DHCP Starvation Attack

Name:Adiba Shaira
Roll:1605097

Steps of Attack:

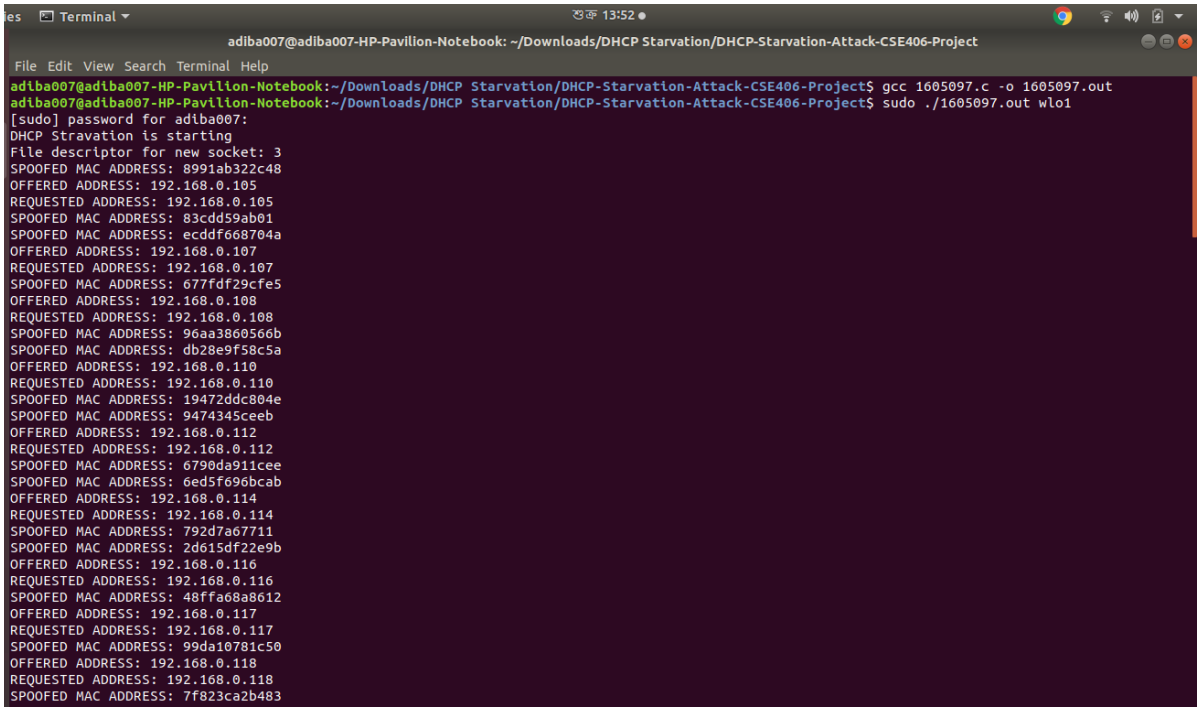
1.Command in Terminal:

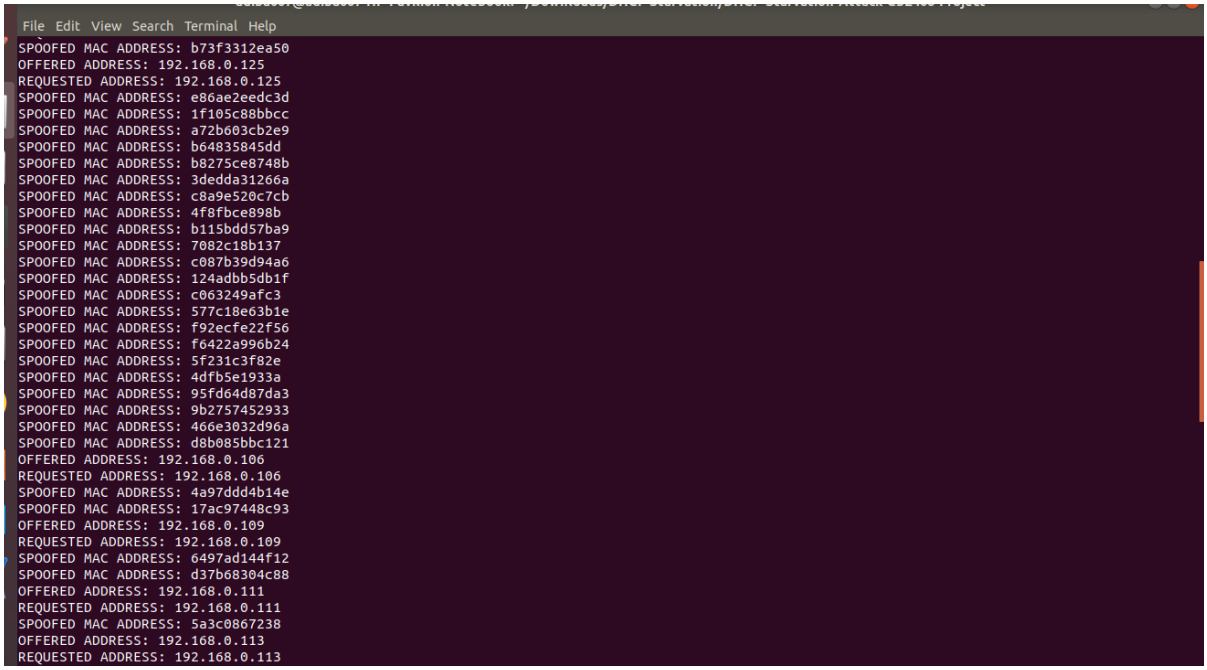
a.gcc <filename> -o <output>

b. Ifconfig-interface name for wireless connection

c. sudo <output> <interface_name>

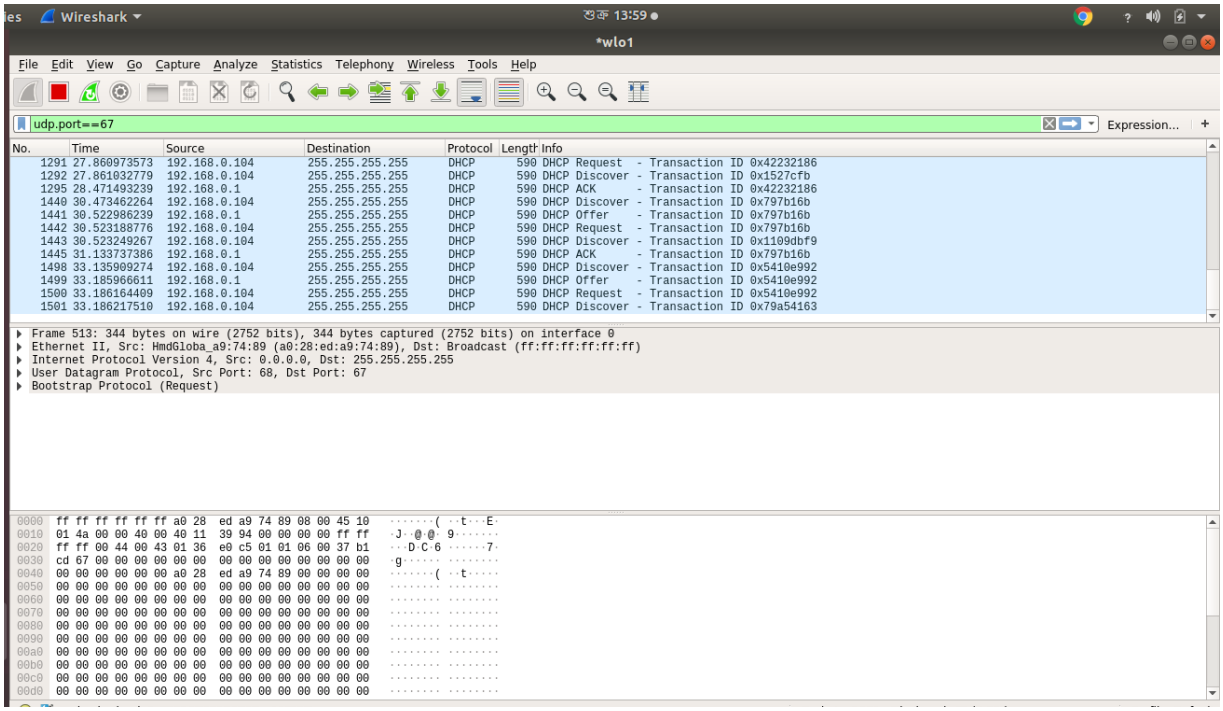
2.Outputs in Command line

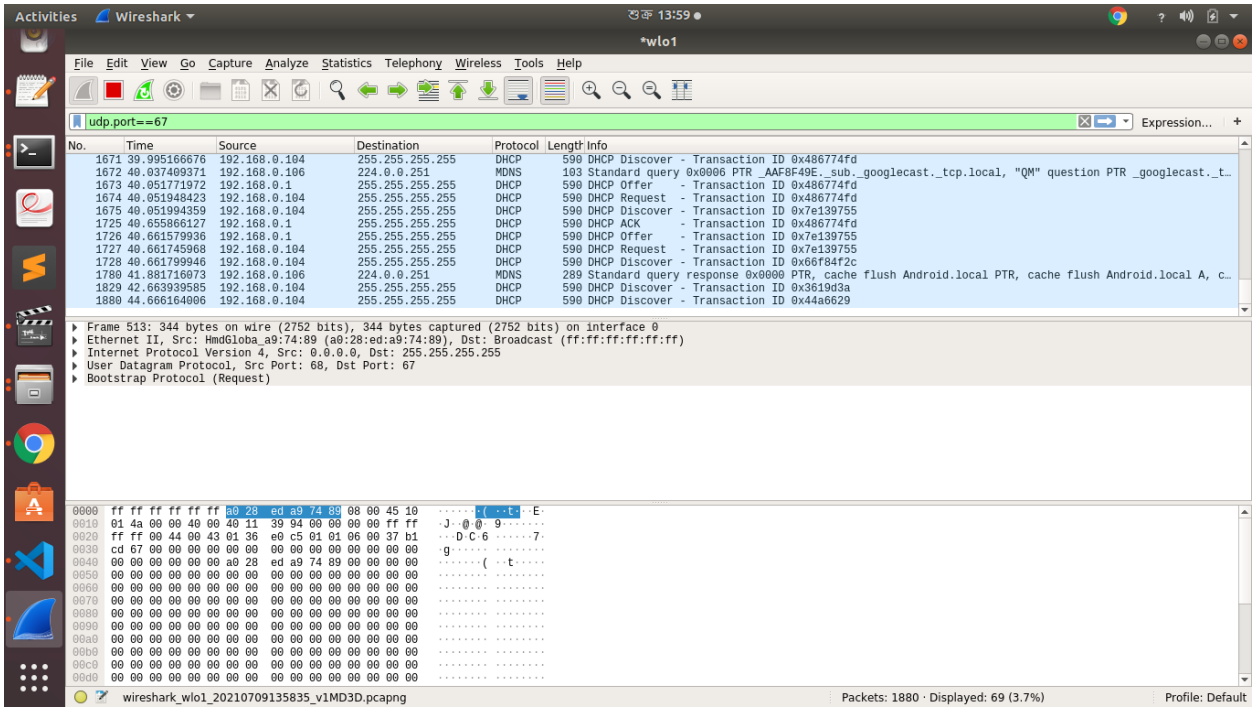




