Kingdom of Saudi Arabia
Ministry of Higher Education
King Faisal University
College of Computer Sciences & Information Technology

NETWORK MAN-IN-THE-MIDDLE ATTACK

Name	Student ID
Basmah Alshakhes	219025990
Jinan Alkhammas	219008852
Ghufran Abdullah	219037244

Supervised by: Dr. Abdulla AL-BoAli

1 Table of Contents

1. EXECUTIVE SUMMARY

The purpose of this report is to provide a thorough analysis of two critical network security sections; Man-in-the-Middle (MITM) attacks (using Ettercap) and ARP poisoning attacks. Ettercap is basically an open-source tool known for its ability to intercept network traffic, capture sensitive information, and conduct active eavesdropping. Moreover, ARP poisoning attacks exploit weaknesses in the Address Resolution Protocol to redirect traffic and gain unauthorized access.

2 OBJECTIVES

The objectives of this project are as follows:

- To understand the methodology of a Man-in-the-Middle (MITM) attack using Ettercap
- To assess the potential impact of a MITM attack on network security
- To examine the interception of network traffic during a MITM attack
- To understand the methodology and techniques involved in ARP poisoning attacks
- To analyze the vulnerabilities exploited by ARP poisoning attacks
- To examine the interception and redirection of network traffic in ARP poisoning attacks

3 FIRST ATTACK: ARP POIONING

3.1 OBJECTIVE OF ARP POIONING ATTACK

This lab aims to provide students with hands-on experience in conducting an ARP poisoning attack using Kali Linux and analyzing its impact using Wireshark. Students will learn how to manipulate the ARP protocol to intercept network traffic and perform a Man-in-the-Middle attack.

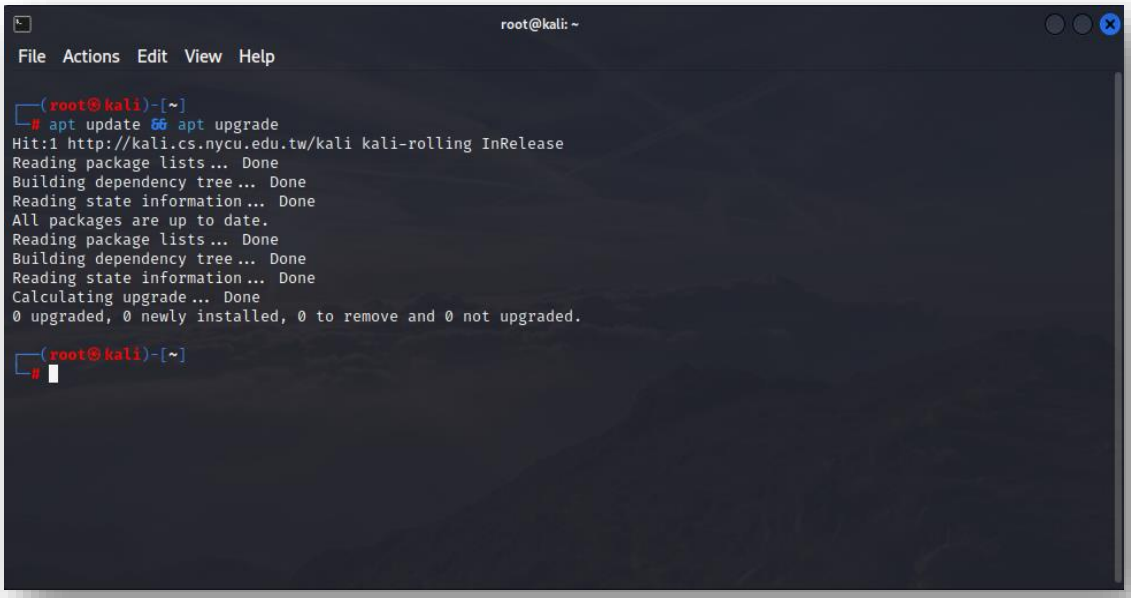
3.2 ABOUT ARP POISONING

ARP Poisoning, or ARP spoofing, is basically a network attack where the attacker manipulates the Address Resolution Protocol (ARP) to intercept and manipulate network traffic. By sending forged ARP messages, the attacker associates their own MAC address with the IP address of another device, redirecting traffic to their machine. This allows the attacker to eavesdrop, modify, or gain unauthorized access to network communications.

