Cyber Security - Security Tools - LABB1

Yazan Al Khalili
Yazanalkhalili77@gmail.com

Ali M S Aljaiab
Ali.jayab10911@gmail.com

Karlstad University

November 2024

1 Introduction

This lab aims at improving our knowledge in security tools, with a focus on using well-known tools such as Nmap, Wireshark, and Ettercap. Each of these tools serves a specific purpose in the field of network security.

- **Nmap**: A tool for scanning networks to find computers, open ports, and services, and to identify vulnerabilities.
- **Wireshark**: A packet analyzer that captures and examines network traffic in real-time to understand device communication and detect issues.
- **Ettercap**: A tool for network security testing, specializing in man-in-the-middle (MITM) attacks to intercept and analyze network data.

2 Lab Environment

To be able to complete the tasks for this lab, the following components are needed:

- **Vulnerable Server**: The Metasploitable server, located in the LAN 1 (Inside) network, with the IP address **192.168.2.10**. This server is intentionally designed with security flaws for testing.
- **Attacker machine**: The Kali Linux machine, located in the LAN 2 (Outside) network, with the IP address **192.168.1.11**. This machine is used to perform attacks on the vulnerable server.
- **Client Machine**: The Windows XP machine, located in the LAN 2 (Outside) network, with the IP address **192.168.1.12**. It simulates a typical user's system interacting with the vulnerable server.
- **Router**: The gateway connects LAN 1 (Inside) and LAN 2 (Outside) networks, allowing communication between the attacker, client and server. It uses IP addresses **192.168.1.1** (outside) and **192.168.2.1** (inside).

3 Experiment

This section outlines the step-by-step process used to conduct the experiment and complete the three tasks. Detailed explanations are provided for each step, along with accompanying images to give a clear visual representation of the actions taken and the outcomes observed.

3.1 Task 1 - Nmap

3.1.1 Step 1

A commonly used command in Nmap is to display host interfaces and routing details. For this lab, we will execute this command on the Kali Linux attacker machine. The command is: `nmap --iflist`.

Figure 1 below shows the Interface section and the Router section.

In the Interface section, two important details must be noted:

- **Loopback Interface (lo):**
  - Internal communication (system talks to itself).
  - IP: `127.0.0.1` (localhost), Subnet: `/8`.
  - Type: Loopback.
- **Ethernet Interface (eth0):**
  - Connects Kali to the lab network.
  - IP: `192.168.1.11`, Subnet: `/24`.
  - MAC: `00:0C:29:F4:41:2D`.
  - Type: Ethernet.

In the router section:

- **Road to Local Devices (192.168.1.0/24):**
  - Allows communication with devices in the same network, like the Windows XP client and the router.
  - Any IP starting with `192.168.1` can be accessed directly.
- **Road to the Outside (Default Gateway: 192.168.1.1):**
  - Acts as the "exit door" to other networks.
  - Enables access to the Metasploitable server (`192.168.2.10`) and external devices.

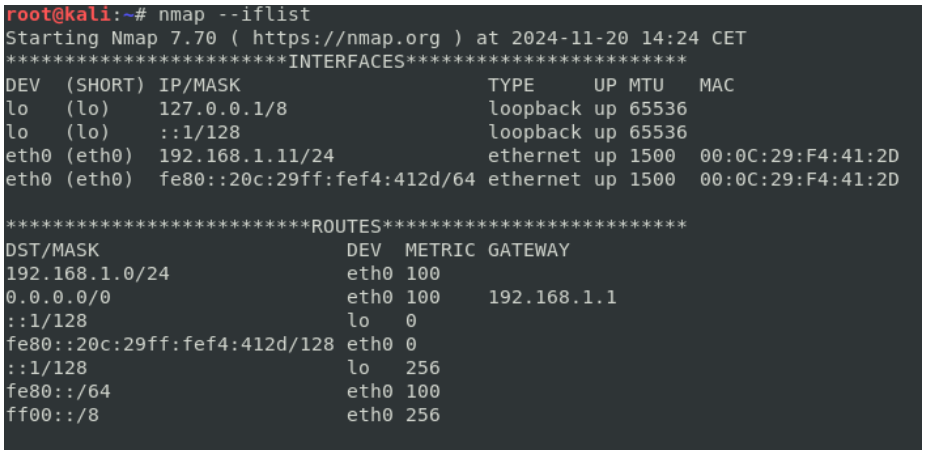

Figure 1: Interfaces and Routes

3.1.2 Step 2

To scan the network and identify all four devices that are up and running, the command `nmap -sn` is used. This command is called an Nmap ping scan.

