

UE18CS335 – COMPUTER NETWORK SECURITY

Lab – 1 SNIFFING AND SPOOFING

Date: 31/01/2021

By:

Nitish S

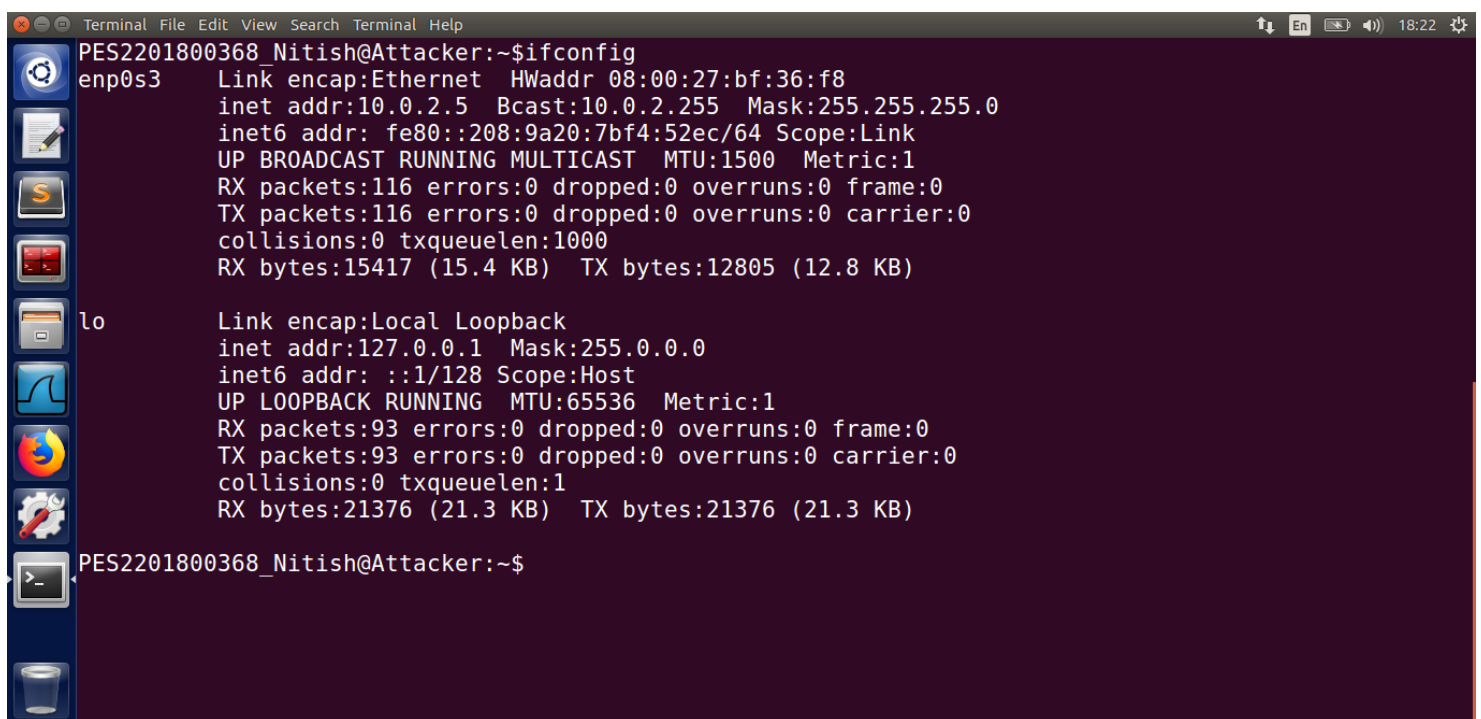
PES2201800368

6 'A'

