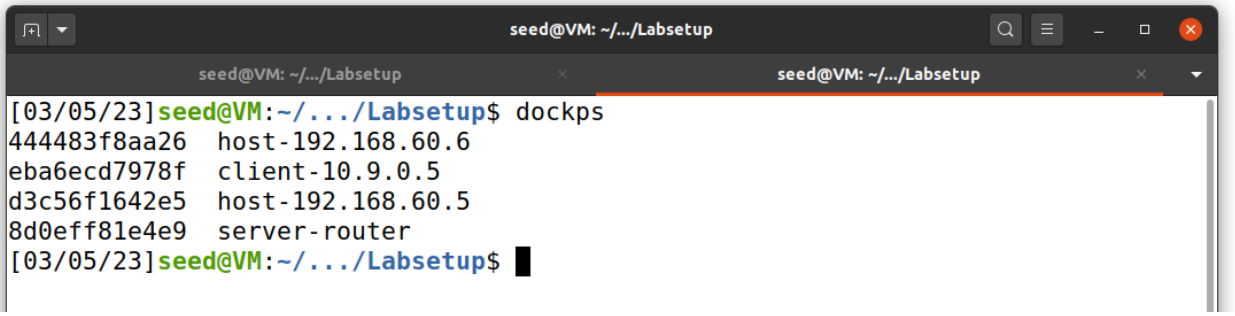
NAME: YASH SNEHAL SHETIYA

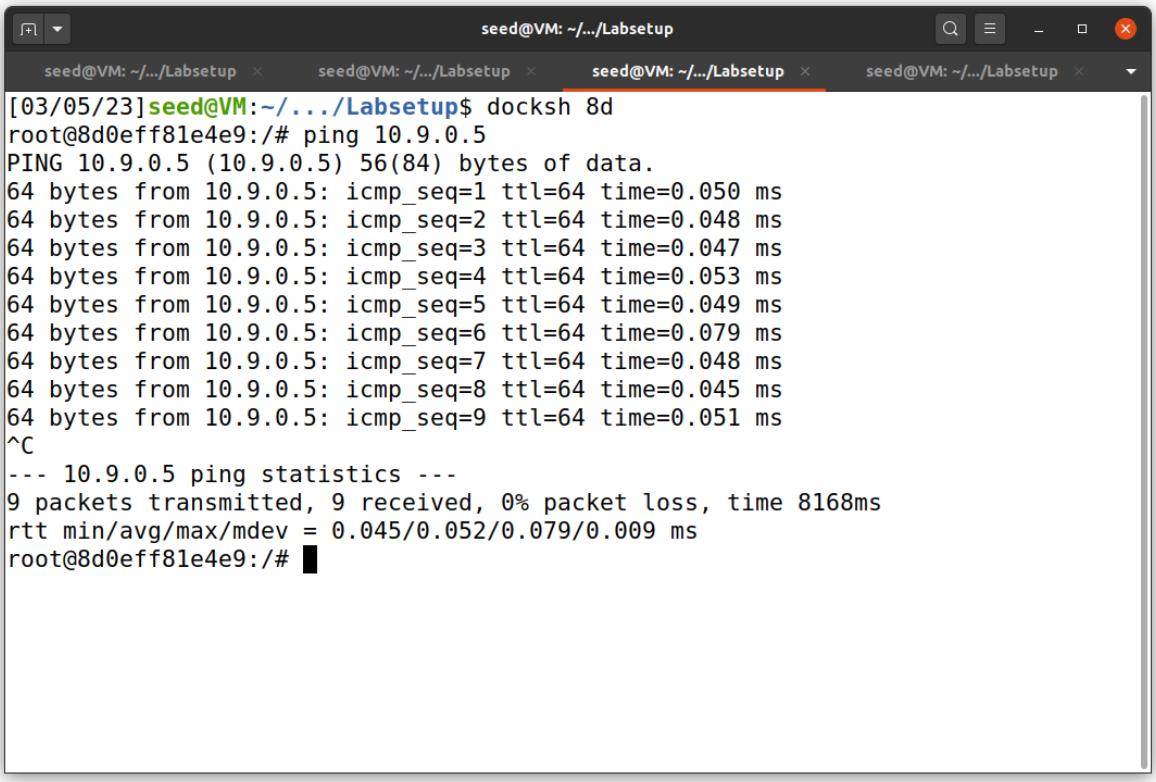
SUID: 9276568741

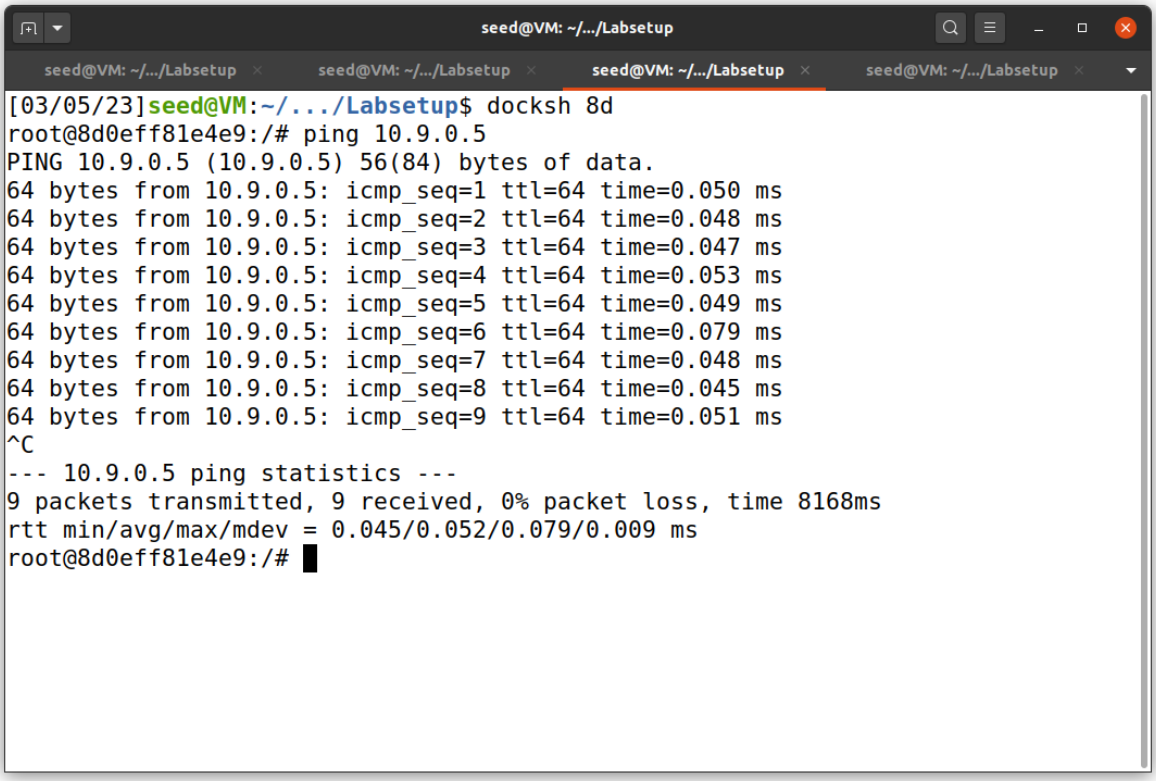
LAB: VPN TUNNELING

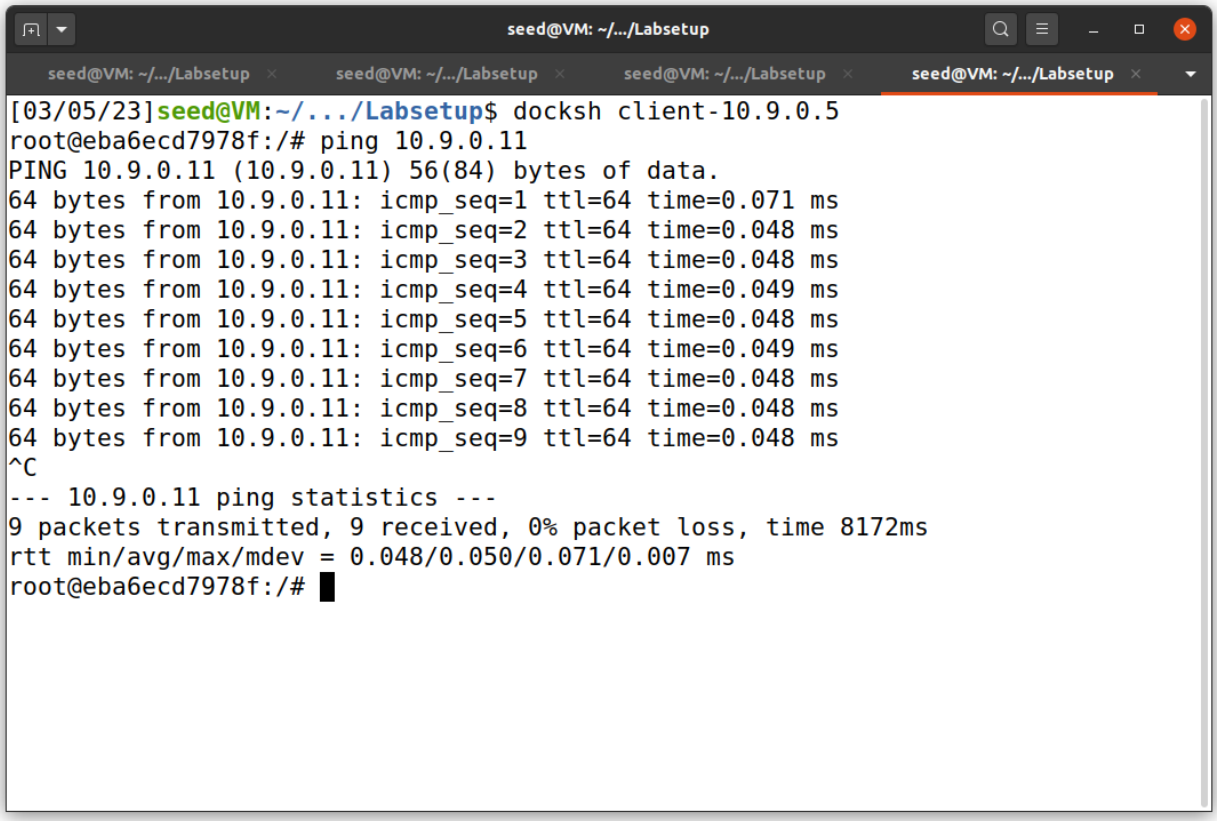
****

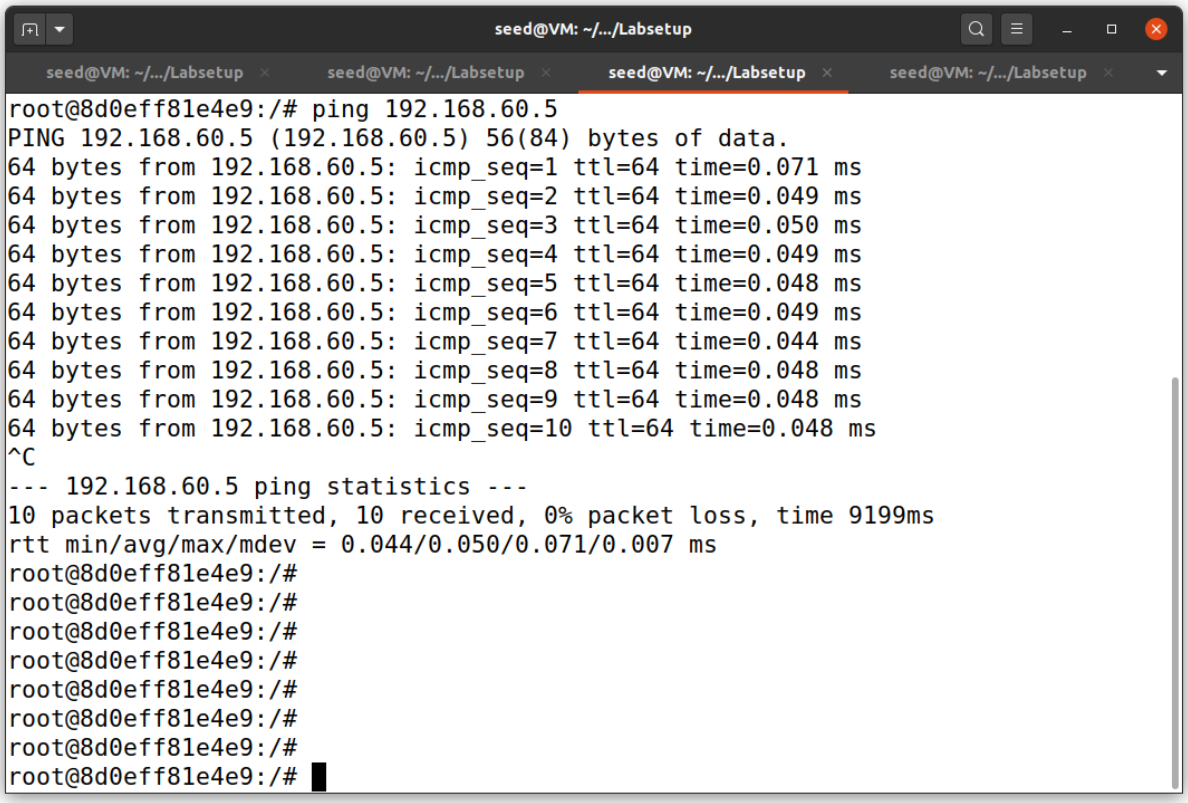
TASK 1:

