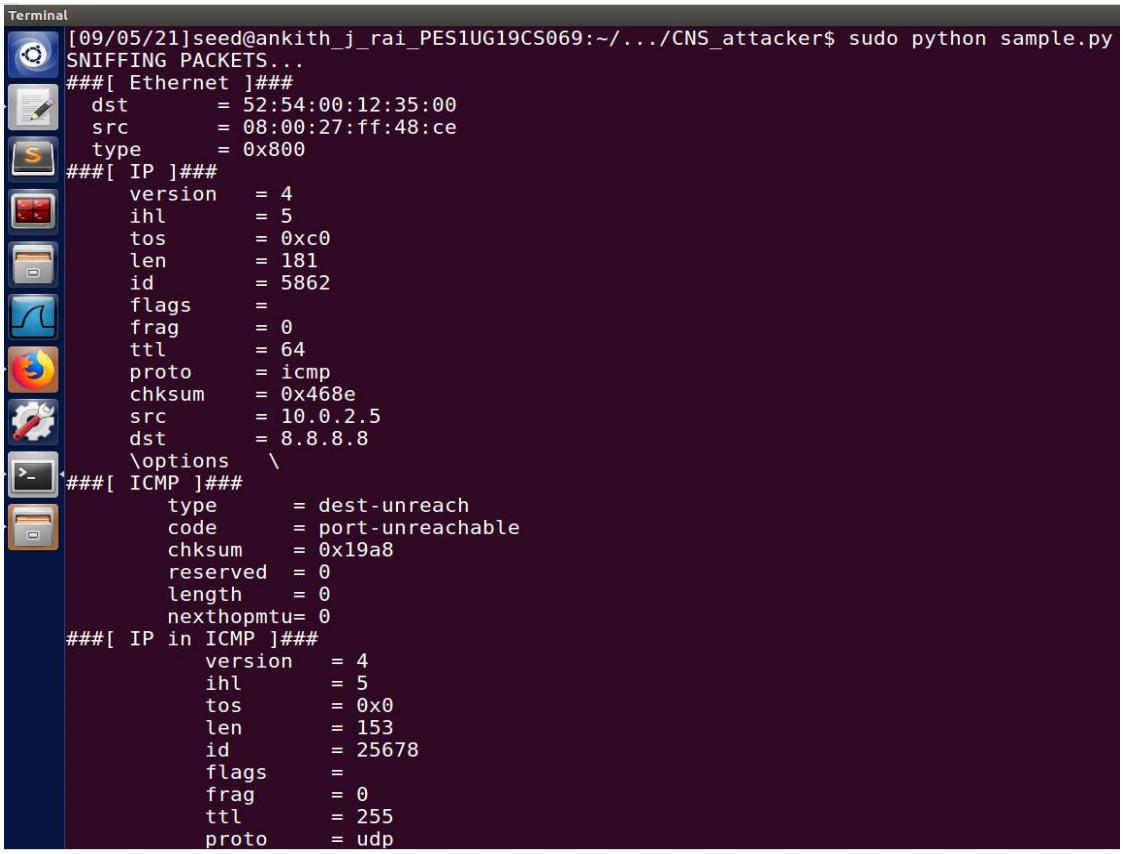
LAB-1 SNIFFING AND SPOOFING

Name : ANKITH J RAI
SRN : PES1UG19CS069
Sec : B

Attacker machine ip address : 10.0.2.5
Victim machine ip address : 10.0.2.7

2.1 Task 1: Sniffing Packets

A) With root privileges



```
Terminal
[09/05/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_attacker$ sudo python sample.py
SNIFFING PACKETS...
###[ Ethernet ]###
  dst       = 52:54:00:12:35:00
  src       = 08:00:27:ff:48:ce
  type      = 0x800
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0xc0
     len       = 181
     id        = 5862
     flags     =
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x468e
     src       = 10.0.2.5
     dst       = 8.8.8.8
     \options   \
###[ ICMP ]###
        type      = dest-unreach
        code      = port-unreachable
        chksum    = 0x19a8
        reserved  = 0
        length    = 0
        nexthopmtu= 0
###[ IP in ICMP ]###
           version   = 4
           ihl       = 5
           tos       = 0x0
           len       = 153
           id        = 25678
           flags     =
           frag      = 0
           ttl       = 255
           proto     = udp
```

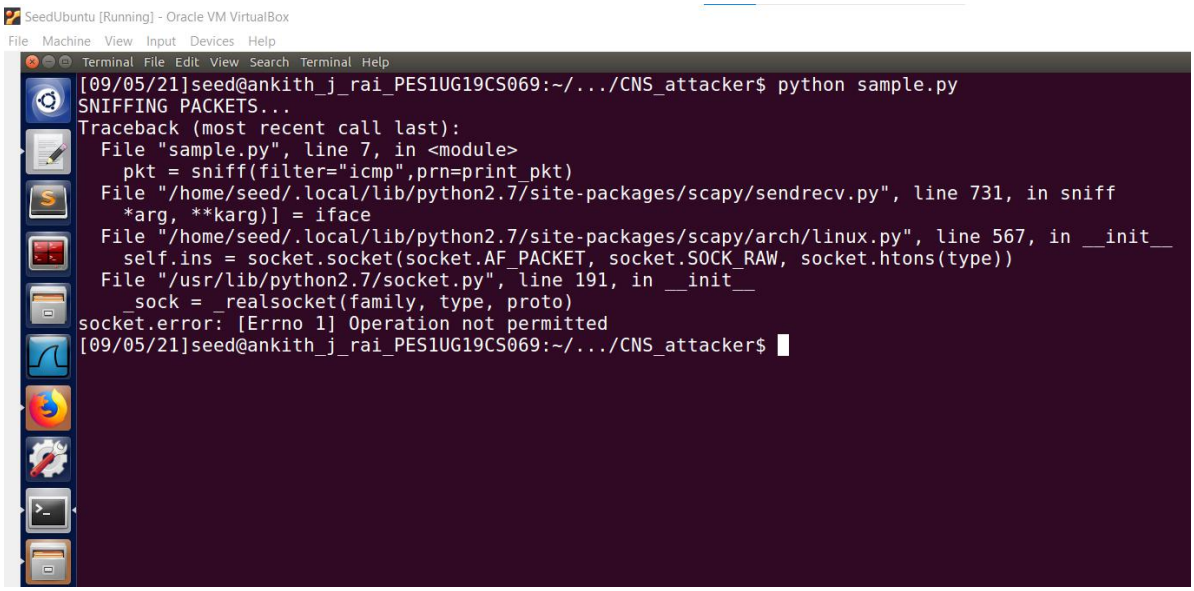
```
Terminal
            chksum    = 0x3af1
            src       = 8.8.8.8
            dst       = 10.0.2.5
            \options   \
###[ UDP in ICMP ]###
               sport     = domain
               dport     = 11037
               len       = 133
               chksum    = 0x869e
###[ DNS ]###
                  id        = 12542
                  qr        = 1
                  opcode    = QUERY
                  aa        = 0
                  tc        = 0
                  rd        = 1
                  ra        = 1
                  z         = 0
                  ad        = 0
                  cd        = 0
                  rcode     = name-error
                  qdcount   = 1
                  ancount   = 0
                  nscount   = 1
                  arcount   = 0
                  \qd        \
                   |###[ DNS Question Record ]###
                   |  qname     = 'tiles.services.mozilla.com.'
                   |  qtype     = A
                   |  qclass    = IN
                  an        = None
                  \ns        \
                   |###[ DNS Resource Record ]###
                   |  rrname    = 'services.mozilla.com.'
                   |  type      = SOA
                   |  rclass    = IN
                   |  ttl       = 503
```

```
Terminal
                | rdlen    = 69
                | rdata    = '\x06ns-679\tawsdns-20\x03net\x00\x11awsdns-hostmaster\x06amazon\xc0#\x00\x00\x00\x01\x00\x00\x1c \x00\
x00\x03\x84\x00\x12u\x00\x00\x01Q\x80'
             ar       = None
```

Q)Explain on which VM you ran this command and why?

