COMPUTER NETWORK SECURITY LAB

LAB 1: SNIFFING AND SPOOFING

OBJECTIVE

a) Understanding the concepts of Sniffing and Spoofing
b) Using tools to Sniff and Spoof packets using Scapy

INTRODUCTION

Packet sniffing and spoofing are the two important concepts in network security; they are two major threats in network communication. Being able to understand these two threats is essential for understanding security measures in networking. There are many packets sniffing and spoofing tools, such as Wireshark, Tcpdump, Netwox, etc.

Sniffing is the process in which all the data packets passing in the network are monitored. Sniffers are usually used by network administrators to monitor and troubleshoot the network traffic. Whereas attackers use Sniffers to monitor and capture data packets to steal sensitive information containing password and user accounts.

Spoofing is the process in which an intruder introduces fake traffic and pretends to be someone else (legal source or the legitimate entity). Spoofing is done by sending packets with incorrect source address over the network.

EXECUTION

Using Tools to Sniff and Spoof Packets using Scapy

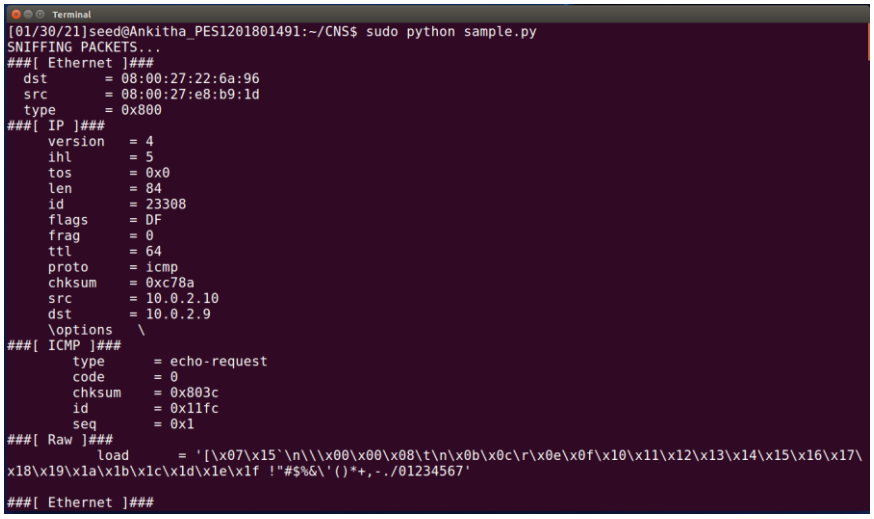
Attacker Machine: 10.0.2.9

Victim Machine: 10.0.2.10

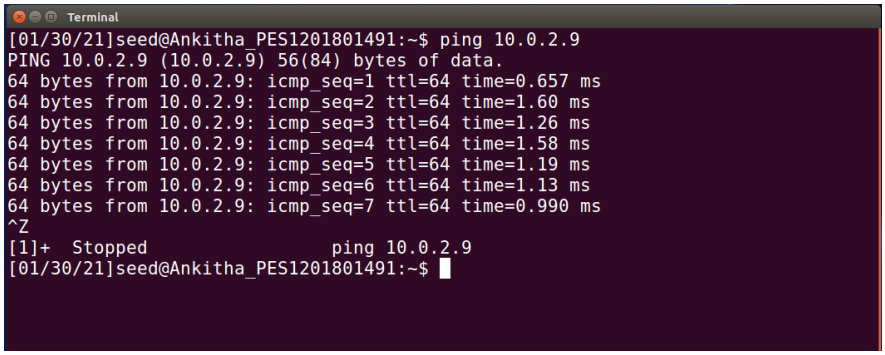
2.1 Task 1: Sniffing Packets

2.1.1 Task 1.1 Sniff IP packets using Scapy

Attacker:



Victim:



OSERVATIONS:

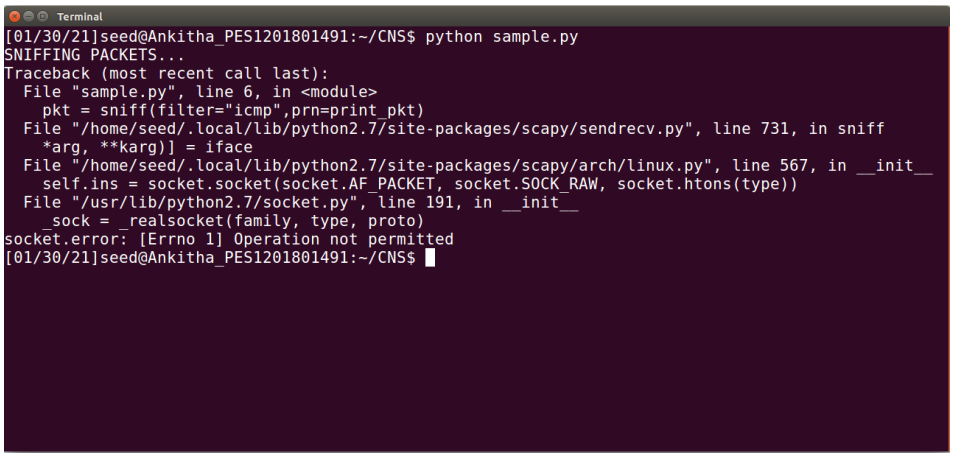
- The program code acts as a sniffer and captures the ping requests sent by another machine on the same network.
- The callback function in the sniff function displays the source, destination IP address and the protocol.