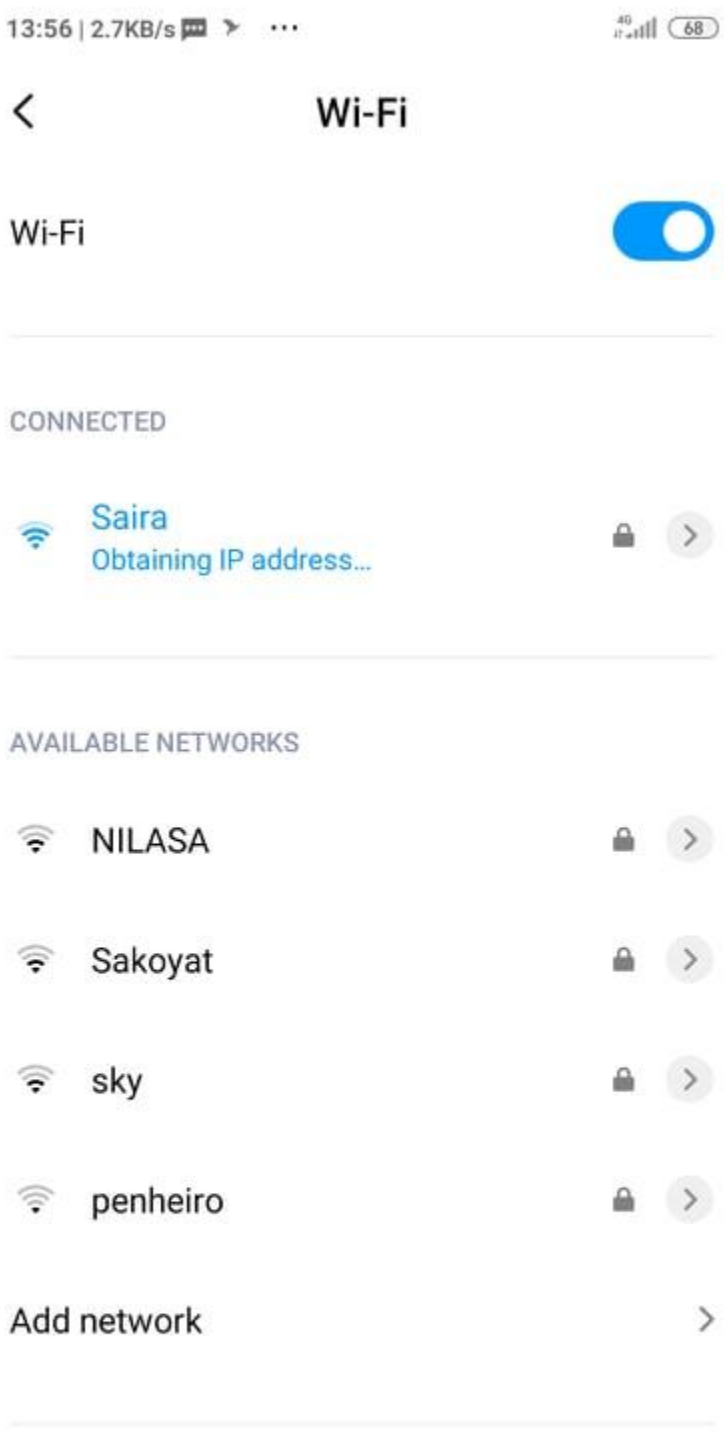
In this step,we can see that DISCOVER packets are being sent continuously and for those offered and requested packets are also generated.In this way all the available IP addresses will be unavailable.