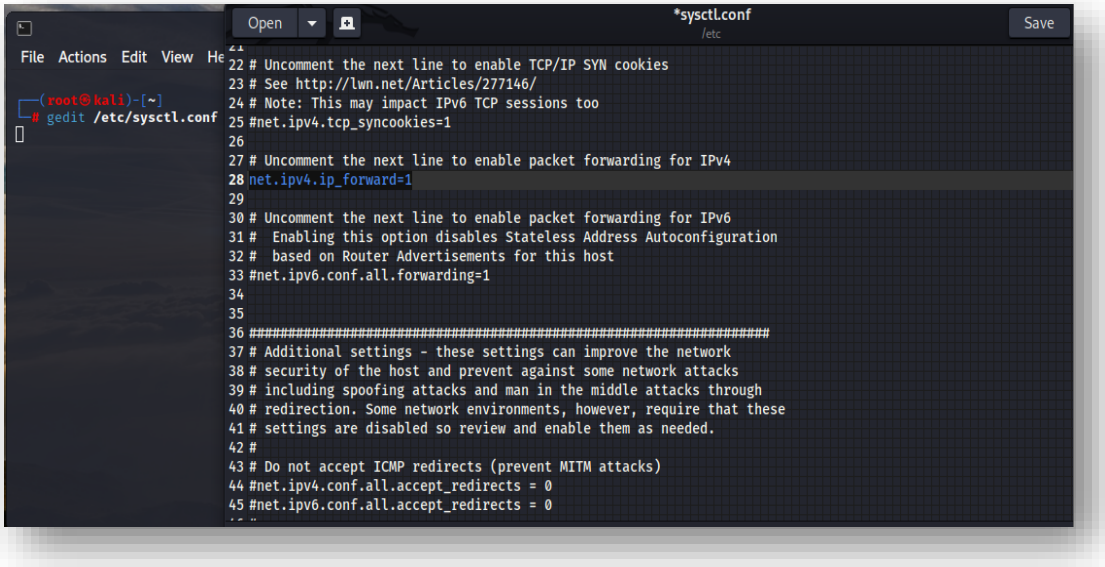
3.3 PROTECTION AGAINST ARP ATTACKS

- Implement ARP spoofing detection tools
- Configure static ARP entries
- Enable port security
- Utilize network monitoring and IDS systems
- Apply network segmentation and access control measures
- Regularly update and patch network devices

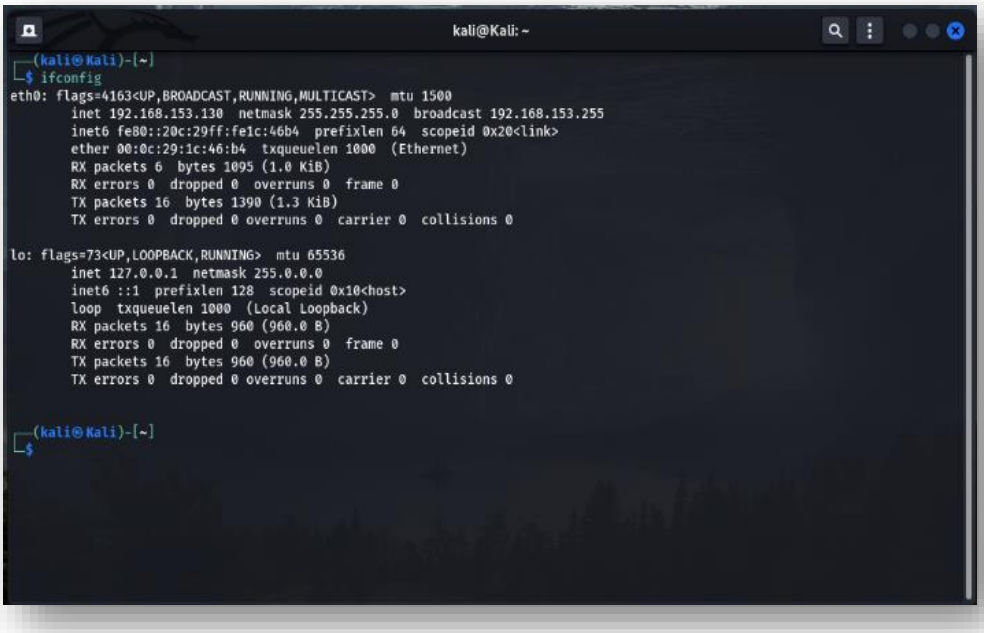
4 EVIDENCE / RELEVANT FINDINGS/ Analysis/ Exhibit of ARP POIONING ATTACK



Firstly, updated and upgraded Kali device to ensure that any available bug is fixed and the new features and packages are successfully installed.

