DHCP Starvation Attack

Najrin Sultana
Student ID : 1605042

Steps of Attack :

Inside Code :

1. Creating a raw socket : A raw socket is created using socket () system call in Linux. Parameters passed are: AF_INET (for IPV4 protocols),SOCK_DGRAM (connectionless, unreliable messages of fixed length),IPPROTO_UDP (DHCP uses UDP in underlying transport layer).

2. Random MAC Address generation : Random addresses are generated for spoofing the chadrr (Client Hardware Address) field in DHCP Discover packets.

3. Making DHCP Discover packets : Raw DHCP Discover packets are used for this attack. Parameters used in this packet are:
Operation Code : Set to 1 (boot request flag ,backward compatible with BOOTP servers)
Hardware Type : Set to 1 (Ethernet)
Hardware Address Length : Length of Mac Address. Set to 6
Hops : Set to 0 so that packet reaches the router of the LAN the attacker is in
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 : Elapsed Time. Set to 0
Flags : Broadcast bit is set to 1 as everyone gets the broadcast message
chaddr : Client's hardware address (Layer 2 address). Set to the spoofed MAC address.
Magic cookie : Set to 0x63825363 in the first four bytes of options field.
DHCP message type is embedded in options field. The 7th byte is set to 1 as DHCP Discover packet in being sent.

4. Sending out DHCP Discover packets : DHCP Discover packets are broadcasted using sendto() system call of Linux using the raw socket opened in the previous step.

5. Receiving DHCP Offer packets : DHCP Offer packets which were sent by the server were received using recvfrom() system call of Linux using the same socket. An ip address is offered with this Offer packet.

6. Making DHCP Request packets : A DHCP Request packet is sent after receiving the offer packet to reserve the IP offered in the Offer packet. Parameters used in this packet are:
Operation Code : Set to 1 (boot request flag, backward compatible with BOOTP servers).
Hardware Type : Set to 1 (Ethernet).
Hardware Address Length : Length of Mac Address. Set to 6.
Hops : Set to 0 so that packet reaches the router of the LAN the attacker is in.
Transaction Identifier : The same transaction ID of the Discover packet.
Seconds : Elapsed Time. Set to 0.

Flags : Broadcast bit is set to 1, tells server it should broadcast its response.
siaddr : set to the server ip address got from the offer packet.
ciaddr : set to the offered ip address to request to reserve it.
chaddr : Client's hardware address (Layer 2 address). Set to the spoofed MAC address.
Magic cookie : Set to 0x63825363 in the first four bytes of options field.
DHCP message type is embedded in options field. The 7th byte is set to 3 as DHCP Discover packet in being sent.

7. Sending out DHCP Request packets : The DHCP Request packet is sent using sendto() system call of Linux using the socket.

Steps 2 to 7 are repeatedly performed to completely exhaust the IP address pool of the server.

N.B : The interface name is set to *wlp3s0* in the code. This should be changed accordingly. The interface name can be found by typing the ifconfig command in the terminal.

Compiling the Code :

