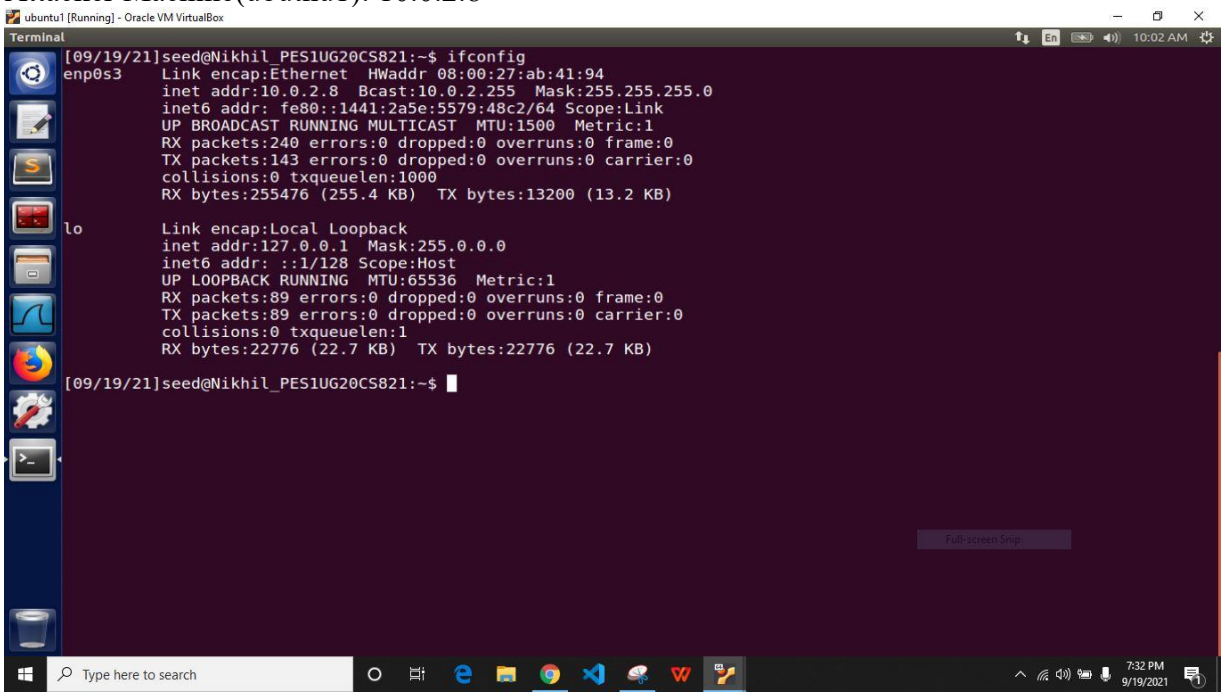
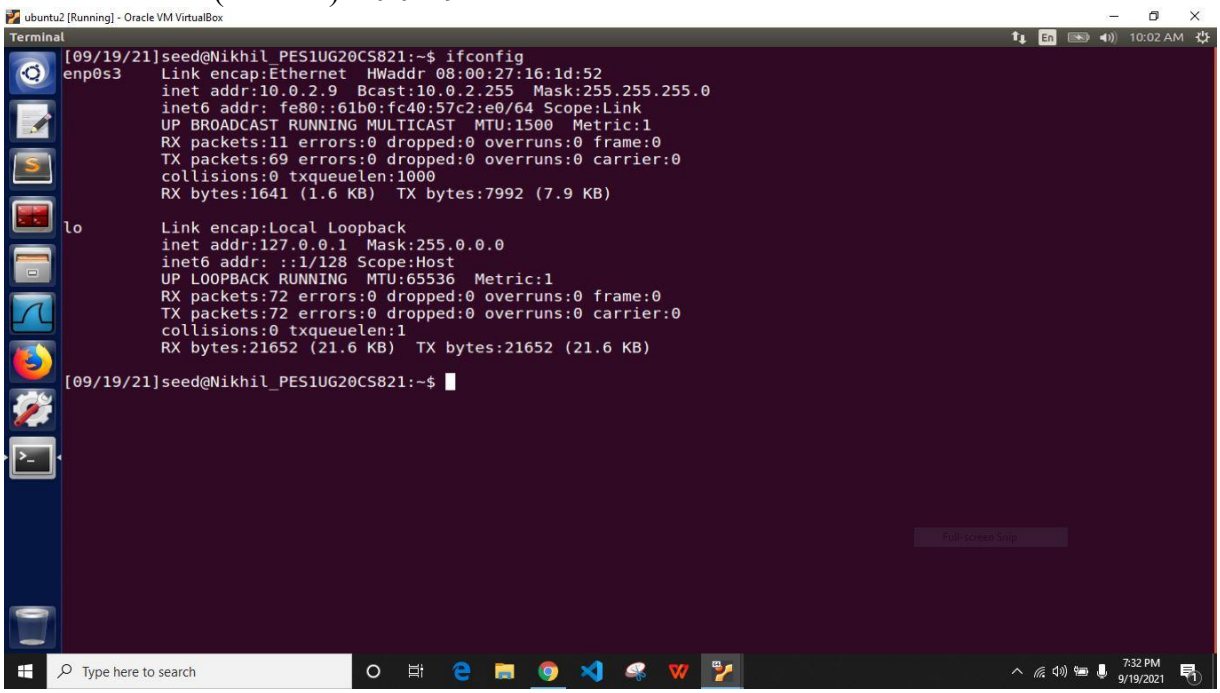
Week 1 Packet Sniffing and Spoofing

