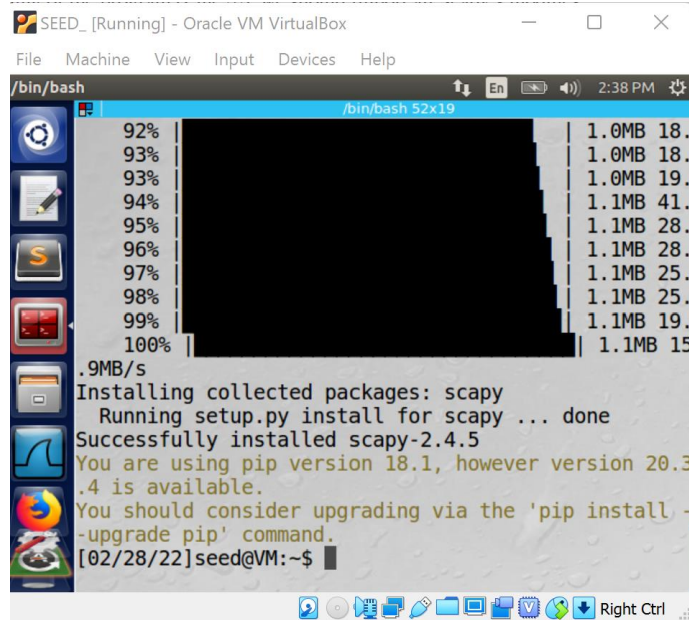


Lab2 Packet Sniffing and spoofing

I. Lab set up

- a. Set up two virtual machines (SEEDLab image) on VirtualBox
 - i. Attacker IP: 10.0.2.15
 - ii. Victim IP: 10.0.2.4
- b. Install scapy by 'sudo pip3 install scapy'



```
SEED_ [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
/bin/bash
92% | 1.0MB 18.
93% | 1.0MB 18.
93% | 1.0MB 19.
94% | 1.1MB 41.
95% | 1.1MB 28.
96% | 1.1MB 28.
97% | 1.1MB 25.
98% | 1.1MB 25.
99% | 1.1MB 19.
100% | 1.1MB 15
.9MB/s
Installing collected packages: scapy
Running setup.py install for scapy ... done
Successfully installed scapy-2.4.5
You are using pip version 18.1, however version 20.3
.4 is available.
You should consider upgrading via the 'pip install -
-upgrade pip' command.
[02/28/22]seed@VM:~$
```

c.

II. Task 1.1A

- a. Use the skeleton code provided to create a sniff.py

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

- b.
- c. To create some traffic on the machine, ping a random website, here I pinged youtube.com

```
[02/28/22]seed@VM:~$ ping youtube.com
PING youtube.com (142.250.80.78) 56(84) bytes of data:
64 bytes from lga34s35-in-f14.1e100.net (142.250.80.78): icmp_seq=1 ttl=58 time=4.24 ms
64 bytes from lga34s35-in-f14.1e100.net (142.250.80.78): icmp_seq=2 ttl=58 time=4.67 ms
64 bytes from lga34s35-in-f14.1e100.net (142.250.80.78): icmp_seq=3 ttl=58 time=3.35 ms
64 bytes from lga34s35-in-f14.1e100.net (142.250.80.78): icmp_seq=4 ttl=58 time=3.77 ms
```

- d.
- e. Make sniff.py executable by running 'chmod a+x sniff.py' and run the file in root by 'sudo ./sniff.py'. We can see echo request and echo reply ICMP packets generated when pinging the website.

```
SEED_ [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

/bin/bash
[3:]
[02/28/22]seed@VM:~$ chmod a+x sniff.py
[02/28/22]seed@VM:~$ sudo ./sniff.py
###[ Ethernet ]###
dst      = 52:54:00:12:35:00
src      = 08:00:27:95:9a:7c
type     = IPv4
###[ IP ]###
version  = 4
ihl      = 5
tos      = 0x0
len      = 84
id       = 25564
flags    = DF
frag     = 0
ttl      = 64
proto    = icmp
chksum   = 0xeb75
src      = 10.0.2.15
dst      = 142.250.80.78
options  \
###[ ICMP ]###
type     = echo-request
code     = 0
chksum   = 0x67a6
id       = 0xc6a
seq      = 0x2c
unused   = ''
###[ Raw ]###
load     = '\xb4(\xdb\xc5\02\00\08\t\n\0b\0c\r\0e\0f\010\011\012\013\014\015\016\017\018\019\01a\01b\01c\01d\01e\01f !"%&'()*+,-./01234567'

###[ Ethernet ]###
dst      = 08:00:27:95:9a:7c
src      = 52:54:00:12:35:00
type     = IPv4
###[ IP ]###
version  = 4
ihl      = 5
tos      = 0x0
len      = 84
id       = 1219
flags    = 
frag     = 0
ttl      = 58
proto    = icmp
chksum   = 0x908f
src      = 142.250.80.78
dst      = 10.0.2.15
options  \
###[ ICMP ]###
type     = echo-reply
```