Figure 2 below shows two subnets being scanned. The first scan targets the subnet `192.168.1.0/24`, which covers all devices in the `192.168.1.x` range, with the goal of discovering live hosts.

The results indicate that 3 hosts are active in the **LAN 2 (outside)** network. Their IP and MAC addresses are shown, and the devices identified are as follows:

- The Kali Linux attacker machine with the IP address `192.168.1.11`.
- The Windows XP client machine with the IP address `192.168.1.12`.
- The gateway for the local network with the IP address `192.168.1.1`.

The second scan targets the subnet `192.168.2.0/24`, which coves all devices in the `192.168.2.x` range. The results indicate that 2 hosts are active in the LAN1 (Inside) network. Their IP and MAC addresses are shown, and the devices identified are as follows:

- The vulnerable server which is called the Metasploitable server, the main target with the IP address `192.168.2.10`.
- The gateway for the local network with the IP address `192.168.2.1`.

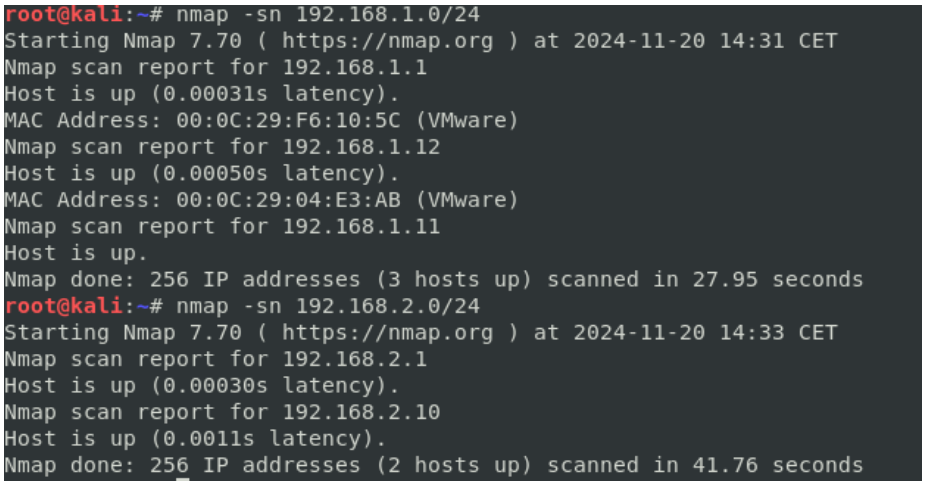

Figure 2: Nmap ping scans

3.1.3 Step 3

To gather information about the Metasploitable server (`192.168.2.10`), several Nmap scans were performed to identify its operating system, open ports, and running services. Figures 3, 4, 5 and 6 shows the results of these scans.

- **Operating System Detection (-O):**
  - The (`nmap -O`) command identified the server as running a Linux-based OS.
- **TCP SYN Scan (-sS):**
  - Identifies open ports like 21 (FTP), 22 (SSH), 80 (HTTP), and 3306 (MySQL), confirming multiple services are running. It is very effective for discovering open ports.
- **TCP Connect Scan (-sT):**
  - Confirms the same open ports as the SYN scan, using a full connection method. It is useful for verification.
- **TCP ACK Scan (-sA):**
  - Reveals that all scanned ports are unfiltered, indicating no firewall protection, meaning the server is fully exposed to network traffic.
- **Service Detection: (-sV):**
  - The nmap -sV command identifies services on open ports and their version numbers, useful for finding outdated or vulnerable software.

