Task 1

Scan Your Local Network for Open Ports

Objective:- Learn to discover open ports on devices in your local network to understand network exposure. Tools: Nmap (free), Wireshark (optional).

📘 DEFINITION:-

Nmap (short for Network Mapper) is a **free and open-source network scanning tool** used to discover hosts, detect open ports, identify services, determine operating systems, and perform security audits on computer networks.

🎯 **What Makes Nmap Special?**

- Scan thousands of hosts in a few seconds.
- Determine what ports are open on a machine.
- Find out which services are running and their versions.
- Guess the operating system of a target.
- Detect firewalls, intrusion detection systems, and more.

1.Install Nmap from official website

✅ **Option 1: Install Using APT (Recommended for Kali Users)**

This is the easiest and safest method on Kali Linux.

1. **Open the terminal**.
2. Update package lists:

   Cmd{ **sudo apt update**}

3. Install Nmap:
   Cmd { sudo apt install nmap -y}

🧩 **Option 2: Manual Installation from the Official Nmap Website**

Use this method if you want the **latest Nmap release** from the developers.

1. Go to the official Nmap download page:
   🔗 https://nmap.org/download.html
2. Scroll to the **"Linux Source Code"** section.
3. Download the latest .tar.bz2 file:

   Cmd <<wget https://nmap.org/dist/nmap-7.94.tar.bz2 >>

4. Extract the downloaded file:

   Cmd { tar -xvjf nmap-7.94.tar.bz2

   cd nmap-7.94 }

5. Compile and install:
   { ./configure
   make
   sudo make install }

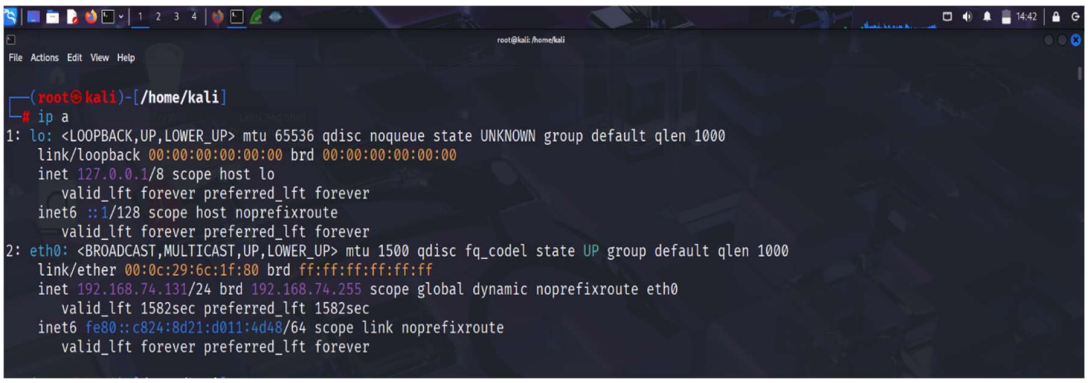
🔐 **Post-Installation Tip:**

Once Nmap is installed, it's important to ensure that:

- You run scans using sudo or as root for full functionality.
- You respect ethical boundaries — only scan networks you are authorized to.

2.Find your local IP range (e.g., 192.168.1.0/24)

To begin scanning the local network for open ports, it is crucial to first identify the IP address and subnet range of the host system. Using the ip a command in Kali Linux, we were able to retrieve detailed network interface information.

