

Computer Network Security

Lab 1

Packet sniffing and spoofing

-PES1201801948
-Bharath S Bhambore
-Section H

Lab Setup :

Attacker Machine :

Machine Name : Ubuntu 16.04 [White Terminal]

IP : 10.0.2.8

Victim Machine :

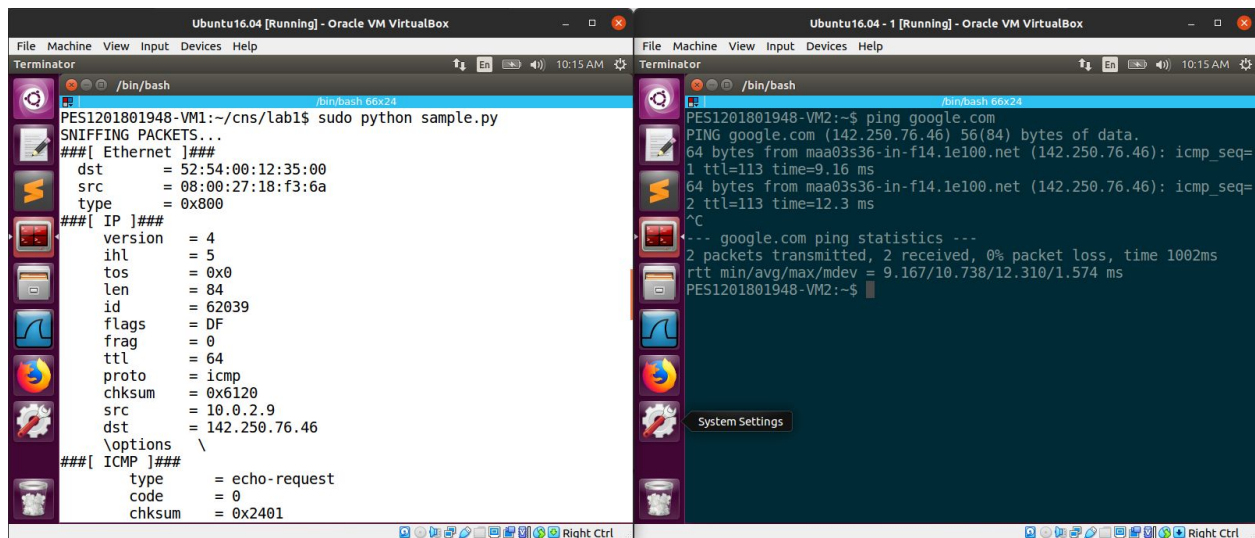
Machine Name : Ubuntu 16.04 -1 [Blue Terminal]

IP : 10.0.2.9

Task 1 : Sniffing Packets

2.1.1 Task 1.1 : Sniffing packets using scapy

Command is run on the attacker machine [10.0.2.8] to sniff the ICMP request packets of the victim machine [10.0.2.9] on the network

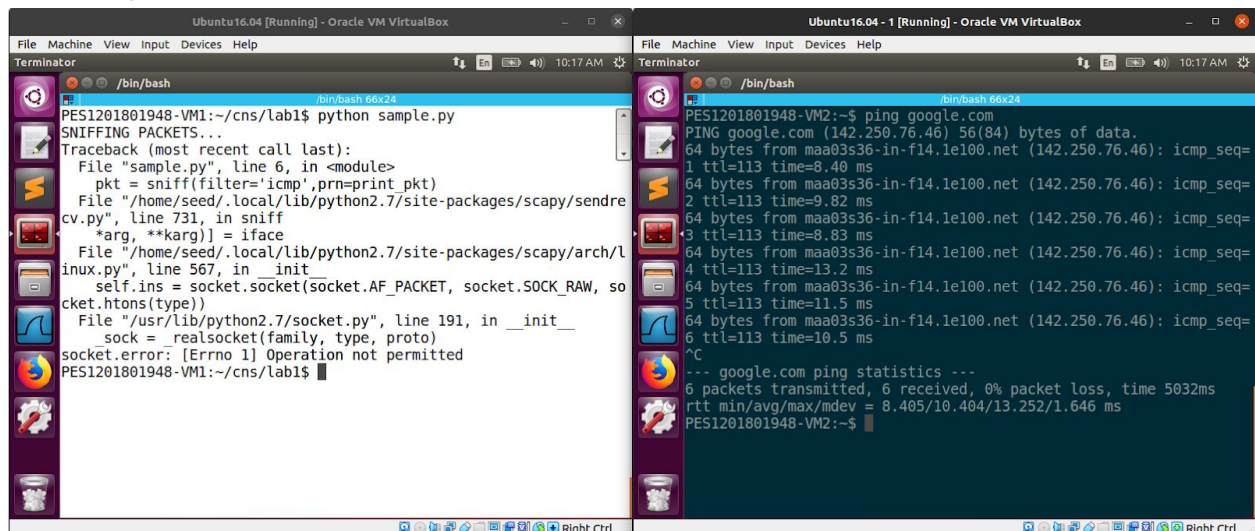


```
Ubuntu16.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
/bin/bash
PES1201801948-VM1:~/cns/lab1$ sudo python sample.py
SNIFFING PACKETS...
###[ Ethernet ]###
  dst      = 52:54:00:12:35:00
  src      = 08:00:27:18:f3:6a
  type     = 0x800
###[ IP ]###
  version  = 4
  ihl      = 5
  tos      = 0x0
  len      = 84
  id       = 62039
  flags    = DF
  frag     = 0
  ttl      = 64
  proto    = icmp
  chksum   = 0x6120
  src      = 10.0.2.9
  dst      = 142.250.76.46
  \options \
###[ ICMP ]###
  type     = echo-request
  code     = 0
  chksum   = 0x2401

Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
/bin/bash
PES1201801948-VM2:~$ ping google.com
PING google.com (142.250.76.46) 56(84) bytes of data:
64 bytes from maa03s36-in-f14.1e100.net (142.250.76.46): icmp_seq=1 ttl=113 time=9.16 ms
64 bytes from maa03s36-in-f14.1e100.net (142.250.76.46): icmp_seq=2 ttl=113 time=12.3 ms
^C
--- google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 9.167/10.738/12.310/1.574 ms
PES1201801948-VM2:~$
```

As we can see in the attacker terminal [Left], it was able to capture the ICMP ping requests of the victim machine.