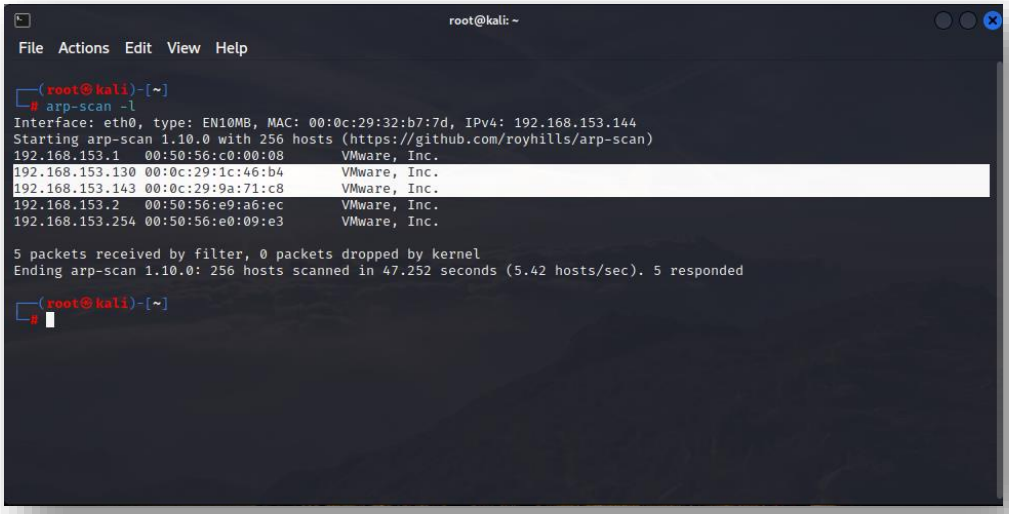


After that, enabled IP forwarding by uncommenting the command "**net.ipv4.ip_forward=1**" in the "**/etc/sysctl.conf**" file.

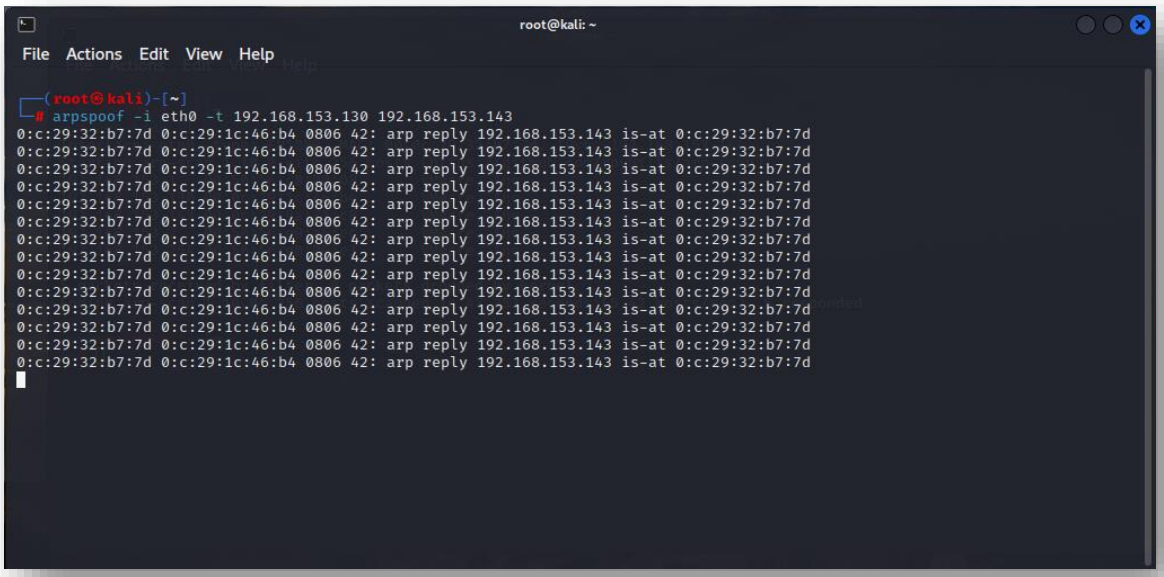


Now, this is my first victim device. Its IP is "**192.168.153.130**".

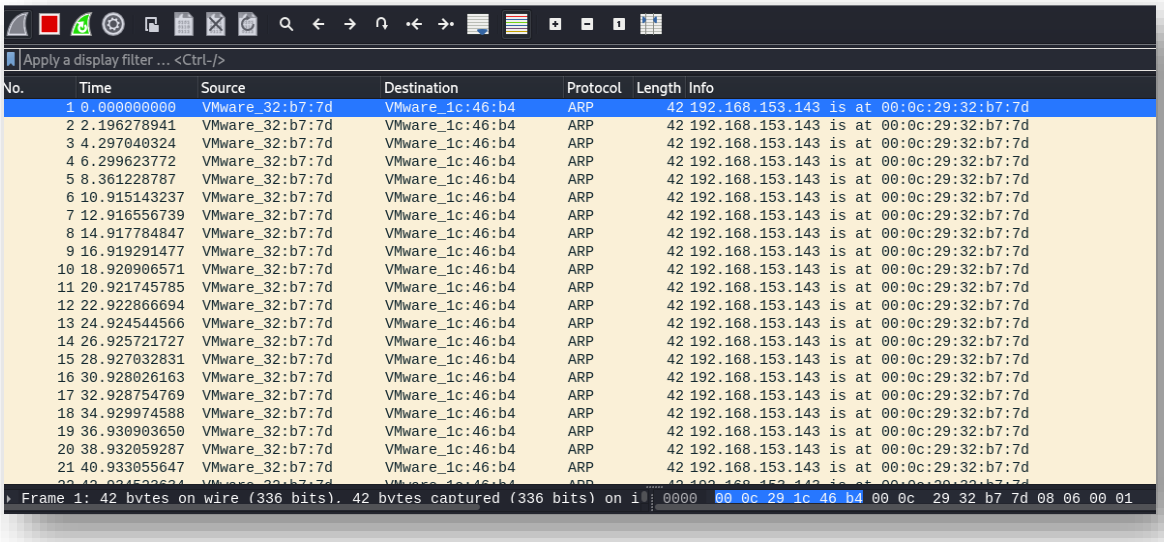
This is my second victim device. Its IP is "**192.168.153.143**".



