NAME: YASH SNEHAL SHETIYA

SUID: 9276568741

LAB REPORT : PACKET SNIFFING AND SPOOFING LAB

Task 1.1: Sniffing Packets

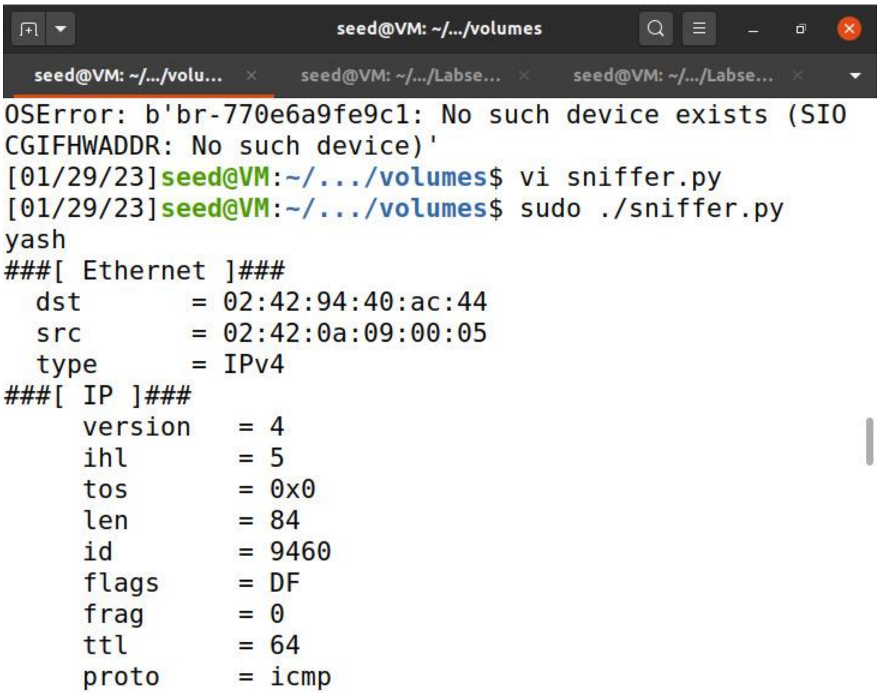
We define a function named print_pkt which will display the packet data and is passed as an argument to /sniff/ method. The sniff method is defined inside the scapy library and it takes in an interface id and further we can provide filters for our own purpose.

For the interface id we run the command ifconfig in another tab and get the required information, we can specify multiple id’s to the sniff method.

Make the python code executable and run it and it will start sniffing packets, we had to make sure to run python using the root privilege because without root privilege we won’t be able to complete our lab.

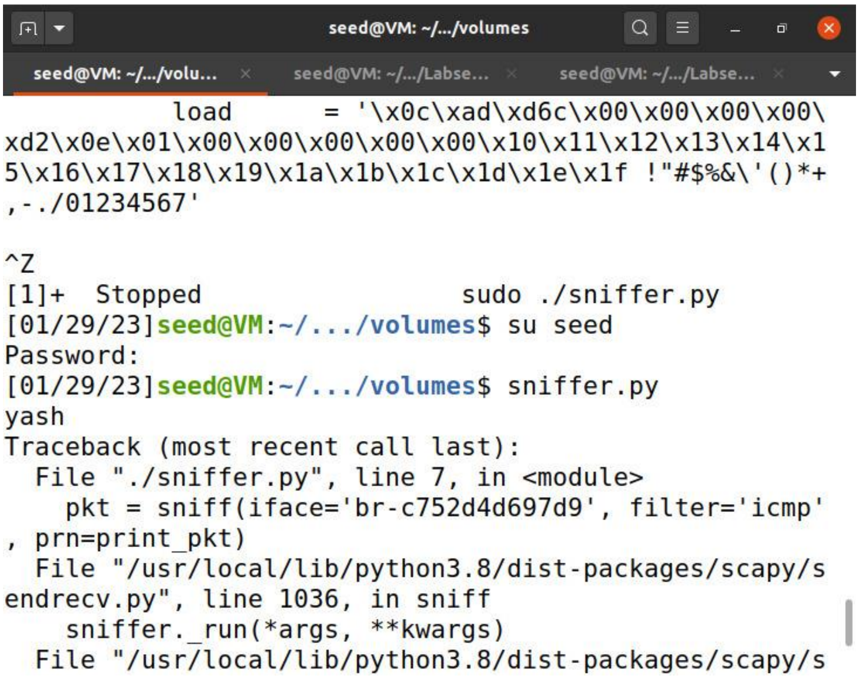
Start a shell on the host A container, we can get the container id by executing the ‘dockps’ command and then using that id start a shell on the particular container by executing docksh. We will have the host A terminal and from here start pinging.

Once the pinging starts we can observe that we can observe the packets being captured.



```
OSError: b'br-770e6a9fe9c1: No such device exists (SIO
CGIFHWADDR: No such device)'
[01/29/23]seed@VM:~/.../volumes$ vi sniffer.py
[01/29/23]seed@VM:~/.../volumes$ sudo ./sniffer.py
yash
###[ Ethernet ]###
  dst       = 02:42:94:40:ac:44
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 9460
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = icmp
```

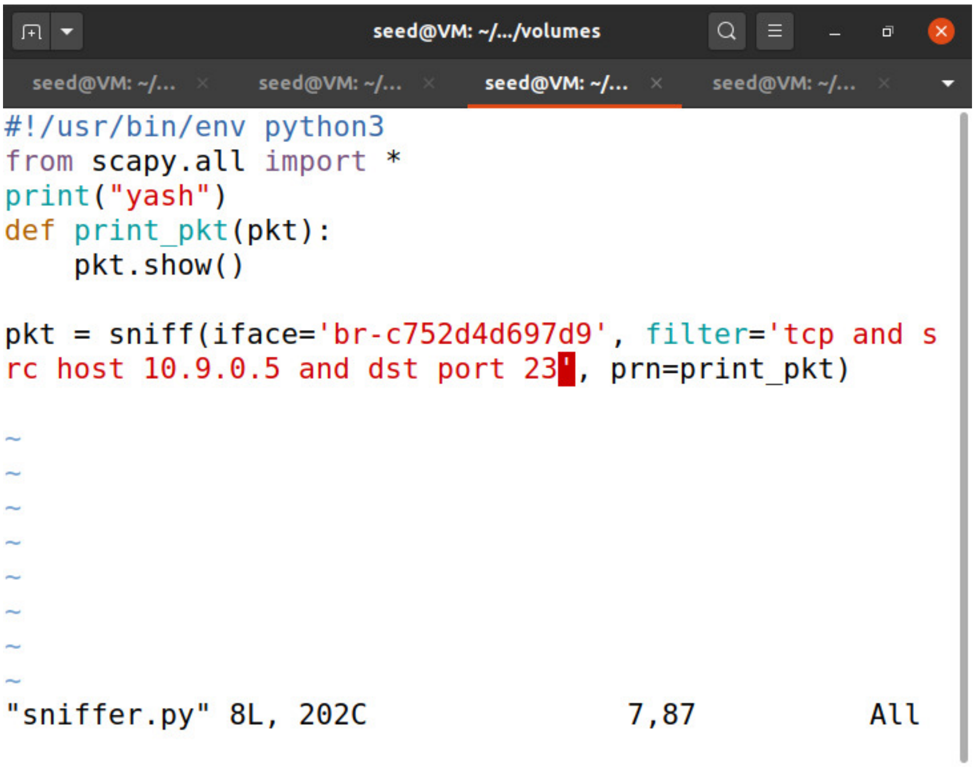
If we don’t use the root privilege and run it we will get the following error message:



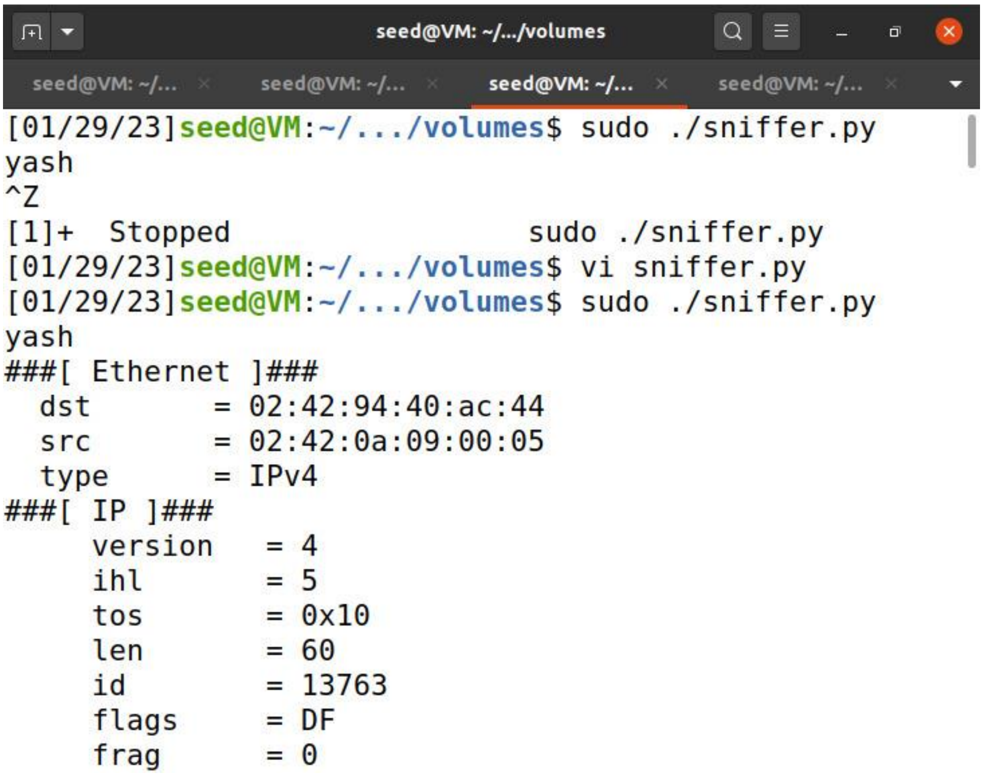
```
           load      = '\x0c\xad\xd6c\x00\x00\x00\x00\
xd2\x0e\x01\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x1
5\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !"#$%&\'()*+
,-./01234567'

^Z
[1]+  Stopped                 sudo ./sniffer.py
[01/29/23]seed@VM:~/.../volumes$ su seed
Password:
[01/29/23]seed@VM:~/.../volumes$ sniffer.py
yash
Traceback (most recent call last):
  File "./sniffer.py", line 7, in <module>
    pkt = sniff(iface='br-c752d4d697d9', filter='icmp'
, prn=print_pkt)
  File "/usr/local/lib/python3.8/dist-packages/scapy/s
endrecv.py", line 1036, in sniff
    sniffer._run(*args, **kwargs)
  File "/usr/local/lib/python3.8/dist-packages/scapy/s
```

Task 1.1 B

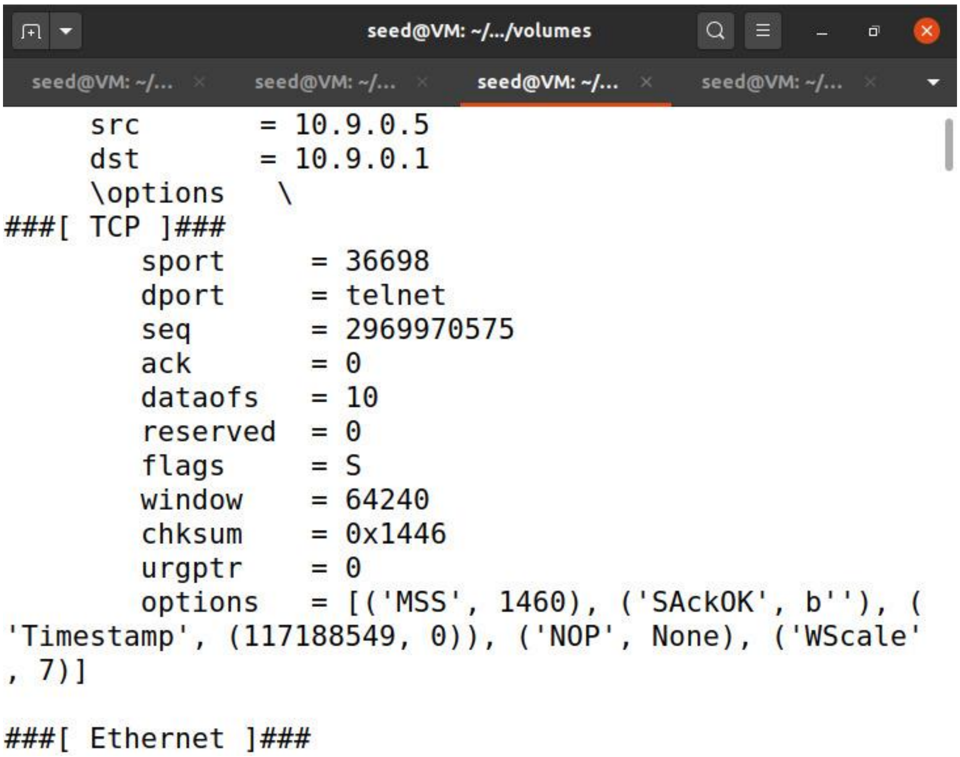
Sniffing TCP packets. For sniffing tcp packets we had to use telnet.



```
#!/usr/bin/env python3
from scapy.all import *
print("yash")
def print_pkt(pkt):
    pkt.show()

pkt = sniff(iface='br-c752d4d697d9', filter='tcp and s
rc host 10.9.0.5 and dst port 23', prn=print_pkt)

~
~
~
~
~
~
~
~
"sniffer.py" 8L, 202C                 7,87          All
```

