Name	Naman Choudhary
SRN	PES2UG20CS209
Section	D

Sniffing and Spoofing Lab

Assignment 1

Lab Task Set-1: Using Tools to Sniff and Spoof Packets using Scapy

Task 1.1: Sniffing Packets

The objective of this task is to learn how to use Scapy to do packet sniffing in Python programs.

Task 1.1 A: Sniff IP packets using Scapy

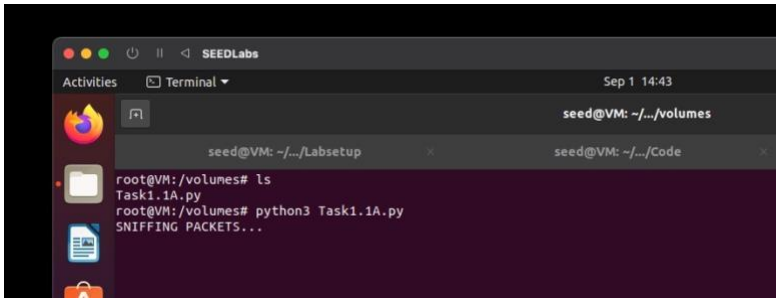
On the Attacker terminal run the command:

 # python3 Task1.1A.py

Explain on which VM you ran this command and why?

- ➔ I ran the command `**python3 Task1.1A.py**` **on the attacker's VM** terminal, as the attacker is the one to sniff the packers from the victim's VM

Provide a screenshot of your observations.



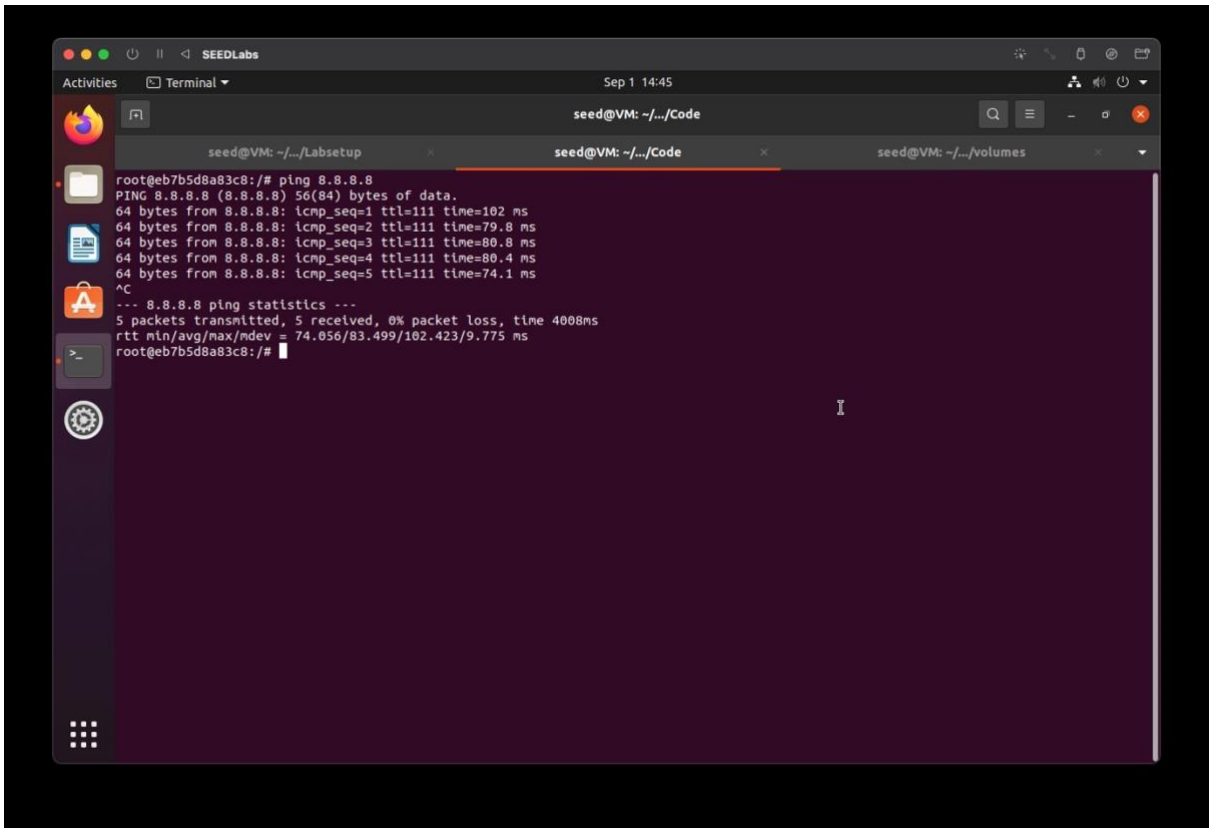
Sniffing on Attacker's VM shown above

From the host A machine's terminal ping a random IP address(8.8.8.8)

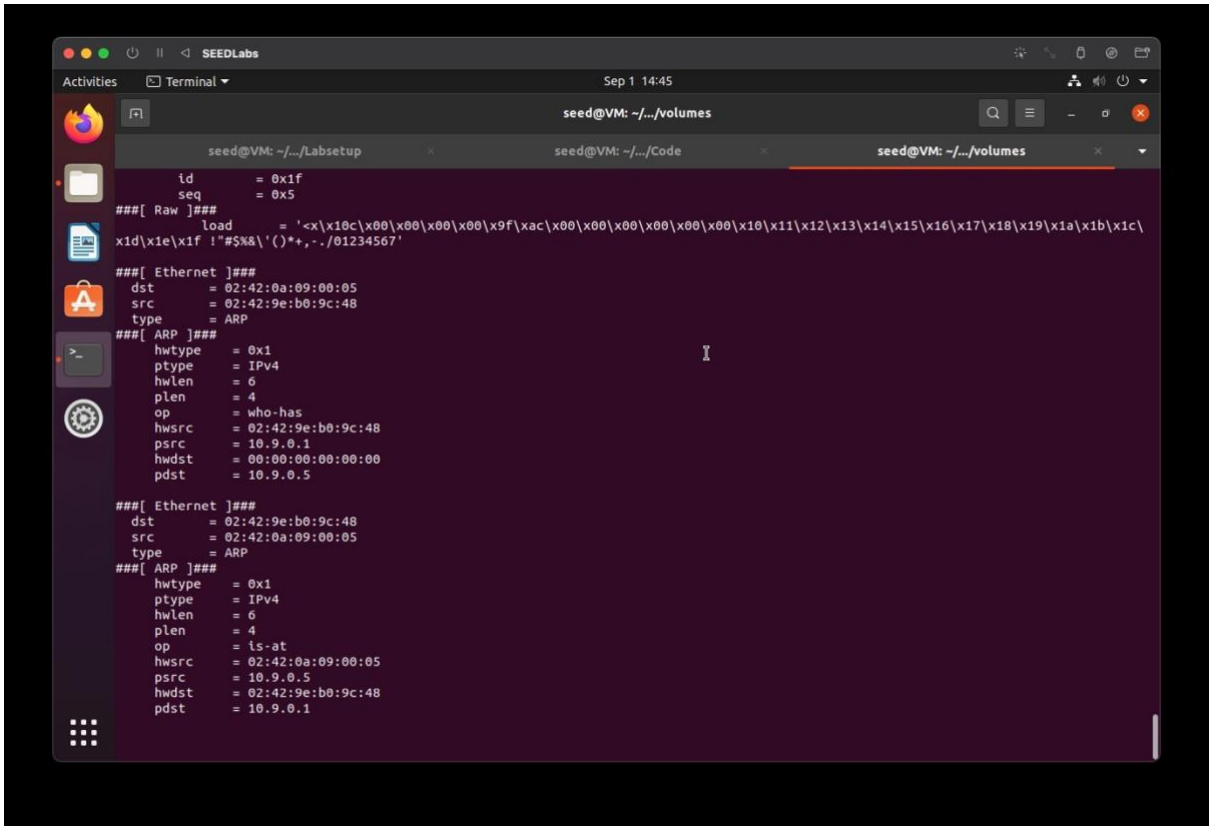
On the Host A terminal run the command:

 # ping 8.8.8.8

Host A terminal:



Attacker's terminal:



Sniffing is successful and its data is being shown on terminal continuously.

Now, we run the same program without root privileges.

On the Attacker terminal run the command:

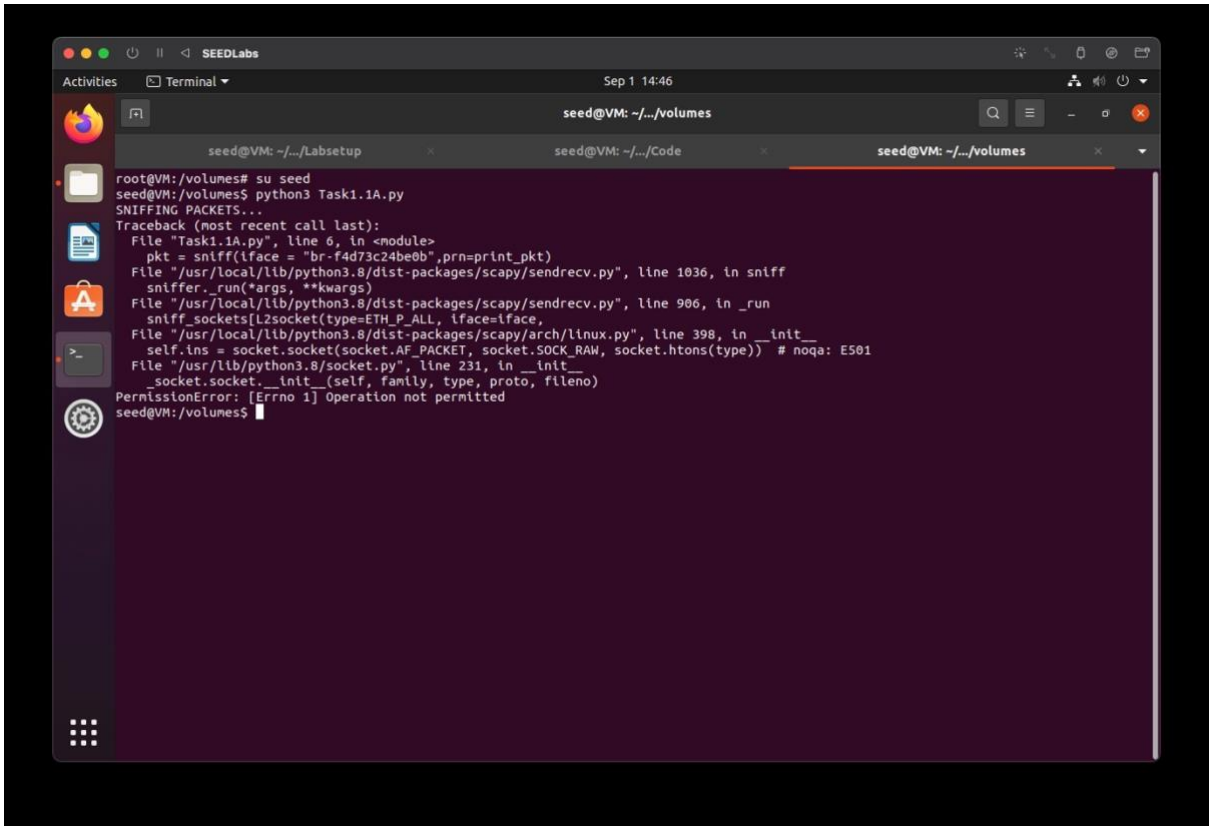
 # su seed

$ python3 Task1.1A.py

Do you find any issues? If so, why?