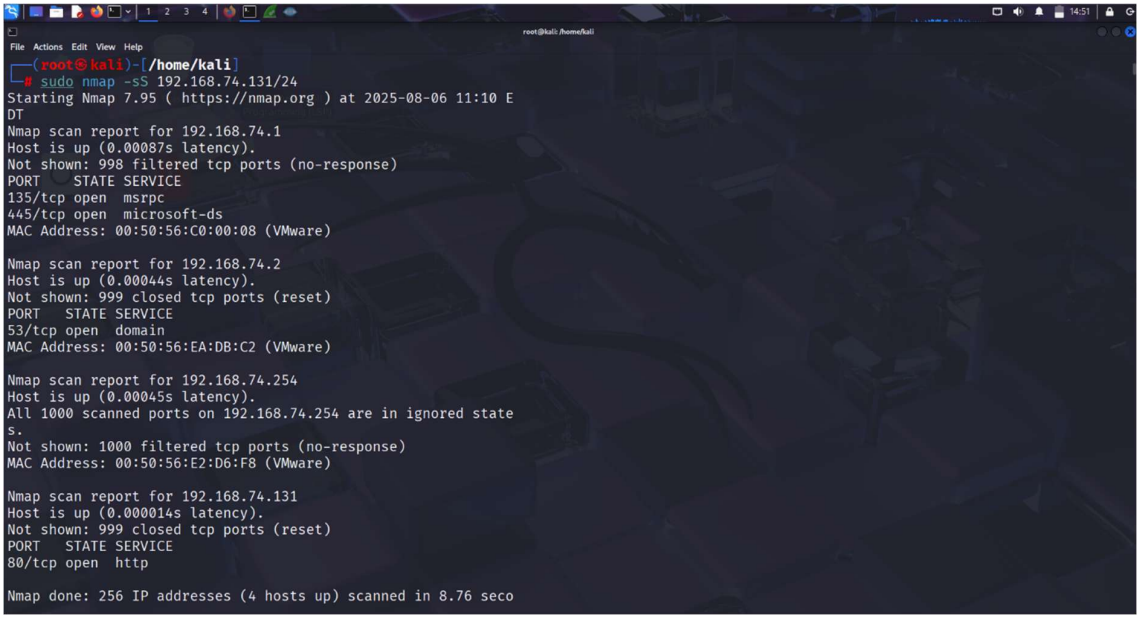


This indicates that the system is part of the 192.168.74.0/24 subnet, which includes IP addresses ranging from 192.168.74.1 to 192.168.74.254. This IP range will be used as the target scope for the Nmap scan.

🔑 Key network details from the output:

- Interface: eth0
- Local IP Address: 192.168.74.131
- Subnet Mask: /24 (255.255.255.0)
- Broadcast Address: 192.168.74.255

This configuration confirms that the system is connected to a private internal network, making it an ideal environment for performing a safe and controlled port scan.



🧠 **Analysis & Observations:**

- Port 135 & 445 are often associated with Windows services (MS-RPC and SMB). If left exposed, these can be vulnerable to exploits like EternalBlue, which was used in the infamous WannaCry ransomware attacks.
- Port 53 indicates a DNS service running on 192.168.74.2, which could be useful for internal name resolution but may also be abused if misconfigured.
- Port 80 on the local Kali machine confirms that a web service (possibly Apache or Nginx) is active. If not secured, this can be a vector for attacks such as XSS or SQL Injection.
- 192.168.74.254 responded to pings but had no open ports visible, suggesting either strict firewall rules or port filtering mechanisms in place.

🔐 **Conclusion:**

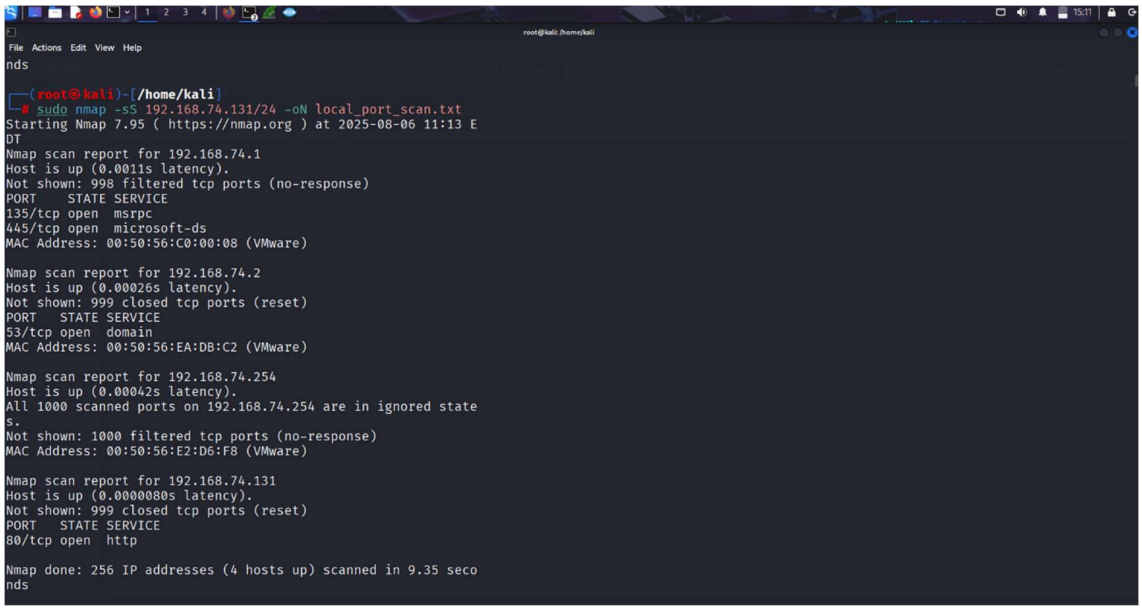
This TCP SYN scan effectively identified active hosts and exposed ports within the local network. The presence of open ports like SMB (445), HTTP (80), and DNS (53) demonstrates potential entry points that could be exploited if not properly secured.