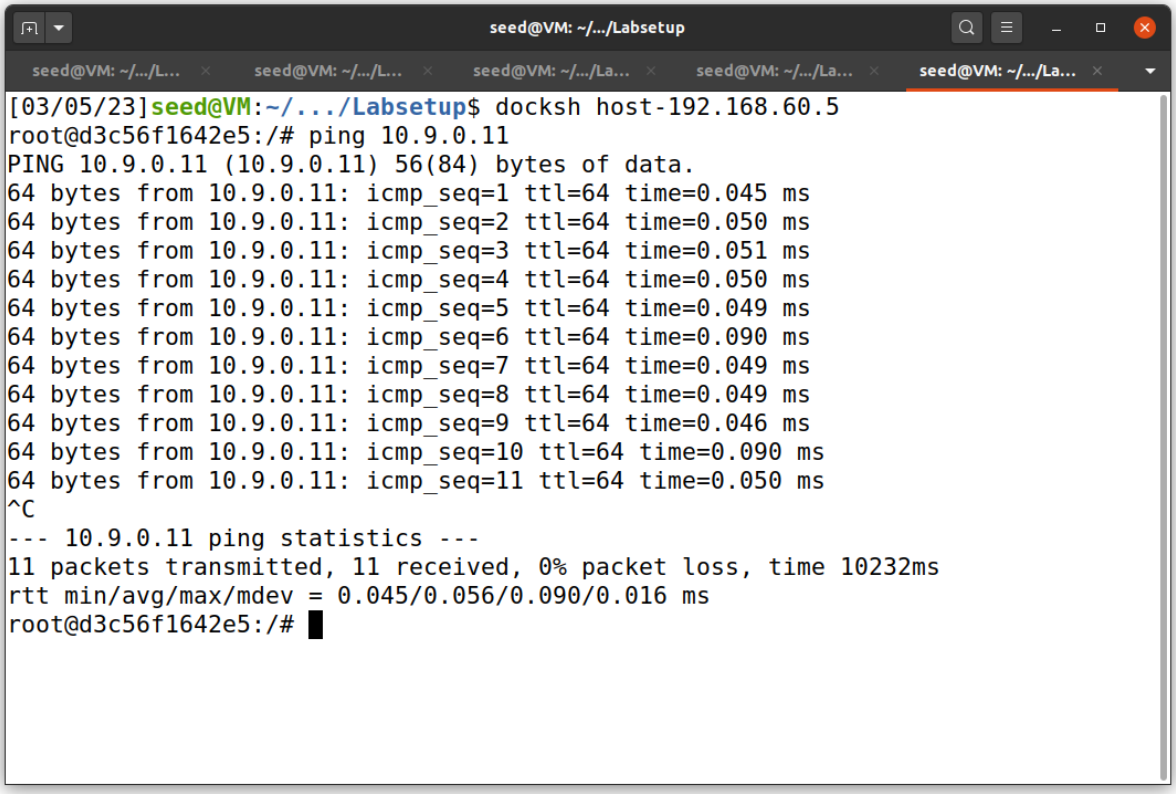




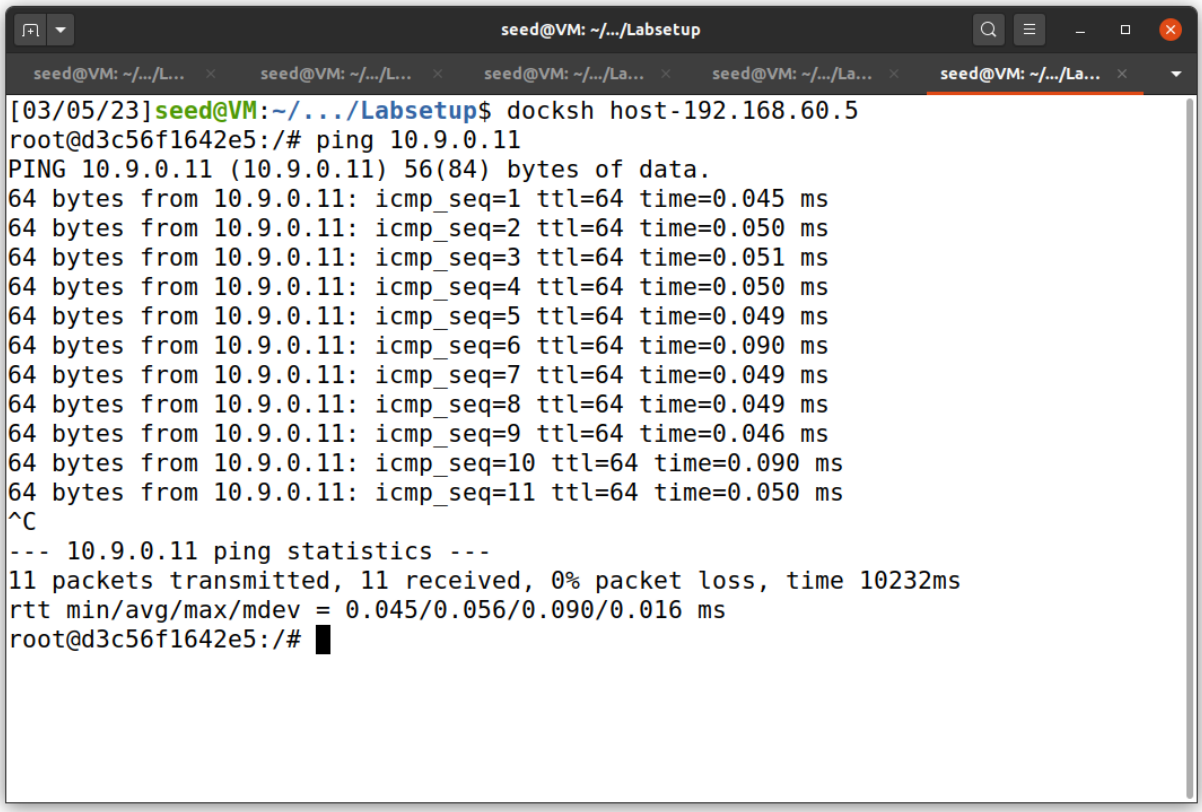

seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup
[03/05/23]seed@VM:~/.../Labsetup$ docksh 8d
root@8d0eff81e4e9:/# ping 10.9.0.5
PING 10.9.0.5 (10.9.0.5) 56(84) bytes of data.
64 bytes from 10.9.0.5: icmp_seq=1 ttl=64 time=0.050 ms
64 bytes from 10.9.0.5: icmp_seq=2 ttl=64 time=0.048 ms
64 bytes from 10.9.0.5: icmp_seq=3 ttl=64 time=0.047 ms
64 bytes from 10.9.0.5: icmp_seq=4 ttl=64 time=0.053 ms
64 bytes from 10.9.0.5: icmp_seq=5 ttl=64 time=0.049 ms
64 bytes from 10.9.0.5: icmp_seq=6 ttl=64 time=0.079 ms
64 bytes from 10.9.0.5: icmp_seq=7 ttl=64 time=0.048 ms
64 bytes from 10.9.0.5: icmp_seq=8 ttl=64 time=0.045 ms
64 bytes from 10.9.0.5: icmp_seq=9 ttl=64 time=0.051 ms
^C
--- 10.9.0.5 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8168ms
rtt min/avg/max/mdev = 0.045/0.052/0.079/0.009 ms
root@8d0eff81e4e9:/#


seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup
[03/05/23]seed@VM:~/.../Labsetup$ docksh client-10.9.0.5
root@eba6ecd7978f:/# ping 10.9.0.11
PING 10.9.0.11 (10.9.0.11) 56(84) bytes of data.
64 bytes from 10.9.0.11: icmp_seq=1 ttl=64 time=0.071 ms
64 bytes from 10.9.0.11: icmp_seq=2 ttl=64 time=0.048 ms
64 bytes from 10.9.0.11: icmp_seq=3 ttl=64 time=0.048 ms
64 bytes from 10.9.0.11: icmp_seq=4 ttl=64 time=0.049 ms
64 bytes from 10.9.0.11: icmp_seq=5 ttl=64 time=0.048 ms
64 bytes from 10.9.0.11: icmp_seq=6 ttl=64 time=0.049 ms
64 bytes from 10.9.0.11: icmp_seq=7 ttl=64 time=0.048 ms
64 bytes from 10.9.0.11: icmp_seq=8 ttl=64 time=0.048 ms
64 bytes from 10.9.0.11: icmp_seq=9 ttl=64 time=0.048 ms
^C
--- 10.9.0.11 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8172ms
rtt min/avg/max/mdev = 0.048/0.050/0.071/0.007 ms
root@eba6ecd7978f:/#

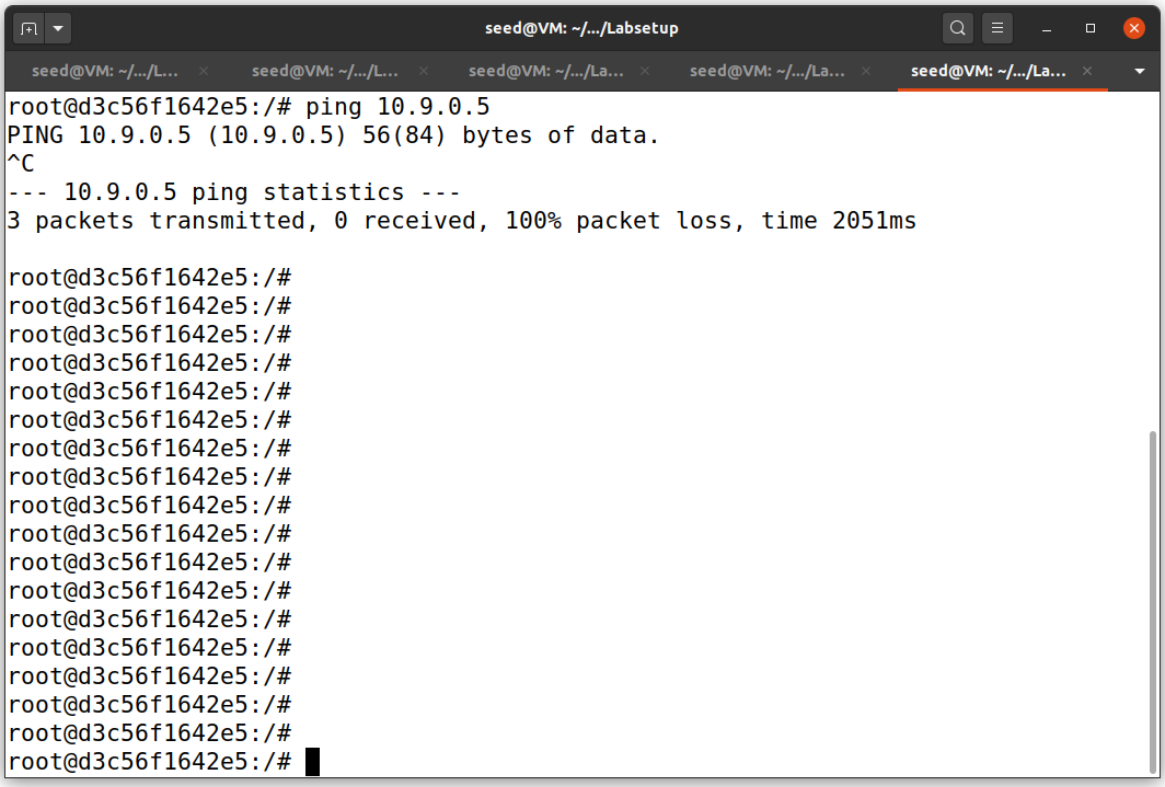




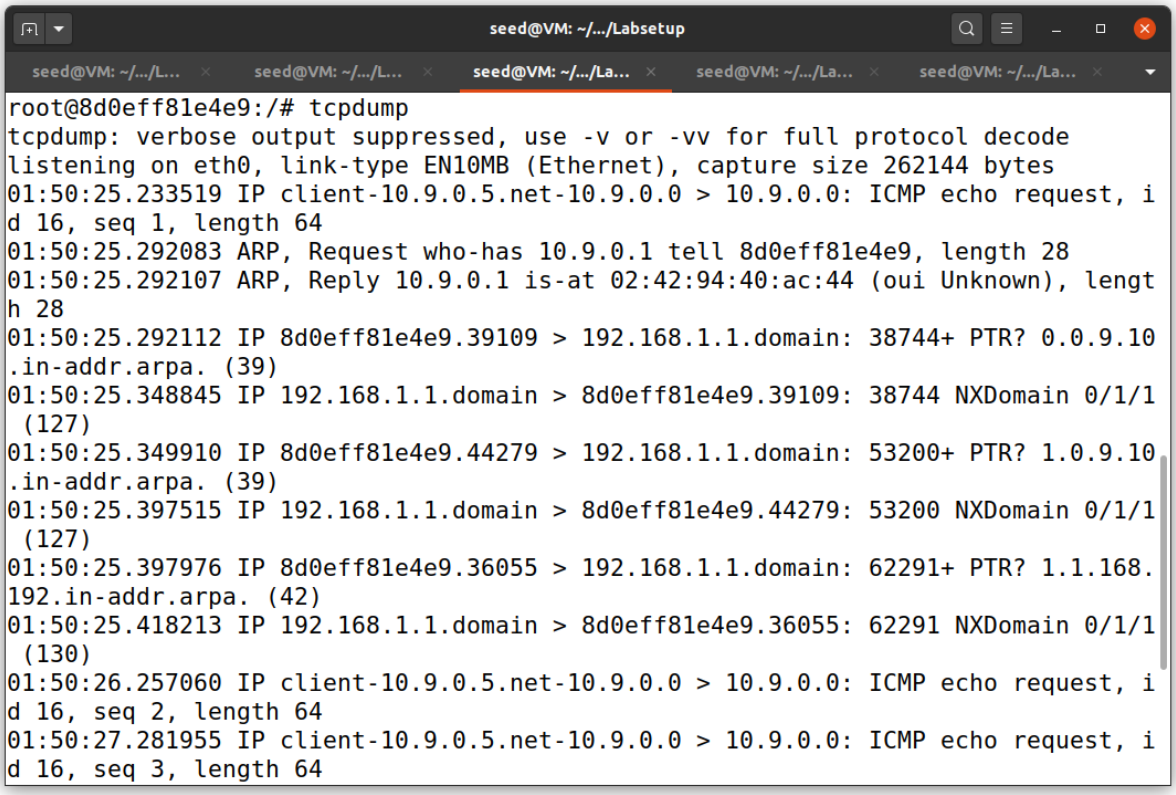
Host U to Host V



Host V to Host U:

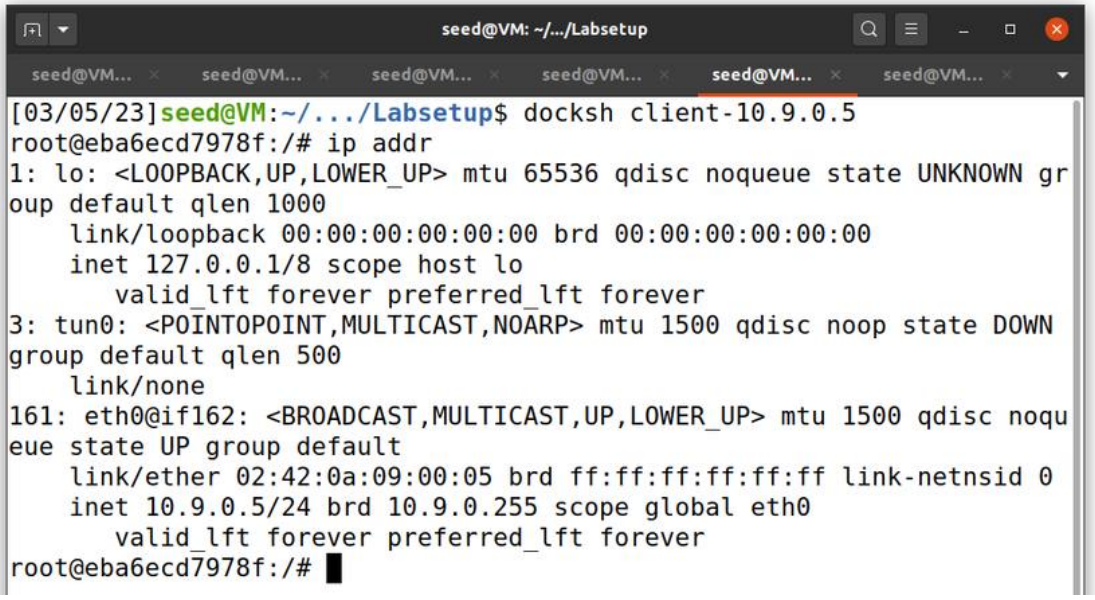


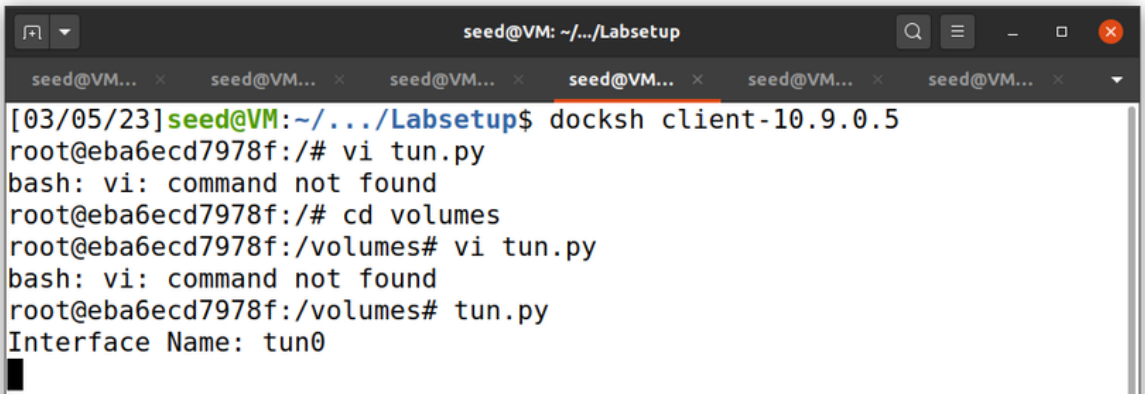
TCP dump at Host U:

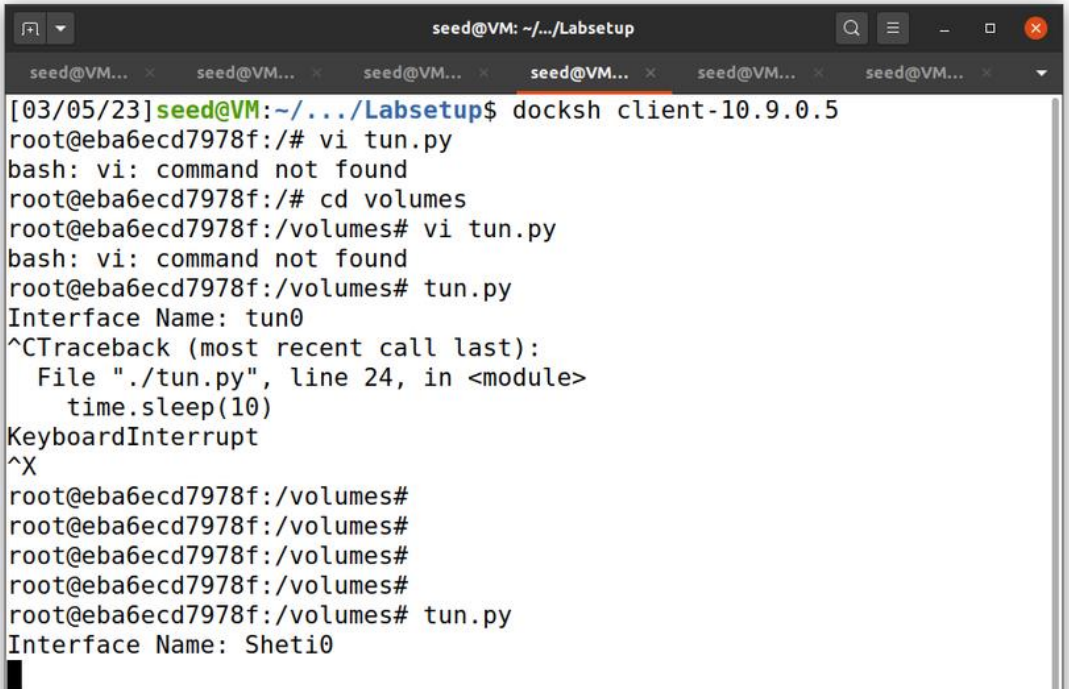


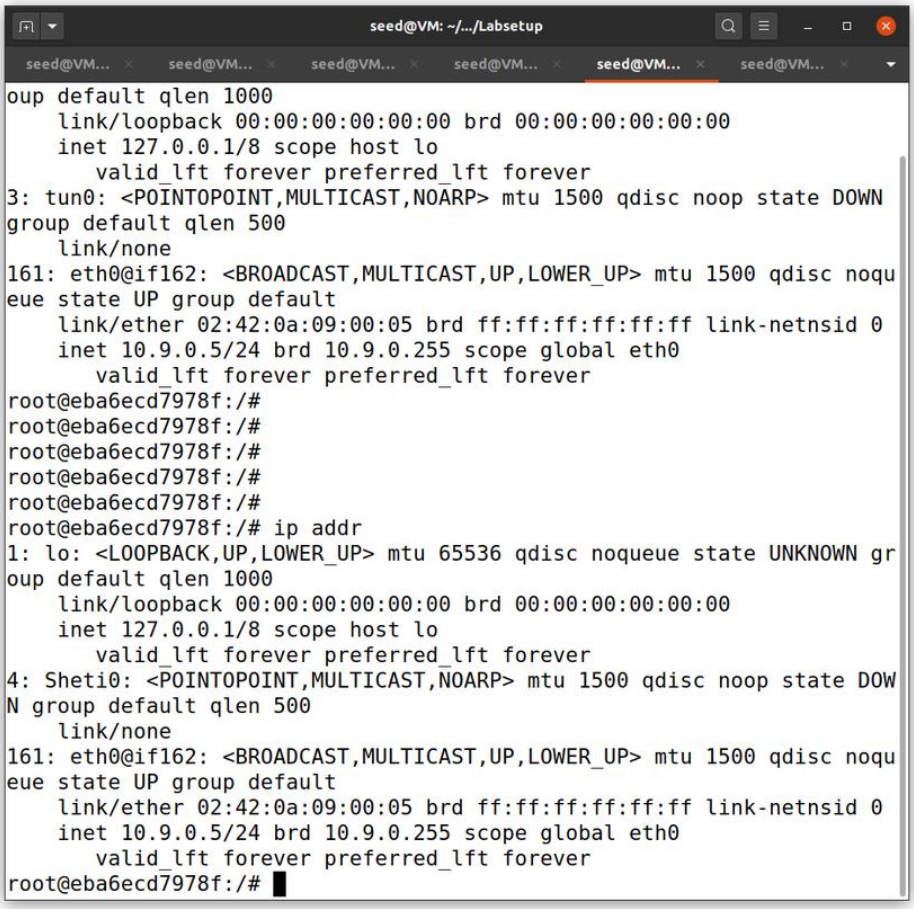
2 a)

Before:



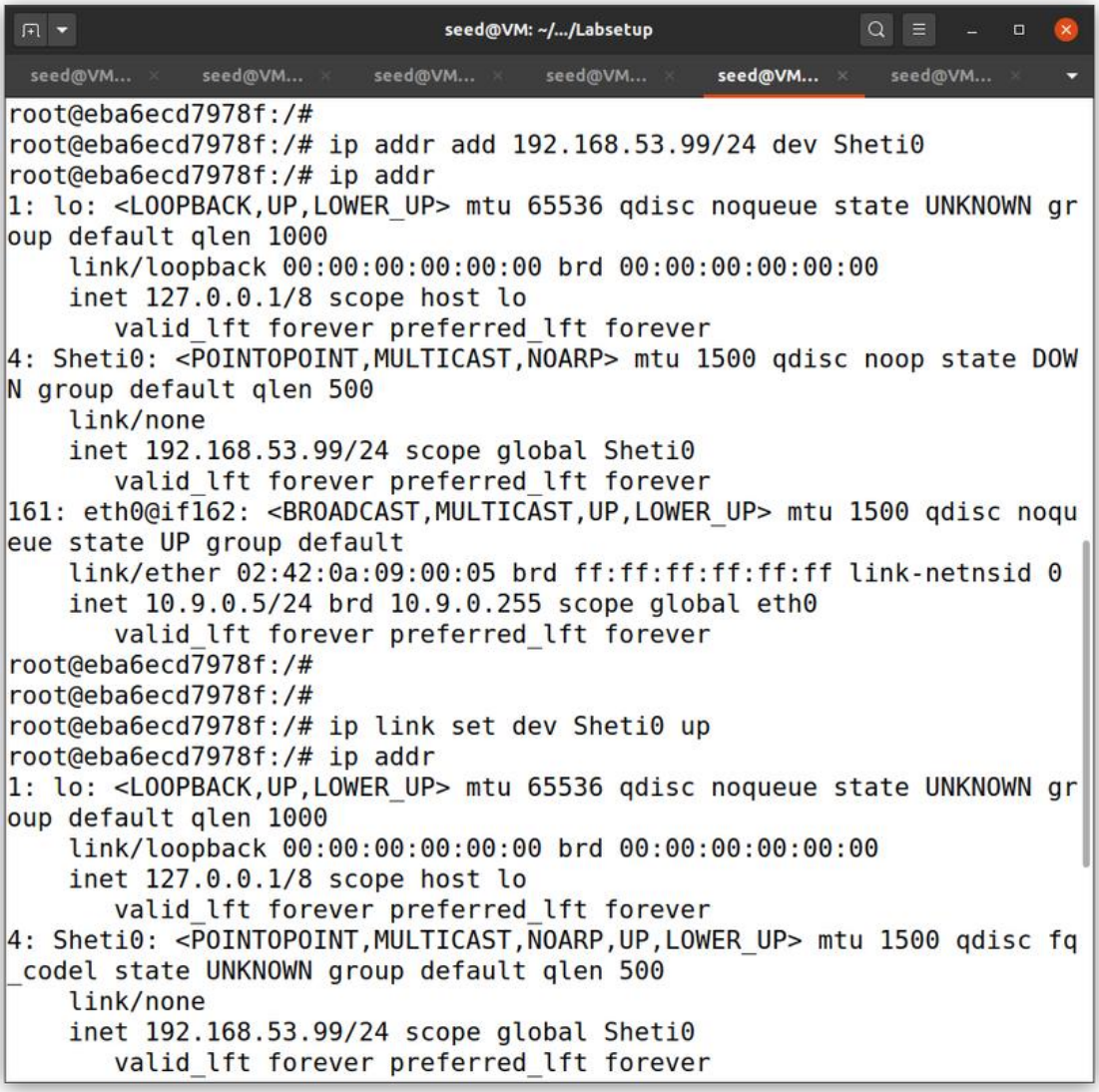

seed@VM: ~/.../Labsetup
seed@VM...
seed@VM...
seed@VM...
seed@VM...
seed@VM...
seed@VM...
[03/05/23]seed@VM:~/.../Labsetup$ docksh client-10.9.0.5
root@eba6ecd7978f:/# vi tun.py
bash: vi: command not found
root@eba6ecd7978f:/# cd volumes
root@eba6ecd7978f:/volumes# vi tun.py
bash: vi: command not found
root@eba6ecd7978f:/volumes# tun.py
Interface Name: tun0


seed@VM: ~/.../Labsetup
seed@VM...
seed@VM...
seed@VM...
seed@VM...
seed@VM...
seed@VM...
[03/05/23]seed@VM:~/.../Labsetup$ docksh client-10.9.0.5
root@eba6ecd7978f:/# vi tun.py
bash: vi: command not found
root@eba6ecd7978f:/# cd volumes
root@eba6ecd7978f:/volumes# vi tun.py
bash: vi: command not found
root@eba6ecd7978f:/volumes# tun.py
Interface Name: tun0
^CTraceback (most recent call last):
File "./tun.py", line 24, in <module>
time.sleep(10)
KeyboardInterrupt
^X
root@eba6ecd7978f:/volumes#
root@eba6ecd7978f:/volumes#
root@eba6ecd7978f:/volumes#
root@eba6ecd7978f:/volumes#
root@eba6ecd7978f:/volumes# tun.py
Interface Name: Sheti0

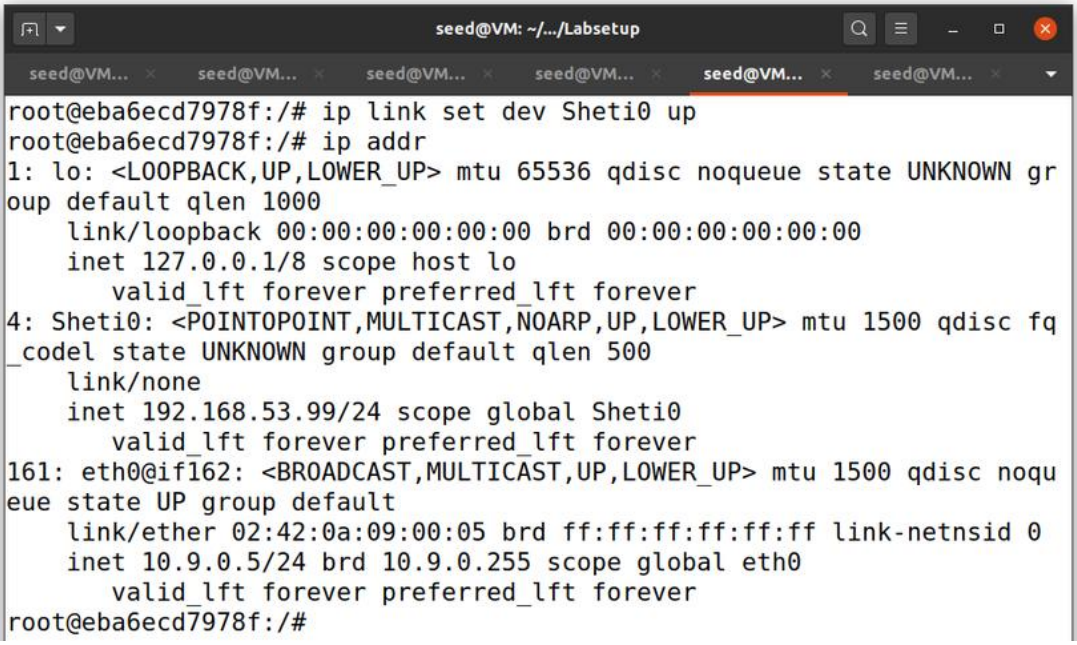


2 b)

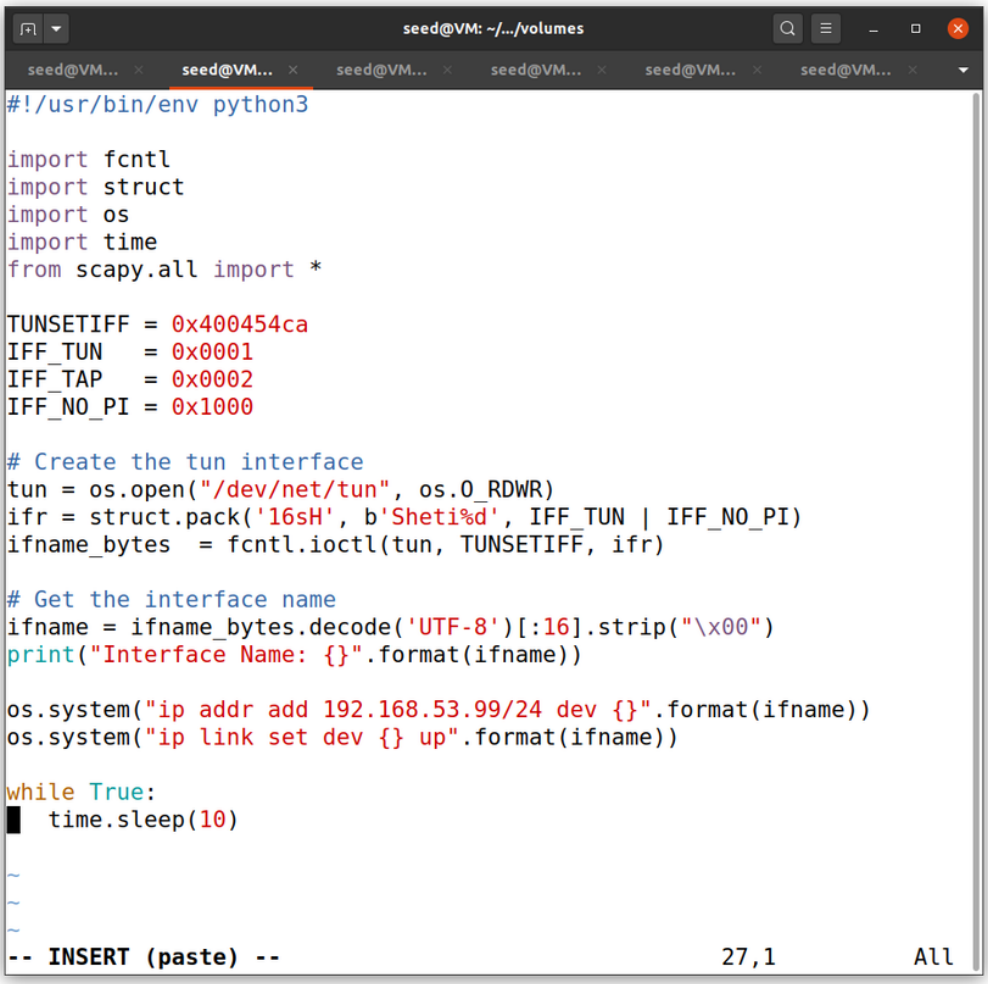
Assigning ip



Bringing it up:

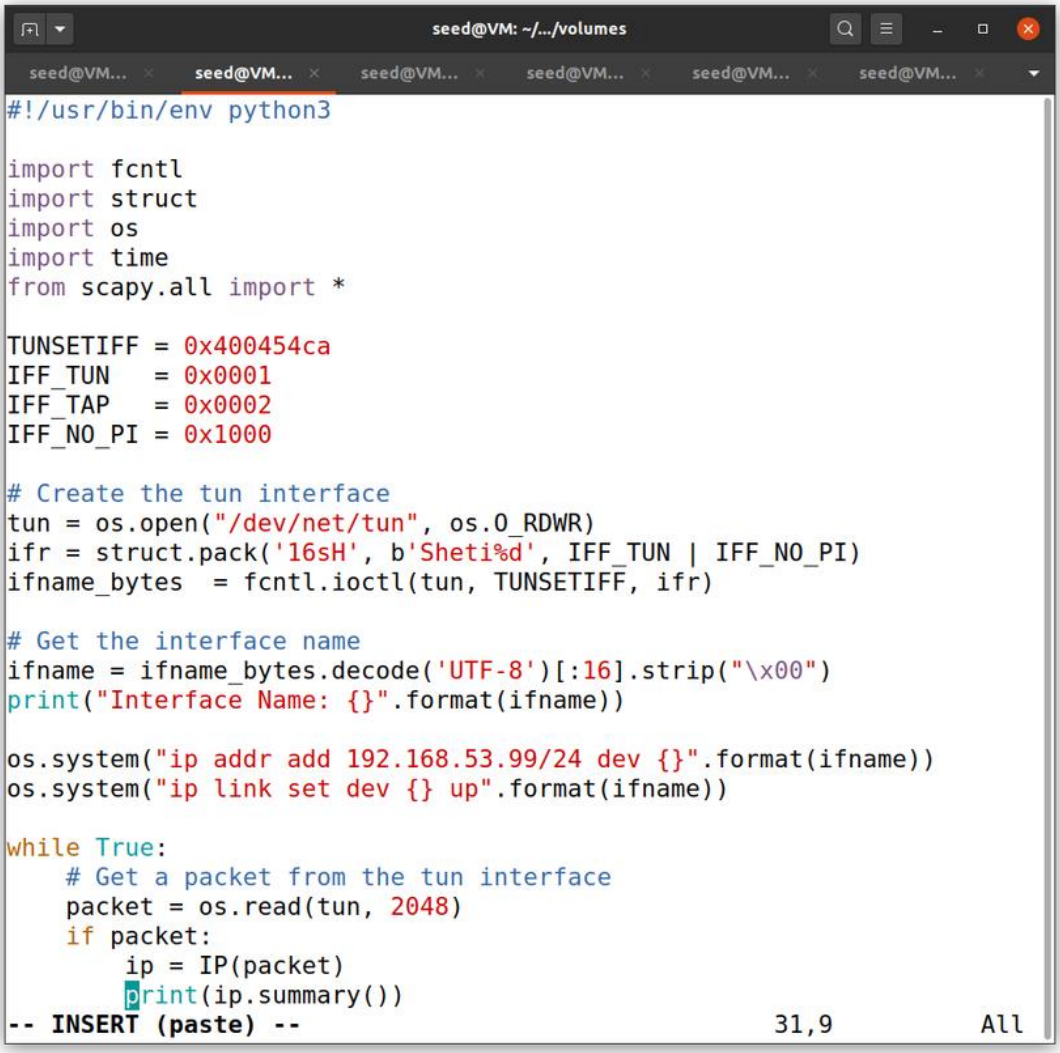

seed@VM: ~/.../Labsetup
seed@VM...
seed@VM...
seed@VM...
seed@VM...
seed@VM...
seed@VM...
root@eba6ecd7978f:/# ip link set dev Sheti0 up
root@eba6ecd7978f:/# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN gr
oup default qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
4: Sheti0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq
_codel state UNKNOWN group default qlen 500
link/none
inet 192.168.53.99/24 scope global Sheti0
valid_lft forever preferred_lft forever
161: eth0@if162: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqu
eue state UP group default
link/ether 02:42:0a:09:00:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0
valid_lft forever preferred_lft forever
root@eba6ecd7978f:/#