Explain on which VM you ran the command “sudo python sample.py “ and why?

The command “sudo python sample.py” is run on the Attacker side. The program code acts as a sniffer and captures the ping requests sent by the victim machine. i.e in this case, the victim(10.0.2.1) pings to the machine 10.0.2.9 and the attacker sniffs all the packets by the victim displaying all the packet details.

Now, we run the same program without root privileges. Do you find any issues? If so, why?
When we try to run the same program without root privileges, it does not execute successfully. This is because certain applications, files, packages etc has certain security privileges that does not

anyone in the system to open the raw sockets. Thus can be accessed only with root privileges. The following image shows the error

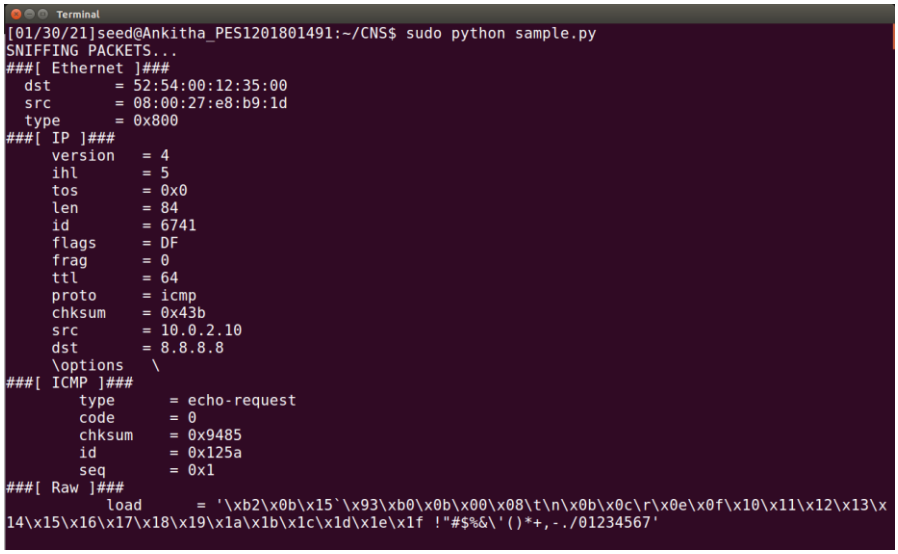


2.1.2 Task 1.2 Capturing ICMP, TCP packet and Subnet

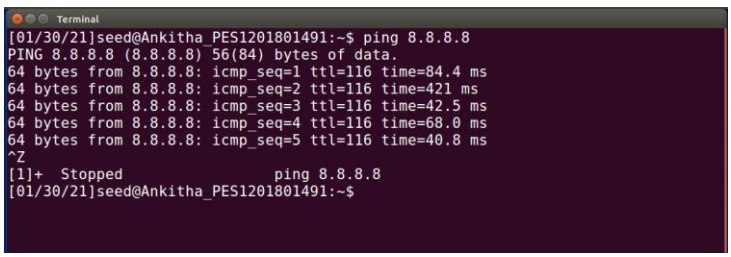
2.1.2.1 Capture only the ICMP packet

When victim pings to 10.0.2.9

Attacker:

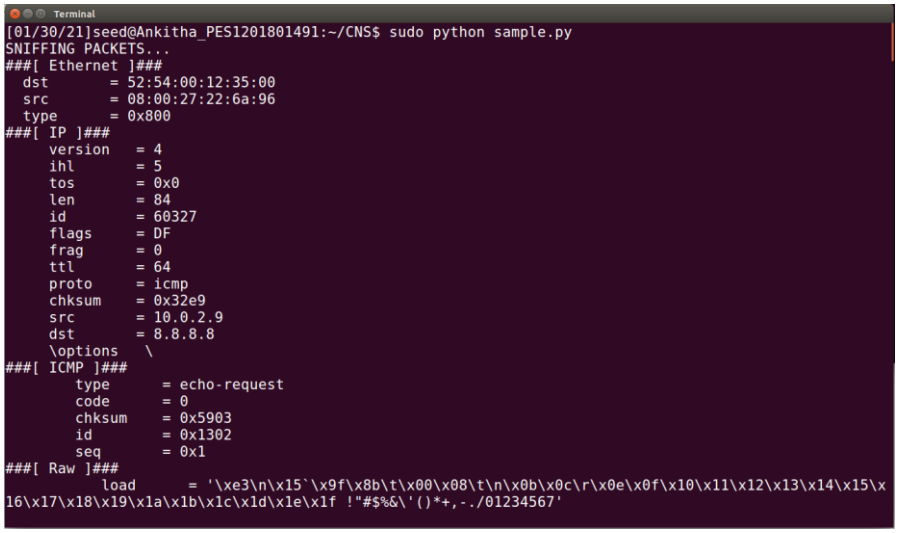


Victim:

