REPORT OF SNIFFING

BY SACHCHITANAND YADAV

SNIFFING

MODULE - 6

Learning Objectives -

- Summarize Sniffing Concepts
- Demonstrate Different Sniffing Techniques
- Use Sniffing Tools
- Explain Sniffing Countermeasures

TABLE OF CONTENTS

7. Detecting Network Sniffing and MAC Flooding Using hping3

7.1 Detecting Network Sniffing
7.2 Introduction to hping3
7.3 Working of hping3
7.4 Command Used
7.5 Conclusion

8. ARP Poisoning Attacks

8.1 ARP Poisoning Using Ettercap
    8.1.1 Working of ARP Poisoning
    8.1.2 Command Explanation
    8.1.3 Conclusion
8.2 ARP Poisoning Using Bettercap
    8.2.1 Attack Procedure
    8.2.2 Conclusion

9. Sniffing Countermeasures

9.1 Use of Encryption
9.2 Replace Hubs with Switches
9.3 Implement Port Security
9.4 Detect Promiscuous Mode
9.5 Monitor ARP Traffic
9.6 Intrusion Detection Systems
9.7 Network Traffic Analysis
9.8 Sniffing Detection Techniques
9.9 Overall Conclusion

10. Module Summary

10.1 Overview of Sniffing Concepts
10.2 Summary of Sniffing Techniques
10.3 Tools Used for Network Sniffing
10.4 Sniffing Countermeasures Discussed
10.5 Importance of Sniffing Detection

Summarize Sniffing Concepts: -

Sniffing Concepts

This section explains the concept of network sniffing and the associated security threats. It describes how a packet sniffer operates, including **active and passive sniffing techniques**, and how attackers exploit sniffers to compromise networks. The section also covers **protocols vulnerable to sniffing**, sniffing activities at the **Data Link Layer of the OSI model**, the role of **hardware protocol analyzers**, the use of **Switched Port Analyzer (SPAN) ports**, as well as techniques such as **wiretapping and lawful interception**.

Network Sniffing

Network sniffing refers to the practice of monitoring and capturing data packets as they travel across a network. It allows an attacker or administrator to observe network communication from a single point and analyze the flow of information between devices.

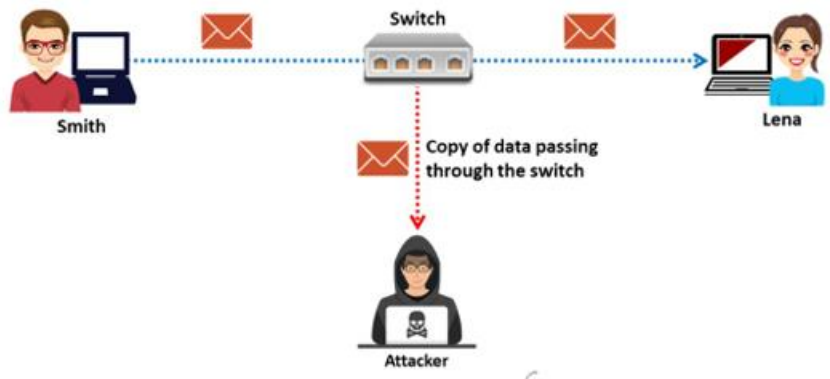
Packet Sniffing

Packet sniffing is the process of capturing all data packets passing through a given network using a software application or a hardware device. By observing these packets, an attacker can gain access to sensitive information transmitted over the network.

Packet sniffing enables the collection of critical data such as:

- Telnet usernames and passwords
- Email traffic
- Login credentials
- DNS traffic
- FTP passwords
- Chat sessions
- Account and configuration information

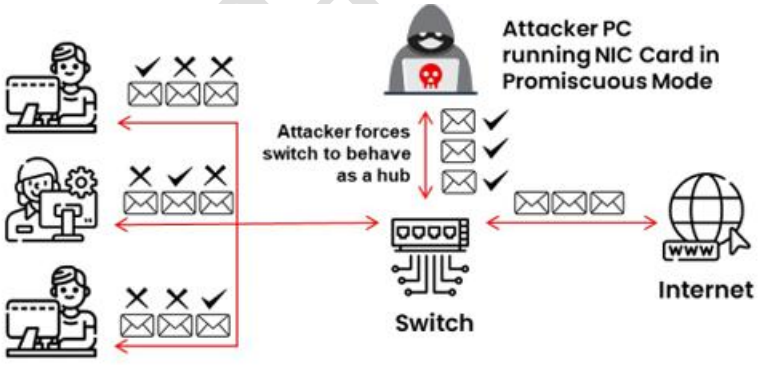
If the data is transmitted without encryption, it becomes easily readable once captured.



How a Sniffer Works

A packet sniffer operates by placing the system's **Network Interface Card (NIC)** into **promiscuous mode**. In this mode, the NIC accepts all packets present on the network segment rather than only those addressed to it.

In switched networks, attackers may use techniques such as MAC flooding or ARP poisoning to force the switch to behave like a hub. When this happens, network traffic is broadcast to all ports, allowing the attacker's system—running in promiscuous mode—to intercept and analyze packets intended for other devices.



Core Definition

Packet sniffing is the process of monitoring and capturing all data packets passing through a network using either software or hardware-based tools. A packet sniffer operates by placing the system's Network Interface Card (NIC) into **promiscuous mode**, which allows it to receive and

analyze all packets transmitted across the network segment, not just those specifically addressed to the system.

Types of Sniffing

Packet sniffing generally occurs in two different network environments:

Passive Sniffing

Passive sniffing is performed in **hub-based or shared networks**, where network traffic is broadcast to all connected devices. The attacker simply listens to the packets flowing through the network without injecting or modifying any traffic, making detection extremely difficult.

Active Sniffing

Active sniffing is used in **switched networks**, where switches normally forward traffic only to the intended destination based on MAC addresses. In this case, attackers must actively interfere with network communication—often by injecting malicious packets—to trick the switch into forwarding traffic to them.

Key Sniffing Techniques

Attackers use several techniques to bypass switch-based security mechanisms:

- **MAC Flooding:** The attacker overwhelms a switch's CAM (Content Addressable Memory) table with fake MAC addresses. Once the table is full, the switch enters a fail-open state and starts broadcasting traffic like a hub.
- **ARP Poisoning / ARP Spoofing:** Forged ARP messages are sent to associate the attacker's MAC address with a legitimate IP address, such as the default gateway, enabling interception of traffic.
- **DHCP Attacks:**
    - *DHCP Starvation:* Exhausts all available IP addresses in the DHCP pool.
    - *Rogue DHCP Server:* A fake DHCP server provides incorrect network configuration details to victims.

- **VLAN Hopping:** Allows attackers to access traffic from other VLANs by bypassing logical network segmentation using techniques such as switch spoofing or double tagging.

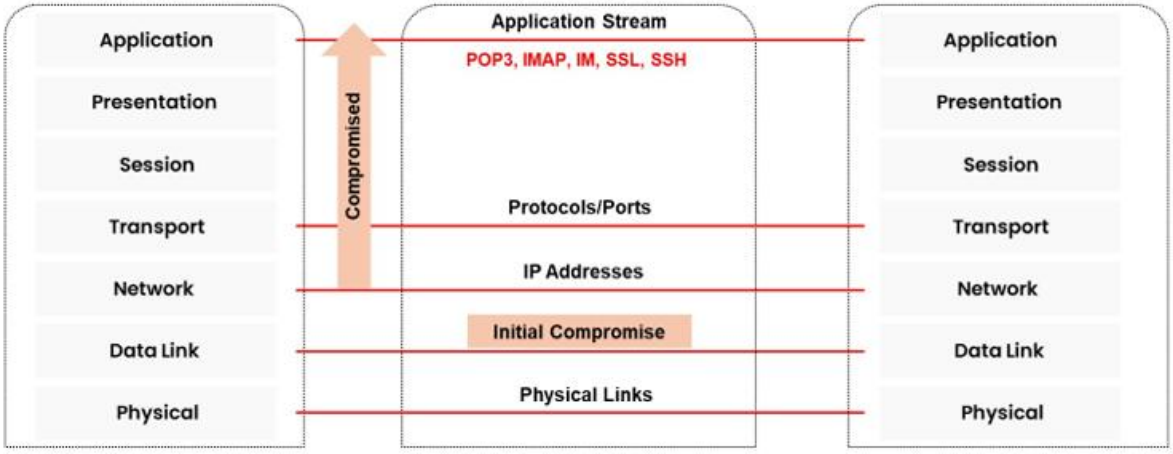
Vulnerable Protocols

Protocols that transmit data without encryption are highly vulnerable to sniffing attacks because sensitive information is sent in cleartext:

- **Telnet / Rlogin:** Transmit usernames and passwords in plaintext.
- **HTTP:** Sends user data without encryption in its default configuration.
- **SMTP, POP, IMAP:** Common email protocols that may expose credentials and message content.
- **FTP / TFTP:** File transfer protocols that do not encrypt authentication details or data.

OSI Model Context

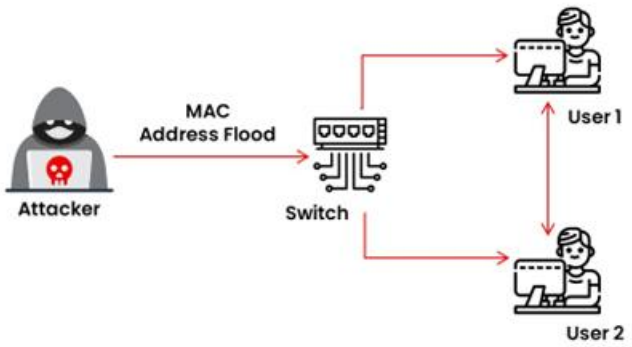
Packet sniffers primarily operate at the **Data Link Layer (Layer 2)** of the OSI model. Since each OSI layer functions independently, sniffing at Layer 2 often remains invisible to higher layers such as the Transport and Application layers, allowing attacks to go undetected.



Perform MAC Flooding using macof

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.



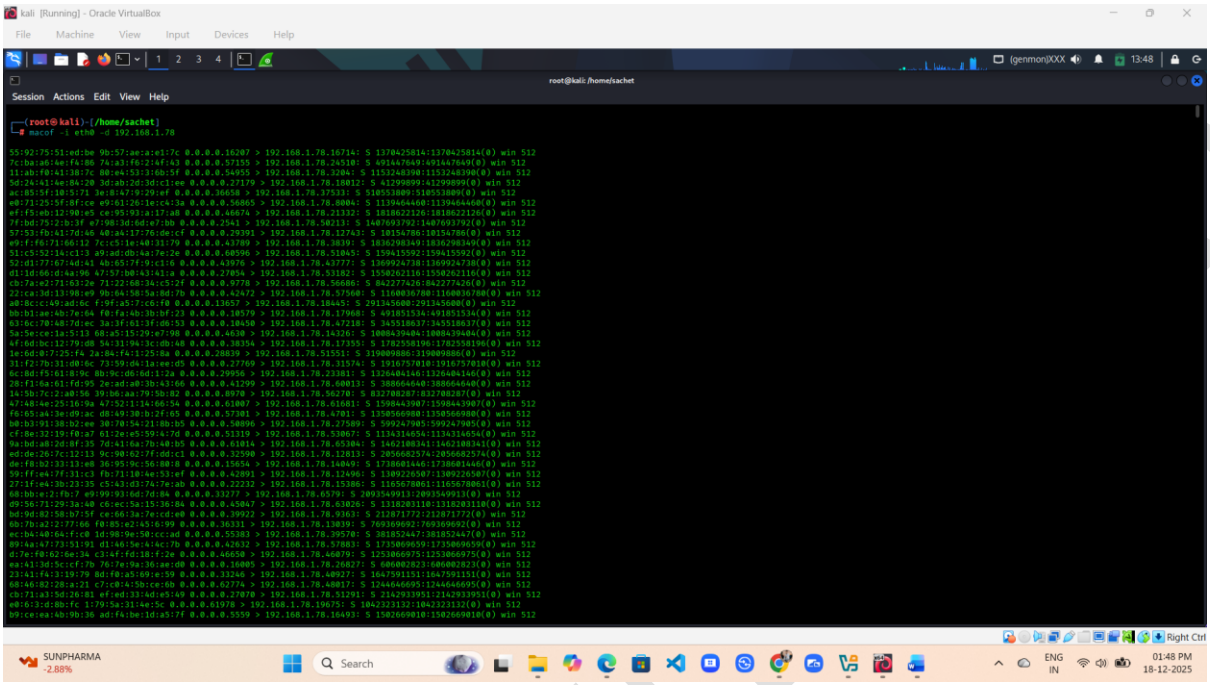
How to use it - :

- Open kali linux / parrot os terminal
- Type sudo apt install macof
- Then get detailed information about macof , simply type man macof
- Perform attack
- Type - : macof -I eth0 -d <Target IP>