Ans) The VM on which the sample.py is run is on the seedubuntu that is the attacker machine.It is run on the attacker machine because the attacker machine is the one which sniffs the packets.

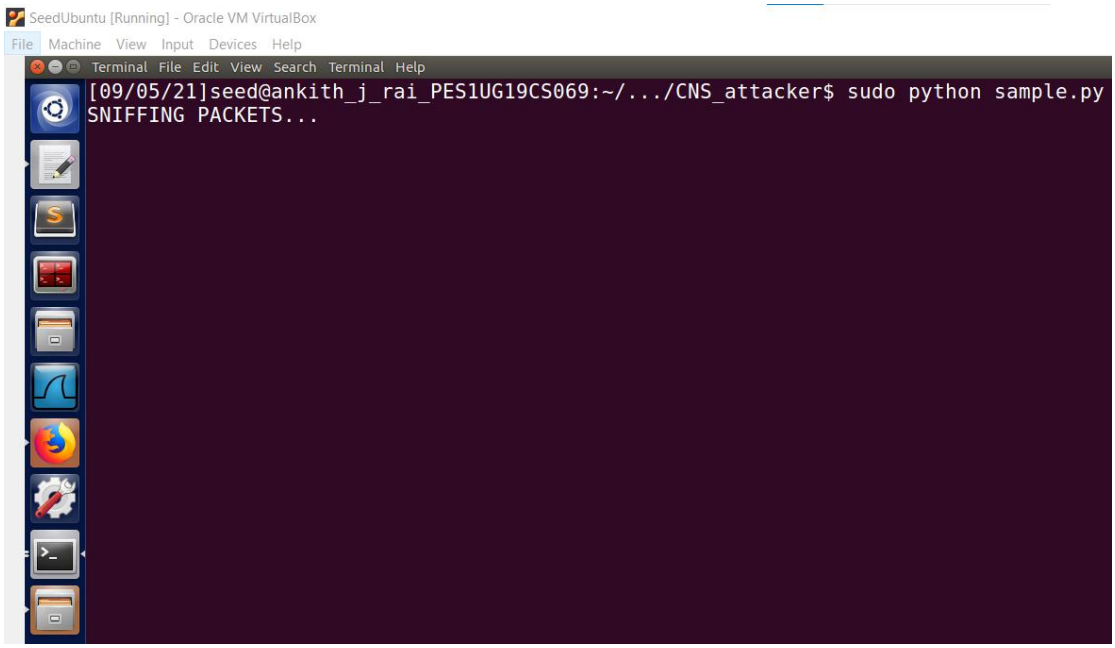
B) <u>Without root privileges</u>



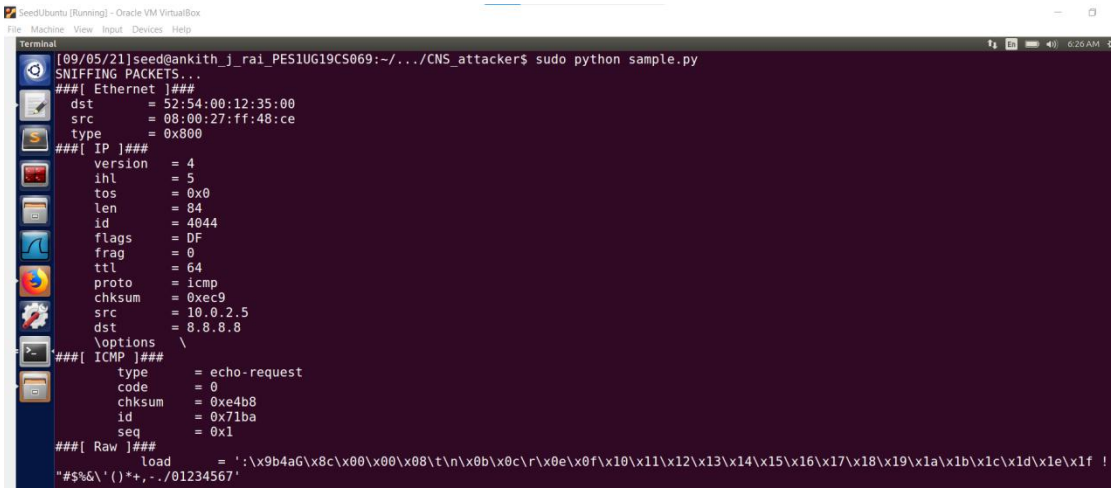
Yes,we face issues.the error we get is a socket error(where the operation of sniffing is not permitted).this is because we are not the root user.

Task 1.2 : Capturing ICMP, TCP packet and Subnet

Capture only the ICMP packet

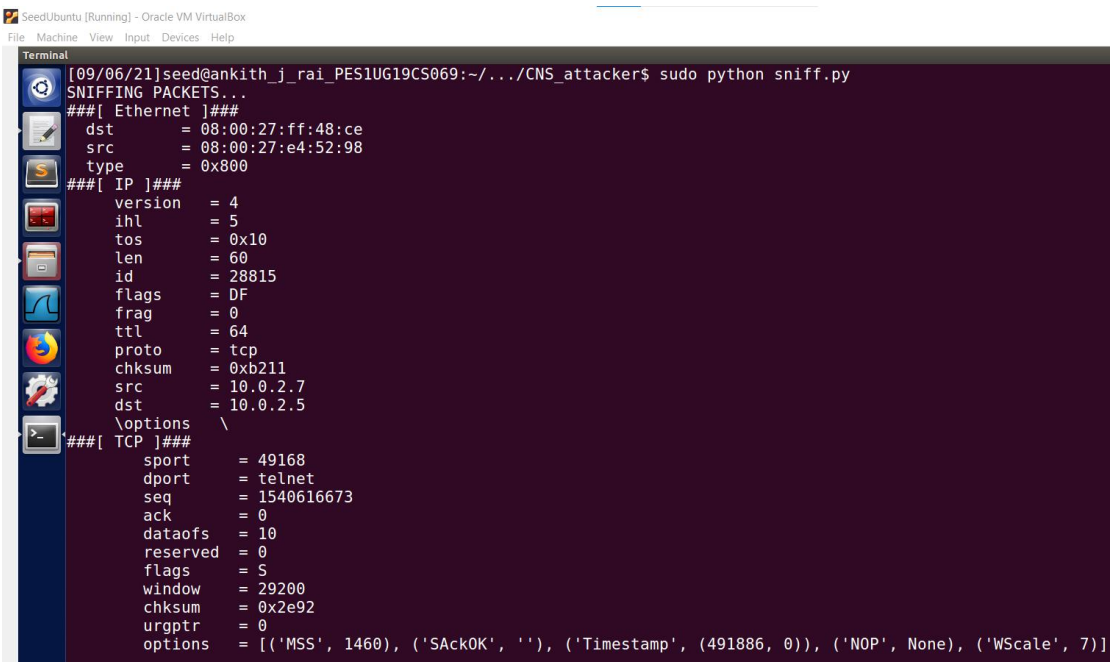


The above screenshot is of the attacker machine before it pings.