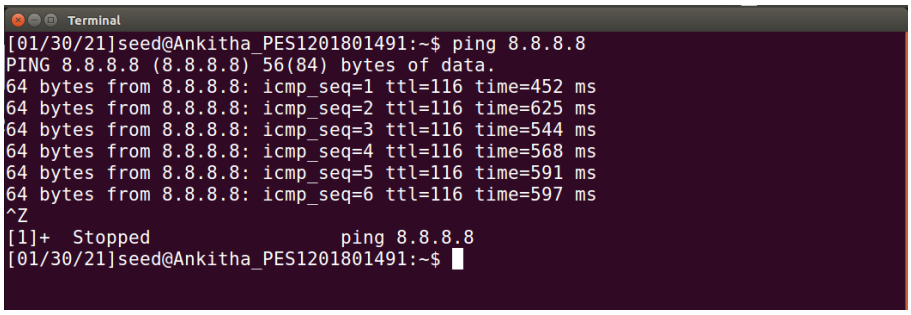


When attacker pings to 8.8.8.8

Attacker:



Victim:



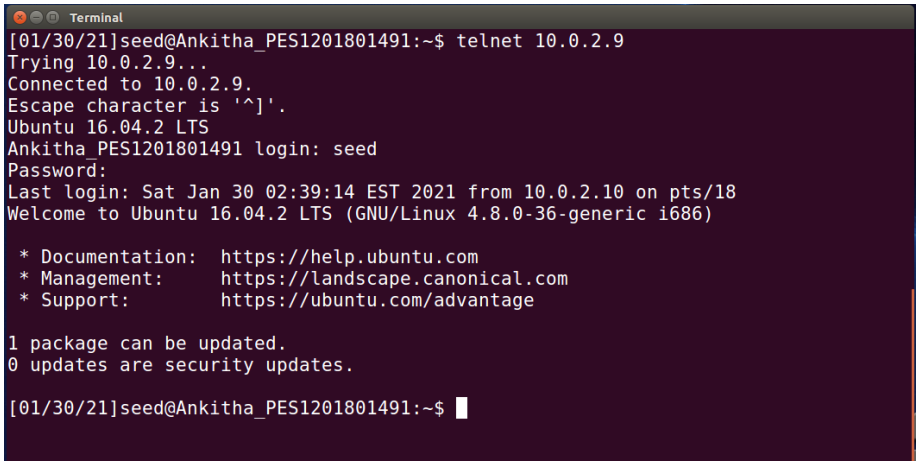
OBSERVATIONS:

- The sniffer program code on the Attacker side has filter is only for the ICMP packets. Hence, when some machine on the same network sends ping requests, the packets get captured by the sniffer.
- Here when the victim sends a ping request to 8.8.8.8, the attacker sniffs all the icmp packets and gets all the packet details.

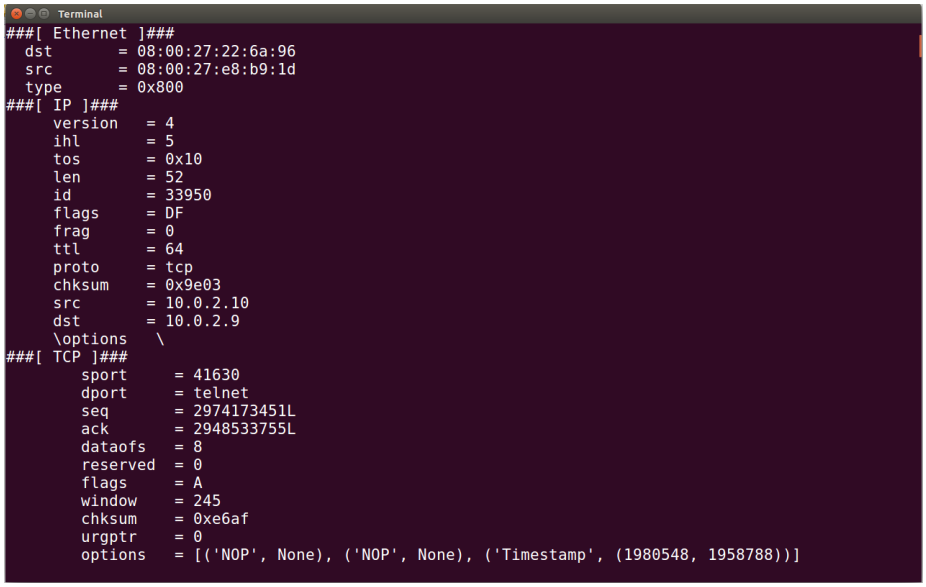
2.1.2.2 Capture any TCP packet that comes from a particular IP and with a destination port number 23

The attacker sniffer code sniffs the TCP traffic from a specific host (10.0.2.10) to port 23.