- f.
- g. Now run sniff.py without root by './sniff.py', we see a permission error saying operation not permitted.

```
SEED_ [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

/bin/bash
[3]+ Stopped sudo ./sniff.py
[02/28/22]seed@VM:~$ ./sniff.py
Traceback (most recent call last):
  File "./sniff.py", line 5, in <module>
    pkt = sniff(filter='icmp', prn=print_pkt)
  File "/usr/local/lib/python3.5/dist-packages/scapy/sendrecv.py", line 1263, in sniff
    sniffer._run(*args, **kwargs)
  File "/usr/local/lib/python3.5/dist-packages/scapy/sendrecv.py", line 1128, in _run
    **karg)) = iface
  File "/usr/local/lib/python3.5/dist-packages/scapy/arch/linux.py", line 487, in _init_socket.AF_PACKET, socket.SOCK_RAW, socket.htons(type)
)
  File "/usr/lib/python3.5/socket.py", line 134, in _init__
    _socket.socket._init__(self, family, type, proto, fi
leno)
PermissionError: [Errno 1] Operation not permitted
[02/28/22]seed@VM:~$
```

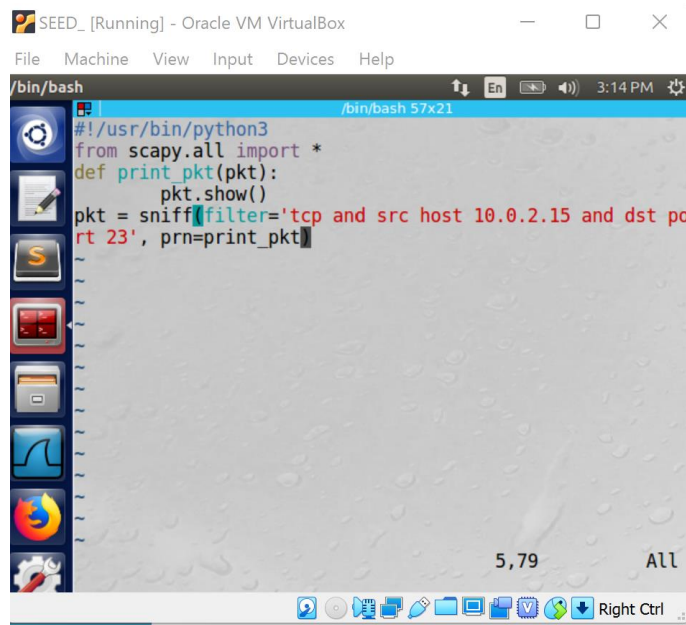
h.

III. Observations of Task 1.1A

- a. Line 5 in sniff.py, the function sniff() from scapy needs root permission to operate. A get-around is to pip install scapy-unroot packet. Otherwise, root privileges are needed for scapy.

IV. Task 1.1B

- a. Capture only ICMP packets (this was set in the skeleton code, see II.(e) for output).
- b. To capture any TCP packet that comes from a particular IP and with a destination port number 23, we need to tweak the sniff.py. Source host IP is the IP of the current machine, which is 10.0.2.15.



SEED_ [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

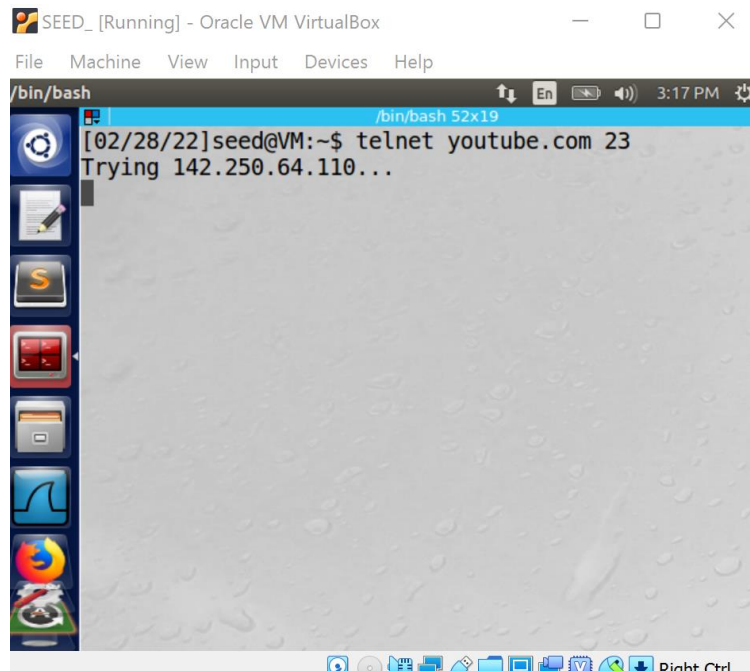
/bin/bash

```
#!/usr/bin/python3
from scapy.all import *
def print_pkt(pkt):
    pkt.show()
pkt = sniff(filter='tcp and src host 10.0.2.15 and dst port 23', prn=print_pkt)
```

5,79 All

Right Ctrl

- c.
- d. To generate some TCP packets, I choose to telnet youtube.com at port 23.