The above screenshot is of the attacker machine after it pings.

Capture any TCP packet that comes from a particular IP and with a destination port number 23(TELNET)



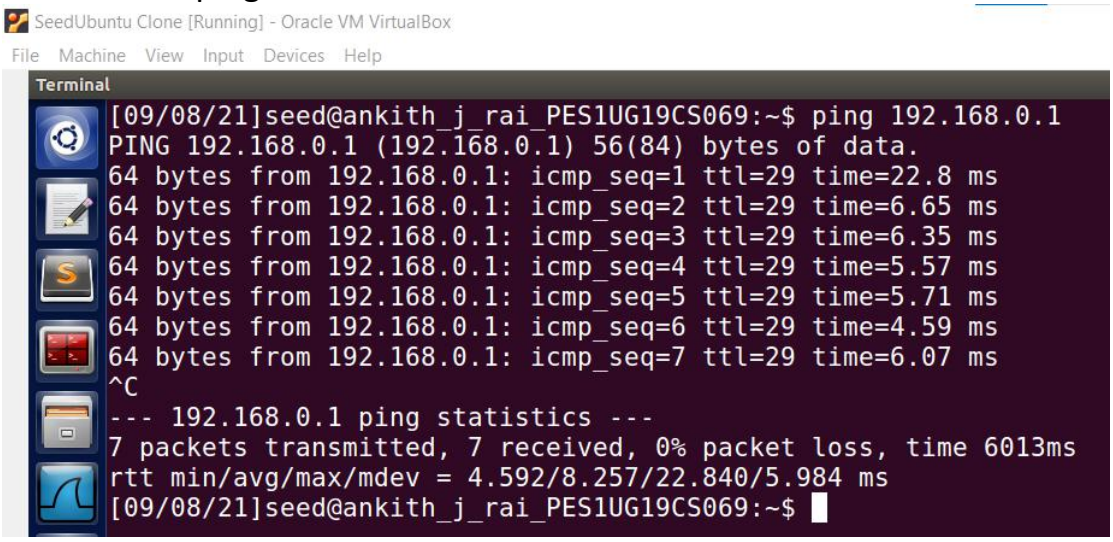
Q) Explain where you will run Telnet.

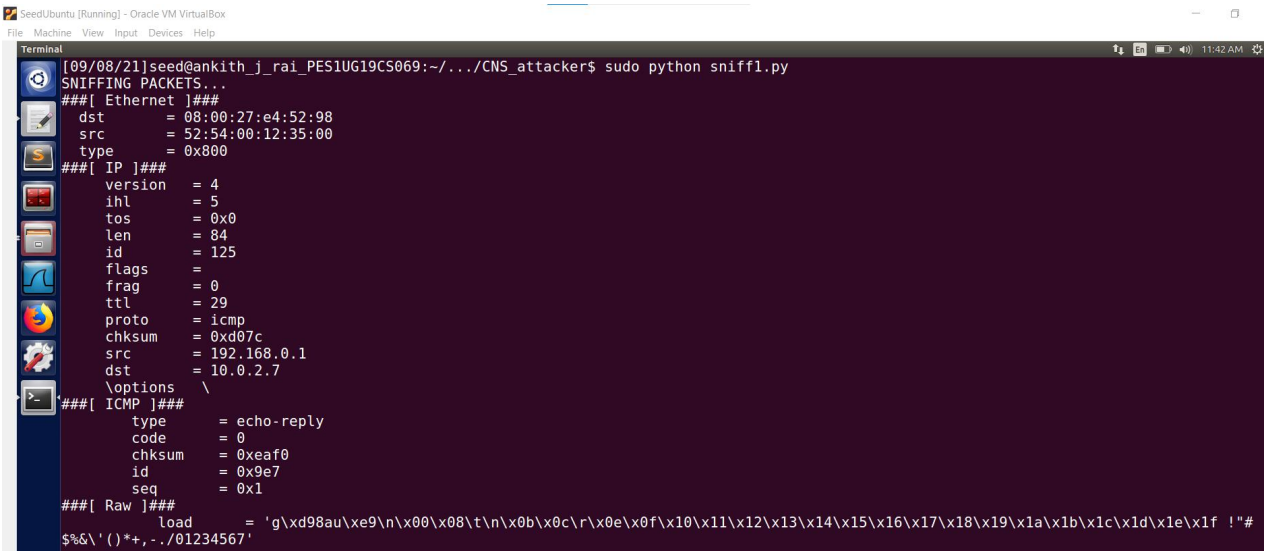
Ans) The telnet is run on the seedubuntu clone machine(which is the victim machine)

The above screenshot tells that the source is 10.0.2.7(victim machine ip address) and the destination is 10.0.2.5(attack machine ip address) and the port is telnet.

Capture packets comes from or to go to a particular subnet

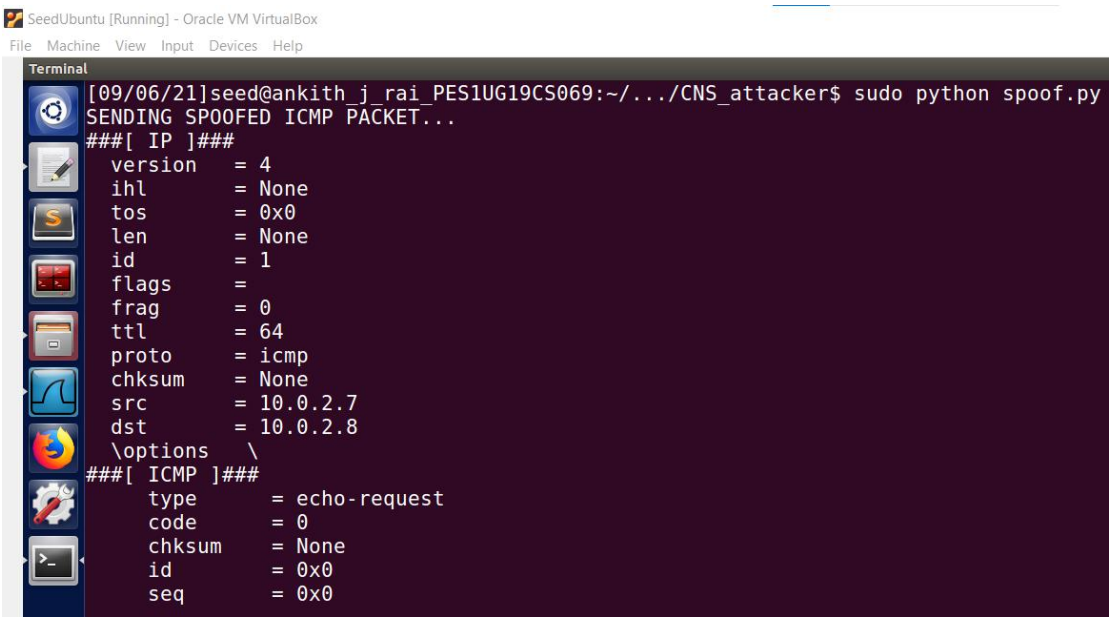
Now we will ping 192.168.0.1 from victim machine