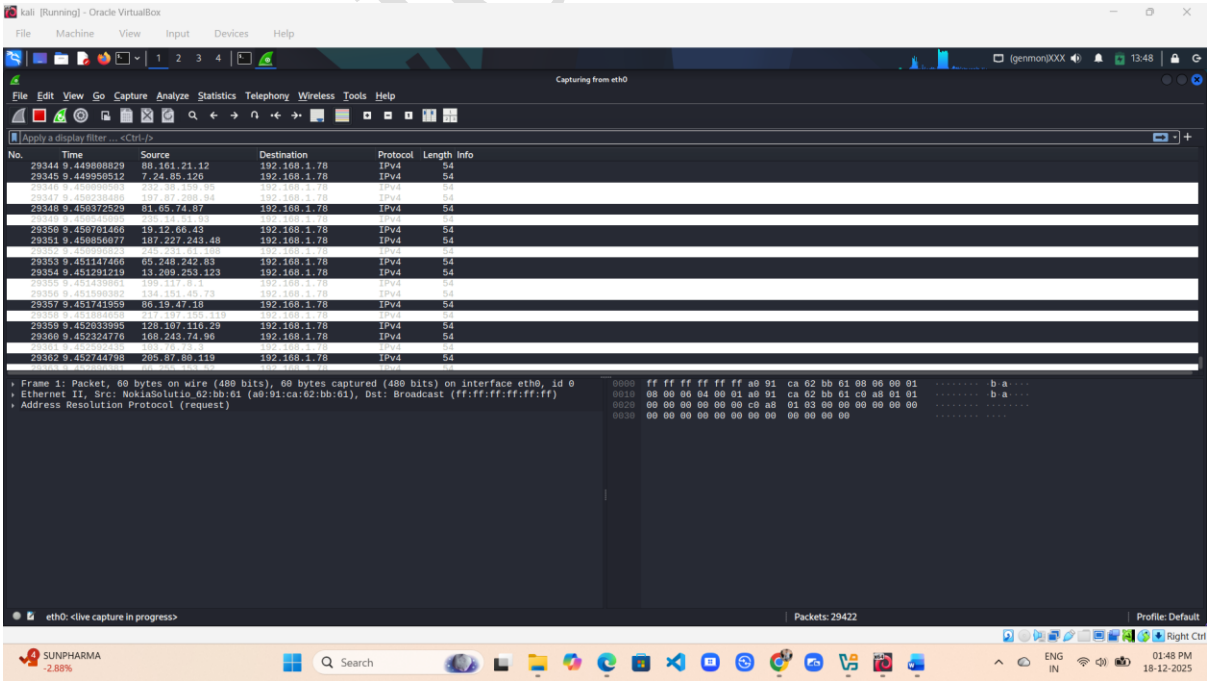
  -I → network interface

  -d → destination ip

- **Mac Flooding attack start**



- Now monitoring the attack using Wireshark
- Open Wireshark
- **Packet sending start**



DHCP Starvation

Definition

DHCP Starvation is a **Denial-of-Service (DoS) attack** in which an attacker deliberately exhausts the pool of IP addresses available on a DHCP server. This is achieved by flooding the server with a large number of **fake DHCP requests**, preventing legitimate devices from obtaining valid IP addresses and accessing the network.

DHCP Starvation Attack

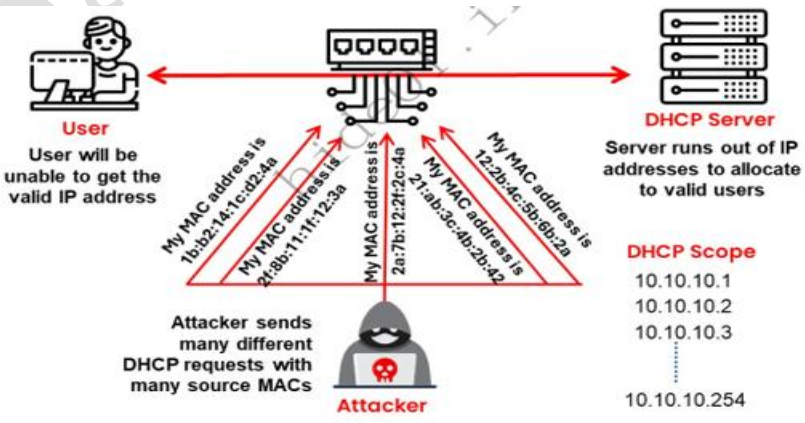
In a DHCP starvation attack, the attacker continuously sends DHCP discovery and request messages using spoofed MAC addresses. The DHCP server responds to each request by assigning an IP address from its available pool. As the number of requests increases rapidly, the pool of available IP addresses is exhausted.

Once the DHCP server runs out of IP addresses, it is unable to assign or renew addresses for legitimate users. As a result, genuine clients fail to join the network or lose existing connectivity, effectively causing a denial-of-service condition.

Impact of DHCP Starvation

- Legitimate users are unable to obtain or renew IP addresses
- Network access is denied to valid hosts
- Overall network availability is disrupted
- Can be combined with rogue DHCP attacks for further exploitation

This attack mainly targets networks that lack proper DHCP security mechanisms.



DHCP Starvation Attack Using Yersinia

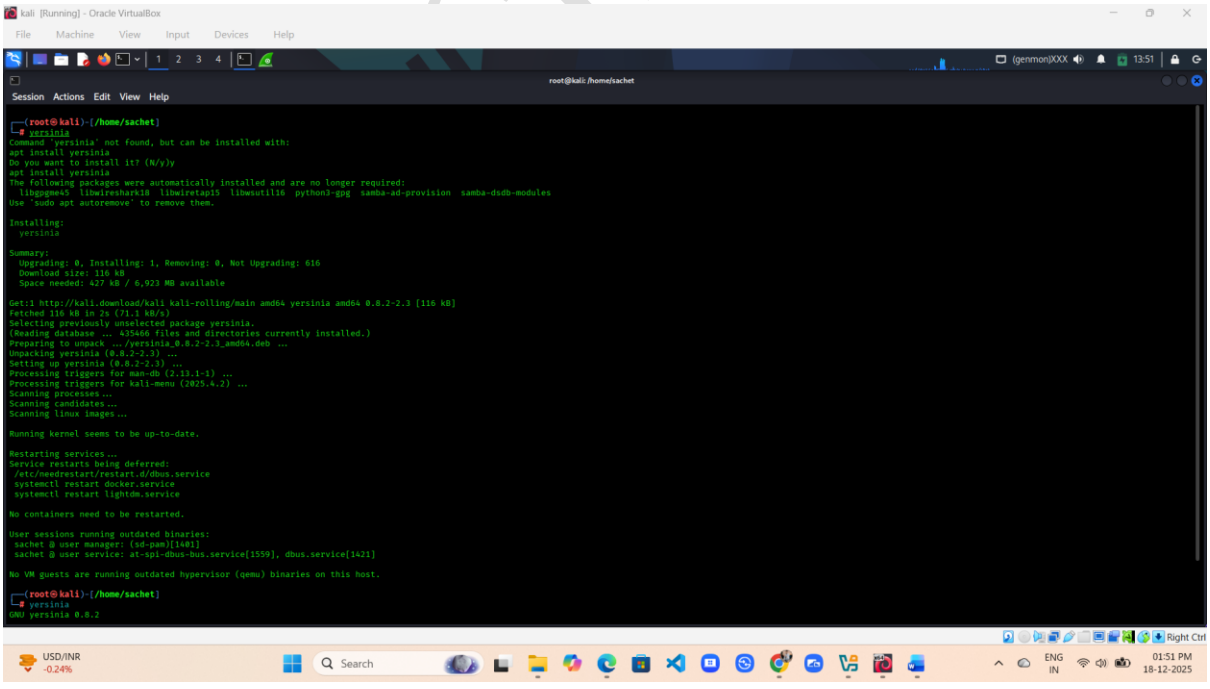
Yersinia – Network Attack Tool

Yersinia is a network penetration testing tool designed to exploit vulnerabilities in **Layer 2 (Data Link Layer)** network protocols such as DHCP, ARP, STP, and CDP. It is commonly used in controlled environments to demonstrate protocol weaknesses and to study network security flaws.

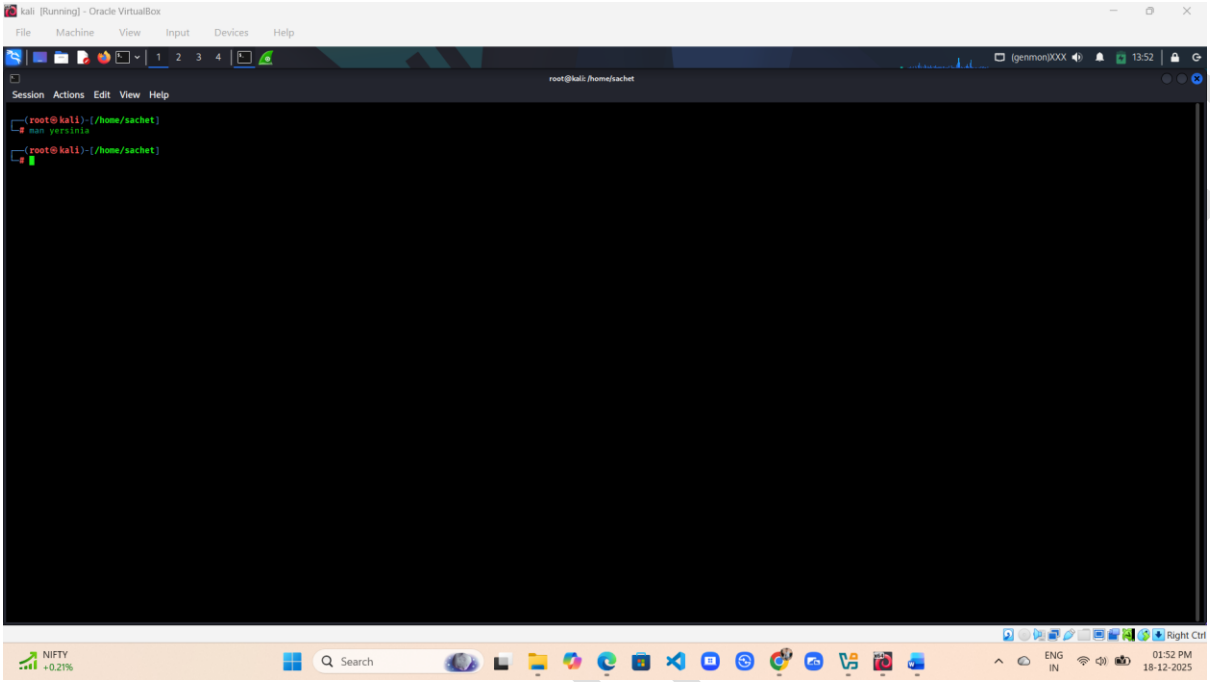
Yersinia can simulate multiple DHCP client requests with spoofed MAC addresses, making it effective for demonstrating how DHCP starvation attacks occur in unsecured networks.