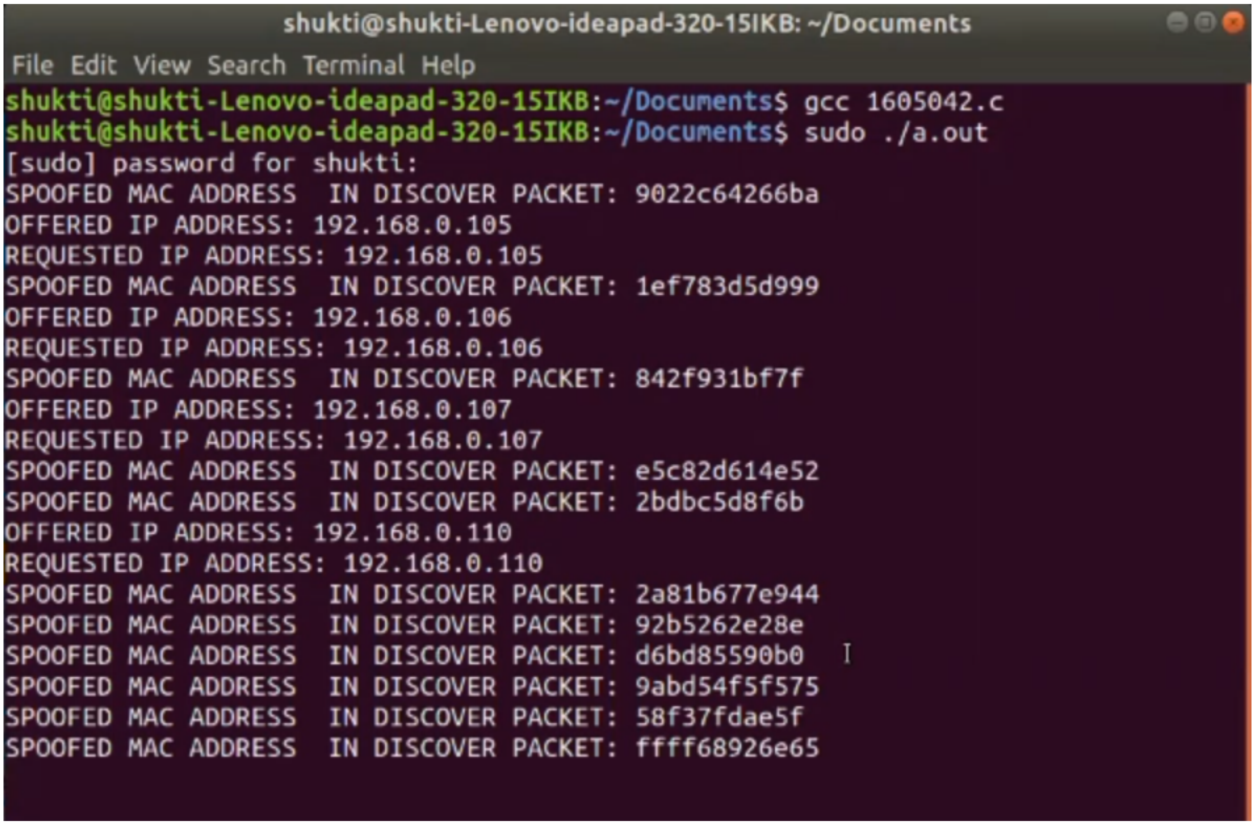
```
gcc <file-name>
```

Running the Code :

```
sudo ./a.out
```

Observed Output in Terminal :