SEED_ [Running] - Oracle VM VirtualBox

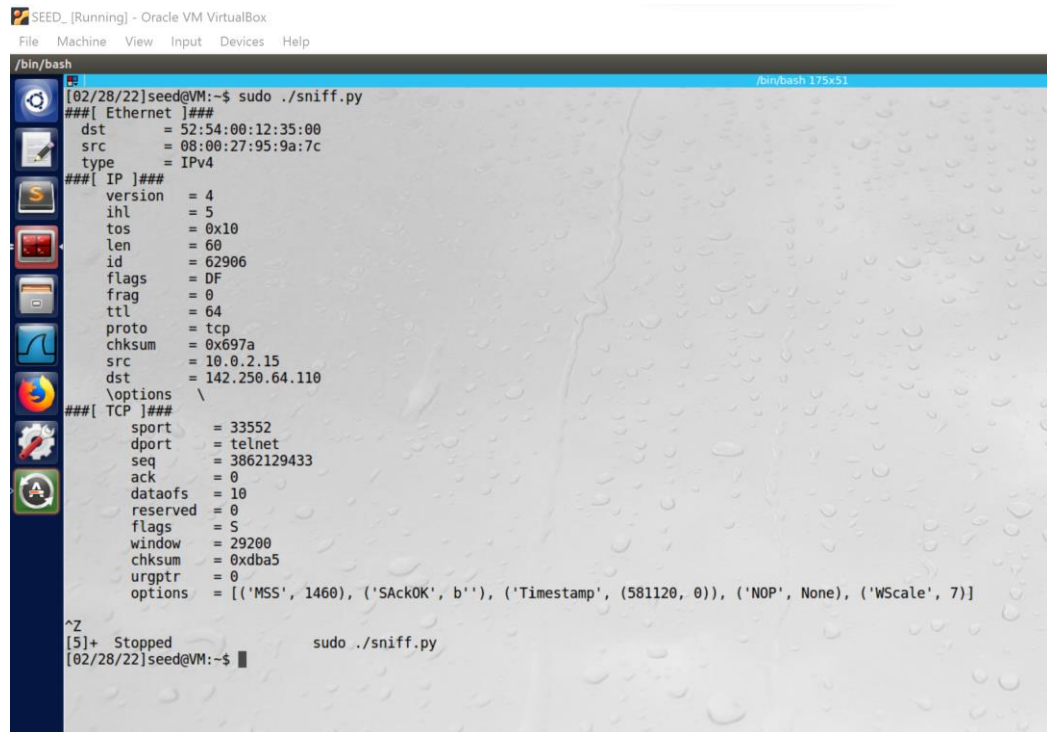
File Machine View Input Devices Help

/bin/bash

```
[02/28/22]seed@VM:~$ telnet youtube.com 23
Trying 142.250.64.110...
```

Right Ctrl

- e.
- f. Then run sniff.py with root, the information of the captured TCP packet is printed out. Noted here only packets from 10.0.2.15 to port 23 was captured.

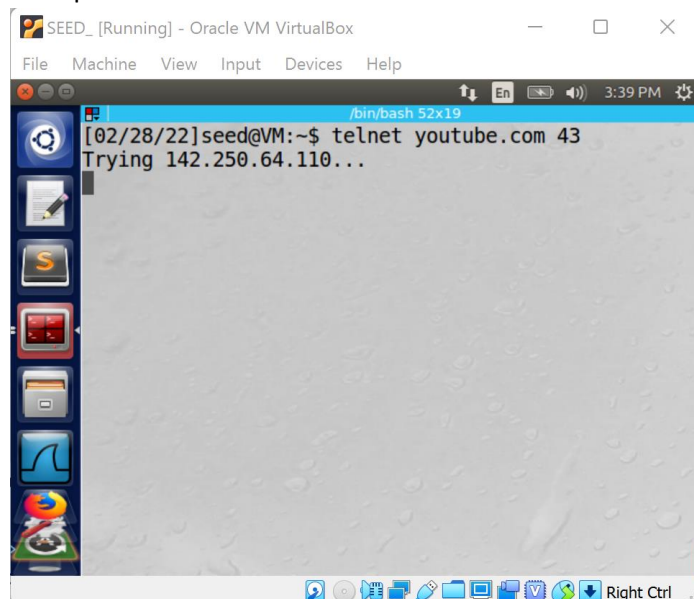


```
SEED_ [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

/bin/bash
[02/28/22]seed@VM:~$ sudo ./sniff.py
###[ Ethernet ]###
dst      = 52:54:00:12:35:00
src      = 08:00:27:95:9a:7c
type     = IPv4
###[ IP ]###
version  = 4
ihl      = 5
tos      = 0x10
len      = 60
id       = 62906
flags    = DF
frag     = 0
ttl      = 64
proto    = tcp
chksum   = 0x697a
src      = 10.0.2.15
dst      = 142.250.64.110
\options
###[ TCP ]###
sport    = 33552
dport    = telnet
seq      = 3862129433
ack      = 0
dataoffs = 10
reserved = 0
flags    = S
window   = 29200
chksum   = 0xdba5
urgptr   = 0
options  = [('MSS', 1460), ('SackOK', b''), ('Timestamp', (581120, 0)), ('NOP', None), ('WScale', 7)]

^Z
[5]+  Stopped                  sudo ./sniff.py
[02/28/22]seed@VM:~$
```

- g.
- h. If create a telnet connection by telnet youtube.com <other ports #>, no packets would be captured.