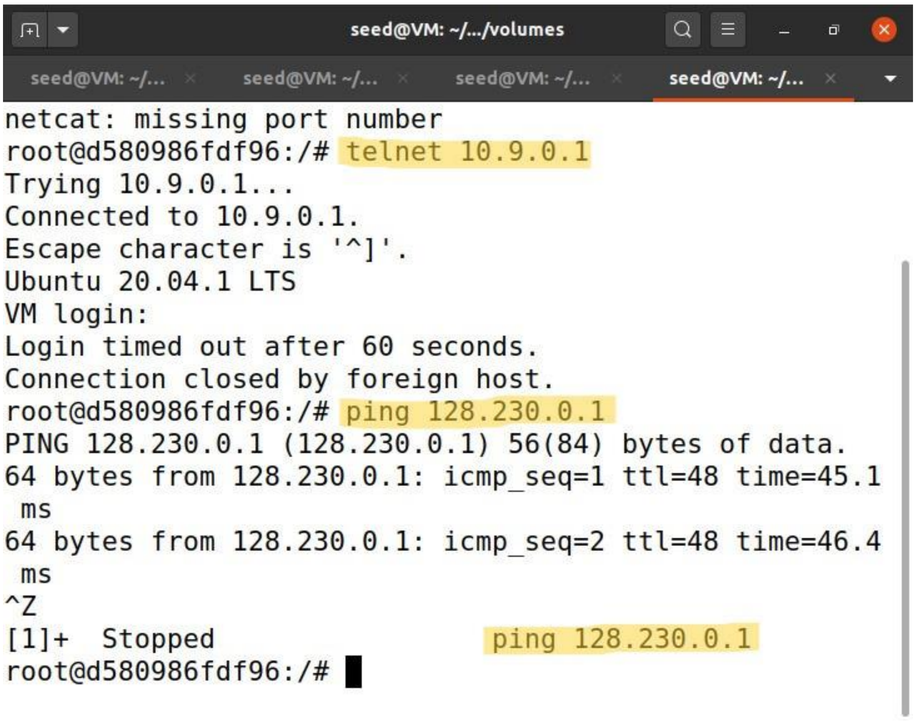


```
[01/29/23]seed@VM:~/.../volumes$ sudo ./sniffer.py
yash
^Z
[1]+  Stopped                 sudo ./sniffer.py
[01/29/23]seed@VM:~/.../volumes$ vi sniffer.py
[01/29/23]seed@VM:~/.../volumes$ sudo ./sniffer.py
yash
###[ Ethernet ]###
  dst       = 02:42:94:40:ac:44
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x10
     len       = 60
     id        = 13763
     flags     = DF
     frag      = 0
```



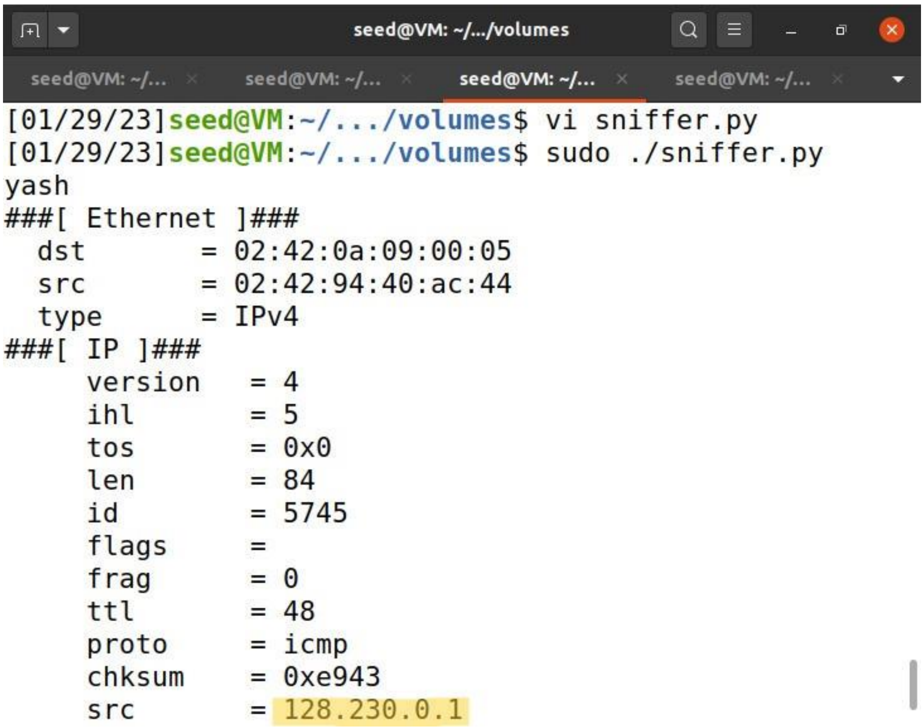
```
     src       = 10.9.0.5
     dst       = 10.9.0.1
     \options   \
###[ TCP ]###
        sport     = 36698
        dport     = telnet
        seq       = 2969970575
        ack       = 0
        dataofs   = 10
        reserved  = 0
        flags     = S
        window    = 64240
        chksum    = 0x1446
        urgptr    = 0
        options   = [('MSS', 1460), ('SAckOK', b''), (
'Timestamp', (117188549, 0)), ('NOP', None), ('WScale'
, 7)]

###[ Ethernet ]###
```

Sniffing with provided subnet address:



```
netcat: missing port number
root@d580986fdf96:/# telnet 10.9.0.1
Trying 10.9.0.1...
Connected to 10.9.0.1.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
VM login:
Login timed out after 60 seconds.
Connection closed by foreign host.
root@d580986fdf96:/# ping 128.230.0.1
PING 128.230.0.1 (128.230.0.1) 56(84) bytes of data.
64 bytes from 128.230.0.1: icmp_seq=1 ttl=48 time=45.1
 ms
64 bytes from 128.230.0.1: icmp_seq=2 ttl=48 time=46.4
 ms
^Z
[1]+  Stopped                 ping 128.230.0.1
root@d580986fdf96:/# 
```