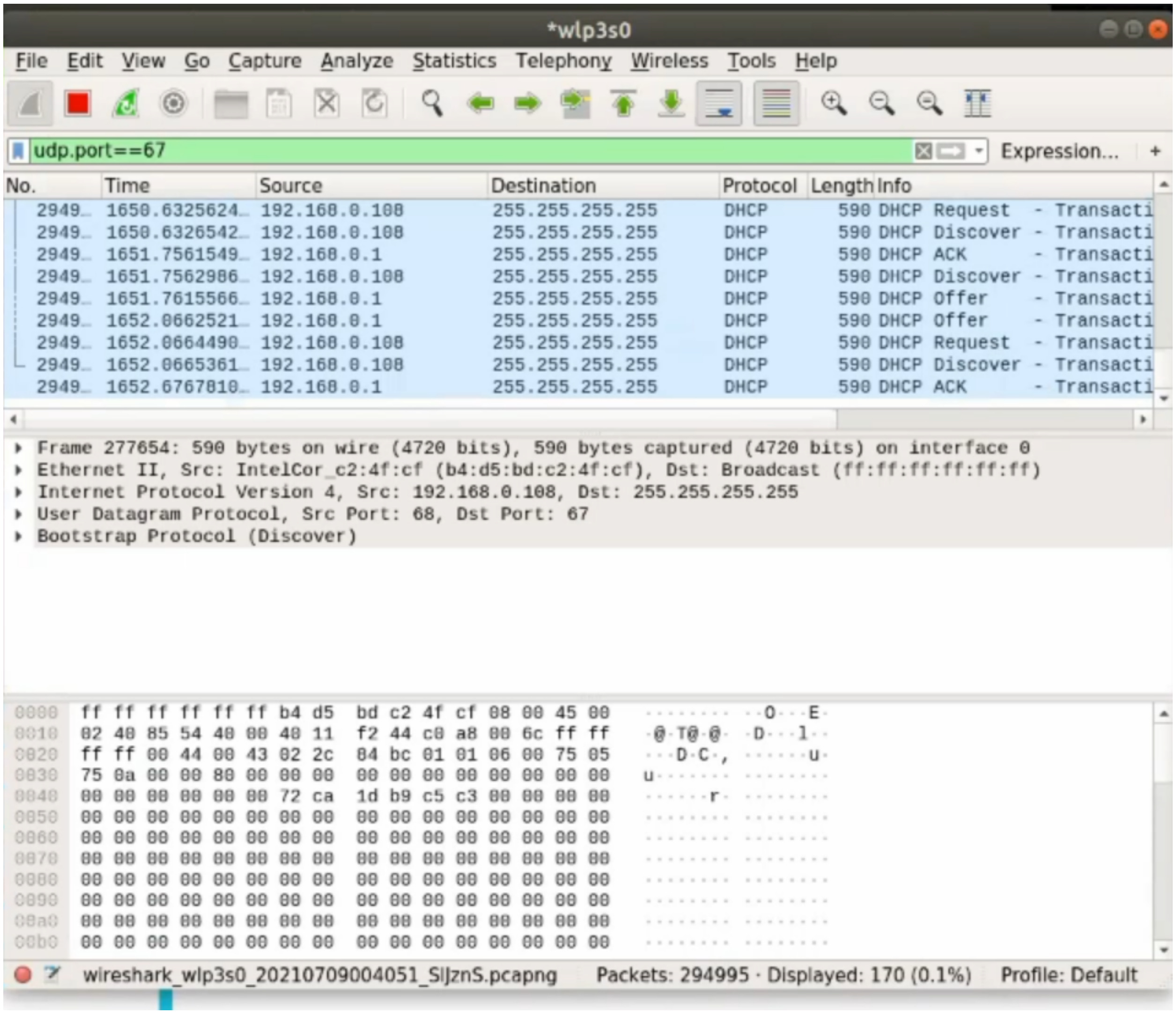
In the beginning, the Offer packets are received. When the IP pool of the server had gotten exhausted i.e it had no more IP to offer, no more Offer packet was received.


shukti@shukti-Lenovo-ideapad-320-15IKB: ~/Documents
File Edit View Search Terminal Help

```
shukti@shukti-Lenovo-ideapad-320-15IKB:~/Documents$ gcc 1605042.c
shukti@shukti-Lenovo-ideapad-320-15IKB:~/Documents$ sudo ./a.out
[sudo] password for shukti:
SPOOFED MAC ADDRESS  IN DISCOVER PACKET: 9022c64266ba
OFFERED IP ADDRESS: 192.168.0.105
REQUESTED IP ADDRESS: 192.168.0.105
SPOOFED MAC ADDRESS  IN DISCOVER PACKET: 1ef783d5d999
OFFERED IP ADDRESS: 192.168.0.106
REQUESTED IP ADDRESS: 192.168.0.106
SPOOFED MAC ADDRESS  IN DISCOVER PACKET: 842f931bf7f
OFFERED IP ADDRESS: 192.168.0.107
REQUESTED IP ADDRESS: 192.168.0.107
SPOOFED MAC ADDRESS  IN DISCOVER PACKET: e5c82d614e52
SPOOFED MAC ADDRESS  IN DISCOVER PACKET: 2bdbc5d8f6b
OFFERED IP ADDRESS: 192.168.0.110
REQUESTED IP ADDRESS: 192.168.0.110
SPOOFED MAC ADDRESS  IN DISCOVER PACKET: 2a81b677e944
SPOOFED MAC ADDRESS  IN DISCOVER PACKET: 92b5262e28e
SPOOFED MAC ADDRESS  IN DISCOVER PACKET: d6bd85590b0
SPOOFED MAC ADDRESS  IN DISCOVER PACKET: 9abd54f5f575
SPOOFED MAC ADDRESS  IN DISCOVER PACKET: 58f37fdae5f
SPOOFED MAC ADDRESS  IN DISCOVER PACKET: ffff68926e65
```

