**NETWORK ATTACK PROJECT**

**Experiment Design and Planning**

For this experiment, we’ll use **Bettercap** and **Wireshark** to demonstrate ARP spoofing attacks, monitoring network traffic between devices. The goal is to analyze vulnerabilities associated with data transmission and internet activity, and understand how attackers intercept and modify data using common network attacks, and how we can detect such attacks so we can prevent it.

**Key Objectives:**

- Conduct ARP spoofing to intercept/sniff traffic with **bettercap**.

- Analyze the intercepted traffic with to identify sensitive information that can be captured.

- Analyze the intercepted packets with **wireshark** so we can identify the ongoing ARP spoofing attack.

**Network Setup:**

* Ensure Bettercap is running on a system with network access permissions and that all devices on the network are connected to the same subnet.
* Ensure that Wireshark is running on the right interface and that is capturing traffic.

**Tool Selection and Utilization :**

* **Bettercap** is a versatile network monitoring and attack tool, suitable for performing various MitM attacks and spoofing operations. It can perform both ARP spoofing attacks and DNS spoofing attacks. We will be focusing on the ARP spoofing attack in this project. Below, I’ll guide you through setting up Bettercap for these attacks.
* **Wireshark** is a renowned network protocol analyzer that provides in-depth visibility into network traffic. Leveraging its advanced capabilities, Wireshark can effectively detect ARP spoofing attacks. By scrutinizing ARP traffic, Wireshark identifies conflicting ARP entries, unusual traffic patterns, and mismatched destination MAC addresses - all indicators of an ongoing ARP spoofing attempt. Wireshark's comprehensive analysis features make it a crucial tool for monitoring and mitigating these types of man-in-the-middle attacks.

**Attack Identification and Analysis :**

**Part A: ARP Spoofing attack**

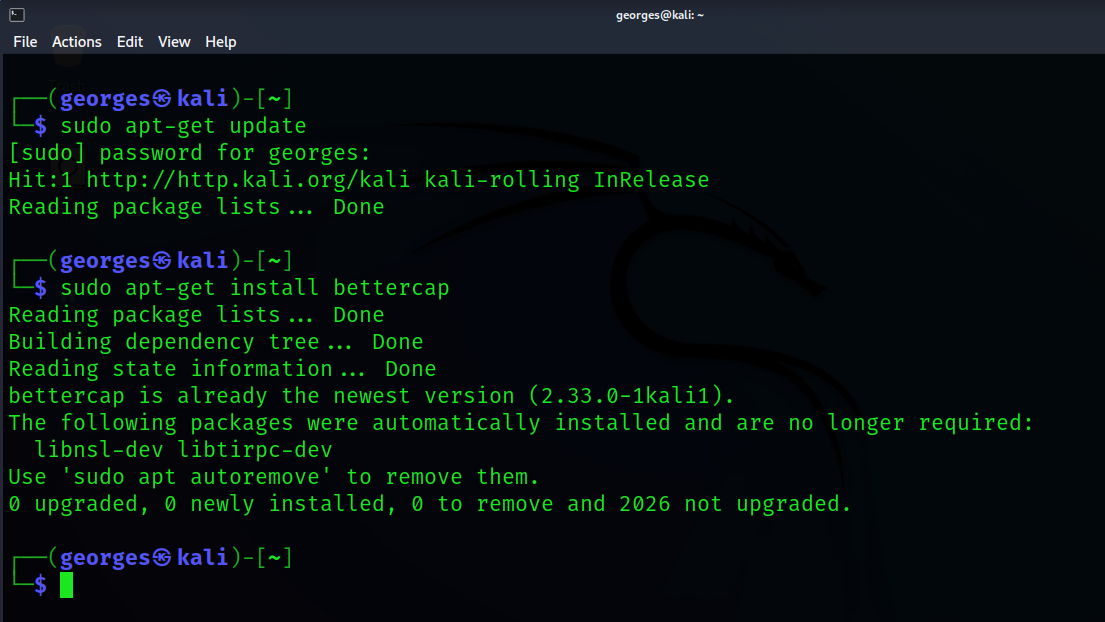
**1. Description** : ARP spoofing allows an attacker to intercept data by associating their MAC address with the IP address of a target device.

