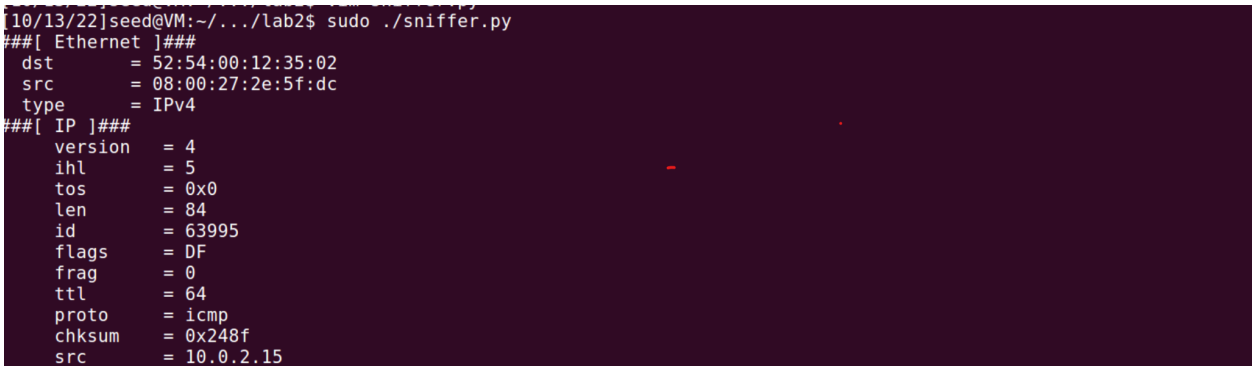
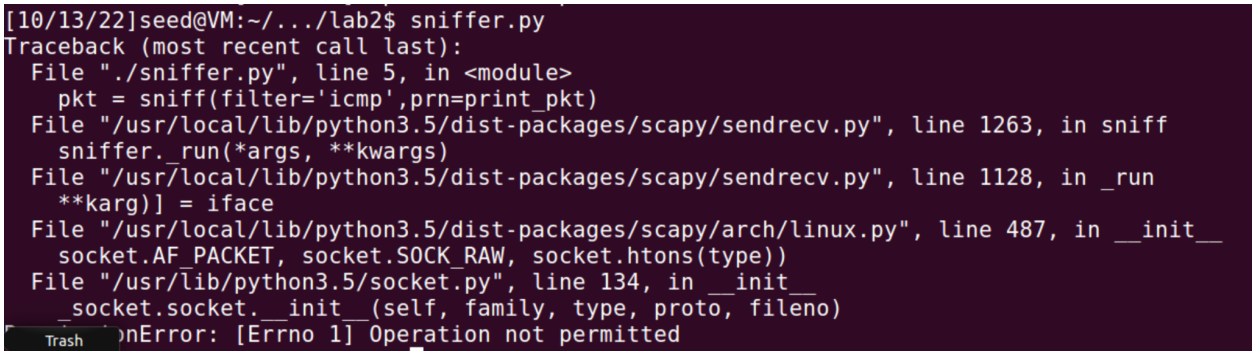
Name: Zhiyuan Ma, Wenting Zheng

1. Task 1.1: Sniffing Packets

   Ping 8.8.8.8 run with root



   Run without root(can't run without root)



   Task 1.1B

   Capture only the ICMP packet

```
#!/usr/bin/python3
from scapy.all import *
def print_pkt(pkt):
        pkt.show()
pkt = sniff(filter='icmp',prn=print_pkt)
```

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

```
#!/usr/bin/python3
from scapy.all import *
def print_pkt(pkt):
        pkt.show()
pkt = sniff(filter='dst port 23',prn=print_pkt)
```

Capture packets comes from or to go to a particular subnet. You can pick any subnet, such as 128.230.0.0/16; you should not pick the subnet that your VM is attached to.

```
#!/usr/bin/python3
from scapy.all import *
def print_pkt(pkt):
        pkt.show()
pkt = sniff(filter = "tcp and src='192.168.2.0/24'",prn=print_pkt)
```

2.2 Task 1.2: Spoofing ICMP Packets

We use sniffer to ping host machine:

```
[10/19/22]seed@VM:~/.../lab2$ sudo ./sniffer.py
Begin emission:
Finished sending 1 packets.
.*
Received 2 packets, got 1 answers, remaining 0 packets
[10/19/22]seed@VM:~/.../lab2$ cat sniffer.py
#!/usr/bin/python3
from scapy.all import *
def print_pkt(pkt):
        pkt.show()
# pkt = sniff(filter = "tcp and src='192.168.2.0/24'",prn=print_pkt)
sr1(IP(dst="192.168.56.1")/ICMP())
```