Victim:



Attacker:



OBSERVATIONS:

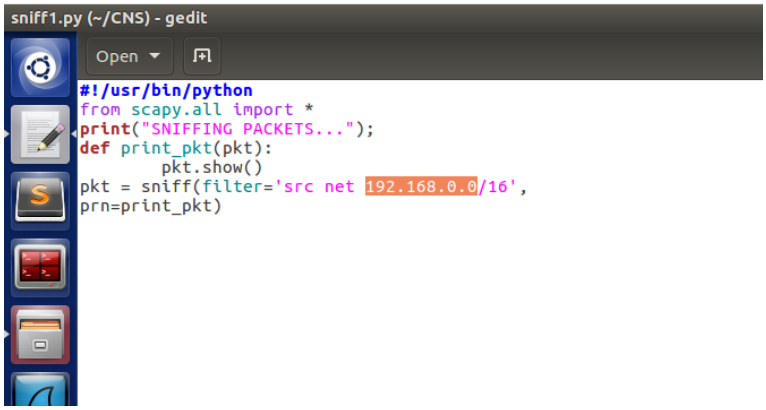
- The attacker sniffs all the TCP traffic from the victim that sends a telnet request to 10.0.2.9.
- From the sniffed packet details we observe the following:
  Protocol: TCP
  IP version: 4
  Sport: 41630
  Dport: telnet (port no 23)

Explain where you will run Telnet.

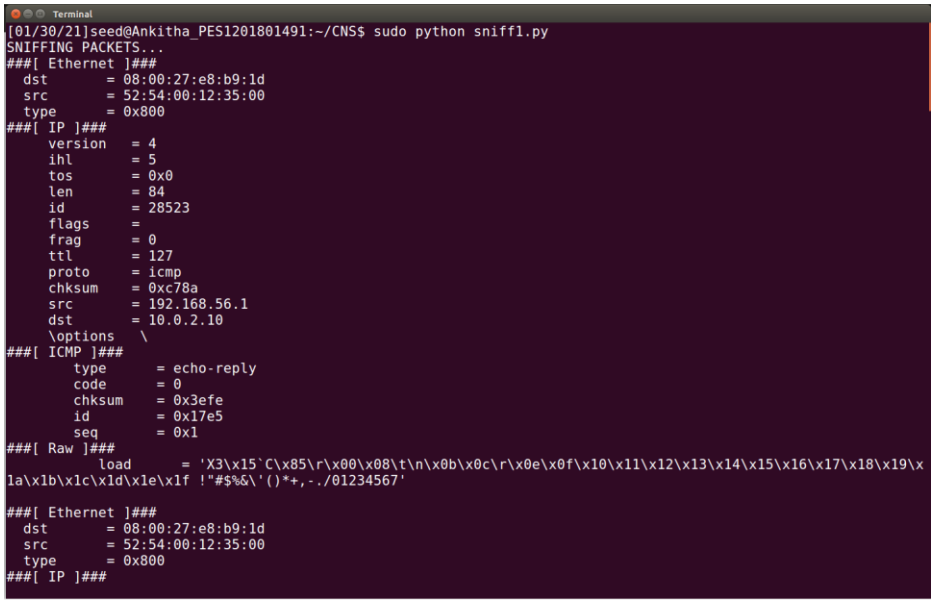
The telnet is run on the victim machine which sends a telnet request to the ip 10.0.2.9 over the port number 23.

2.1.2.2.1 Capture packets that comes from or go to a particular subnet

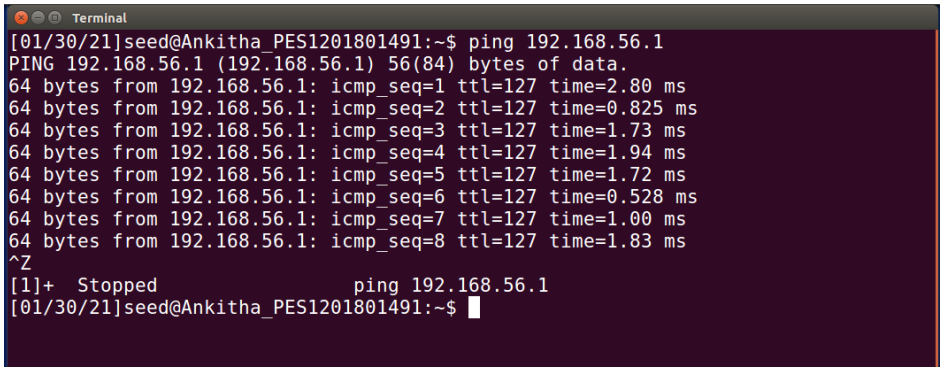
The subnet chosen for trial is the subnet of which host machine is part of. i.e 192.168.0.0/16



Attacker:



Victim:



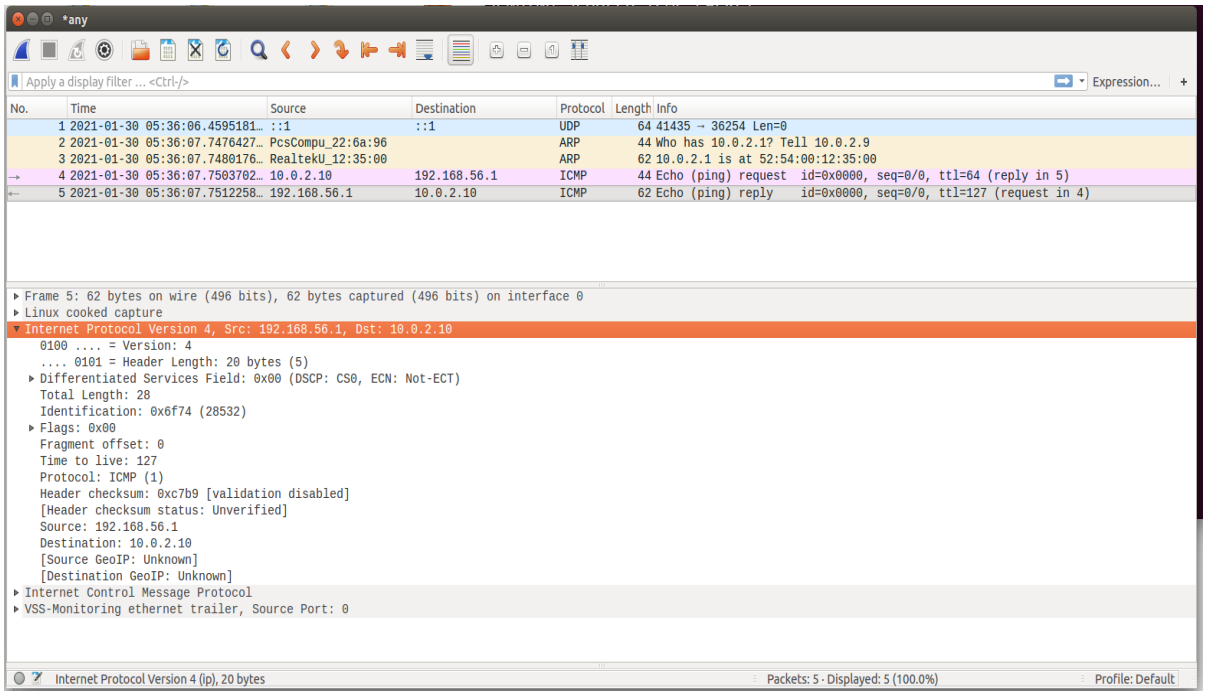
OBSERVATIONS:

- When the victim sends ICMP packets to 192.168.254.1, the sniffer program captures the packets sent out from 192.168.254.1 and thus retrieves the packet details

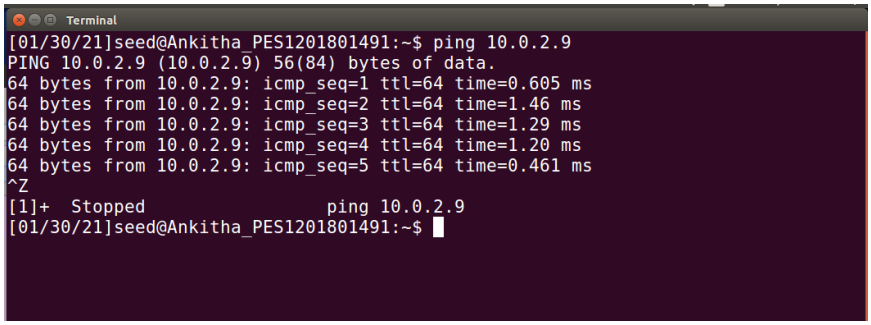
2.1.3 Task 2: Spoofing

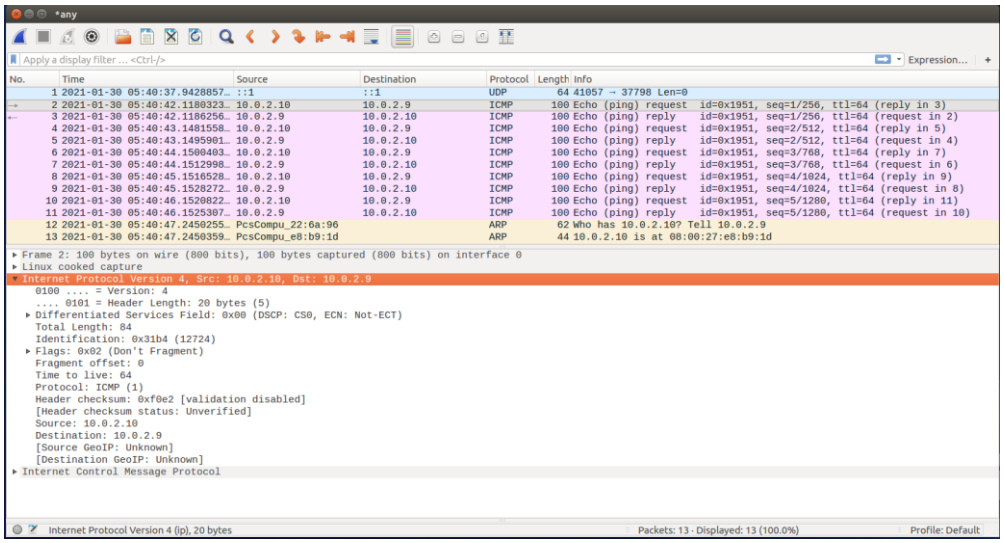
The program code spoofs ICMP echo request packets and sends them to another VM on the same network. The spoofed request is formed by creating our own packet with the header specifications. Here we create an ICMP header. Similarly, we fill the IP header with source IP address of any machine within the local network and destination IP address of any remote machine on the internet which is alive.