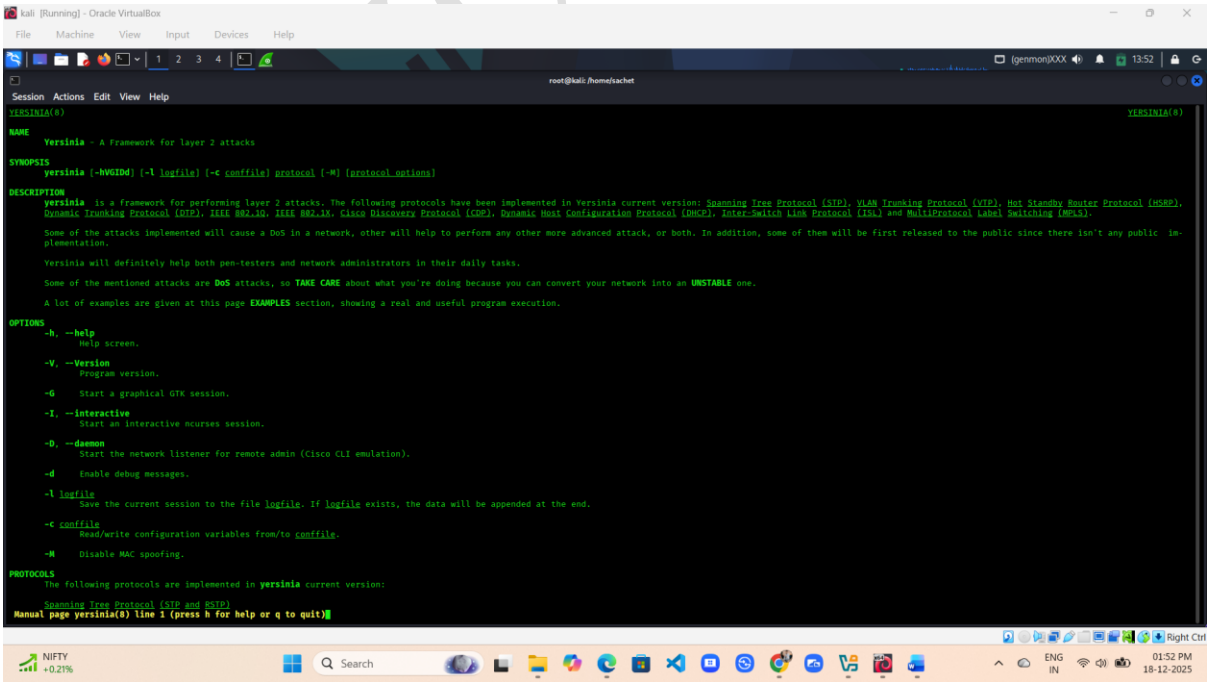
How to use it -:

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

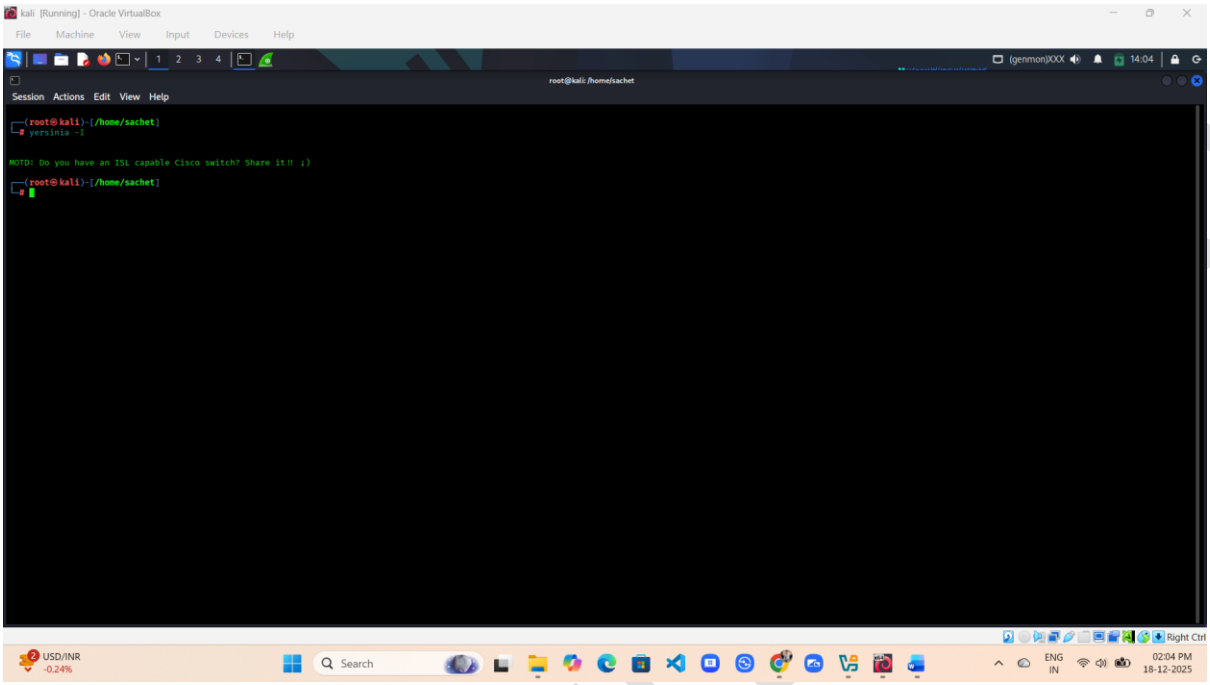


- get detailed information about macof , simply type man yersinia

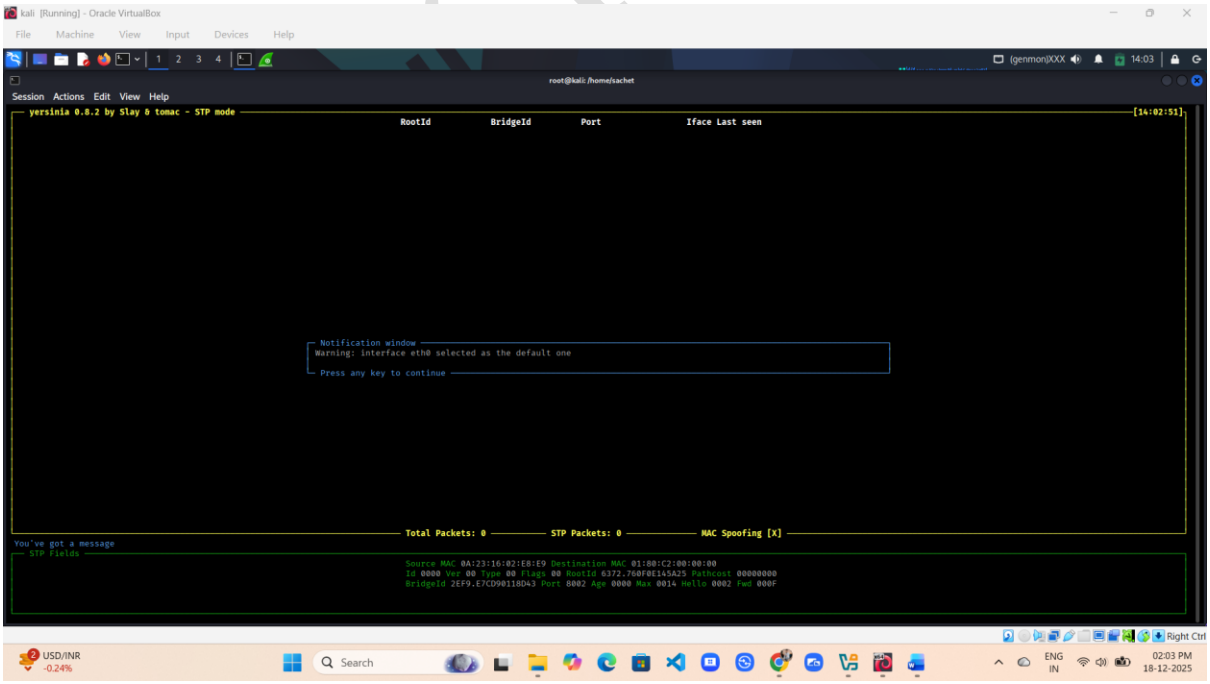


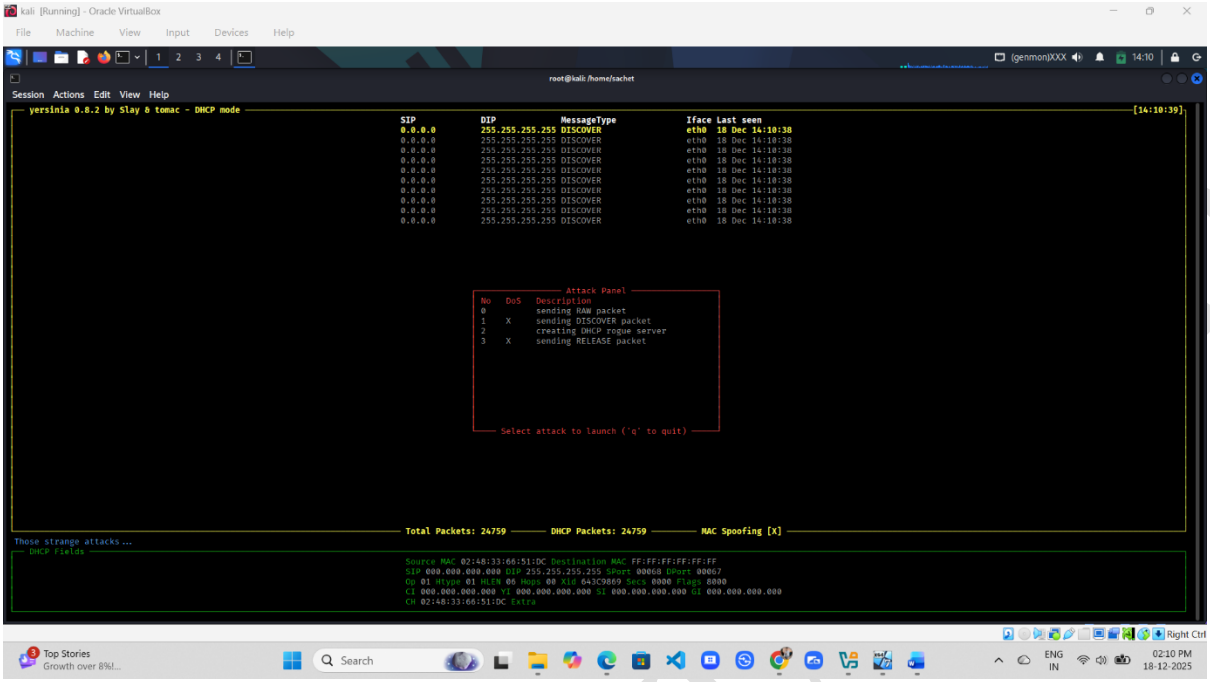


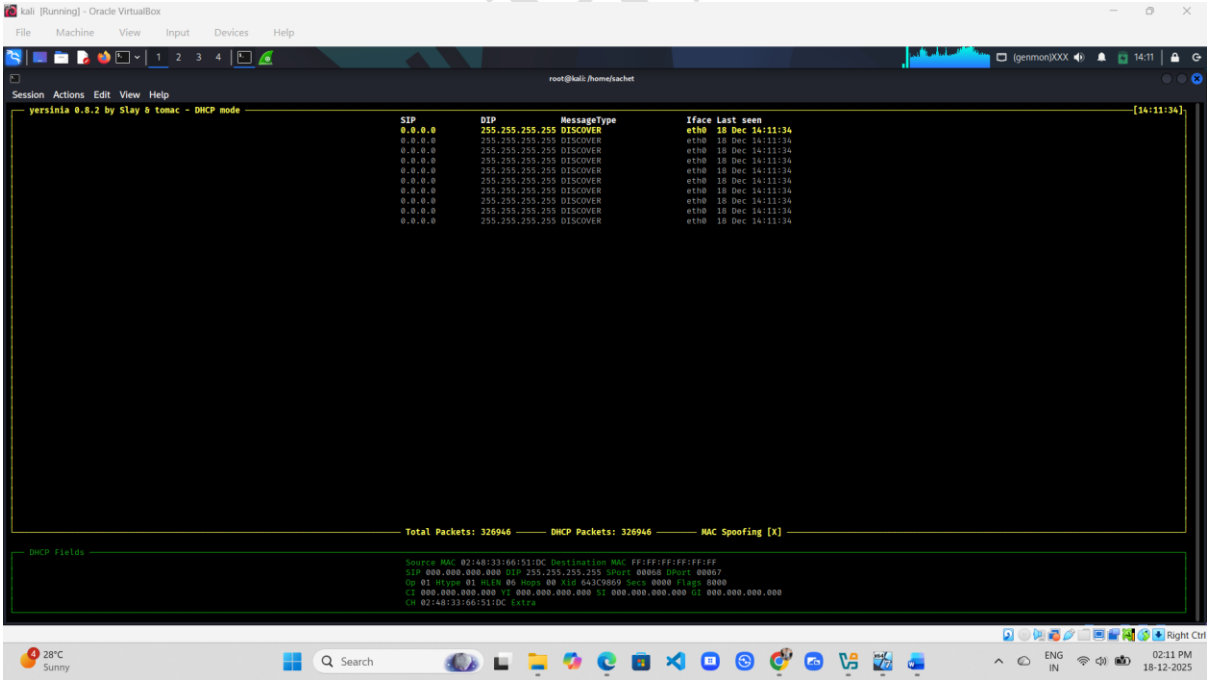
Type – yersinia -I



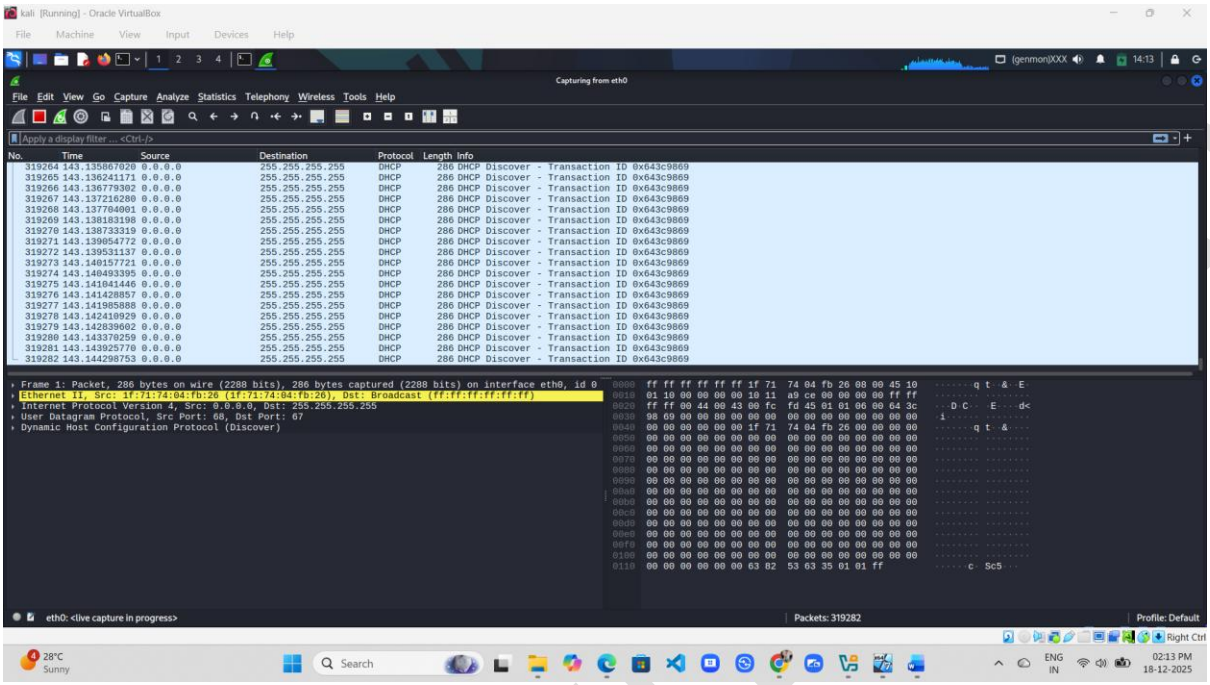
- **Now perform attack**




kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
root@kali: /home/sachet
Session Actions Edit View Help
yersinia 0.8.2 by Slay & tomac - DHCP mode
SIP DIP MessageType Iface Last seen
0.0.0.0 255.255.255.255 DISCOVER eth0 18 Dec 14:10:38
Attack Panel
No DoS Description
0 sending RAW packet
1 X sending DISCOVER packet
2 creating DHCP rogue server
3 X sending RELEASE packet
Select attack to launch ('q' to quit)
Total Packets: 24759
DHCP Packets: 24759
MAC Spoofing [X]
Those strange attacks ...
DHCP Fields
Top Stories
Growth over 8%!...
Search
ENG
IN
02:10 PM
18-12-2025


kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
root@kali: /home/sachet
Session Actions Edit View Help
yersinia 0.8.2 by Slay & tomac - DHCP mode
SIP DIP MessageType Iface Last seen
0.0.0.0 255.255.255.255 DISCOVER eth0 18 Dec 14:11:34
Total Packets: 326946
DHCP Packets: 326946
MAC Spoofing [X]
DHCP Fields
28°C
Sunny
Search
ENG
IN
02:11 PM
18-12-2025

- **Here , DHCP Packets are send to the target**



Conclusion

DHCP Starvation exposes a critical weakness in networks that rely on DHCP without adequate protection. By exhausting the IP address pool, an attacker can deny network access to legitimate users without directly targeting them. This attack highlights the importance of implementing security measures such as DHCP snooping, port security, and MAC address filtering to ensure network availability and resilience.

Perform Network Sniffing using Various Sniffing Tools

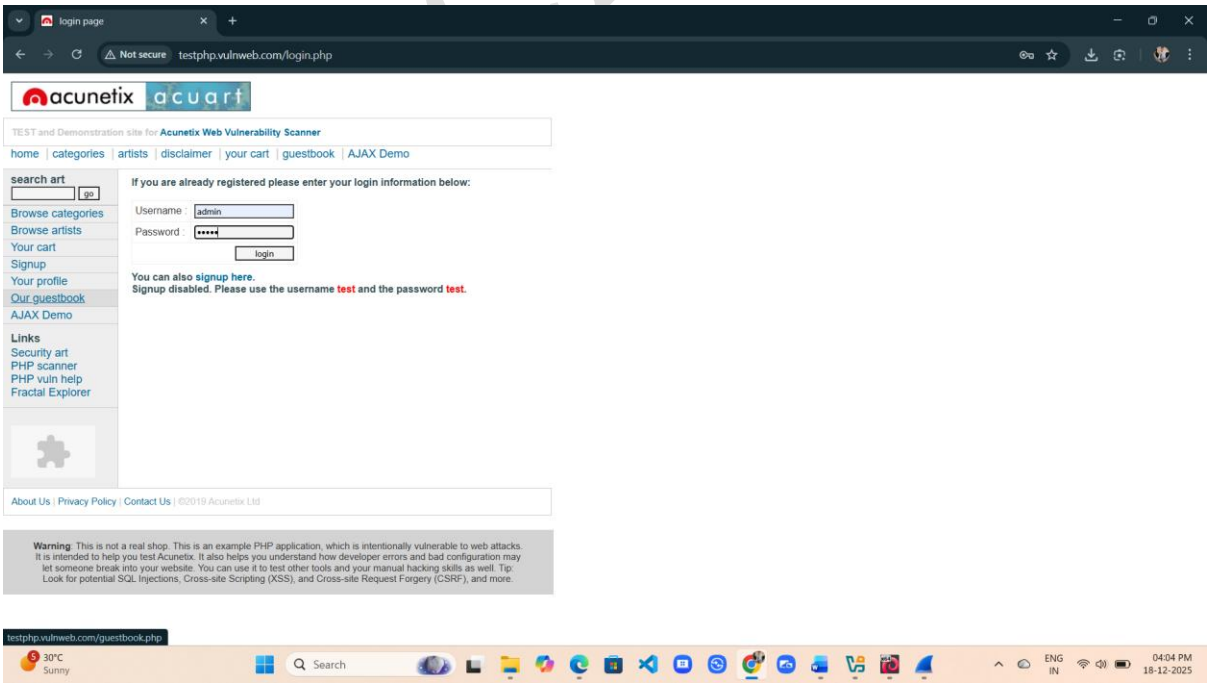
Network sniffing is widely used by ethical hackers and penetration testers to monitor and analyze network traffic. Various sniffing tools make this task efficient and practical.

Lab Scenario

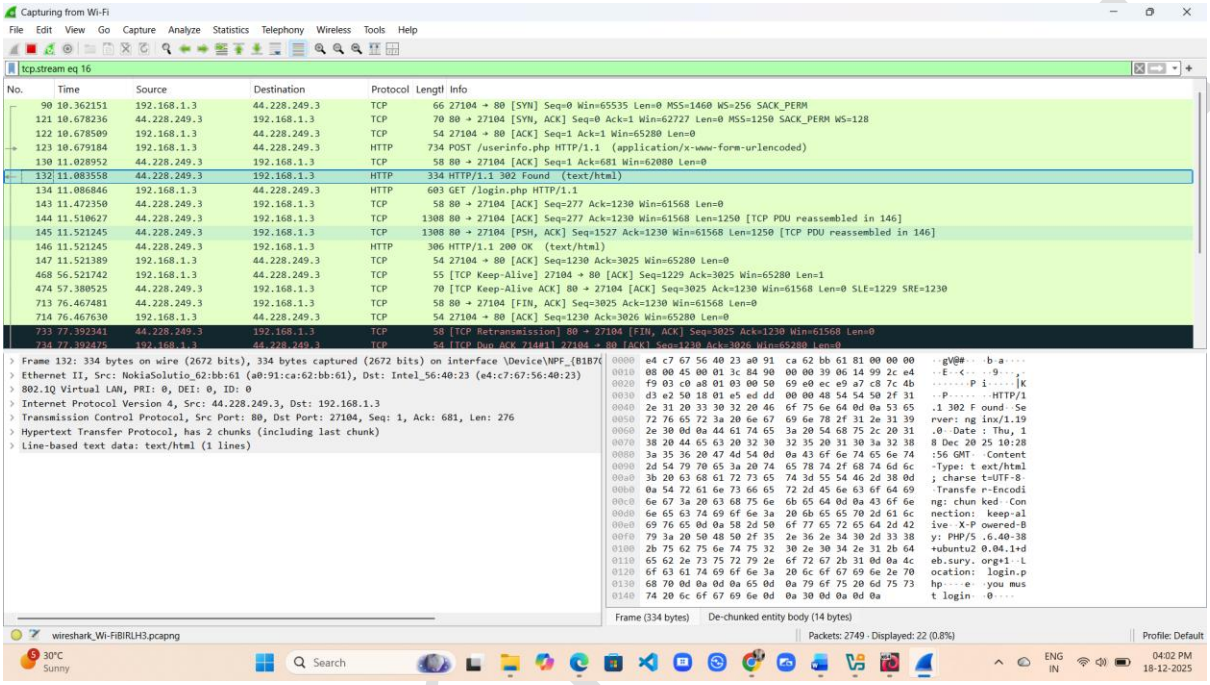
Data transmitted over the HTTP protocol travels in plain-text format, which makes it vulnerable to Man-in-the-Middle (MITM) attacks. Network administrators use sniffing tools to troubleshoot network issues, analyze security problems, and debug protocol implementations. However, attackers can also misuse sniffing tools such as Wireshark to capture and analyze traffic flowing between systems on a network.

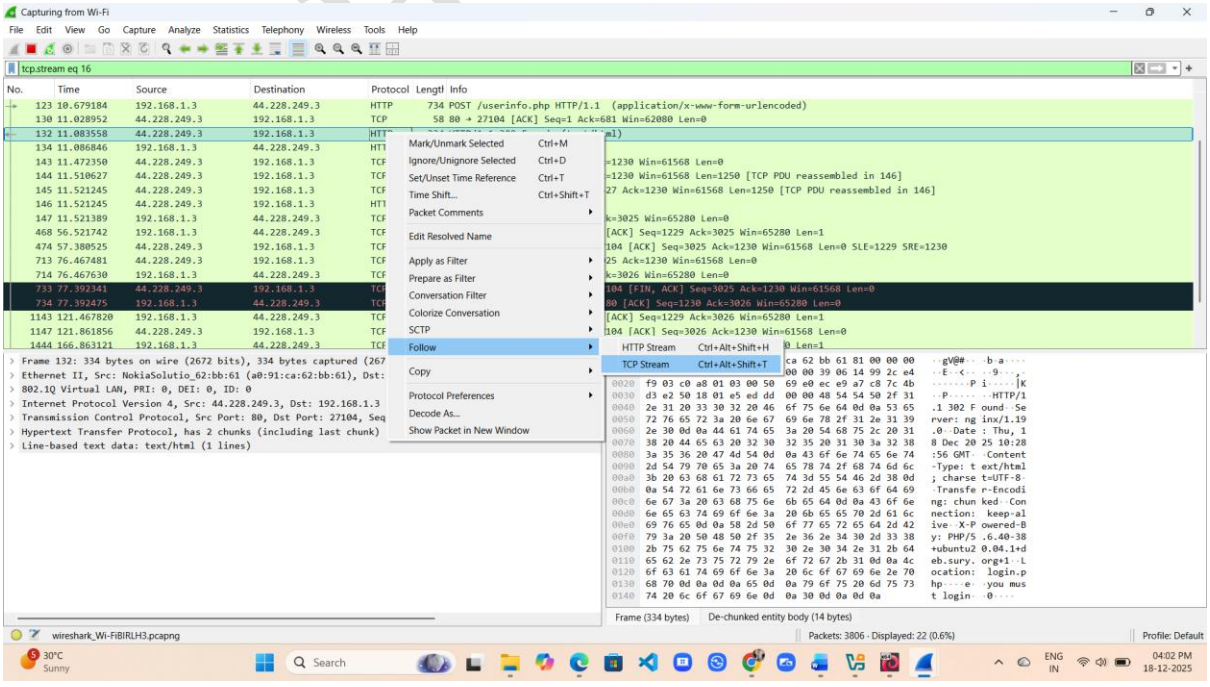
Perform Password Sniffing using Wireshark