Name: Nikhil T M
SRN: PES1UG20CS821
Subject:Computer Network Security

Attacker Machine(ubutnu1): 10.0.2.8



Victim Machine(ubuntu2): 10.0.2.9

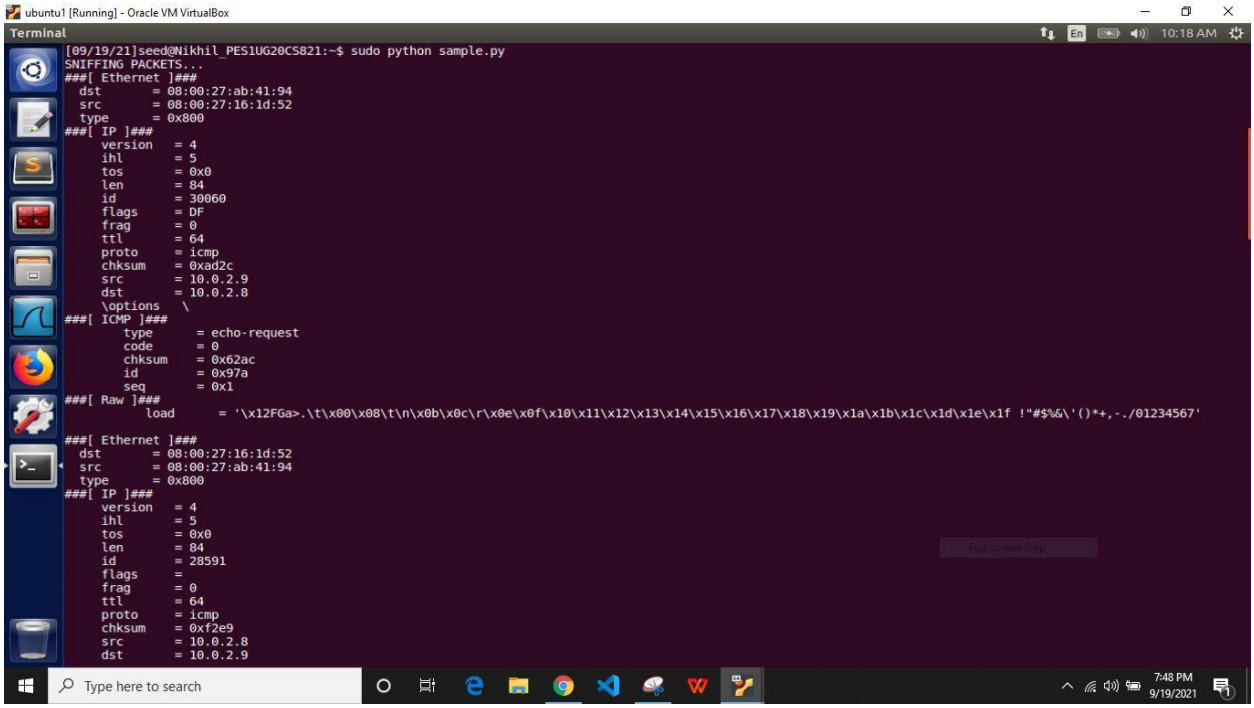


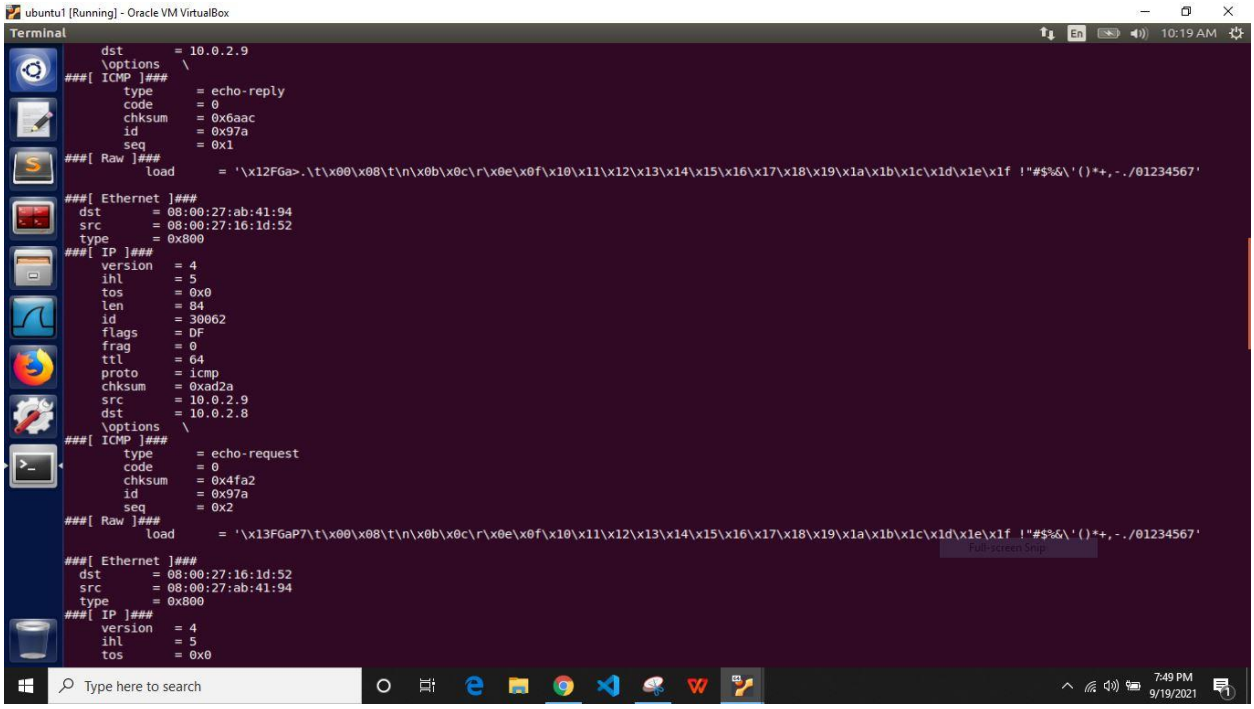
Task 1: Sniffing Packet

Task 1.1 Sniff IP packets using Scapy

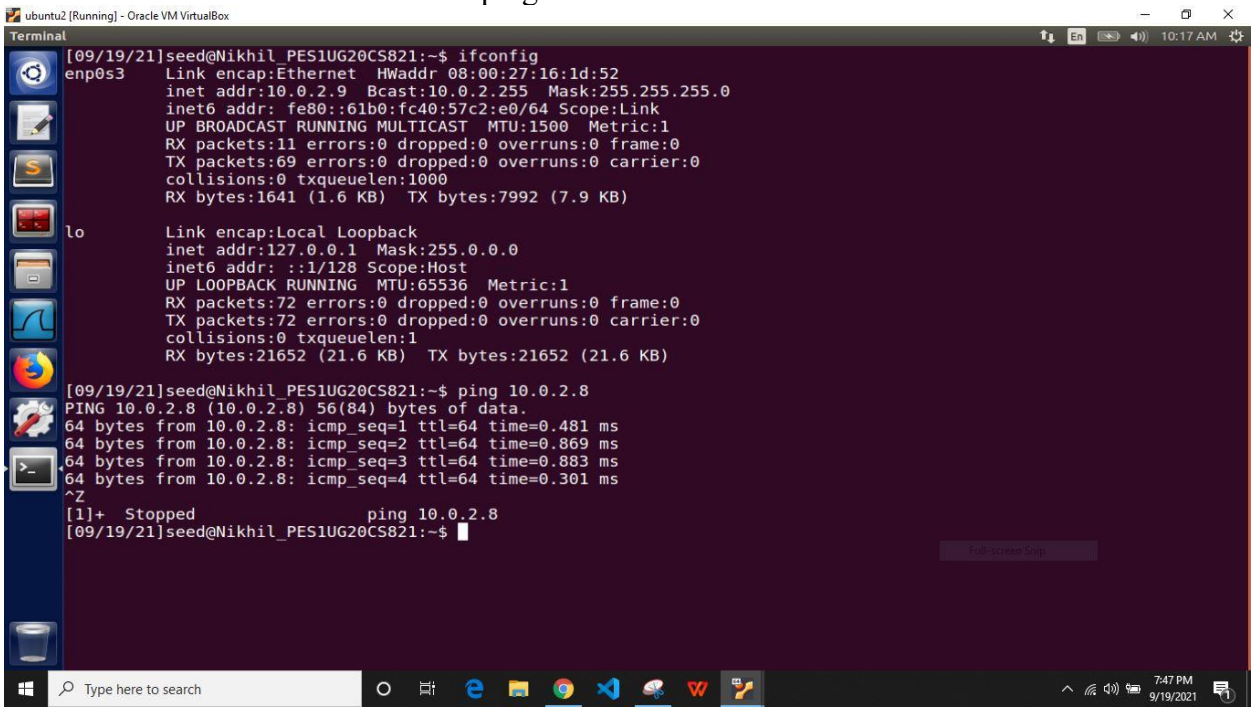
We run the command sudo python sample.py on the attacker machine (ubuntu1) of IP address 10.0.2.8.because attacker is the one who want to sniff the packets.

Observation On attacker machine:we run the code on the attacker machine using sudo python sample.py and capture the packets.



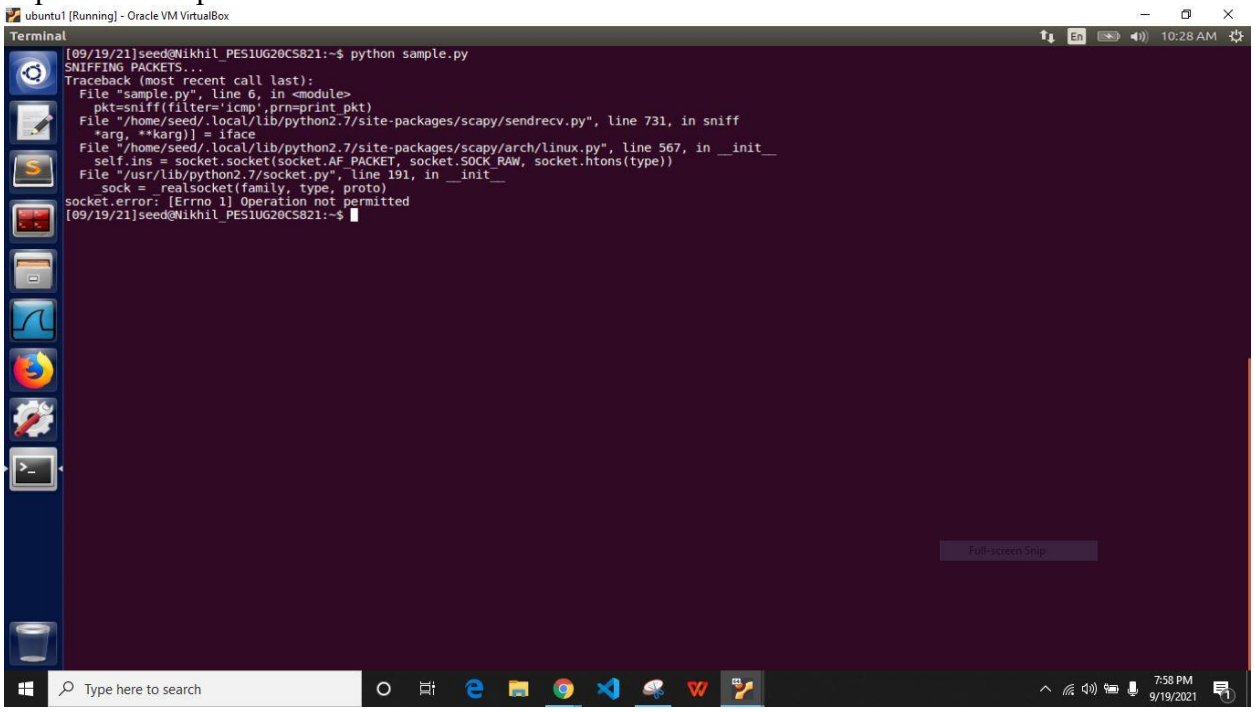


Observations On victim machine:we ping the attacker



We get to know that we can sniff the packets and gather information such such as packets source and destination IP address and mac address along with the port number ,length of the packet,ICMP type etc.

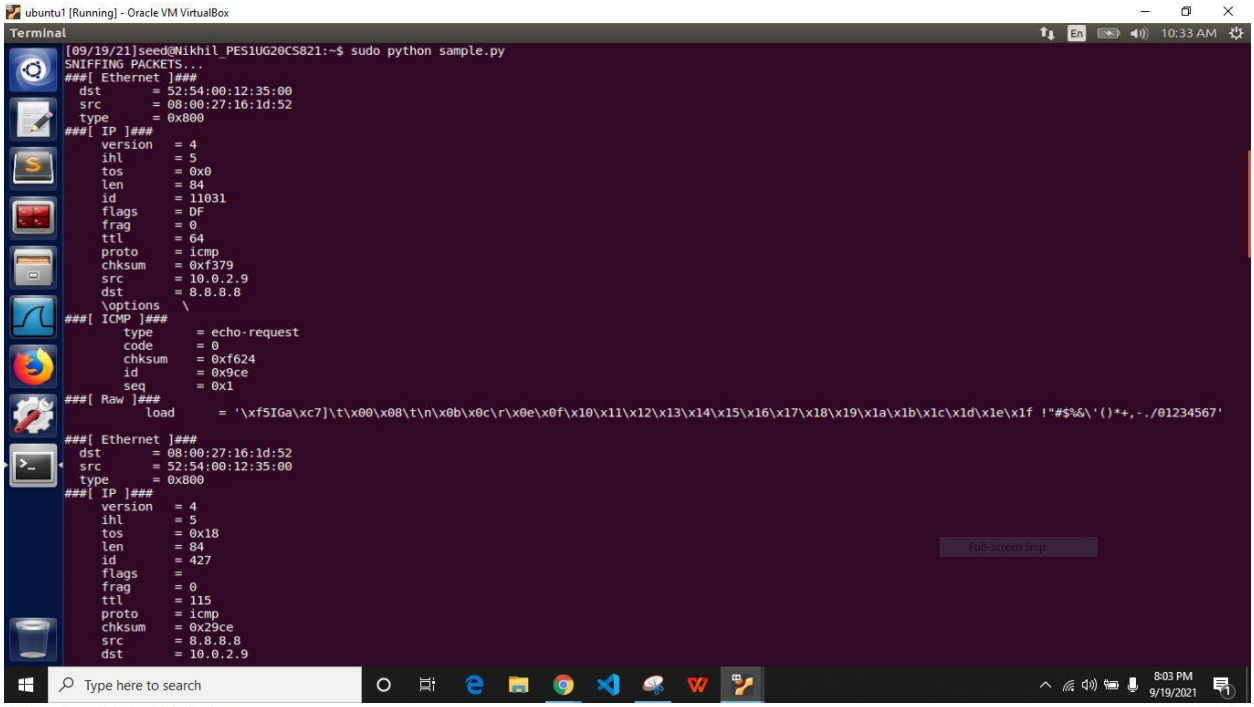
When we run the same program without root (sudo) privileges we end up with an error “operation not permitted “ as shown below