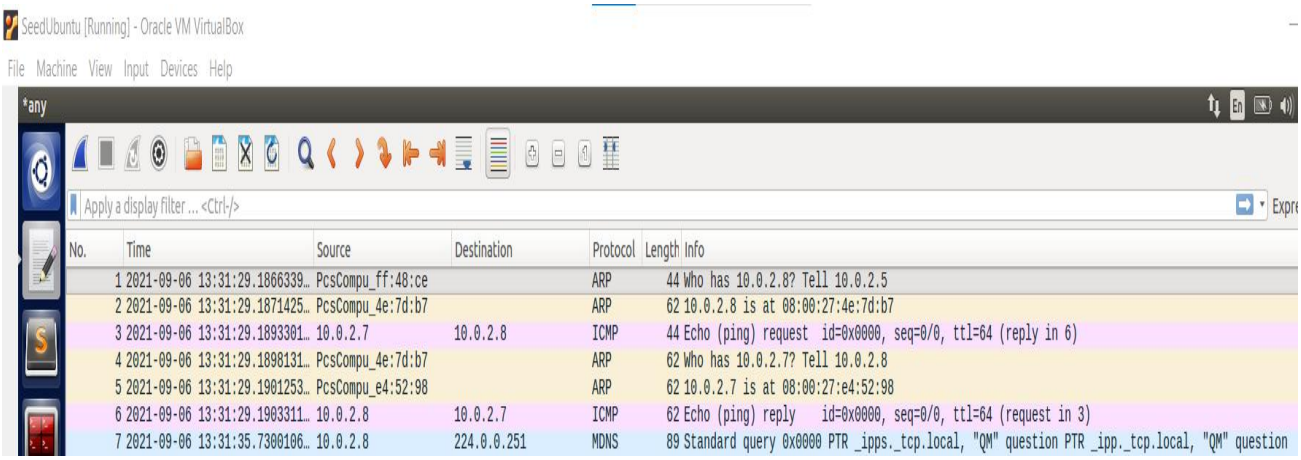


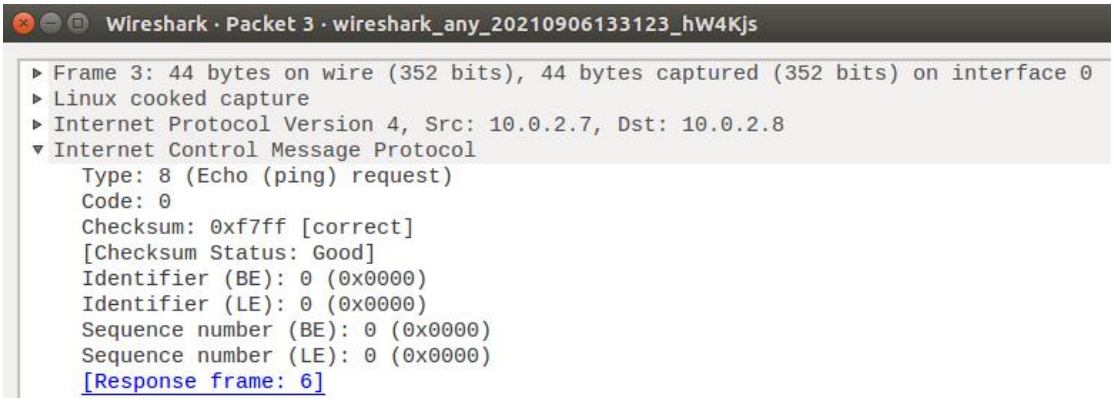


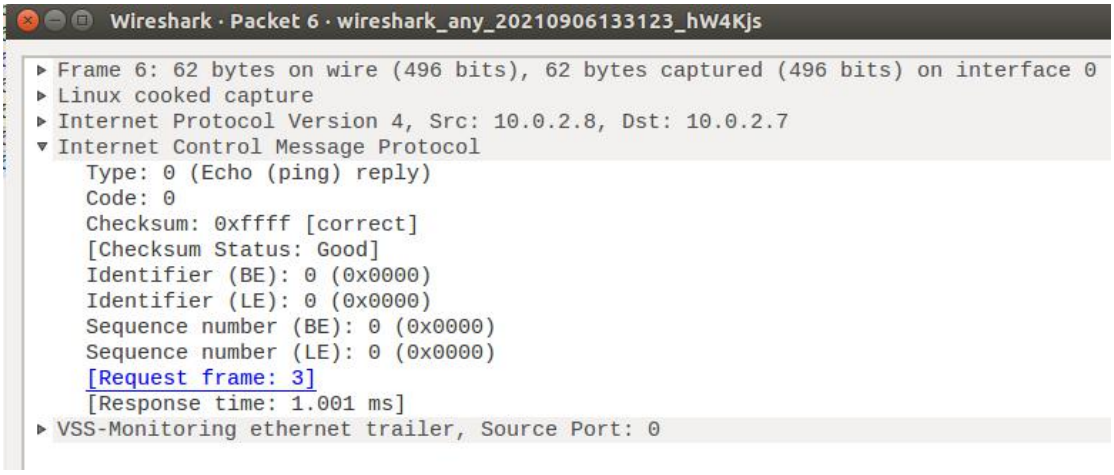
The above screenshot shows that attacker is sniffing the echo-reply sent from src:192.168.0.1 to dst:10.0.2.7

Task 2: Spoofing



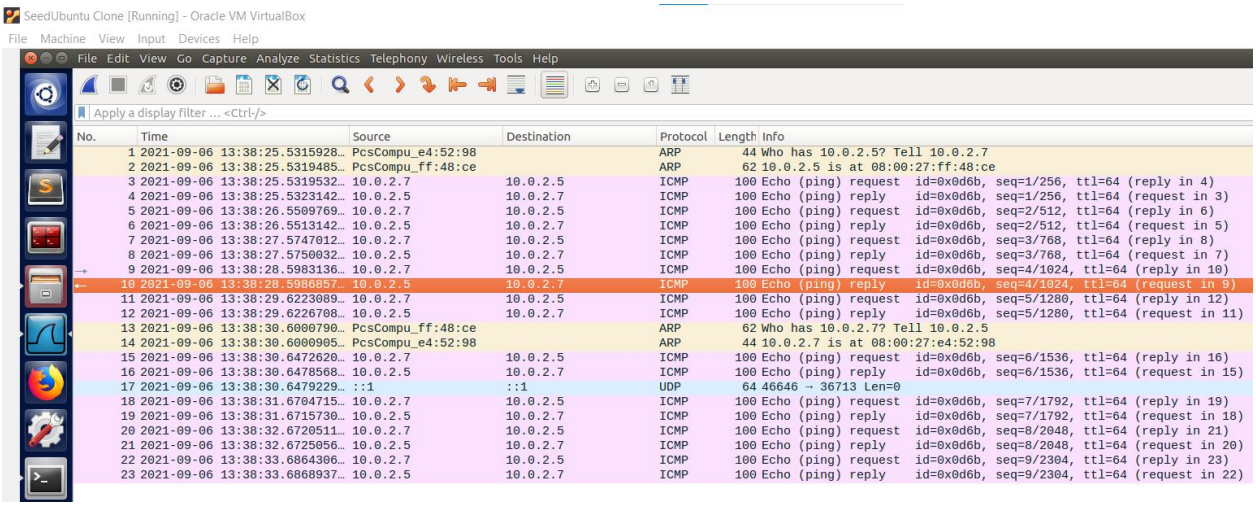






The above wireshark screenshots is of the response from the live machine.