In a real-world scenario, it is important to:

- Regularly audit open ports
- Disable unnecessary services
- Use firewalls to restrict unauthorized access
- Apply patches and updates to all networked systems

4.Note down IP addresses and open ports found.

In this scan, I executed a TCP SYN scan on my entire local subnet using the following command.

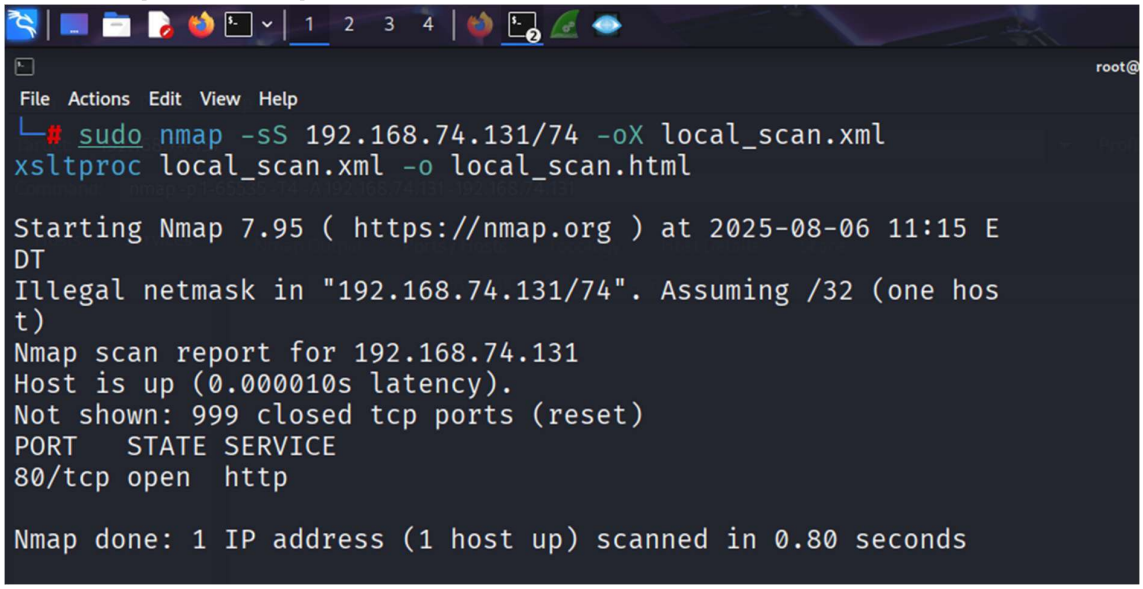


This scan analyzed all devices in the 192.168.74.0/24 network range and saved the output as a .txt file. It successfully detected 4 active hosts:

- 192.168.74.1
  - Open Ports: 135/tcp (MSRPC), 445/tcp (Microsoft-DS)
  - These ports are commonly associated with Windows networking services.
- 192.168.74.2
  - Open Port: 53/tcp (DNS)
  - This suggests the device might be acting as a DNS server or resolver.
- 192.168.74.254
  - All ports filtered; no response
  - Likely protected by a firewall.
- 192.168.74.131 (My Kali VM)
  - Open Port: 80/tcp (HTTP)
  - Indicates a web server or service is active on my own system.

This scan helped identify which machines are active on my local network and what services are exposed. Saving the results to a .txt file made documentation easier for the research report.