2. Using Tools to Sniff and Spoof Packets using Scapy

Command: `sudo apt-get install Scapy`

Attacker Machine: 10.0.2.5

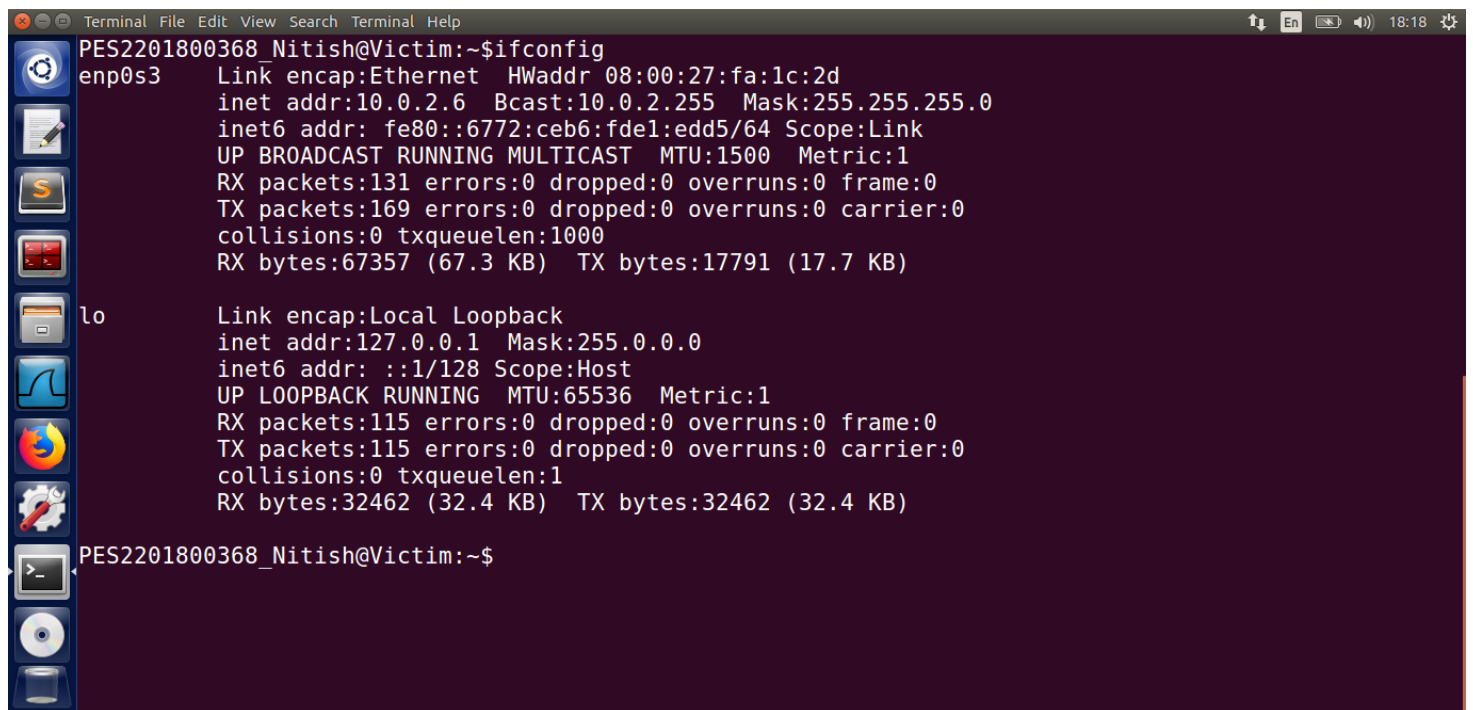


```
Terminal File Edit View Search Terminal Help
PES2201800368_Nitish@Attacker:~$ifconfig
enp0s3  Link encap:Ethernet  HWaddr 08:00:27:bf:36:f8
        inet addr:10.0.2.5  Bcast:10.0.2.255  Mask:255.255.255.0
        inet6 addr: fe80::208:9a20:7bf4:52ec/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:116 errors:0 dropped:0 overruns:0 frame:0
        TX packets:116 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:15417 (15.4 KB)  TX bytes:12805 (12.8 KB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:93 errors:0 dropped:0 overruns:0 frame:0
        TX packets:93 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1
        RX bytes:21376 (21.3 KB)  TX bytes:21376 (21.3 KB)

PES2201800368_Nitish@Attacker:~$
```

Victim Machine: 10.0.2.6



```
Terminal File Edit View Search Terminal Help
PES2201800368_Nitish@Victim:~$ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:fa:1c:2d
          inet addr:10.0.2.6  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::6772:ceb6:fde1:edd5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:131 errors:0 dropped:0 overruns:0 frame:0
          TX packets:169 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:67357 (67.3 KB)  TX bytes:17791 (17.7 KB)

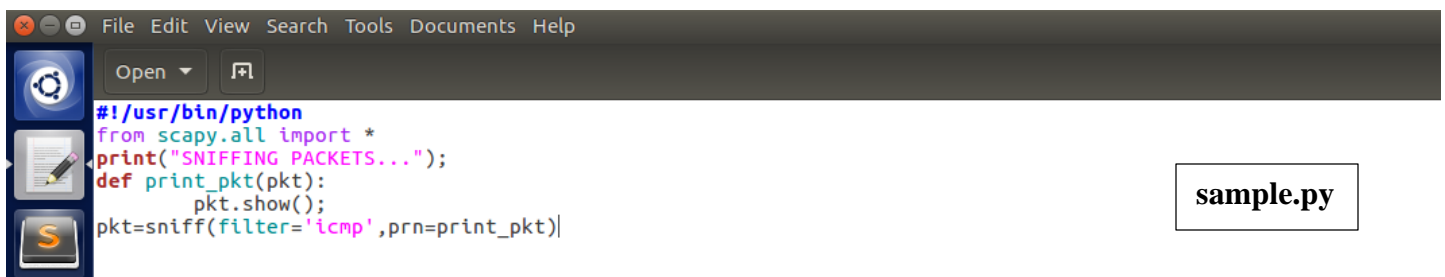
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:115 errors:0 dropped:0 overruns:0 frame:0
          TX packets:115 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:32462 (32.4 KB)  TX bytes:32462 (32.4 KB)

PES2201800368_Nitish@Victim:~$
```

2.1 Task 1: Sniffing Packets

2.1.1 Task 1.1 Sniff IP packets using Scapy

Command : `sudo python sample.py`



```
File Edit View Search Tools Documents Help
Open [icon]
#!/usr/bin/python
from scapy.all import *
print("SNIFFING PACKETS...");
def print_pkt(pkt):
    pkt.show();
pkt=sniff(filter='icmp',prn=print_pkt)|
```

sample.py

Explain on which VM you ran this command and why? Provide a screenshot of your observations.

Ans: - The above command has to be run on attacker side (10.0.2.5) because we are trying to sniff packets from victim on attacker.