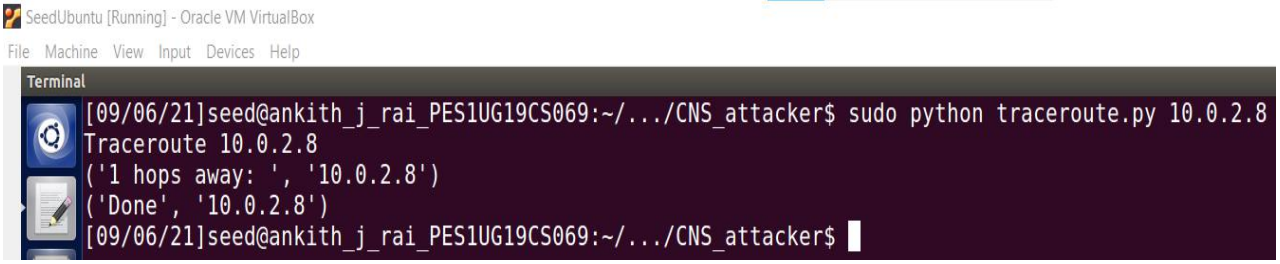
Screenshot of ping 10.0.2.5

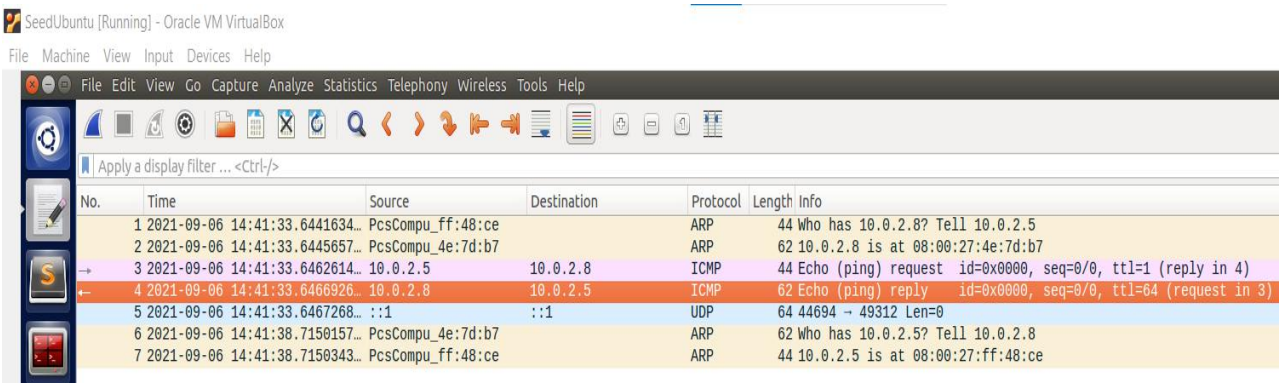


The above is the screenshot of the wireshark on pinging 10.0.2.5

<u>Task 3:</u> Traceroute

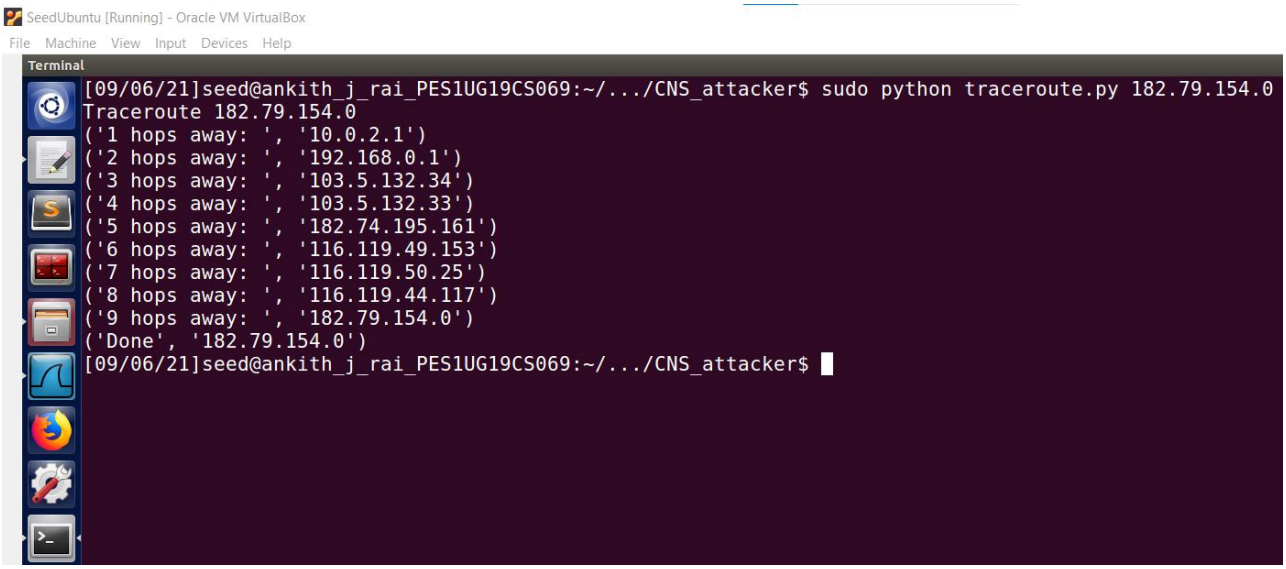
<u>Case 1</u>:let the host be **10.0.2.8**