Subnet is highlighted



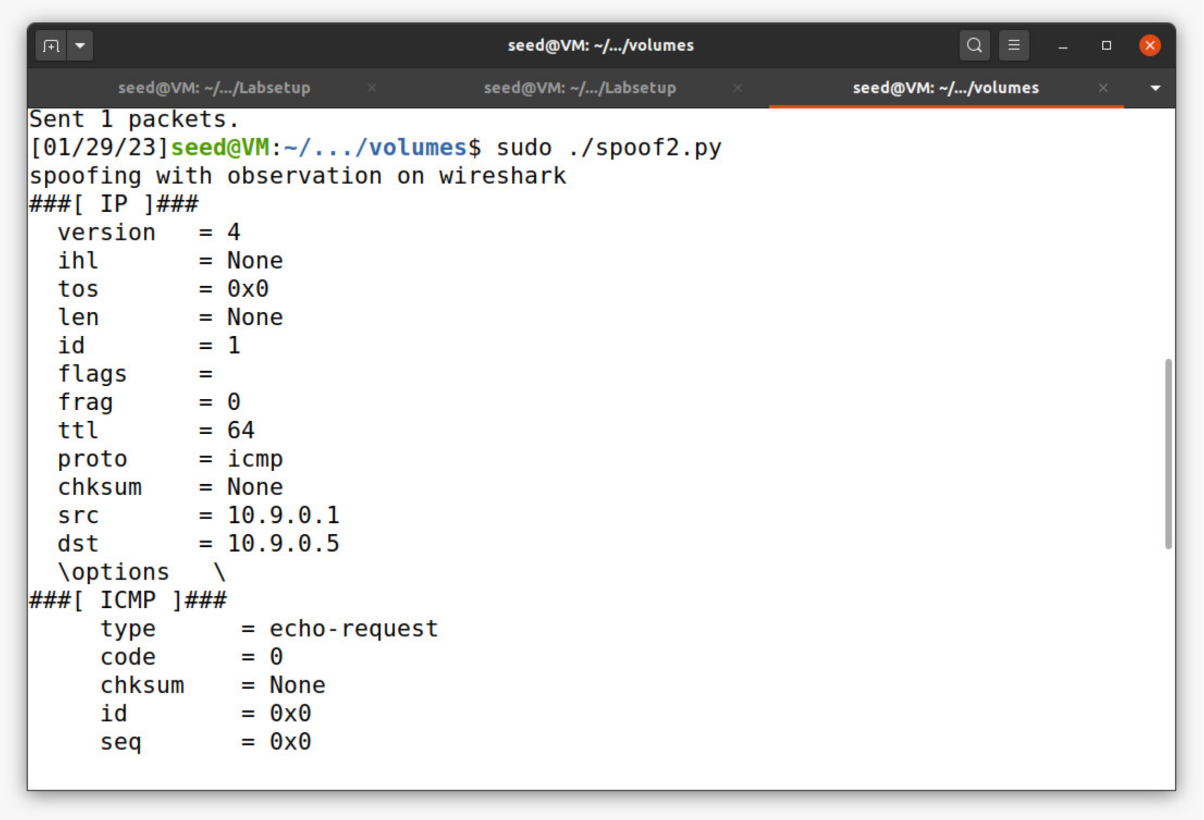
```
[01/29/23]seed@VM:~/.../volumes$ vi sniffer.py
[01/29/23]seed@VM:~/.../volumes$ sudo ./sniffer.py
yash
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:94:40:ac:44
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 5745
     flags     =
     frag      = 0
     ttl       = 48
     proto     = icmp
     chksum    = 0xe943
     src       = 128.230.0.1
```

TASK 1.2 Spoofing

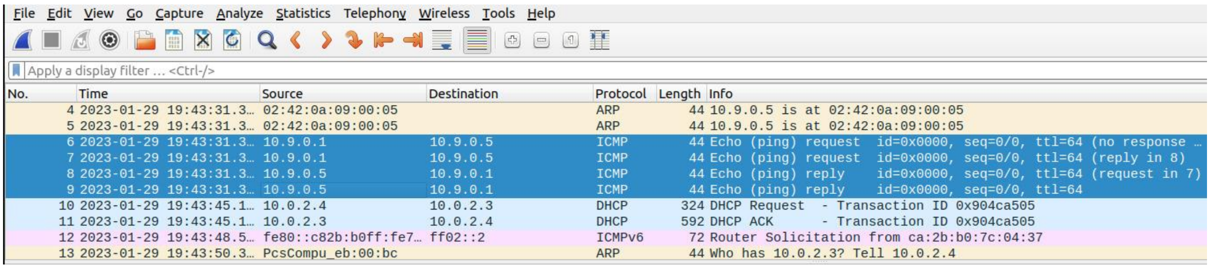
The IP method defines an ip layer. We set the source to the attacker's ip and the destination to one of the host IP's. Then we use ICMP method and packet is constructed, the division operator here is for overloading. It adds ICMP as payload of IP.

In this task, we can see the packet information in the terminal and from wireshark we can view the captured packets.

Spoofing to an existing address results in a reply.

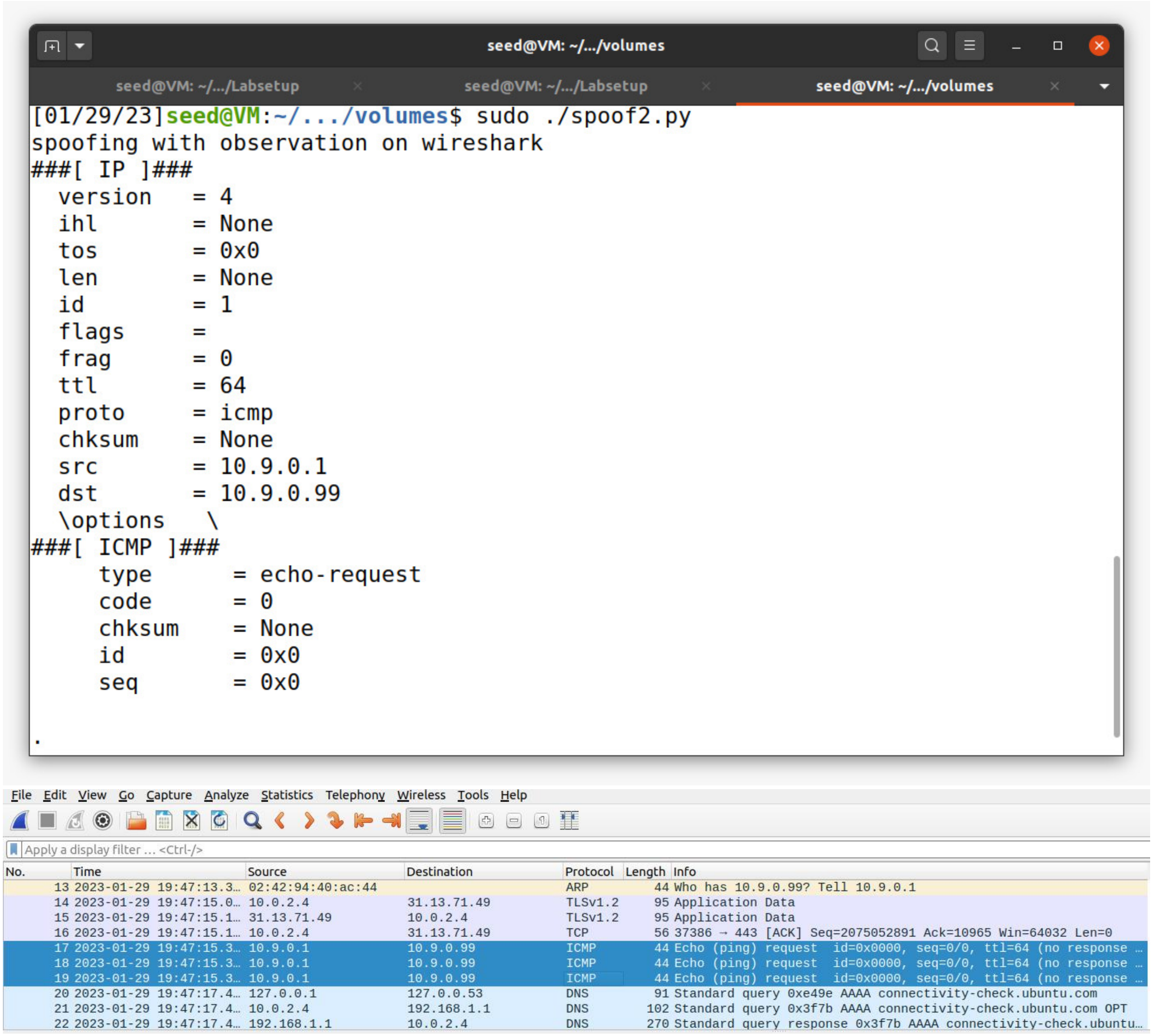


```
Sent 1 packets.
[01/29/23]seed@VM:~/.../volumes$ sudo ./spoof2.py
spoofing with observation on wireshark
###[ IP ]###
  version   = 4
  ihl       = None
  tos       = 0x0
  len       = None
  id        = 1
  flags     =
  frag      = 0
  ttl       = 64
  proto     = icmp
  chksum    = None
  src       = 10.9.0.1
  dst       = 10.9.0.5
  \options   \
###[ ICMP ]###
     type      = echo-request
     code      = 0
     chksum    = None
     id        = 0x0
     seq       = 0x0
```