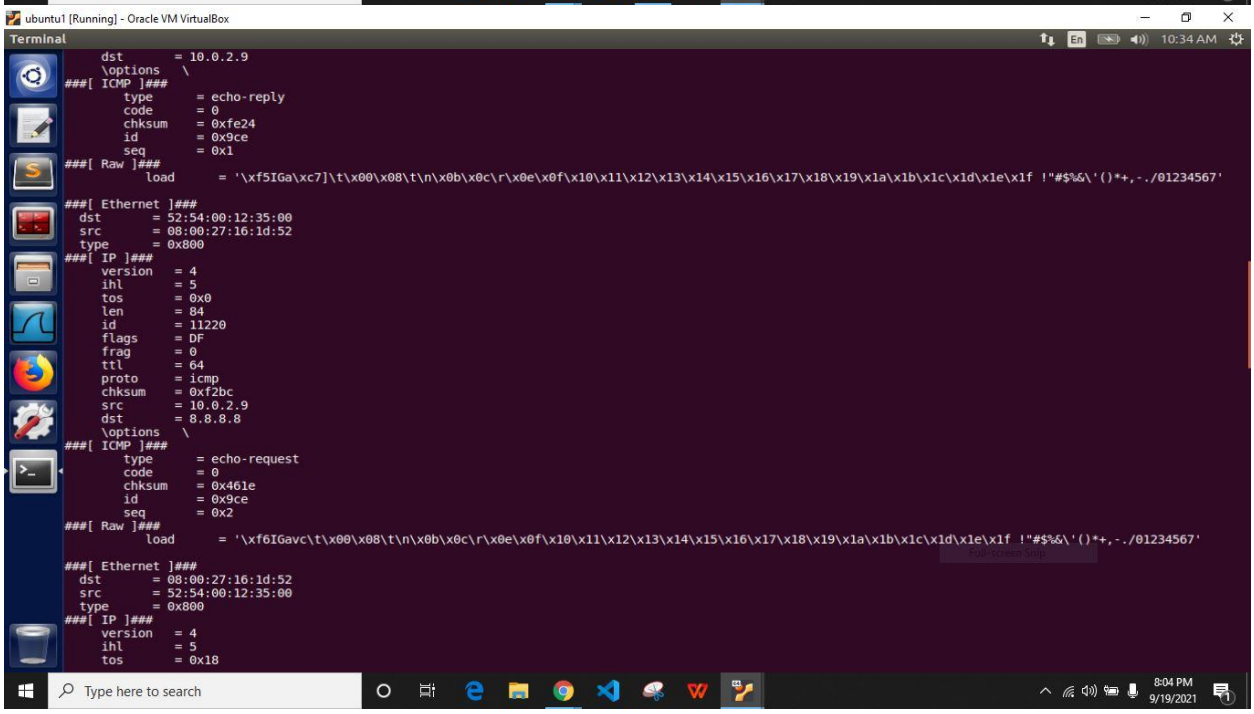


Task 1.2 Capturing ICMP, TCP packet and Subnet

Task 1.2.1 Capture only the ICMP packet

Observation on attacker machine



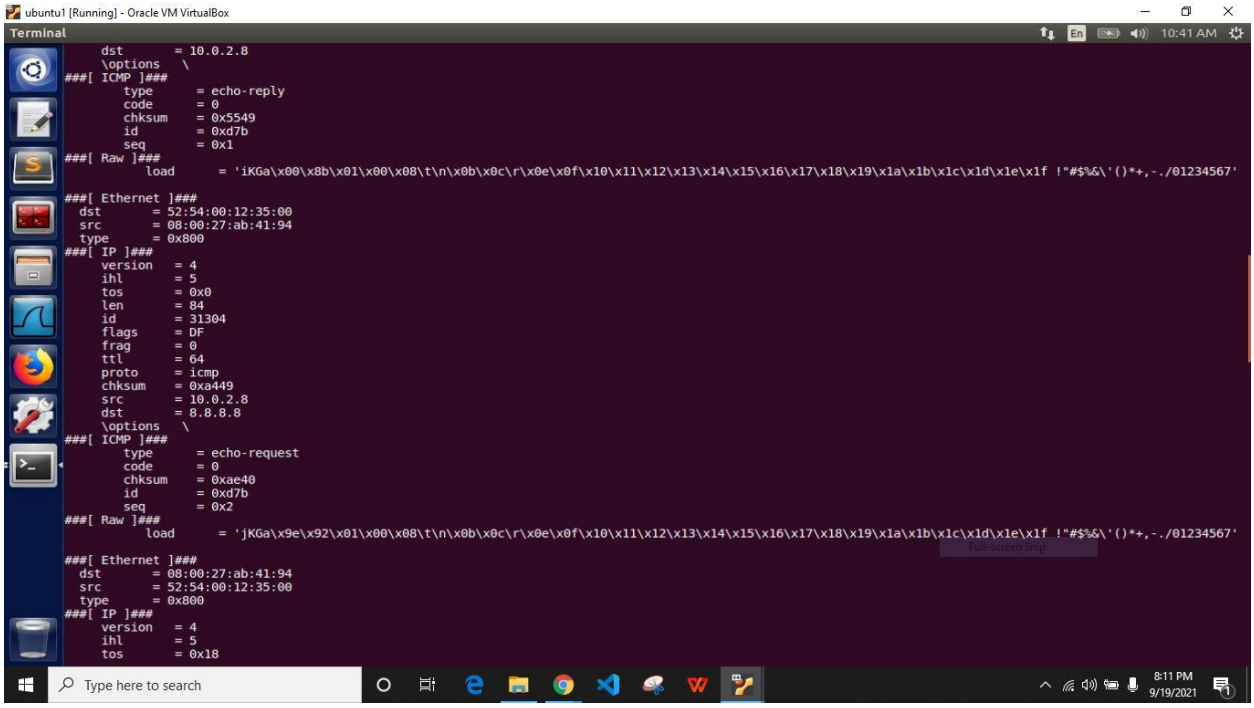


Observation on victim machine while pinging 8.8.8.8

```
ubuntu2 [Running] - Oracle VM VirtualBox
Terminal
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:16:1d:52
          inet addr:10.0.2.9  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::61b0:fc40:57c2:e0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:11 errors:0 dropped:0 overruns:0 frame:0
          TX packets:69 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1641 (1.6 KB)  TX bytes:7992 (7.9 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:72 errors:0 dropped:0 overruns:0 frame:0
          TX packets:72 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:21652 (21.6 KB)  TX bytes:21652 (21.6 KB)

[09/19/21]seed@Nikhil_PES1UG20CS821:~$ ping 10.0.2.8
PING 10.0.2.8 (10.0.2.8) 56(84) bytes of data.
64 bytes from 10.0.2.8: icmp_seq=1 ttl=64 time=0.481 ms
64 bytes from 10.0.2.8: icmp_seq=2 ttl=64 time=0.869 ms
64 bytes from 10.0.2.8: icmp_seq=3 ttl=64 time=0.883 ms
64 bytes from 10.0.2.8: icmp_seq=4 ttl=64 time=0.301 ms
^Z
[1]+  Stopped                 ping 10.0.2.8
[09/19/21]seed@Nikhil_PES1UG20CS821:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=115 time=9.10 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=115 time=9.02 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=115 time=10.0 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=115 time=8.95 ms
^Z
[2]+  Stopped                 ping 8.8.8.8
[09/19/21]seed@Nikhil_PES1UG20CS821:~$
```

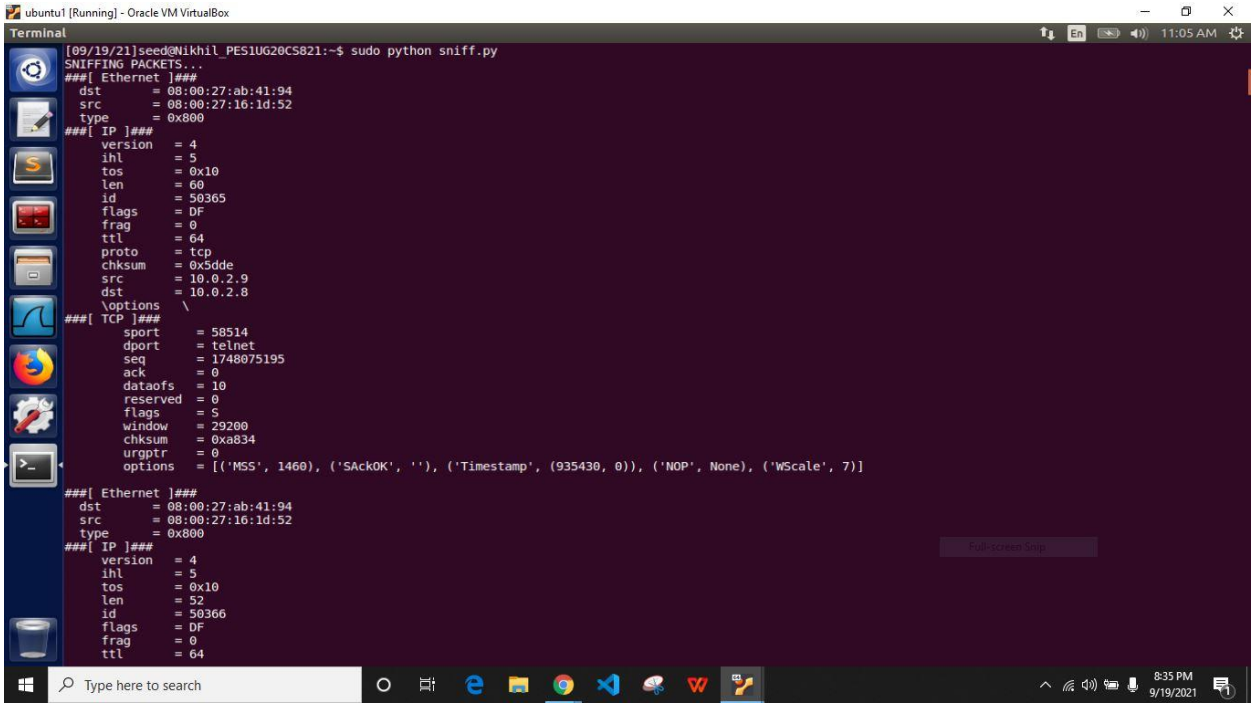
While running the same program on the attacker machine along with pinging 8.8.8.8 on the same virtual machine in another terminal.we can identify those two with the help of source IP address and Mac address.