**2. Setup Steps** :

- Install Bettercap by running:

**sudo apt-get update**

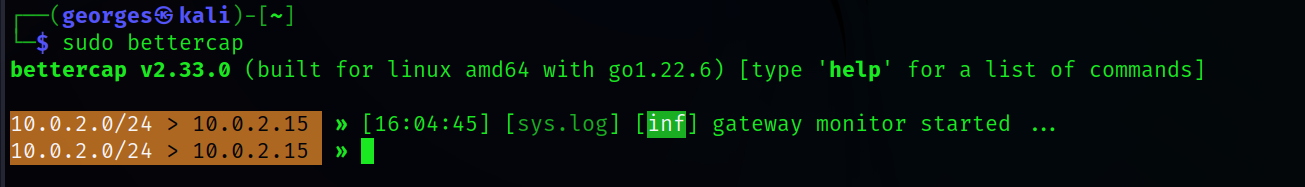
**sudo apt-get install bettercap**



**3. Executing ARP Spoofing in Bettercap** :

- Launch Bettercap with root privileges:

**sudo bettercap**

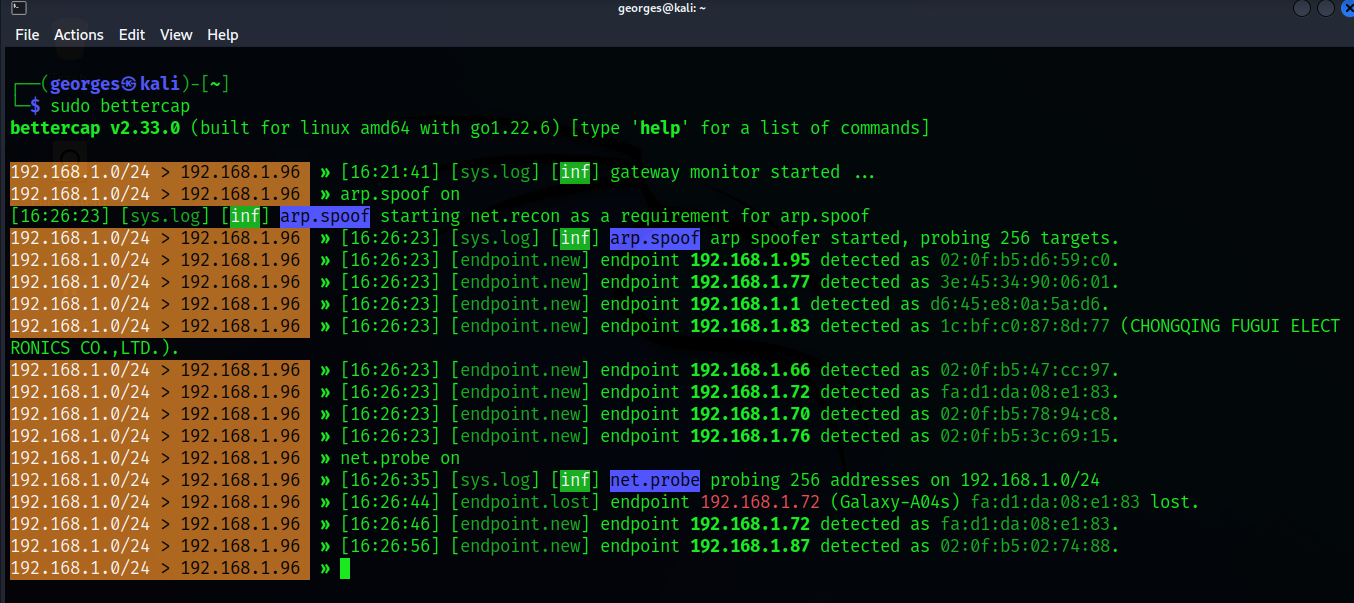
****

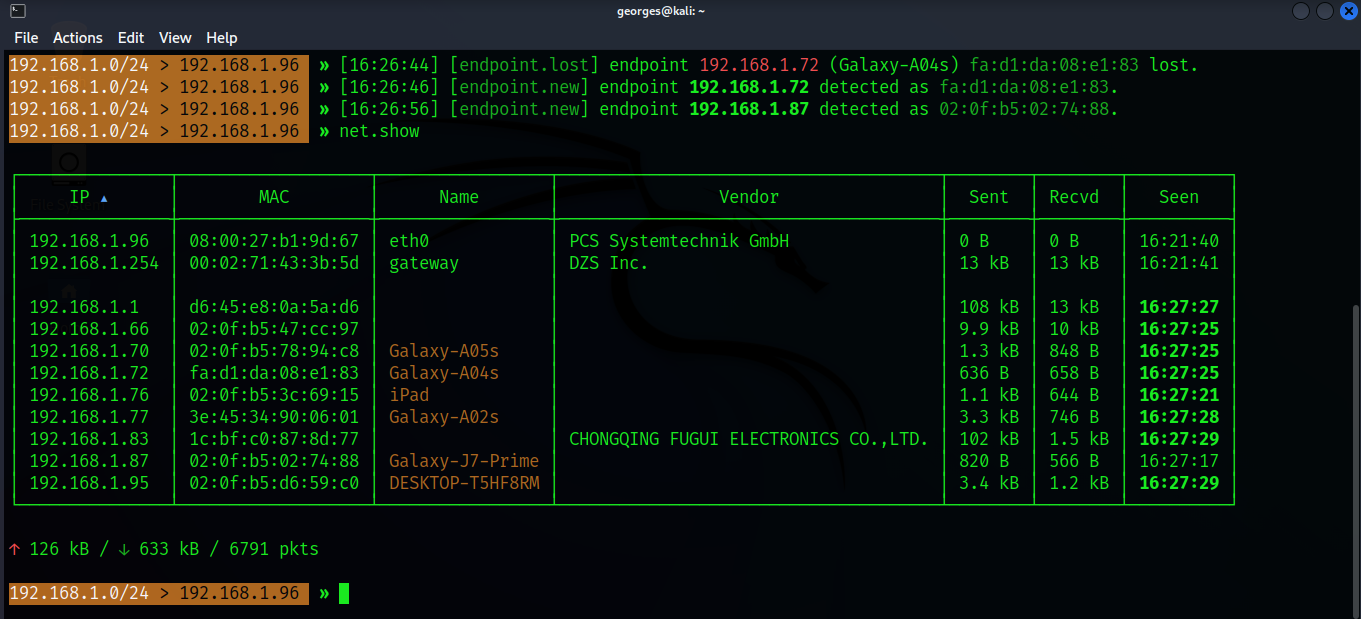
- Enable ARP spoofing and specify the target IP (replace `TARGET\_IP` with the IP of the target device and `GATEWAY\_IP` with the router’s IP):