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
4	2023-01-29 19:43:31.3…	02:42:0a:09:00:05		ARP	44	10.9.0.5 is at 02:42:0a:09:00:05
5	2023-01-29 19:43:31.3…	02:42:0a:09:00:05		ARP	44	10.9.0.5 is at 02:42:0a:09:00:05
6	2023-01-29 19:43:31.3…	10.9.0.1	10.9.0.5	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (no response …
7	2023-01-29 19:43:31.3…	10.9.0.1	10.9.0.5	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 8)
8	2023-01-29 19:43:31.3…	10.9.0.5	10.9.0.1	ICMP	44	Echo (ping) reply id=0x0000, seq=0/0, ttl=64 (request in 7)
9	2023-01-29 19:43:31.3…	10.9.0.5	10.9.0.1	ICMP	44	Echo (ping) reply id=0x0000, seq=0/0, ttl=64
10	2023-01-29 19:43:45.1…	10.0.2.4	10.0.2.3	DHCP	324	DHCP Request - Transaction ID 0x904ca505
11	2023-01-29 19:43:45.1…	10.0.2.3	10.0.2.4	DHCP	592	DHCP ACK - Transaction ID 0x904ca505
12	2023-01-29 19:43:48.5…	fe80::c82b:b0ff:fe7…	ff02::2	ICMPv6	72	Router Solicitation from ca:2b:b0:7c:04:37
13	2023-01-29 19:43:50.3…	PcsCompu_eb:00:bc		ARP	44	Who has 10.0.2.3? Tell 10.0.2.4

Spoofing to an address which does not exist resulted in no replies as seen through wireshark. The request was sent to the destination but it couldn't get it so we got no replies.


seed@VM: ~/.../volumes
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup
seed@VM: ~/.../volumes
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

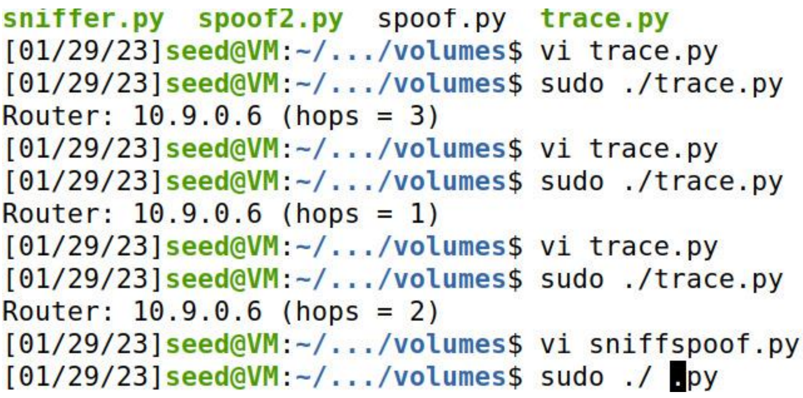
```
[01/29/23]seed@VM:~/.../volumes$ sudo ./spoof2.py
spoofing with observation on wireshark
###[ IP ]###
  version   = 4
  ihl       = None
  tos       = 0x0
  len       = None
  id        = 1
  flags     =
  frag      = 0
  ttl       = 64
  proto     = icmp
  chksum    = None
  src       = 10.9.0.1
  dst       = 10.9.0.99
  \options   \
###[ ICMP ]###
     type      = echo-request
     code      = 0
     chksum    = None
     id        = 0x0
     seq       = 0x0

.
```

No.	Time	Source	Destination	Protocol	Length	Info
13	2023-01-29 19:47:13.3…	02:42:94:40:ac:44		ARP	44	Who has 10.9.0.99? Tell 10.9.0.1
14	2023-01-29 19:47:15.0…	10.0.2.4	31.13.71.49	TLSv1.2	95	Application Data
15	2023-01-29 19:47:15.1…	31.13.71.49	10.0.2.4	TLSv1.2	95	Application Data
16	2023-01-29 19:47:15.1…	10.0.2.4	31.13.71.49	TCP	56	37386 → 443 [ACK] Seq=2075052891 Ack=10965 Win=64032 Len=0
17	2023-01-29 19:47:15.3…	10.9.0.1	10.9.0.99	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (no response …
18	2023-01-29 19:47:15.3…	10.9.0.1	10.9.0.99	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (no response …
19	2023-01-29 19:47:15.3…	10.9.0.1	10.9.0.99	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (no response …
20	2023-01-29 19:47:17.4…	127.0.0.1	127.0.0.53	DNS	91	Standard query 0xe49e AAAA connectivity-check.ubuntu.com
21	2023-01-29 19:47:17.4…	10.0.2.4	192.168.1.1	DNS	102	Standard query 0x3f7b AAAA connectivity-check.ubuntu.com OPT
22	2023-01-29 19:47:17.4…	192.168.1.1	10.0.2.4	DNS	270	Standard query response 0x3f7b AAAA connectivity-check.ubuntu…

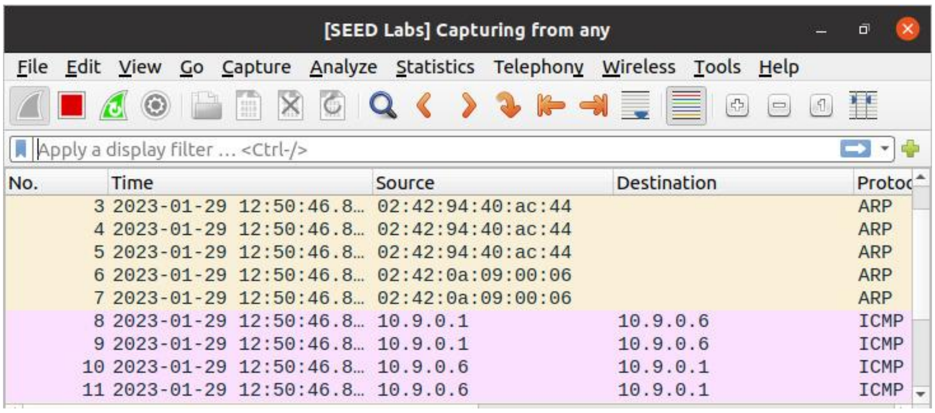
TASK 1.3 Trace route

TTL stands for time to live factor. It measures the distance between the source and the destination. We send the packet using function sr1, it waits for the reply from the destination.

