**Lab2: Cryptography and Network Security**

**Task 1.1A**

The following code is used to sniff the packets from the Host A and print the information about the ICMP packets. The interface is found by using the ifconfig command and is updated in the code.

A screenshot of a computer

Description automatically generated

Running the program with the root privilege the captured packets output is the following.

A screenshot of a computer

Description automatically generated

Running the program without the root privilege.

A screenshot of a computer

Description automatically generated

When the program is run without the root privilege, it gives “PermissionError: Operation not permitted” because scapy requires root privilege to perform network operations and without privilege the operation is not permitted.

**Task 1.1B**

**Capture only the ICMP packet.**

A screenshot of a computer

Description automatically generated

**Output**

A screenshot of a computer

Description automatically generated

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

**A screenshot of a computer

Description automatically generated**

**Output**

A screenshot of a computer

Description automatically generated

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.

A screenshot of a computer

Description automatically generated

To send packets from the destination subnet mask in the host A the following code was executed to capture for destination port 21 and 23 i.e., FTP and Telnet respectively.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Task 1.2: Spoofing ICMP Packets**

Here we will spoof IP packets with an arbitrary source IP address and will spoof ICMP echo request packets and send.

Spoofing ICMP packets.

A screenshot of a computer

Description automatically generated

From host B when we ping to host A we get the following result.

A screenshot of a computer

Description automatically generated

**Task 1.3: Traceroute**

The objective of this task is to use scapy to estimate the distance, in terms of number of routers, between your VM and a selected destination. Here I am sending an ICMP packet to the destination 8.8.4.4 and have set the time to live to 1. Once it reaches the first router it would send the error message that the TTL has exceeded. So in each following round we increment the value of TTL and send another packet and get address of next router till we reach our destination. This is done by using an infinite while loop and incrementing the TTL in each iteration.

A screenshot of a computer program

Description automatically generated

Execution of Traceroute program for 8.8.4.4

A screenshot of a computer

Description automatically generated

**Task 1.4: Sniffing and-then Spoofing**

A screenshot of a computer program

Description automatically generated

1. Pinging 1.2.3.4

When we try to ping 1.2.3.4 which is non-existing host on the internet, we can see that the spoofed packets are sent in reply with ttl = 64.

A screenshot of a computer

Description automatically generated

1. Pinging 10.9.0.99

When we try to ping 10.9.0.99 which is a non-existing host on the LAN it says that destination is not reachable

A screenshot of a computer

Description automatically generated

To find the router for that destination we run “ip route get” for both 1.2.3.4 and 10.9.0.99

A screenshot of a computer

Description automatically generated

Here we can see that there is route to 1.2.3.4 from 10.9.0.1 but in case of 10.9.0.99 there is no route because the ARP protocol is not able to find the route when we send the ping.

In the case of 1.2.3.4 it is on the internet, so the ARP can find the router that routes the packet sent and is able to convert the IP to MAC address whereas in 10.9.0.99 ARP is not able to find any route to this destination.

1. Pinging 8.8.4.4

A screenshot of a computer

Description automatically generated

When we try to ping 8.8.4.4 which is an existing host on the internet, we can see that the spoofed packets are sent in reply with ttl less than 64