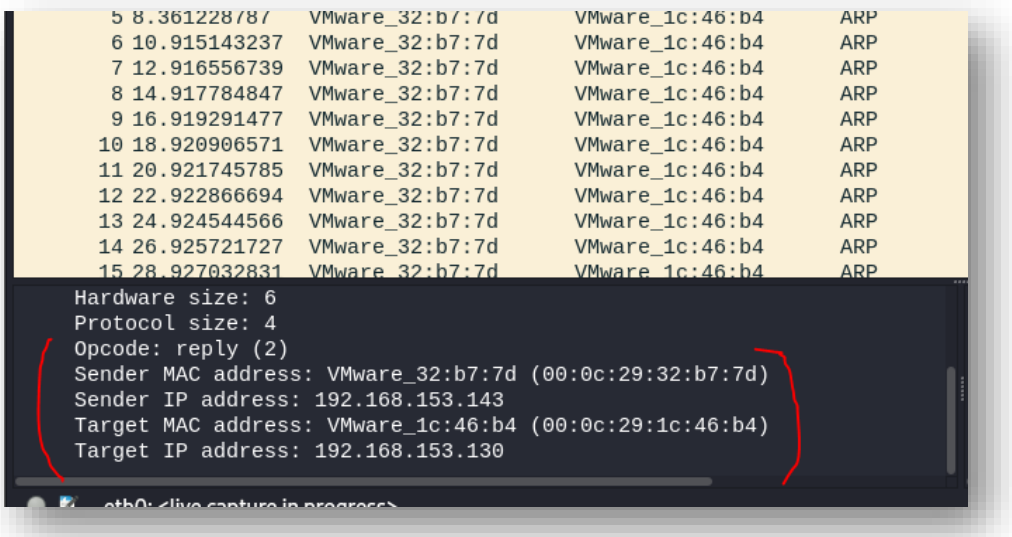
Now, on host device, we used the "**arp-scan -l**" to scan the local network. You can see that it has successfully detected the IP and MAC addresses of both of the victim devices.



Now, here we started the ARP Poisoning attack using "**arpspoof**". It will redirect the traffic between both the victims and that will pass through our host machine.



If we open **Wireshark** on our host machine, you can see that it shows the ARP packets that are being transferred.



Here, you can see that it shows the following values:

Sender:

MAC – X:X:X:X:b7:7d

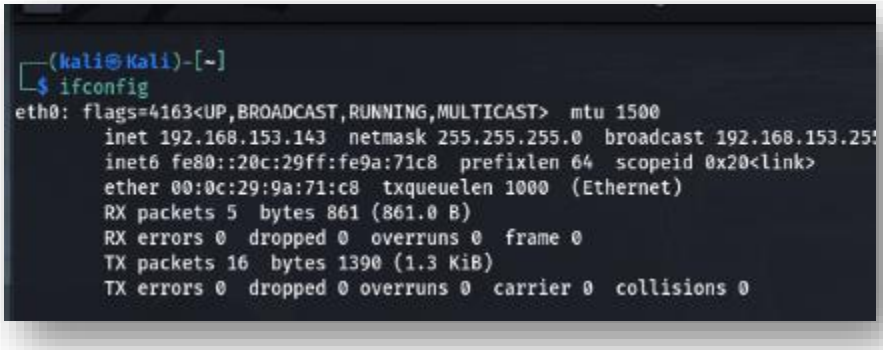
IP – 192.168.153.143

Target:

MAC – X:X:X:X:46:b4

IP – 192.168.153.130

However, the MAC of IP (**192.168.253.143**) is (**X:X:X:X:71:c8**)



Thus, we have successfully performed the ARP Poisoning attack.

REVIEW QUESTION OF ARP POISONING ATTACK LAB

1. **How did you identify the IP addresses of Victim 1 and Victim 2?**

   We identified the IP addresses of both the victims by using the "**ifconfig**" command. Since both the victims are my own devices, so we can easily access them. However, you can also identify victims' IP addresses by using the "**arp-scan -l**" command. It scans the local network.