- ➔ Yes, there are issues when the command is run, as the Operation is not permitted without root privilages. As seed user is not the root user, it does not have the required administrative access

Provide a screenshot of your observations.



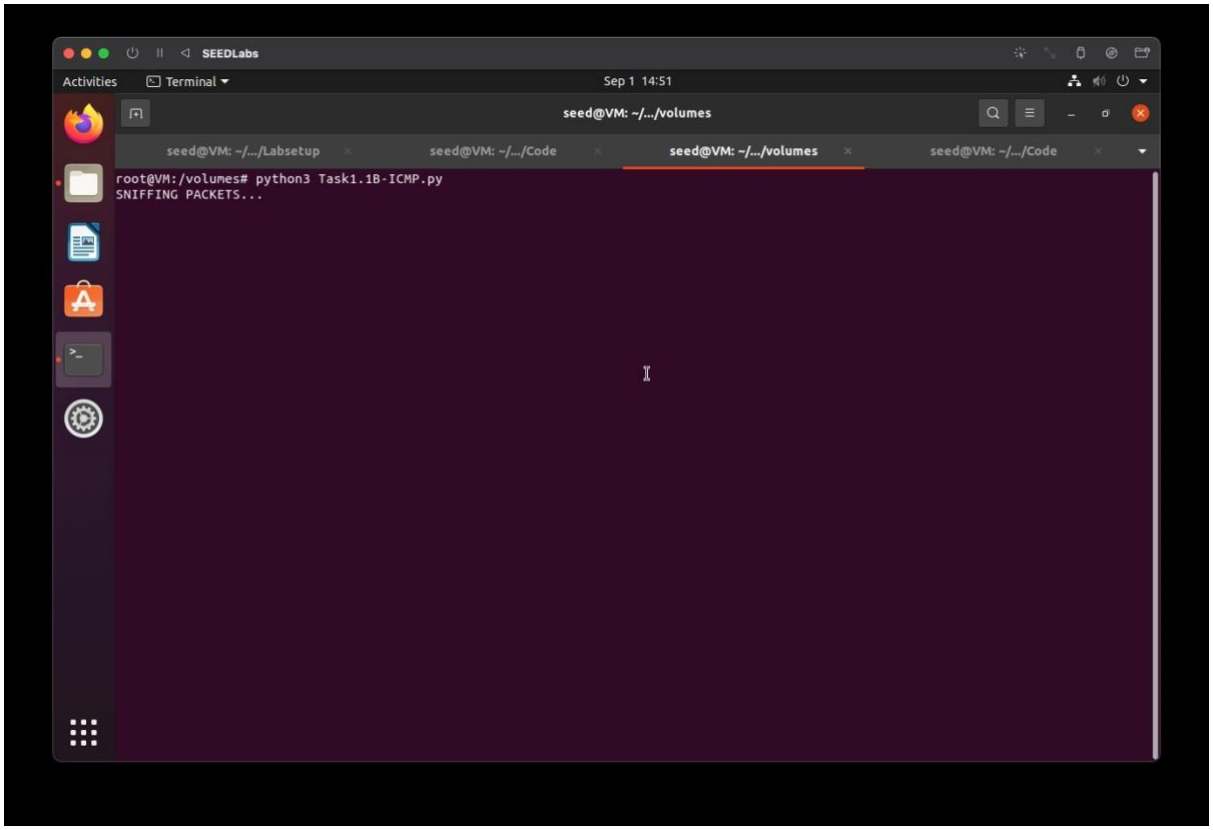
Task 1.1 B : Capturing ICMP, TCP packet and Subnet

Capture only the ICMP packet

On the Attacker terminal run the command:

 # python3 Task1.1B-ICMP.py

Provide a screenshot of your observations



The attacker starts sniffing.

From the host A machine's terminal ping a random IP address(8.8.8.8)

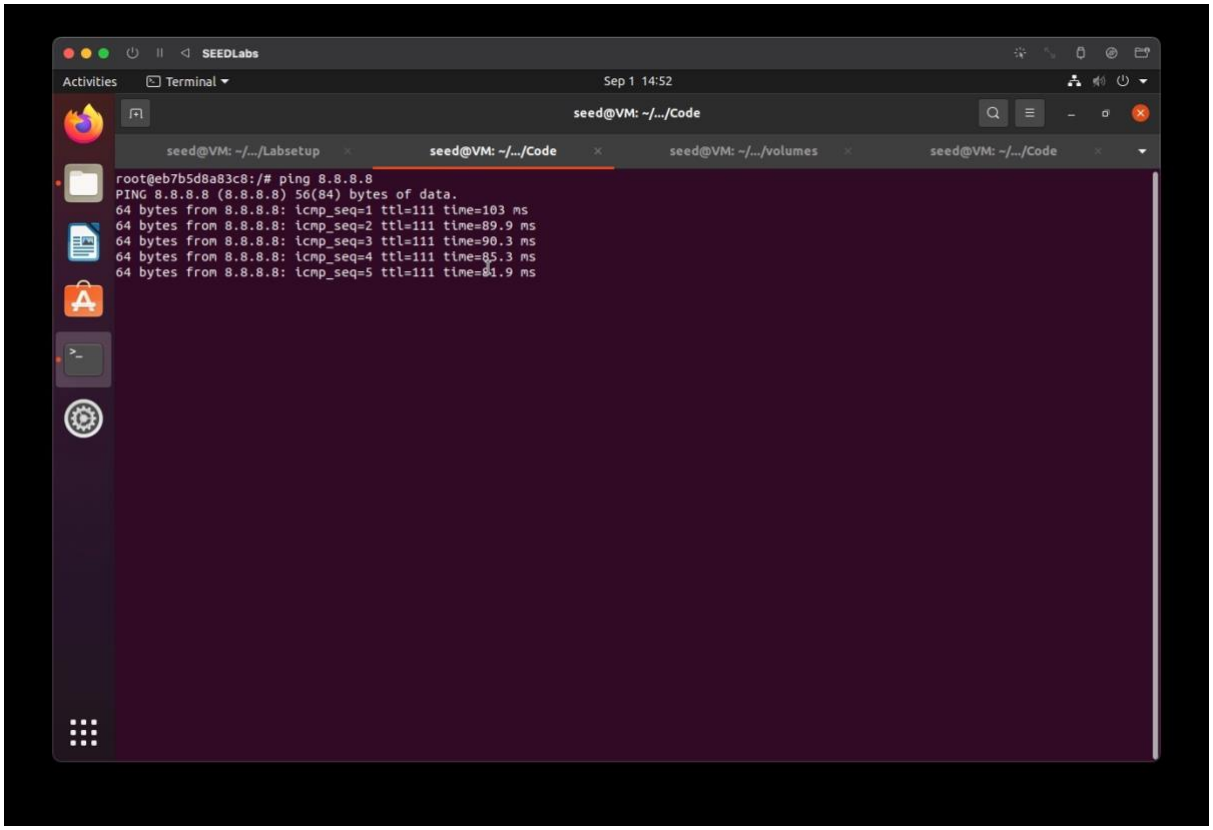
On the Host A terminal run the command:

 # ping 8.8.8.8

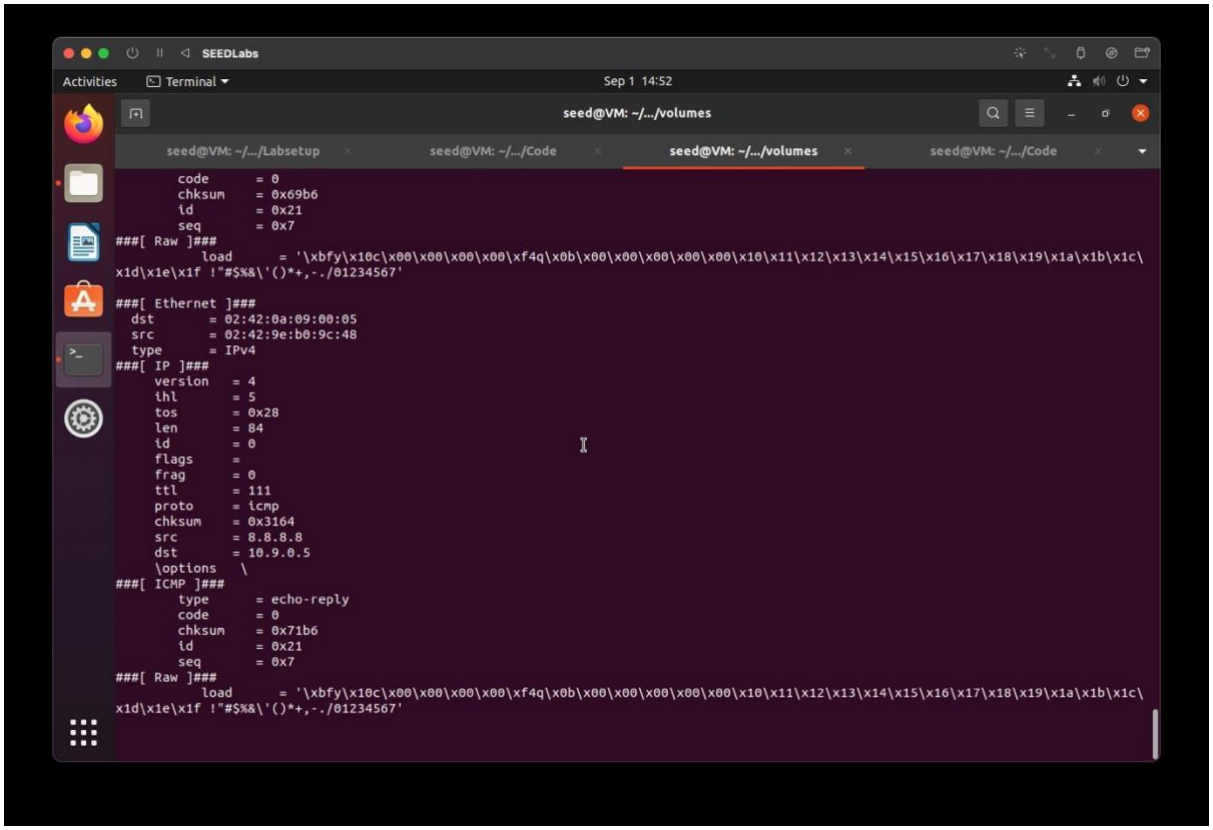
The ICMP packets are captured by the sniffer program.