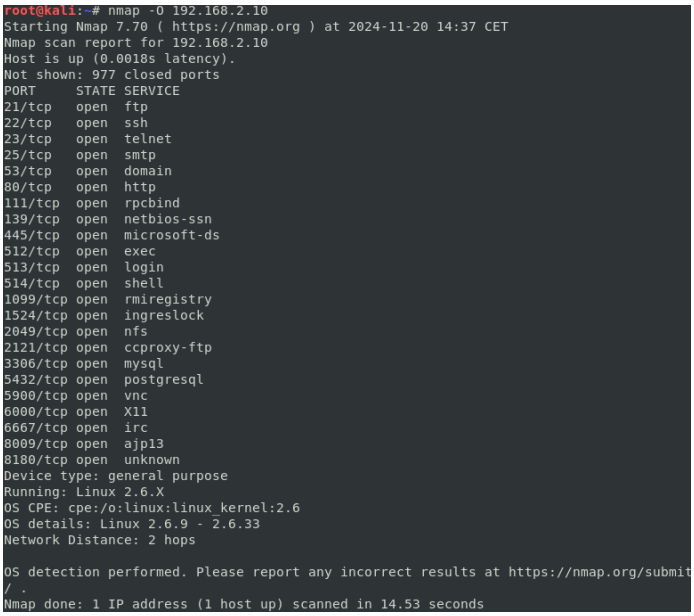

Figure 3: Operating System Detection

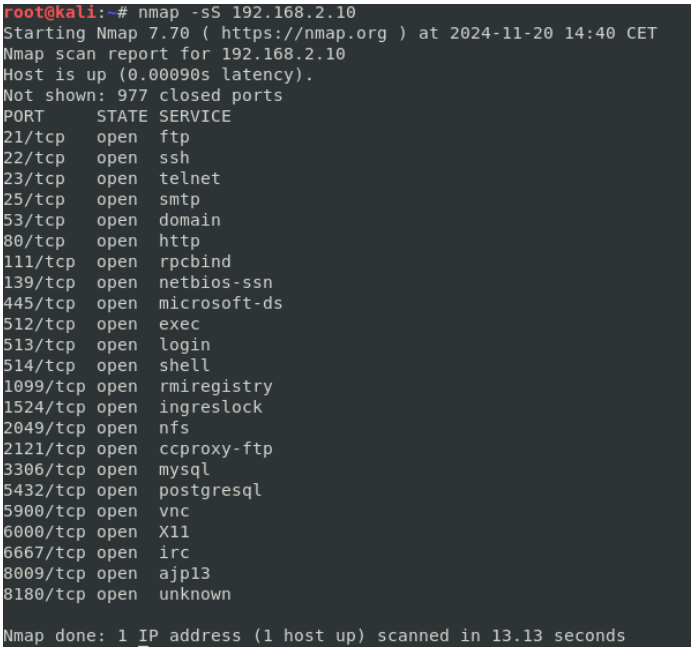

Figure 4: TCP SYN Scan

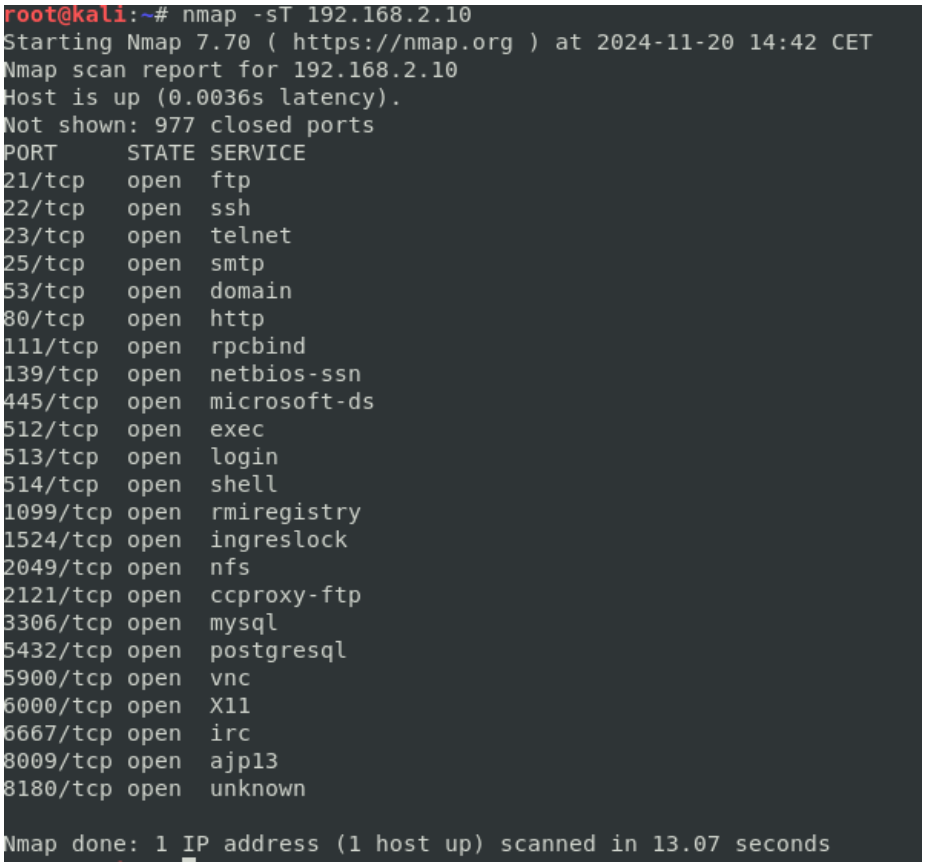

Figure 5: TCP Connect Scan

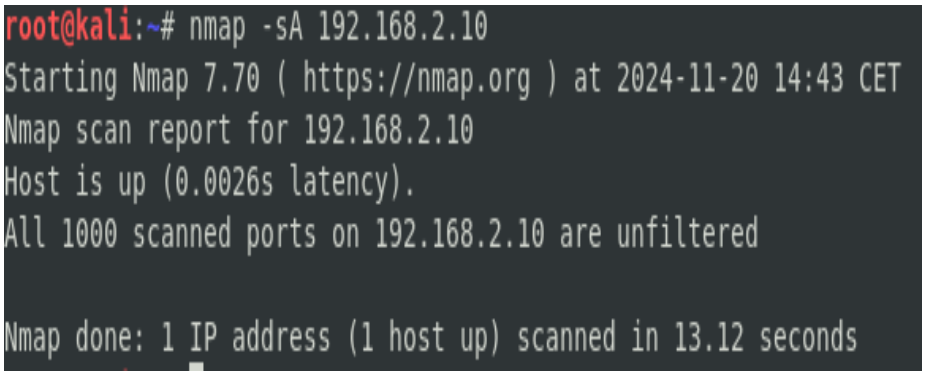

Figure 6: TCP ACK Scan

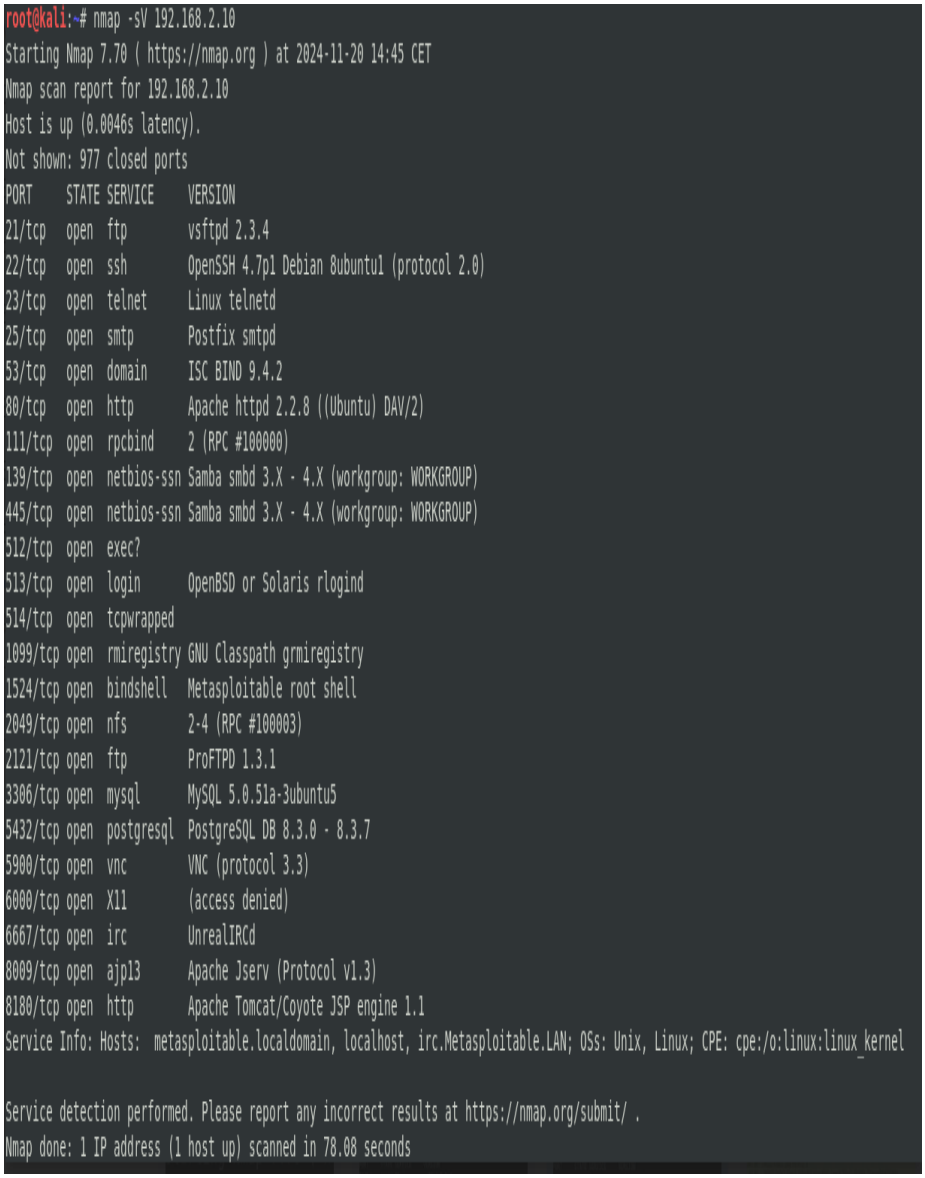

Figure 7: Service and Version Detection

4 Task 2 - Wireshark

4.1 Capturing HTTP Traffic

The main goal of this task is to capture and analyze the network traffic between the **Metasploitable server** and the **client machine** using Wireshark on the **Kali Linux VM**. Filtering the traffic is essential to narrow down the captured data, saving time and focusing on relevant traffic. A **capture filter** is used to limit the capture to **HTTP** traffic only.

- First, check the **eth0** interface in the interface section of Wireshark. The **eth0** interface is the main network interface on the **Kali VM**.
- Second, turn off **Promiscuous Mode**. Promiscuous mode allows Wireshark to capture all traffic on the network, even traffic not addressed to the Kali machine. It is important to turn this off to avoid unnecessary traffic capture.
- Third, set the **capture filter** to `tcp port 80` to limit the capture to only **HTTP** traffic, as we are interested in sniffing **HTTP** packets.