2. **What command did you use to perform the ARP poisoning attack?**

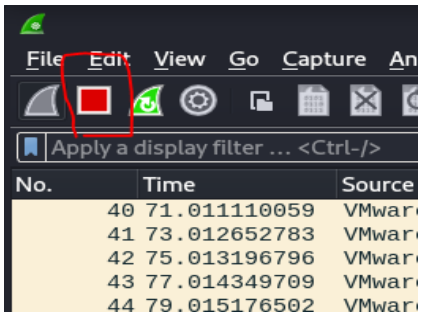
   The command that we used to perform ARP Poisoning is "**arpspoof -i <interface> -t <victim1_ip> <victim2_ip>**". Here, we will replace **<interface>** with the appropriate network interface (**eth0** or **wlan0**) and **<victim1_ip>** and **<victim2_ip>** with the IP addresses of Victim 1 and Victim 2 respectively.

3. **Which tool did you use to capture network traffic?**

   We used **Wireshark** to capture network traffic.

4. **How did you stop the packet capture in Wireshark?**

   In order to stop the packet capturing in Wireshark, we will click the "**Stop**" button (represented by a red square icon) in the toolbar.



5. **Explain the process you followed to filter the captured packets.**

   We simply entered "**ARP**" in the "**Filter Bar**" shown above the captured packets. Since there wasn't much activity going on, so it didn't require much filtering.

6. **What changes did you observe in the MAC addresses during the ARP poisoning attack.**

As stated above, the MAC address, which can allow the attacker to capture the communication which wasn't intended for him.

7. **Describe the potential impact of ARP poisoning on network traffic.**

   ARP Poisoning attack can lead to the modification, interception, and redirection of network traffic, allowing attackers to eavesdrop on sensitive information and alter the data. It can also cause DoS attacks, session hijacking, and network instability.

8. **How did you analyze the packet payloads for sensitive information?**

   The packet payloads can be analyzed by capturing the network traffic, examining packet headers and payloads, and looking for patterns or content that may indicate sensitive data such as login credentials or financial information.

9. **Name two prevention measures against ARP Poisoning.**

   Following are the two main prevention measures against ARP Poisoning:

   - **ARP Spoofing Detection Software**: Deploying specialized tools that monitor network traffic and detect ARP spoofing attacks. For eg: Snort, XArp, ARPMiner.
   - **ARP Spoofing Prevention Techniques**: Implementing measures such as static ARP table entries, ARP spoofing detection and prevention protocols (ARPWatch, ARP Guard, DAI), network segmentation, and network access control to prevent ARP spoofing incidents.

5 SECOND ATTACK: MAN-IN-THE-MIDDLE

5.1 OBJECTIVES OF MAN-IN-THE-MIDDLE ATTACK

The objectives of this project are as follows:

- To understand the methodology of a Man-in-the-Middle (MITM) attack using Ettercap
- To assess the potential impact of a MITM attack on network security
- To examine the interception of network traffic during a MITM attack

5.2 ABOUT ETTERCAP

Ettercap is basically a free and open-source network security tool for man-in-the-middle attacks. It is used for computer network protocol analysis and security auditing. It intercepts traffic on a network segment, captures passwords, and conducts active eavesdropping as well. It also features sniffing of live connections, content filtering on the fly, and many other interesting tricks.

5.3 PROTECTION

Several tools are available to protect yourself from Ettercap:

- **XArp** – It is a graphical utility that can detect attempts to spoof MAC addresses using the ARP protocol and counter it. It can work in Windows and in Linux
- **Snort** – It is a well-known system to counter intrusions. It can detect attacks on the ARP protocol
- **ArpON** – It is a small service that monitors the ARP table and protects it from spoofing MAC addresses.

5.4 FUNCTIONALITIES

It works by putting the network interface into promiscuous mode and by ARP poisoning the target machines. Ettercap acts as a MITM and unleashes various attacks on the victims. It also includes plugin support so that its features can be extended by adding new plugins.

5.5 FEACUTRE OF ETTERCAP

Following are the primary features of Ettercap:

- Ettercap supports active and passive dissection of many protocols
- It provides features for network and host analysis
- It can help to inject characters into a server or to a client while maintaining a live connection
- It can sniff the username, password, and even the data of an SSH1 connection
- It can also sniff the HTTP SSL secured data, even when the connection is made through a proxy
- It can determine the OS of the victim host, its network adapter, and can also hijack DNS requests

5.6 MODES OF OPERATION

Ettercap offers four modes of operation:

- **IP-Based**: Packets are filtered based on IP source and destination
- **MAC-Based**: Packets are filtered based on MAC address, useful for sniffing connections through a gateway
- **ARP-Based**: It uses ARP poisoning to sniff on a switched LAN between two hosts (full-duplex)
- **PublicARP**-Based: It uses ARP poisoning to sniff on a switched LAN from a victim host to all other hosts (half-duplex).