Observed Output in Wireshark :

After sending the Request packet, it was observed that the Acknowledgement packet was also received. It confirms the reservation of the corresponding IP address.



```
udp.port==67
```

```
No.       Time            Source          Destination      Protocol  Length Info
  2949…  1650.6325624…  192.168.0.108   255.255.255.255  DHCP          590 DHCP Request  - Transacti
  2949…  1650.6326542…  192.168.0.108   255.255.255.255  DHCP          590 DHCP Discover - Transacti
  2949…  1651.7561549…  192.168.0.1     255.255.255.255  DHCP          590 DHCP ACK      - Transacti
  2949…  1651.7562986…  192.168.0.108   255.255.255.255  DHCP          590 DHCP Discover - Transacti
  2949…  1651.7615566…  192.168.0.1     255.255.255.255  DHCP          590 DHCP Offer    - Transacti
  2949…  1652.0662521…  192.168.0.1     255.255.255.255  DHCP          590 DHCP Offer    - Transacti
  2949…  1652.0664490…  192.168.0.108   255.255.255.255  DHCP          590 DHCP Request  - Transacti
  2949…  1652.0665361…  192.168.0.108   255.255.255.255  DHCP          590 DHCP Discover - Transacti
  2949…  1652.6767810…  192.168.0.1     255.255.255.255  DHCP          590 DHCP ACK      - Transacti
```