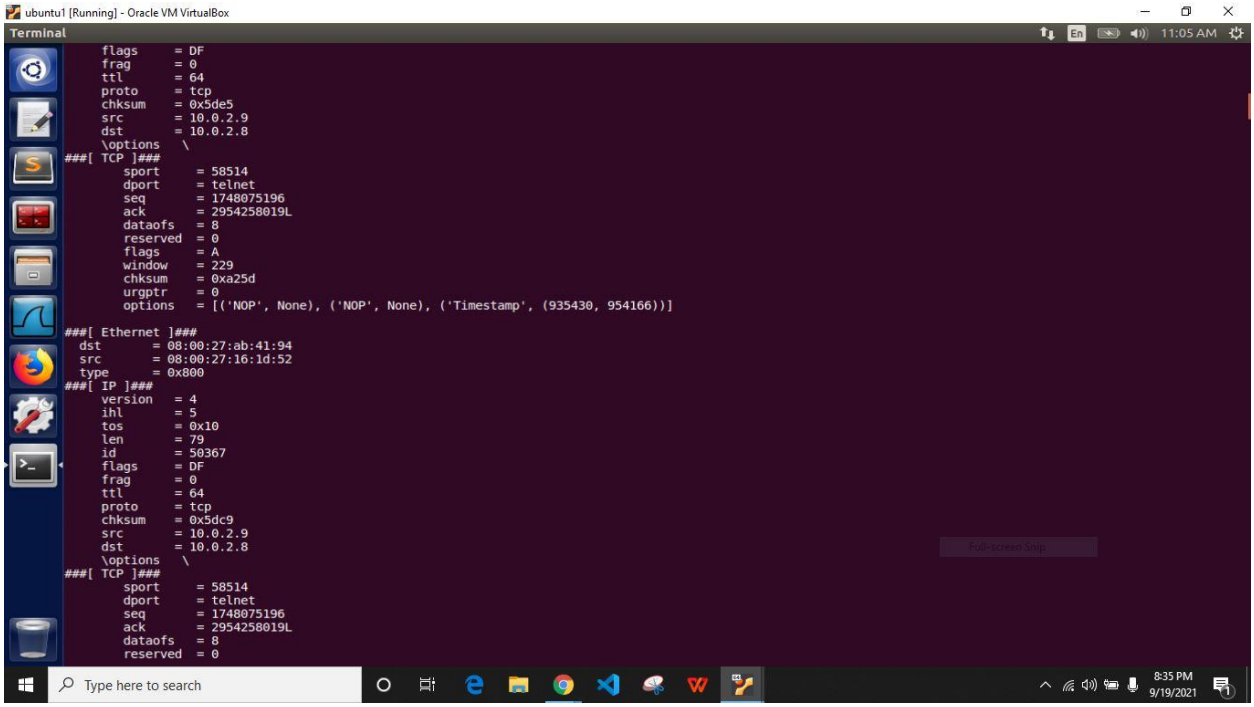
```
ubuntu1 [Running] - Oracle VM VirtualBox
Terminal
[09/19/21]seed@Nikhil_PES1UG20CS821:~$ sudo python sample.py
SNIFFING PACKETS...
###[ Ethernet ]###
  dst       = 52:54:00:12:35:00
  src       = 08:00:27:ab:41:94
  type      = 0x800
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 31197
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0xa4b4
     src       = 10.0.2.8
     dst       = 8.8.8.8
     \options   \
###[ ICMP ]###
        type      = echo-request
        code      = 0
        chksum    = 0x4d49
        id        = 0xd7b
        seq       = 0x1
###[ Raw ]###
           load      = 'iKGa\x00\x8b\x01\x00\x08\t\n\x0b\x0c\r\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !"#$%&\'()*+,-./01234567'

###[ Ethernet ]###
  dst       = 08:00:27:ab:41:94
  src       = 52:54:00:12:35:00
  type      = 0x800
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x18
     len       = 84
     id        = 431
     flags     =
     frag      = 0
     ttl       = 115
     proto     = icmp
     chksum    = 0x29cb
     src       = 8.8.8.8
     dst       = 10.0.2.8
```



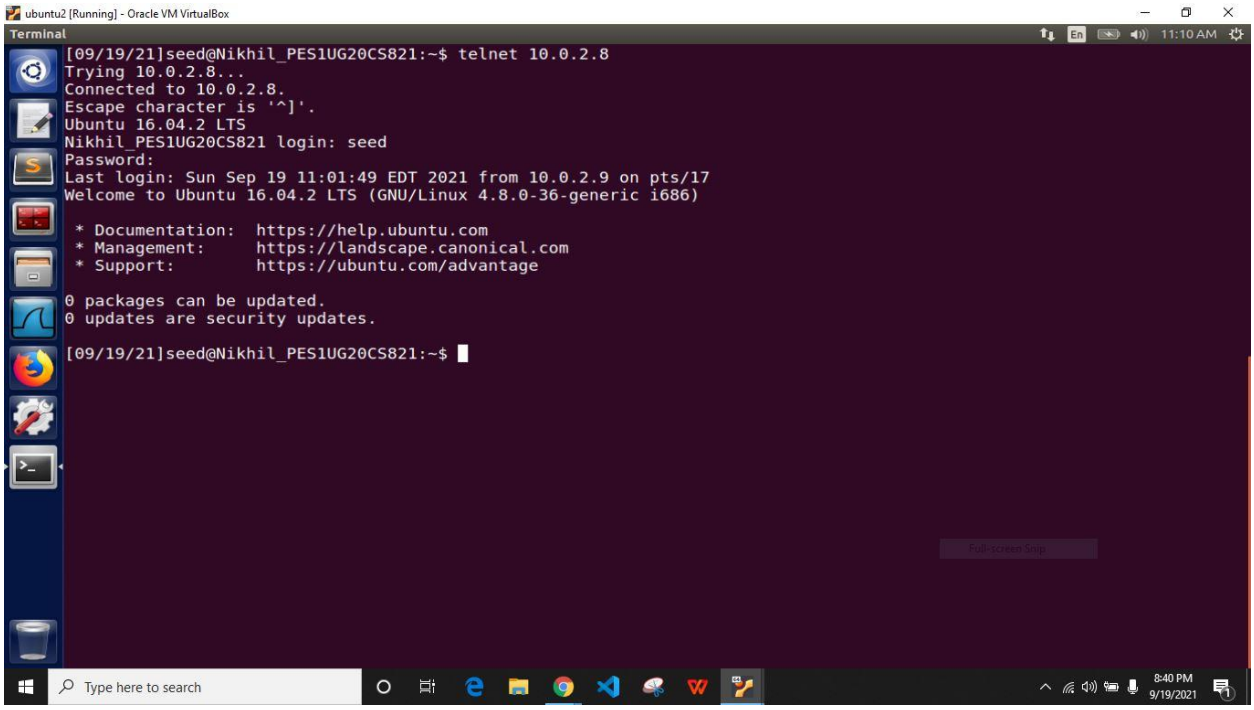
Task 1.2.2 Capture any TCP packet that comes from a particular IP and with a destination port number 23.