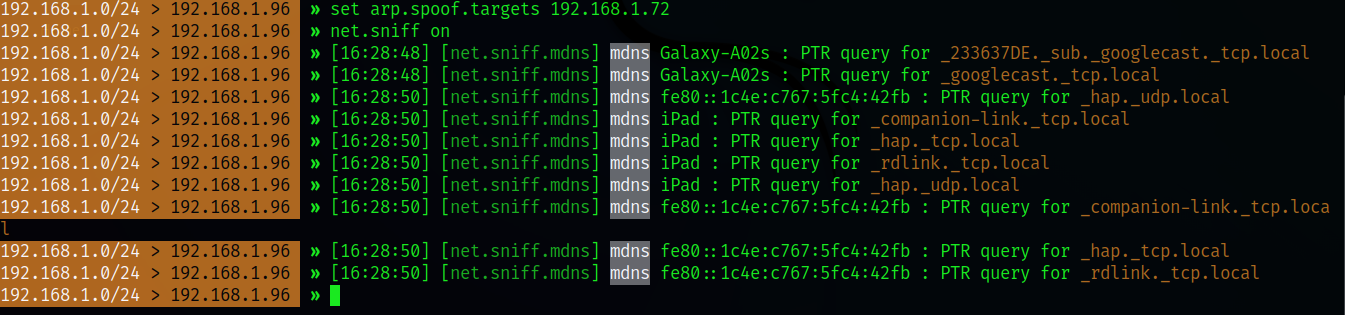
**arp.spoof on**

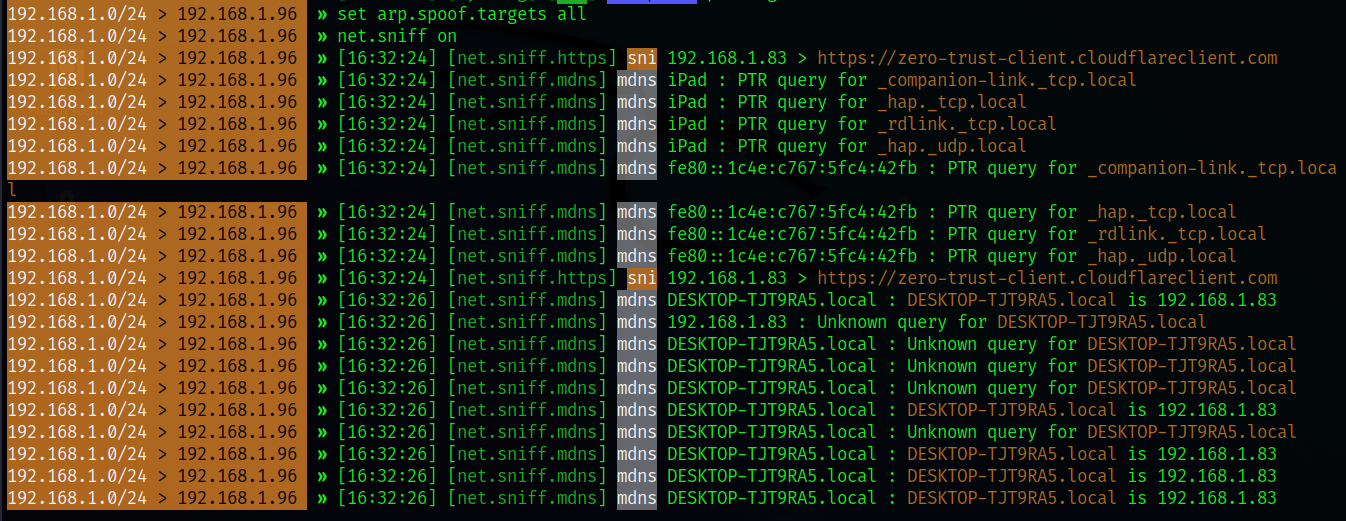
- Enable the `net.probe` module to scan for devices on the network:

**net.probe on**

****

** net.show (**to view all the connected devices to the network in a table showing their ipv4/ipv6 addresses, their mac addresses, and their Hostname/Device name**)**

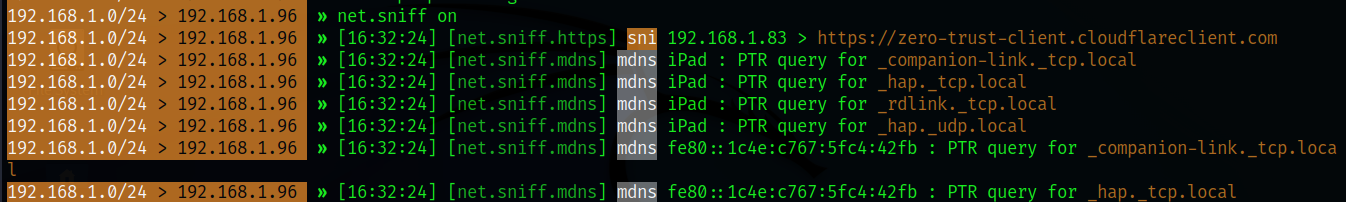
 **set arp.spoof.targets TARGET\_IP** (for a specific target)