Attacker:

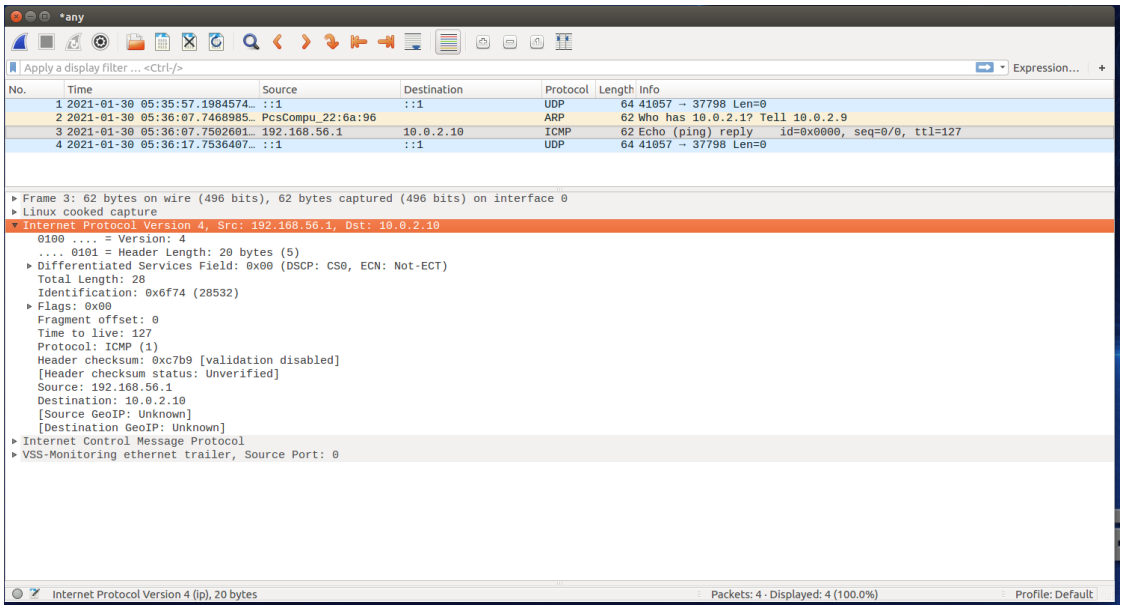


Victim:





Reply received by the victim from host with ip 192.168.56.1 which had received a spoof request from the attacker.