▶For better visualization and report inclusion, I exported the scan results in both XML and HTML format using the following commands:-



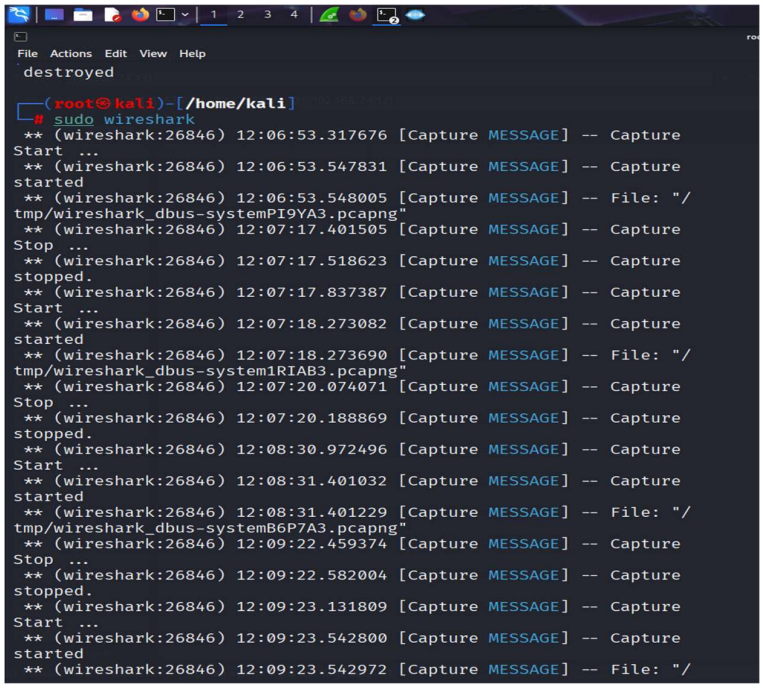
Though I accidentally used /74 instead of /24, Nmap corrected it to /32, meaning it only scanned my own machine (192.168.74.131). The scan result showed:

- Open Port: 80/tcp (HTTP)

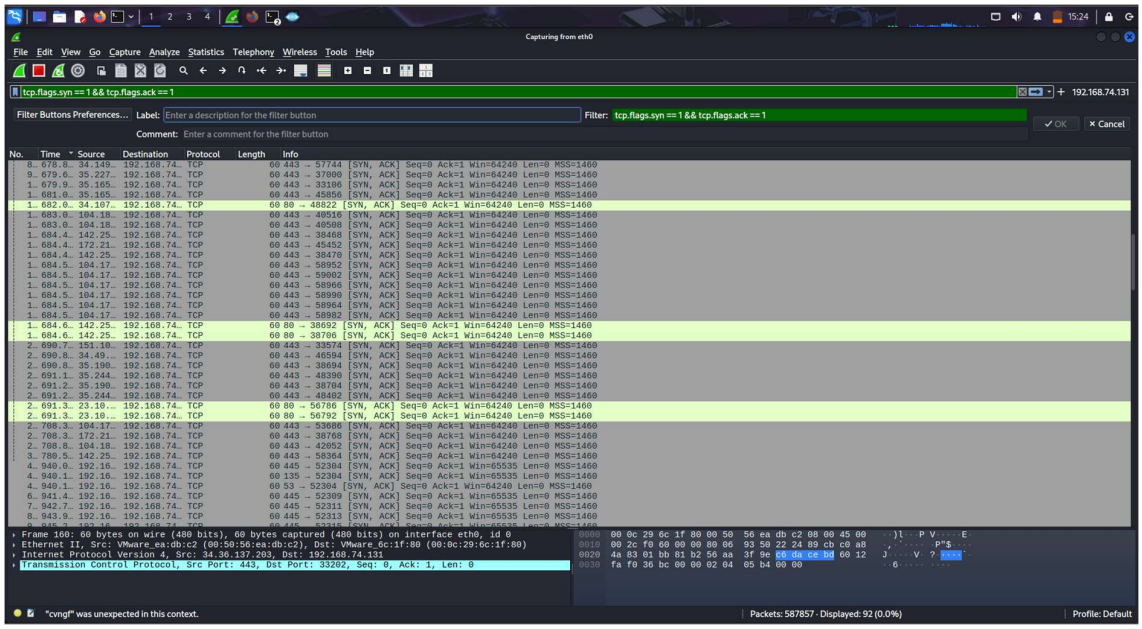
The HTML file generated from the XML format provides a clean and structured visual summary, useful for including in the final internship documentation. It can be opened in any browser and shared with non-technical reviewers.