Observation on attacker machine





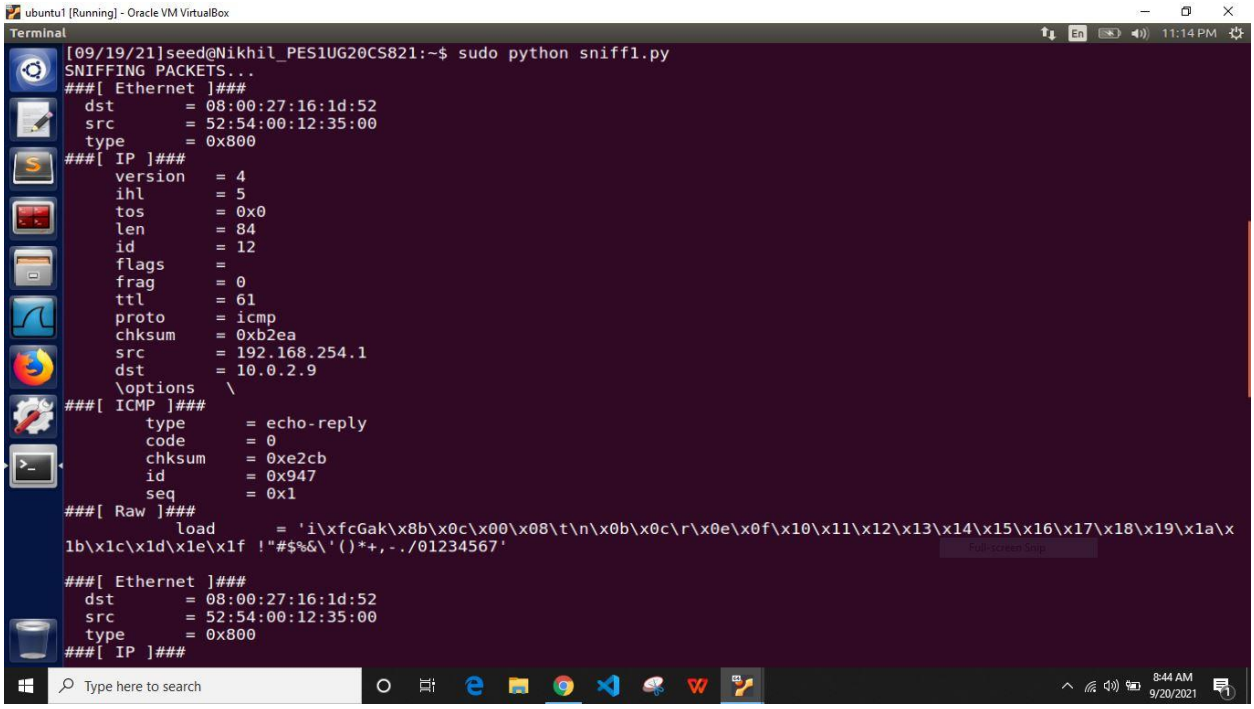
Observation victim machine



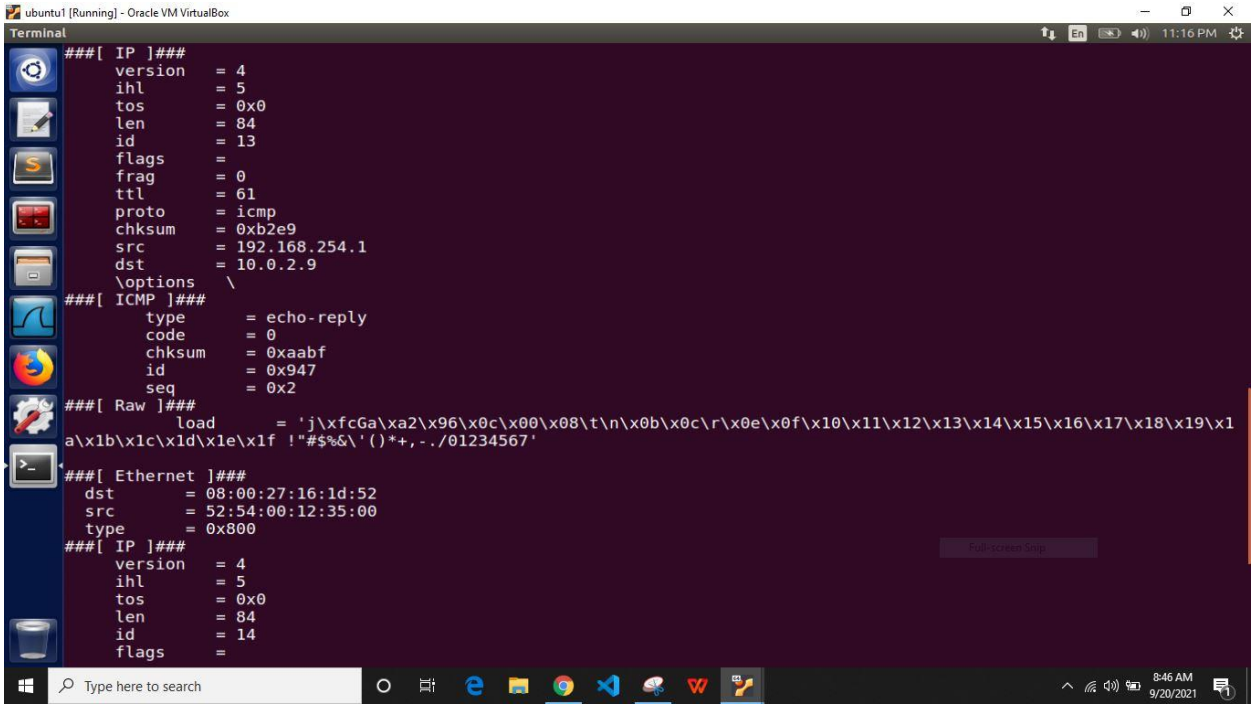
We run the telnet on the victim machine because Telnet is a computer protocol that provides two-way interactive communication compatibility for computers on the internet and local area networks.here we achieve this by using telnet on victim machine to get connect with the attacker machine.