Running the program repeatedly while changing the ttl value in the program.

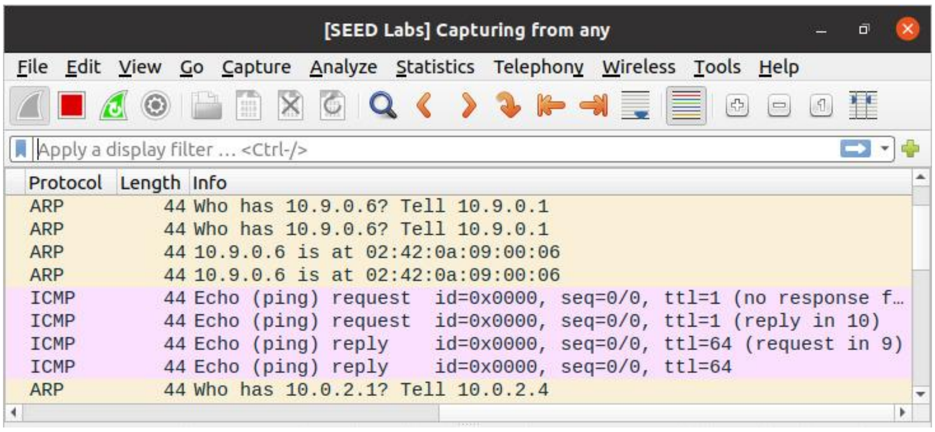


```
sniffer.py  spoof2.py  spoof.py  trace.py
[01/29/23]seed@VM:~/.../volumes$ vi trace.py
[01/29/23]seed@VM:~/.../volumes$ sudo ./trace.py
Router: 10.9.0.6 (hops = 3)
[01/29/23]seed@VM:~/.../volumes$ vi trace.py
[01/29/23]seed@VM:~/.../volumes$ sudo ./trace.py
Router: 10.9.0.6 (hops = 1)
[01/29/23]seed@VM:~/.../volumes$ vi trace.py
[01/29/23]seed@VM:~/.../volumes$ sudo ./trace.py
Router: 10.9.0.6 (hops = 2)
[01/29/23]seed@VM:~/.../volumes$ vi sniffspoof.py
[01/29/23]seed@VM:~/.../volumes$ sudo ./ .py
```



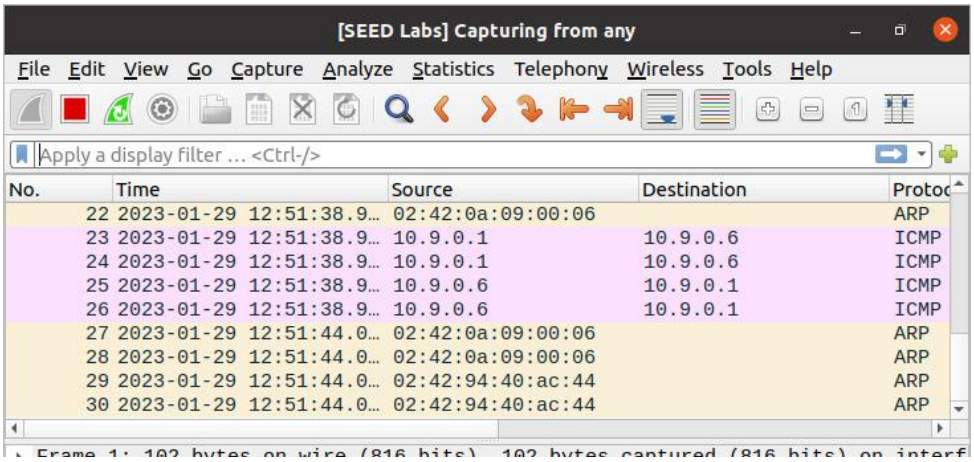
[SEED Labs] Capturing from any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

No.	Time	Source	Destination	Protoc
3	2023-01-29 12:50:46.8…	02:42:94:40:ac:44		ARP
4	2023-01-29 12:50:46.8…	02:42:94:40:ac:44		ARP
5	2023-01-29 12:50:46.8…	02:42:94:40:ac:44		ARP
6	2023-01-29 12:50:46.8…	02:42:0a:09:00:06		ARP
7	2023-01-29 12:50:46.8…	02:42:0a:09:00:06		ARP
8	2023-01-29 12:50:46.8…	10.9.0.1	10.9.0.6	ICMP
9	2023-01-29 12:50:46.8…	10.9.0.1	10.9.0.6	ICMP
10	2023-01-29 12:50:46.8…	10.9.0.6	10.9.0.1	ICMP
11	2023-01-29 12:50:46.8…	10.9.0.6	10.9.0.1	ICMP



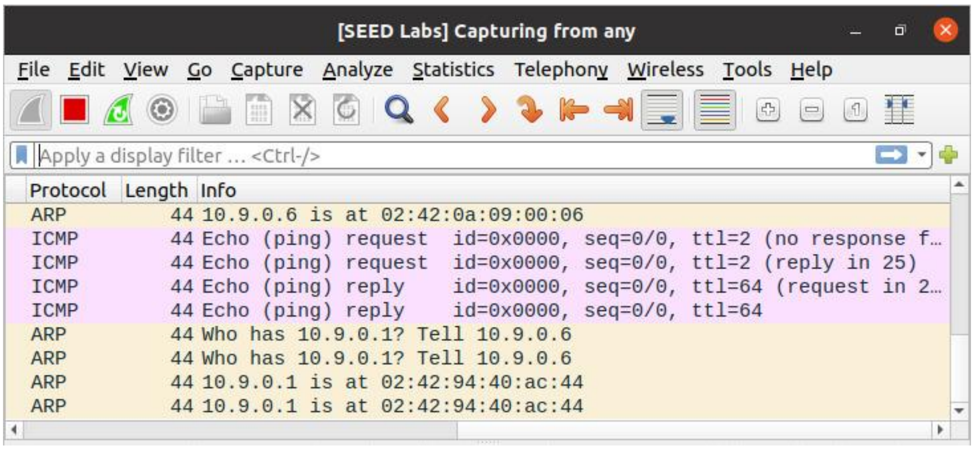
[SEED Labs] Capturing from any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

Protocol	Length	Info
ARP	44	Who has 10.9.0.6? Tell 10.9.0.1
ARP	44	Who has 10.9.0.6? Tell 10.9.0.1
ARP	44	10.9.0.6 is at 02:42:0a:09:00:06
ARP	44	10.9.0.6 is at 02:42:0a:09:00:06
ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=1 (no response f…
ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=1 (reply in 10)
ICMP	44	Echo (ping) reply id=0x0000, seq=0/0, ttl=64 (request in 9)
ICMP	44	Echo (ping) reply id=0x0000, seq=0/0, ttl=64
ARP	44	Who has 10.0.2.1? Tell 10.0.2.4