- Entered username and passwords

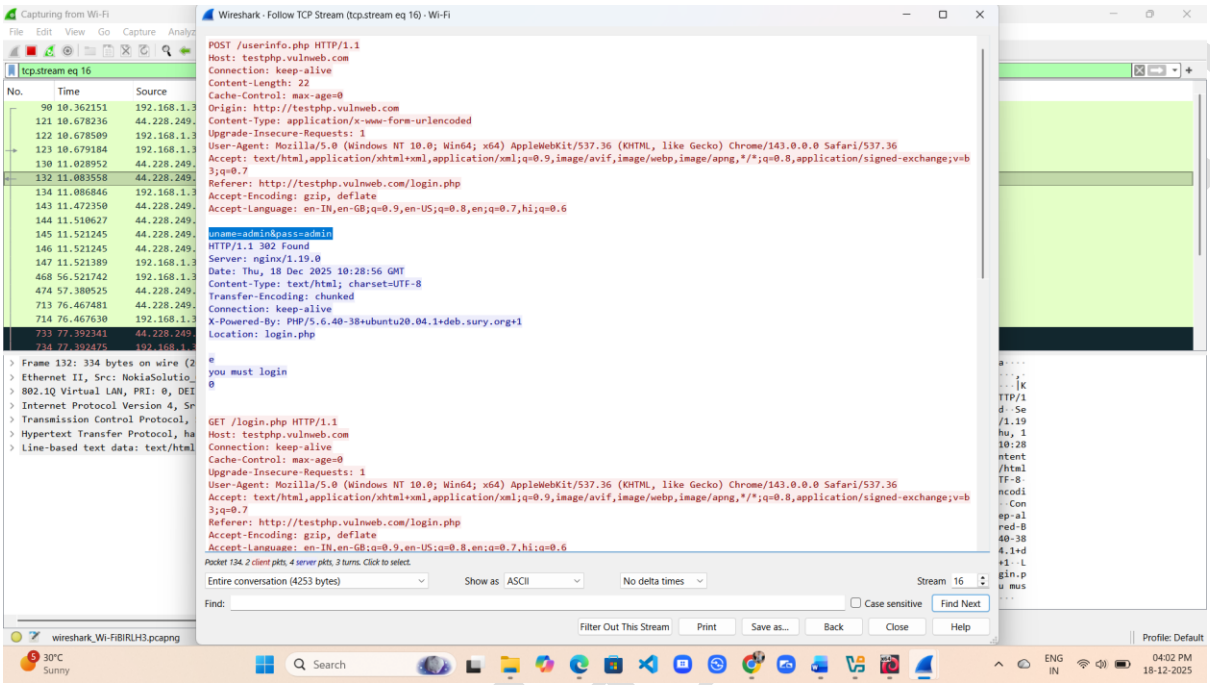


- Now back to the wireshark and find POST packet
- Here , post packet .. Open it





- It captures the Credentials



Detecting Network Sniffing

Ethical hackers and pen testers don't just learn how attacks work—they learn how attacks **fail**. Sniffing lives in the shadows, but networks whisper when something's wrong. The trick is listening the *right* way.

Back in the day, hubs made sniffing laughably easy—plug in, capture everything, go home. Switches changed the game. Today, sniffing needs tricks, and tricks leave scars.

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

Working of hping3

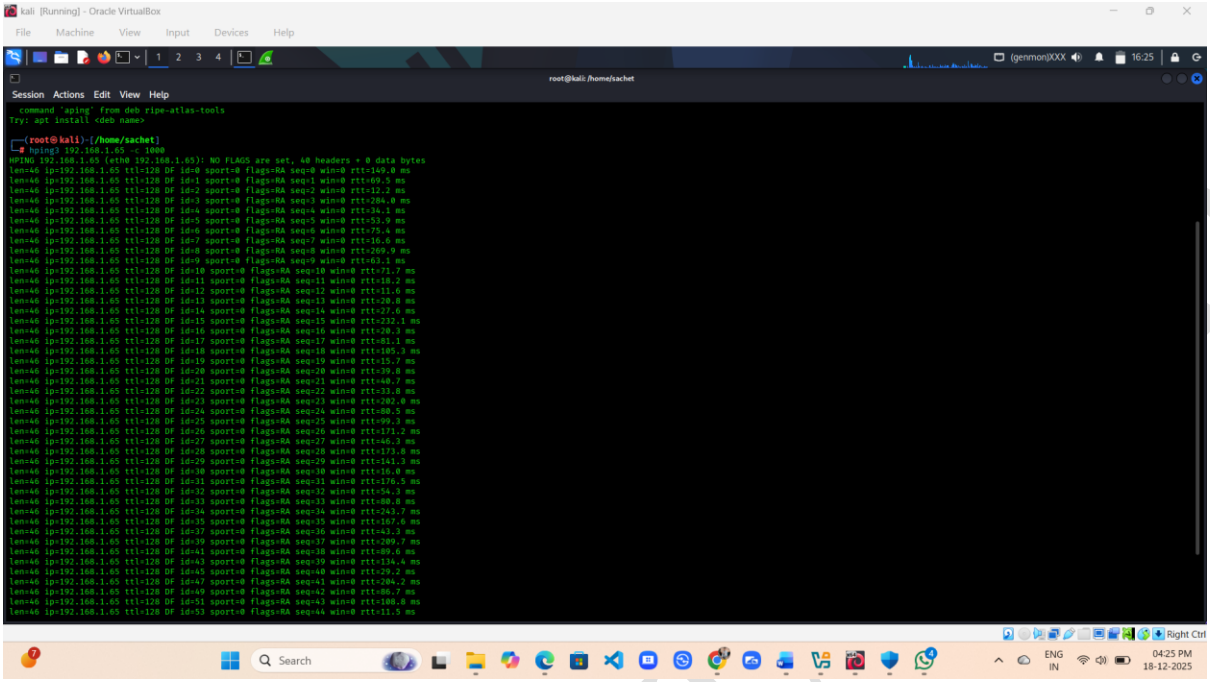
hping3 is a command-line packet crafting tool used to send custom TCP/IP packets to a target system. It allows control over packet type, flags, count, and speed. By analyzing the target's responses—or lack of response—network behavior, firewall rules, and traffic handling can be studied. When large numbers of packets are sent, abnormal network activity becomes visible in monitoring tools like Wireshark.

How to use it -:

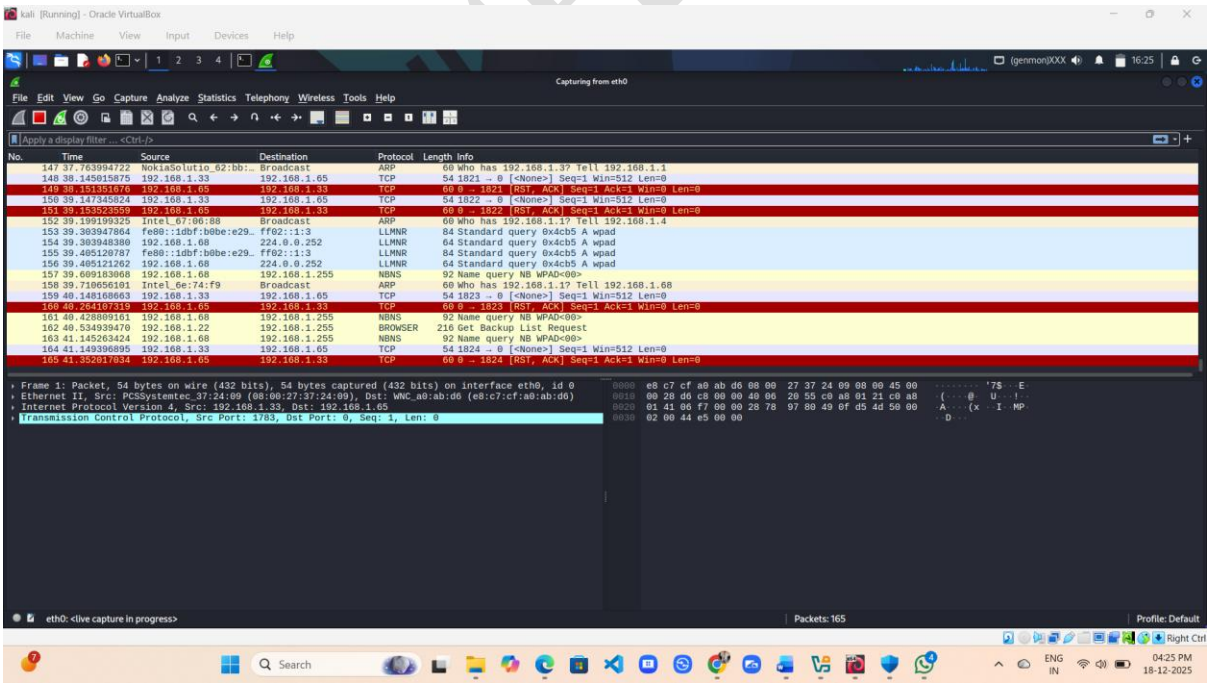
- Open kali linux Terminal and type **man hping3** – To get Detailed information about hping3

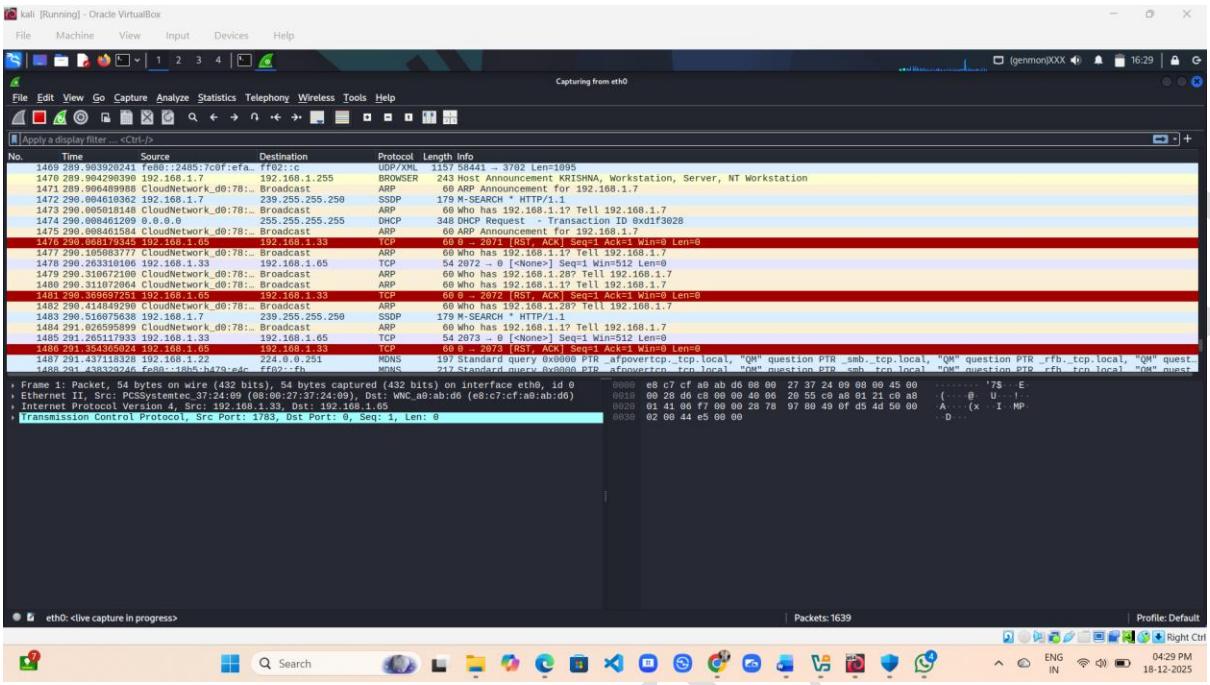
Command - hping3 192.168.1.59 -c 1000

- Attack Start



- Now open wireshark to analys packets
- Packets are send to the target





Conclusion

hping3 demonstrates how crafted packets can stress a network and expose security weaknesses. It highlights the need for proper firewall configuration, rate limiting, and intrusion detection systems to prevent misuse and protect network stability.

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

ARP Poisoning is a **Man-in-the-Middle (MITM) attack** technique in which an attacker exploits the Address Resolution Protocol (ARP) to intercept communication between two networked devices. The attacker deceives both the victim machine and the network gateway by sending forged ARP replies, causing traffic meant for one device to be redirected through the attacker's system.

Working of ARP Poisoning Attack

During the attack, forged ARP messages were continuously sent to both the victim and the gateway. These messages falsely claimed that the attacker's MAC address corresponded to the legitimate IP addresses of each device. Because ARP lacks authentication, both devices accepted the false mappings without verification.