5.7 SUPPORTED DISTRIBUTIONS

The following distributions have been tested in both 32 and 64 bit flavors where possible:

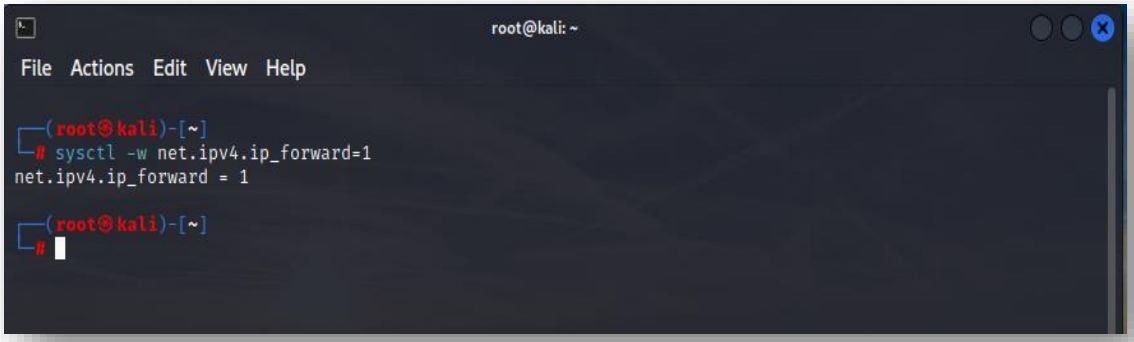
- Debian / Ubuntu (Includes derivatives such as Kali, BackTrack, Mint, etc)
- Fedora
- Gentoo
- Pentoo
- Mac OSX (Snow Leopard & Lion)
- FreeBSD
- OpenBSD
- NetBSD

5.8 UNSUPPORTED DISTRIBUTIONS

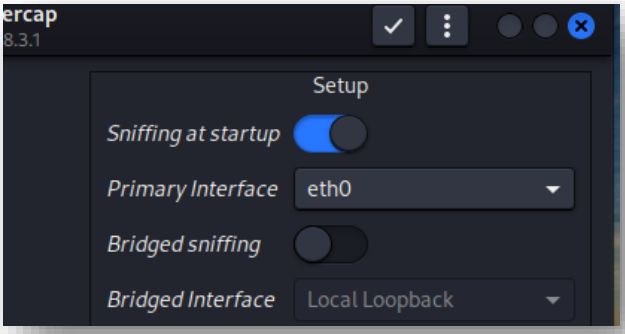
Installation may work on the following distributions, but they are not supported. Additional settings may be required for compilation and/or use:

- OpenSuSe
- Solaris
- Windows Vista
- Windows 7
- Windows 8

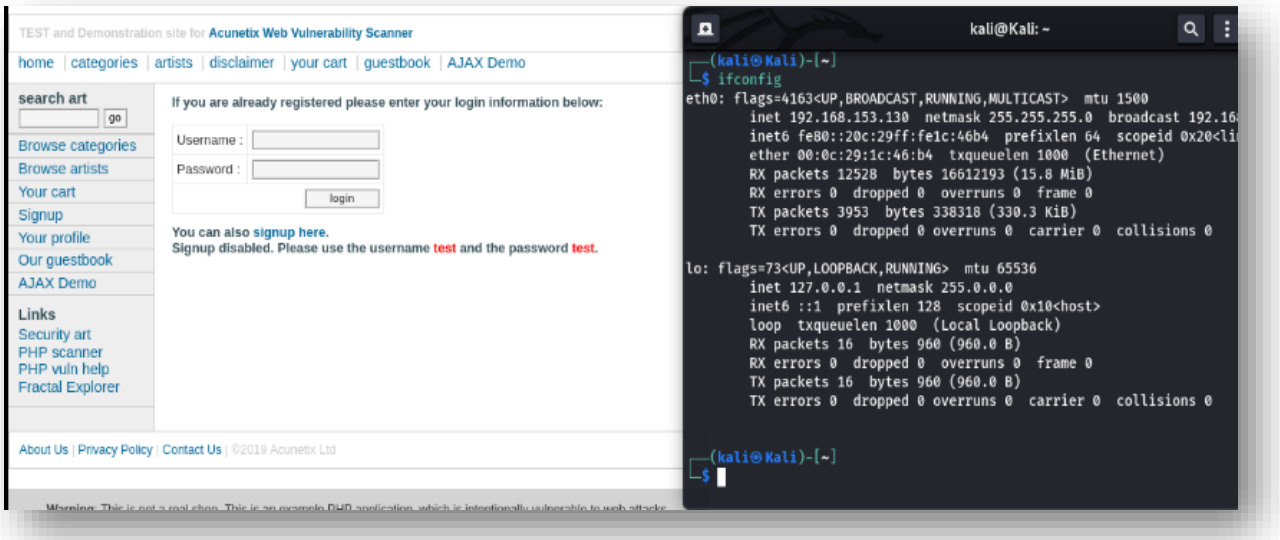
6 EVIDENCE / RELEVANT FINDINGS/ Analysis/ Exhibit of Man-In-The-Middle