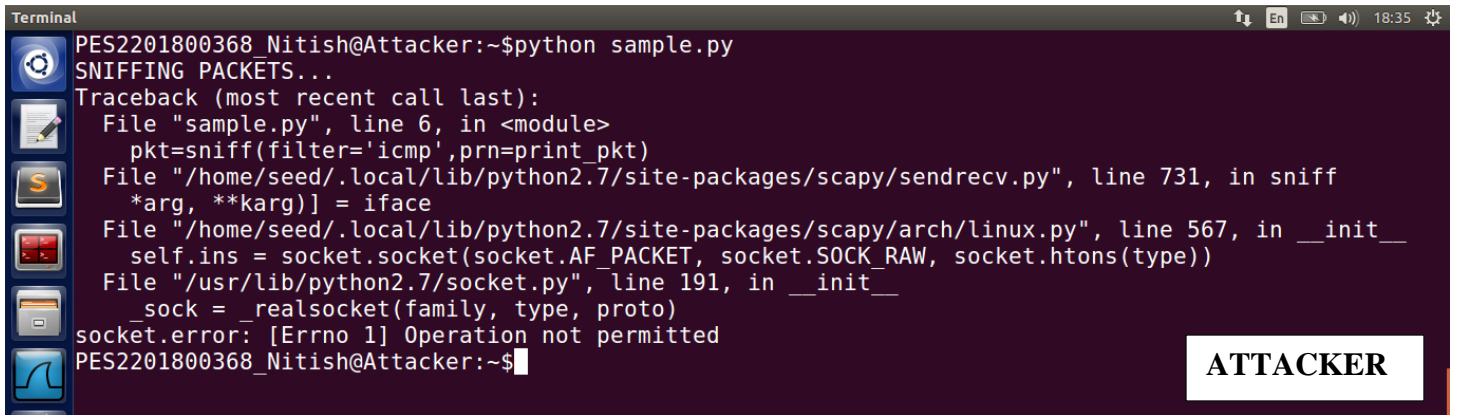
```
Terminal
PES2201800368_Nitish@Attacker:~$gedit sample.py
PES2201800368_Nitish@Attacker:~$sudo python sample.py
SNIFFING PACKETS...
###[ Ethernet ]###
dst      = 08:00:27:bf:36:f8
src      = 08:00:27:fa:1c:2d
type     = 0x800
###[ IP ]###
version  = 4
ihl      = 5
tos      = 0x0
len      = 84
id       = 5391
flags    = DF
frag     = 0
ttl      = 64
proto    = icmp
chksum   = 0xd90
src      = 10.0.2.6
dst      = 10.0.2.5
\options \
###[ ICMP ]###
type     = echo-request
code     = 0
```

ATTACKER

```
Terminal
len      = 84
id       = 5391
flags    = DF
frag     = 0
ttl      = 64
proto    = icmp
chksum   = 0xd90
src      = 10.0.2.6
dst      = 10.0.2.5
\options \
###[ ICMP ]###
type     = echo-request
code     = 0
chksum   = 0xf5e9
id       = 0xdaa
seq      = 0x1
###[ Raw ]###
load     = 'U\x07\x14'\x9f\x00\x01\x00\x08\t\n\x0b\x0c\r\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !"#%&\'()*+,-./01234567'
###[ Ethernet ]###
dst      = 08:00:27:fa:1c:2d
src      = 08:00:27:bf:36:f8
type     = 0x800
###[ IP ]###
```

ATTACKER

Command: python sample.py

A terminal window with a dark background. The prompt is 'PES2201800368 Nitish@Attacker:~\$'. The user has run 'python sample.py'. The output is 'SNIFFING PACKETS...' followed by a traceback. The error is 'socket.error: [Errno 1] Operation not permitted' at line 191 of '/usr/lib/python2.7/socket.py'. A white box with the word 'ATTACKER' is in the bottom right corner.

```
PES2201800368 Nitish@Attacker:~$python sample.py
SNIFFING PACKETS...
Traceback (most recent call last):
  File "sample.py", line 6, in <module>
    pkt=sniff(filter='icmp',prn=print_pkt)
  File "/home/seed/.local/lib/python2.7/site-packages/scapy/sendrecv.py", line 731, in sniff
    *arg, **karg)] = iface
  File "/home/seed/.local/lib/python2.7/site-packages/scapy/arch/linux.py", line 567, in __init__
    self.ins = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, socket.htons(type))
  File "/usr/lib/python2.7/socket.py", line 191, in __init__
    _sock = _realsocket(family, type, proto)
socket.error: [Errno 1] Operation not permitted
PES2201800368_Nitish@Attacker:~$
```

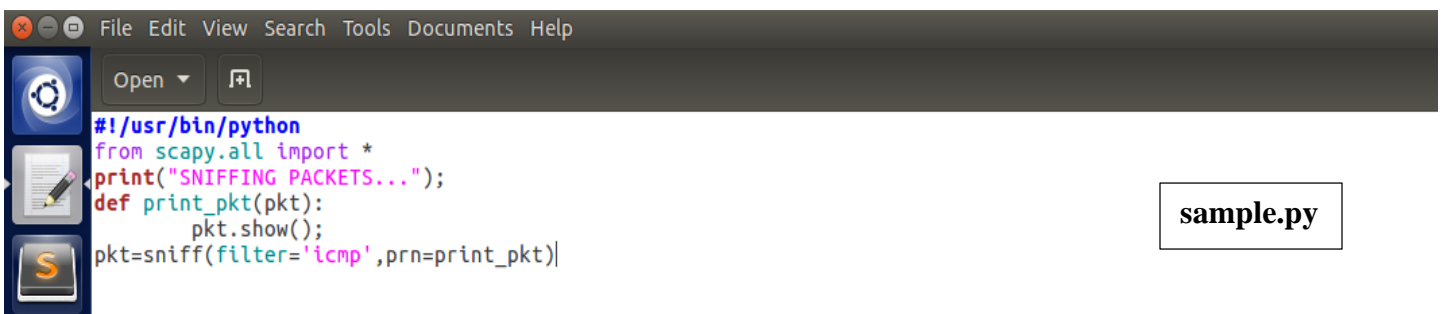
ATTACKER

Now, we run the same program without root privileges. Do you find any issues? If so, why? Provide a screenshot of your observations.

Ans: - There is an error in running the program without root privileges because root privileges are required to put the network adapter in promiscuous mode, without which the sniffer program will not run.