Running without Sudo :



```
Ubuntu16.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
/bin/bash
PES1201801948-VM1:~/cns/lab1$ python sample.py
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
PES1201801948-VM1:~/cns/lab1$

Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
/bin/bash
PES1201801948-VM2:~$ ping google.com
PING google.com (142.250.76.46) 56(84) bytes of data:
64 bytes from maa03s36-in-f14.1e100.net (142.250.76.46): icmp_seq=1 ttl=113 time=8.40 ms
64 bytes from maa03s36-in-f14.1e100.net (142.250.76.46): icmp_seq=2 ttl=113 time=9.82 ms
64 bytes from maa03s36-in-f14.1e100.net (142.250.76.46): icmp_seq=3 ttl=113 time=8.83 ms
64 bytes from maa03s36-in-f14.1e100.net (142.250.76.46): icmp_seq=4 ttl=113 time=13.2 ms
64 bytes from maa03s36-in-f14.1e100.net (142.250.76.46): icmp_seq=5 ttl=113 time=11.5 ms
64 bytes from maa03s36-in-f14.1e100.net (142.250.76.46): icmp_seq=6 ttl=113 time=10.5 ms
^C
--- google.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5032ms
rtt min/avg/max/mdev = 8.405/10.404/13.252/1.646 ms
PES1201801948-VM2:~$
```

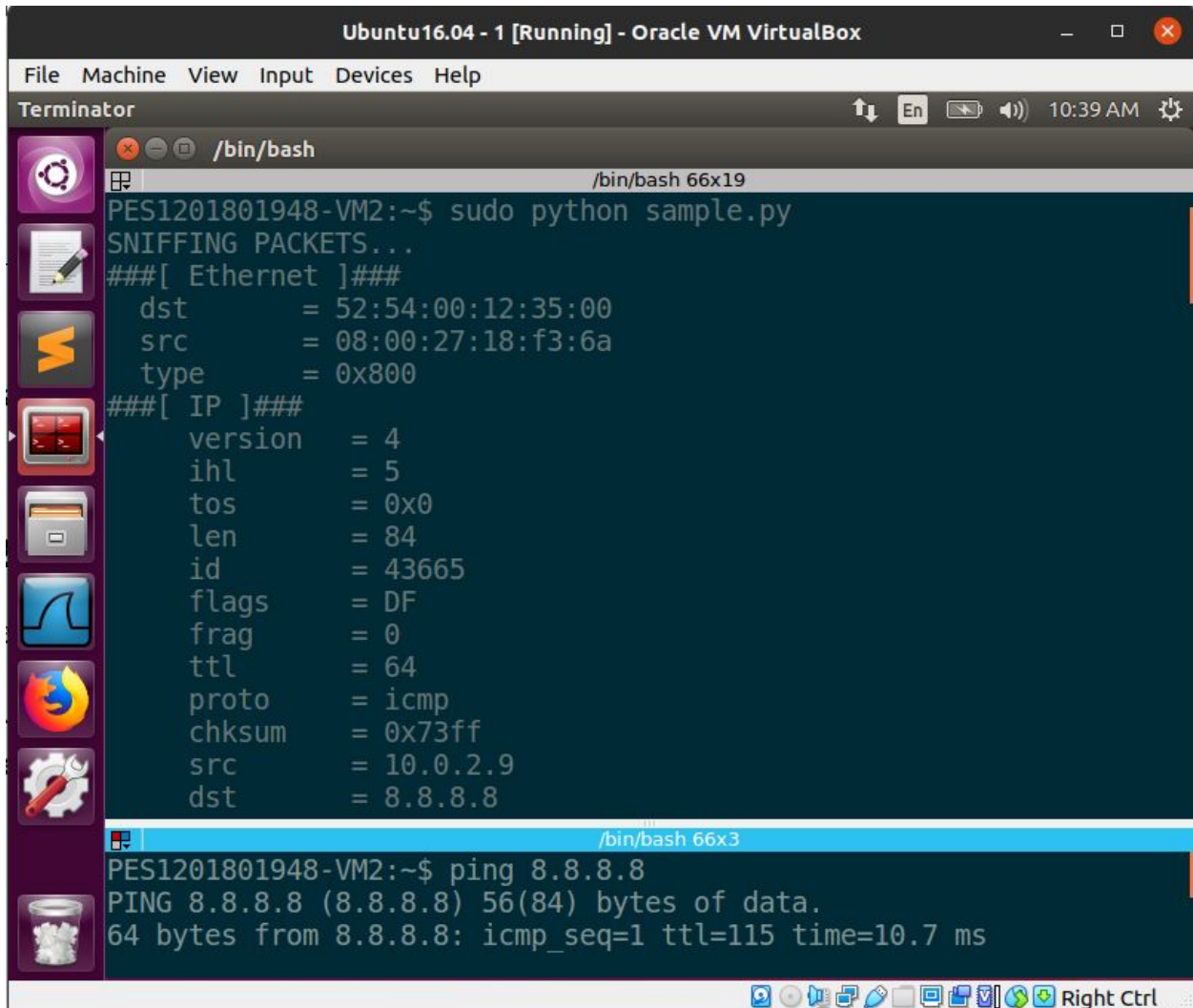
Error : Operation not permitted

Sniffing requires root privileges therefore the socket itself is not initialized, hence why it failed to capture the icmp packets