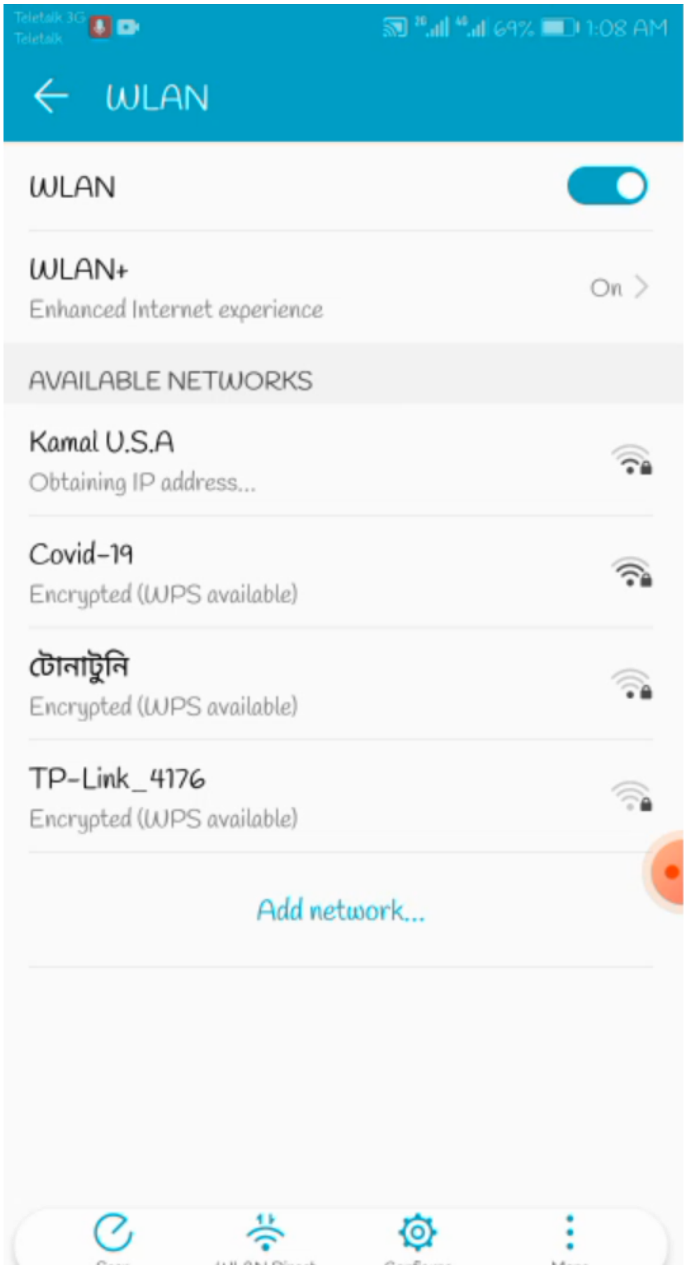
```
▸ Frame 277654: 590 bytes on wire (4720 bits), 590 bytes captured (4720 bits) on interface 0
▸ Ethernet II, Src: IntelCor_c2:4f:cf (b4:d5:bd:c2:4f:cf), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▸ Internet Protocol Version 4, Src: 192.168.0.108, Dst: 255.255.255.255
▸ User Datagram Protocol, Src Port: 68, Dst Port: 67
▸ Bootstrap Protocol (Discover)
```

```
0000  ff ff ff ff ff ff b4 d5  bd c2 4f cf 08 00 45 00   ········ ··O···E·
0010  02 40 85 54 40 00 40 11  f2 44 c0 a8 00 6c ff ff   ·@·T@·@· ·D···l··
0020  ff ff 00 44 00 43 02 2c  84 bc 01 01 06 00 75 05   ···D·C·, ······u·
0030  75 0a 00 00 80 00 00 00  00 00 00 00 00 00 00 00   u······· ········
0040  00 00 00 00 00 00 72 ca  1d b9 c5 c3 00 00 00 00   ······r· ········
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
```

```
wireshark_wlp3s0_20210709004051_SIJznS.pcapng    Packets: 294995 · Displayed: 170 (0.1%)    Profile: Default
```

Observed Output in Victim's Screen :

When the IP pool of the server was exhausted, it was tried to connect to the same DHCP server (i.e to connect to the same router) from a Smartphone.



WLAN

WLAN

WLAN+
Enhanced Internet experience
On

AVAILABLE NETWORKS

Kamal U.S.A
Obtaining IP address...