2.1.2 Task 1.2 Capturing ICMP, TCP packet and Subnet

2.1.2.1 Capture only the ICMP packet

A screenshot of a text editor window. The title bar says 'File Edit View Search Tools Documents Help'. The code is as follows:#!/usr/bin/python
from scapy.all import *
print("SNIFFING PACKETS...");
def print_pkt(pkt):
 pkt.show();
pkt=sniff(filter='icmp',prn=print_pkt)|
A white box with the text 'sample.py' is on the right side.

```
#!/usr/bin/python
from scapy.all import *
print("SNIFFING PACKETS...");
def print_pkt(pkt):
    pkt.show();
pkt=sniff(filter='icmp',prn=print_pkt)|
```

sample.py

Command: `sudo python sample.py`

Open another terminal on the same VM (10.0.2.5) and ping 8.8.8.8

Command: `ping 8.8.8.8`

The ICMP packets are captured by the sniffer program. Provide a screenshot of your observations.

```
Terminal
PES2201800368 Nitish@Attacker:~$sudo python sample.py
SNIFFING PACKETS...
###[ Ethernet ]###
dst      = 52:54:00:12:35:00
src      = 08:00:27:bf:36:f8
type     = 0x800
###[ IP ]###
version  = 4
ihl      = 5
tos      = 0x0
len      = 84
id       = 61904
flags    = DF
frag     = 0
ttl      = 64
proto    = icmp
chksum   = 0x2cc4
src      = 10.0.2.5
dst      = 8.8.8.8
\options \
###[ ICMP ]###
type     = echo-request
code     = 0
chksum   = 0x8b7
id       = 0xd67

Terminal
PES2201800368 Nitish@Attacker:~$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=110 time=45.7 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=110 time=45.1 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=110 time=53.7 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=110 time=31.8 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=110 time=57.6 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=110 time=61.9 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=110 time=46.6 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=110 time=55.2 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=110 time=43.9 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=110 time=34.3 ms
64 bytes from 8.8.8.8: icmp_seq=13 ttl=110 time=1477 ms
64 bytes from 8.8.8.8: icmp_seq=14 ttl=110 time=1196 ms
64 bytes from 8.8.8.8: icmp_seq=15 ttl=110 time=1460 ms
64 bytes from 8.8.8.8: icmp_seq=16 ttl=110 time=1356 ms
```

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

The below code sniffs the TCP traffic from a specific host (10.0.2.6) to port 23.

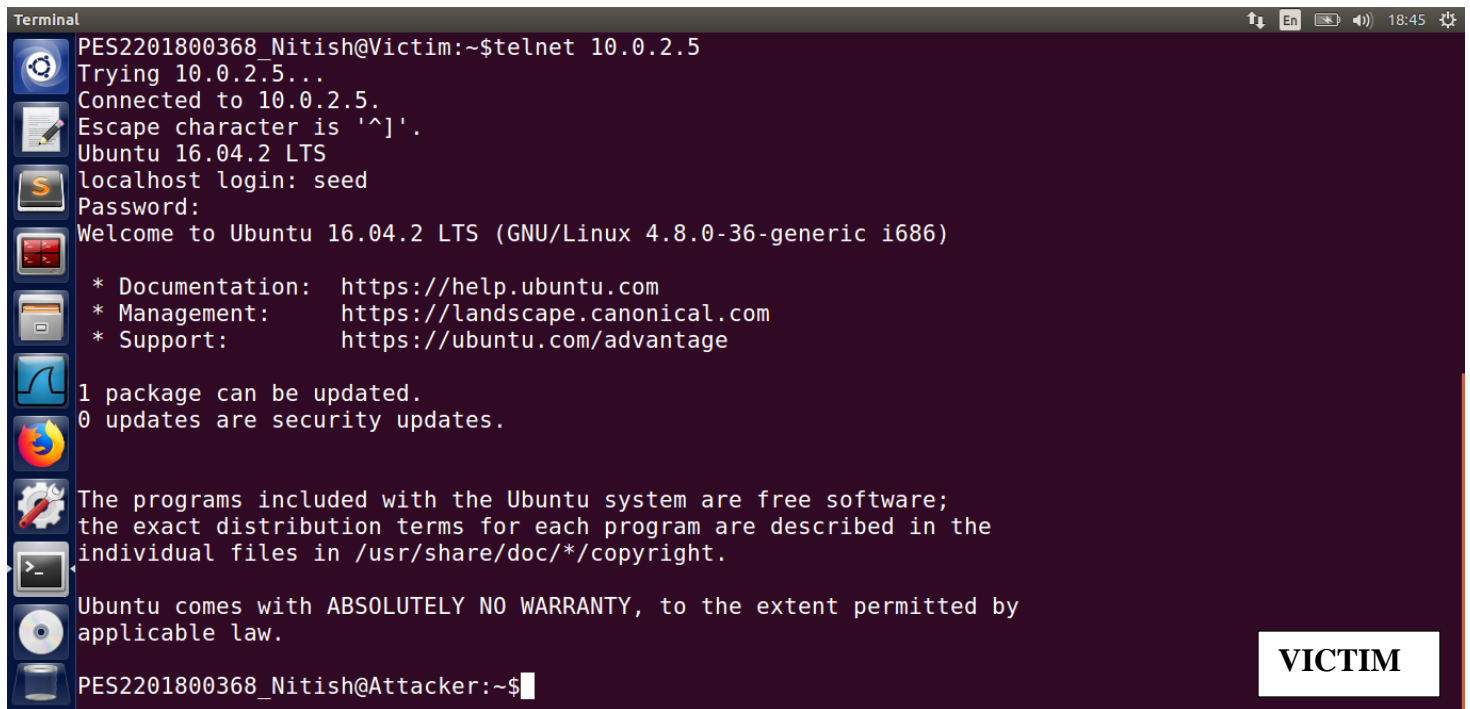
```
File Edit View Search Tools Documents Help
Open Save
#!/usr/bin/python
from scapy.all import *
print("SNIFFING PACKETS...");
def print_pkt(pkt):
    pkt.show();
pkt=sniff(filter='tcp and (src host=10.0.2.6 and dst port 23)',prn=print_pkt)
```

sniff.py

Command: telnet 10.0.2.5

Explain where you will run Telnet. Provide screenshots of your observations.