2.1.2 Task 1.2 Capturing ICMP, TCP packet and Subnet

2.1.2.1 Capture only the ICMP packet

sample.py is run with the ICMP packet filter, ping is also run on the same machine



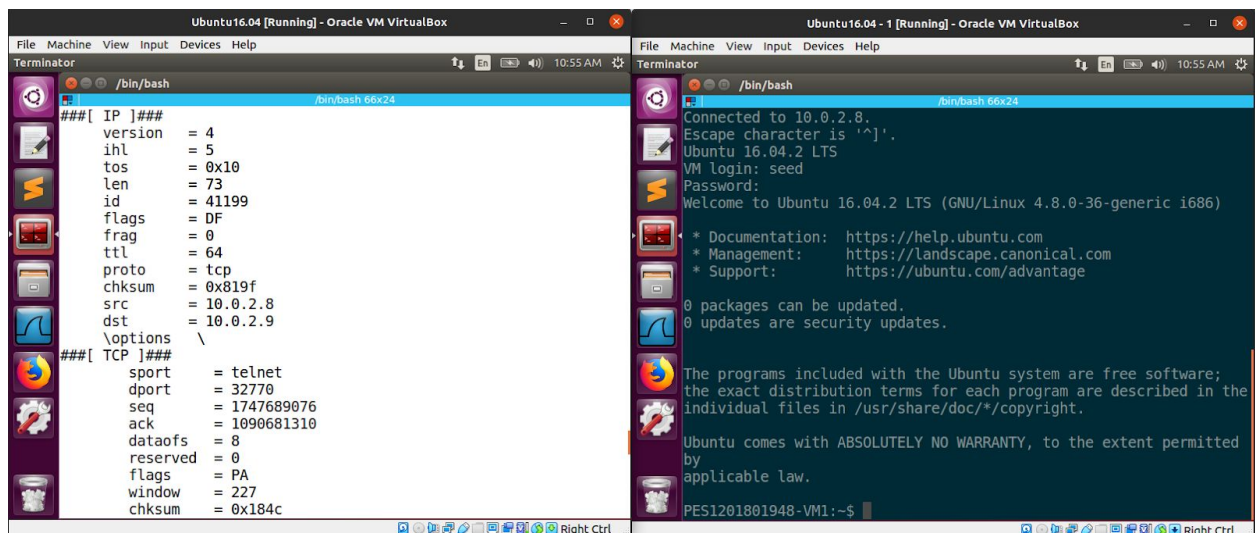
The screenshot shows a VirtualBox window titled "Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox". Inside, a "Terminator" terminal window is open, displaying the output of a Python script and a ping command. The terminal has a dark blue background and a light blue title bar. The output of the script shows packet details for an ICMP echo request from 10.0.2.9 to 8.8.8.8. Below this, the output of a ping command is shown, indicating a successful connection with a time of 10.7 ms.

```
Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator 10:39 AM
/bin/bash
PES1201801948-VM2:~$ sudo python sample.py
SNIFFING PACKETS...
###[ Ethernet ]###
  dst      = 52:54:00:12:35:00
  src      = 08:00:27:18:f3:6a
  type     = 0x800
###[ IP ]###
  version  = 4
  ihl      = 5
  tos      = 0x0
  len      = 84
  id       = 43665
  flags    = DF
  frag     = 0
  ttl      = 64
  proto    = icmp
  checksum = 0x73ff
  src      = 10.0.2.9
  dst      = 8.8.8.8
/bin/bash 66x19
PES1201801948-VM2:~$ 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=115 time=10.7 ms
/bin/bash 66x3
```

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

Telnet request uses TCP packets on port 23, therefore it is run on the Victim Machine [10.0.2.9], while the python script is run on the attacker machine [10.0.2.8] to sniff out the TCP packets.

Telnet is used to virtually access another machine and provide a 2 way channel between them.