Code:

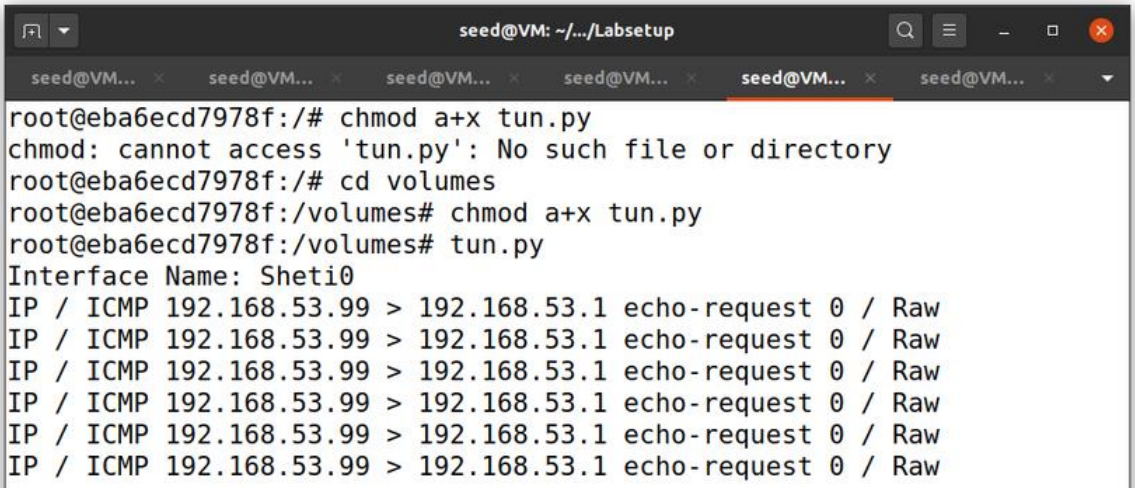

seed@VM: ~/.../volumes
seed@VM...
seed@VM...
seed@VM...
seed@VM...
seed@VM...
seed@VM...
#!/usr/bin/env python3
import fcntl
import struct
import os
import time
from scapy.all import *
TUNSETIFF = 0x400454ca
IFF_TUN = 0x0001
IFF_TAP = 0x0002
IFF_NO_PI = 0x1000
# Create the tun interface
tun = os.open("/dev/net/tun", os.O_RDWR)
ifr = struct.pack('16sH', b'Sheti%d', IFF_TUN | IFF_NO_PI)
ifname_bytes = fcntl.ioctl(tun, TUNSETIFF, ifr)
# Get the interface name
ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
print("Interface Name: {}".format(ifname))
os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
os.system("ip link set dev {} up".format(ifname))
while True:
time.sleep(10)
-- INSERT (paste) --
27,1
All

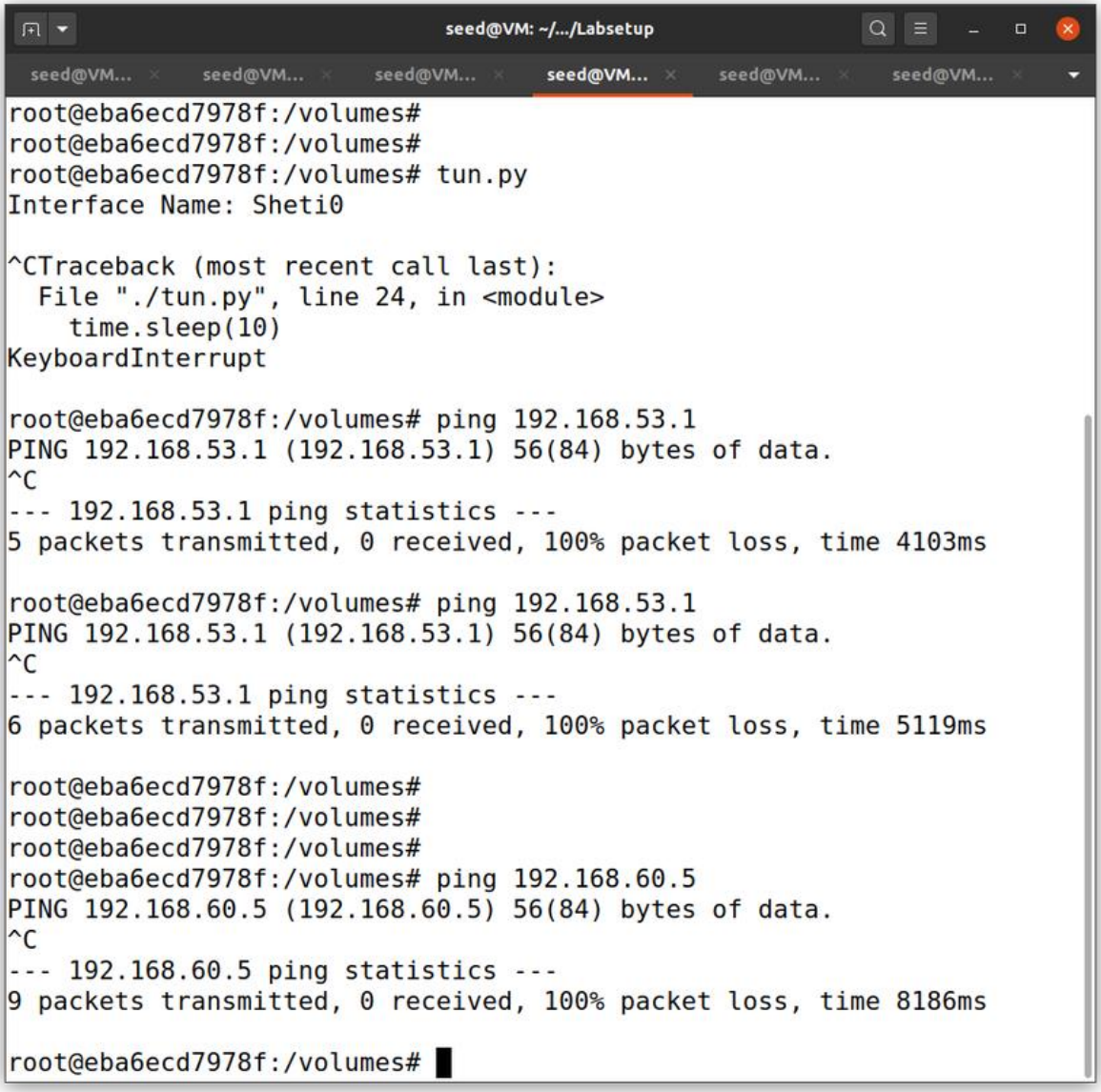
2 c)

Code:



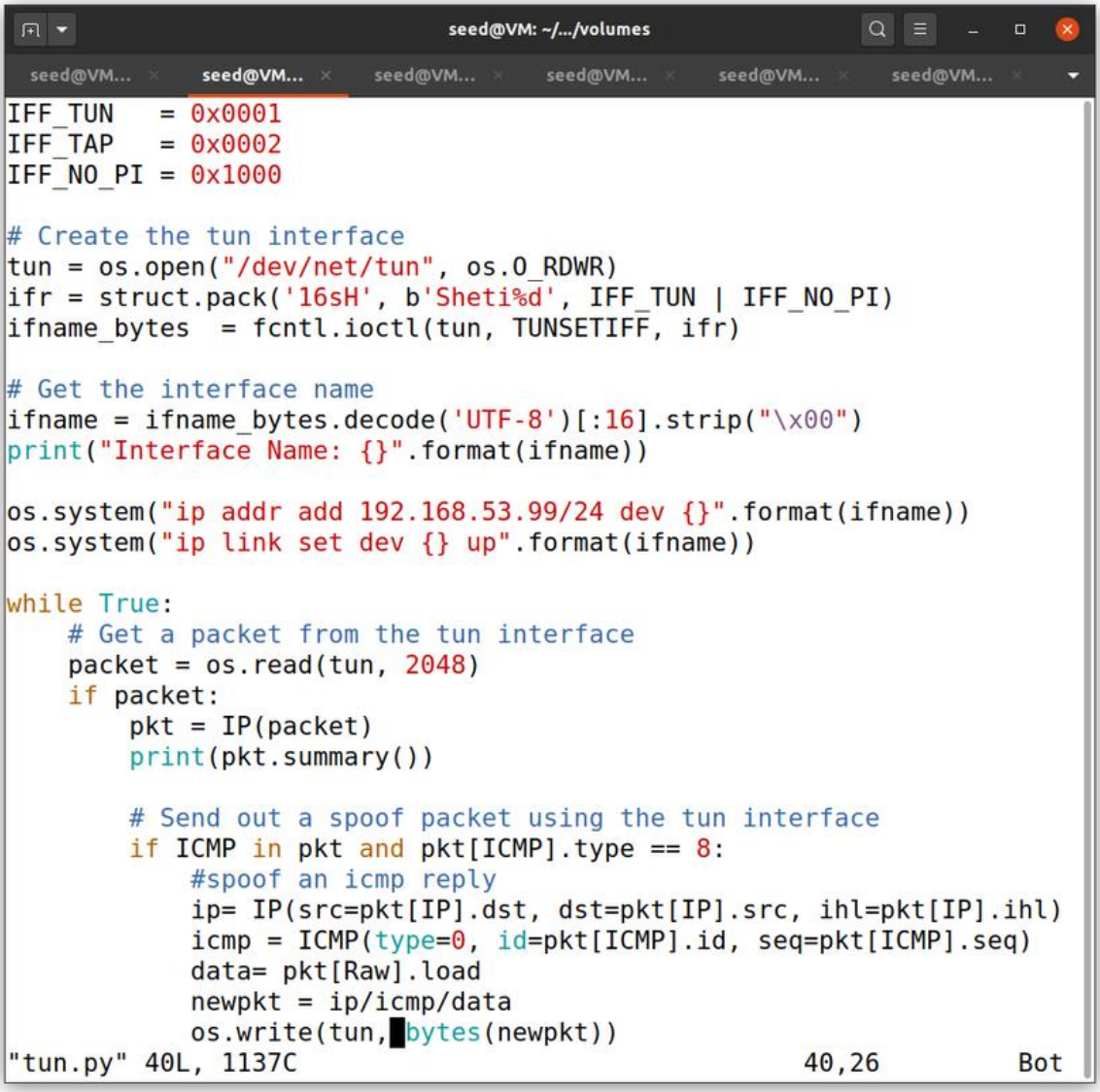
External network:



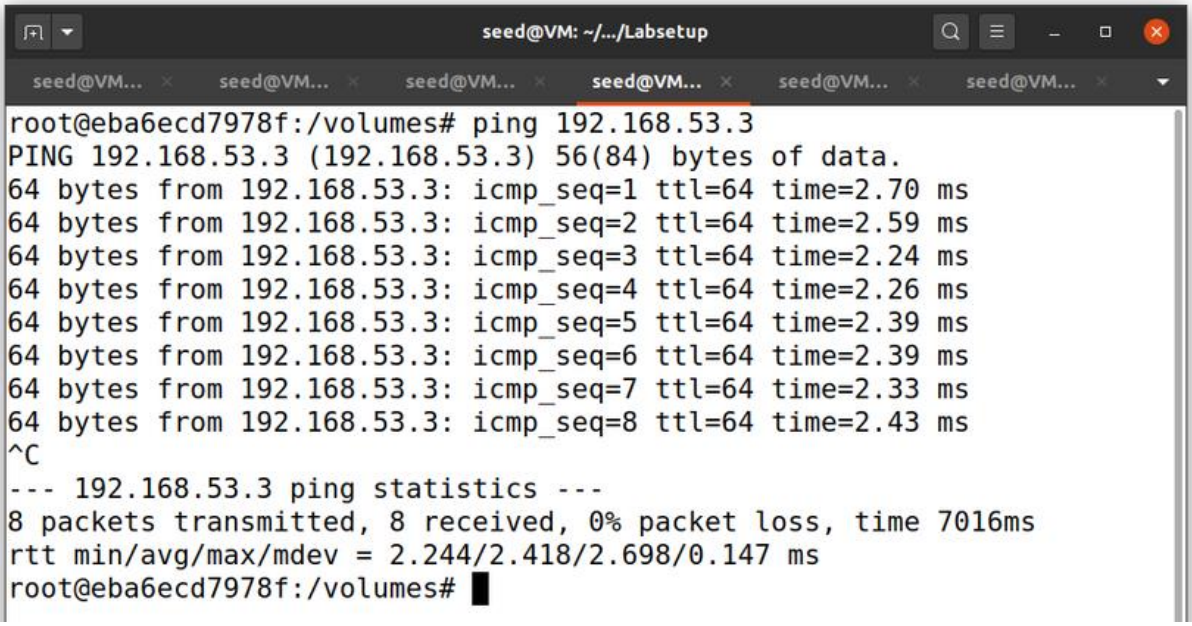

seed@VM: ~/.../Labsetup
seed@VM...
seed@VM...
seed@VM...
seed@VM...
seed@VM...
seed@VM...
root@eba6ecd7978f:/volumes#
root@eba6ecd7978f:/volumes#
root@eba6ecd7978f:/volumes# tun.py
Interface Name: Sheti0
^CTraceback (most recent call last):
File "./tun.py", line 24, in <module>
time.sleep(10)
KeyboardInterrupt
root@eba6ecd7978f:/volumes# ping 192.168.53.1
PING 192.168.53.1 (192.168.53.1) 56(84) bytes of data.
^C
--- 192.168.53.1 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4103ms
root@eba6ecd7978f:/volumes# ping 192.168.53.1
PING 192.168.53.1 (192.168.53.1) 56(84) bytes of data.
^C
--- 192.168.53.1 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5119ms
root@eba6ecd7978f:/volumes#
root@eba6ecd7978f:/volumes#
root@eba6ecd7978f:/volumes#
root@eba6ecd7978f:/volumes# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
^C
--- 192.168.60.5 ping statistics ---
9 packets transmitted, 0 received, 100% packet loss, time 8186ms
root@eba6ecd7978f:/volumes#

2 d)