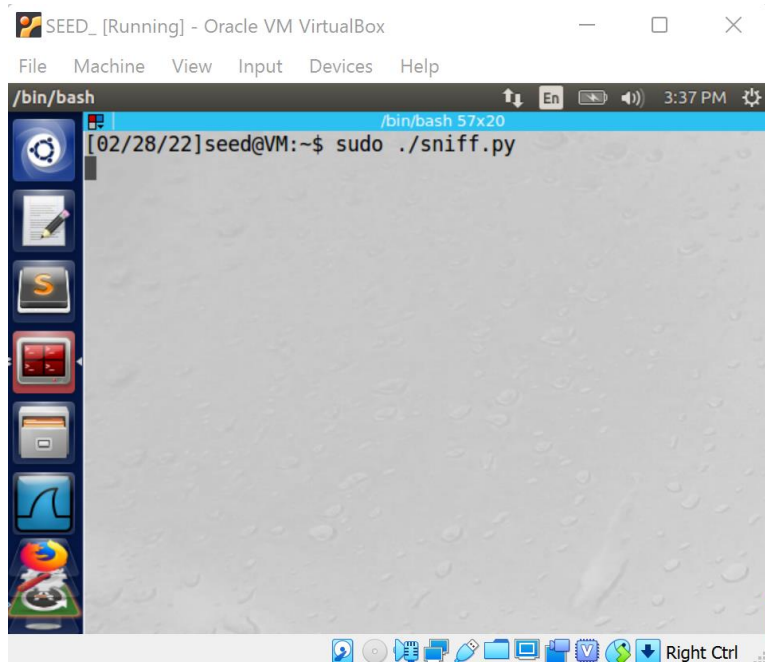


```
SEED_ [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

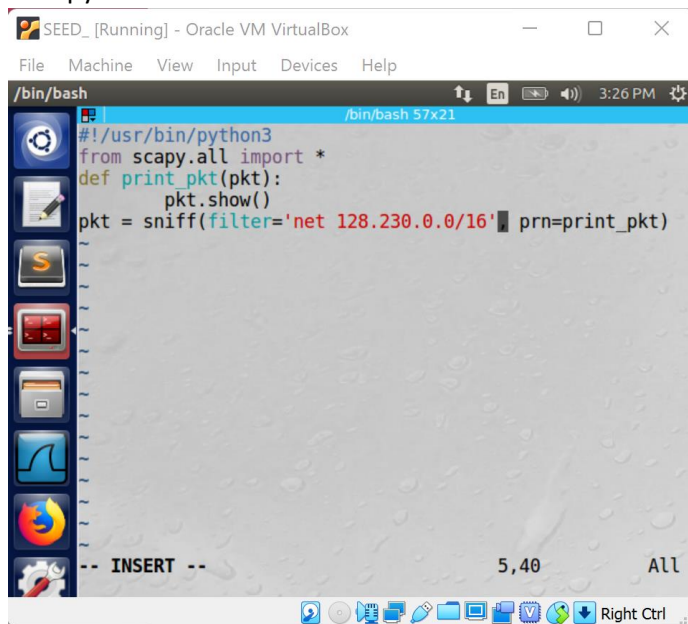
/bin/bash 52x19
[02/28/22]seed@VM:~$ telnet youtube.com 43
Trying 142.250.64.110...

```

- i.



- j.
- k. To capture packets comes from or to go to a particular subnet (128.230.0.0/16), tweak sniff.py as follows.



- l.
- m. On terminal, we telnet this subnet by 'telnet 128.230.0.0/16 23'

```

SEED_ [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
/bin/bash
[02/28/22]seed@VM:~$ telnet 128.230.0.0 23
Trying 128.230.0.0...