5.Optionally analyze packet capture with Wireshark

- **Open Wireshark:-**



- **Start capturing on interface (e.g., eth0 or wlan0)**
- **In parallel, run the nmap scan again**
- **Watch SYN/ACK packets (filter with tcp.flags.syn == 1 && tcp.flags.ack == 1)**



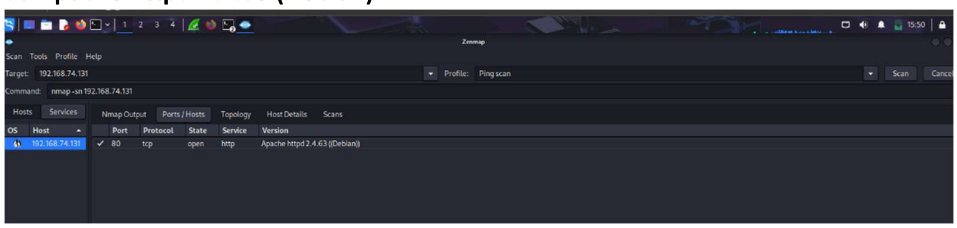
In this Wireshark capture session, I filtered traffic using the expression tcp.flags.syn == 1 && tcp.flags.ack == 1 to isolate TCP SYN-ACK packets. These packets are part of the three-way TCP handshake and indicate that the target host is acknowledging an incoming connection request — confirming the presence of an open port.

From the filtered results, I observed SYN-ACK responses from the IP address 192.168.74.131 to various source IPs. The highlighted packets confirm successful connection attempts, particularly to port 443 (HTTPS) and port 80 (HTTP), which are commonly used for secure and non-secure web traffic, respectively.

This analysis highlights how Wireshark can be used to visually confirm open ports and active services by interpreting real-time traffic. The session also reflects the effectiveness of combining Nmap scans with packet analysis, which together provide both summary and deep inspection of network behavior.

6.Research common services running on those ports.

Using Zenmap (the graphical interface for Nmap), I scanned the local host at IP 192.168.74.131. The scan revealed that port 80 (TCP) was open, and the service running on this port was identified as:**Apache httpd 2.4.63 (Debian)**



This confirms that the host is running a web server that is accessible via the standard HTTP protocol. Apache is a widely used open-source web server, and version details help in determining if the server is up to date or potentially vulnerable.

Note:- Based on the Nmap scan results, several ports were found open on local network devices. Port 80 (HTTP) and 443 (HTTPS) indicated web services, while port 53 (DNS) and port 135 (MSRPC) were related to core network communication and RPC services respectively. Port 445, associated with SMB, was particularly noteworthy due to its historical exploitation in ransomware attacks such as WannaCry. Recognizing the services behind open ports is critical in assessing the vulnerability and exposure of devices in a network.

7.Identify potential security risks from open ports.

As part of the local network scanning exercise using Nmap and Zenmap, we identified multiple active devices along with their open ports and associated services. While discovering open ports is a critical step in understanding the exposure of a network, it is equally important to assess the security risks posed by those ports and the services running on them.

🧠 What This Step Is Really About

When we scan our local network, it's not just a technical process of finding ports — it's an exercise in thinking like a hacker and defending like a security analyst.

Each open port is like a *door* into a system. Some doors are protected, some are locked but can be picked, and some are left wide open without anyone noticing. This step is about finding those doors, understanding what's behind them, and asking:

- Should this door even exist?
- If yes, is it locked properly?
- If not, how dangerous is it?

🧪 Services Identified (Recap of My Scan)

After scanning my local subnet 192.168.74.0/24 using Nmap with TCP SYN scans (-sS) and analyzing it through Zenmap, I discovered the following devices and services:

IP Address	Port(s)	Service(s) Identified	OS/Details
192.168.74.1	135, 445	MSRPC, Microsoft-DS (SMB)	Likely Windows Host
192.168.74.2	53	DNS (Domain Name Service)	Internal resolver or VM guest
192.168.74.131	80	Apache 2.4.63 HTTP (Debian)	My Kali VM
192.168.74.254	—	All ports filtered or ignored	VMNet gateway or firewall

Now, let's go beyond the numbers and analyze what these services truly mean in terms of network exposure and risk.

🧩 **1. Port 80 (HTTP - Apache Server)**

- **Where?** 192.168.74.131 (My Kali VM)
- **What is it?** Apache HTTP Server v2.4.63 — A popular open-source web server.
- **How it behaves?** Accepts connections over port 80 (unencrypted HTTP).

🔍 **Human Thinking:**

"Imagine sending your passwords and personal data on a postcard that anyone can read. That's HTTP."