The image shows two side-by-side terminal windows from Oracle VM VirtualBox. The left window is titled 'Ubuntu16.04 [Running] - Oracle VM VirtualBox' and shows a telnet session. The right window is titled 'Ubuntu16.04-1 [Running] - Oracle VM VirtualBox' and shows the telnet client's output.

```
###[ IP ]###
version      = 4
ihl          = 5
tos          = 0x10
len          = 73
id           = 41199
flags        = DF
frag         = 0
ttl          = 64
proto        = tcp
chksum       = 0x819f
src          = 10.0.2.8
dst          = 10.0.2.9
\options     \
###[ TCP ]###
sport        = telnet
dport        = 32770
seq          = 1747689076
ack          = 1090681310
dataofs      = 8
reserved     = 0
flags        = PA
window       = 227
chksum       = 0x184c
```

```
Connected to 10.0.2.8.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM 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

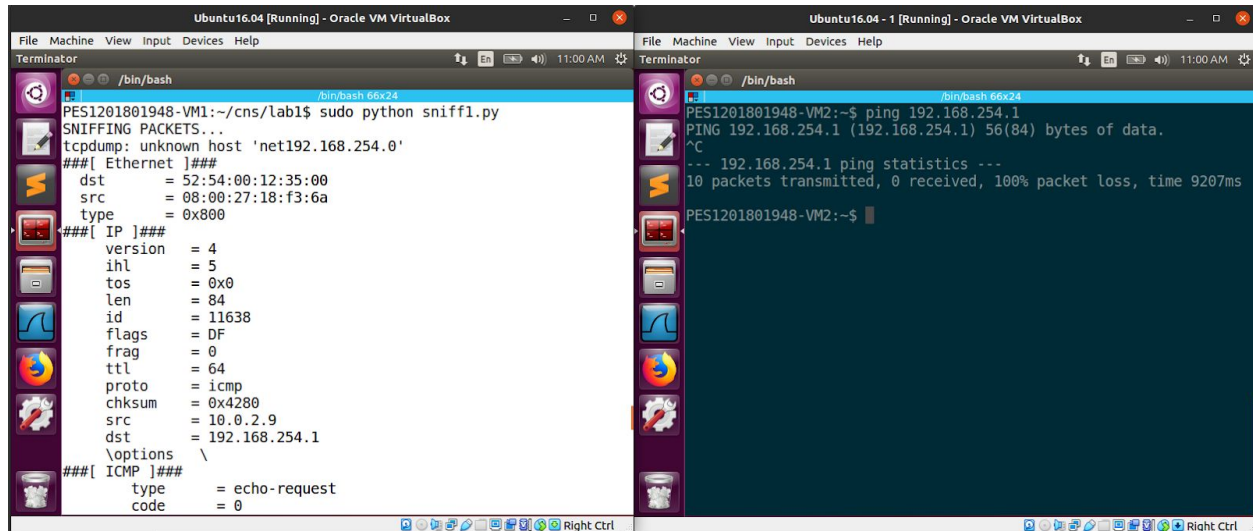
0 packages can be updated.
0 updates are security updates.

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.

PES1201801948-VM1:~$
```

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



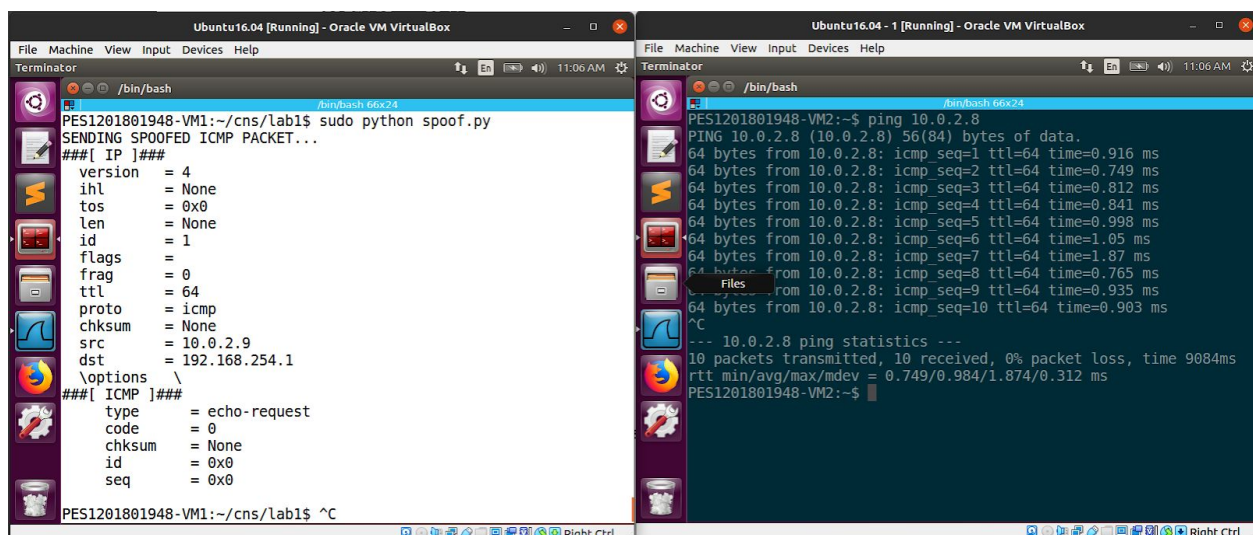
The left screenshot shows a terminal window titled 'Ubuntu16.04 [Running] - Oracle VM VirtualBox'. The user runs 'sudo python sniff1.py'. The output shows a sniffed packet from an Ethernet interface with IP 10.0.2.9 to 192.168.254.1. The right screenshot shows a terminal window titled 'Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox'. The user runs 'ping 192.168.254.1'. The output shows a successful ping with 10 packets transmitted, 0 received, and 100% packet loss, which is unusual for a successful ping.

```
File Machine View Input Devices Help
Terminator
/bin/bash
PES1201801948-VM1:~/cns/lab1$ sudo python sniff1.py
SNIFFING PACKETS...
tcpdump: unknown host 'net192.168.254.0'
###[ Ethernet ]###
  dst      = 52:54:00:12:35:00
  src      = 08:00:27:18:f3:6a
  type     = 0x800
###[ IP ]###
  version  = 4
  ihl      = 5
  tos      = 0x0
  len      = 84
  id       = 11638
  flags    = DF
  frag     = 0
  ttl      = 64
  proto    = icmp
  chksum   = 0x4280
  src      = 10.0.2.9
  dst      = 192.168.254.1
  \options \
###[ ICMP ]###
  type     = echo-request
  code     = 0

File Machine View Input Devices Help
Terminator
/bin/bash
PES1201801948-VM2:~$ ping 192.168.254.1
PING 192.168.254.1 (192.168.254.1) 56(84) bytes of data.
^C
--- 192.168.254.1 ping statistics ---
10 packets transmitted, 0 received, 100% packet loss, time 9207ms
PES1201801948-VM2:~$
```

192.168.254.1 is the router's IP address that we ping from the victim machine, while the attacker machine is able to capture the packets transferred in the subnet specified.