**set arp.spoof.targets TARGET\_IP** (for a all the devices on the netowrk)

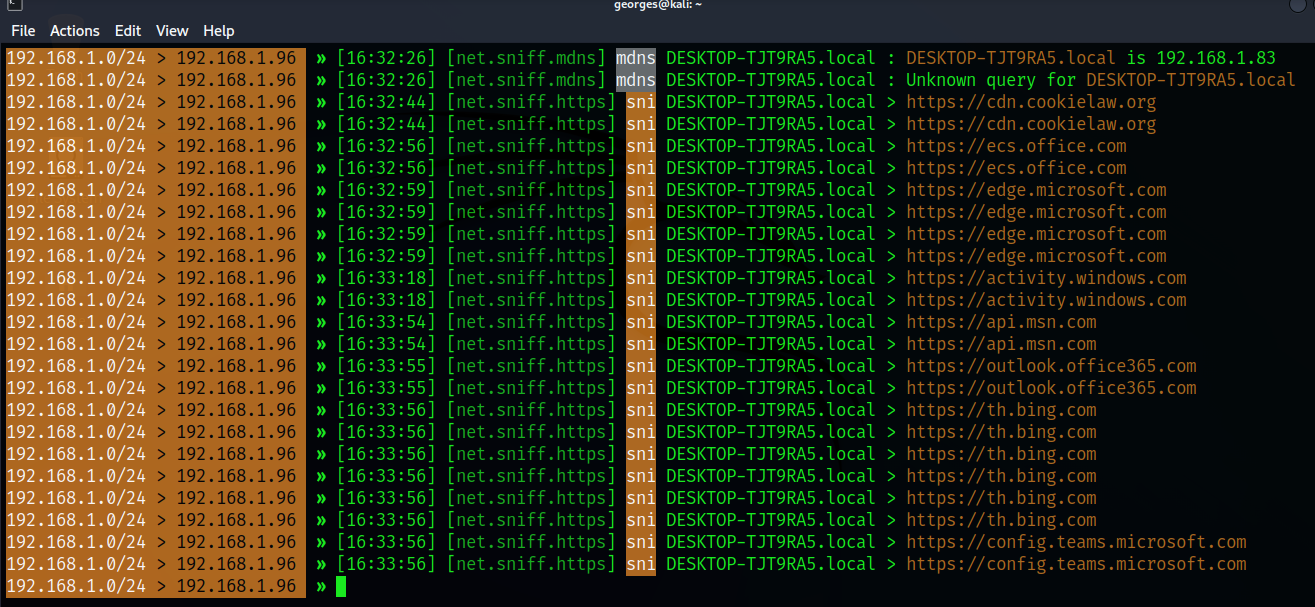
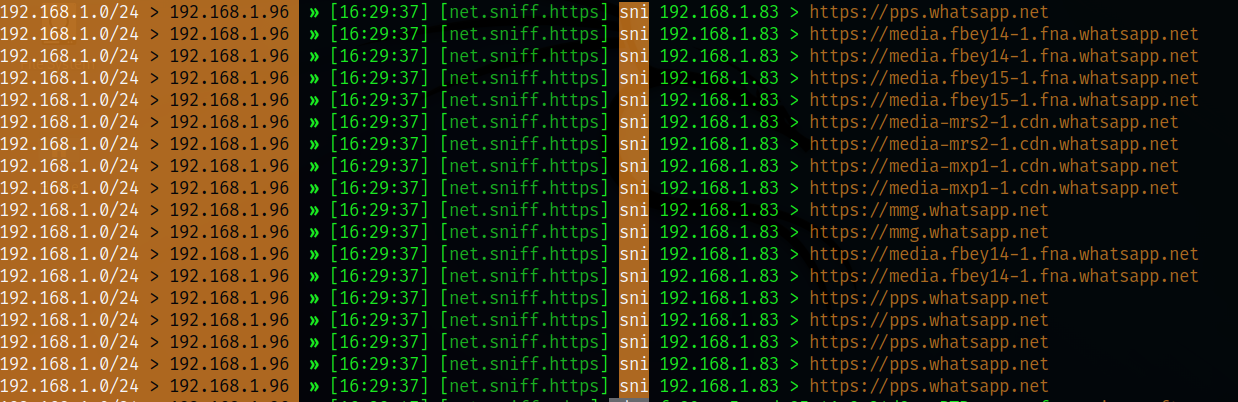
**4. Traffic Interception and Analysis :**