Provide a screenshot of your observations.

Host A terminal:



Attacker's terminal:



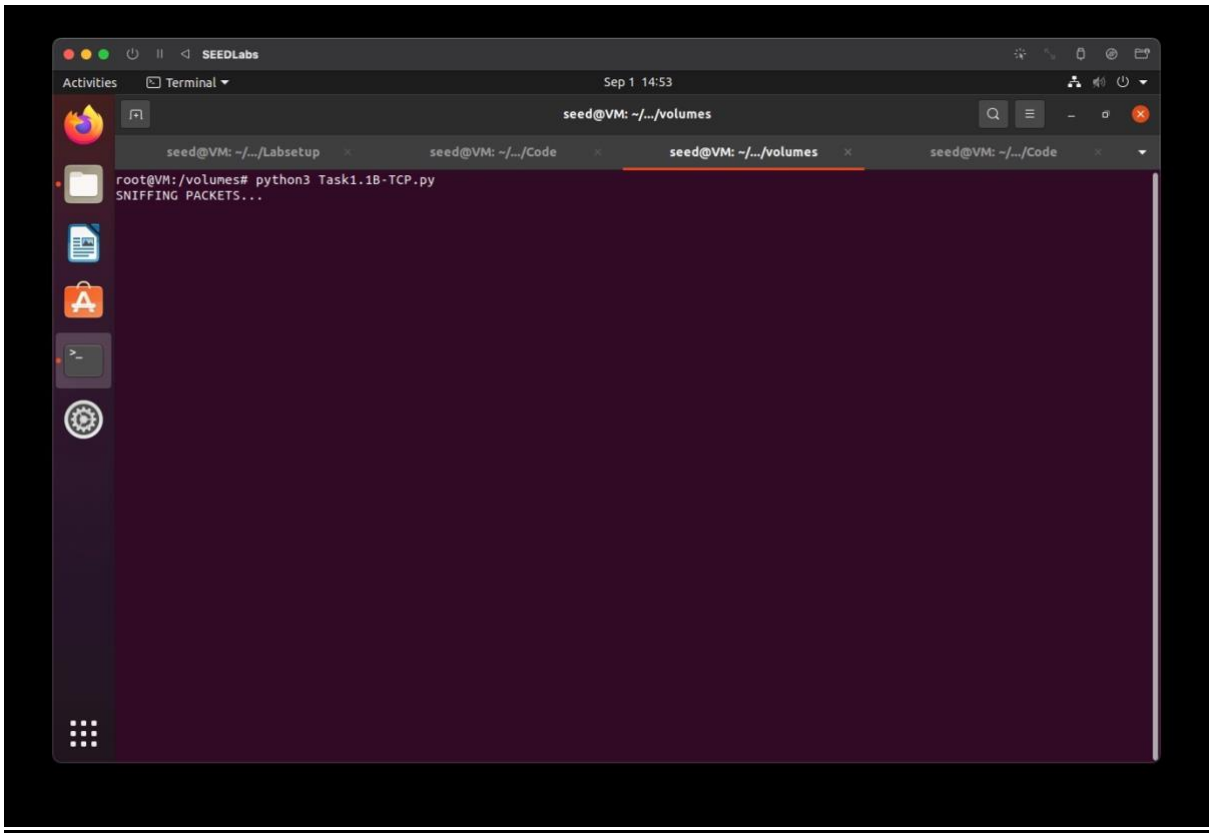
Sniffing attack is successful.

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

On the Attacker terminal run the command:

 # python3 Task1.1B-TCP.py

Provide a screenshot of your observations



The attacker starts sniffing.

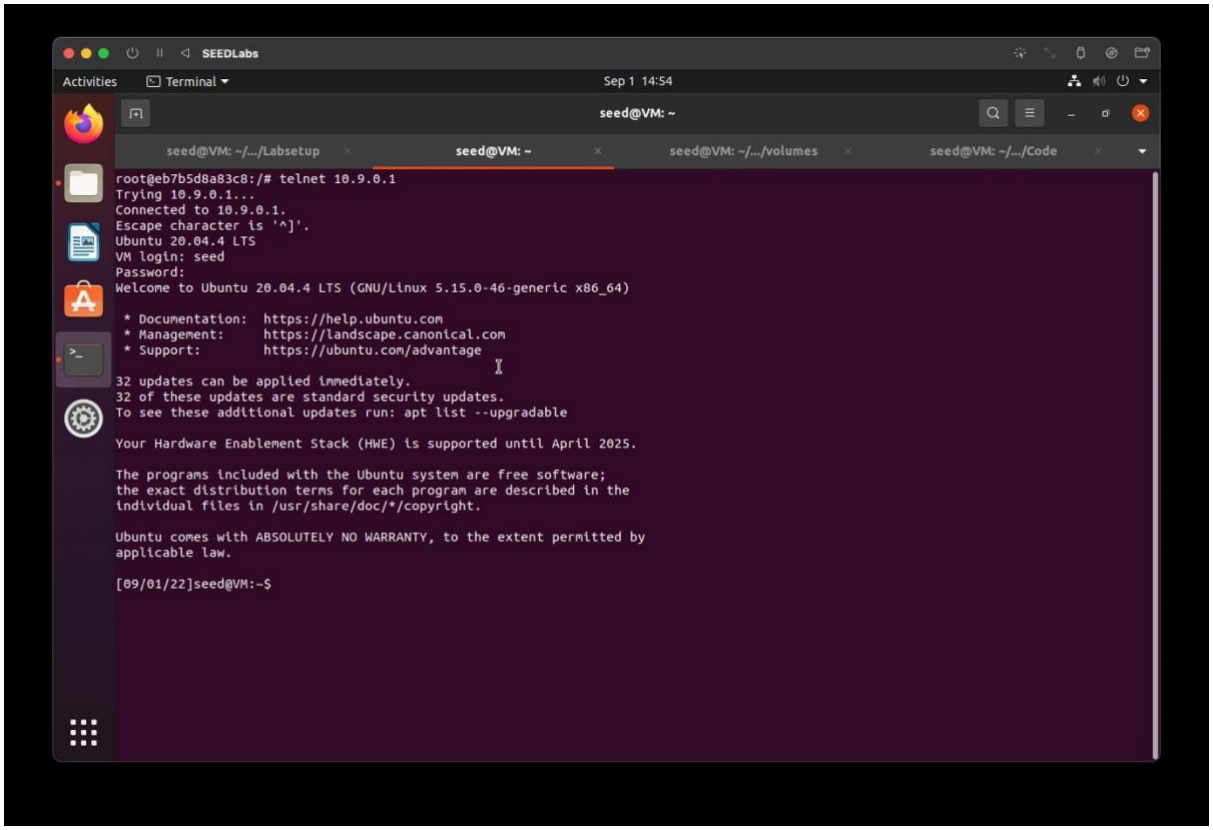
From the host A machine's terminal telnet to a random IP address.

On the Host A terminal run the command:

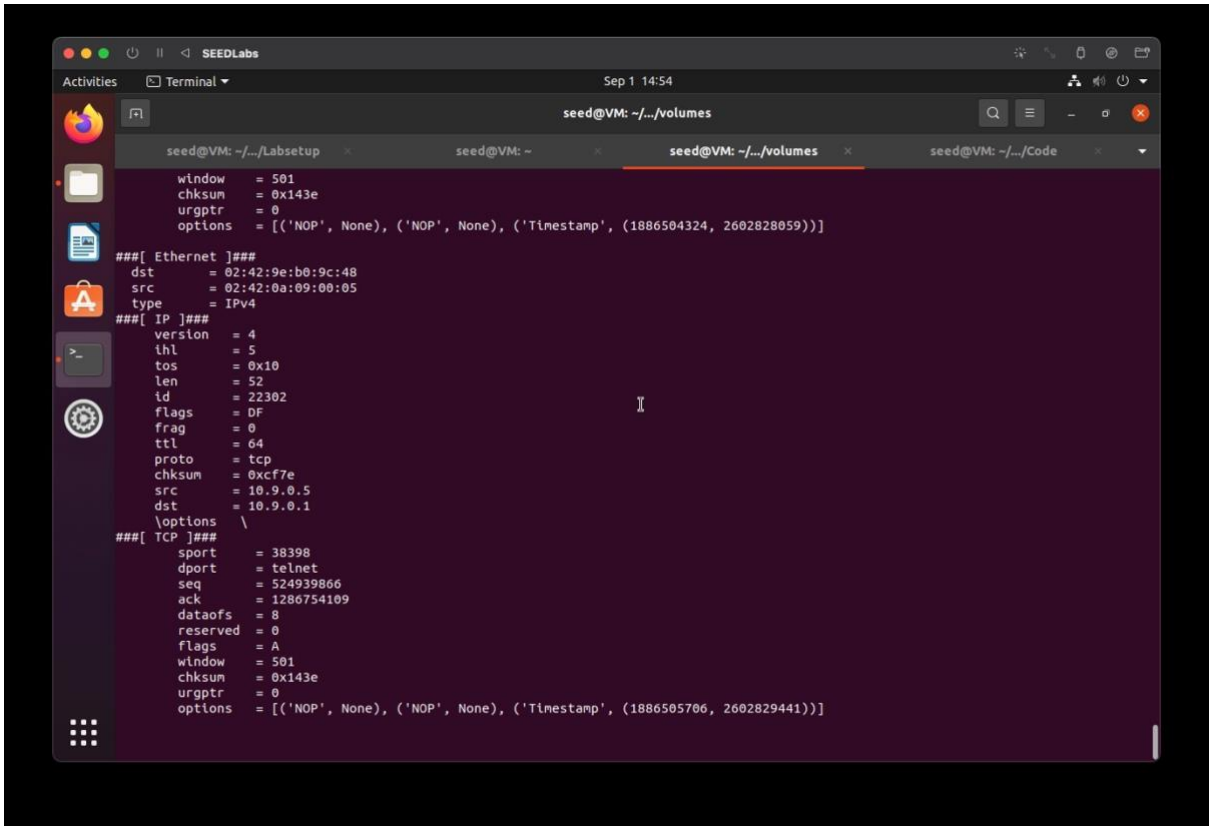
 # telnet 10.9.0.1

Provide screenshots of your observations.