```

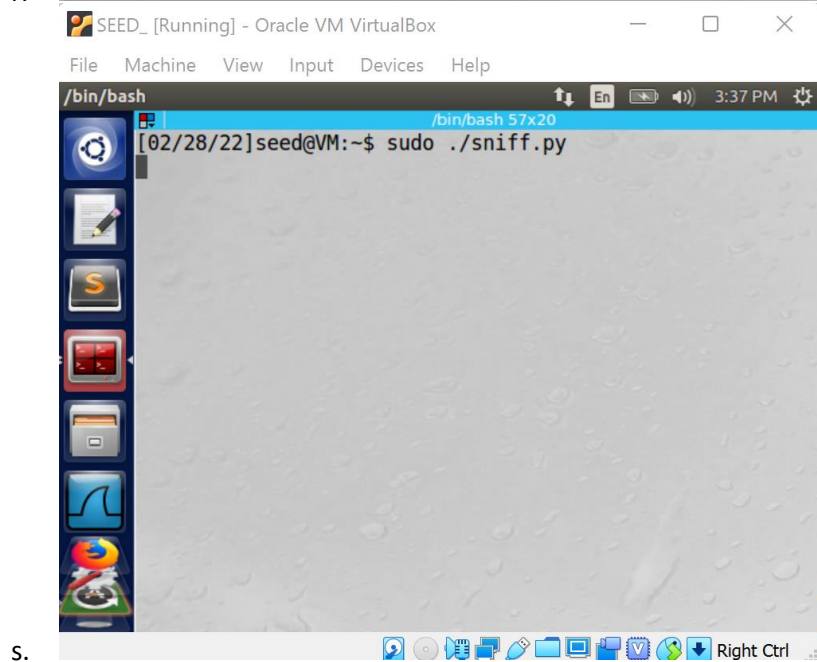
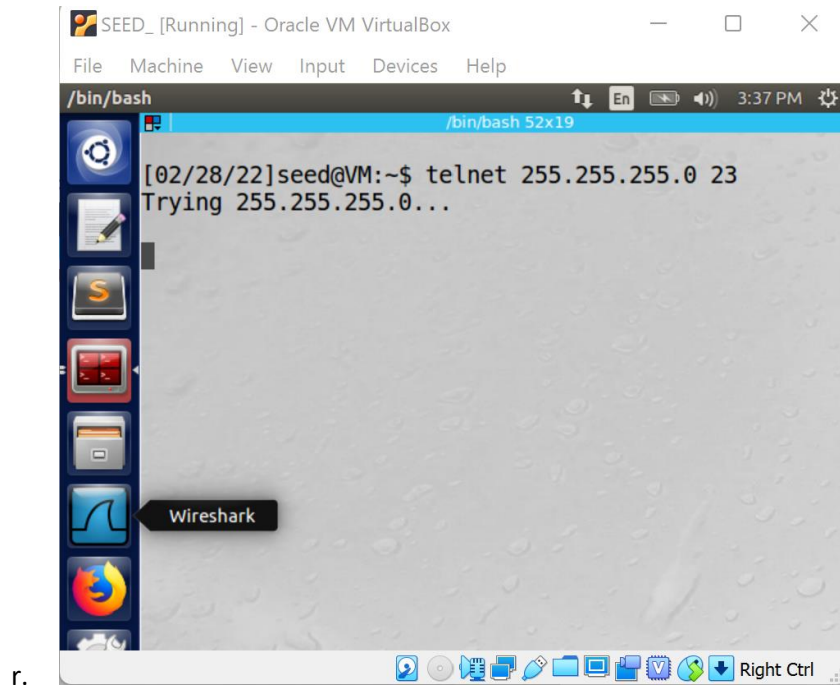
- n.
- o. Run sniff.py with root, the information of the packets goes or comes from the subnet specified is printed out.

```

SEED_ [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
/bin/bash
[02/28/22]seed@VM:~$ sudo ./sniff.py
#### Ethernet ####
dst      = 52:54:00:12:35:00
src      = 08:00:27:95:9a:7c
type     = IPv4
#### IP ####
version  = 4
ihl      = 5
tos      = 0x10
len      = 60
id       = 18096
flags    = DF
frag     = 0
ttl      = 64
proto    = tcp
chksum   = 0x6707
src      = 10.0.2.15
dst      = 128.230.0.0
options  \
#### TCP ####
sport    = 56932
dport    = telnet
seq      = 3721515407
ack      = 0
dataofs  = 10
reserved = 0
flags    = S
window   = 29200
chksum   = 0x8d23
urgptr   = 0
options  = [('MSS', 1460), ('AckOK', b''), ('Timestamp', (767488, 0)), ('NOP', None), ('WScale', 7)]
^Z
[0]+  Stopped                  sudo ./sniff.py
[02/28/22]seed@VM:~$

```

- p.
- q. If we telnet another subnet, no packets would be captured by this filter.



V. Task 1.2

- a. Use the skeleton code to create `spoof.py`, in it, create a source IP that is arbitrary. Here I choose `10.0.0.0` as arbitrary IP. Insert a line `'a.src='10.0.0.0'`


```
/bin/bash
#!/usr/bin/python3
from scapy.all import *
a = IP()
a.src='10.0.0.0'
a.dst='10.0.2.15'
b=ICMP()
p=a/b
send(p)
```

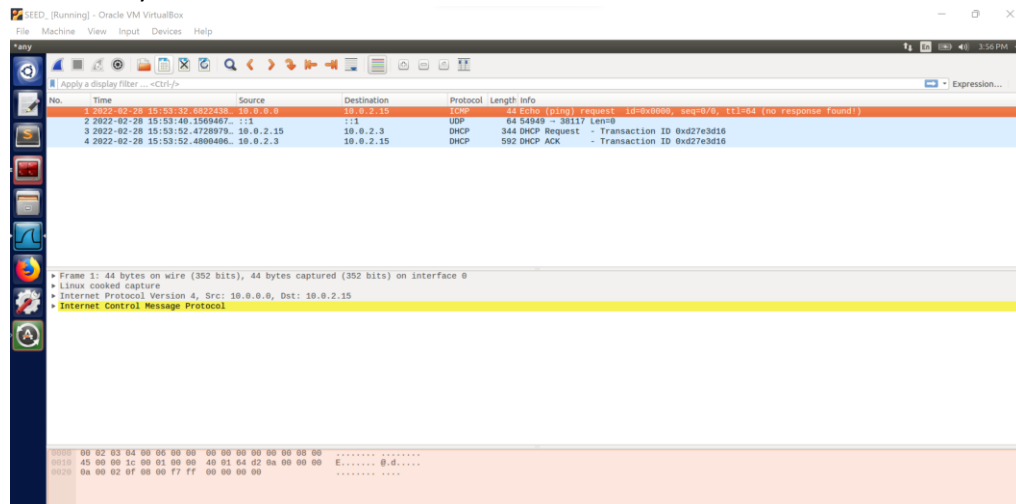
"spoof.py" 8L, 110C 1,1 All

- b.
- c. Make spoof.py executable by running 'chmod a+x spoof.py' and run it with root privileges. It outputs 'Sent 1 packet'.

```
/bin/bash
[02/28/22]seed@VM:~$ sudo ./spoof.py
Sent 1 packets.
[02/28/22]seed@VM:~$
```

- d.
- e. At the same time, open Wireshark to capture the packet, we can see an ICMP echo request packets were captured, with source IP of 10.0.0.0 (arbitrary IP specified in