2.1.3 Task 2: Spoofing

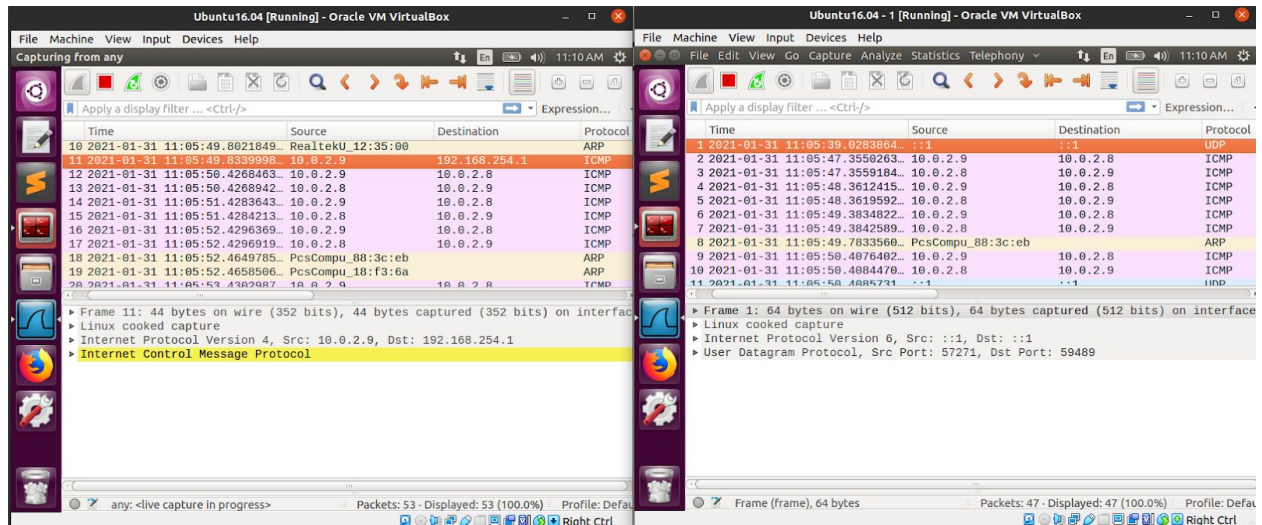


The left screenshot shows a terminal window titled 'Ubuntu16.04 [Running] - Oracle VM VirtualBox'. The user runs 'sudo python spoof.py'. The output shows a spoofed ICMP packet from 10.0.2.9 to 192.168.254.1. The right screenshot shows a terminal window titled 'Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox'. The user runs 'ping 10.0.2.8'. The output shows a successful ping with 10 packets transmitted, 10 received, and 0% packet loss.

```
File Machine View Input Devices Help
Terminator
/bin/bash
PES1201801948-VM1:~/cns/lab1$ sudo python spoof.py
SENDING SPOOFED ICMP PACKET...
###[ IP ]###
  version  = 4
  ihl      = None
  tos      = 0x0
  len      = None
  id       = 1
  flags    = 
  frag     = 0
  ttl      = 64
  proto    = icmp
  chksum   = None
  src      = 10.0.2.9
  dst      = 192.168.254.1
  \options \
###[ ICMP ]###
  type     = echo-request
  code     = 0
  chksum   = None
  id       = 0x0
  seq      = 0x0
PES1201801948-VM1:~/cns/lab1$ ^C

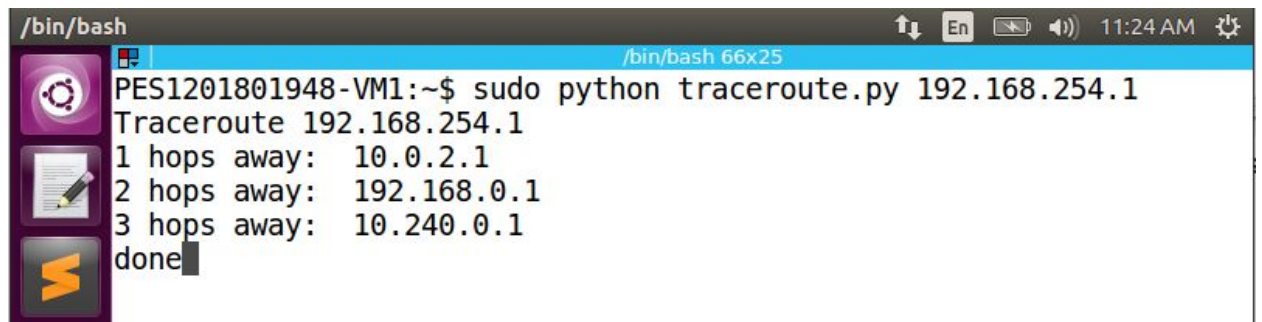
File Machine View Input Devices Help
Terminator
/bin/bash
PES1201801948-VM2:~$ ping 10.0.2.8
PING 10.0.2.8 (10.0.2.8) 56(84) bytes of data.
64 bytes from 10.0.2.8: icmp_seq=1 ttl=64 time=0.916 ms
64 bytes from 10.0.2.8: icmp_seq=2 ttl=64 time=0.749 ms
64 bytes from 10.0.2.8: icmp_seq=3 ttl=64 time=0.812 ms
64 bytes from 10.0.2.8: icmp_seq=4 ttl=64 time=0.841 ms
64 bytes from 10.0.2.8: icmp_seq=5 ttl=64 time=0.998 ms
64 bytes from 10.0.2.8: icmp_seq=6 ttl=64 time=1.05 ms
64 bytes from 10.0.2.8: icmp_seq=7 ttl=64 time=1.87 ms
64 bytes from 10.0.2.8: icmp_seq=8 ttl=64 time=0.765 ms
64 bytes from 10.0.2.8: icmp_seq=9 ttl=64 time=0.935 ms
64 bytes from 10.0.2.8: icmp_seq=10 ttl=64 time=0.903 ms
^C
--- 10.0.2.8 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9084ms
rtt min/avg/max/mdev = 0.749/0.984/1.874/0.312 ms
PES1201801948-VM2:~$
```