[SEED Labs] Capturing from any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.   Time                      Source              Destination    Protoc
   22 2023-01-29 12:51:38.9…   02:42:0a:09:00:06                  ARP
   23 2023-01-29 12:51:38.9…   10.9.0.1            10.9.0.6       ICMP
   24 2023-01-29 12:51:38.9…   10.9.0.1            10.9.0.6       ICMP
   25 2023-01-29 12:51:38.9…   10.9.0.6            10.9.0.1       ICMP
   26 2023-01-29 12:51:38.9…   10.9.0.6            10.9.0.1       ICMP
   27 2023-01-29 12:51:44.0…   02:42:0a:09:00:06                  ARP
   28 2023-01-29 12:51:44.0…   02:42:0a:09:00:06                  ARP
   29 2023-01-29 12:51:44.0…   02:42:94:40:ac:44                  ARP
   30 2023-01-29 12:51:44.0…   02:42:94:40:ac:44                  ARP
› Frame 1: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interf
```



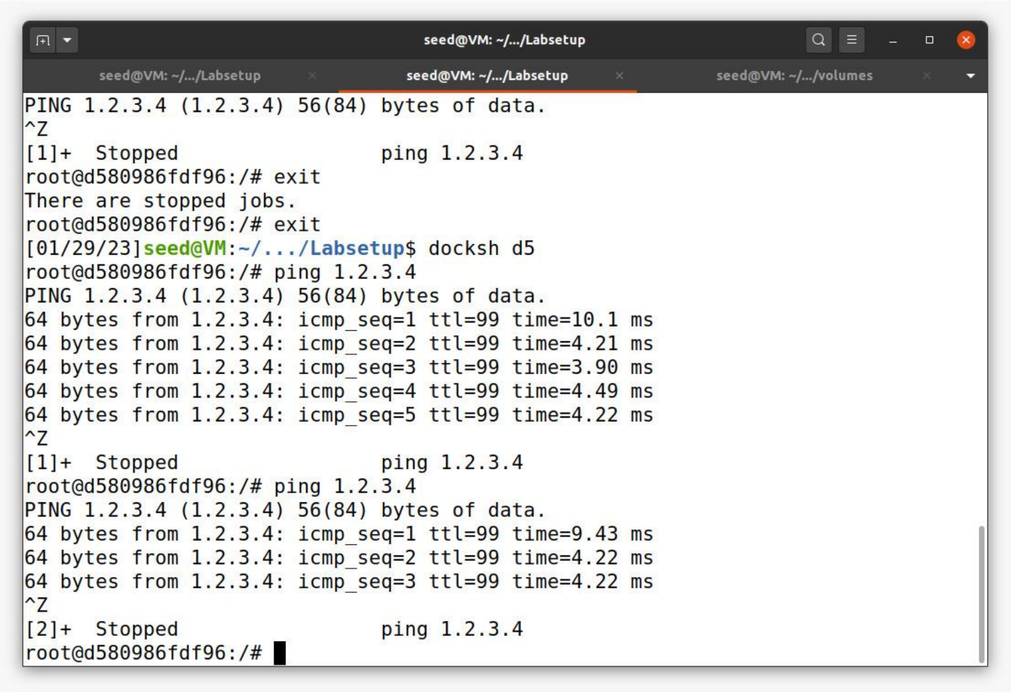
[SEED Labs] Capturing from any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
Protocol  Length Info
ARP           44 10.9.0.6 is at 02:42:0a:09:00:06
ICMP          44 Echo (ping) request  id=0x0000, seq=0/0, ttl=2 (no response f…
ICMP          44 Echo (ping) request  id=0x0000, seq=0/0, ttl=2 (reply in 25)
ICMP          44 Echo (ping) reply    id=0x0000, seq=0/0, ttl=64 (request in 2…
ICMP          44 Echo (ping) reply    id=0x0000, seq=0/0, ttl=64
ARP           44 Who has 10.9.0.1? Tell 10.9.0.6
ARP           44 Who has 10.9.0.1? Tell 10.9.0.6
ARP           44 10.9.0.1 is at 02:42:94:40:ac:44
ARP           44 10.9.0.1 is at 02:42:94:40:ac:44
```

Task 1.4

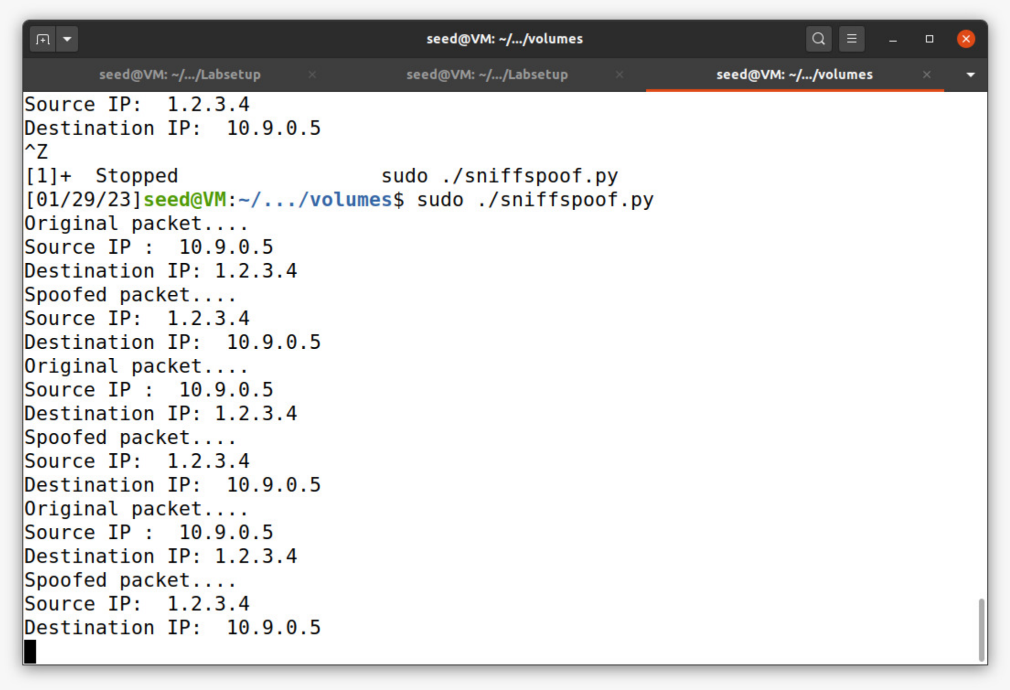
We will find a spoof method which takes a sniff packet as an argument. We retrive the source ip and destination ip from the sniff packets and create a new ip packet. The packet's destination is stored as source IP and packet's source ip is stored as destination IP, so that we can send a reply and the reply seems to be valid from a non existing source IP and then we construct the packet to be spoofed.