Once the ARP tables were poisoned:

- The attacker successfully intercepted network traffic
- Data packets flowed through the attacker's system
- Sensitive information transmitted in plaintext became visible

How to use it -:

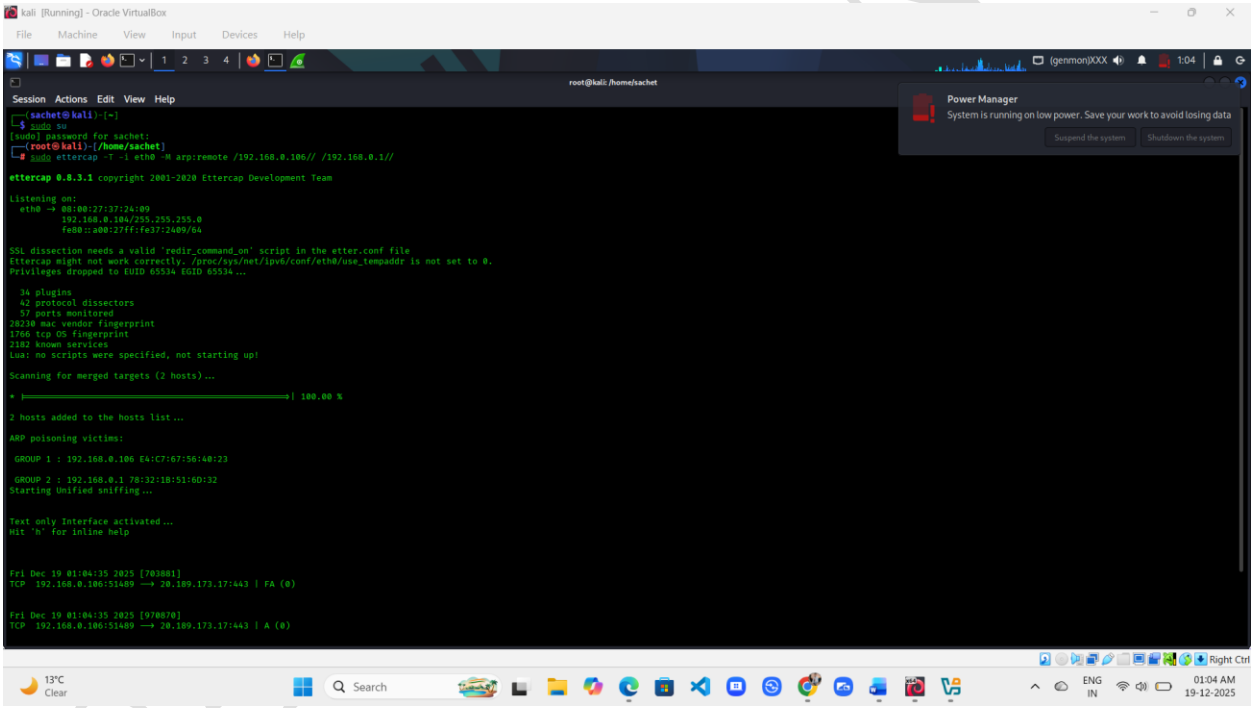
• Open Kali linux

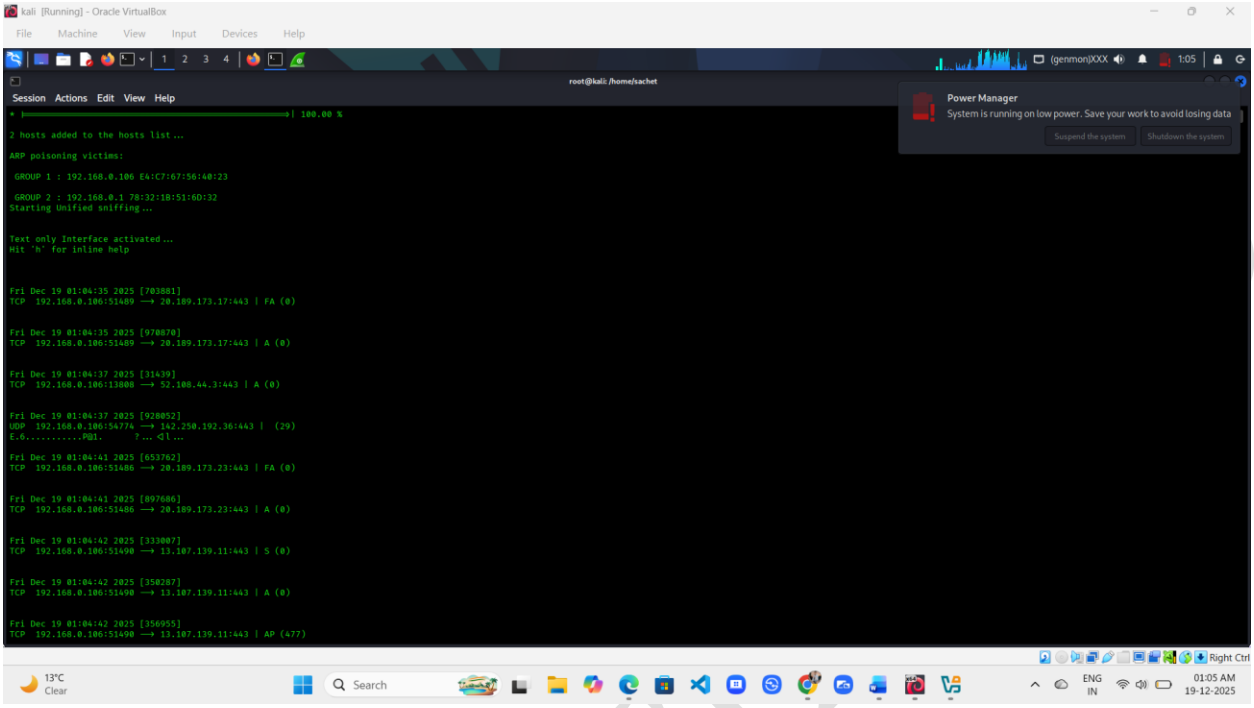
Command –

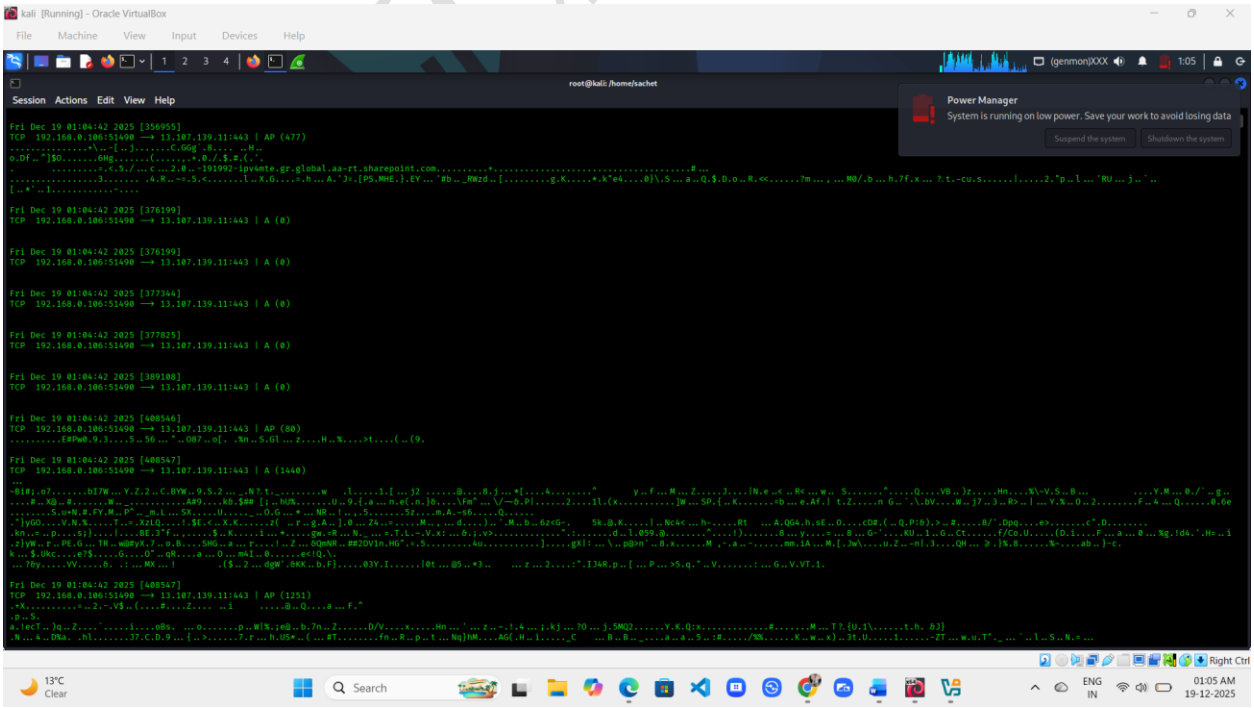
```
sudo ettercap -T -i eth0 -M arp:remote /10.81.39.20// /10.81.39.194//
```

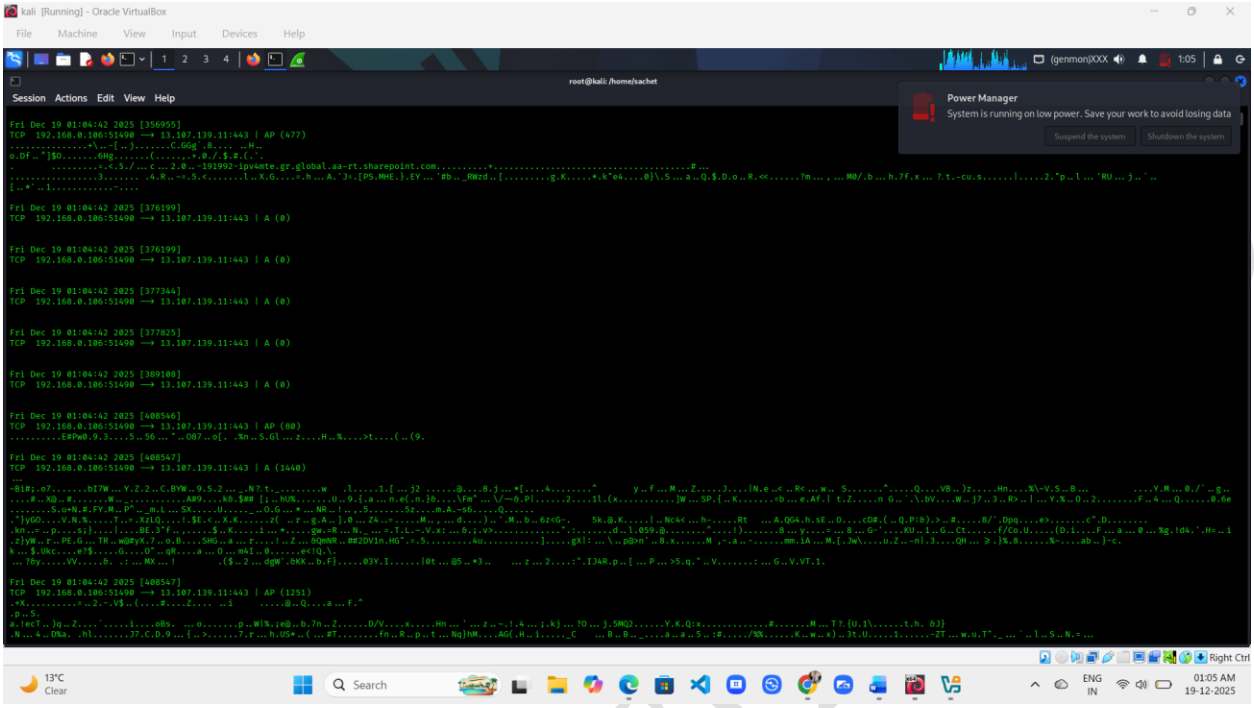
was used to perform an **ARP poisoning Man-in-the-Middle attack** in a controlled environment.