Host A Terminal:



Attacker's terminal:

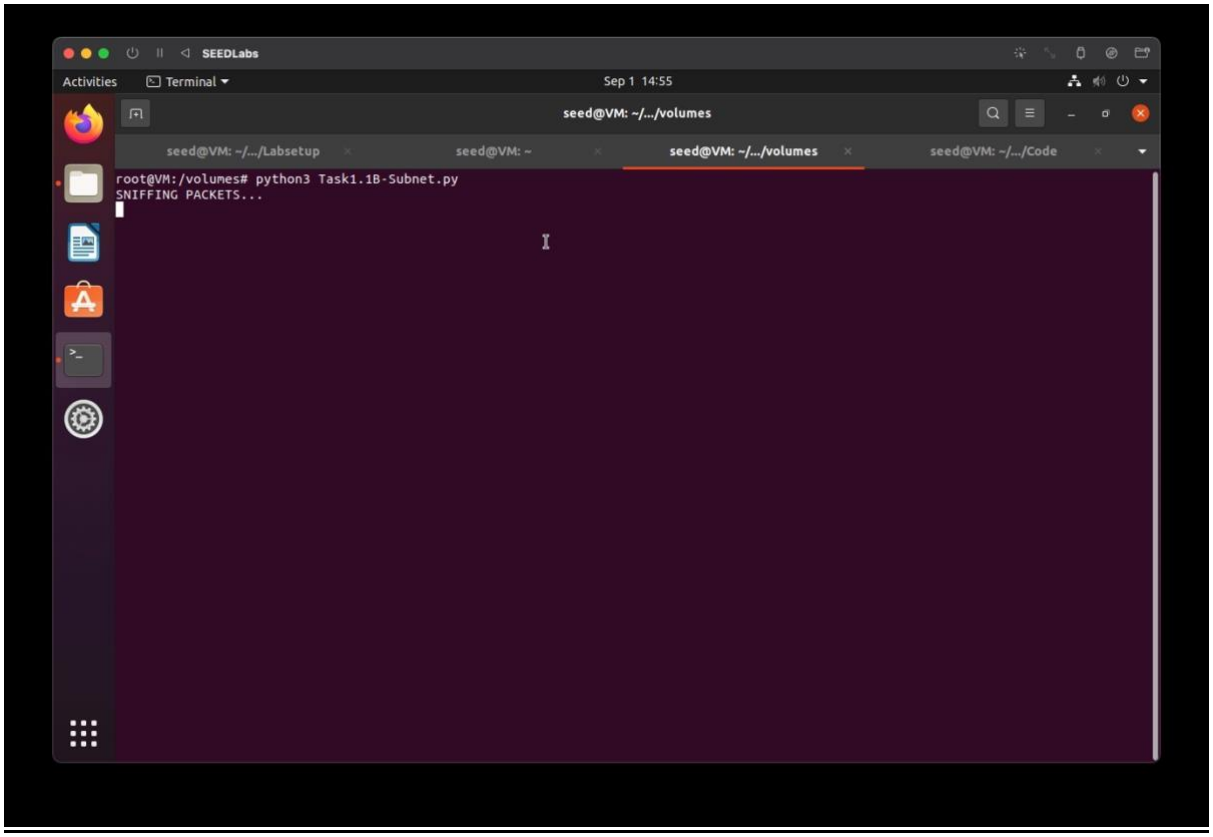


Capture packets that come from or go to a particular subnet

On the Attacker terminal run the command:

 # python3 Task1.1B-Subnet.py

Provide a screenshot of your observations

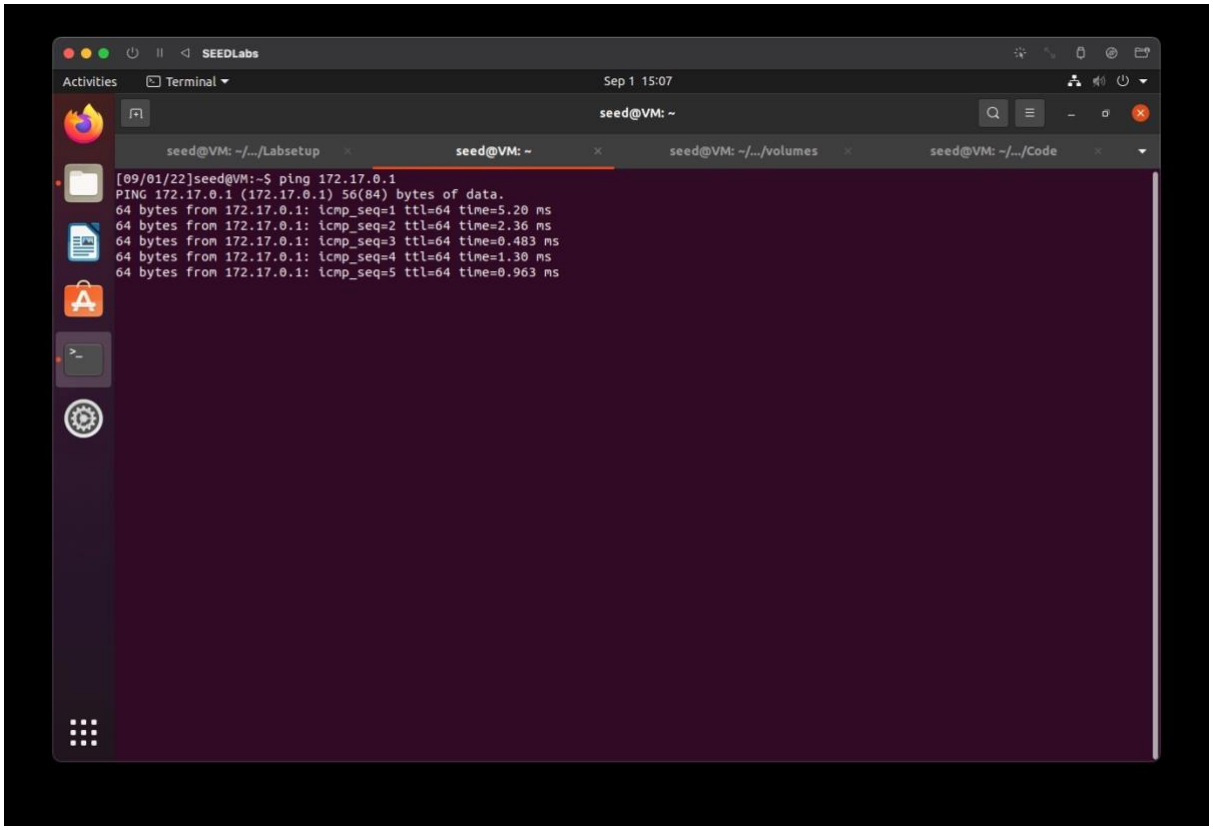


On the Host A terminal run the command

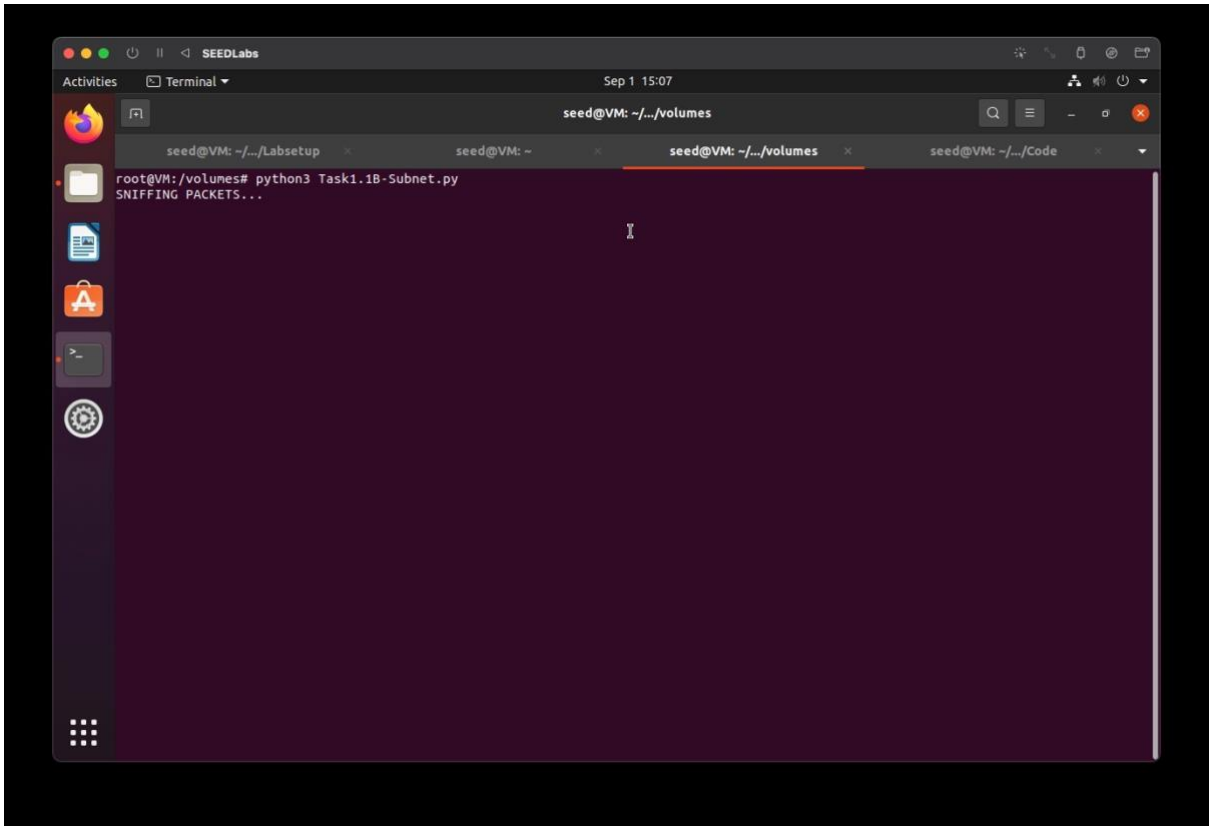
 # ping 172.17.0.1

Provide screenshots of your observations.

Host A terminal:



Attacker's terminal:



Task 1.2: Spoofing

The objective of this task is to spoof IP packets with an arbitrary source IP address.