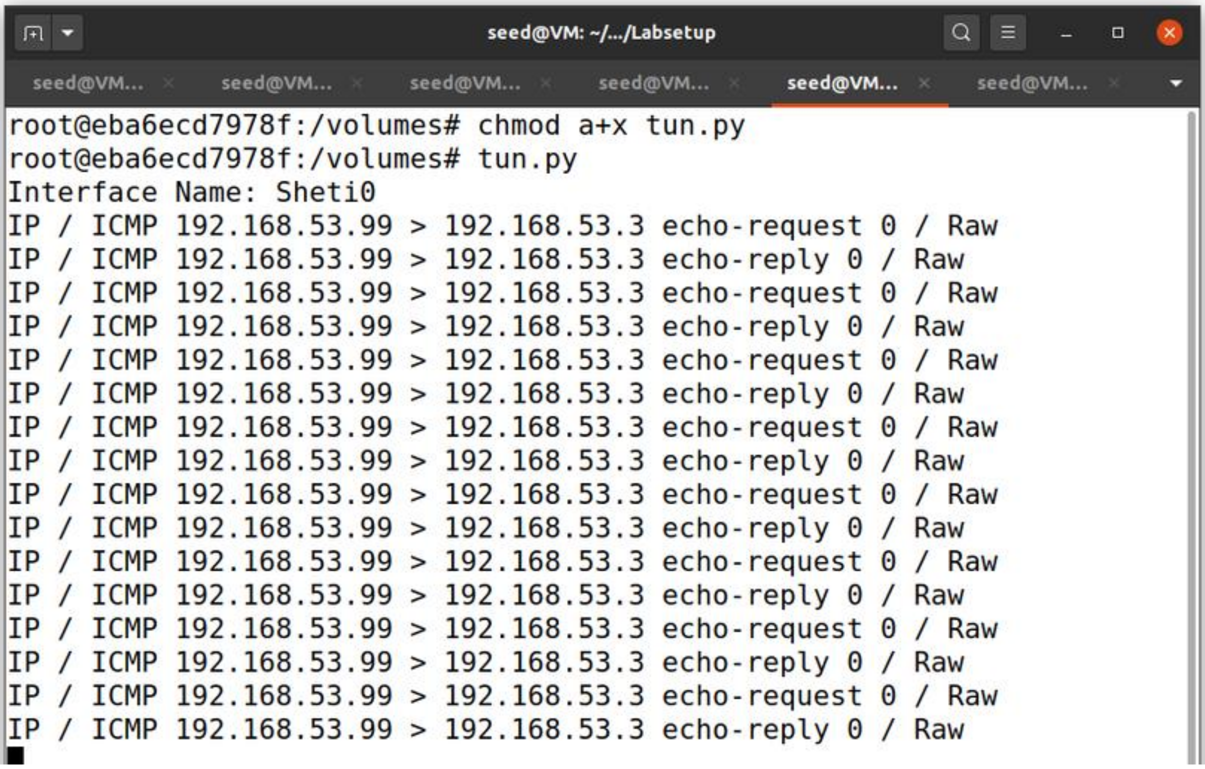
Code: Sending spoof packet



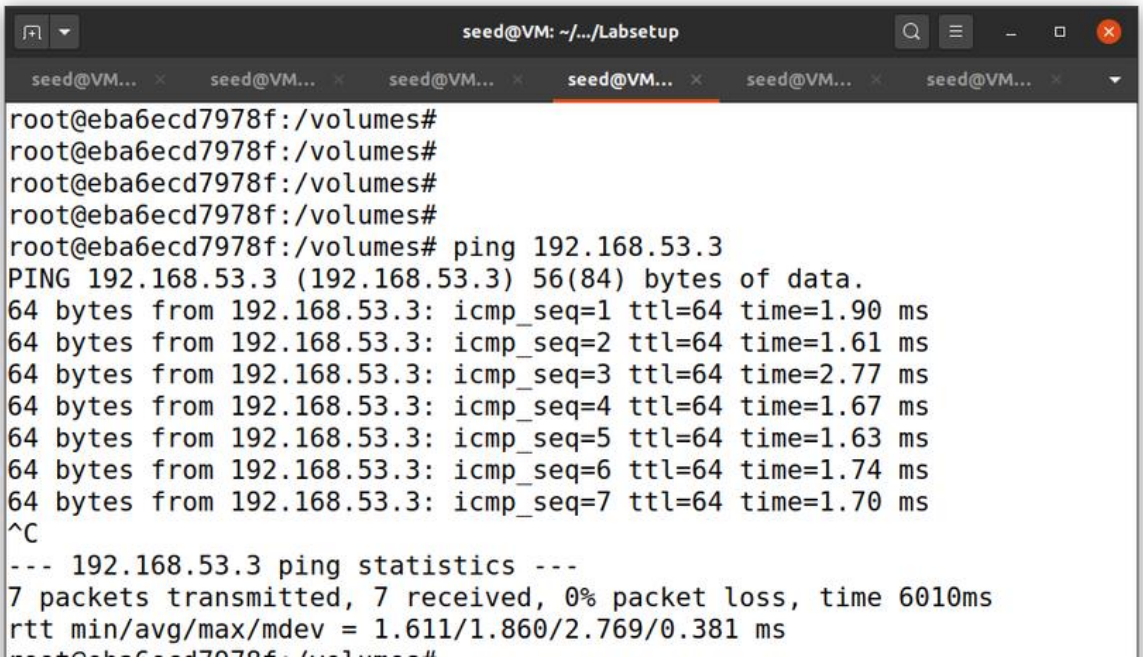
Pinging:



Code exec:

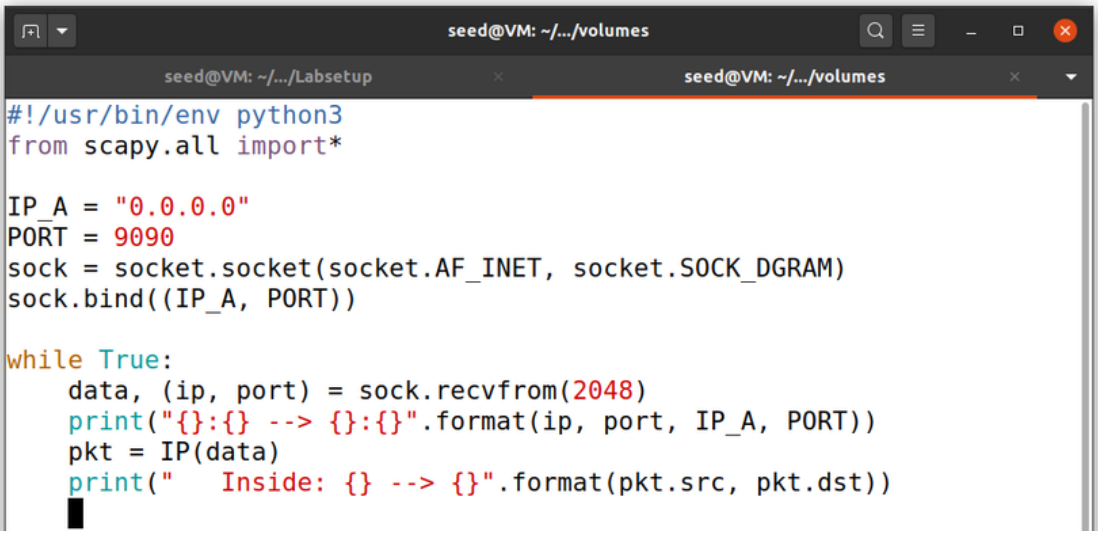


We can see that packets were sent and received:

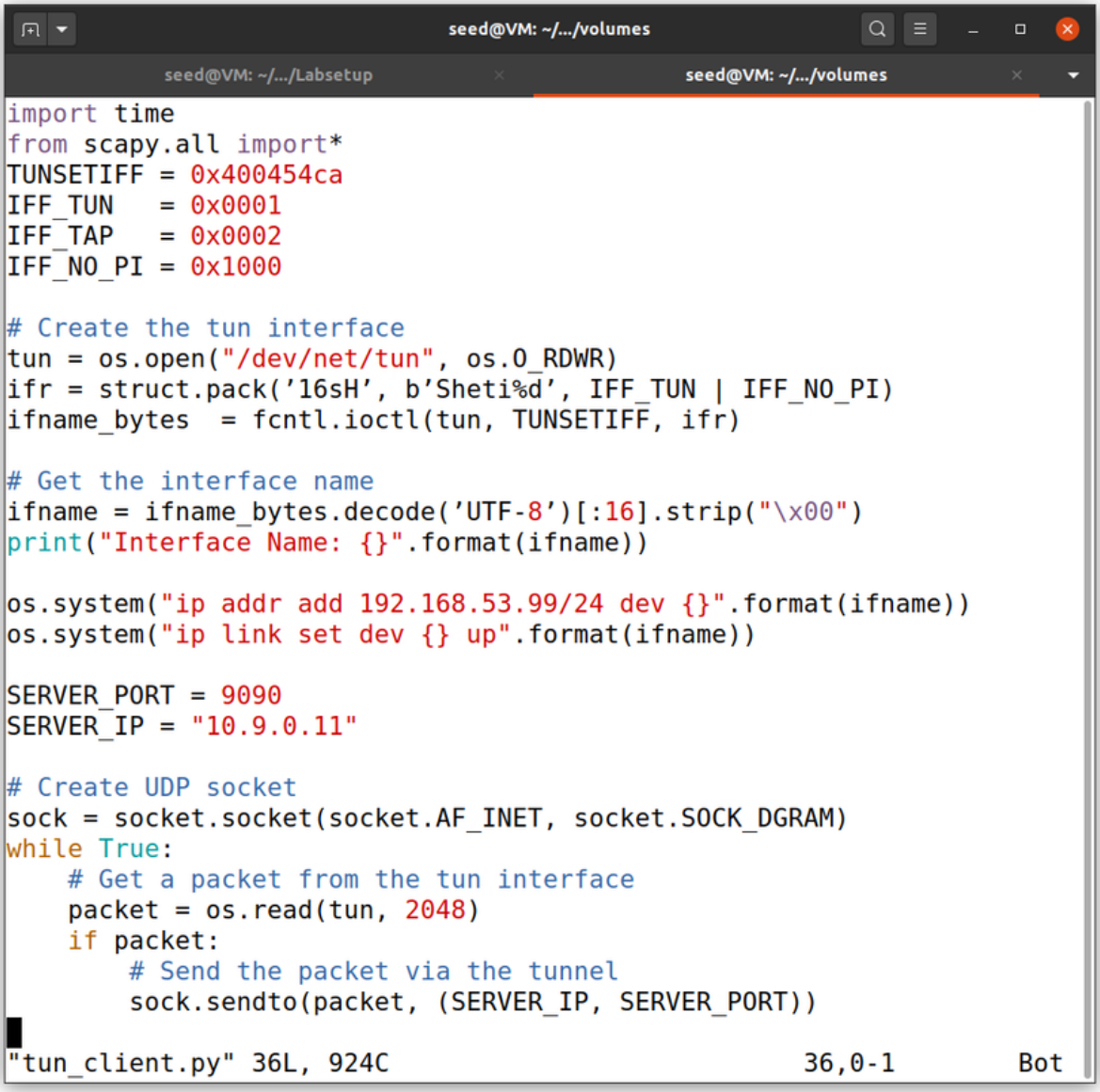


3)

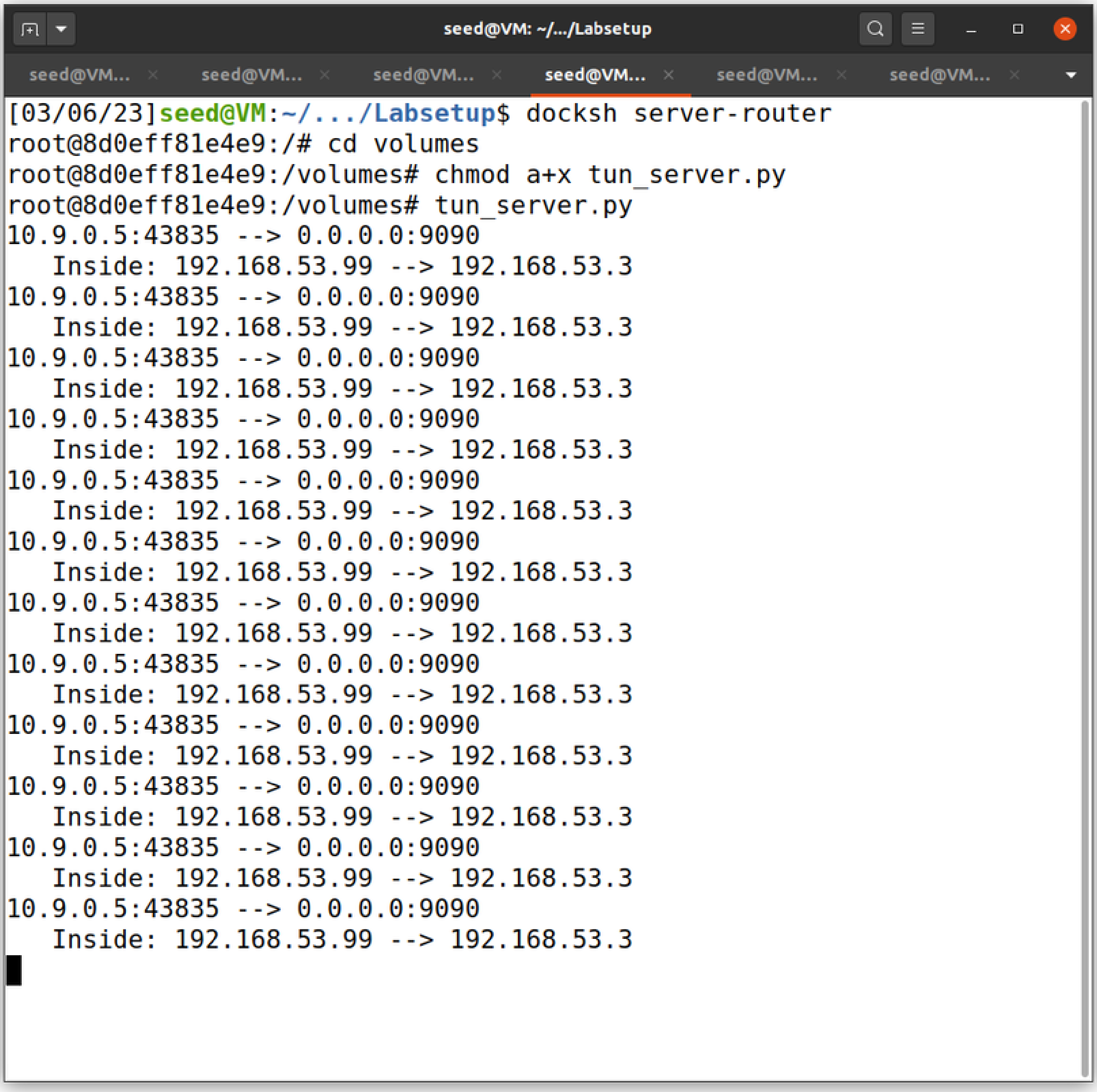
Server code:



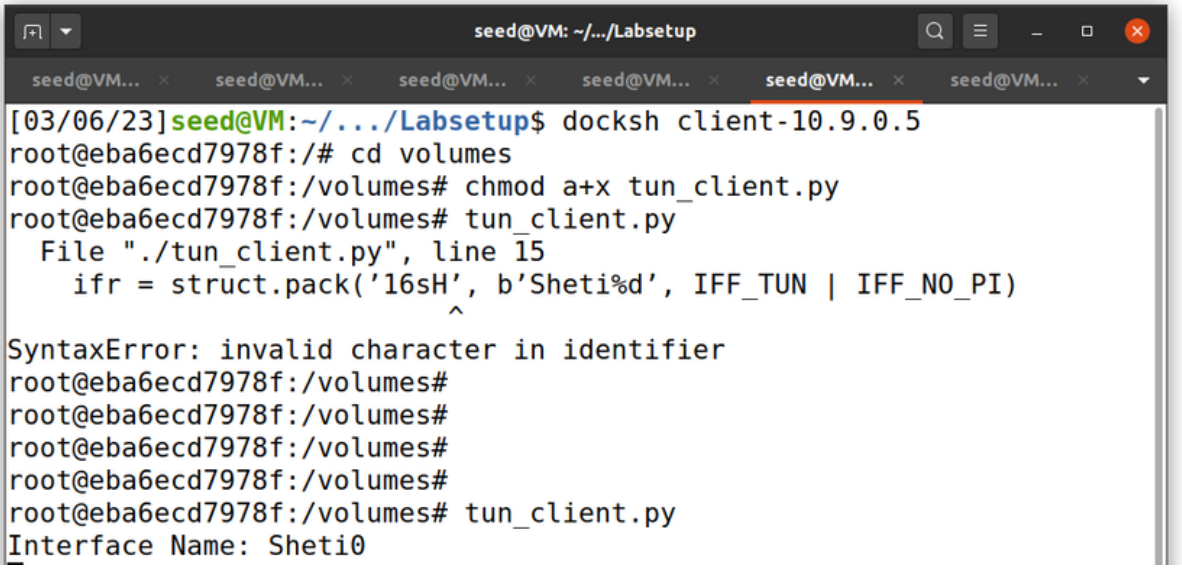
Client side:



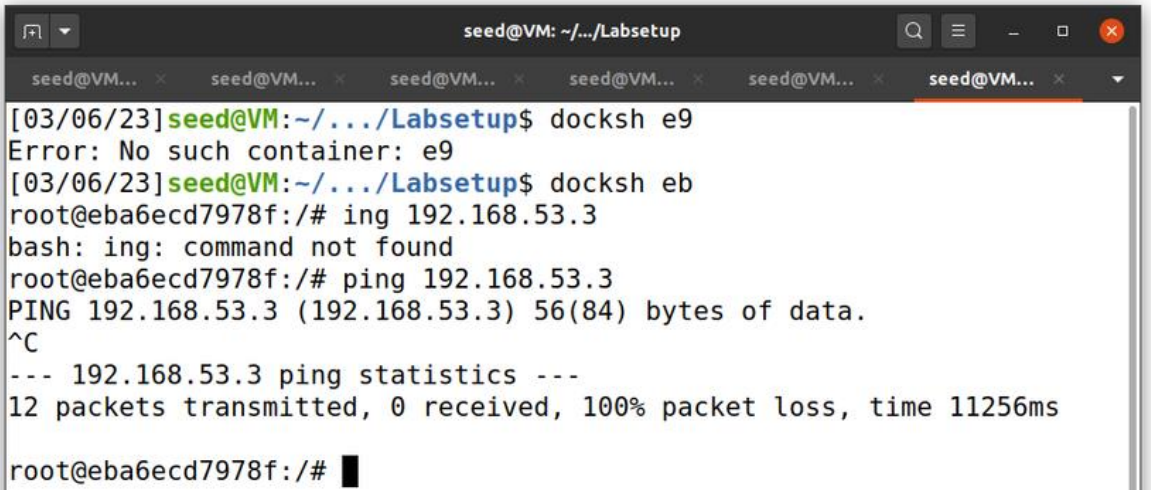
Server side output:



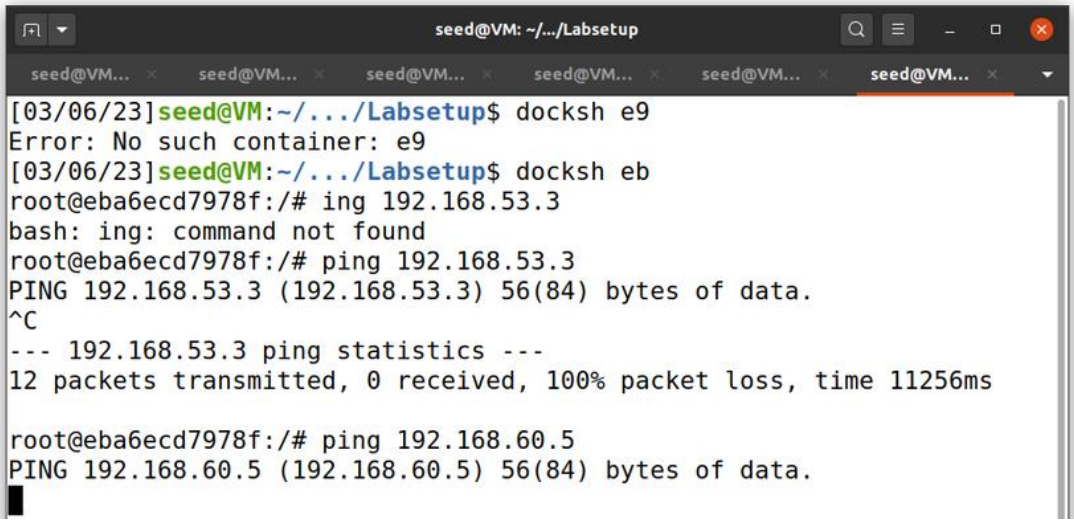
Client side ouput:



Pinging interface:



Pinging in another network:



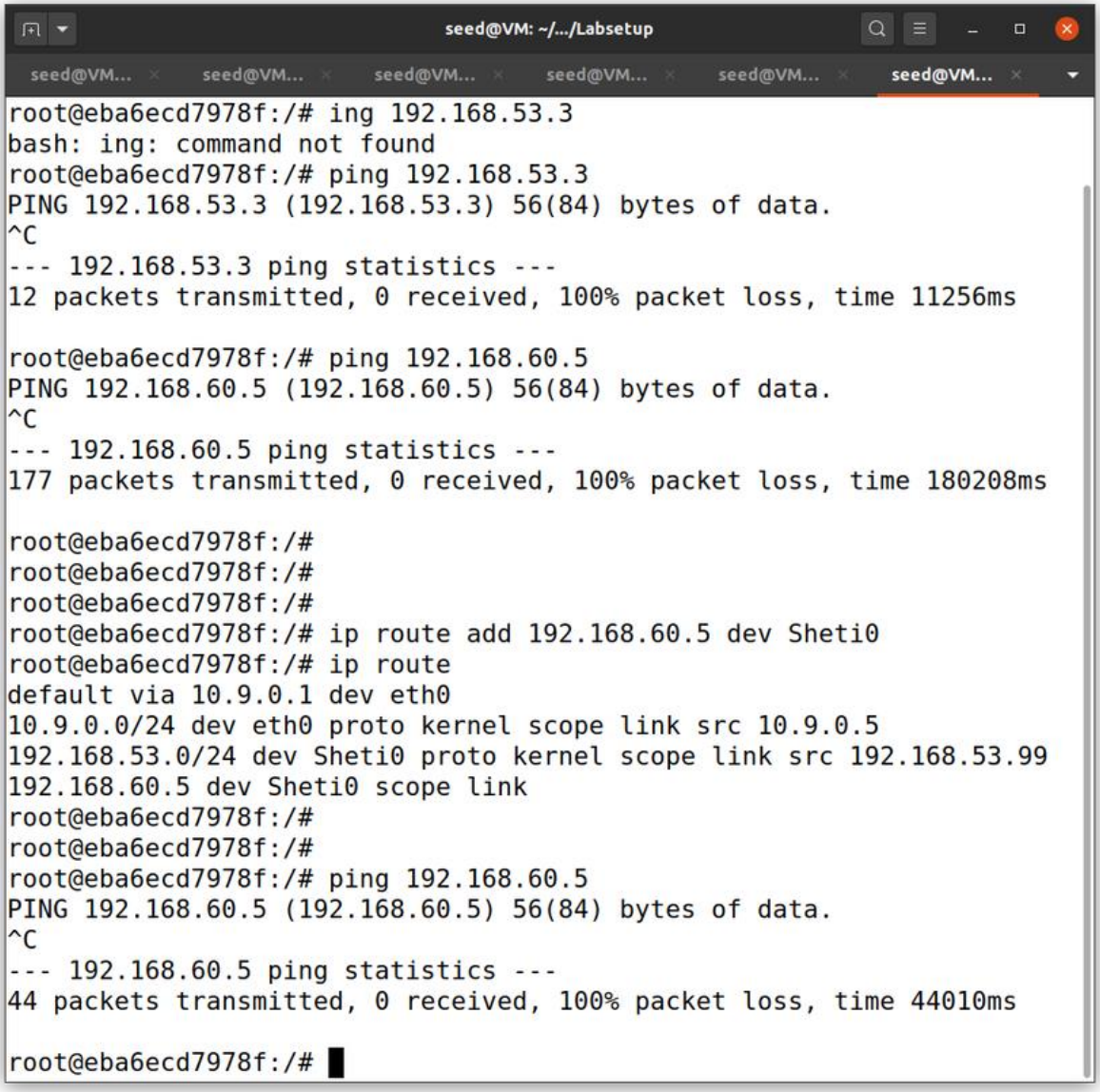
No response outputs:

```
seed@VM: ~/.../Labsetup
10.9.0.5:43835 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.53.3
10.9.0.5:43835 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.53.3
10.9.0.5:43835 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.53.3
10.9.0.5:43835 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.53.3
10.9.0.5:43835 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.53.3
10.9.0.5:43835 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.53.3
10.9.0.5:43835 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.53.3
10.9.0.5:43835 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.53.3
10.9.0.5:43835 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.53.3
10.9.0.5:43835 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.53.3
^CTraceback (most recent call last):
  File "./tun_server.py", line 10, in <module>
    data, (ip, port) = sock.recvfrom(2048)
KeyboardInterrupt

root@8d0eff81e4e9:/volumes#
root@8d0eff81e4e9:/volumes#
root@8d0eff81e4e9:/volumes#
root@8d0eff81e4e9:/volumes#
root@8d0eff81e4e9:/volumes#
root@8d0eff81e4e9:/volumes# tun_server.py
```

```
root@8d0eff81e4e9:/volumes#
root@8d0eff81e4e9:/volumes#
root@8d0eff81e4e9:/volumes#
root@8d0eff81e4e9:/volumes#
root@8d0eff81e4e9:/volumes#
root@8d0eff81e4e9:/volumes# tun_server.py
```

Adding ip route:



We can see once we added the route, it is responsive:

seed@VM: ~/.../Labsetup

```
  File "./tun_server.py", line 10, in <module>
    data, (ip, port) = sock.recvfrom(2048)
KeyboardInterrupt

root@8d0eff81e4e9:/volumes#
root@8d0eff81e4e9:/volumes#
root@8d0eff81e4e9:/volumes#
root@8d0eff81e4e9:/volumes#
root@8d0eff81e4e9:/volumes#
root@8d0eff81e4e9:/volumes# tun_server.py
10.9.0.5:37681 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:37681 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:37681 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:37681 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:37681 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:37681 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:37681 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:37681 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.60.5
^CTraceback (most recent call last):
  File "./tun_server.py", line 10, in <module>
    data, (ip, port) = sock.recvfrom(2048)
KeyboardInterrupt

root@8d0eff81e4e9:/volumes#
```

Task 4:

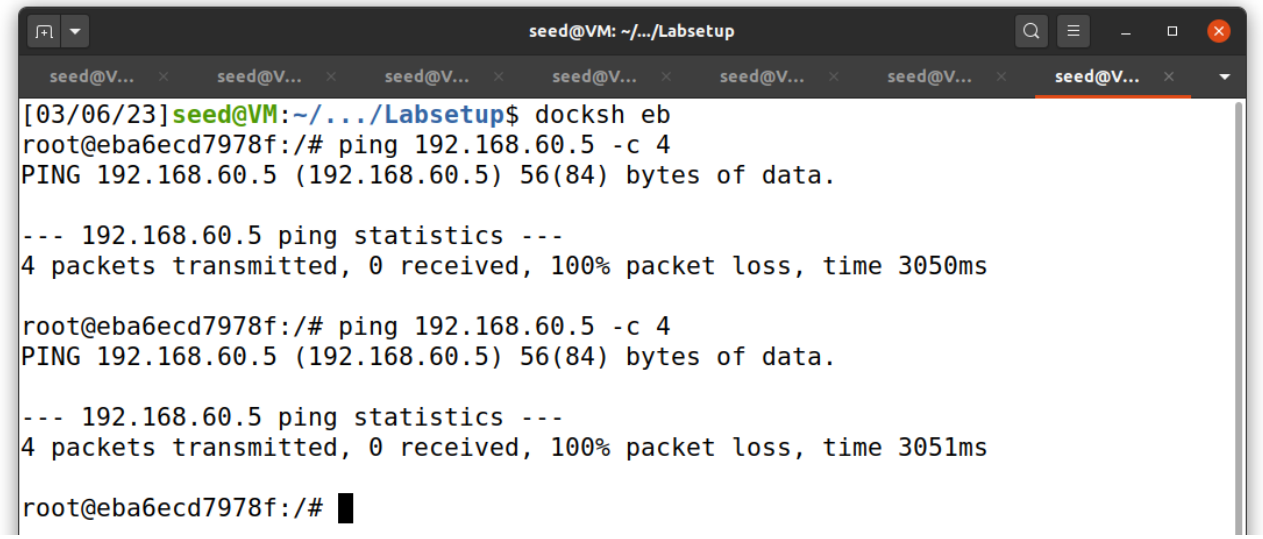
Server side :

```
[03/06/23]seed@VM:~/.../volumes$ vi tun.py
[03/06/23]seed@VM:~/.../volumes$ cd ../
[03/06/23]seed@VM:~/.../Labsetup$ docksh server-router
root@8d0eff81e4e9:/# cd volumes
root@8d0eff81e4e9:/volumes# chmod a+x tun_server.py
root@8d0eff81e4e9:/volumes# tun_server.py
Interface Name: Sheti0
10.9.0.5:37283 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:37283 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:37283 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:37283 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:37283 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:37283 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:37283 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.60.5
10.9.0.5:37283 --> 0.0.0.0:9090
   Inside: 192.168.53.99 --> 192.168.60.5
```

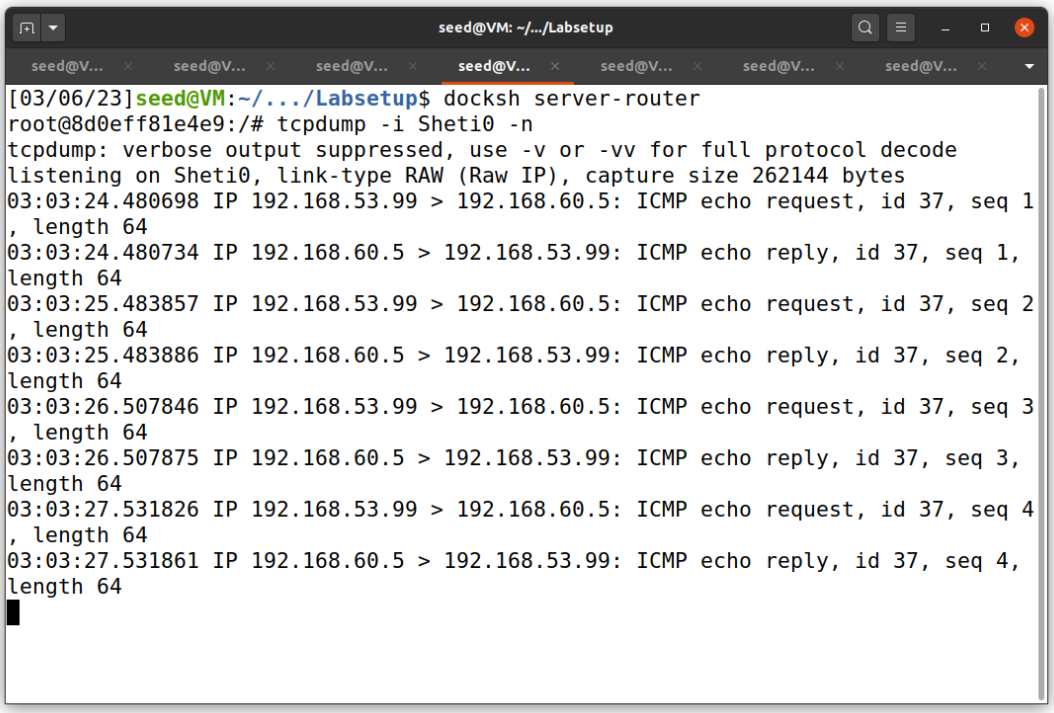
Client side:

```
[03/06/23]seed@VM:~/.../Labsetup$ docksh eb
root@eba6ecd7978f:/# cd volumes
root@eba6ecd7978f:/volumes# chmod a+x tun_client.py
root@eba6ecd7978f:/volumes# tun_client.py
Interface Name: Sheti0
```

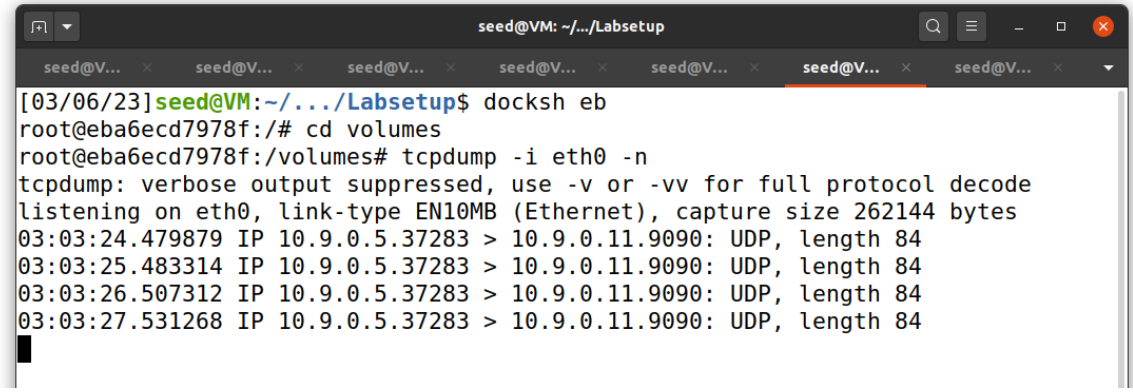
Pinging:



Tcp dump at server:



Tcp dump:



At host V:

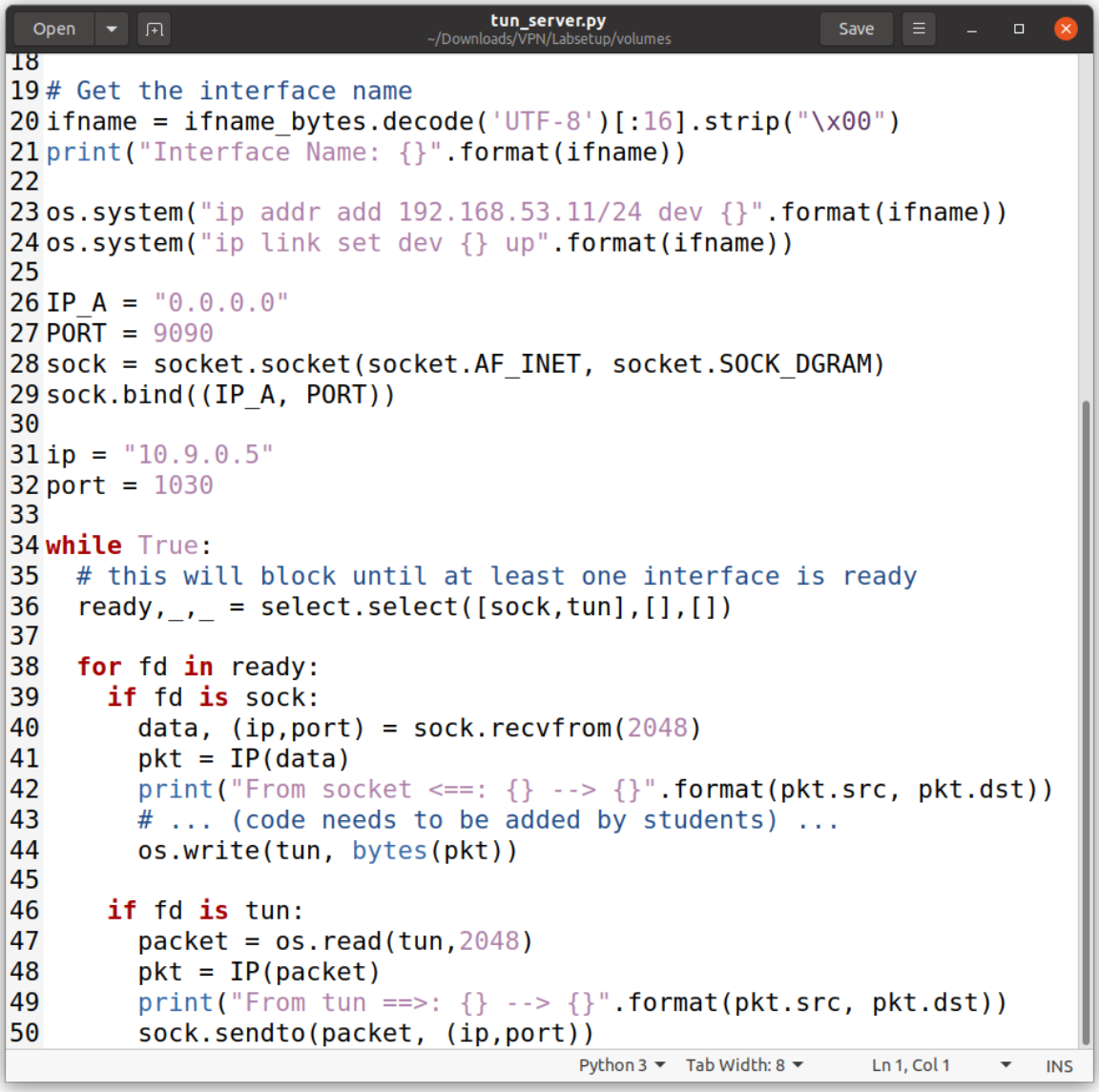
When tried to ping a private network, we can see that we are able to ping and get replies in the server but it is not showing on the host as the dual traffic hasn't been configured yet.