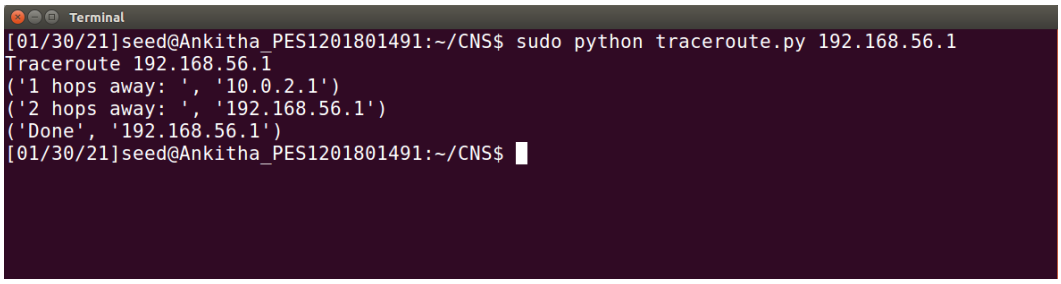


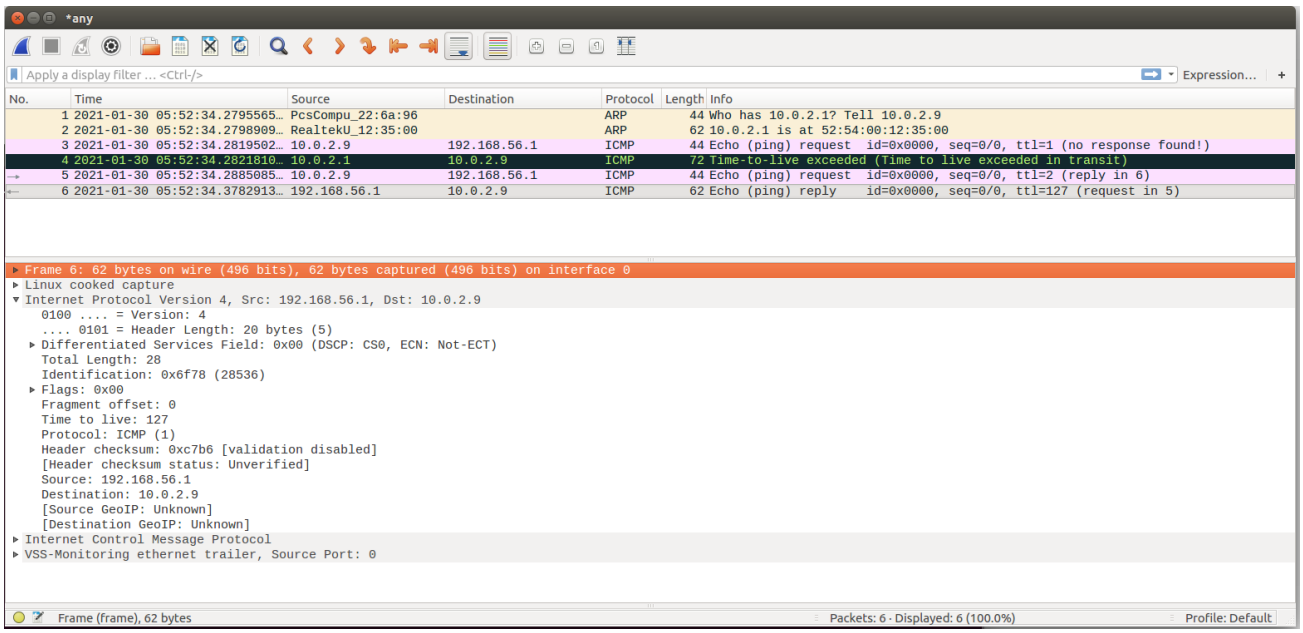
OBSERVATIONS:

- The spoof ICMP echo request packets sent to the host with ip 192.168.56.1 was accepted and an echo reply packet was sent to the spoofed IP address(i.e victim: 10.0.2.10)
- The reply from the spoofed packet was observed in the wireshark of the victim and attacker machine.

2.1.4 Task 3: Traceroute

Implemented a simple traceroute tool using Scapy to estimate the distance, in terms of number of routers, between your VM and a selected destination.





OBSERVATIONS:

- The packet will be dropped by the first router, which will send us an ICMP error message, telling that the TTL has exceeded.
- The wireshark capture shows the ICMP requests sent with increasing TTL (i.e the second packet is sent out with TTL value of 2)

2.1.5 Task 4: Sniffing and-then Spoofing

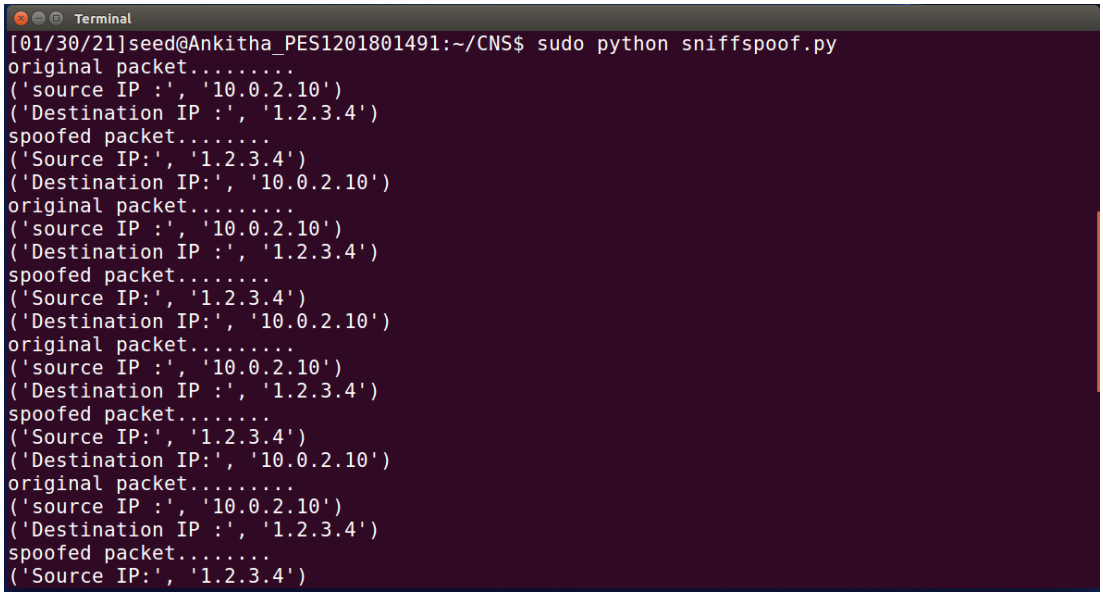
In this task, victim machine pings a non-existing IP address “1.2.3.4”. As the attacker machine is on the same network, it sniffs the request packet, creates a new echo reply packet with IP and ICMP header and sends it to the victim machine. Hence, the user will always receive an echo reply from a non-existing IP address indicating that the machine is alive.