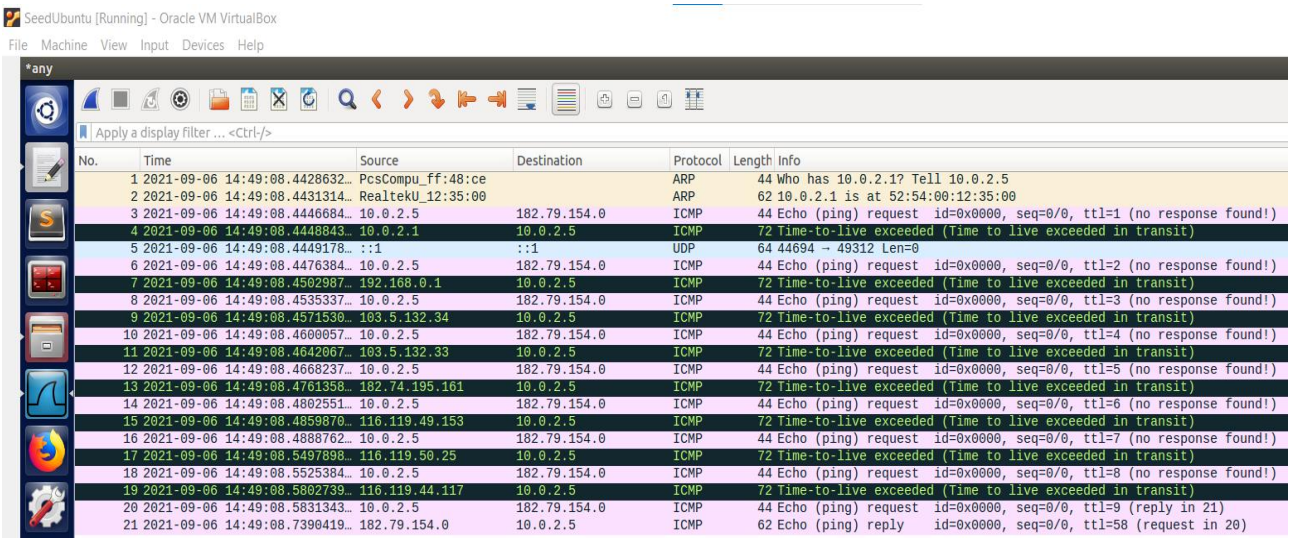




No.	Time	Source	Destination	Protocol	Length	Info
1	2021-09-06 14:41:33.6441634…	PcsCompu_ff:48:ce		ARP	44	Who has 10.0.2.8? Tell 10.0.2.5
2	2021-09-06 14:41:33.6445657…	PcsCompu_4e:7d:b7		ARP	62	10.0.2.8 is at 08:00:27:4e:7d:b7
3	2021-09-06 14:41:33.6462614…	10.0.2.5	10.0.2.8	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=1 (reply in 4)
4	2021-09-06 14:41:33.6466926…	10.0.2.8	10.0.2.5	ICMP	62	Echo (ping) reply id=0x0000, seq=0/0, ttl=64 (request in 3)
5	2021-09-06 14:41:33.6467268…	::1	::1	UDP	64	44694 → 49312 Len=0
6	2021-09-06 14:41:38.7150157…	PcsCompu_4e:7d:b7		ARP	62	Who has 10.0.2.5? Tell 10.0.2.8
7	2021-09-06 14:41:38.7150343…	PcsCompu_ff:48:ce		ARP	44	10.0.2.5 is at 08:00:27:ff:48:ce

Case2: let the host be **182.79.154.0**





No.	Time	Source	Destination	Protocol	Length	Info
1	2021-09-06 14:49:08.4428632…	PcsCompu_ff:48:ce		ARP	44	Who has 10.0.2.1? Tell 10.0.2.5
2	2021-09-06 14:49:08.4431314…	RealtekU_12:35:00		ARP	62	10.0.2.1 is at 52:54:00:12:35:00
3	2021-09-06 14:49:08.4446684…	10.0.2.5	182.79.154.0	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=1 (no response found!)
4	2021-09-06 14:49:08.4448843…	10.0.2.1	10.0.2.5	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
5	2021-09-06 14:49:08.4449178…	::1	::1	UDP	64	44694 → 49312 Len=0
6	2021-09-06 14:49:08.4476384…	10.0.2.5	182.79.154.0	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=2 (no response found!)
7	2021-09-06 14:49:08.4502987…	192.168.0.1	10.0.2.5	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
8	2021-09-06 14:49:08.4535337…	10.0.2.5	182.79.154.0	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=3 (no response found!)
9	2021-09-06 14:49:08.4571530…	103.5.132.34	10.0.2.5	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
10	2021-09-06 14:49:08.4600057…	10.0.2.5	182.79.154.0	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=4 (no response found!)
11	2021-09-06 14:49:08.4642067…	103.5.132.33	10.0.2.5	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
12	2021-09-06 14:49:08.4668237…	10.0.2.5	182.79.154.0	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=5 (no response found!)
13	2021-09-06 14:49:08.4761358…	182.74.195.161	10.0.2.5	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
14	2021-09-06 14:49:08.4802551…	10.0.2.5	182.79.154.0	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=6 (no response found!)
15	2021-09-06 14:49:08.4859870…	116.119.49.153	10.0.2.5	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
16	2021-09-06 14:49:08.4888762…	10.0.2.5	182.79.154.0	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=7 (no response found!)
17	2021-09-06 14:49:08.5497898…	116.119.50.25	10.0.2.5	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
18	2021-09-06 14:49:08.5525384…	10.0.2.5	182.79.154.0	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=8 (no response found!)
19	2021-09-06 14:49:08.5802739…	116.119.44.117	10.0.2.5	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
20	2021-09-06 14:49:08.5831343…	10.0.2.5	182.79.154.0	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=9 (reply in 21)
21	2021-09-06 14:49:08.7390419…	182.79.154.0	10.0.2.5	ICMP	62	Echo (ping) reply id=0x0000, seq=0/0, ttl=58 (request in 20)

From the above wireshark screenshot we can see the error response message as **Time to live exceeded** in transit

Wireshark · Packet 6 · wireshark_any_20210906144902_3gir8P

```
▶ Frame 6: 44 bytes on wire (352 bits), 44 bytes captured (352 bits) on interface 0
▶ Linux cooked capture
▶ Internet Protocol Version 4, Src: 10.0.2.5, Dst: 182.79.154.0
▼ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xf7ff [correct]
    [Checksum Status: Good]
    Identifier (BE): 0 (0x0000)
    Identifier (LE): 0 (0x0000)
    Sequence number (BE): 0 (0x0000)
    Sequence number (LE): 0 (0x0000)
  ▶ [No response seen]
```