Ans: - The above command has to be run at other machines except 10.0.2.5, as we are trying to connect to 10.0.2.5. It makes no sense to run it on the same machine. Hence, we run the command on the victim machine here i.e. 10.0.2.6



```
Terminal
PES2201800368_Nitish@Victim:~$telnet 10.0.2.5
Trying 10.0.2.5...
Connected to 10.0.2.5.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
localhost login: seed
Password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

PES2201800368_Nitish@Attacker:~$
```

VICTIM

2.1.2.2.1 iii) Capture packets comes from or to go to a particular subnet



```
File Edit View Search Tools Documents Help
Open Save

#!/usr/bin/python
from scapy.all import *
print("SNIFFING PACKETS...");
def print_pkt(pkt):
    pkt.show();
pkt=sniff(filter='src net 192.168.56.0/24',prn=print_pkt)
```

sniff1.py

Command: ping 192.168.56.1

Provide a screenshot of your observations.

Command: sudo python sniff1.py

Provide a screenshot of your observations

```
Terminal
PES2201800368 Nitish@Attacker:~$sudo python sniff1.py
SNIFFING PACKETS...
###[ Ethernet ]###
  dst      = 08:00:27:fa:1c:2d
  src      = 52:54:00:12:35:00
  type     = 0x800
###[ IP ]###
  version  = 4
  ihl      = 5
  tos      = 0x0
  len      = 84
  id       = 190
  flags    =
  frag     = 0
  ttl      = 127
  proto    = icmp
  chksum   = 0x363c
  src      = 192.168.56.1
  dst      = 10.0.2.6
  \options \
###[ ICMP ]###
  type     = echo-reply
  code     = 0
  chksum   = 0x919d
```

ATTACKER

```
Terminal File Edit View Search Terminal Help
PES2201800368 Nitish@Victim:~$ping 192.168.56.1
PING 192.168.56.1 (192.168.56.1) 56(84) bytes of data.
64 bytes from 192.168.56.1: icmp_seq=1 ttl=127 time=1.15 ms
64 bytes from 192.168.56.1: icmp_seq=2 ttl=127 time=0.644 ms
64 bytes from 192.168.56.1: icmp_seq=3 ttl=127 time=0.698 ms
64 bytes from 192.168.56.1: icmp_seq=4 ttl=127 time=1.56 ms
64 bytes from 192.168.56.1: icmp_seq=5 ttl=127 time=0.919 ms
64 bytes from 192.168.56.1: icmp_seq=6 ttl=127 time=1.33 ms
64 bytes from 192.168.56.1: icmp_seq=7 ttl=127 time=0.645 ms
64 bytes from 192.168.56.1: icmp_seq=8 ttl=127 time=0.663 ms
64 bytes from 192.168.56.1: icmp_seq=9 ttl=127 time=0.710 ms
```

VICTIM

2.1.3 Task 2: Spoofing

VM1 Attacker Machine: 10.0.2.9

VM2 Victim Machine: 10.0.2.10

```
spooof.py (~/) - gedit
#!/usr/bin/python
from scapy.all import *
print("SENDING SPOOFED ICMP PACKETS...");
IPLayer=IP();
IPLayer.src="10.0.2.6"
IPLayer.dst="192.168.56.1"
ICMPpkt=ICMP()
pkt=IPLayer/ICMPpkt
pkt.show()
send(pkt,verbose=0)
```

spooof.py

Command: sudo python spooof.py

Provide a screenshot of your observations.

Show from Wireshark capture that the live machine sends back an ICMP response.

Command: ping 10.0.2.5

Open Wireshark and observe the ICMP packets as they are being captured.

Provide screenshots of your observations.

```
Terminal
PES2201800368 Nitish@Attacker:~$sudo python spooof.py
SENDING SPOOFED ICMP PACKETS...
###[ IP ]###
version    = 4
ihl        = None
tos        = 0x0
len        = None
id         = 1
flags      = 
frag       = 0
ttl        = 64
proto      = icmp
chksum     = None
src        = 10.0.2.6
dst        = 192.168.56.1
\options   \
###[ ICMP ]###
type       = echo-request
code       = 0
chksum     = None
id         = 0x0
seq        = 0x0
PES2201800368_Nitish@Attacker:~$
```

ATTACKER

Capturing from enp0s3

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	2021-01-29 19:18:24.5997983...	PcsCompu_bf:36:f8	Broadcast	ARP	42	Who has 10.0.2.1? Tell 10.0.2.5
2	2021-01-29 19:18:24.6002605...	RealtekU_12:35:00	PcsCompu_bf:36:f8	ARP	60	10.0.2.1 is at 52:54:00:12:35:00
3	2021-01-29 19:18:24.6023184...	10.0.2.6	192.168.56.1	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (re...
4	2021-01-29 19:18:24.6034520...	192.168.56.1	10.0.2.6	ICMP	60	Echo (ping) reply id=0x0000, seq=0/0, ttl=127 (r...

WIRESHARK CAPTURE FROM ATTACKER

Capturing from enp0s3

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	2021-01-29 19:18:24.5999049...	PcsCompu_bf:36:f8	Broadcast	ARP	60	Who has 10.0.2.1? Tell 10.0.2.5
2	2021-01-29 19:18:24.5999218...	RealtekU_12:35:00	PcsCompu_bf:36:f8	ARP	60	10.0.2.1 is at 52:54:00:12:35:00
3	2021-01-29 19:18:24.6023463...	10.0.2.6	192.168.56.1	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (re...
4	2021-01-29 19:18:24.6033261...	192.168.56.1	10.0.2.6	ICMP	60	Echo (ping) reply id=0x0000, seq=0/0, ttl=127 (r...