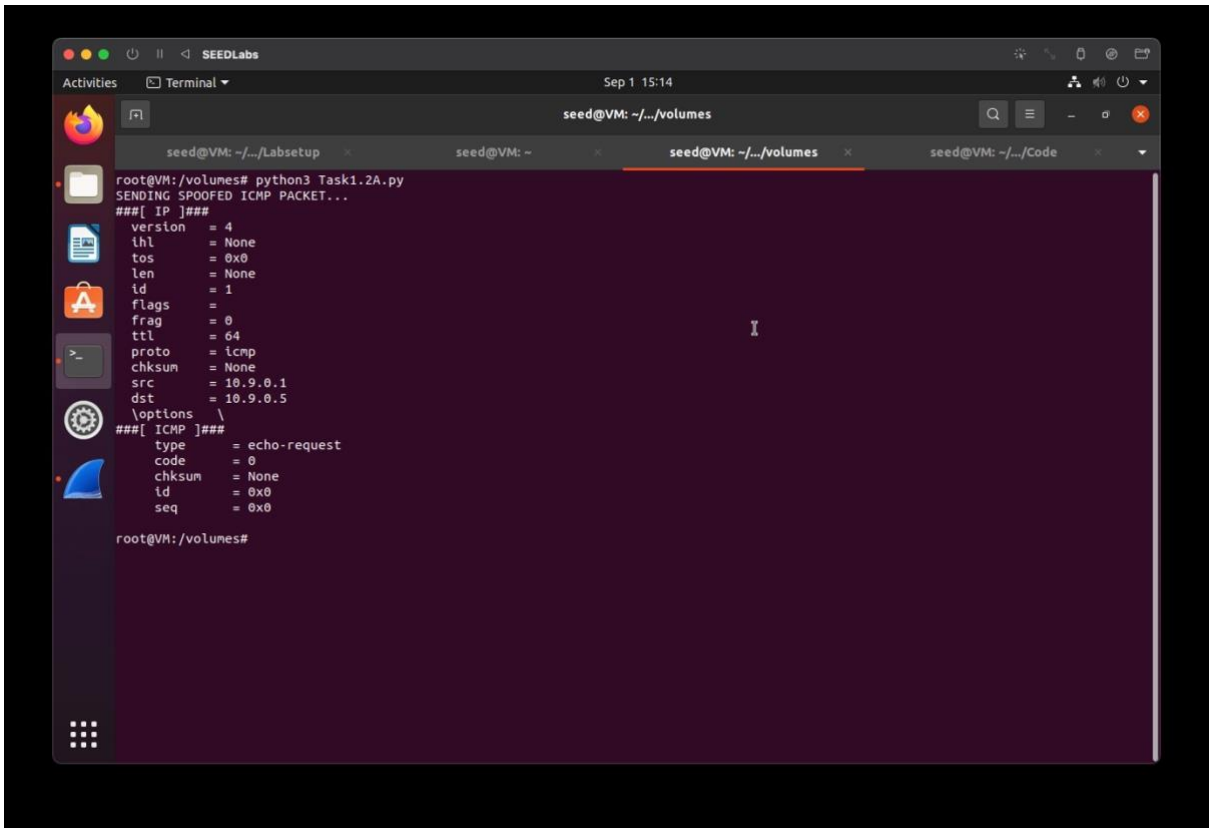
Wireshark is kept open before executing the program.

On the Attacker terminal run the command:

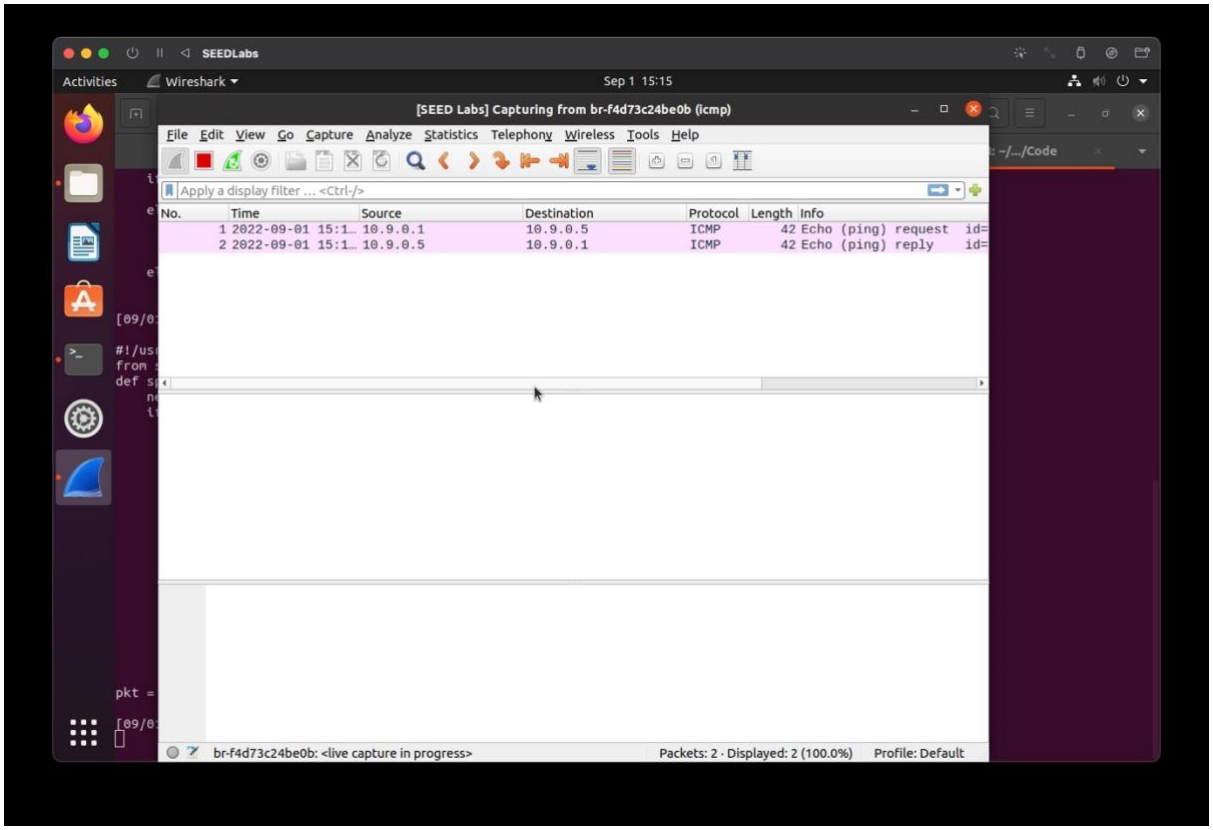
 # python3 Task1.2A.py

Provide a screenshot of your observations.

Attacker's terminal:



Wireshark:



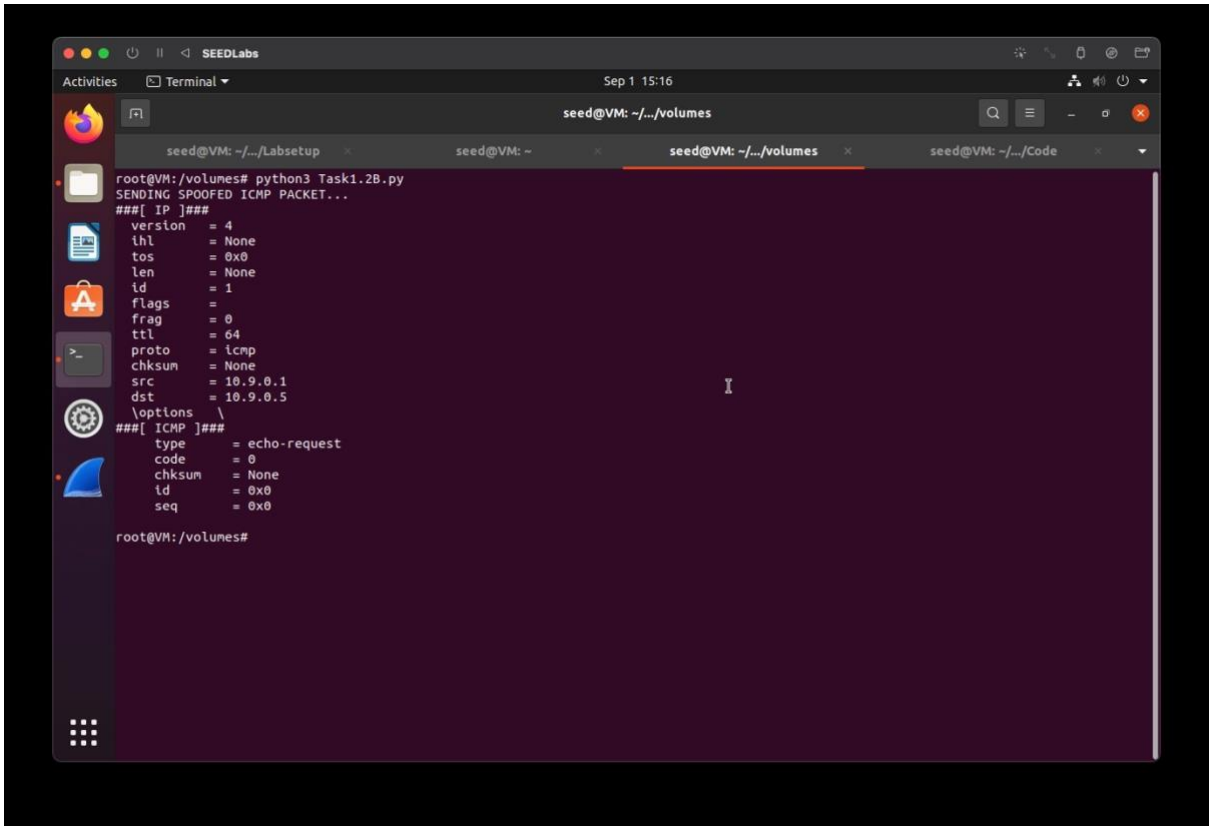
<u>With an arbitrary source IP address:</u>

On the Attacker terminal run the command:

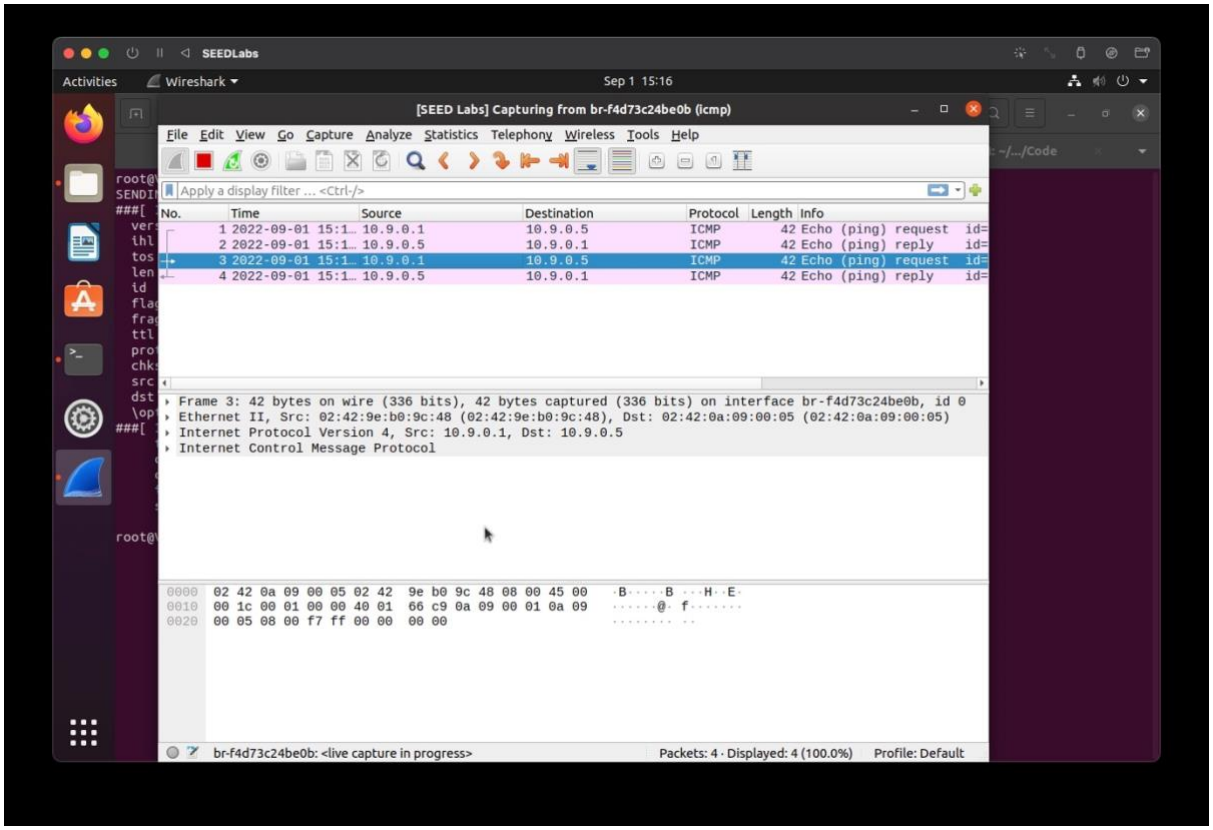
 # python3 Task1.2B.py

Provide a screenshot of your observations.

Attacker's terminal:


SEEDLabs
Activities
Terminal
Sep 1 15:16
seed@VM: ~/.../volumes
seed@VM: ~/.../Labsetup
seed@VM: ~
seed@VM: ~/.../volumes
seed@VM: ~/.../Code
root@VM:/volumes# python3 Task1.2B.py
SENDING SPOOFED ICMP PACKET...
###[IP]###
version = 4
ihl = None
tos = 0x0
len = None
id = 1
flags =
frag = 0
ttl = 64
proto = icmp
chksum = None
src = 10.9.0.1
dst = 10.9.0.5
\options \
###[ICMP]###
type = echo-request
code = 0
chksum = None
id = 0x0
seq = 0x0
root@VM:/volumes#

Wireshark:


SEEDLabs
Activities
Wireshark
Sep 1 15:16
[SEED Labs] Capturing from br-f4d73c24be0b (icmp)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
1 2022-09-01 15:1… 10.9.0.1 10.9.0.5 ICMP 42 Echo (ping) request id=
2 2022-09-01 15:1… 10.9.0.5 10.9.0.1 ICMP 42 Echo (ping) reply id=
3 2022-09-01 15:1… 10.9.0.1 10.9.0.5 ICMP 42 Echo (ping) request id=
4 2022-09-01 15:1… 10.9.0.5 10.9.0.1 ICMP 42 Echo (ping) reply id=
Frame 3: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface br-f4d73c24be0b, id 0
Ethernet II, Src: 02:42:9e:b0:9c:48 (02:42:9e:b0:9c:48), Dst: 02:42:0a:09:00:05 (02:42:0a:09:00:05)
Internet Protocol Version 4, Src: 10.9.0.1, Dst: 10.9.0.5
Internet Control Message Protocol
0000 02 42 0a 09 00 05 02 42 9e b0 9c 48 08 00 45 00
0010 00 1c 00 01 00 00 40 01 66 c9 0a 09 00 01 0a 09
0020 00 05 08 00 f7 ff 00 00 00 00
br-f4d73c24be0b: <live capture in progress>
Packets: 4 · Displayed: 4 (100.0%)
Profile: Default

When we use an arbitrary IP address, we notice in wireshark that the echo ping requests are not returned with a reply.

Task 1.3: Traceroute

The objective of this task is to implement a simple traceroute tool using Scapy to estimate the distance, in terms of number of routers, between the VM and a selected destination.

Using the scapy library, we periodically increase the ttl value to check for different readings on Wireshark.

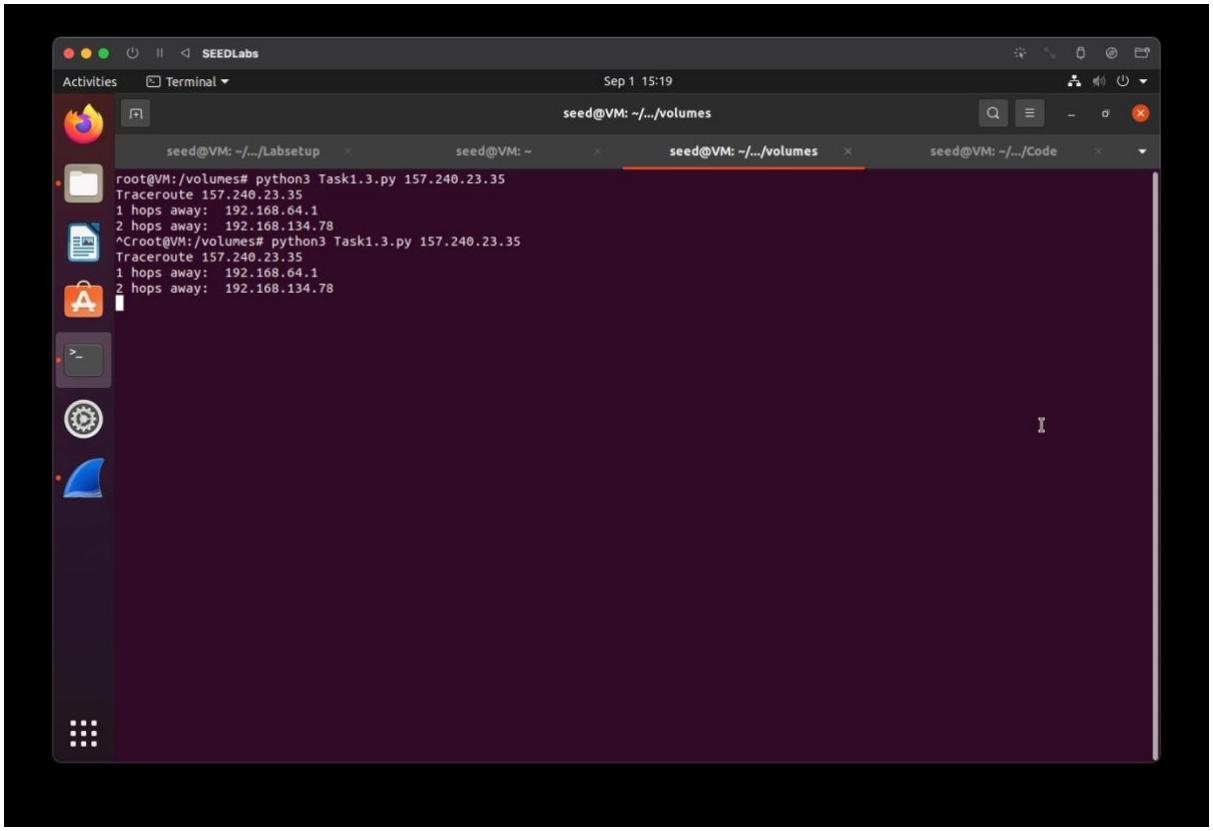
On the Attacker terminal run the command:

 # python3 Task1.3.py 157.240.23.35

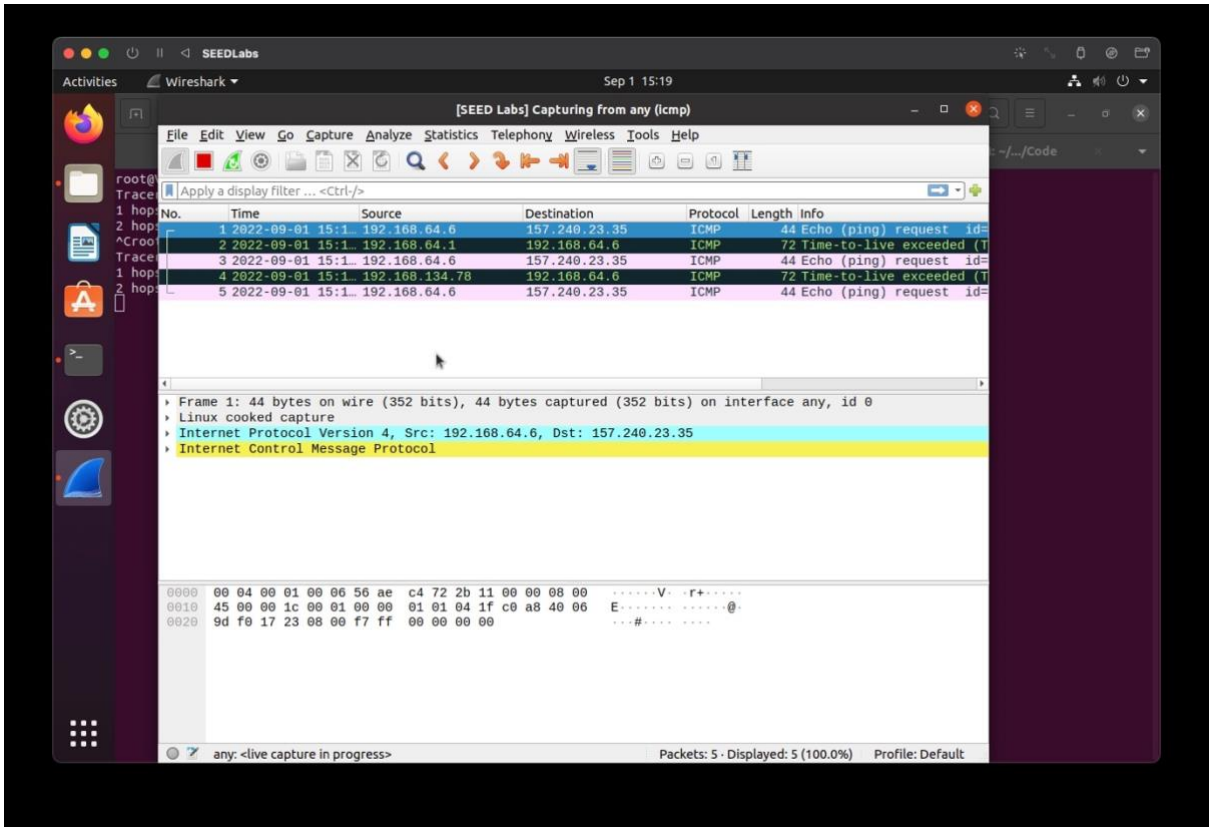
157.240.23.35 is the IP address for facebook.com

For ttl=1,

Attacker's terminal:

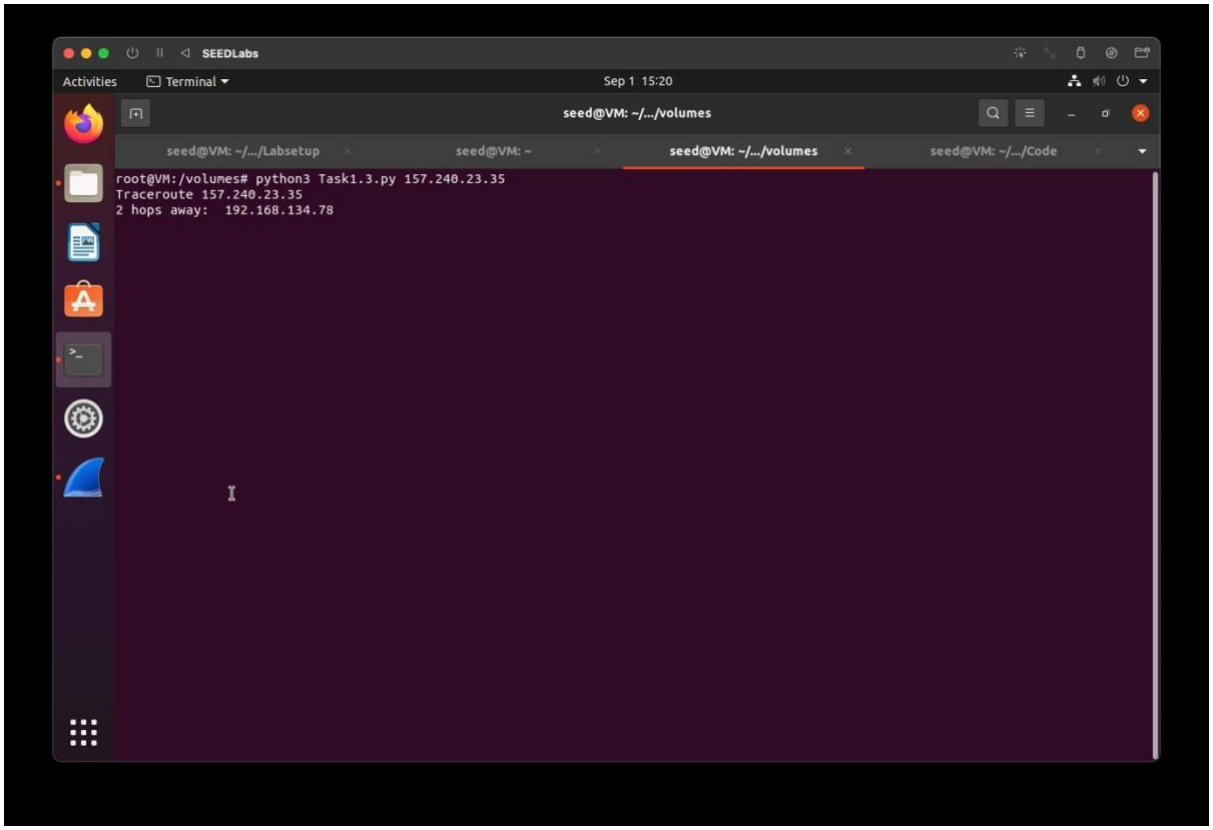


Wireshark:

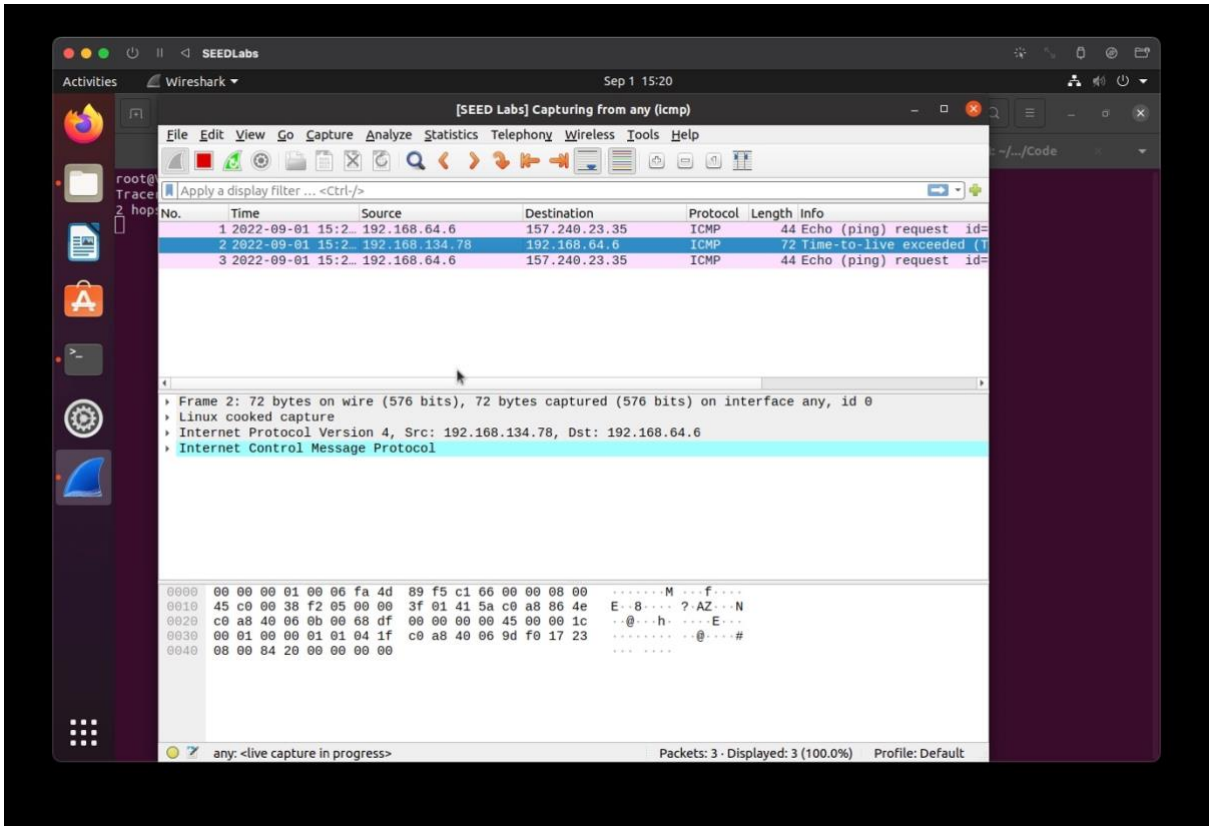


For ttl=2,

Attacker's terminal:


SEEDLabs
Activities
Terminal
Sep 1 15:20
seed@VM: ~/.../volumes
seed@VM: ~/.../Labsetup
seed@VM: ~
seed@VM: ~/.../volumes
seed@VM: ~/.../Code
root@VM:/volumes# python3 Task1.3.py 157.240.23.35
Traceroute 157.240.23.35
2 hops away: 192.168.134.78

Wireshark:


SEEDLabs
Activities
Wireshark
Sep 1 15:20
[SEED Labs] Capturing from any (icmp)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
1 2022-09-01 15:2... 192.168.64.6 157.240.23.35 ICMP 44 Echo (ping) request id=
2 2022-09-01 15:2... 192.168.134.78 192.168.64.6 ICMP 72 Time-to-live exceeded (T
3 2022-09-01 15:2... 192.168.64.6 157.240.23.35 ICMP 44 Echo (ping) request id=
Frame 2: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface any, id 0
Linux cooked capture
Internet Protocol Version 4, Src: 192.168.134.78, Dst: 192.168.64.6
Internet Control Message Protocol
0000 00 00 00 01 00 06 fa 4d 89 f5 c1 66 00 00 08 00
0010 45 c0 00 38 f2 05 00 00 3f 01 41 5a c0 a8 86 4e
0020 c0 a8 40 06 0b 00 68 df 00 00 00 00 45 00 00 1c
0030 00 01 00 00 01 01 04 1f c0 a8 40 06 9d f0 17 23
0040 08 00 84 20 00 00 00 00
any: <live capture in progress>
Packets: 3 · Displayed: 3 (100.0%)
Profile: Default

Task 1.4: Sniffing and-then Spoofing

In this task, the victim machine pings a non-existing IP address “1.2.3.4”. As the attacker machine is on the same network, it sniffs the request packet, creates a new echo reply packet with IP and ICMP header and sends it to the victim machine. Hence, the user will always receive an echo reply from a non-existing IP address indicating that the machine is alive.

On the Attacker terminal run the command:

 # python3 Task1.4.py

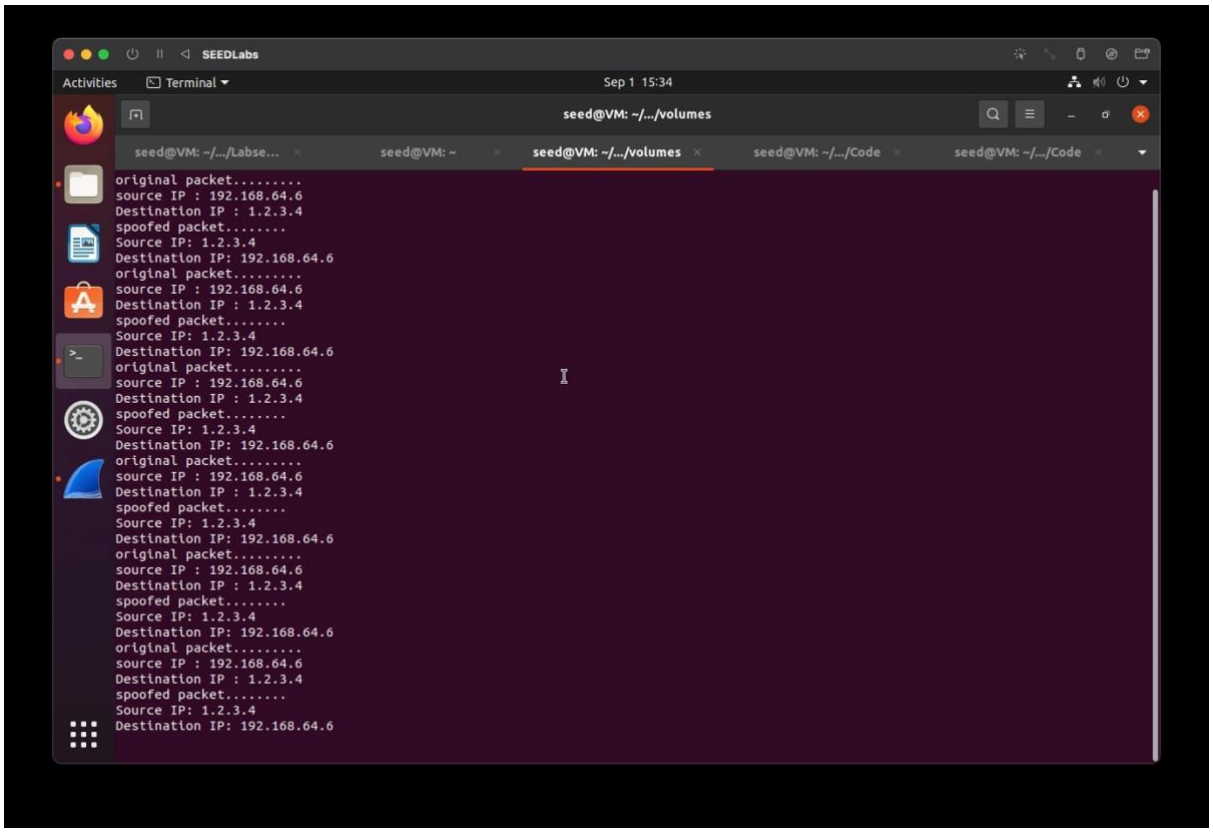
From the host A machine’s terminal ping 1.2.3.4