Wireshark capture of both machines :



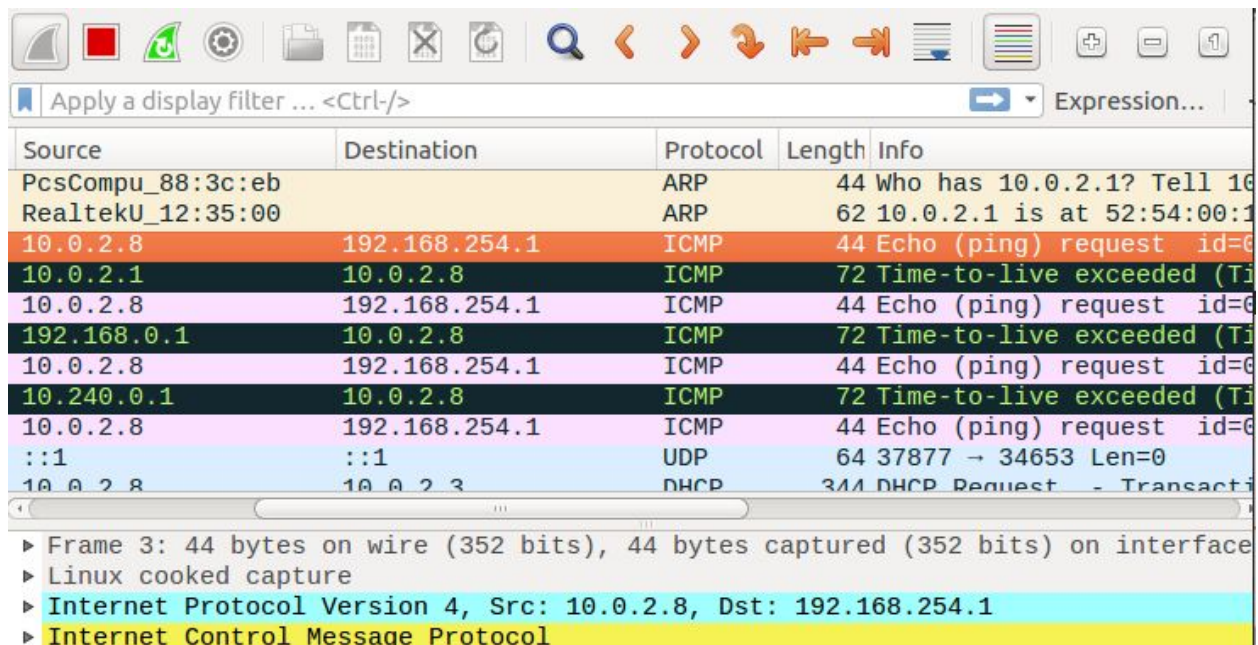
We can see that the destination is successfully spoofed in this attack.

2.1.4 Task 3: Traceroute



Traceroute of the router is found using the above script

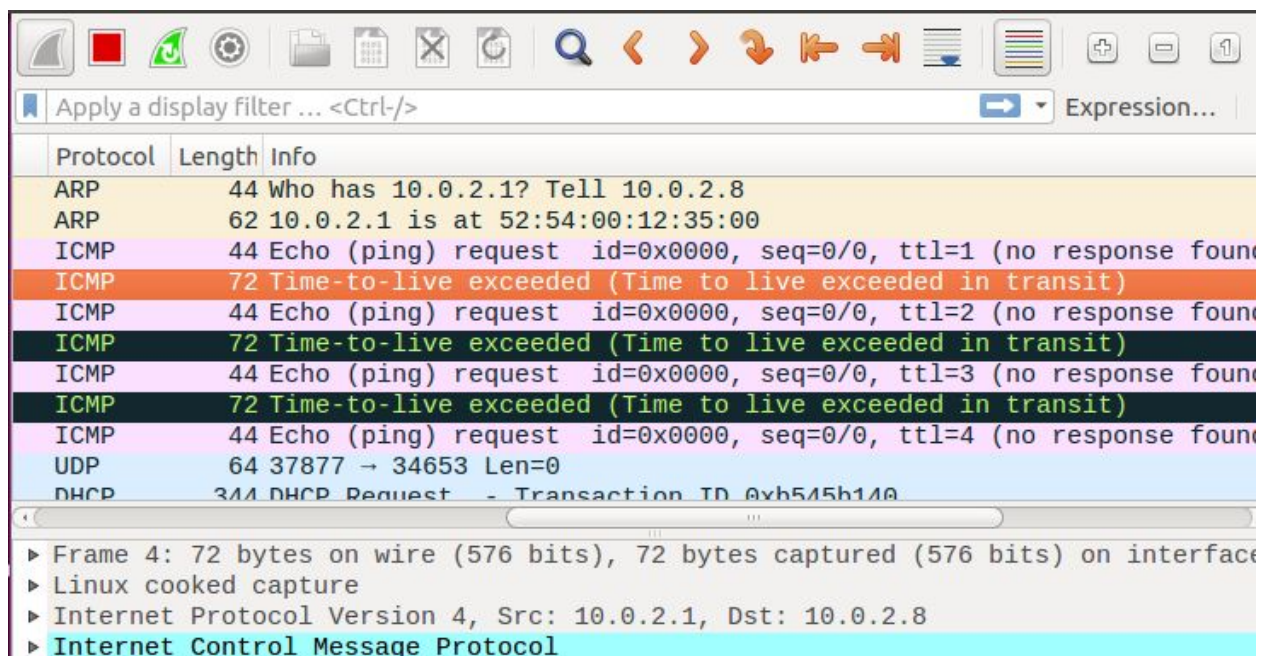
Wireshark capture :