Task 1.2.3 Capture packets comes from or to go to a particular subnet

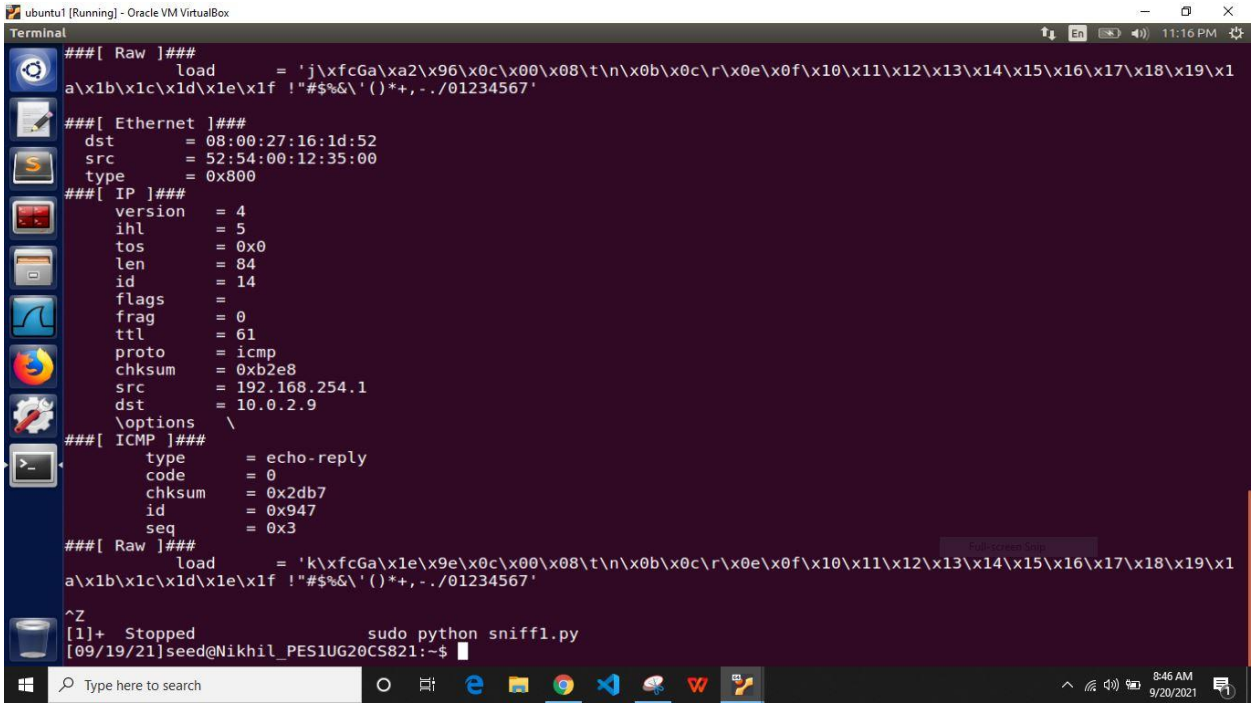
Observation on attacker machine



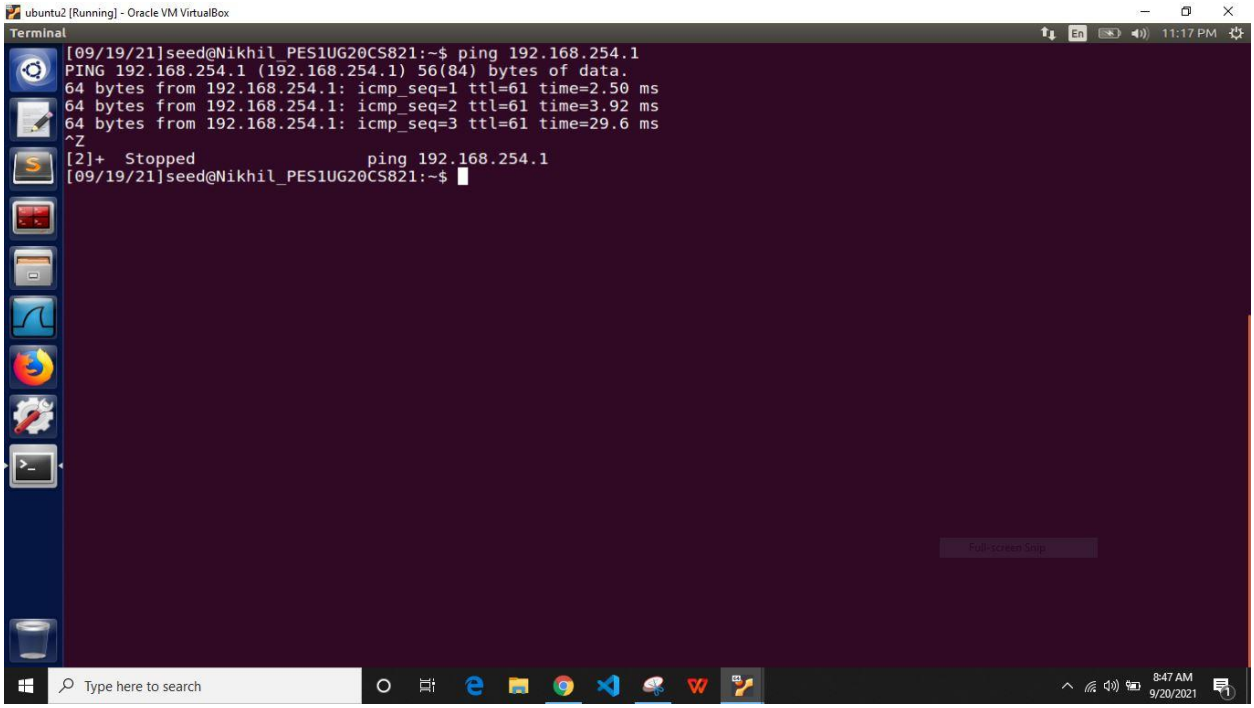
```
[09/19/21]seed@Nikhil_PES1UG20CS821:~$ sudo python sniff1.py
SNIFFING PACKETS...
###[ Ethernet ]###
  dst       = 08:00:27:16:1d:52
  src       = 52:54:00:12:35:00
  type      = 0x800
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 12
     flags     =
     frag      = 0
     ttl       = 61
     proto     = icmp
     chksum    = 0xb2ea
     src       = 192.168.254.1
     dst       = 10.0.2.9
     \options   \
###[ ICMP ]###
        type      = echo-reply
        code      = 0
        chksum    = 0xe2cb
        id        = 0x947
        seq       = 0x1
###[ Raw ]###
           load      = 'i\xfcGak\x8b\x0c\x00\x08\t\n\x0b\x0c\r\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x
1b\x1c\x1d\x1e\x1f !"#$%&\'()*+,-./01234567'

###[ Ethernet ]###
  dst       = 08:00:27:16:1d:52
  src       = 52:54:00:12:35:00
  type      = 0x800
###[ IP ]###
```



```
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 13
     flags     =
     frag      = 0
     ttl       = 61
     proto     = icmp
     chksum    = 0xb2e9
     src       = 192.168.254.1
     dst       = 10.0.2.9
     \options   \
###[ ICMP ]###
        type      = echo-reply
        code      = 0
        chksum    = 0xaabf
        id        = 0x947
        seq       = 0x2
###[ Raw ]###
           load      = 'j\xfcGa\xa2\x96\x0c\x00\x08\t\n\x0b\x0c\r\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1
a\x1b\x1c\x1d\x1e\x1f !"#$%&\'()*+,-./01234567'

###[ Ethernet ]###
  dst       = 08:00:27:16:1d:52
  src       = 52:54:00:12:35:00
  type      = 0x800
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 14
     flags     =
```



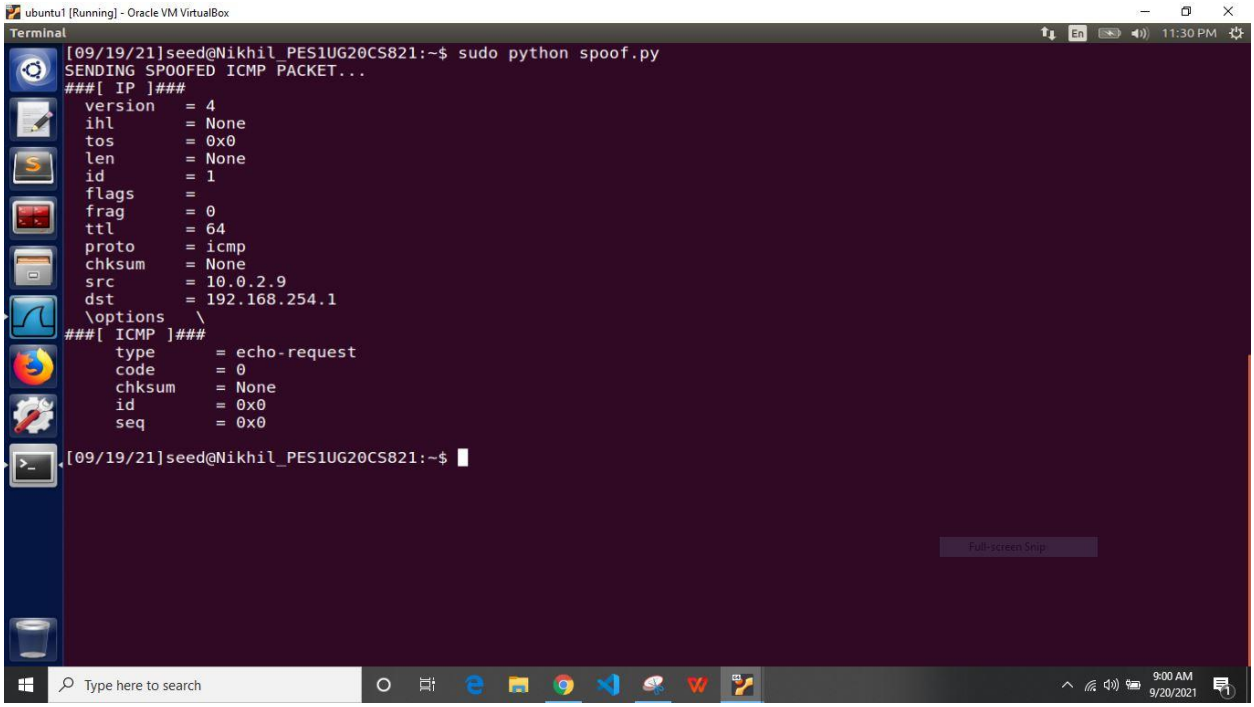
Observation on victim machine



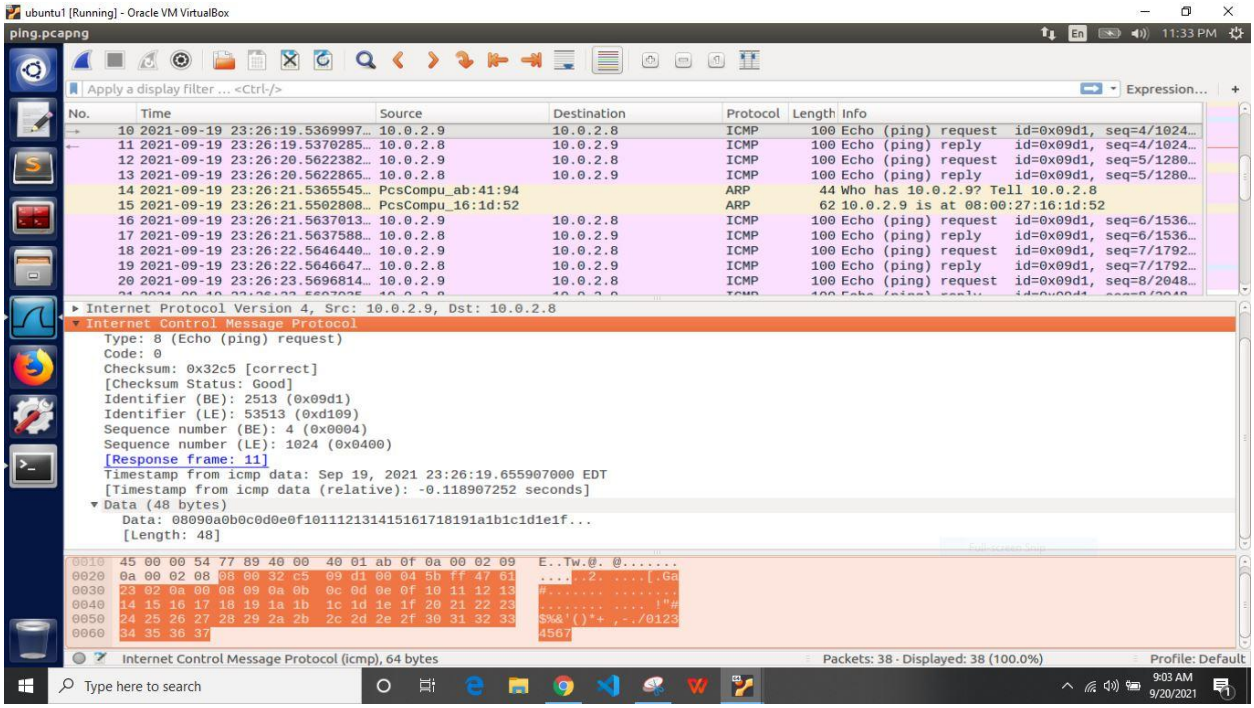
In this task the packets are sniffed from the particular subnet 192.168.254.0/24.
Whenever the victim try to access the IP address from 192.168.254.0 to 192.168.254.24 those packets are sniffed from the attacker.