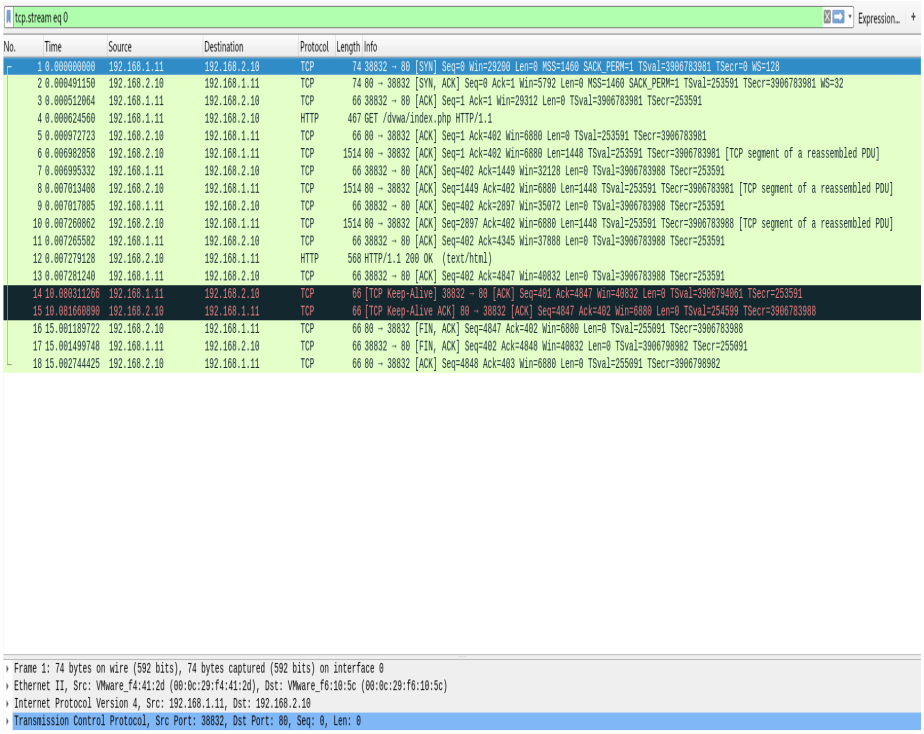

Figure 8: HTTP Traffic

4.2 Generating HTTP Traffic

To further test Wireshark and its capabilities, we generated some HTTP traffic by simply opening a web browser and connecting to **http://192.168.2.10/dvwa/index.php**. After connecting, we logged in using the credentials **admin/password**.

After successfully logging in, we wanted to capture and examine the specific HTTP traffic generated by this action in Wireshark. The figure below shows the captured HTTP request and response in Wireshark. The HTTP request is highlighted in **red**, and the HTTP response is highlighted in **blue**.

- **Request:** The `POST` method is used to send data to the server, in this case, for logging into the DVWA (`/dvwa/login.php`).
  - **Form Data:**
    - `username=admin`
    - `password=password`
    - `Login=Login`
- **HTTP Response:** The response shows a `302 Found` status, which is a redirection (likely to the next page after a successful login). The `Location` header specifies that the user is being redirected to `index.php`.
- **Conclusion:** This packet successfully captured the log-in attempt with the username (`admin`) and password (`password`) transmitted in plain text over `HTTP`. These credentials were part of the `POST` request sent to the Metasploitable server.

```
[1 bytes missing in capture file].POST /dvwa/login.php HTTP/1.1
Host: 192.168.2.10
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.2.10/dvwa/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 44
Cookie: security=high; PHPSESSID=7671c723f25261d5a9e8301636a7aecc
DNT: 1
Connection: keep-alive
Upgrade-Insecure-Requests: 1

username=admin&password=password&Login=LoginHTTP/1.1 302 Found
Date: Wed, 20 Nov 2024 13:17:09 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Location: index.php
Content-Length: 0
Keep-Alive: timeout=15, max=98
Connection: Keep-Alive
Content-Type: text/html
```

Figure 9: HTTP Traffic Captured From Browser

5 Task 3 - Ettercap

The goal of this task is to intercept network traffic between the client machine (running Windows XP) and the Metasploitable server, to capture the username and password sent over the network. One way to achieve this is by poisoning the ARP cache (Address Resolution Protocol) of the machines in the network, and then sniffing the traffic for credentials using Wireshark.

What is ARP poisoning?
ARP poisoning is a network attack where an attacker sends fake ARP messages to associate their device's MAC address with the IP address of another device on the network, typically the router or a target machine. This allows the attacker to intercept, monitor, or manipulate network traffic between devices.

Down below is a step by step process on how to complete this task successfully.

5.1 Step-by-Step Process:

5.1.1 Enable Port Forwarding on Kali Linux:

- This will allow Kali to forward packets it intercepts to the appropriate destination.
- To do this we changed `/proc/sys/net/ipv4/ip_forward` from 0 to 1 using:
  - `echo 1 > /proc/sys/net/ipv4/ip_forward`

5.1.2 Use Ettercap for ARP Poisoning:

- **Set Interface:** Select `eth0` for network connection. This is the network interface used by Kali Linux to connect to the network.
- **Scan for Hosts:** Scan the local network to find devices. This action scans the local network to identify all devices that are connected, including the ones we want which are the Metasploitable server and the Windows XP client.
- **Add Targets:** Set the router (`192.168.1.1`) and client (`192.168.1.12`) as targets. The router (IP: 192.168.1.1) is one of the targets for ARP poisoning. By poisoning its ARP cache, we redirect traffic from the client and server through Kali Linux. The Windows XP client (IP: 192.168.1.12) is the second target. It will also have its ARP cache poisoned to redirect its traffic to Kali Linux. Figure 10 shows the targets.
- **ARP Poisoning:** Now we choose **"ARP poisoning..."** from the **Man-in-the-Middle** (MITM) menu.