It can be seen that packets have arrived at host V

```
03:11:48.479401 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 38, seq 1
, length 64
03:11:48.479417 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 38, seq 1,
length 64
03:11:49.484560 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 38, seq 2
, length 64
03:11:49.484577 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 38, seq 2,
length 64
03:11:50.507828 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 38, seq 3
, length 64
03:11:50.507843 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 38, seq 3,
length 64
03:11:51.531841 IP 192.168.53.99 > 192.168.60.5: ICMP echo request, id 38, seq 4
, length 64
03:11:51.531856 IP 192.168.60.5 > 192.168.53.99: ICMP echo reply, id 38, seq 4,
length 64
03:11:53.483211 ARP, Request who-has 192.168.60.11 tell 192.168.60.5, length 28
03:11:53.483587 ARP, Request who-has 192.168.60.5 tell 192.168.60.11, length 28
03:11:53.483601 ARP, Reply 192.168.60.5 is-at 02:42:c0:a8:3c:05, length 28
03:11:53.483604 ARP, Reply 192.168.60.11 is-at 02:42:c0:a8:3c:0b, length 28
```

Task 5:

CODES:


Open
tun_server.py
~/Downloads/VPN/Labsetup/volumes
Save
18
19 # Get the interface name
20 ifname = ifname_bytes.decode('UTF-8')[:16].strip("\x00")
21 print("Interface Name: {}".format(ifname))
22
23 os.system("ip addr add 192.168.53.11/24 dev {}".format(ifname))
24 os.system("ip link set dev {} up".format(ifname))
25
26 IP_A = "0.0.0.0"
27 PORT = 9090
28 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
29 sock.bind((IP_A, PORT))
30
31 ip = "10.9.0.5"
32 port = 1030
33
34 while True:
35   # this will block until at least one interface is ready
36   ready,_,_ = select.select([sock,tun],[],[])
37
38   for fd in ready:
39   if fd is sock:
40   data, (ip,port) = sock.recvfrom(2048)
41   pkt = IP(data)
42   print("From socket <==: {} --> {}".format(pkt.src, pkt.dst))
43   # ... (code needs to be added by students) ...
44   os.write(tun, bytes(pkt))
45
46   if fd is tun:
47   packet = os.read(tun,2048)
48   pkt = IP(packet)
49   print("From tun ==>: {} --> {}".format(pkt.src, pkt.dst))
50   sock.sendto(packet, (ip,port))
Python 3
Tab Width: 8
Ln 1, Col 1
INS

tun_client.py
~/Downloads/VPN/Labsetup/volumes

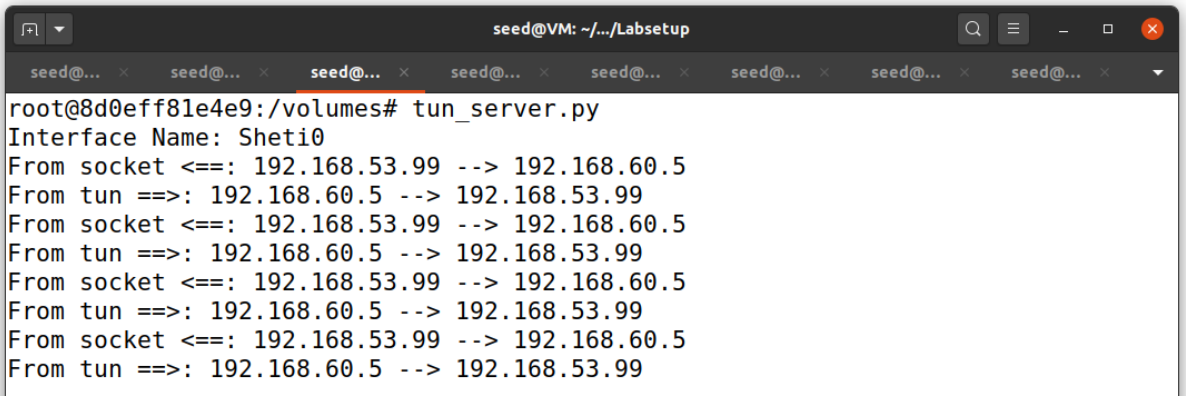
```
22
23 os.system("ip addr add 192.168.53.99/24 dev {}".format(ifname))
24 os.system("ip link set dev {} up".format(ifname))
25
26 SERVER_PORT = 9090
27 SERVER_IP = "10.9.0.11"
28
29 ip = "10.9.0.5"
30 port = 1030
31
32 os.system("ip route add 192.168.60.0/24 dev {}".format(ifname))
33
34 # Create UDP socket
35 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
36
37
38 while True:
39   # this will block until at least one interface is ready
40   ready,_,_ = select.select([sock,tun],[],[])
41
42   for fd in ready:
43     if fd is sock:
44       data, (ip,port) = sock.recvfrom(2048)
45       pkt = IP(data)
46       print("From socket ==>: {} --> {}".format(pkt.src, pkt.dst))
47       # ... (code needs to be added by students) ...
48       os.write(tun, bytes(pkt))
49
50     if fd is tun:
51       packet = os.read(tun,2048)
52       pkt = IP(packet)
53       print("From tun ==>: {} --> {}".format(pkt.src, pkt.dst))
54       sock.sendto(packet, (SERVER_IP, SERVER_PORT))
```

Client side:

seed@VM: ~/.../Labsetup

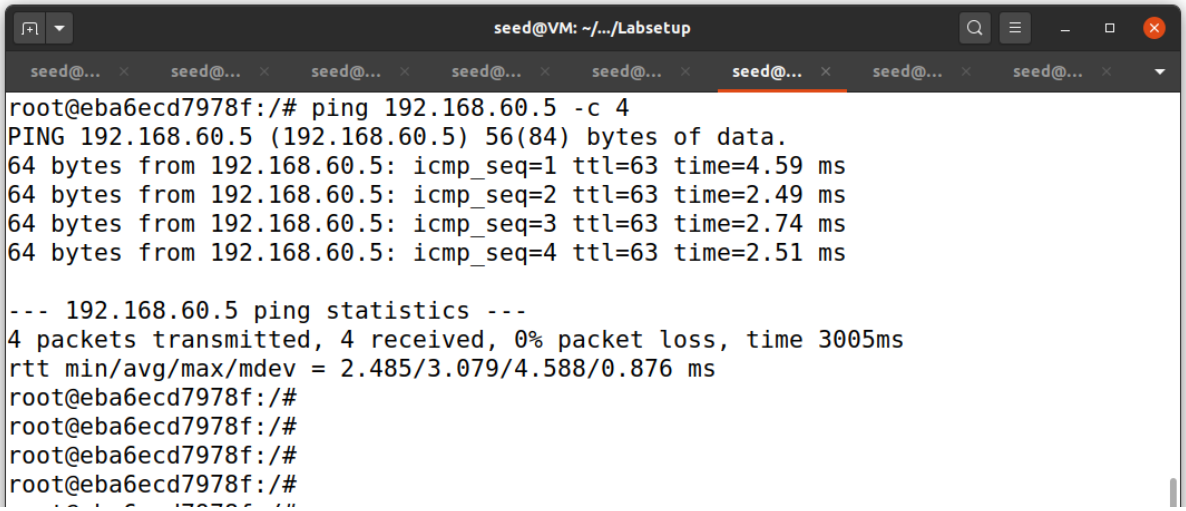
```
root@eba6ecd7978f:/volumes# tun_client.py
Interface Name: Sheti0
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
```

Server side:

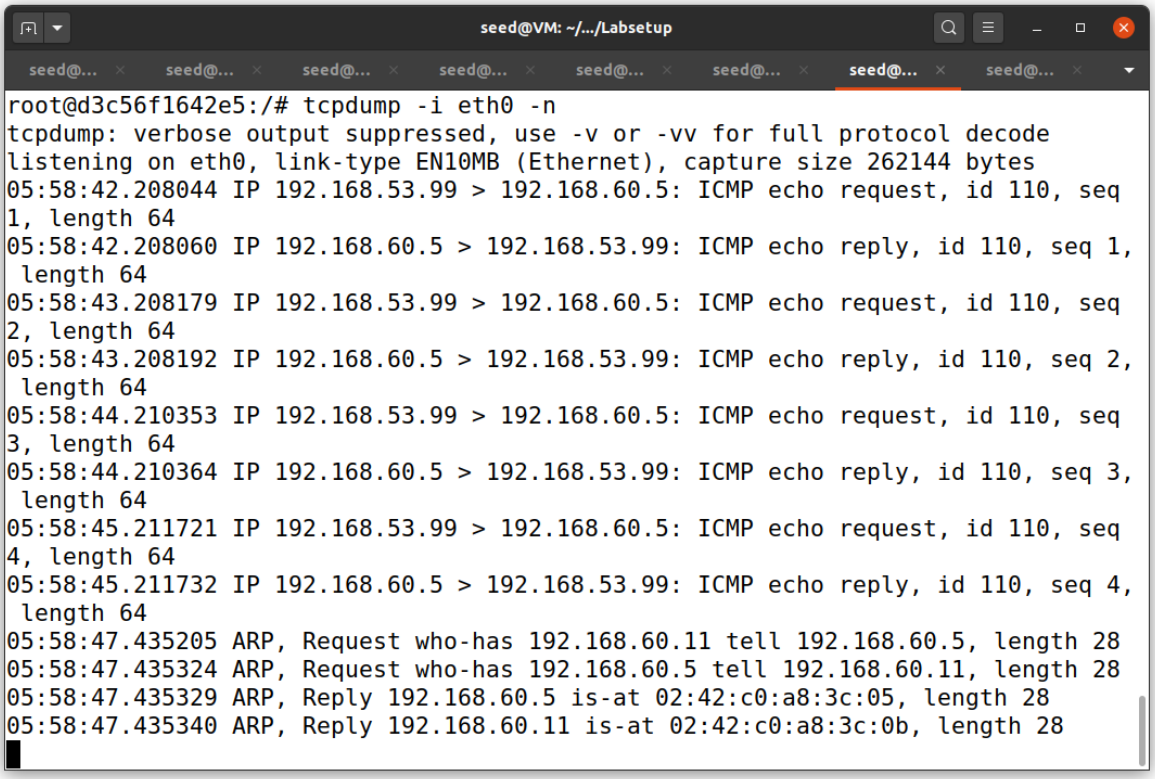


Pinging:

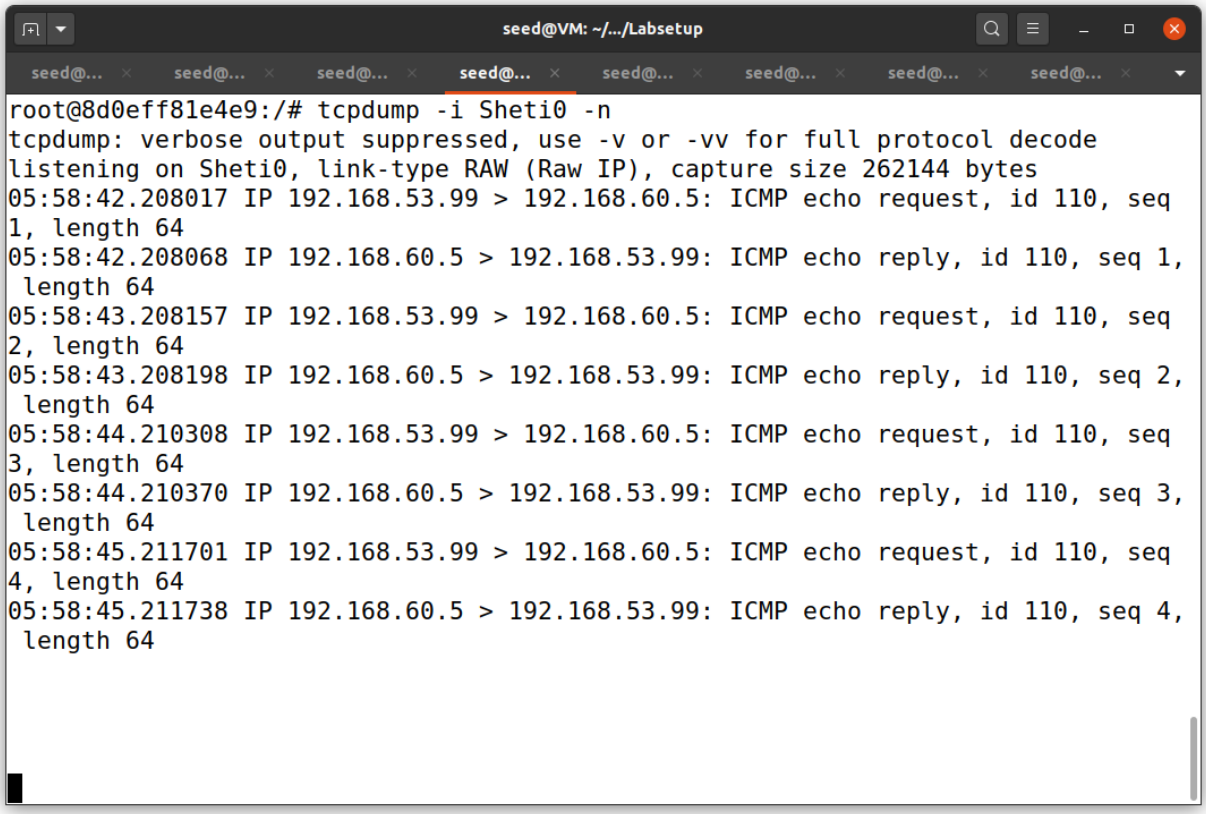
We can see that its successful as Host U and V are both getting requests and replies.