WIRESHARK CAPTURE FROM VICTIM BEFORE PING 10.0.2.5

Capturing from enp0s3

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	2021-01-29 19:19:54.6584976...	PcsCompu_bf:36:f8	Broadcast	ARP	60	Who has 10.0.2.1? Tell 10.0.2.5
2	2021-01-29 19:19:54.6585133...	RealtekU_12:35:00	PcsCompu_bf:36:f8	ARP	60	10.0.2.1 is at 52:54:00:12:35:00
3	2021-01-29 19:19:54.6673495...	10.0.2.6	192.168.56.1	ICMP	60	Echo (ping) request id=0x0000, seq=0/0, ttl=64...
4	2021-01-29 19:19:54.6679127...	192.168.56.1	10.0.2.6	ICMP	60	Echo (ping) reply id=0x0000, seq=0/0, ttl=12...
5	2021-01-29 19:20:02.8630969...	10.0.2.6	10.0.2.5	ICMP	98	Echo (ping) request id=0x1040, seq=1/256, ttl=...
6	2021-01-29 19:20:02.8639633...	10.0.2.5	10.0.2.6	ICMP	98	Echo (ping) reply id=0x1040, seq=1/256, ttl=...
7	2021-01-29 19:20:03.8639199...	10.0.2.6	10.0.2.5	ICMP	98	Echo (ping) request id=0x1040, seq=2/512, ttl=...
8	2021-01-29 19:20:03.8650749...	10.0.2.5	10.0.2.6	ICMP	98	Echo (ping) reply id=0x1040, seq=2/512, ttl=...
9	2021-01-29 19:20:04.8653996...	10.0.2.6	10.0.2.5	ICMP	98	Echo (ping) request id=0x1040, seq=3/768, ttl=...
10	2021-01-29 19:20:04.8659262...	10.0.2.5	10.0.2.6	ICMP	98	Echo (ping) reply id=0x1040, seq=3/768, ttl=...
11	2021-01-29 19:20:05.8956694...	10.0.2.6	10.0.2.5	ICMP	98	Echo (ping) request id=0x1040, seq=4/1024, ttl=...
12	2021-01-29 19:20:05.8962620...	10.0.2.5	10.0.2.6	ICMP	98	Echo (ping) reply id=0x1040, seq=4/1024, ttl=...

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
 Ethernet II, Src: PcsCompu_bf:36:f8 (08:00:27:bf:36:f8), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Address Resolution Protocol (request)

0000	ff ff ff ff ff ff 08 00	27 bf 36 f8 08 06 00 01 '.6.....
0010	08 00 06 04 00 01 08 00	27 bf 36 f8 0a 00 02 05 '.6.....
0020	00 00 00 00 00 0a 00 00	02 01 00 00 00 00 00 00
0030	00 00 00 00 00 00 00 00	00 00 00 00

enp0s3: <live capture in progress> Packets: 12 · Displayed: 12 (100.0%) Profile: Default

WIRESHARK CAPTURE FROM VICTIM AFTER PING 10.0.2.5

2.1.4 Task 3: Traceroute



The screenshot shows a code editor window with a dark theme. The menu bar includes File, Edit, View, Search, Tools, Documents, and Help. The status bar at the bottom right shows the time as 21:53. The script content is as follows:

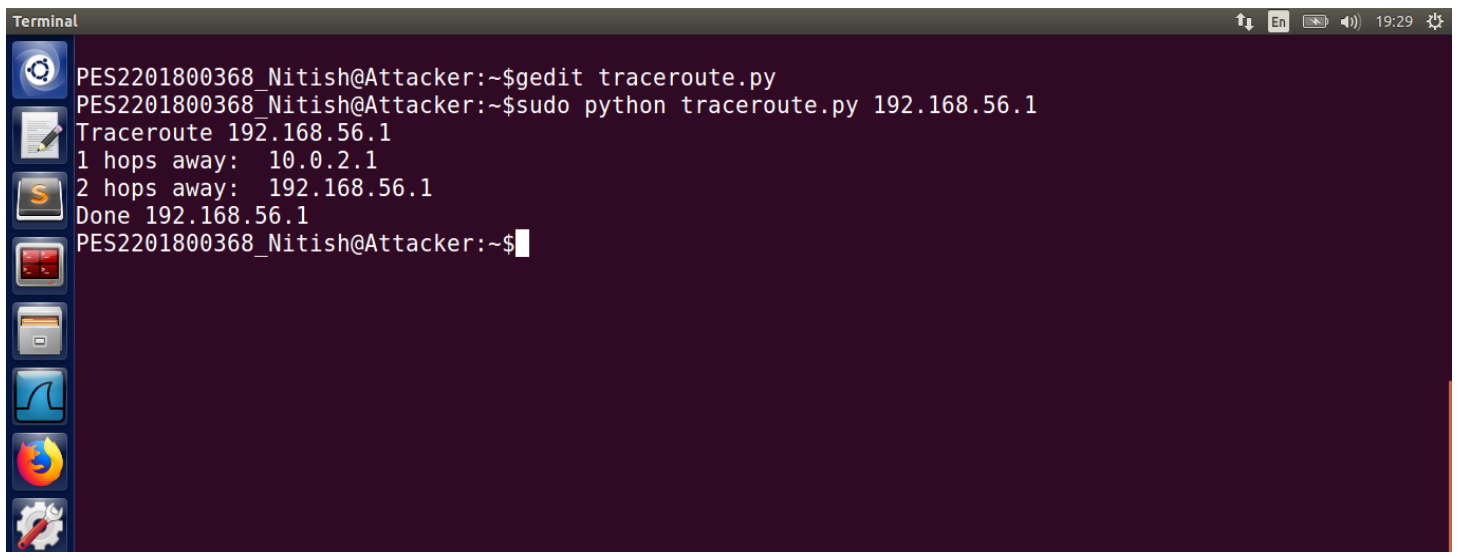
```
from scapy.all import *
'''Usage: ./traceroute.py "hostname or ip address"'''
host=sys.argv[1];
print("Traceroute "+host)
ttl=1
while 1:
    IPlayer=IP()
    IPlayer.dst=host
    IPlayer.ttl=ttl
    ICMPpkt=ICMP()
    pkt=IPlayer/ICMPpkt
    replypkt=sr1(pkt,verbose=0)
    if replypkt is None:
        break
    elif replypkt[ICMP].type==0:
        print "%d hops away: "%ttl,replypkt[IP].src
        print "Done",replypkt[IP].src
        break
    else:
        print "%d hops away: "%ttl,replypkt[IP].src
        ttl+=1
```

A box labeled **traceroute.py** is positioned to the right of the code editor.