We could see the response on the wireshark of host machine

1	0.000000	PcsCompu_2e:5f:dc	Broadcast	ARP	60	Who has 192.168.56.1? Tell 192.168.56.101
2	0.000018	0a:00:27:00:00:11	PcsCompu_2e:5f:dc	ARP	42	192.168.56.1 is at 0a:00:27:00:00:11
3	0.019223	192.168.56.101	192.168.56.1	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (reply in 4)
4	0.019258	192.168.56.1	192.168.56.101	ICMP	42	Echo (ping) reply id=0x0000, seq=0/0, ttl=128 (request in 3)

2.3 Task 1.3: Traceroute

We print first 10 jumping spot of ping 8.8.8.8

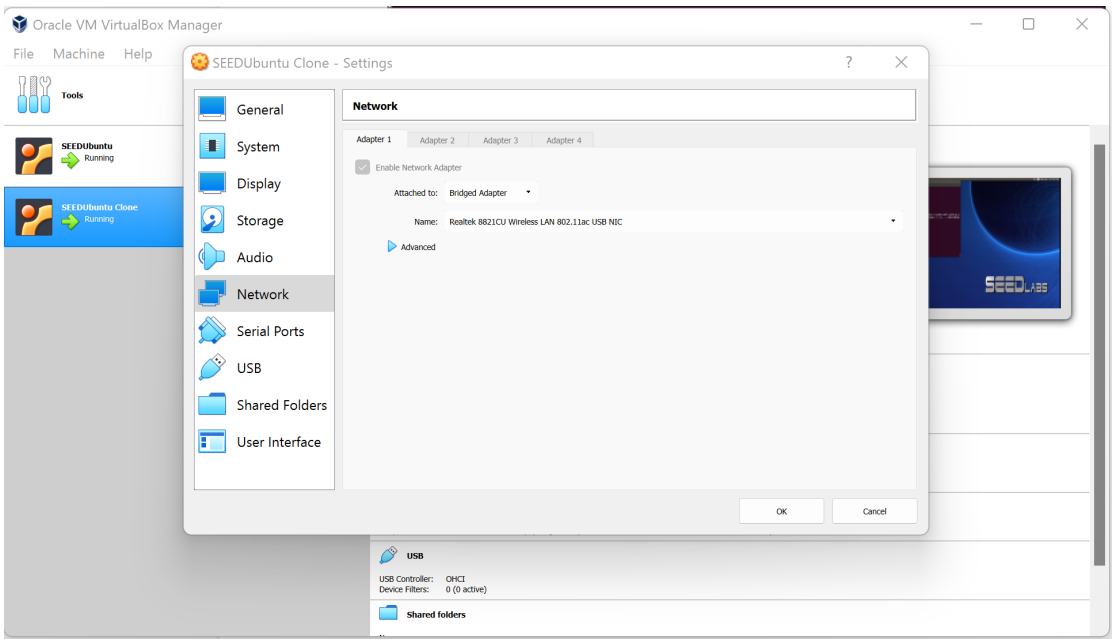
```
[10/19/22]seed@VM:~/.../lab2$ cat sniffer.py
#!/usr/bin/python3
from scapy.all import *
def print_pkt(pkt):
        pkt.show()
# pkt = sniff(filter = "tcp and src='192.168.2.0/24'",prn=print_pkt)
for x in range(10):
        resp =sr1(IP(ttl=x+1,dst="8.8.8.8")/ICMP())
        print(resp[0].summary())
```

The result is as follows:

```
.*
Received 2 packets, got 1 answers, remaining 0 packets
IP / ICMP 10.0.2.2 > 10.0.2.15 time-exceeded ttl-zero-during-transit / IPerror / ICMPerror
Begin emission:
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
IP / ICMP 169.234.0.1 > 10.0.2.15 time-exceeded ttl-zero-during-transit / IPerror / ICMPerror
Begin emission:
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
IP / ICMP 128.195.249.129 > 10.0.2.15 time-exceeded ttl-zero-during-transit / IPerror / ICMPerror
Begin emission:
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
IP / ICMP 10.255.0.46 > 10.0.2.15 time-exceeded ttl-zero-during-transit / IPerror / ICMPerror
Begin emission:
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
IP / ICMP 128.200.2.205 > 10.0.2.15 time-exceeded ttl-zero-during-transit / IPerror / ICMPerror
Begin emission:
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
IP / ICMP 128.200.2.222 > 10.0.2.15 time-exceeded ttl-zero-during-transit / IPerror / ICMPerror
Begin emission:
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
IP / ICMP 128.200.2.242 > 10.0.2.15 time-exceeded ttl-zero-during-transit / IPerror / ICMPerror
Begin emission:
```