VM 1 (Victim): 10.0.2.10

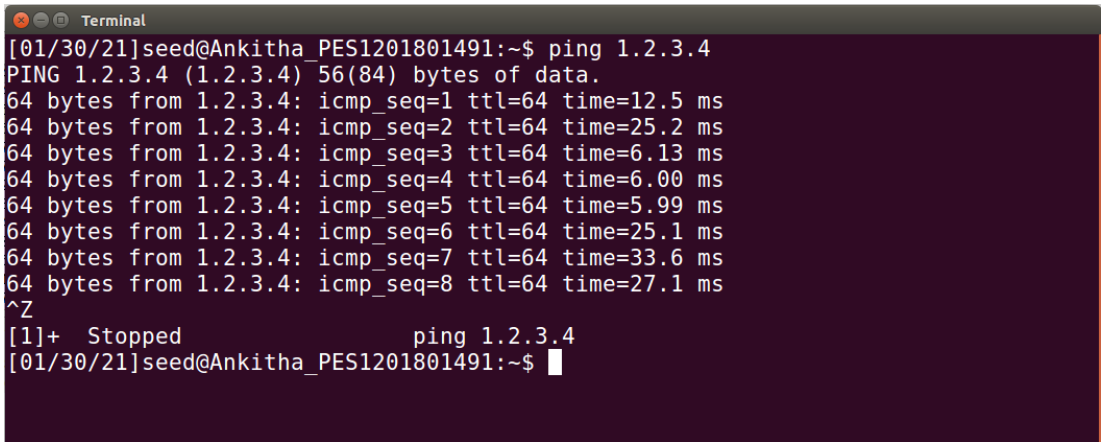
Ping X: 1.2.3.4

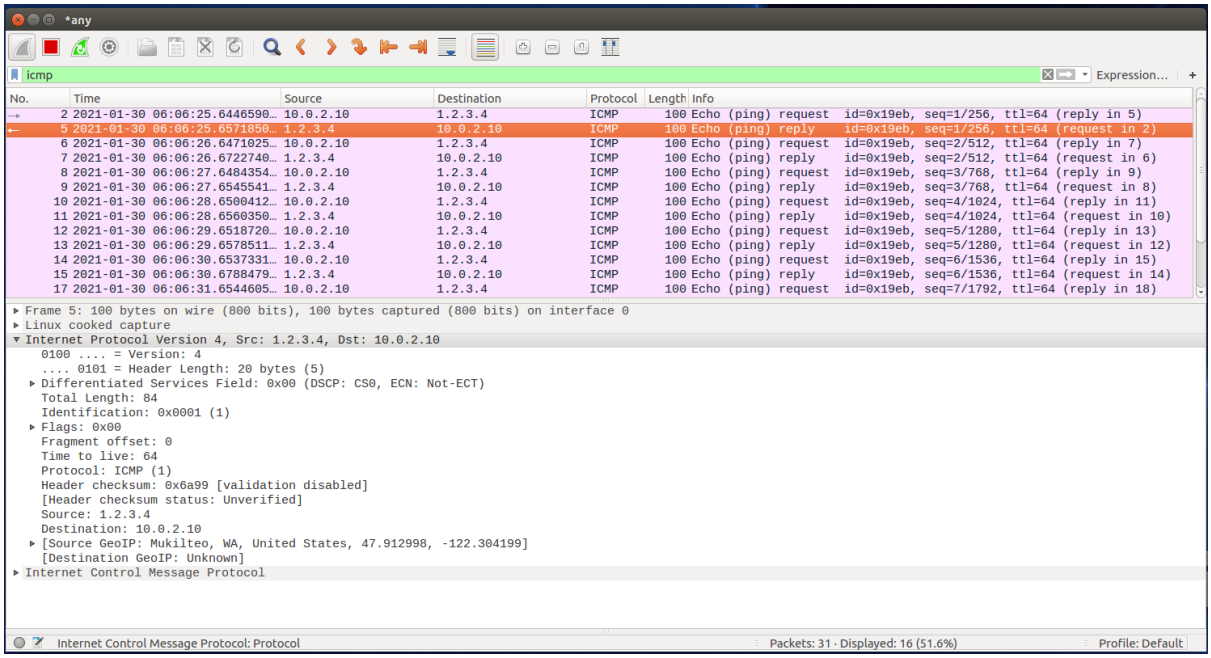
VM 2 (attacker with sniffer-spoofer running): 10.0.2.9

Attacker:



Victim:





OBSERVATIONS:

- We observe that 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.