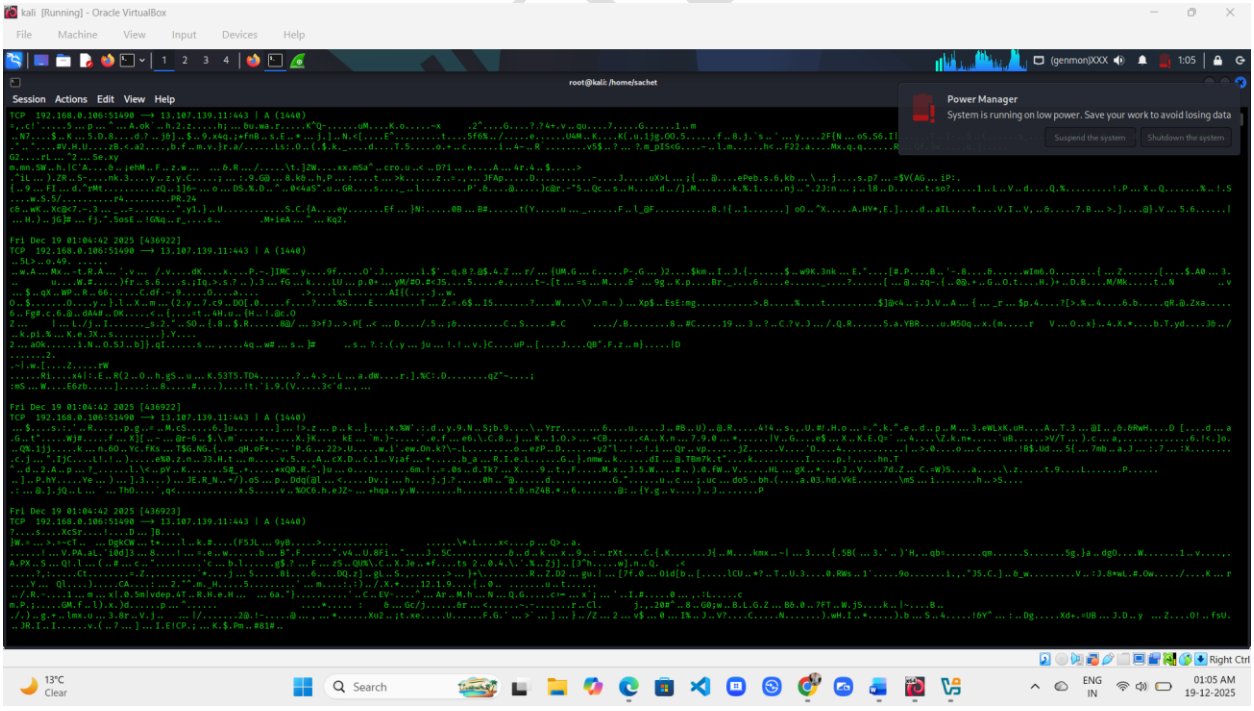
- `sudo` → required because you're touching the network stack
- `ettercap` → MITM framework
- `-T` → text-only interface
- `-i eth0` → specifies the network interface
- `-M arp:remote` → selects ARP-based MITM mode
- `/IP1/ /IP2/` → two endpoints whose traffic is being intercepted

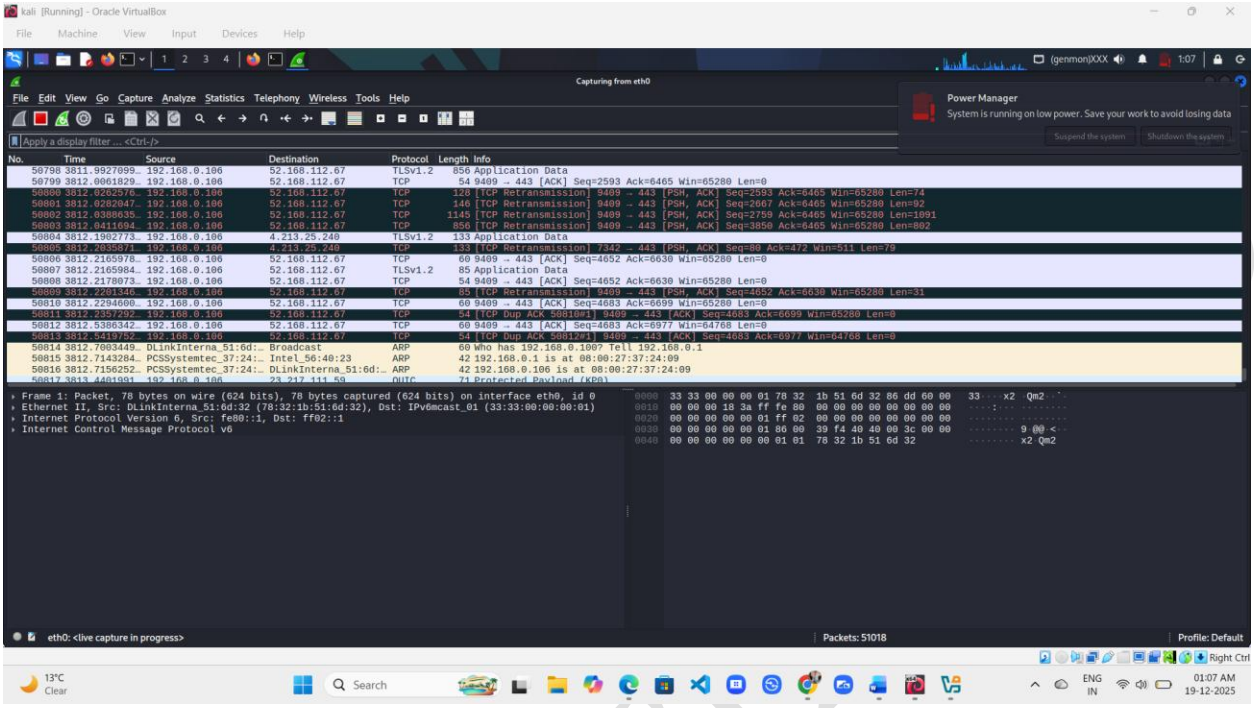



kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
root@kali: /home/sachet
Session Actions Edit View Help
Power Manager
System is running on low power. Save your work to avoid losing data
Suspend the system
Shutdown the system
2 hosts added to the hosts list ...
ARP poisoning victims:
GROUP 1 : 192.168.0.106 E4:C7:67:56:40:23
GROUP 2 : 192.168.0.1 78:32:1B:51:6D:32
Starting Unified sniffing ...
Text only Interface activated ...
Hit 'h' for inline help
Fri Dec 19 01:04:35 2025 [703881]
TCP 192.168.0.106:51489 --> 20.189.173.17:443 | FA (0)
Fri Dec 19 01:04:35 2025 [970870]
TCP 192.168.0.106:51489 --> 20.189.173.17:443 | A (0)
Fri Dec 19 01:04:37 2025 [31439]
TCP 192.168.0.106:13808 --> 52.108.44.3:443 | A (0)
Fri Dec 19 01:04:37 2025 [928052]
Fri Dec 19 01:04:41 2025 [653762]
TCP 192.168.0.106:51486 --> 20.189.173.23:443 | FA (0)
Fri Dec 19 01:04:41 2025 [897686]
TCP 192.168.0.106:51486 --> 20.189.173.23:443 | A (0)
Fri Dec 19 01:04:42 2025 [333007]
TCP 192.168.0.106:51490 --> 13.107.139.11:443 | S (0)
Fri Dec 19 01:04:42 2025 [358287]
TCP 192.168.0.106:51490 --> 13.107.139.11:443 | A (0)
Fri Dec 19 01:04:42 2025 [356955]
TCP 192.168.0.106:51490 --> 13.107.139.11:443 | AP (477)
Right Ctrl
13°C Clear
Search
ENG IN
01:05 AM 19-12-2025


kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
root@kali: /home/sachet
Session Actions Edit View Help
Power Manager
System is running on low power. Save your work to avoid losing data
Suspend the system
Shutdown the system
Fri Dec 19 01:04:42 2025 [356955]
TCP 192.168.0.106:51490 --> 13.107.139.11:443 | AP (477)
Fri Dec 19 01:04:42 2025 [376199]
TCP 192.168.0.106:51490 --> 13.107.139.11:443 | A (0)
Fri Dec 19 01:04:42 2025 [376199]
TCP 192.168.0.106:51490 --> 13.107.139.11:443 | A (0)
Fri Dec 19 01:04:42 2025 [377344]
TCP 192.168.0.106:51490 --> 13.107.139.11:443 | A (0)
Fri Dec 19 01:04:42 2025 [377825]
TCP 192.168.0.106:51490 --> 13.107.139.11:443 | A (0)
Fri Dec 19 01:04:42 2025 [389108]
TCP 192.168.0.106:51490 --> 13.107.139.11:443 | A (0)
Fri Dec 19 01:04:42 2025 [408546]
TCP 192.168.0.106:51490 --> 13.107.139.11:443 | AP (80)
Fri Dec 19 01:04:42 2025 [408547]
TCP 192.168.0.106:51490 --> 13.107.139.11:443 | A (1440)
Fri Dec 19 01:04:42 2025 [408547]
TCP 192.168.0.106:51490 --> 13.107.139.11:443 | AP (1251)
Right Ctrl
13°C Clear
Search
ENG IN
01:05 AM 19-12-2025


kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
root@kali: /home/sachet
Session Actions Edit View Help
Power Manager
System is running on low power. Save your work to avoid losing data
Suspend the system
Shutdown the system
Right Ctrl
13°C
Clear
Search
ENG
IN
01:05 AM
19-12-2025


kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
root@kali: /home/sachet
Session Actions Edit View Help
Power Manager
System is running on low power. Save your work to avoid losing data
Suspend the system
Shutdown the system
Right Ctrl
13°C
Clear
Search
ENG
IN
01:05 AM
19-12-2025



Conclusion

ARP Poisoning demonstrates a significant weakness in the ARP protocol due to its lack of authentication. By exploiting this vulnerability, attackers can silently position themselves between communicating devices and intercept sensitive data. This experiment highlights the importance of implementing security measures such as dynamic ARP inspection, encrypted communication protocols, and network monitoring to prevent MITM attacks.

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.

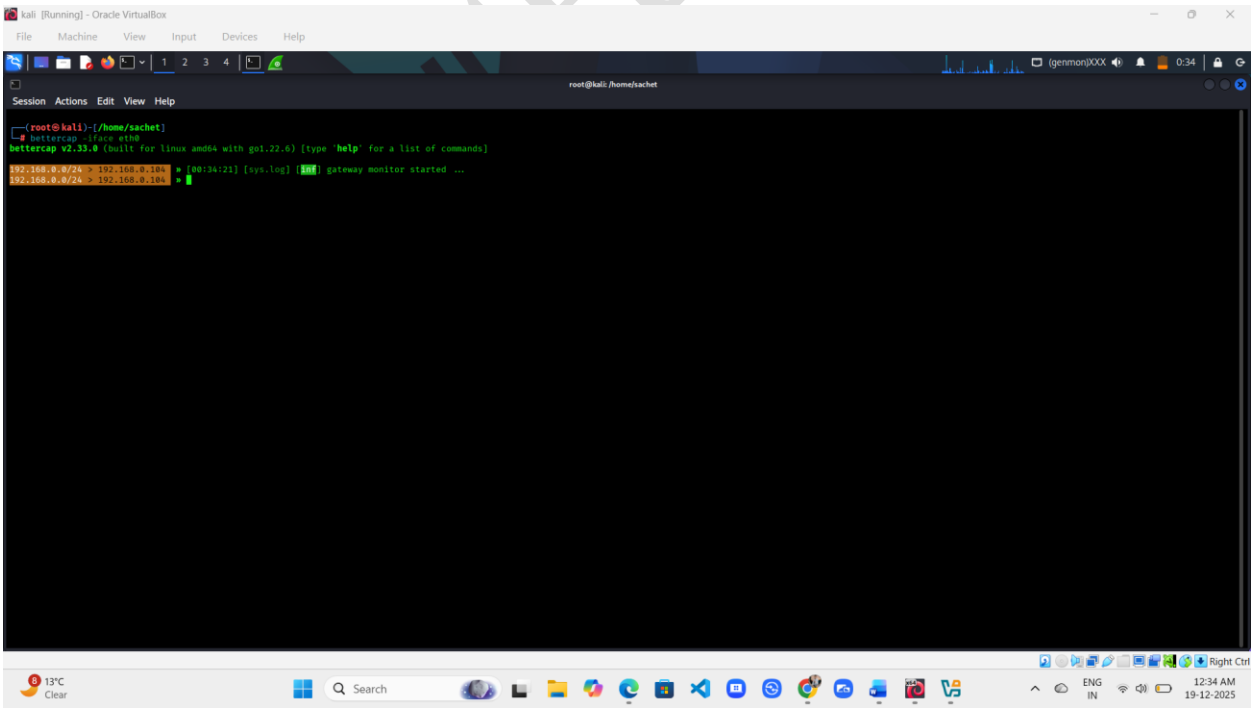
ARP Poisoning is a Man-in-the-Middle (MITM) attack in which an attacker exploits the lack of authentication in the ARP protocol to intercept communication between a victim system and the network gateway.

Bettercap is a modern, powerful, and flexible network security tool used for MITM attacks, packet sniffing, traffic manipulation, credential harvesting, and protocol analysis. It is widely used in penetration testing to demonstrate real-world network vulnerabilities.