Pinging a random address which does not exist:



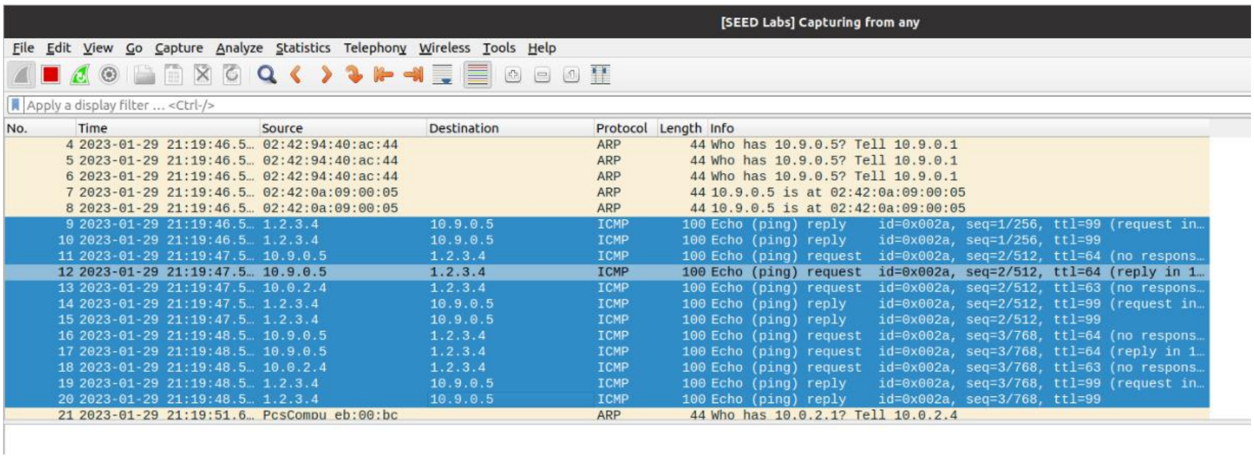
```
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
^Z
[1]+  Stopped                 ping 1.2.3.4
root@d580986fdf96:/# exit
There are stopped jobs.
root@d580986fdf96:/# exit
[01/29/23]seed@VM:~/.../Labsetup$ docksh d5
root@d580986fdf96:/# ping 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
64 bytes from 1.2.3.4: icmp_seq=1 ttl=99 time=10.1 ms
64 bytes from 1.2.3.4: icmp_seq=2 ttl=99 time=4.21 ms
64 bytes from 1.2.3.4: icmp_seq=3 ttl=99 time=3.90 ms
64 bytes from 1.2.3.4: icmp_seq=4 ttl=99 time=4.49 ms
64 bytes from 1.2.3.4: icmp_seq=5 ttl=99 time=4.22 ms
^Z
[1]+  Stopped                 ping 1.2.3.4
root@d580986fdf96:/# 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=99 time=9.43 ms
64 bytes from 1.2.3.4: icmp_seq=2 ttl=99 time=4.22 ms
64 bytes from 1.2.3.4: icmp_seq=3 ttl=99 time=4.22 ms
^Z
[2]+  Stopped                 ping 1.2.3.4
root@d580986fdf96:/# 
```

It can be observed how the source ip of the spoofed packet is the ip address (1.2.3.4) which does not exist and the destination is the host who initially sent the original packet.



```
Source IP:  1.2.3.4
Destination IP:  10.9.0.5
^Z
[1]+  Stopped                 sudo ./sniffspoof.py
[01/29/23]seed@VM:~/.../volumes$ sudo ./sniffspoof.py
Original packet....
Source IP :  10.9.0.5
Destination IP: 1.2.3.4
Spoofed packet....
Source IP:  1.2.3.4
Destination IP:  10.9.0.5
Original packet....
Source IP :  10.9.0.5
Destination IP: 1.2.3.4
Spoofed packet....
Source IP:  1.2.3.4
Destination IP:  10.9.0.5
Original packet....
Source IP :  10.9.0.5
Destination IP: 1.2.3.4
Spoofed packet....
Source IP:  1.2.3.4
Destination IP:  10.9.0.5
```

Observation through Wireshark: we can the original packet and the spoofed packet with the help of the ip addresses that we switched.


[SEED Labs] Capturing from any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
4	2023-01-29 21:19:46.5…	02:42:94:40:ac:44		ARP	44	Who has 10.9.0.5? Tell 10.9.0.1
5	2023-01-29 21:19:46.5…	02:42:94:40:ac:44		ARP	44	Who has 10.9.0.5? Tell 10.9.0.1
6	2023-01-29 21:19:46.5…	02:42:94:40:ac:44		ARP	44	Who has 10.9.0.5? Tell 10.9.0.1
7	2023-01-29 21:19:46.5…	02:42:0a:09:00:05		ARP	44	10.9.0.5 is at 02:42:0a:09:00:05
8	2023-01-29 21:19:46.5…	02:42:0a:09:00:05		ARP	44	10.9.0.5 is at 02:42:0a:09:00:05
9	2023-01-29 21:19:46.5…	1.2.3.4	10.9.0.5	ICMP	100	Echo (ping) reply id=0x002a, seq=1/256, ttl=99 (request in…
10	2023-01-29 21:19:46.5…	1.2.3.4	10.9.0.5	ICMP	100	Echo (ping) reply id=0x002a, seq=1/256, ttl=99
11	2023-01-29 21:19:47.5…	10.9.0.5	1.2.3.4	ICMP	100	Echo (ping) request id=0x002a, seq=2/512, ttl=64 (no respons…
12	2023-01-29 21:19:47.5…	10.9.0.5	1.2.3.4	ICMP	100	Echo (ping) request id=0x002a, seq=2/512, ttl=64 (reply in 1…
13	2023-01-29 21:19:47.5…	10.0.2.4	1.2.3.4	ICMP	100	Echo (ping) request id=0x002a, seq=2/512, ttl=63 (no respons…
14	2023-01-29 21:19:47.5…	1.2.3.4	10.9.0.5	ICMP	100	Echo (ping) reply id=0x002a, seq=2/512, ttl=99 (request in…
15	2023-01-29 21:19:47.5…	1.2.3.4	10.9.0.5	ICMP	100	Echo (ping) reply id=0x002a, seq=2/512, ttl=99
16	2023-01-29 21:19:48.5…	10.9.0.5	1.2.3.4	ICMP	100	Echo (ping) request id=0x002a, seq=3/768, ttl=64 (no respons…
17	2023-01-29 21:19:48.5…	10.9.0.5	1.2.3.4	ICMP	100	Echo (ping) request id=0x002a, seq=3/768, ttl=64 (reply in 1…
18	2023-01-29 21:19:48.5…	10.0.2.4	1.2.3.4	ICMP	100	Echo (ping) request id=0x002a, seq=3/768, ttl=63 (no respons…
19	2023-01-29 21:19:48.5…	1.2.3.4	10.9.0.5	ICMP	100	Echo (ping) reply id=0x002a, seq=3/768, ttl=99 (request in…
20	2023-01-29 21:19:48.5…	1.2.3.4	10.9.0.5	ICMP	100	Echo (ping) reply id=0x002a, seq=3/768, ttl=99
21	2023-01-29 21:19:51.6…	PcsCompu eb:00:bc		ARP	44	Who has 10.0.2.1? Tell 10.0.2.4