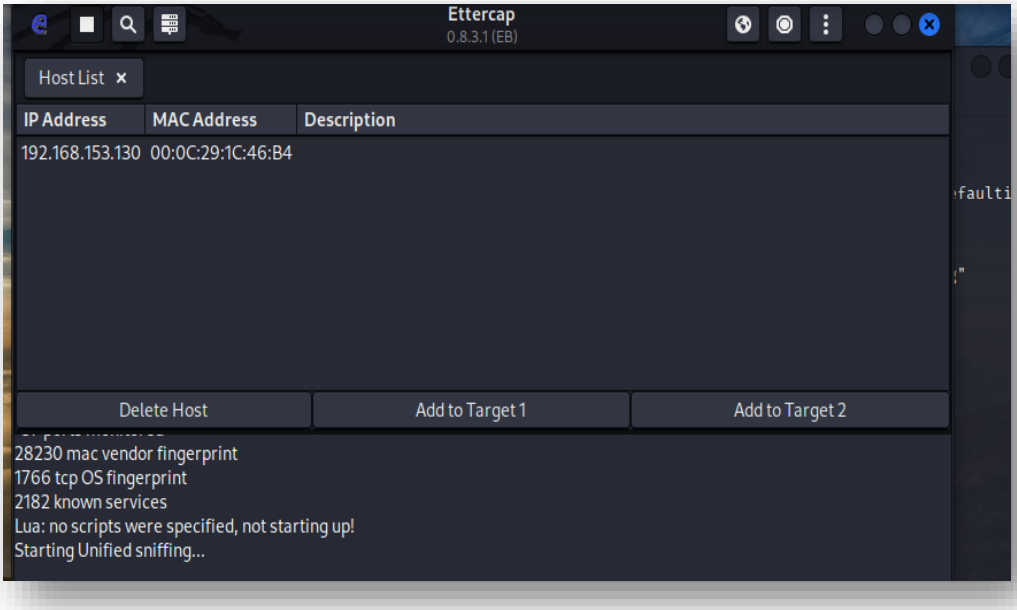
Firstly, we've entered the "**sysctl -w net.ipv4.ip_forward=1**" command in Kali. This command allows the flow of packets through your computer, as you're acting b/w the router and the client.



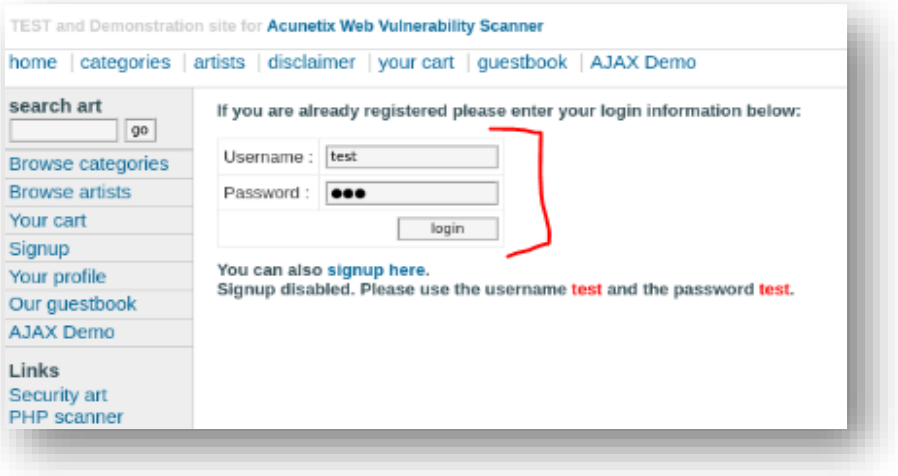
After that, we've opened Ettercap in my host OS (Kali). Note that Ettercap comes pre-installed in Kali.



After that, we opened a "**test login page**" in other Kali device which will be acting as the target for the purpose of this demonstration. The IP of that machine is **192.168.153.130**.