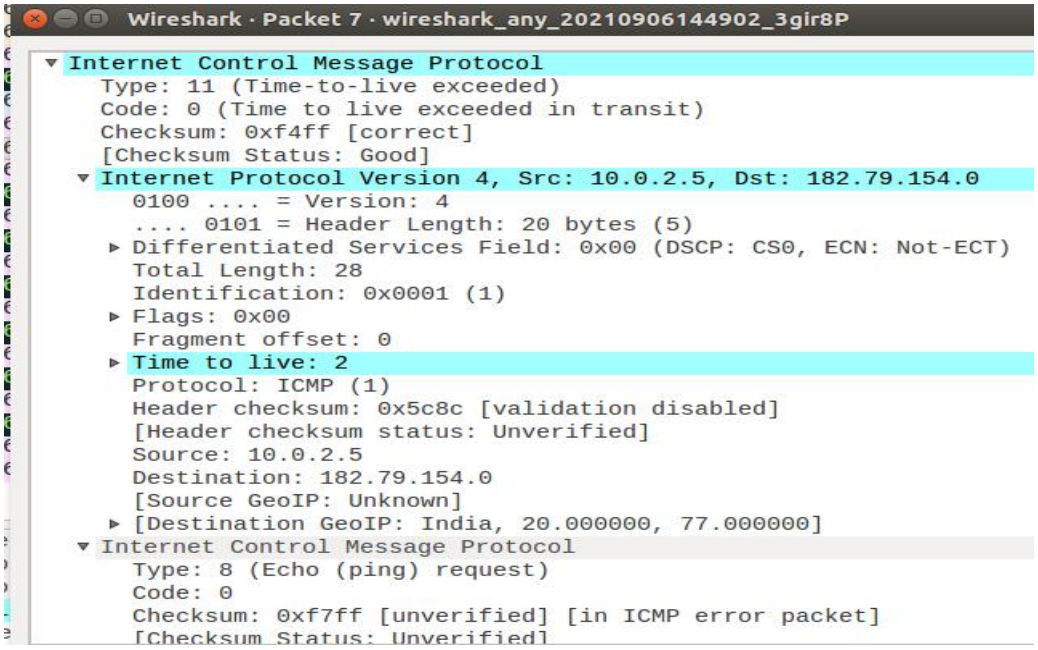
Wireshark · Packet 4 · wireshark_any_20210906144902_3gir8P

```
▶ Frame 4: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface 0
▶ Linux cooked capture
▶ Internet Protocol Version 4, Src: 10.0.2.1, Dst: 10.0.2.5
▼ Internet Control Message Protocol
    Type: 11 (Time-to-live exceeded)
    Code: 0 (Time to live exceeded in transit)
    Checksum: 0xf4ff [correct]
    [Checksum Status: Good]
  ▶ Internet Protocol Version 4, Src: 10.0.2.5, Dst: 182.79.154.0
  ▶ Internet Control Message Protocol
```

The below two screenshots indicate ttl to be increased from 1 to 2

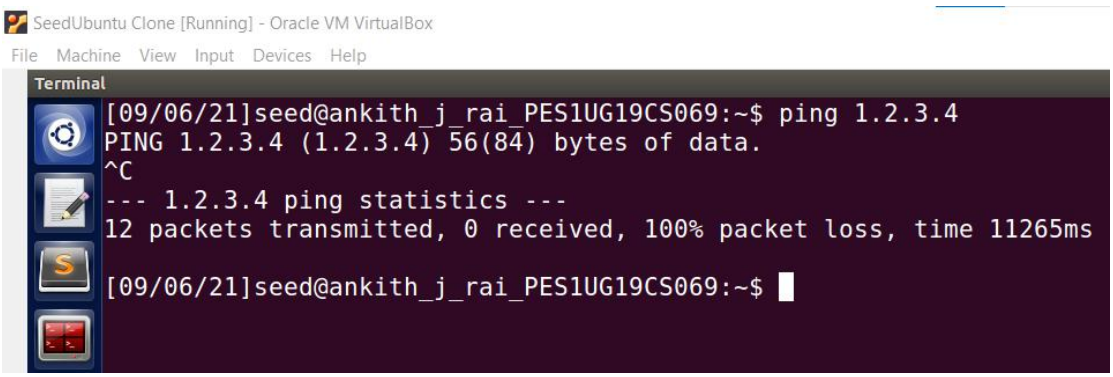
Wireshark · Packet 4 · wireshark_any_20210906144902_3gir8P

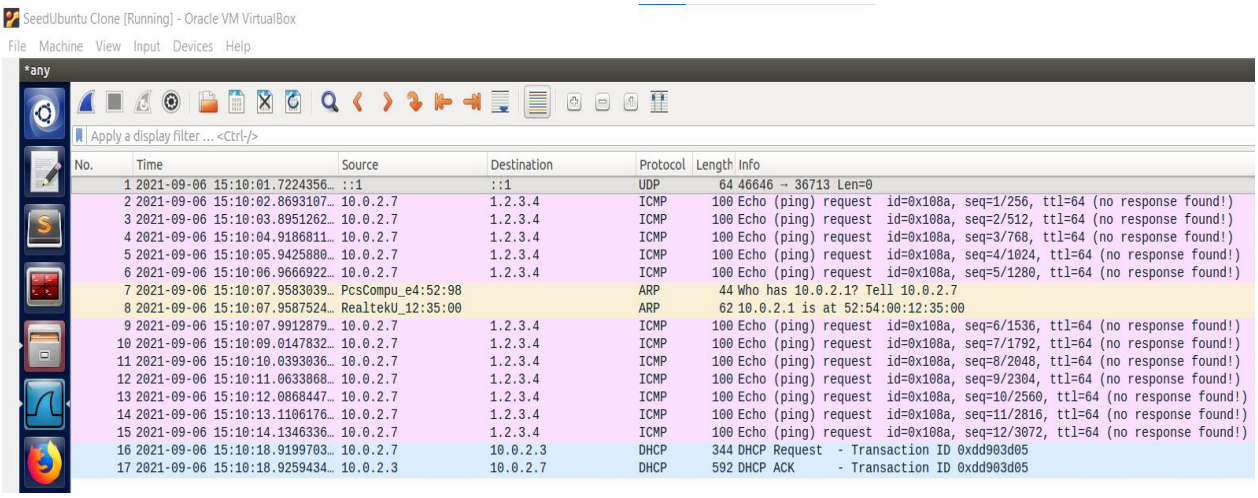
```
▼ Internet Control Message Protocol
    Type: 11 (Time-to-live exceeded)
    Code: 0 (Time to live exceeded in transit)
    Checksum: 0xf4ff [correct]
    [Checksum Status: Good]
  ▼ Internet Protocol Version 4, Src: 10.0.2.5, Dst: 182.79.154.0
      0100 .... = Version: 4
      .... 0101 = Header Length: 20 bytes (5)
    ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      Total Length: 28
      Identification: 0x0001 (1)
    ▶ Flags: 0x00
      Fragment offset: 0
    ▶ Time to live: 1
      Protocol: ICMP (1)
      Header checksum: 0x5d8c [validation disabled]
      [Header checksum status: Unverified]
      Source: 10.0.2.5
      Destination: 182.79.154.0
      [Source GeoIP: Unknown]
    ▶ [Destination GeoIP: India, 20.000000, 77.000000]
  ▼ Internet Control Message Protocol
      Type: 8 (Echo (ping) request)
      Code: 0
      Checksum: 0xf7ff [unverified] [in ICMP error packet]
      [Checksum Status: Unverified]
```