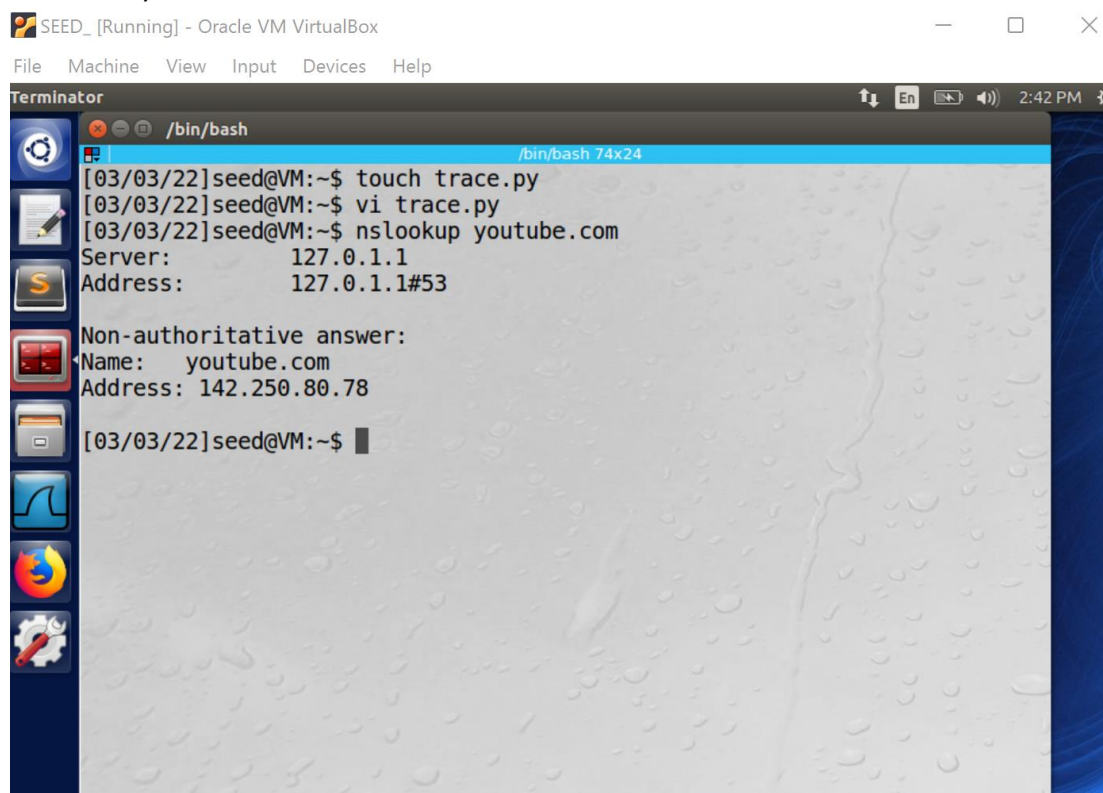
script) and destination IP of 10.0.2.15. The spoof of an ICMP echo request packet with an arbitrary source IP address was successful.



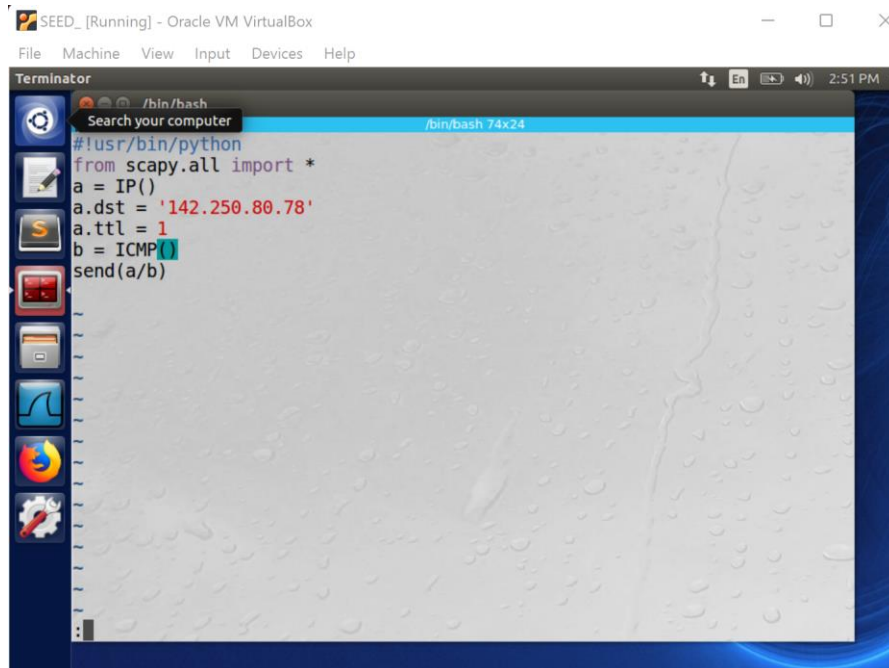
f.