Outputs in Wireshark:





New Client unavailable to join:



Though the correct password has been given to connect to router “Saira”,the mobile is not able to connect to the server.It’s happening because all the IP addresses have been occupied by the client(attacker).So,the attack is successful.

Explanation of the Attack:

1.Initializing a socket:

```
int create_DHCP_socket()
```

Using this function,I created a raw socket using linux’s socket() system call in.

```
sock = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);
```

AF_INET : IPV4 Internet Protocols
SOCK_DGRAM : Supports Datagrams
IPPROTO_UDP : DHCP uses UDP in the underlying transport layer.

2.Setting Client Address:

```
int setHardwareAddress()
```

Random Mac Addresses are generated in this function.Random addresses are generated for spoofing

the chadrr (Client Hardware Address) field in DHCP Discover packets.

3.Creating DHCP DISCOVER Packets:

```
int makeDHCPDiscoverPacket(int sock)
```

For flooding,raw DHCP DISCOVER packets have to be created.Following are the parameters used:
Operation Code : Set to 1 (As client i.e. attacker is sending discover packets)
Hops: Set to 0 so that packet reaches the router of the LAN in which the attacker remains
Hardware Type: Set to 1 (Ethernet)
Hardware Address Length: Length of Mac Address. Set to 6
Transaction Identifier:A 32-bit identification field generated by the client,to allow it to match up the request with replies received
from DHCP. Set to a random number of uint32_t. Using random() function.
Seconds: Set to 0
Flags : Set broadcast bit to 1 so that everyone gets the message
ciaddr : Set to 0 as it is set by the client when the client has confirmed it's ip.

yiaddr : Client's IP address; set by the server to inform the client
of the client's IP Address. So we need to set this to 0
siaddr : IP Address of the next server for the client to use in the configuration process
giaddr : Relay agent (gateway) IP address; filled in by the relay
agent with the address of the interface through which Dynamic Host Configuration Protocol (DHCP) message was received. So we need to set this to 0
chaddr: Client's hardware address (Layer 2 address). Set to the
spoofed MAC address.
Magic cookie :Set to 0x63825363

Sending DHCP Packets:

```
int sendPacket(void *buffer, int buffer_size, int sock, struct sockaddr_in
*dest)
```

After making DHCP DISCOVER packets,they are being sent by the sendPacket function.Continuous sending of DHCP DISCOVER packets is the key to this attack.

Receiving DHCP Packets:

```
int getDHCPOfferPacket(int sock)
```

Using this method,from server side we can get DHCP OFFER packet.After some time REQUEST packet is being sent from the client and finally the ACK packet from server side.This and above process is continuously done until all the available ip addresses are spoofed.

Evaluation of the Attack:

Though the victim's machine tried to join the server and the password was also correct,it couldn't.It happened because all the ip addresses were unknowingly given to the attacker by the server.The only way victim could join the server was to restart the router/server.

Measures to defend the attack:

1.Enabling mac address check.The DHCP server compares the **chaddr** field of a received DHCP request with the source MAC address in the frame header. If they are the same, the DHCP server verifies this

request as legal and processes it. If they are not the same, the server discards the DHCP request.
2.Enabling port security in case of wired connection
3.Limiting the number of DHCP DISCOVER packets through a single port.