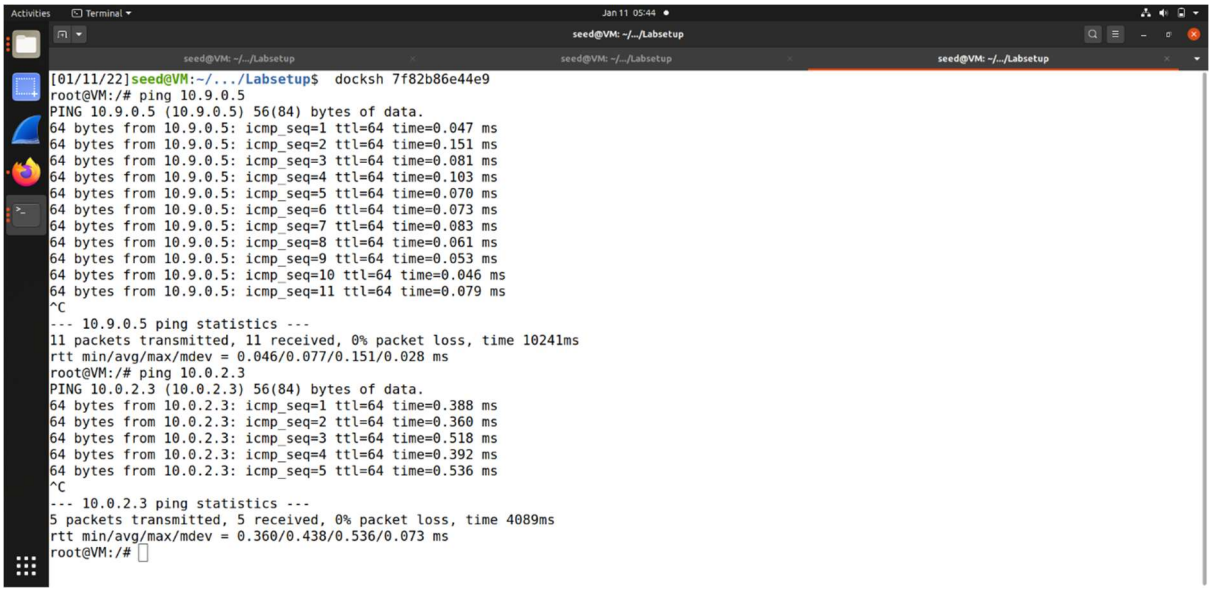
Task2:

packets captured by the sniff program:

running with pernicious mode=0

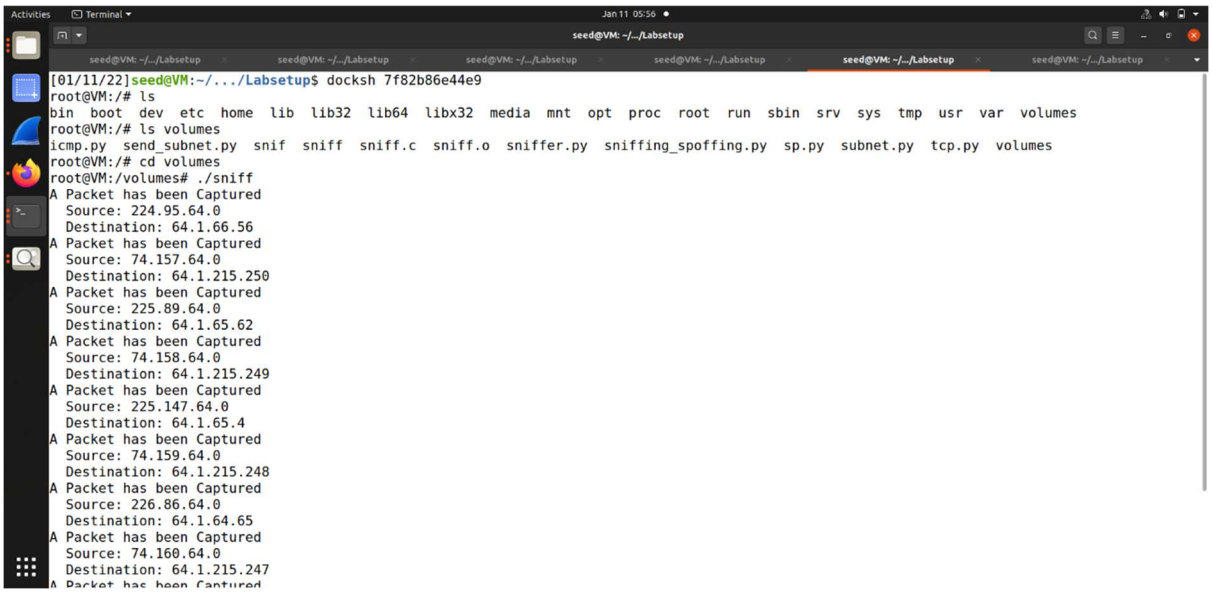


The program in the phot up is the captures being sent by ping 10.9.0.5:

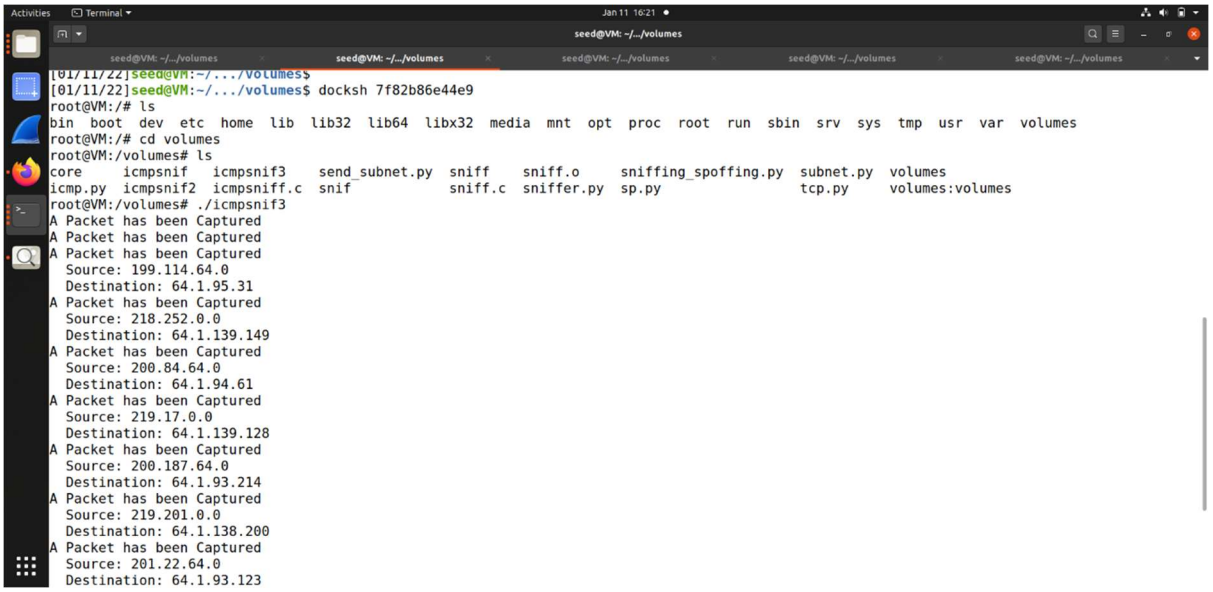


packets captured by the sniff program:

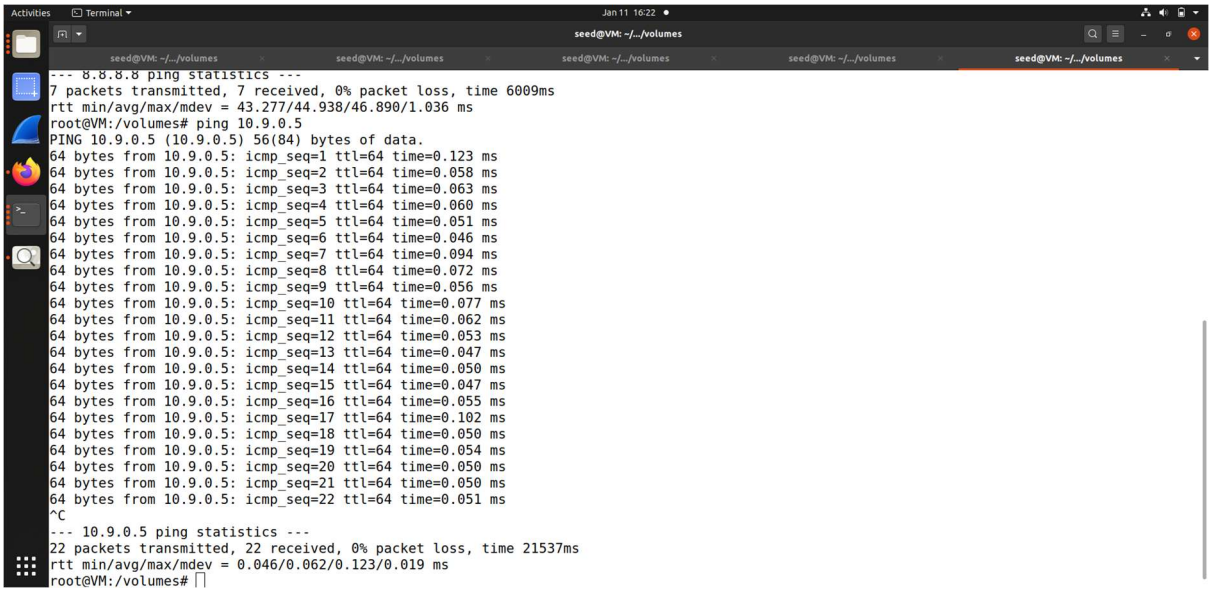
running with pernicious mode=1



Program that captures Icmp packets between two hosts on the network:



Sending packets using ping 10.9.0.5:



We have used the pcap library which contains the main functions for the sniffing program:

1.pcap_open_live: to capture packets going on the network(network card)

2.pcap_compile: used for the packets filter

3.pcap_setfilter: used for packets filter

4.pcap_loop: for receiving packets

Question 2:

On order to capture packets on the network level(network card) we need to use raw socket,

To use Raw socket u need root privalige, using raw socket could be very sensitive in the term of the user.

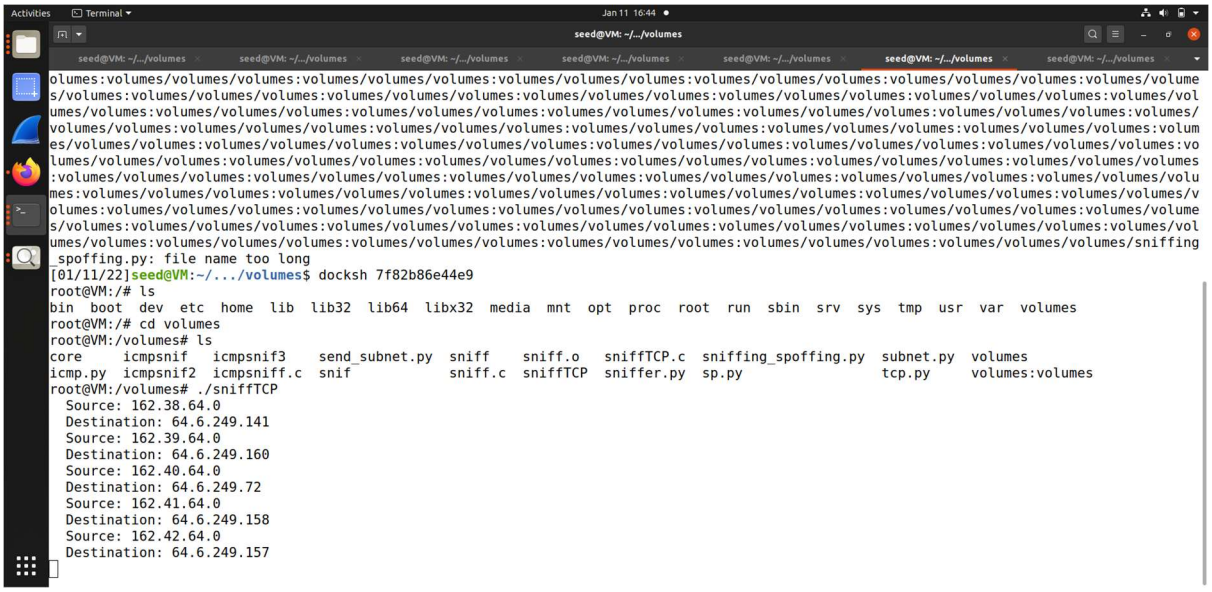
Question 3:

When the premicious mode is off we can sniff only the packets that goes on our network card and was meant to be sent to us(to our network card),

But if the premicious mode is on more packets can be captured by our network card which may not necceseeraly meant to be sent to our network card.