🔓 **Risk Breakdown:**

- **Unencrypted Protocol:** Since it's not using HTTPS (SSL/TLS), all communication is in *plain text*. This means anyone using tools like Wireshark on the same network could *sniff usernames, passwords, cookies, or search queries*.
- **Fingerprinting Vulnerability:** The exact server version (Apache 2.4.63) is exposed. Attackers can Google "Apache 2.4.63 vulnerabilities" and find known exploits.
- **Common CVEs:**
    - CVE-2023-25690: Rewrite rule injection vulnerability.
    - CVE-2022-31813: Denial-of-Service in mod_proxy module.

🛡️ **Mitigation Recommendations:**

- **If not needed, disable it.**
- Use **HTTPS with SSL/TLS certificates** (even self-signed for local).
- Remove or restrict version disclosure headers.
- Apply security modules: mod_security, mod_evasive.

🧩 **2. Ports 135 & 445 (MSRPC + SMB - Windows File Sharing)**

- **Where?** 192.168.74.1
- **What is it?** Microsoft Remote Procedure Call (RPC) and SMB file-sharing protocols.
- **How it behaves?** Allows remote communication for services like file and printer sharing.

🔍 **Human Thinking:**

"This is the same tech that let ransomware like WannaCry spread across the world in hours."

🔓 **Risk Breakdown:**

- **Historically Vulnerable:** These ports are **notorious attack surfaces**, especially port 445 used by SMBv1.

- **WannaCry (2017)** and **NotPetya (2017)** ransomware used **SMB vulnerabilities (EternalBlue)** to propagate. Those attacks began *exactly* through these ports.
- **No Authentication / Poor Configuration:** If file shares are configured for "Everyone" or with weak permissions, an attacker can **read/copy/exfiltrate files** from your device without even hacking it.

🛡️ **Mitigation Recommendations:**

- **Disable SMBv1.**
- Restrict port 445 and 135 using firewalls to **local only** access.
- Patch Windows regularly to close SMB-related CVEs.
- Avoid anonymous shares or "guest" access.

🧩 **3. Port 53 (DNS)**

- **Where?** 192.168.74.2
- **What is it?** A DNS server – resolves domain names to IP addresses.
- **How it behaves?** Accepts and processes DNS queries.

🔍 **Human Thinking:**

"A DNS server is like your internet phonebook. If it's hacked, it could send you to the wrong house."

🔓 **Risk Breakdown:**

- **DNS Amplification Attacks:** If exposed to the internet, this can be used in **DDoS attacks**.
- **Spoofing & Poisoning:** An attacker could inject fake DNS responses, tricking users into visiting **phishing websites**.
- **Recursive Resolution Open to All:** If the DNS server allows queries from any IP, it can be **abused by attackers as an open resolver**.

🛡️ **Mitigation Recommendations:**

- Block external DNS access.
- Disable recursion for unauthorized IPs.
- Monitor DNS logs for anomalies (e.g., large packet sizes).

🚫 **What Was Not Found – And Why That's Good**

Sometimes, what you *don't* find is just as important as what you do. Here's what I **did not** detect:

Service	Port	Status	Security Meaning
Telnet	23	Not Found	✅ Good – it's outdated and insecure
SSH	22	Not Found	✅ Good – remote access locked down
RDP	3389	Not Found	✅ Good – no unprotected desktop access

Had these services been active, especially without encryption or brute-force protection, it would have opened up new threat vectors.

🔬 CVE Search & Vulnerability Research

I used **https://cvedetails.com** to research known vulnerabilities in the identified services.

Examples:

- **Apache 2.4.63**
  - CVE-2023-31122 – File disclosure via mod_macro
  - CVE-2022-30522 – Denial of service via HTTP/2
- **SMB on Windows**
  - CVE-2017-0144 (EternalBlue) – Used by WannaCry
- **BIND (DNS)**
  - CVE-2023-2828 – DoS via crafted DNS responses

Each of these has a **CVSS score** (Common Vulnerability Scoring System), which helps assess severity.

🧠 Final Thoughts: Thinking Like an Attacker – and a Defender

What this task really taught me is:

- Attackers **don't need to break down the door** — they just look for the ones you forgot to lock.
- Even a **harmless-looking open port** can become a serious weakness.
- Every service should be questioned: *"Do I need it? If yes, is it secured? If not, what's the risk?"*

📌 Overall Conclusion

Open ports are more than just technical data — they are **entry points**, **risk zones**, and **possible vulnerabilities**. By identifying and analyzing them:

- We reduce our **attack surface**.
- We learn to **think proactively about security**.
- We protect users, data, and systems from being exposed, abused, or compromised.

This step was a powerful, real-world lesson in **network visibility, threat detection, and responsible system administration** — exactly what a modern security analyst must understand.