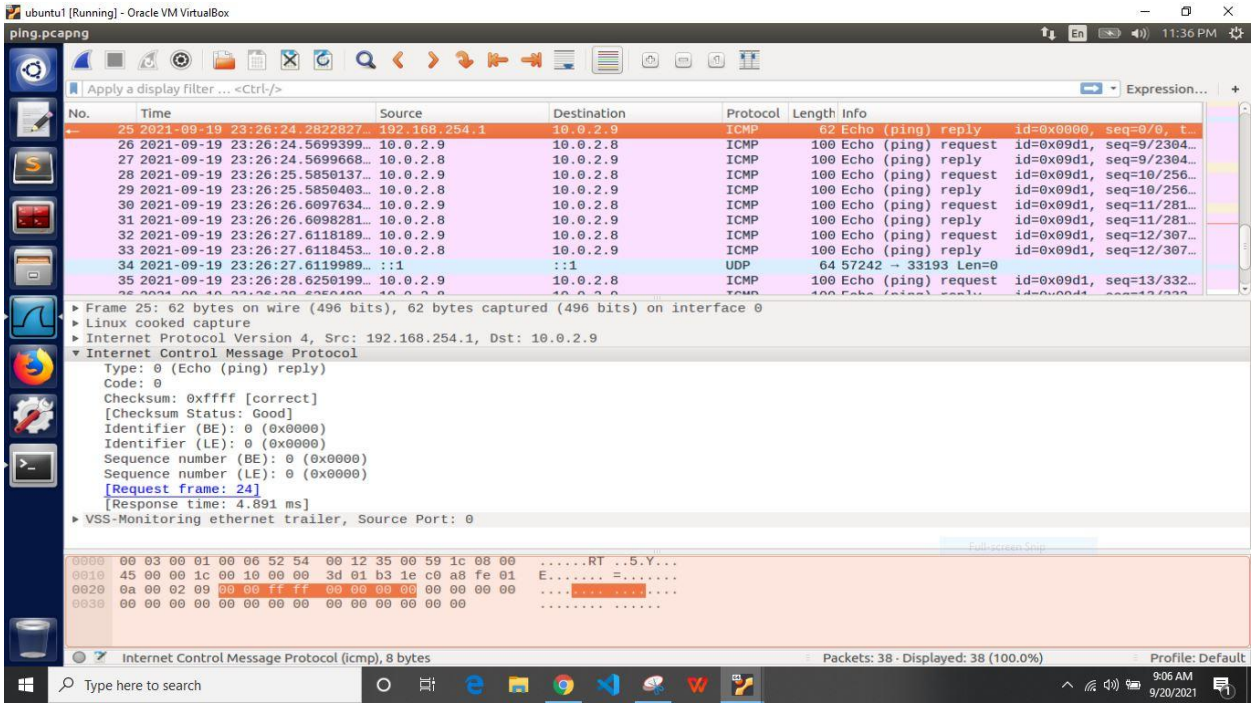
Task 2: Spoofing

Observation on attacker machine

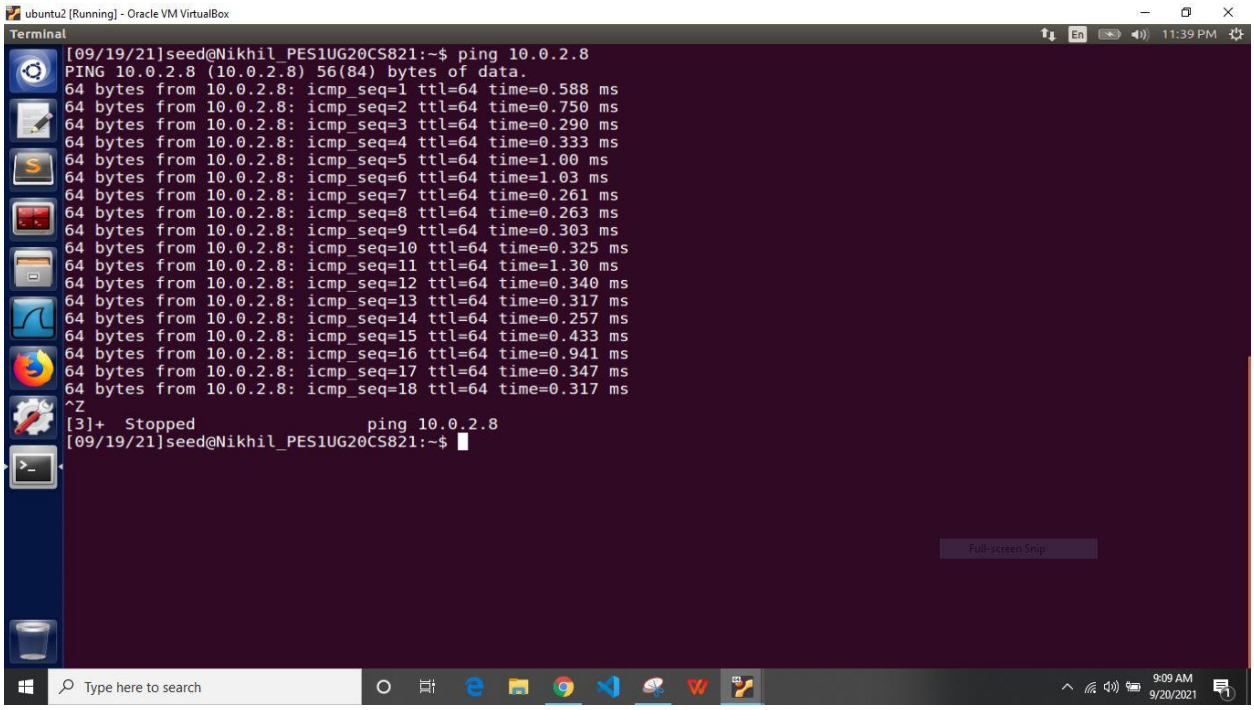


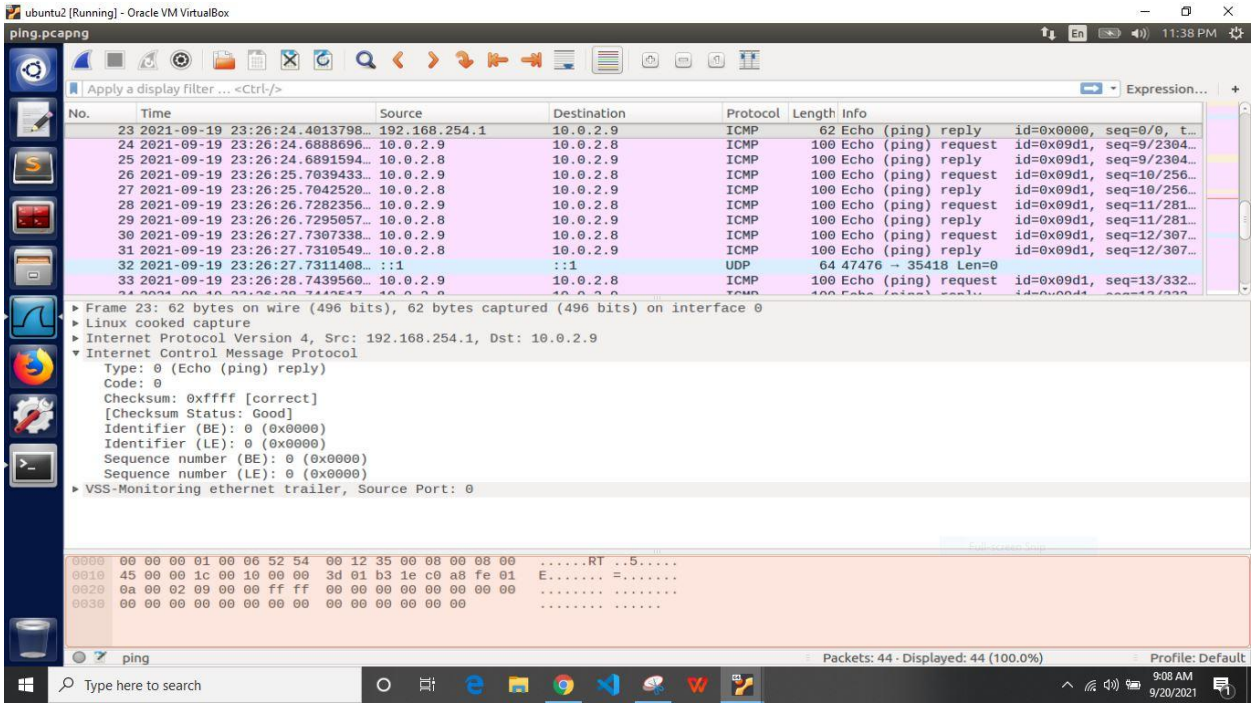
Observation on wireshark





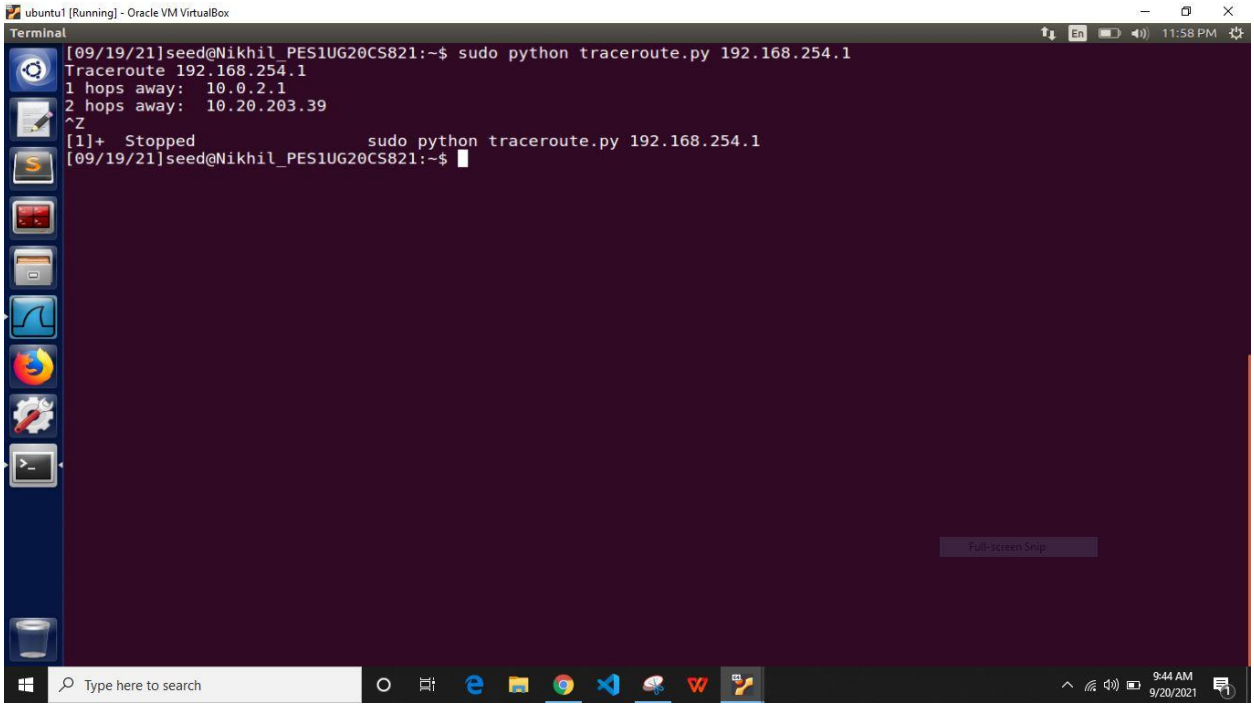
Observation on victim machine

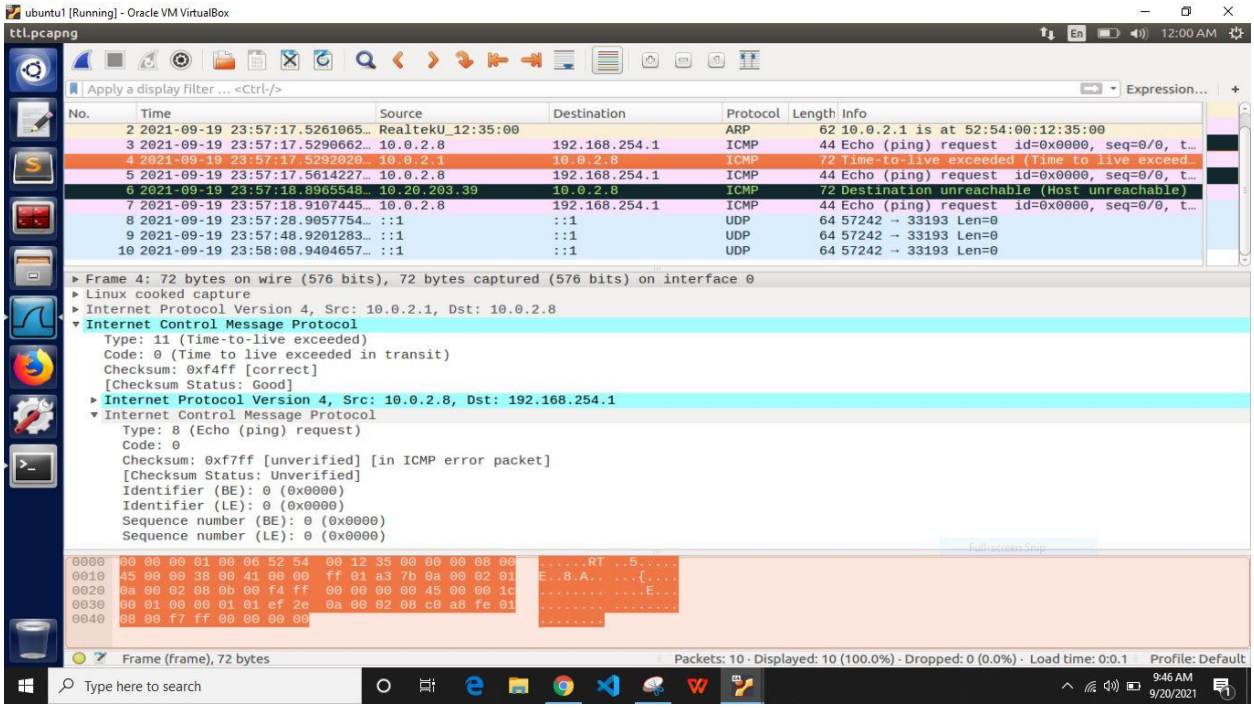




In this task the user is going to ping the attacker machine (10.0.2.8) then the user spoofs the packets and sends the packets to the IP address 192.168.254.1 with the victim IP address then the victim gets the reply from the IP address 192.168.254.1 even though he pinged 10.0.2.8.

Task 3: Traceroute

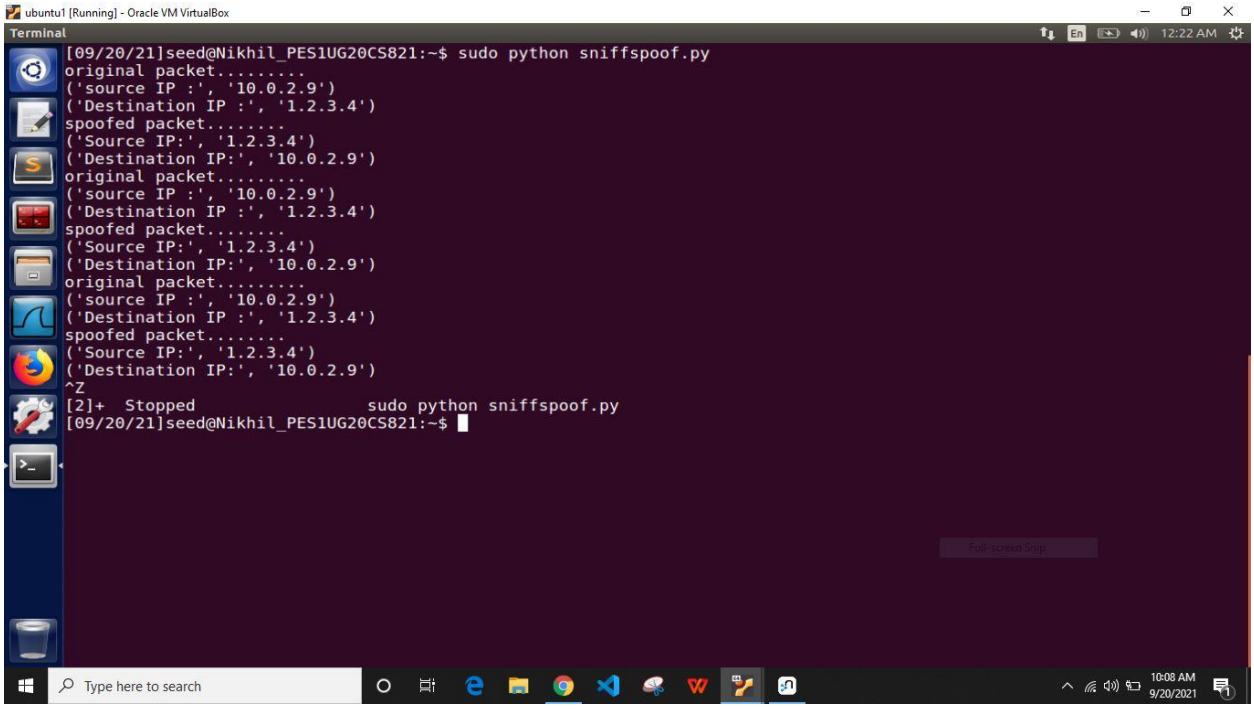




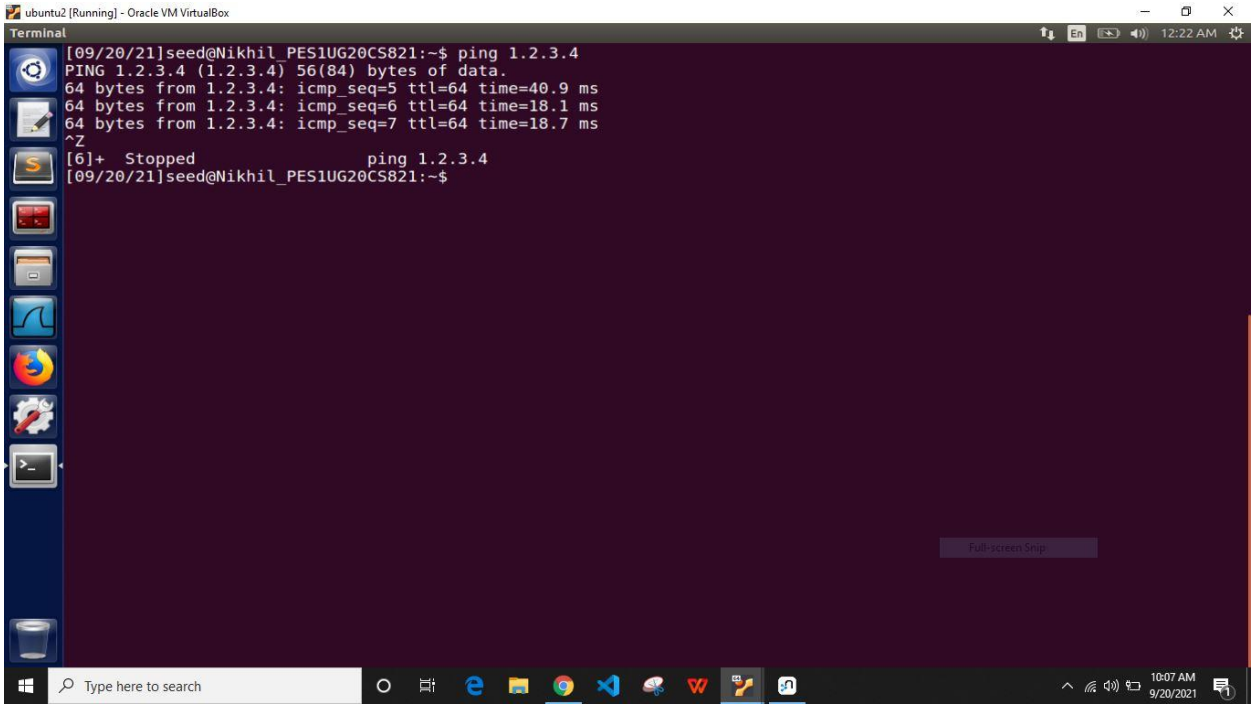
in this task we will do traceroute which is used to find the hops that the packet goes through in the network. We increase the ttl (time to live) value of the packets at last the error response form the router is obtained which shows Time to live exceed.

Task 4: Sniffing and-then Spoofing

Observation on attacker machine



Observation on victim machine



In this task both sniffing and spoofing are done by the attacker. The victim is going to ping the non existing IP address 1.2.3.4 but the victim will get the response this is done by the attacker. The attacker will sniff the packets and spoof it but the victim doesn't get to know that he is getting response from the attacker instead of machine having IP address 1.2.3.4.when the same same IP address is pinged from the same machine it results in no response.