2.4 Task 1.4: Sniffing and-then Spoofing

First we choose bridged adaptor options in two vm, then ping 8.8.8.8 in one vm and run sniffer.py in another vm



```
[10/20/22]seed@VM:~$ 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=58 time=6.84 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=58 time=8.79 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=58 time=12.5 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=58 time=9.46 ms
```

We could see the echo package in another vm using sniffer, by setting filter to icmp package only:

```
###[ Ethernet ]###
  dst       = 08:00:27:2e:5f:dc
  src       = 58:24:29:9a:2a:9e
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 0
     flags     =
     frag      = 0
     ttl       = 58
     proto     = icmp
     chksum    = 0x5926
     src       = 8.8.8.8
     dst       = 192.168.86.203
     \options   \
###[ ICMP ]###
        type      = echo-reply
```

Then, we would focus on identify the dst address of the ip layer:

```
[10/20/22]seed@VM:~/.../lab2$ sudo ./sniffer.py
8.8.8.8
192.168.86.203
8.8.8.8
192.168.86.203
8.8.8.8
192.168.86.203
8.8.8.8
192.168.86.203
8.8.8.8
192.168.86.203
8.8.8.8
192.168.86.203
8.8.8.8
192.168.86.203
```

```
^C[10/20/22]seed@VM:~/.../lab2$ cat sniffer.py
#!/usr/bin/python3
from scapy.all import *
def print_pkt(pkt):
        print(pkt[1].dst)
pkt = sniff(filter='icmp',prn=print_pkt)
[10/20/22]seed@VM:~/.../lab2$
```

So the logic is if we identify the package to some specific destination, in this case 8.8.8.8, we would use sniffer to send a reply to the source address. To do this, we ping 8.8.8.9, which won't have any echo reply, we would use sniffer to make the other vm believe there is a reply.

First, there is no reply:

```
[10/20/22]seed@VM:~/.../lab2$ sudo ./sniffer.py
8.8.8.9
8.8.8.9
8.8.8.9
8.8.8.9
8.8.8.9
8.8.8.9
8.8.8.9
8.8.8.9
^C[10/20/22]seed@VM:~/.../lab2$
```

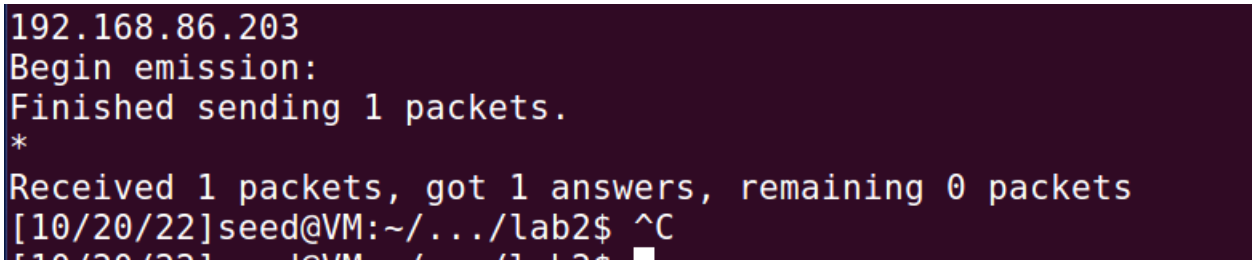
```
PING 8.8.8.9 (8.8.8.9) 56(84) bytes of data.
^C
--- 8.8.8.9 ping statistics ---
12 packets transmitted, 0 received, 100% packet loss, time 11248ms

[10/20/22]seed@VM:~$
```

The code is as

```
#!/usr/bin/python3
from scapy.all import *
def print_pkt(pkt):
        if(pkt[1].dst=="8.8.8.9"):
                print(pkt[1].src)
                sr1(IP(dst=pkt[1].src,src="8.8.8.9")/ICMP(type=0, code=0))
pkt = sniff(filter='icmp',prn=print_pkt,count=1)
[10/20/22]seed@VM:~/.../lab2$
```

follows:

We send a echo reply package to the sender:


192.168.86.203
Begin emission:
Finished sending 1 packets.
*
Received 1 packets, got 1 answers, remaining 0 packets
[10/20/22]seed@VM:~/.../lab2$ ^C