VI. Task 1.3

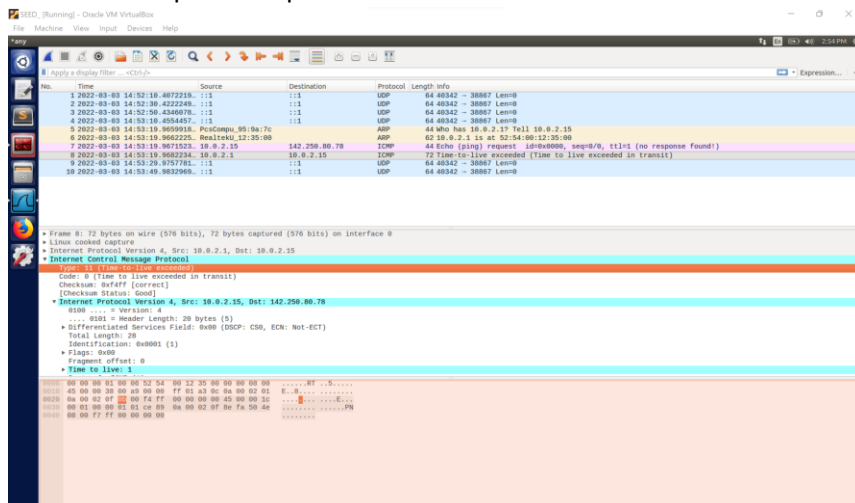
- In this task, I will try to send a packet to youtube.com. Use nslookup to find the IP address of youtube.com.



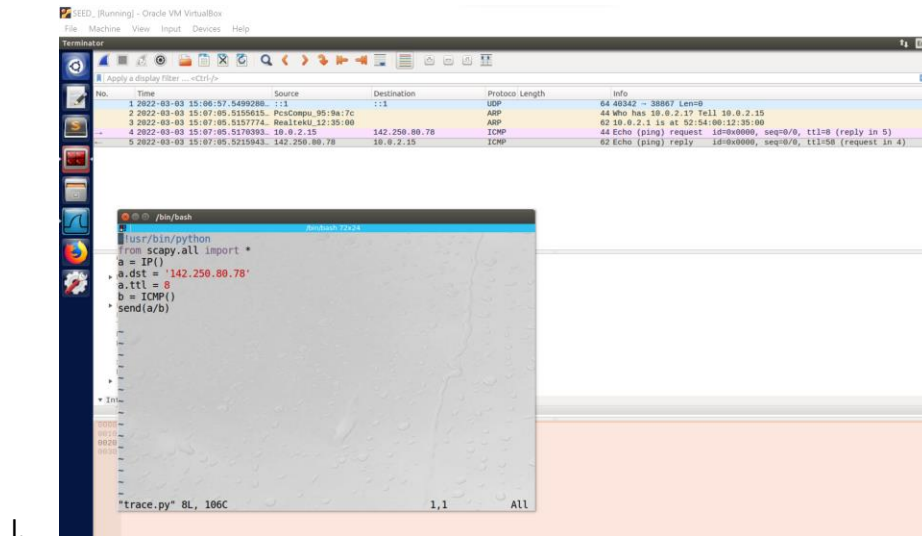
-
-
- Create trace.py using the skeleton code provided. Set ttl field to 1 at the first run. Set destination IP address as youtube's IP Address.



- d.
- e. Run trace.py and capture packets on Wireshark in the meanwhile. We can see that time-to-live exceeded since ttl was set to 1. We can get the IP address of the first router at the last ICMP packet captured.



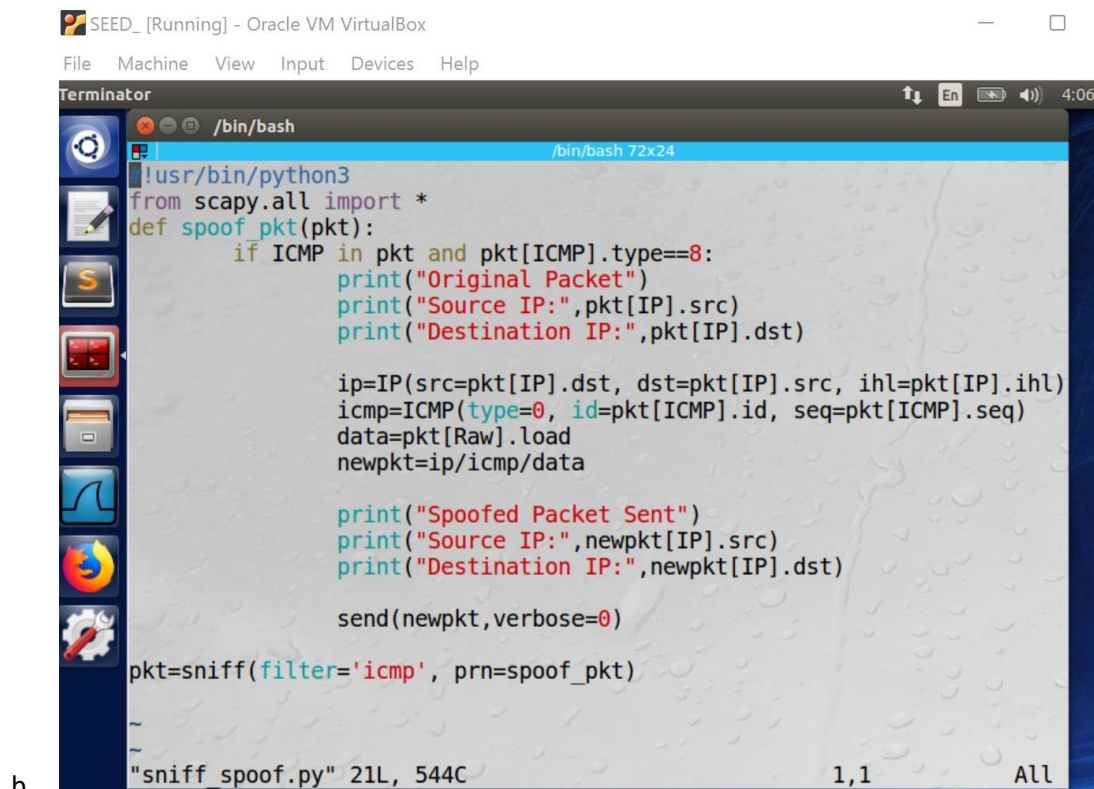
- f.
- g. Increase ttl field value by 1 every time if time-to-live exceeded. The IP address of the next router can be found by looking at the last ICMP packet information.