- Enable packet capture to log intercepted traffic:

**net.sniff on**

****

**5. Results :**



As you can see, we can intercept the user’s internet activity in real time. For example, we can see that the device **192.168.1.83** is using **Whatsapp** , **pps.whatsapp.net** tells us that he is sending/receiving text messages, **media-xxxxxxxxx….net** links shows us that the user is sending/receiving media messages such as voice notes, images, videos…

**- Observation** : We can observe sensitive data visible in the intercepted sniffed information, such as real-time user activity and behavior through links of the sites or apps he is using. Note that we can get unencrypted login credentials during this attack too.

**- Analysis** : ARP spoofing redirects traffic to Bettercap, allowing you to view or modify it.

**- Issues that can be encountered and troubleshooting**: You may encounter a problem characterized by the inability of the tool of probing devices on the network and sniffing the data. To solve it, make sure to change your machine’s network settings to Bridged Adapter instead of NAT or Internal Network or others…

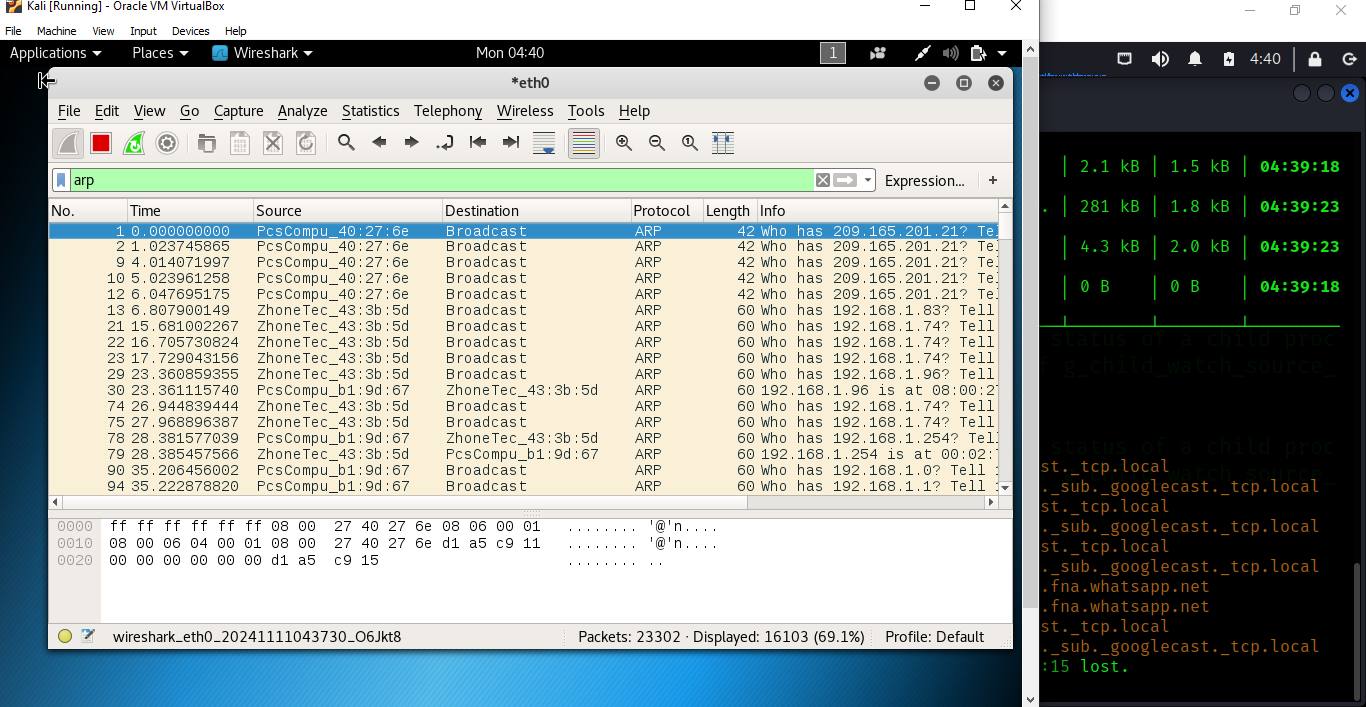
**Part B: Detecting the Attack with Wireshark during the attack’s progress**