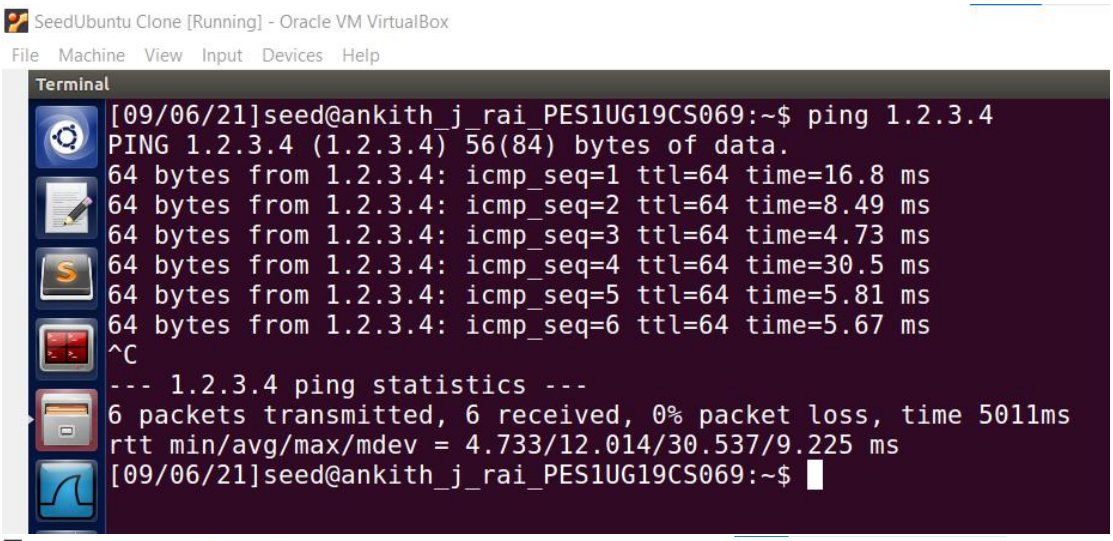
Task 4: Sniffing and-then Spoofing

Screenshot of ping 1.2.3.4

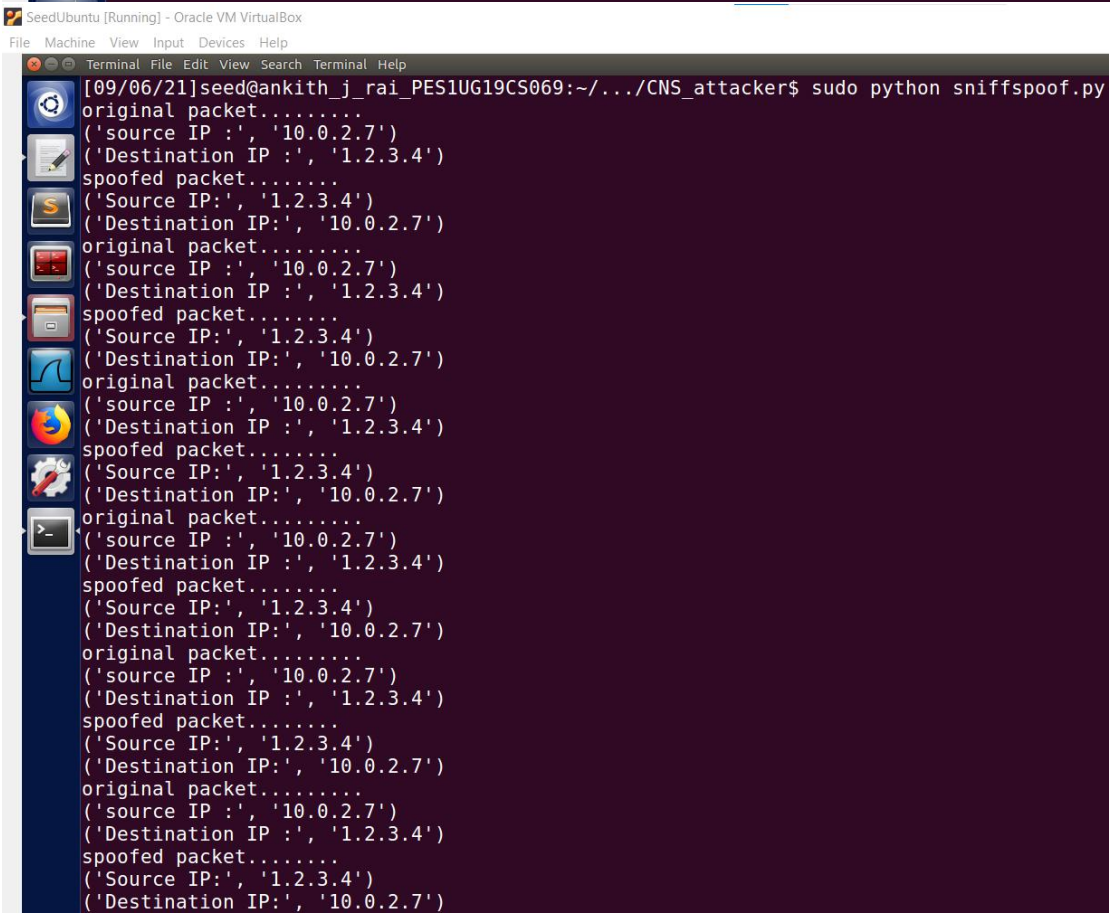




The sniffspoof.py is run on SeedUbuntu(attacker machine).The attacker machine sniffs packets coming from a machine and then the attacker machine spoofs the machine that the earlier machine(victim machine) is trying to ping.Hence making the victim machine believe that the machine which it is trying to ping is alive even though it is unreachable.



```
[09/06/21]seed@ankith_j_rai_PES1UG19CS069:~$ ping 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
64 bytes from 1.2.3.4: icmp_seq=1 ttl=64 time=16.8 ms
64 bytes from 1.2.3.4: icmp_seq=2 ttl=64 time=8.49 ms
64 bytes from 1.2.3.4: icmp_seq=3 ttl=64 time=4.73 ms
64 bytes from 1.2.3.4: icmp_seq=4 ttl=64 time=30.5 ms
64 bytes from 1.2.3.4: icmp_seq=5 ttl=64 time=5.81 ms
64 bytes from 1.2.3.4: icmp_seq=6 ttl=64 time=5.67 ms
^C
--- 1.2.3.4 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5011ms
rtt min/avg/max/mdev = 4.733/12.014/30.537/9.225 ms
[09/06/21]seed@ankith_j_rai_PES1UG19CS069:~$ 
```



```
[09/06/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_attacker$ sudo python sniffspoof.py
original packet.........
('source IP :', '10.0.2.7')
('Destination IP :', '1.2.3.4')
spoofed packet........
('Source IP:', '1.2.3.4')
('Destination IP:', '10.0.2.7')
original packet.........
('source IP :', '10.0.2.7')
('Destination IP :', '1.2.3.4')
spoofed packet........
('Source IP:', '1.2.3.4')
('Destination IP:', '10.0.2.7')
original packet.........
('source IP :', '10.0.2.7')
('Destination IP :', '1.2.3.4')
spoofed packet........
('Source IP:', '1.2.3.4')
('Destination IP:', '10.0.2.7')
original packet.........
('source IP :', '10.0.2.7')
('Destination IP :', '1.2.3.4')
spoofed packet........
('Source IP:', '1.2.3.4')
('Destination IP:', '10.0.2.7')
original packet.........
('source IP :', '10.0.2.7')
('Destination IP :', '1.2.3.4')
spoofed packet........
('Source IP:', '1.2.3.4')
('Destination IP:', '10.0.2.7')
original packet.........
('source IP :', '10.0.2.7')
('Destination IP :', '1.2.3.4')
spoofed packet........
('Source IP:', '1.2.3.4')
('Destination IP:', '10.0.2.7')
```

From the screenshots we can see that the attacker machine has successfully sniffed and spoofed the packets.