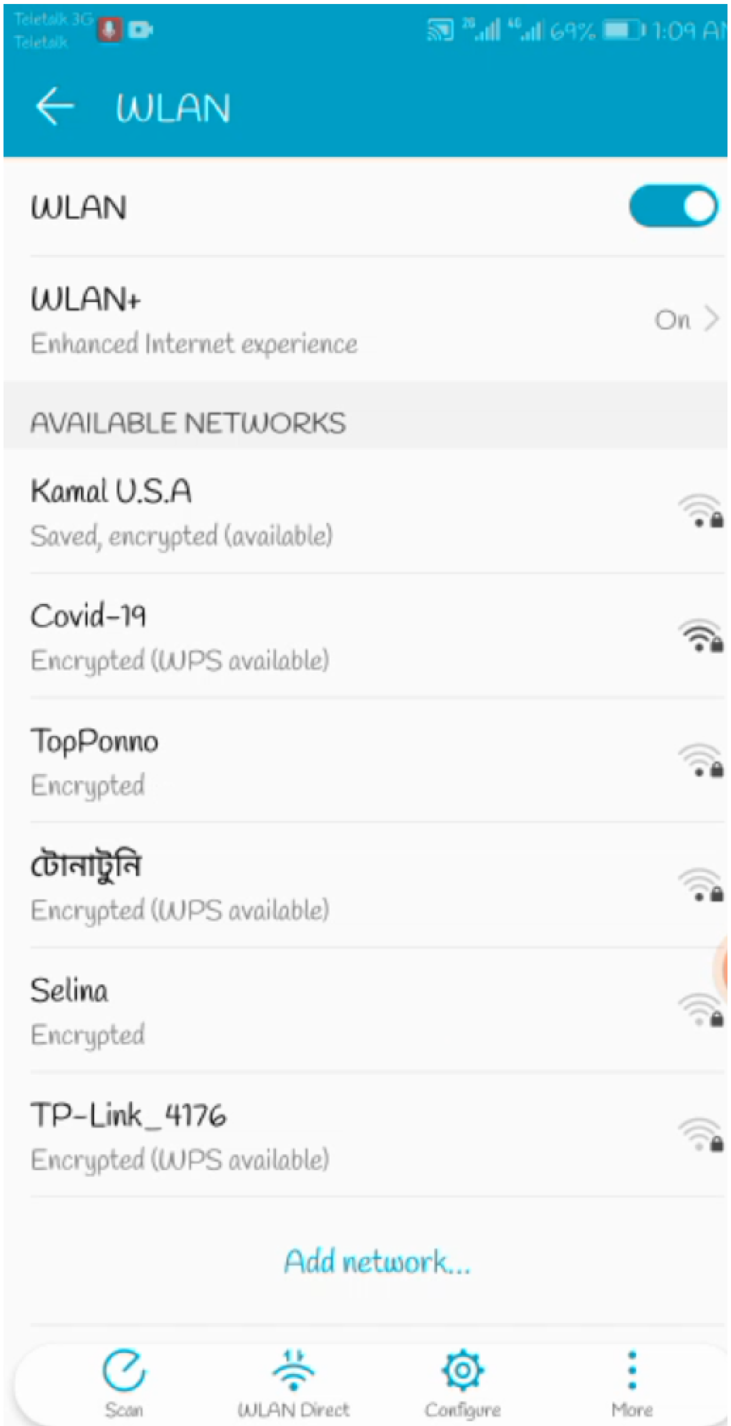
Covid-19
Encrypted (WPS available)

টোনাটুনি
Encrypted (WPS available)

TP-Link_4176
Encrypted (WPS available)

Add network...

As the server had no more IP to allocate for it, it failed to obtain any IP address and could not connect.



WLAN

WLAN+
Enhanced Internet experience
On

AVAILABLE NETWORKS

Kamal U.S.A
Saved, encrypted (available)

Covid-19
Encrypted (WPS available)

TopPonno
Encrypted

টোনাটুনি
Encrypted (WPS available)

Selina
Encrypted

TP-Link_4176
Encrypted (WPS available)

Add network...

Video Demonstration :

A video demonstration of a successful attack can be found in the link given below :

https://drive.google.com/file/d/1Finj7YpJ5K-3HrjwDWIDnEXM8WqBlJI7/view?usp=sharing

Assessment of the Attack :

This attack was successful because as all the IP Addresses were allocated to the attacker unknowingly, no new user could be assigned an IP address. Although the victim machine tried to join the network i.e tried to obtain an IP address, it failed to do so because the IP pool of the server was exhausted and so the server had no more IP left to offer to the new user. This was the goal of the DHCP Starvation Attack and so it seems the attack was successful.

Possible Countermeasure :

During the attack as DHCP requests are encapsulated with the same source MAC address, to prevent DHCP starvation attack, MAC address check on the DHCP server can be enabled. The DHCP server will then compare 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 will verify this request as legal and process it. If they are not the same, the server will discard the DHCP request.