Command:

sudo python traceroute.py 192.168.56.1

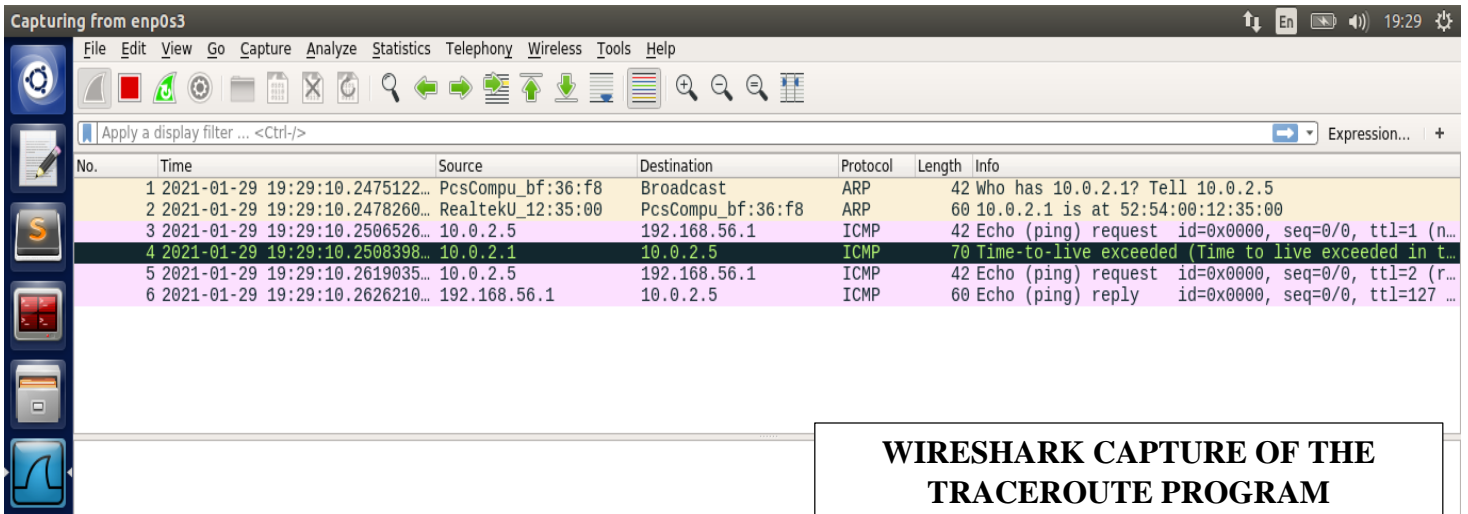
On running the above python code, provide a screenshot of the response.



The screenshot shows a terminal window with a dark background. The command history and output are as follows:

```
PES2201800368_Nitish@Attacker:~$gedit traceroute.py
PES2201800368_Nitish@Attacker:~$sudo python traceroute.py 192.168.56.1
Traceroute 192.168.56.1
1 hops away: 10.0.2.1
2 hops away: 192.168.56.1
Done 192.168.56.1
PES2201800368_Nitish@Attacker:~$
```

Provide a screenshot of the Wireshark capture that shows the ICMP requests sent with increasing TTL and the error response from the routers with a message as “Time to live exceeded”.