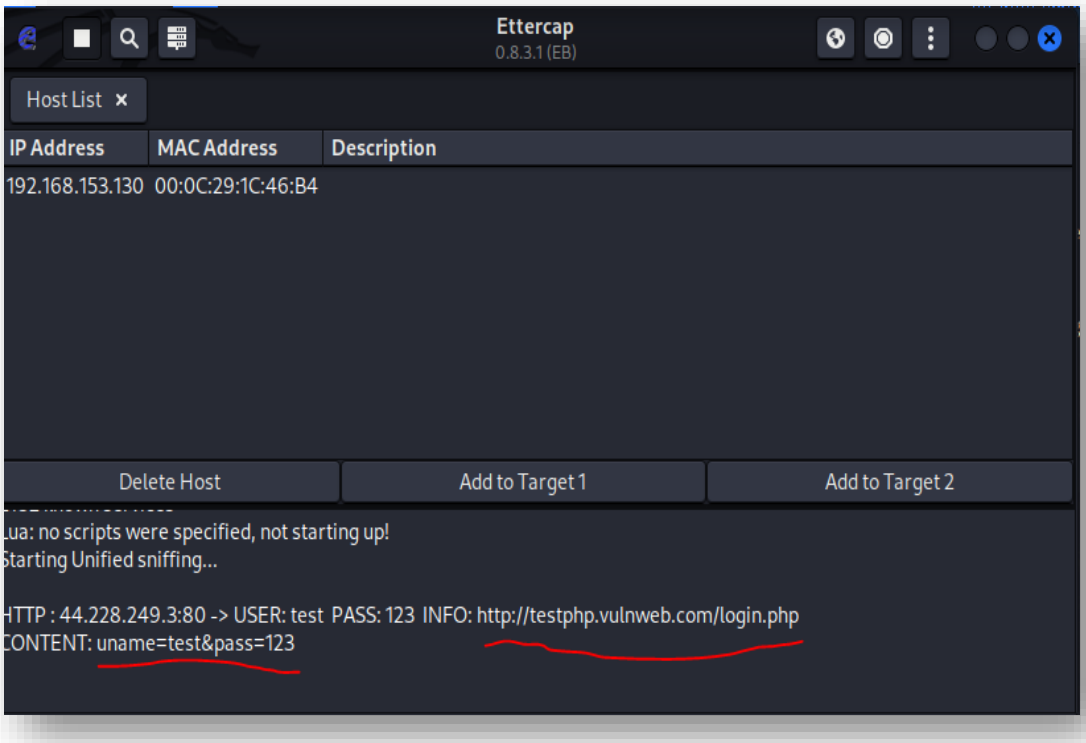


Here, you can see that Ettercap (which was running on the host device) has detected the victim machine.

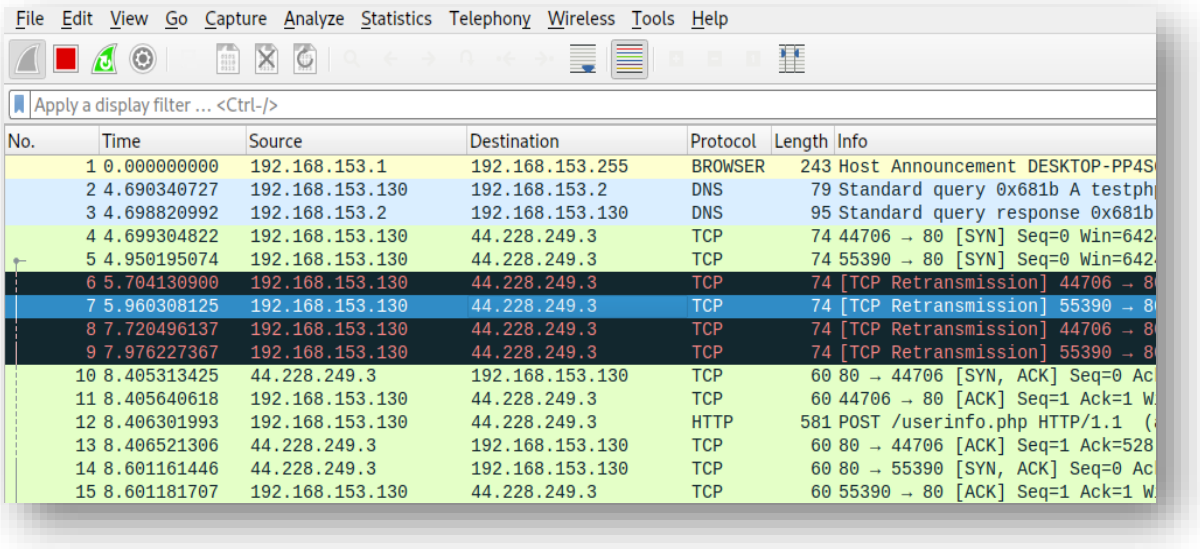


Now, we will switch back to the victim machine and will enter random details in the username and password fields. This is what we've entered:

- Username – test
- Password – 123



In the host machine, you can see that Ettercap has successfully sniffed the details which the victim entered. Thus, MITM attack is successfully performed.



Where you'll be running ettercap the Wireshark was snapshot, it was running in the background throughout the attack.

Here basically, the highlighted packet shows the IP of the victim communicating with the target site.

When you finish, you'll go to Wireshark and see that it captures packets. This means that victim’s traffic is passing through us (the host).

7 CONCLUSION

In conclusion, network forensics is a critical process that involves the collection, analysis, and interpretation of network data to identify and investigate security incidents. It plays an important role in determining the source of the attack, determining the extent of the damage caused, and developing strategies to prevent future attacks. Network forensics investigation requires specialized skills and tools to effectively analyze network traffic and identify potential threats. It is essential for organizations to have a strong network forensics capability to protect their assets and maintain business continuity.