1. **Using Wireshark**:

* **Start Wireshark**: Launch Wireshark on your machine.
* **Select Network Interface**: Choose the appropriate interface to monitor.
* **Capture Traffic**: Start capturing packets.
* **Filter for ARP**: Use the display filter : arp

1. **Analyze ARP Packets**:

* Look for multiple ARP responses for the same IP address.
* Check for different MAC addresses claiming the same IP.

1. **Results :**

Based on the Wireshark capture shown in the image, there are several indicators that suggest an ARP spoofing attack is taking place:

1. **Multiple ARP Responses for the Same IP Address:** The capture shows several ARP responses with different MAC addresses claiming ownership of the same IP addresses, such as 209.165.201.21 and 192.168.1.x addresses.
2. **Unusual ARP Traffic Volume:** There is a high volume of ARP requests and responses being captured, which could indicate an attacker is aggressively sending ARP messages to poison the ARP cache of devices on the network.
3. **Suspicious MAC Addresses:** The MAC addresses involved in the ARP traffic, such as PcsCompu\_b1:9d:67, appear to be unusual and may belong to the attacker's machine(s) impersonating legitimate hosts.

- **Observation** : Multiple ARP responses with different MAC addresses claiming the same IP address indicate a possible ARP spoofing attack.

- **Analysis** : This suggests the attacker is attempting to poison the ARP cache, which can lead to traffic interception and disruption of normal network communication.

- **Issues that can be encountered and troubleshooting:** You may be unable to run **Wireshark** properly if you chose the wrong interface. Make sure to choose the right interface by checking it through commands like **ifconfig** or **ip a**.

**Experiment Execution and Documentation**

- **Bettercap Commands**: The commands are recorded step-by-step above in **Part A**, with explanations of each command’s purpose.

- **Wireshark steps:** The steps to do are recorded step-by-step above in **Part B**, with explanations of each step and filter purpose.

- **Results and Observations** : Screenshots and text logs captured during “ARP spoofing attacks” in **Part A**, and the “Wireshark packets interception” in **Part B**, are inserted above to demonstrate the experiment’s effectiveness.

- **Troubleshooting** : Any issues that can be encountered mostly, are documented at the end of both **Part A** and **Part B**.

**Critical Thinking and Problem Solving**

During the experiment, you might face unexpected network issues. Here are some examples:

**- Challenge** 1: Network devices not appearing in “net.probe” results.

**- Solution** 1: Restart Bettercap or the network interface and re-enable “net.probe”.

**- Challenge** 2: ARP spoofing fails to intercept traffic from certain devices

**- Solution** 2: Ensure that the target devices are on the same network segment and do not have static ARP entries that could bypass the attack.

**Presentation and Communication**

- **Structure** : My report is divided into sections, and used screenshots and command snippets for clarity.

- **Methodologies** : In my report , I describe how each “Bettercap” feature works and its role in network attacks, and did the same for “Wireshark” and its role in detecting the attack.

- **Insights and Mitigations** : As recommendations, I suggest using HTTPS for sensitive sites and implementing network segmentation to prevent unauthorized ARP and DNS spoofing.

Lab Security

Presented by Georges Machhour - 202110271