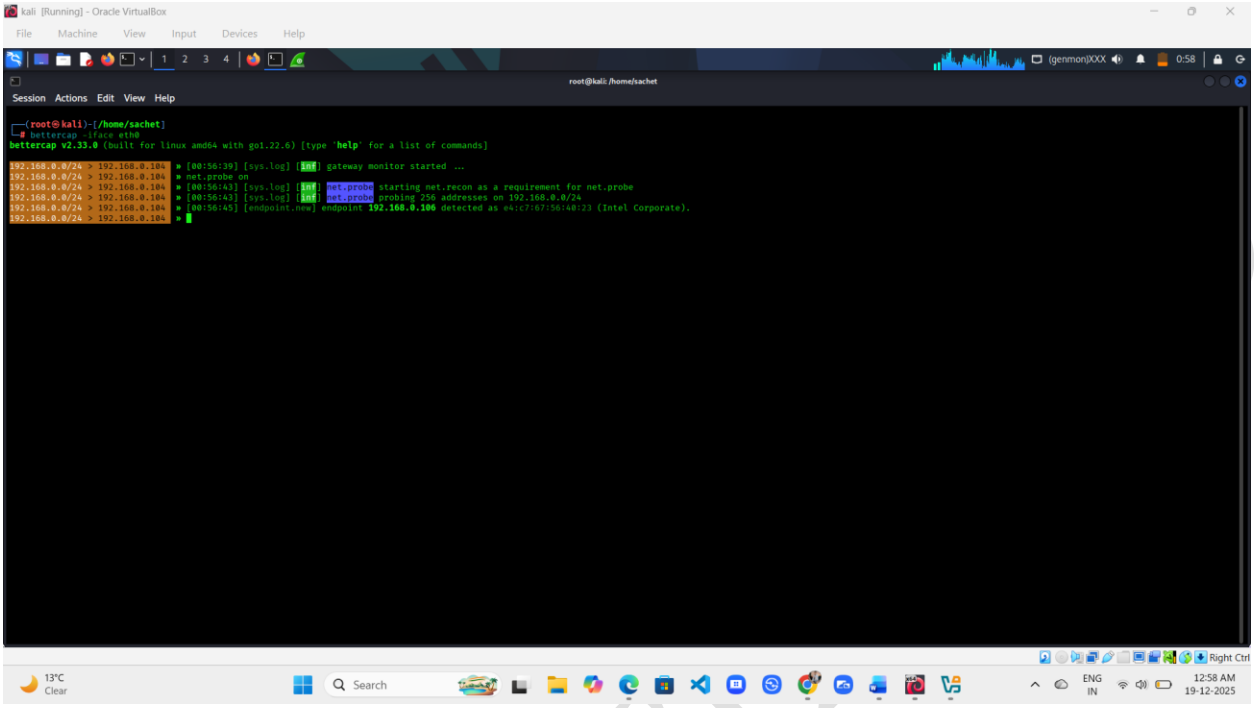
How to use it - :

- Open Kali linux and open terminal
- And type bettercap

Command - sudo bettercap -iface eth0



- Type net.probe on – to scan devices in network



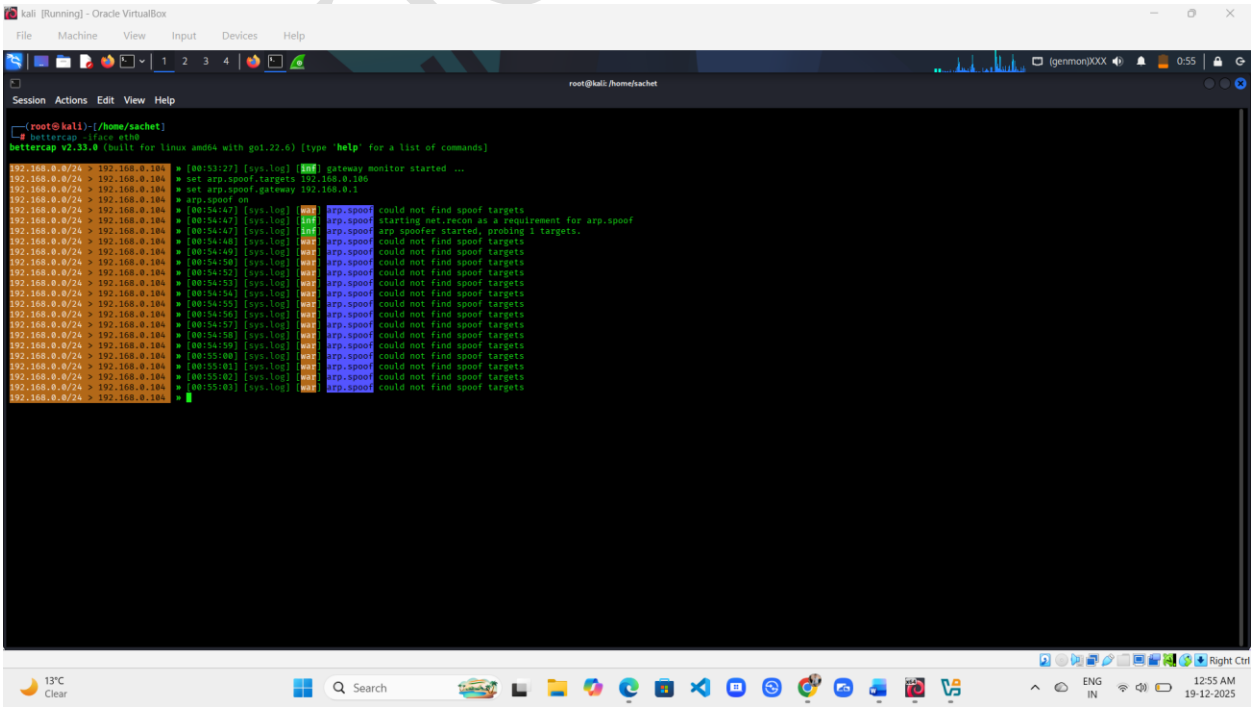
• Now set a target

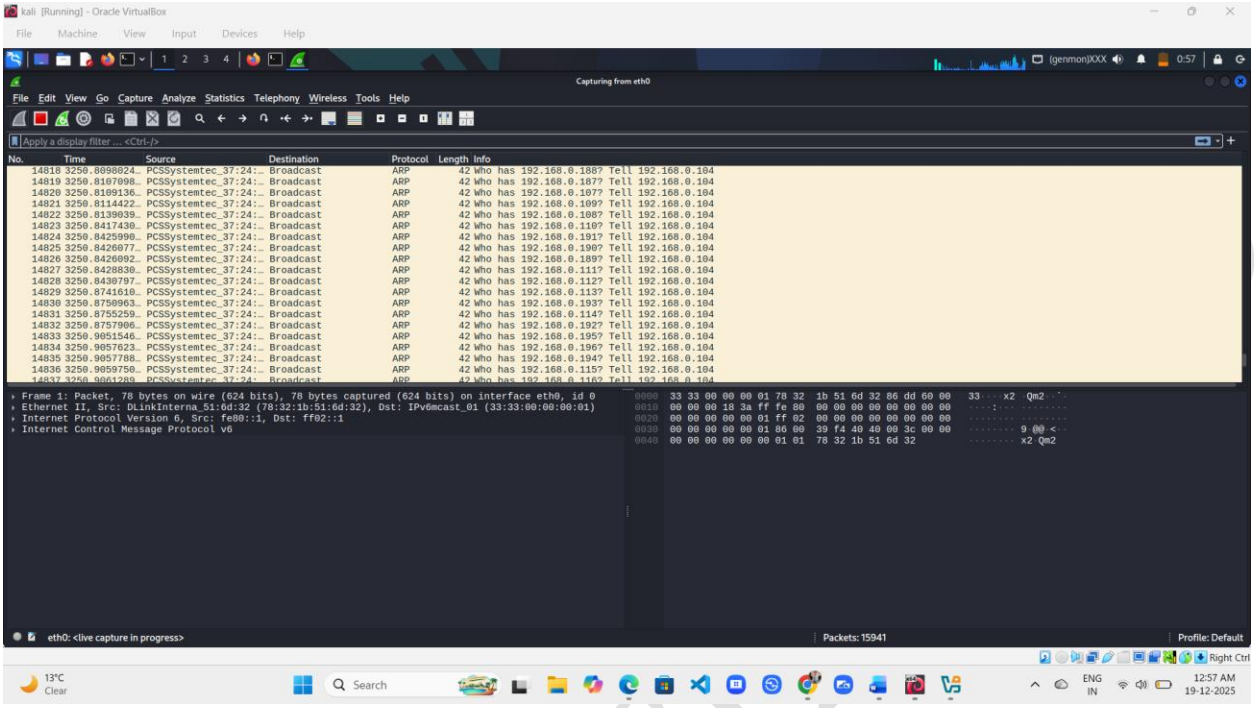
Command -:

set arp.spoof.targets 10.81.39.20

set arp.spoof.gateway 10.81.39.194

arp.spoof on.





Conclusion

This study shows that attacks like sniffing, MAC flooding, DHCP starvation, and ARP poisoning exploit weaknesses in trusted network protocols. Without proper security measures, attackers can intercept or disrupt network communication. Implementing encryption and network security controls is essential to prevent such attacks.

Sniffing Countermeasures

The earlier discussion showed how attackers capture network traffic using various sniffing techniques and tools. Sniffing becomes especially dangerous when it is **passive**, as passive sniffers do not generate noticeable traffic and are extremely difficult to detect—particularly on shared Ethernet environments.

To counter sniffing attacks, a network must rely on **prevention, detection, and control mechanisms** rather than assumptions of trust. The following countermeasures and defensive techniques are commonly used to protect networks against sniffing.

---

1. Use Encryption

Old rule, still undefeated: **what can't be read can't be stolen**.

- Use **HTTPS instead of HTTP**
- Use **SSH instead of Telnet**
- Implement **IPsec, TLS, SSL**
- Enable **WPA2/WPA3** for wireless networks

Even if traffic is sniffed, encryption renders captured data useless.

---

2. Replace Hubs with Switches

Hubs gossip. Switches mind their business.

- Switches forward packets only to the intended destination
- Reduces the effectiveness of passive sniffing
- Prevents attackers from seeing all network traffic

This is basic hygiene, not optional luxury.

---

3. Implement Port Security

Discipline beats cleverness.

- Bind MAC addresses to switch ports
- Limit the number of MACs per port
- Disable unused ports

If a rogue device connects, the switch notices—and reacts.

---

4. Detect Promiscuous Mode

A NIC in promiscuous mode listens when it shouldn't.

- Send crafted ARP or ICMP packets
- Monitor abnormal responses
- Use tools like **Nmap** and **Sniffer detection scripts**

Sniffers try to hide. Detection pokes them until they flinch.

---

5. Monitor ARP Traffic

Sniffing often walks hand-in-hand with ARP poisoning.

- Use **Dynamic ARP Inspection (DAI)**
- Monitor ARP tables for frequent changes
- Detect duplicate IP–MAC mappings

ARP lies loudly—if you're watching.

---

6. Use Intrusion Detection Systems (IDS)

Let machines watch the machines.

- Detect unusual traffic patterns
- Identify packet flooding and ARP spoofing
- Alert administrators in real time

Tools like **Snort** and **Suricata** don't sleep.

---

7. Network Traffic Analysis

Patterns reveal intent.

- Monitor bandwidth usage
- Detect abnormal packet rates

- Identify unauthorized packet capture behavior

When someone listens too closely, the network feels it.

Sniffing Detection Techniques

- Monitoring systems for **promiscuous mode**
- Observing **ARP cache anomalies**
- Identifying **unexpected latency or packet loss**
- Detecting **high-volume or malformed packets**

Passive sniffing is stealthy—but not perfect.

Conclusion

Sniffing attacks thrive on trust and silence. Defensive networks survive on **encryption, monitoring, and discipline**. While passive sniffers are difficult to detect, combining strong encryption, switched networks, ARP inspection, and intrusion detection significantly reduces the risk of successful sniffing attacks.

Module Summary

10.1 Overview of Sniffing Concepts

This module introduced the fundamental concept of network sniffing and its role in both network administration and cyber attacks. It explained how data packets travel across a network and how sniffers capture this data at the Data Link Layer of the OSI model. The module highlighted the security risks associated with sniffing, especially when sensitive information is transmitted without encryption.

10.2 Summary of Sniffing Techniques

Various sniffing techniques were discussed, including MAC flooding, DHCP starvation, ARP poisoning, spoofing attacks, and VLAN hopping. Each technique demonstrated how attackers exploit weaknesses in network protocols and switching mechanisms to intercept or disrupt network communication. These techniques emphasize that trusted protocols can become attack vectors if not properly secured.

10.3 Tools Used for Network Sniffing

The module illustrated the use of multiple sniffing and attack tools such as Wireshark, macof, Yersinia, hping3, Ettercap, and Bettercap. These tools were used in controlled environments to capture network traffic, analyze packet flow, and demonstrate real-world vulnerabilities. Understanding these tools helps ethical hackers and network administrators assess and strengthen network security.

10.4 Sniffing Countermeasures Discussed

Effective countermeasures such as encryption, switched networks, port security, DHCP snooping, Dynamic ARP Inspection, and Intrusion Detection Systems were emphasized. The module showed that preventing sniffing attacks requires layered security rather than relying on a single defense mechanism. Proper configuration and continuous monitoring play a critical role in reducing attack surfaces.

10.5 Importance of Sniffing Detection

Sniffing detection is essential because many sniffing attacks are passive and difficult to identify. The module highlighted techniques such as detecting promiscuous mode, monitoring ARP anomalies, and analyzing abnormal traffic patterns. Early detection allows administrators to respond quickly, minimize data exposure, and maintain network integrity.

THANK YOU