Source	Destination	Protocol	Length	Info
PcsCompu_88:3c:eb		ARP	44	Who has 10.0.2.1? Tell 10.0.2.8
RealtekU_12:35:00		ARP	62	10.0.2.1 is at 52:54:00:12:35:00
10.0.2.8	192.168.254.1	ICMP	44	Echo (ping) request id=0
10.0.2.1	10.0.2.8	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
10.0.2.8	192.168.254.1	ICMP	44	Echo (ping) request id=0
192.168.0.1	10.0.2.8	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
10.0.2.8	192.168.254.1	ICMP	44	Echo (ping) request id=0
10.240.0.1	10.0.2.8	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
10.0.2.8	192.168.254.1	ICMP	44	Echo (ping) request id=0
::1	::1	UDP	64	37877 → 34653 Len=0
10.0.2.8	10.0.2.1	DHCP	344	DHCP Request - Transaction ID 0xb545b140

Frame 3: 44 bytes on wire (352 bits), 44 bytes captured (352 bits) on interface
Linux cooked capture
Internet Protocol Version 4, Src: 10.0.2.8, Dst: 192.168.254.1
Internet Control Message Protocol

TTL exceeded response from router



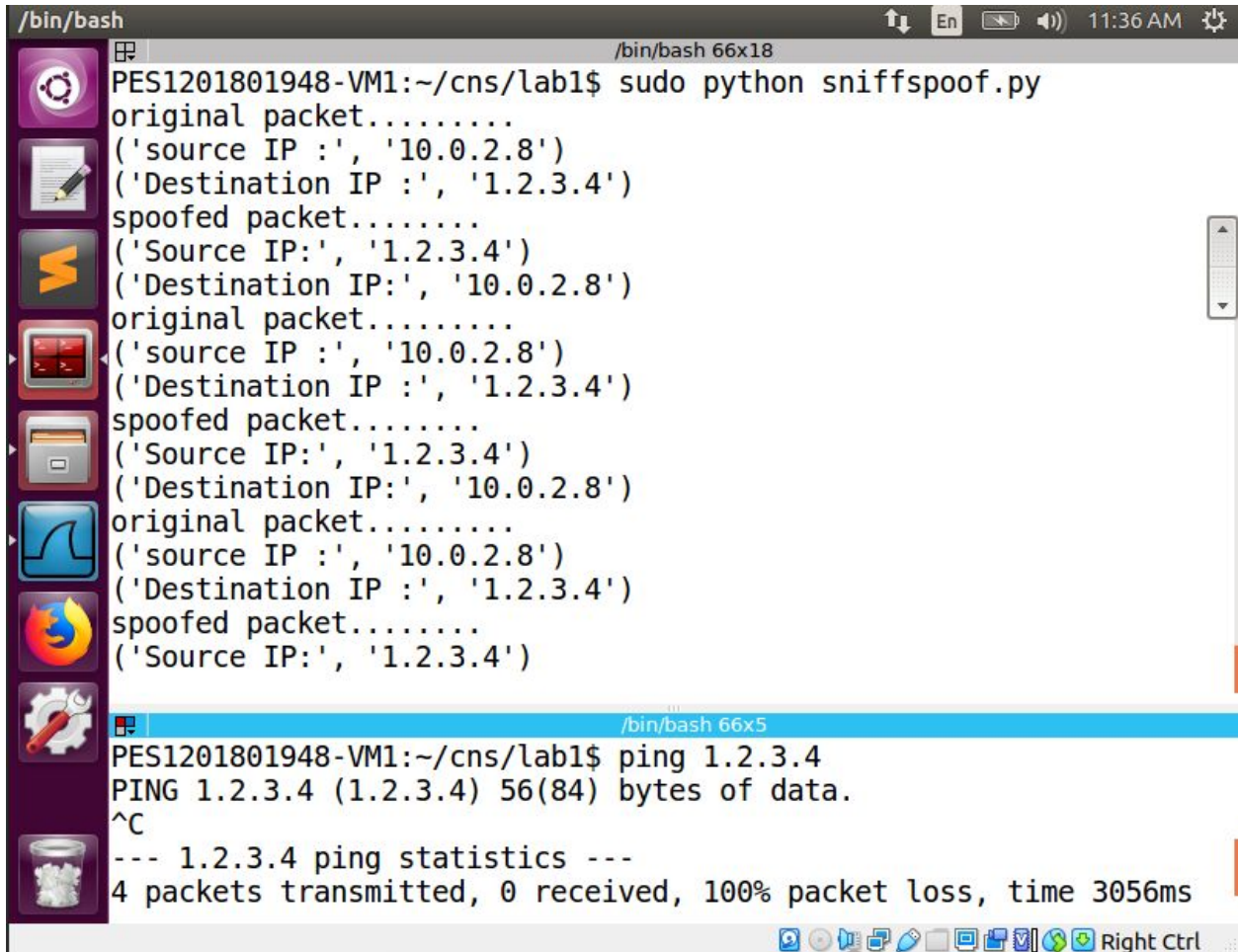
Protocol	Length	Info
ARP	44	Who has 10.0.2.1? Tell 10.0.2.8
ARP	62	10.0.2.1 is at 52:54:00:12:35:00
ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=1 (no response found)
ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=2 (no response found)
ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=3 (no response found)
ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)
ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=4 (no response found)
UDP	64	37877 → 34653 Len=0
DHCP	344	DHCP Request - Transaction ID 0xb545b140