I.

VII. Task 1.4

- Create sniff_spoof.py using the code of handsonsecurity.net, *Network Security, Chapter 15 slides, pp 28 by Wenliang Du*.



b.

- On the victim machine, we ping a random IP, here I choose 10.0.0.1, and it is not reachable when pinging it.

SEED_Clone1 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
Terminator
/bin/bash
[03/03/22]seed@VM:~$ ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
From 64.18.144.1 icmp_seq=2 Destination Net Unreachable
From 64.18.144.1 icmp_seq=3 Destination Net Unreachable
From 64.18.144.1 icmp_seq=4 Destination Net Unreachable
^Z
[19]+  Stopped                  ping 10.0.0.1
```

- d.
- e. On attacker machine, make sniff_spoof.py executable and then run it, while running it, ping 10.0.0.1 again on the victim machine, we can see it is receiving replies from 10.0.0.1 which is not alive.

```
SEED_Clone1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Terminator
/bin/bash
[03/03/22]seed@VM:~$ ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
From 64.18.144.1 icmp_seq=2 Destination Net Unreachable
From 64.18.144.1 icmp_seq=3 Destination Net Unreachable
From 64.18.144.1 icmp_seq=4 Destination Net Unreachable
^Z
[19]+  Stopped                  ping 10.0.0.1

[03/03/22]seed@VM:~$ ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=11.9 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=8.14 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=11.0 ms
^Z
[20]+  Stopped                  ping 10.0.0.1

[03/03/22]seed@VM:~$

SEED_ [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Terminator
/bin/bash
[03/03/22]seed@VM:~$ sudo python sniff_spoof.py
Original Packet
('Source IP:', '10.0.2.4')
('Destination IP:', '10.0.0.1')
Spoofed Packet Sent
('Source IP:', '10.0.0.1')
('Destination IP:', '10.0.2.4')
Original Packet
('Source IP:', '10.0.2.4')
('Destination IP:', '10.0.0.1')
Spoofed Packet Sent
('Source IP:', '10.0.0.1')
('Destination IP:', '10.0.2.4')
Original Packet
('Source IP:', '10.0.2.4')
('Destination IP:', '10.0.0.1')
Spoofed Packet Sent
('Source IP:', '10.0.0.1')
('Destination IP:', '10.0.2.4')
^Z
[9]+  Stopped                  sudo python sniff_spoof.py
[03/03/22]seed@VM:~$
```

- f.
- g. Stop the program on attacker machine and try to ping 10.0.0.1 again, the destination net unreachable message would show up again, indicating that the sniff_spoof.py successfully faked the echo request.

```
SEED_Clone1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Terminator
/bin/bash
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=19.4 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=17.5 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=12.5 ms
^Z
[24]+  Stopped                  ping 10.0.0.1

[03/03/22]seed@VM:~$ ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
From 64.18.144.1 icmp_seq=6 Destination Net Unreachable
From 64.18.144.1 icmp_seq=7 Destination Net Unreachable
From 64.18.144.1 icmp_seq=10 Destination Net Unreachable
From 64.18.144.1 icmp_seq=12 Destination Net Unreachable
From 64.18.144.1 icmp_seq=14 Destination Net Unreachable
From 64.18.144.1 icmp_seq=26 Destination Net Unreachable
From 64.18.144.1 icmp_seq=27 Destination Net Unreachable
From 64.18.144.1 icmp_seq=29 Destination Net Unreachable
^Z
[25]+  Stopped                  ping 10.0.0.1

[03/03/22]seed@VM:~$ ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
From 64.18.144.1 icmp_seq=2 Destination Net Unreachable
From 64.18.144.1 icmp_seq=7 Destination Net Unreachable
^Z
[26]+  Stopped                  ping 10.0.0.1

[03/03/22]seed@VM:~$

SEED_ [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Terminator
/bin/bash
Spoofed Packet Sent
('Source IP:', '10.0.0.1')
('Destination IP:', '10.0.2.4')
Original Packet
('Source IP:', '10.0.2.4')
('Destination IP:', '10.0.0.1')
Spoofed Packet Sent
('Source IP:', '10.0.0.1')
('Destination IP:', '10.0.2.4')
Original Packet
('Source IP:', '10.0.2.4')
('Destination IP:', '10.0.0.1')
Wireshark Packet Sent
('Source IP:', '10.0.0.1')
('Destination IP:', '10.0.2.4')
Firefox Web Browser
('Source IP:', '10.0.2.4')
('Destination IP:', '10.0.0.1')
Spoofed Packet Sent
('Source IP:', '10.0.0.1')
('Destination IP:', '10.0.2.4')
^Z
[11]+  Stopped                  sudo python sniff_spoof.py
[03/03/22]seed@VM:~$
```

- h.