On the Host A terminal run the command:

 # ping 1.2.3.4

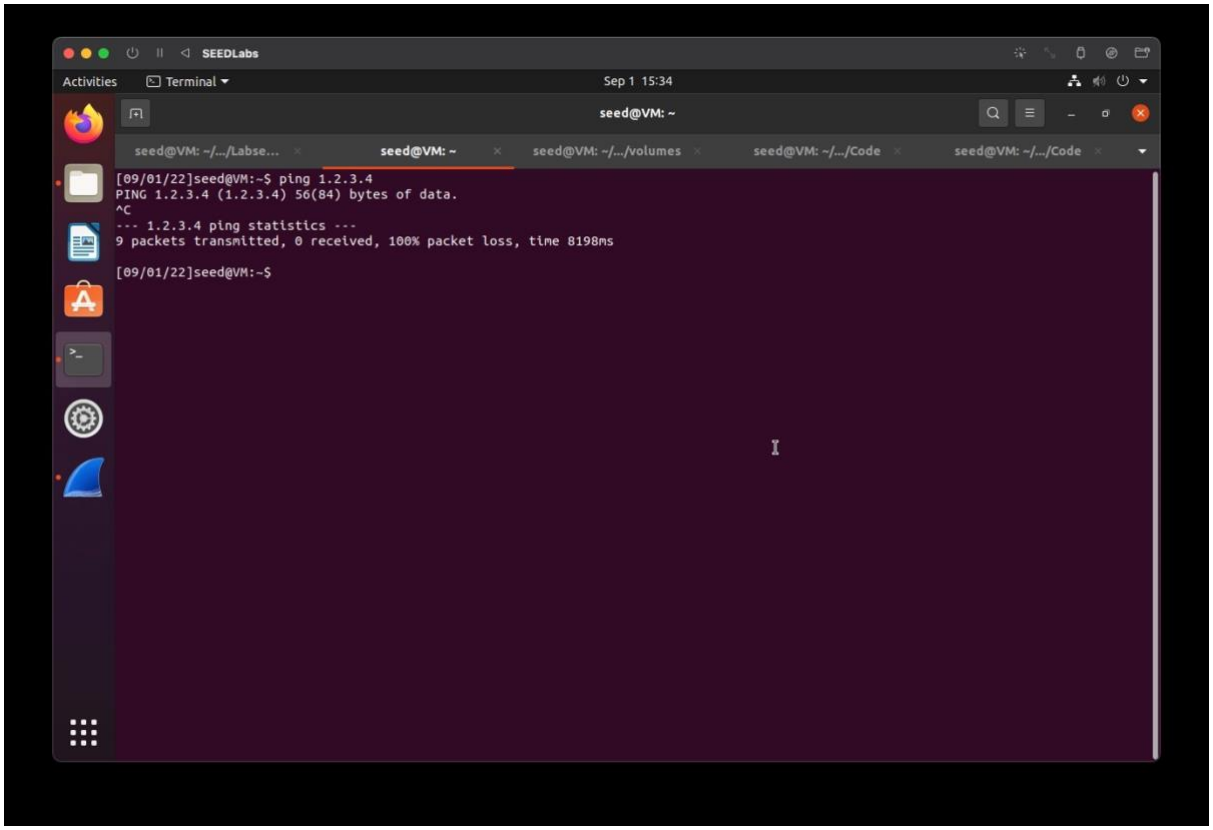
Provide a screenshot of your observations.

Attacker’s terminal:

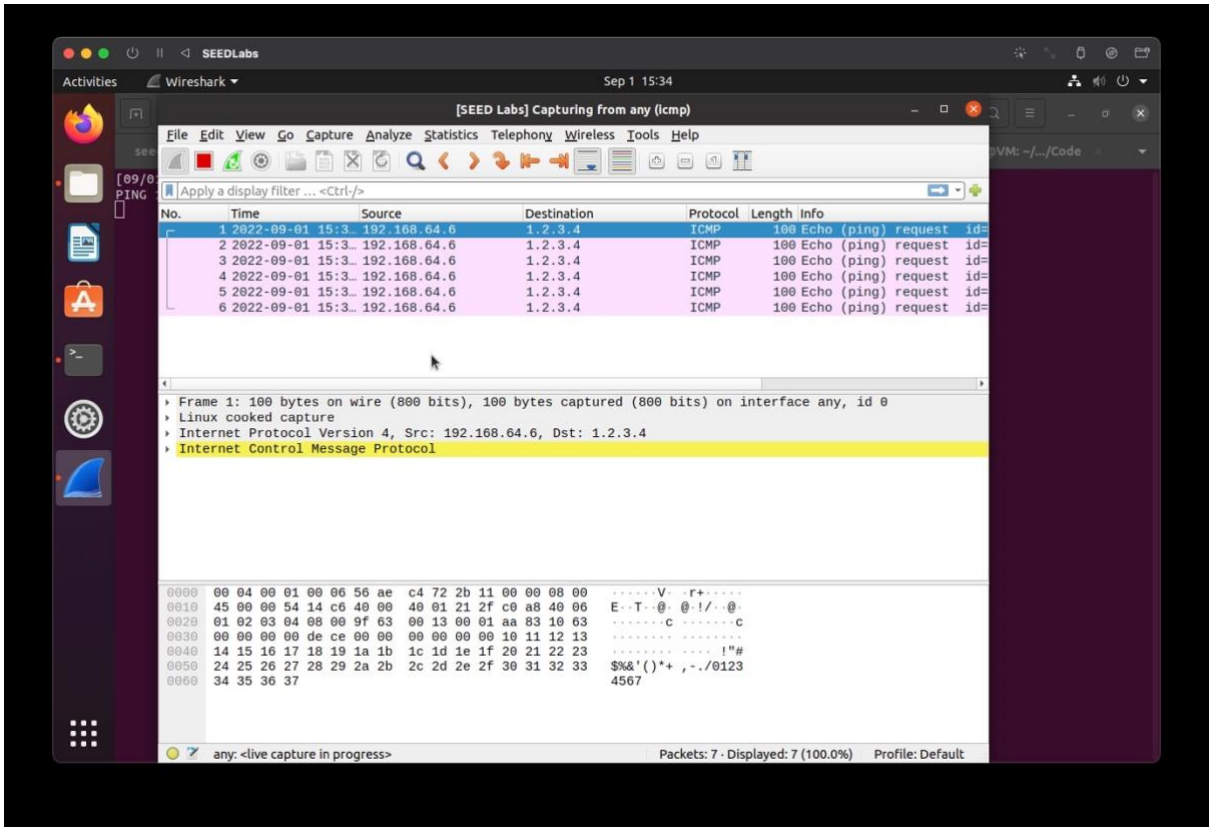


```
original packet.........
source IP : 192.168.64.6
Destination IP : 1.2.3.4
spoofed packet........
Source IP: 1.2.3.4
Destination IP: 192.168.64.6
original packet.........
source IP : 192.168.64.6
Destination IP : 1.2.3.4
spoofed packet........
Source IP: 1.2.3.4
Destination IP: 192.168.64.6
original packet.........
source IP : 192.168.64.6
Destination IP : 1.2.3.4
spoofed packet........
Source IP: 1.2.3.4
Destination IP: 192.168.64.6
original packet.........
source IP : 192.168.64.6
Destination IP : 1.2.3.4
spoofed packet........
Source IP: 1.2.3.4
Destination IP: 192.168.64.6
original packet.........
source IP : 192.168.64.6
Destination IP : 1.2.3.4
spoofed packet........
Source IP: 1.2.3.4
Destination IP: 192.168.64.6
original packet.........
source IP : 192.168.64.6
Destination IP : 1.2.3.4
spoofed packet........
Source IP: 1.2.3.4
Destination IP: 192.168.64.6
```

Host A terminal:


SEEDLabs
Activities
Terminal
Sep 1 15:34
seed@VM: ~
seed@VM: ~/.../Labse...
seed@VM: ~
seed@VM: ~/.../volumes
seed@VM: ~/.../Code
seed@VM: ~/.../Code
[09/01/22]seed@VM:~$ ping 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
^C
--- 1.2.3.4 ping statistics ---
9 packets transmitted, 0 received, 100% packet loss, time 8198ms
[09/01/22]seed@VM:~$

Wireshark:


SEEDLabs
Activities
Wireshark
Sep 1 15:34
[SEED Labs] Capturing from any (icmp)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
1 2022-09-01 15:3... 192.168.64.6 1.2.3.4 ICMP 100 Echo (ping) request id=
2 2022-09-01 15:3... 192.168.64.6 1.2.3.4 ICMP 100 Echo (ping) request id=
3 2022-09-01 15:3... 192.168.64.6 1.2.3.4 ICMP 100 Echo (ping) request id=
4 2022-09-01 15:3... 192.168.64.6 1.2.3.4 ICMP 100 Echo (ping) request id=
5 2022-09-01 15:3... 192.168.64.6 1.2.3.4 ICMP 100 Echo (ping) request id=
6 2022-09-01 15:3... 192.168.64.6 1.2.3.4 ICMP 100 Echo (ping) request id=
Frame 1: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface any, id 0
Linux cooked capture
Internet Protocol Version 4, Src: 192.168.64.6, Dst: 1.2.3.4
Internet Control Message Protocol
0000 00 04 00 01 00 06 56 ae c4 72 2b 11 00 00 08 00
0010 45 00 00 54 14 c6 40 00 40 01 21 2f c0 a8 40 06
0020 01 02 03 04 08 00 9f 63 00 13 00 01 aa 83 10 63
0030 00 00 00 00 de ce 00 00 00 00 00 00 10 11 12 13
0040 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23
0050 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33
0060 34 35 36 37
any: <live capture in progress>
Packets: 7 · Displayed: 7 (100.0%)
Profile: Default