5.1.3 Capture Traffic in Wireshark:

- Now that Ettercap is prepared to be in use we start capturing on `eth0`, and we have to also ensure that no active filters are applied.

5.1.4 Generate HTTP Traffic:

- On the client machine, we use `telnet 192.168.2.10` with credentials. This action will generate network traffic between the client and the server

5.1.5 Examine Packets:

- In Wireshark, we can now follow the TCP stream to view the login credentials in plain text. Figure 11 shows the login credentials.

5.1.6 Stop ARP Poisoning:

- Now that we are done, we stop the ARP poisoning in Ettercap.

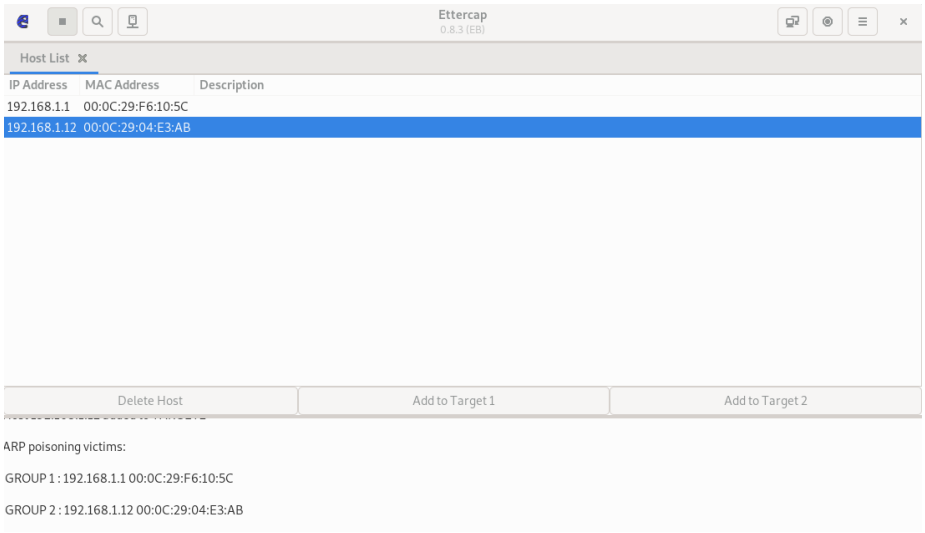

Figure 10: Targets

Figure 11: Credentials for Metasploitable machine

6 Conclusion

In this lab, we successfully demonstrated how to use network sniffing and ARP poisoning to intercept and capture sensitive data, such as login credentials, using tools like Wireshark and Ettercap. By poisoning the ARP cache of the

client and router, we were able to redirect traffic through the Kali Linux machine, allowing us to monitor and capture plain-text credentials transmitted over insecure protocols like Telnet and HTTP.

We also used Nmap for network scanning to discover the devices on the local network, identify open ports, and determine which services were running on the target systems. This information was crucial in understanding the network structure and identifying potential vulnerabilities. Nmap helped us find key services, such as HTTP and Telnet, that could be exploited to capture credentials.

Through this exercise, we gained insight into the risks of using unencrypted communication protocols, such as Telnet and HTTP, and the dangers of Man-in-the-Middle (MITM) attacks. It's important to secure network communications by using encryption (e.g., HTTPS, SSH) to prevent unauthorized access and protect sensitive data.

In conclusion, ARP poisoning is an effective technique for sniffing network traffic in a local network, which highlights the importance of network security practices like using secure protocols, implementing network segmentation, and using tools like firewalls and intrusion detection systems to protect against such attacks. Tools like Nmap also play a key role in finding vulnerabilities and preparing for attacks, emphasizing the need for regular network scans and security audits.