WIRESHARK CAPTURE OF THE TRACEROUTE PROGRAM

2.1.5 Task 4: Sniffing and-then Spoofing



sniffspooof.py

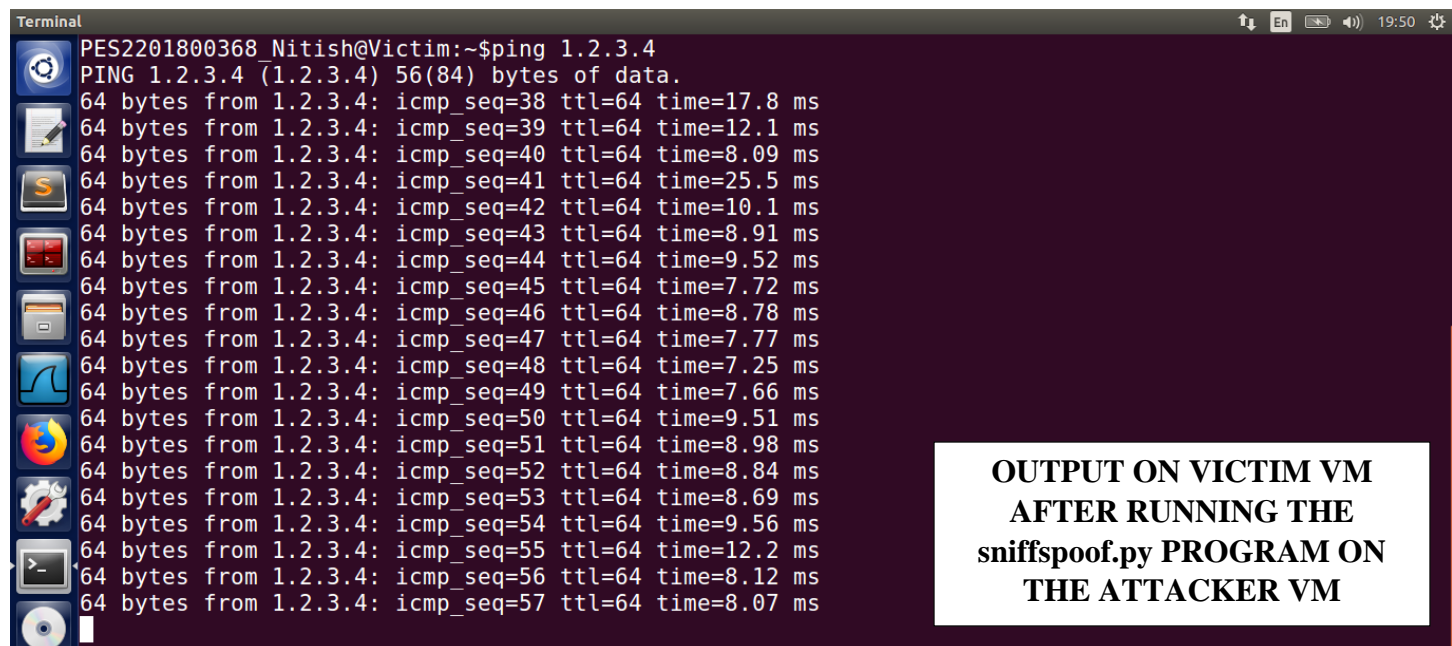
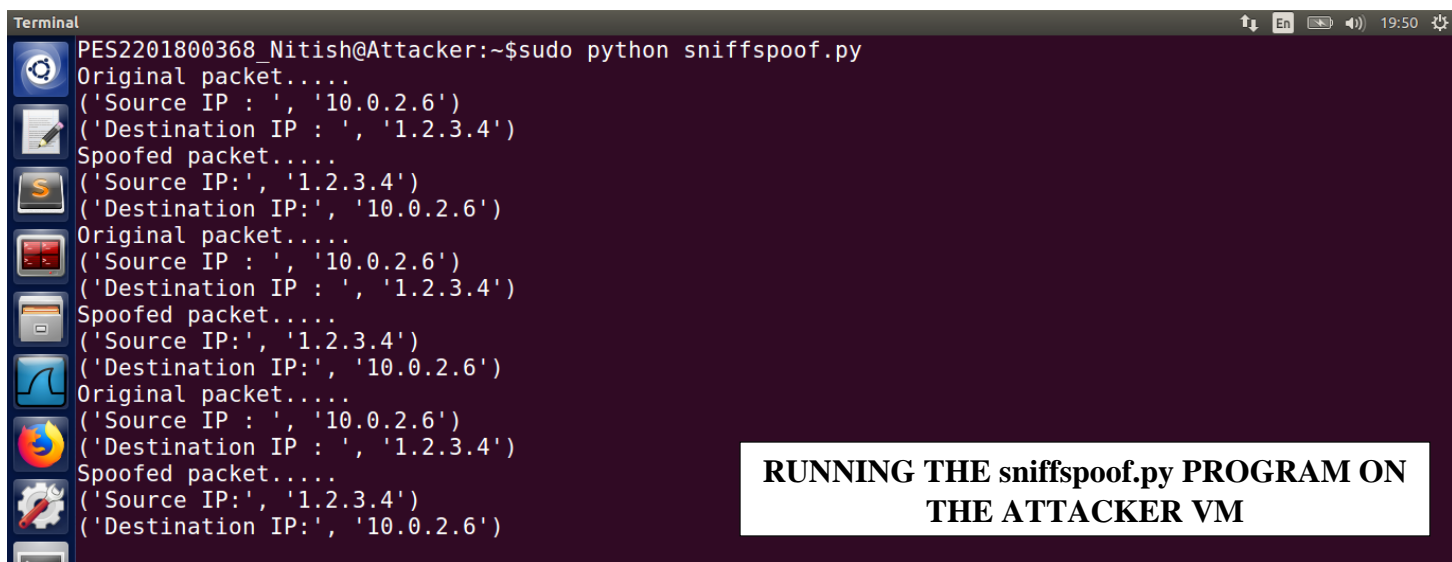
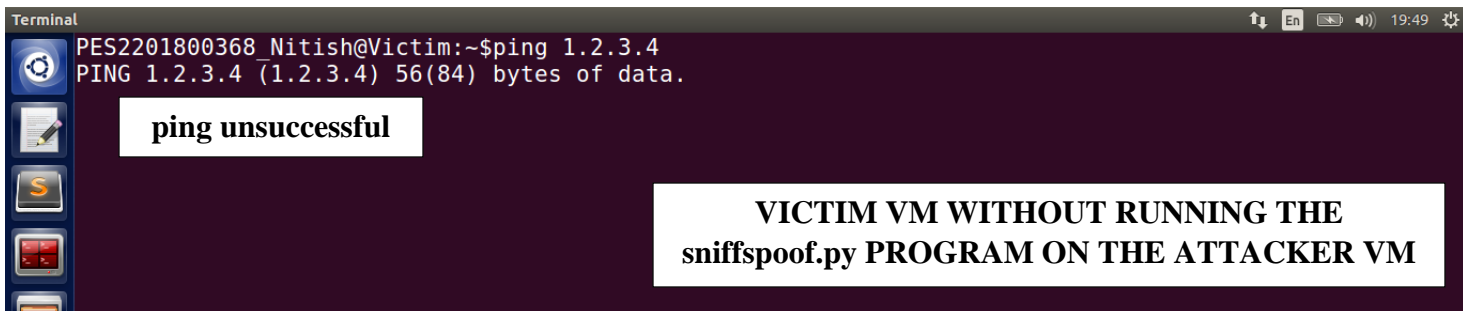
Command: ping 1.2.3.4

Provide a screenshot of your observations.

Open another terminal in the same VM and execute the following command

Command: sudo python sniffspooof.py

Provide a screenshot of your observations.



* On running above code, we see that our spoofer program sends spoofed ICMP responses to the ICMP requests set by the victim machine. The victim machine pings a non-existing IP address, but gets back ICMP response.