Frame 4: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface
Linux cooked capture
Internet Protocol Version 4, Src: 10.0.2.1, Dst: 10.0.2.8
Internet Control Message Protocol

As we can see, the TTL of the icmp packets is increasing

2.1.5 Task 4: Sniffing and-then Spoofing

Running in the same machine itself



The screenshot shows a terminal window titled "/bin/bash" with a system menu on the left. The terminal output shows the execution of a Python script named "sniffspoof.py". The script displays three pairs of "original packet" and "spoofed packet" information. Each pair shows the source and destination IP addresses. The original packets have source IP '10.0.2.8' and destination IP '1.2.3.4'. The spoofed packets have source IP '1.2.3.4' and destination IP '10.0.2.8'. Below this, the user runs a "ping 1.2.3.4" command, which shows a 100% packet loss. The terminal window has a status bar at the bottom with system icons and the text "Right Ctrl".

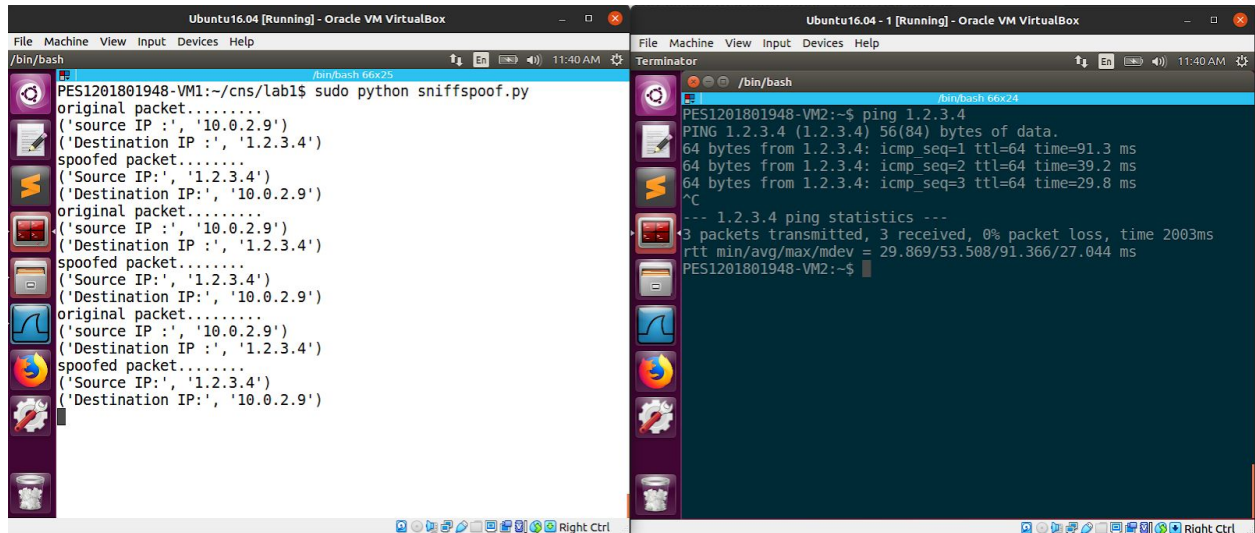
```
/bin/bash
PES1201801948-VM1:~/cns/lab1$ sudo python sniffspoof.py
original packet.....
('source IP :', '10.0.2.8')
('Destination IP :', '1.2.3.4')
spoofed packet.....
('Source IP:', '1.2.3.4')
('Destination IP:', '10.0.2.8')
original packet.....
('source IP :', '10.0.2.8')
('Destination IP :', '1.2.3.4')
spoofed packet.....
('Source IP:', '1.2.3.4')
('Destination IP:', '10.0.2.8')
original packet.....
('source IP :', '10.0.2.8')
('Destination IP :', '1.2.3.4')
spoofed packet.....
('Source IP:', '1.2.3.4')

PES1201801948-VM1:~/cns/lab1$ ping 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
^C
--- 1.2.3.4 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3056ms
```

We can also sniff and spoof packets sent from the victim machine instead of testing it on the same machine only.

As shown below,

The attack runs successfully implying that the victim, even though it pinged a non-existent IP address, the attacker was able to send spoofed ICMP packets.



```
Ubuntu16.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
/bin/bash
PES1201801948-VM1:~/cns/lab1$ sudo python sniffspoof.py
original packet.....
('source IP:', '10.0.2.9')
('Destination IP:', '1.2.3.4')
spoofed packet.....
('Source IP:', '1.2.3.4')
('Destination IP:', '10.0.2.9')
original packet.....
('source IP:', '10.0.2.9')
('Destination IP:', '1.2.3.4')
spoofed packet.....
('Source IP:', '1.2.3.4')
('Destination IP:', '10.0.2.9')
original packet.....
('source IP:', '10.0.2.9')
('Destination IP:', '1.2.3.4')
spoofed packet.....
('Source IP:', '1.2.3.4')
('Destination IP:', '10.0.2.9')

Ubuntu16.04 - 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminator
/bin/bash
PES1201801948-VM2:~$ 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=64 time=91.3 ms
64 bytes from 1.2.3.4: icmp_seq=2 ttl=64 time=39.2 ms
64 bytes from 1.2.3.4: icmp_seq=3 ttl=64 time=29.8 ms
^C
--- 1.2.3.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 29.869/53.508/91.366/27.044 ms
PES1201801948-VM2:~$
```