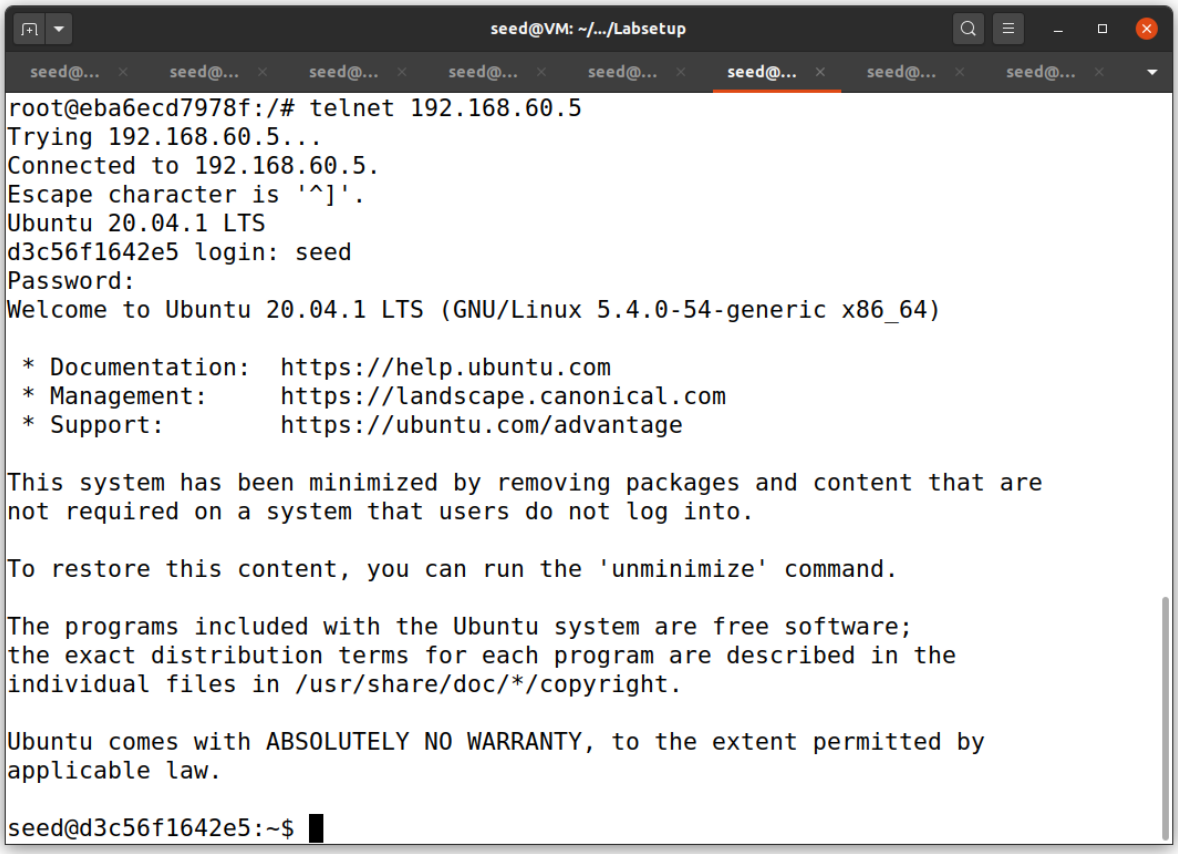
Tcp dump at server:



Tcp dump:



Establishing telnet connection:



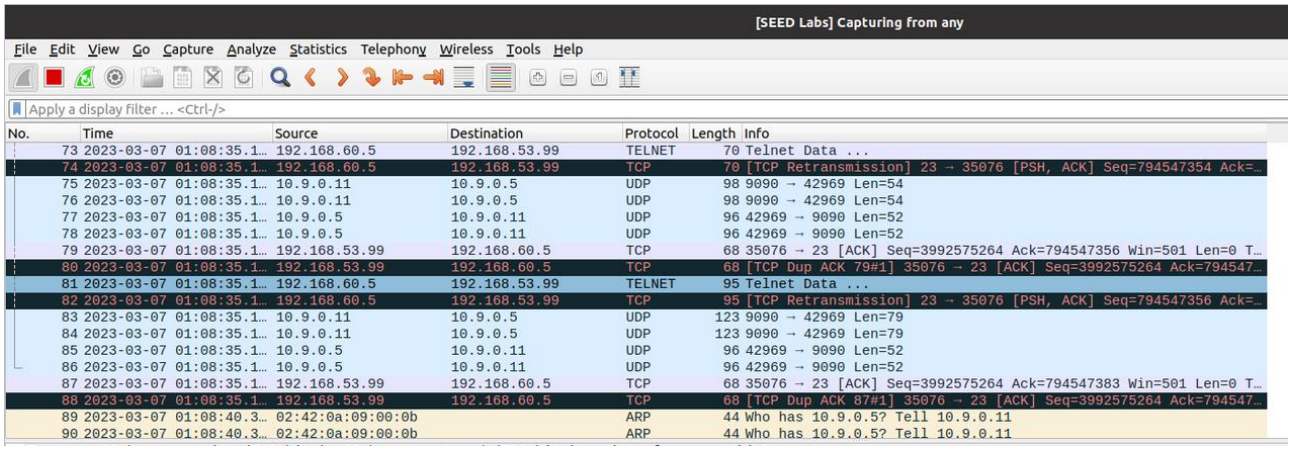
Now when I Run “who am I ” on the telnet connection, it gave me reply as “seed”. This was done multiple times and the result of this can be seen in this manner:

WE CAN SEE THAT IN THE SCREENSHOT OF TASK 6.

```
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
From socket ==>: 192.168.60.5 --> 192.168.53.99
From tun ==>: 192.168.53.99 --> 192.168.60.5
```

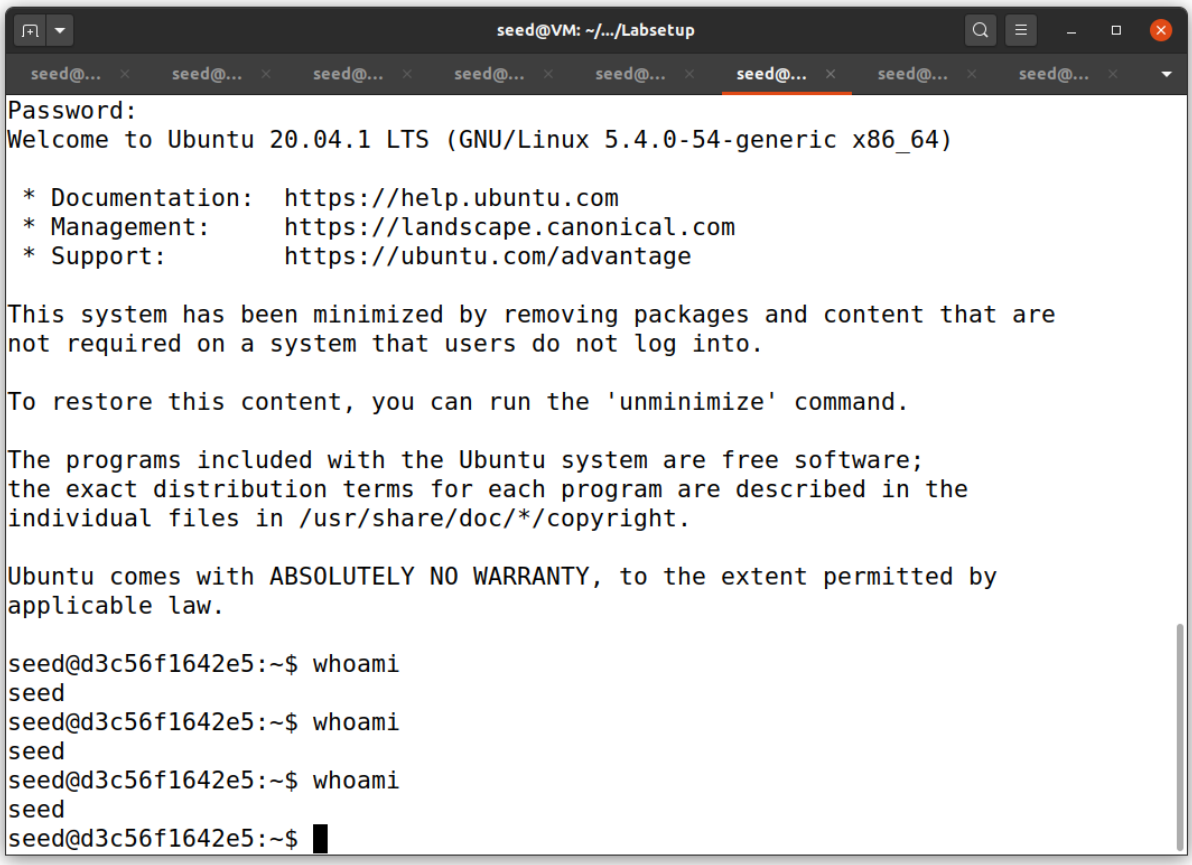
```
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
From tun ==>: 192.168.60.5 --> 192.168.53.99
From socket <==: 192.168.53.99 --> 192.168.60.5
```

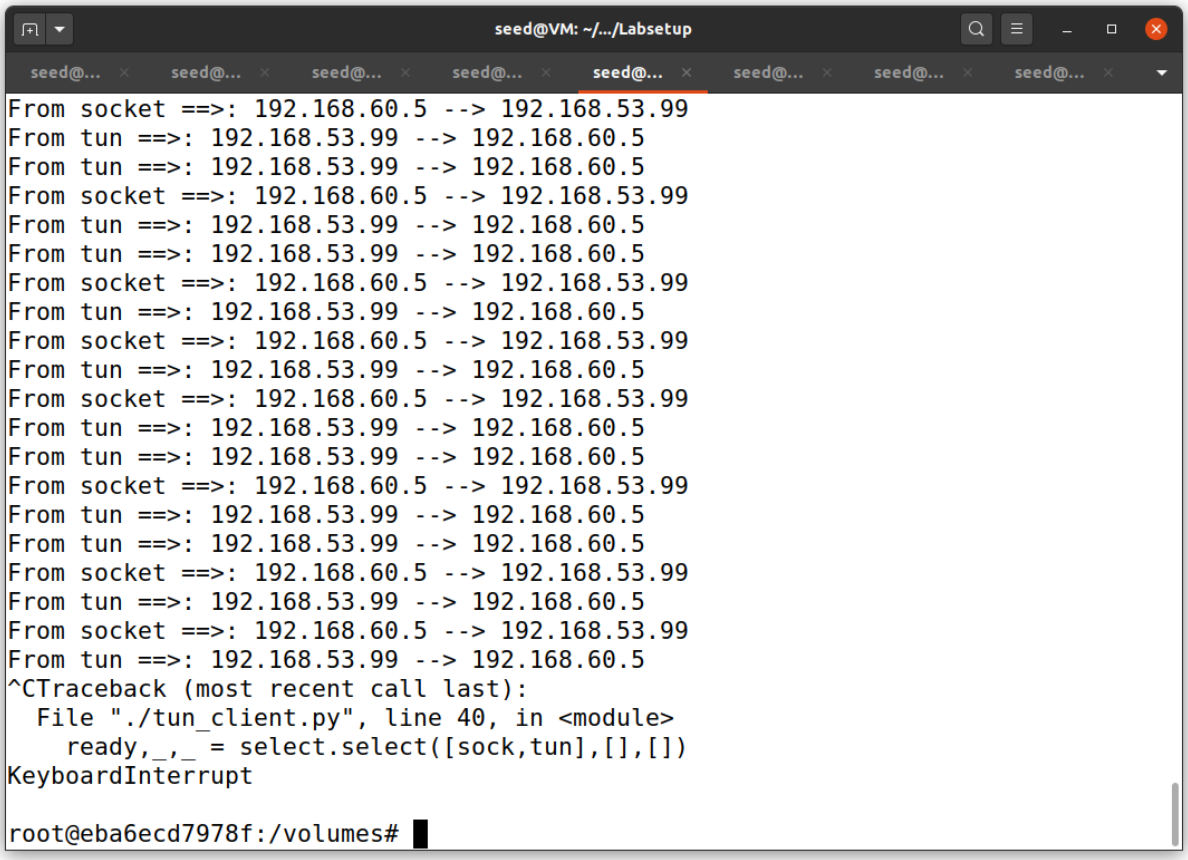
Wireshark proof



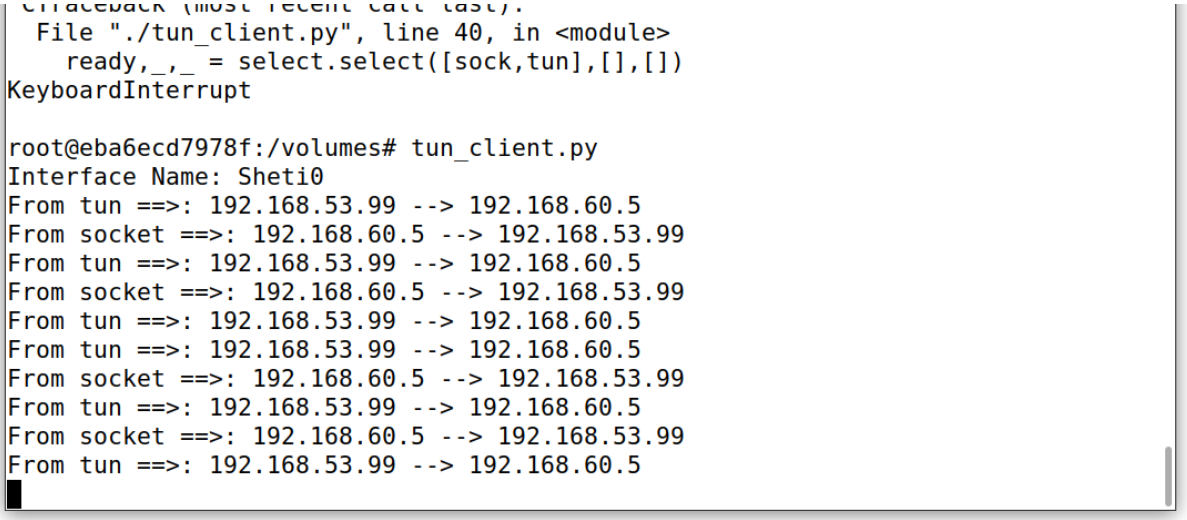
Task 6:

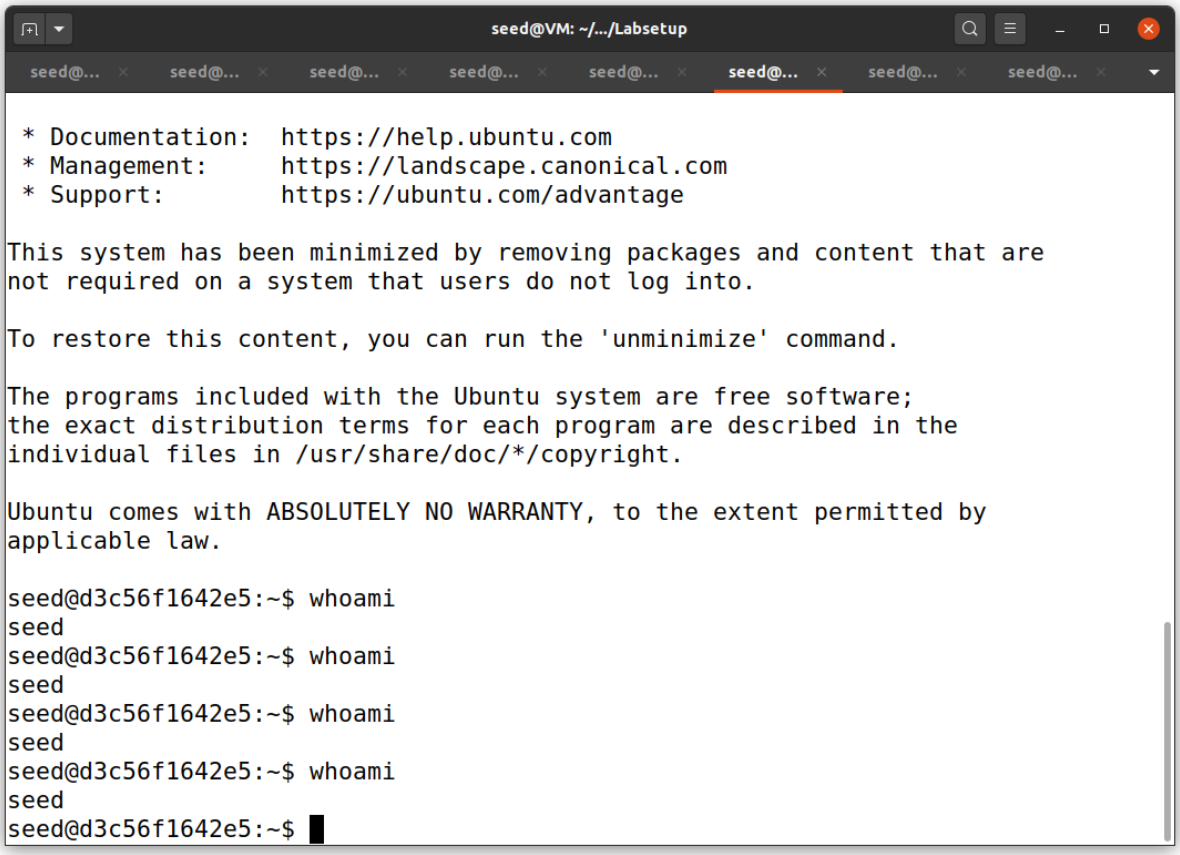
Connection freezes once I disconnect from telnet connection from one side:





Re-establishing connection:





After breaking the connection at the client side, we can observe that the telnet froze, then when we re-establish the side we can see we are able to type in the telnet.