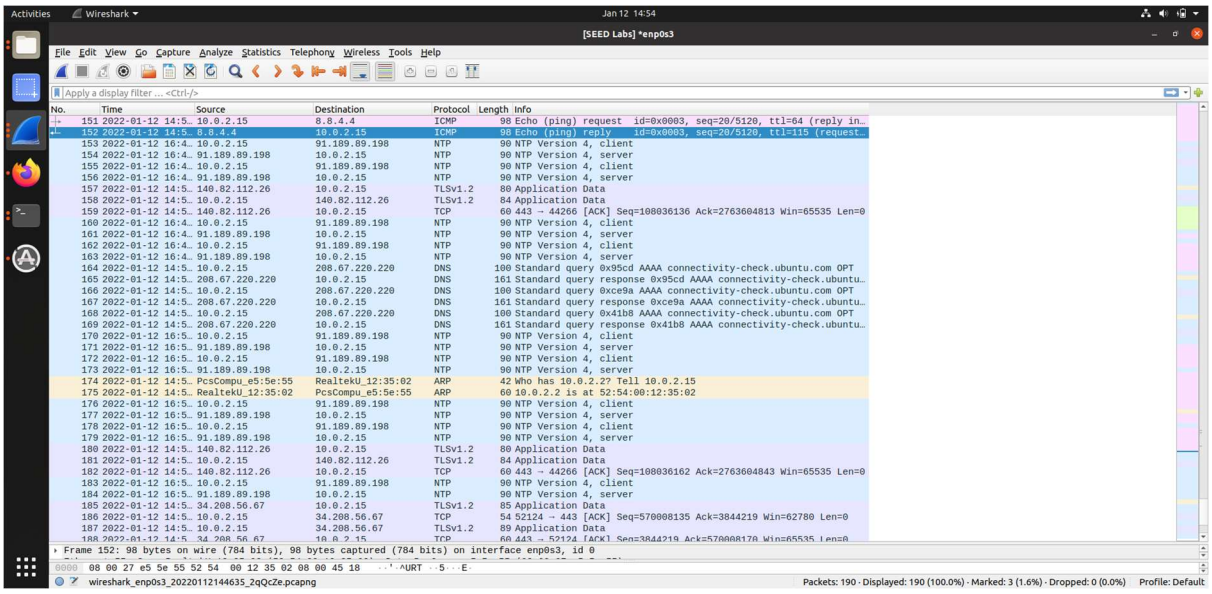
<u>Program that captures TCP with a given range which is 1-100:</u>

TCP program to capture tcp packets with a given port range:



Task2.2A:

Wireshark photo shows the packets that was spoofed,ping sent to 8.8.4.4



Task2.2B:

As seen on the photo below the packet has been spoofed succefully:

Question 4:

No, applying this on our code may cause an error on the code , so it will not work

Question 5:

Calculating checksum for the ip header is not needed ,

It's calculated by our operating system

Question 6:

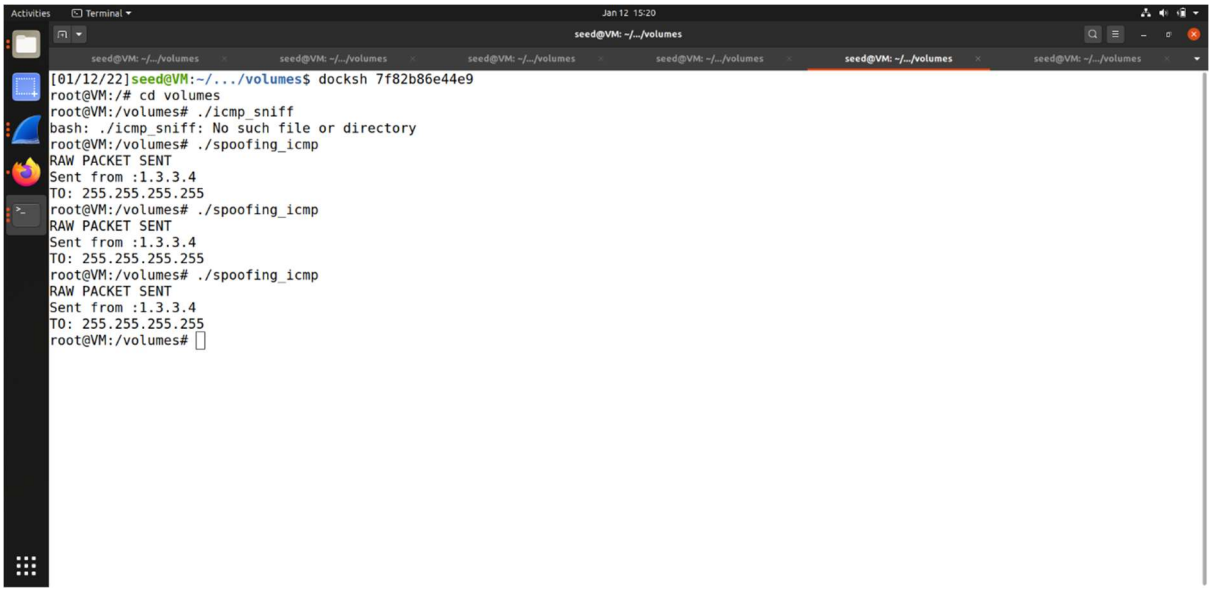
Raw packets allows us to deal with packets and protocols on low level so it requires a root privilege,

It's up to the user to allow using raw sockets , because it could harm the system and could be dangerous

<u>Task2.3-sniff and then spoof:</u>

Ping sent to 1.3.3.4:

Running spoofing_icmp program to spoof ICMP request:



We have tried our best to make it work but was just getting icmp request without replies as seen on the wireshark photo below: