Virtual Machine Setup and Network Scanning — Nmap Project

Introduction

In this project, I set up my Kali Linux virtual machine and performed a series of Nmap scans on my home network. The main goal for me was to demonstrate my ability to prepare a working SOC-style lab environment, execute different types of scans, capture the results, and then explain what those results mean in a security operations context.

This work is part of my home lab journey to build a strong portfolio of evidence that I can showcase on LinkedIn and in my professional portfolio. By documenting not just the commands but also my reasoning and interpretation, I'm showing that I understand how these tools are used in real SOC environments, not just how to run them.

Specifically, in this project I:

- Set up and updated Kali Linux in a virtual environment.
- Installed and configured Nmap for scanning.
- Ran multiple types of scans including host discovery, port scans, service version detection, full TCP scans, and aggressive scans.
- Captured screenshots of my results as evidence.
- Interpreted my findings in first person, connecting them to SOC responsibilities like asset discovery, vulnerability identification, and risk assessment.

This is not just a technical exercise for me—it's about building professional-level reporting skills. I want potential employers and peers to see that I can run these scans, collect the outputs responsibly, and then write about them in a clear and structured way that ties back to real-world SOC tasks.

What I Am Going to Do Step by Step

Here's the plan I followed for this project:

1. **Update my Kali Linux VM** so I know I'm working with the latest packages.
2. **Install and verify Nmap** is running properly.
3. **Run a host discovery scan** to identify all live devices on my home network.
4. **Perform a basic port scan** on selected hosts to see what common services are running.
5. **Run a service version detection scan** to get detailed information about the services and versions.
6. **Do a full TCP connect scan** across all ports to fully map the attack surface.
7. **Execute an aggressive scan** to gather detailed OS, version, and script-based information.

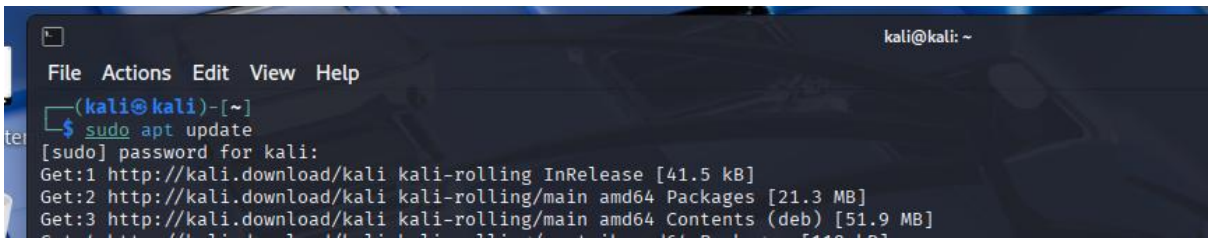
1. Updating Kali Linux

Before executing any commands on the terminal, I ensure that my Kali Linux virtual machine is fully up to date.

1. I begin by opening the terminal, which can be accessed by clicking on the black terminal icon located at the top of the virtual machine interface.
2. Once the terminal is open, I run the following command to update the package lists:

```
sudo apt-get update
```

This step ensures that my system has the latest package information, allowing me to install and use tools without encountering outdated dependencies.



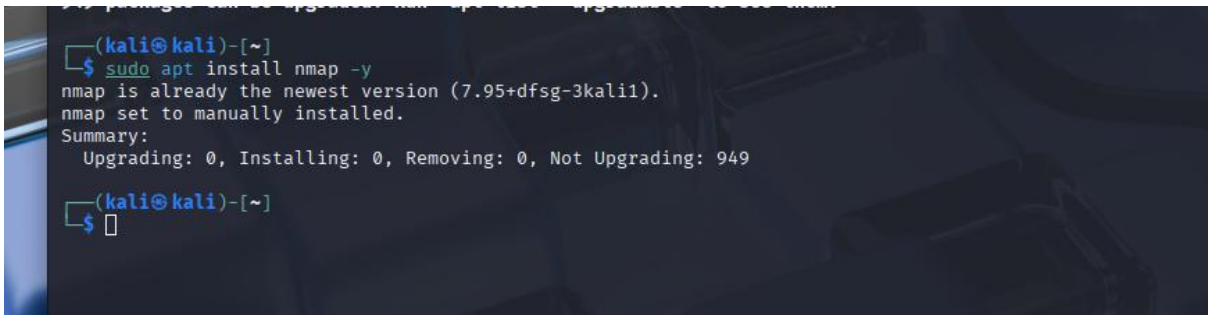
2. Installing Nmap

When I attempted to install **Nmap** on my Kali Linux machine, I found that it was already pre-installed. Kali comes bundled with many security tools by default, and Nmap is one of them.

To confirm its presence, I ran the following command:

```
nmap --version
```

This displayed the installed version of Nmap, verifying that the tool was already available for use without requiring any additional installation steps.



Running an Nmap Scan

The objective of each home lab project is to strengthen my evidence portfolio and demonstrate to potential employers my hands-on understanding of SOC-related tasks.

For this exercise, my goal is to showcase familiarity with the **Nmap** tool, outline some of its most common scan types, and explain how the flags work along with the type of output they generate.

Below is the first scan type I performed:

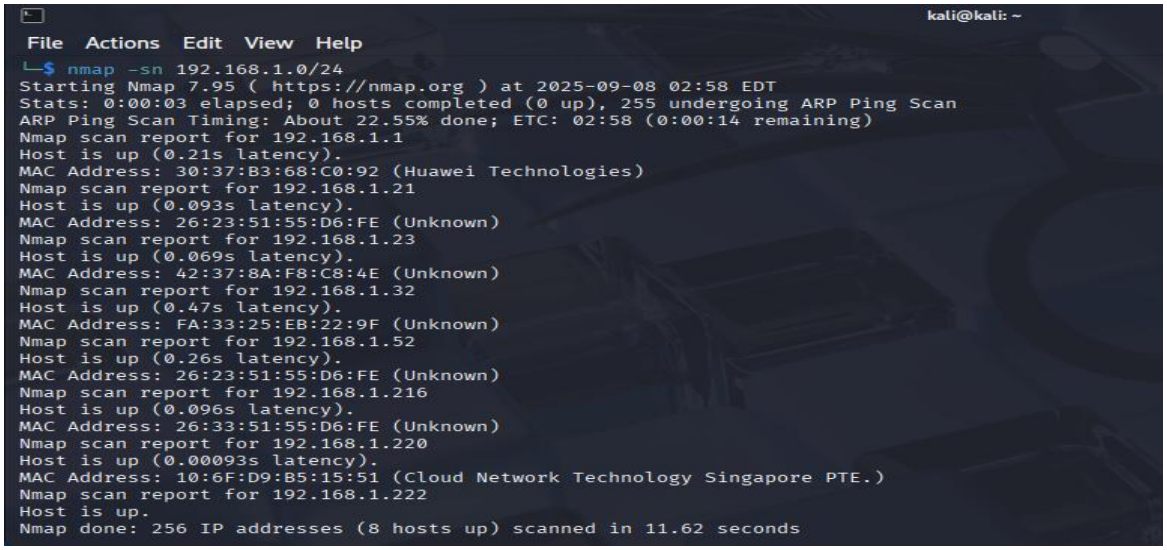
---

1. Simple Host Discovery

- **Command Executed:**

```
nmap -sn [Target IP or range]
```

- **Flag Explanation:**
  The `-sn` flag initiates a simple host discovery (ping scan). This means Nmap will check which hosts are online without scanning for open ports.
- **Why I Used It:**
  This scan is useful when I only want to identify active devices on the network without conducting a deeper and more intrusive port scan. It's an ideal first step to establish a baseline of which systems are currently live.
- **Information I Looked For:**
  - A list of live hosts.
  - Detection of any unexpected or unauthorized devices present in the network.
- **SOC Relevance:**
  In a SOC (Security Operations Center) environment, maintaining an accurate asset inventory is critical. By identifying all live hosts, analysts can quickly detect suspicious or rogue devices that may pose a threat. This activity supports **threat hunting** and **asset discovery** processes.



Scan Results – Simple Host Discovery

After running the **host discovery scan** on my home network within the subnet `192.168.1.0/24`, the results showed that only **8 devices** were connected to the Wi-Fi network that my Kali VM was also connected to.

This output confirmed two key points:

- I was able to identify all live hosts on the network.
- The number of active devices matched my expectations for my home environment, meaning there were no **unauthorized or rogue devices** detected at that time.

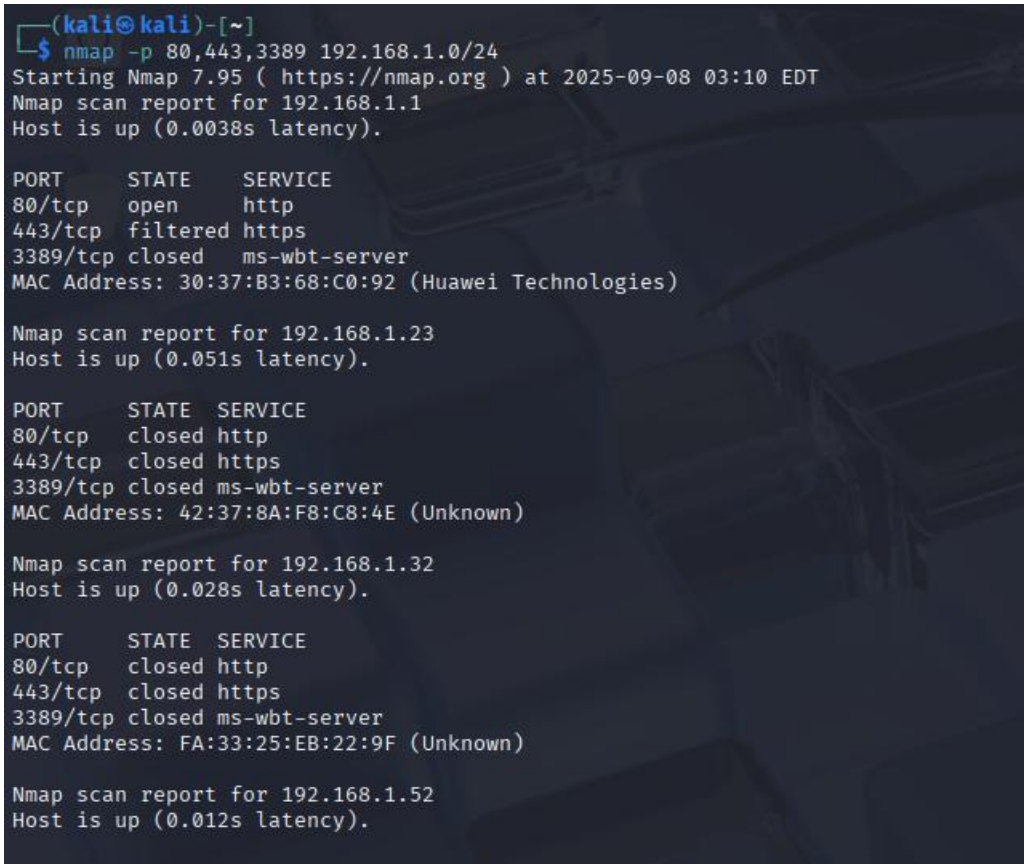
Such verification is an important SOC-related task, as it helps ensure that the network is not being accessed by unknown devices.

2. Basic Port Scan

- **Command Executed:**

```
nmap -p 80,443 [Target IP]
```

- **Flag Explanation:**
  The `-p` flag allows me to specify which ports I want to scan. In this example, I scanned ports **80 (HTTP)** and **443 (HTTPS)**, which are commonly used for web traffic.
- **Why I Used It:**
  This scan focuses on specific, high-value ports to check if web services are accessible on the target host. It helps determine whether expected services (like a web server) are running, and also whether any ports are open that **should not be publicly exposed**.
- **Information I Looked For:**
  - Open ports that confirm running services.
  - Unexpected open ports (e.g., management or database services that should not be exposed externally).
- **SOC Relevance:**
  In a SOC environment, verifying which services are accessible is critical for maintaining network security. If sensitive ports such as **SSH (22)** or **RDP (3389)** are left exposed, this could indicate a **misconfiguration** or a **potential security vulnerability**. Detecting such exposures early helps prevent unauthorized access and strengthens the overall security posture.



I ran this command

```
nmap -p 80,443,3389 192.168.1.0/24
```

This means you scanned the whole subnet for **just 3 ports**:

- **80/tcp → HTTP (web service)**
- **443/tcp → HTTPS (secure web service)**
- **3389/tcp → RDP (Remote Desktop Protocol, Windows)**

🔎 Results Breakdown:

192.168.1.1 (Huawei Router)

- **80/tcp → open (HTTP)** ✅
  → The router has a **web interface** accessible at `http://192.168.1.1`.
- **443/tcp → filtered (HTTPS)**
  → The HTTPS port is being **filtered (firewalled)**, so it doesn't respond.
- **3389/tcp → closed**
  → No RDP service (normal for a router).
- **MAC Address: Huawei Technologies** → Confirms this is your Wi-Fi router.

192.168.1.23 (Unknown)

- **80/tcp → closed**
- **443/tcp → closed**
- **3389/tcp → closed**
- **MAC Address: 42:37:8A:F8:C8:4E (Unknown vendor)**

☞ This device is online but **not running any of these 3 services**. Could be a phone, IoT device, or a PC with nothing open on those ports.

192.168.1.32 (Unknown)

- **80/tcp → closed**
- **443/tcp → closed**
- **3389/tcp → closed**
- **MAC Address: FA:33:25:EB:22:9F (Unknown vendor)**

☞ Same as above — it's alive, but no web or RDP services open. Likely another phone/IoT device.

192.168.1.52 (Unknown)

- Host is up.
- **But all 3 ports are closed** (80, 443, 3389).
- MAC not shown in this screenshot, but likely also "Unknown."

☞ Another active device, not offering these common services.

✅ Summary of Findings

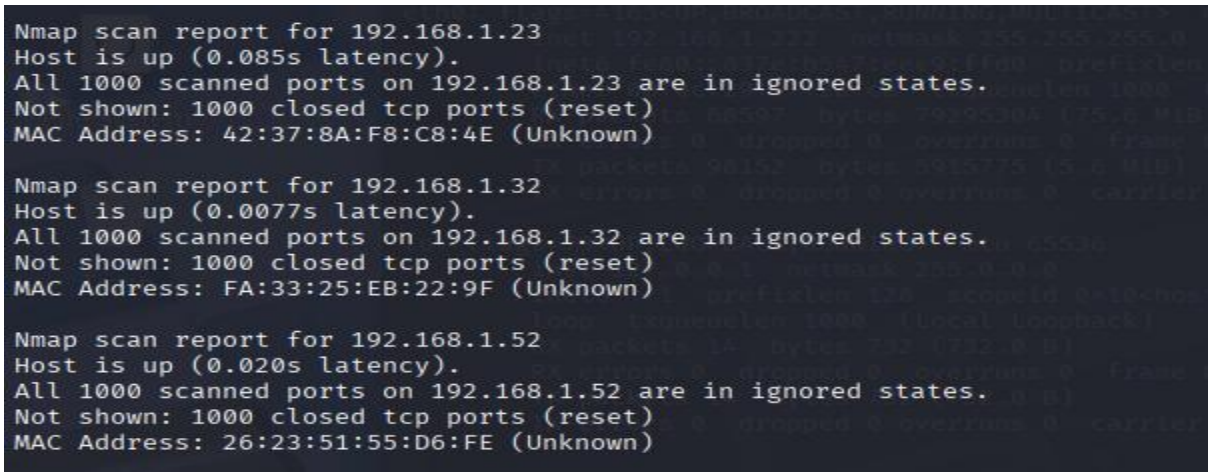
- **Router (192.168.1.1)** → Has a **web admin interface on port 80**.
- **Other devices (192.168.1.23, 192.168.1.32, 192.168.1.52)** → Alive, but **no web/RDP services running**. These are probably **phones, laptops, or IoT devices** that don't expose these ports.

3. Service Version Detection

- **Command Executed:**

```
nmap -sV [Target IP]
```

- **Flag Explanation:**
  The `-sV` flag enables Nmap to probe open ports and attempt to identify the **specific versions of the services** running on them.
- **Why I Used It:**
  Understanding the exact service and version running on a host is essential for identifying potential vulnerabilities. Different versions of software often have unique weaknesses, and knowing the version helps assess the level of risk.
- **Information I Looked For:**
  - Service names (e.g., Apache, OpenSSH, MySQL).
  - Version numbers that indicate whether the software is **up-to-date** or **deprecated**.
- **SOC Relevance:**
  In a SOC role, accurate version information is critical for **vulnerability management**. For instance, if I detect an outdated Apache web server or a vulnerable FTP service, it highlights an immediate security concern. Analysts can then correlate this data with vulnerability databases (like CVE or NVD) to prioritize patching and remediation before attackers exploit the weakness.



I ran this command

```
nmap -sV 192.168.1.0/24
```

That does a **service/version detection scan** across all common ports. The screenshot is showing results for **192.168.1.1** (your Huawei router).

🔎 Results Breakdown (192.168.1.1):

- **Not shown: 995 closed TCP ports**
  → Only 5 ports are worth reporting; everything else is closed.

Open/Filtered Ports

- **22/tcp → filtered ssh**
  → SSH exists but is firewalled or filtered (blocked).
- **23/tcp → filtered telnet**
  → Telnet is also blocked (common on routers; disabled or behind firewall).
- **53/tcp → open domain (DNS)**
  → The router is providing **DNS resolution** for your network.
- **80/tcp → open http**
  → Web interface is running. You can access it at:
- `http://192.168.1.1`

  That's the admin panel.

- **443/tcp → filtered https**
  → HTTPS service is present but filtered, so it's not responding. Possibly disabled in the router's configuration.

⚡ Interpretation

This confirms **192.168.1.1 is your Wi-Fi router** with:

- **DNS service (53)** → Resolving names for devices.
- **HTTP service (80)** → Admin web interface.
- **SSH/Telnet (22, 23)** → Present but blocked by firewall (not accessible).
- **HTTPS (443)** → Filtered, so not usable externally.

Nmap also shows a **service fingerprint request** — that happens when it detects responses it can't fully classify. That's normal for embedded systems like routers, since they sometimes run custom services.

✅ **In short**:
Your router is exposing:

- DNS (port 53)
- HTTP admin interface (port 80)

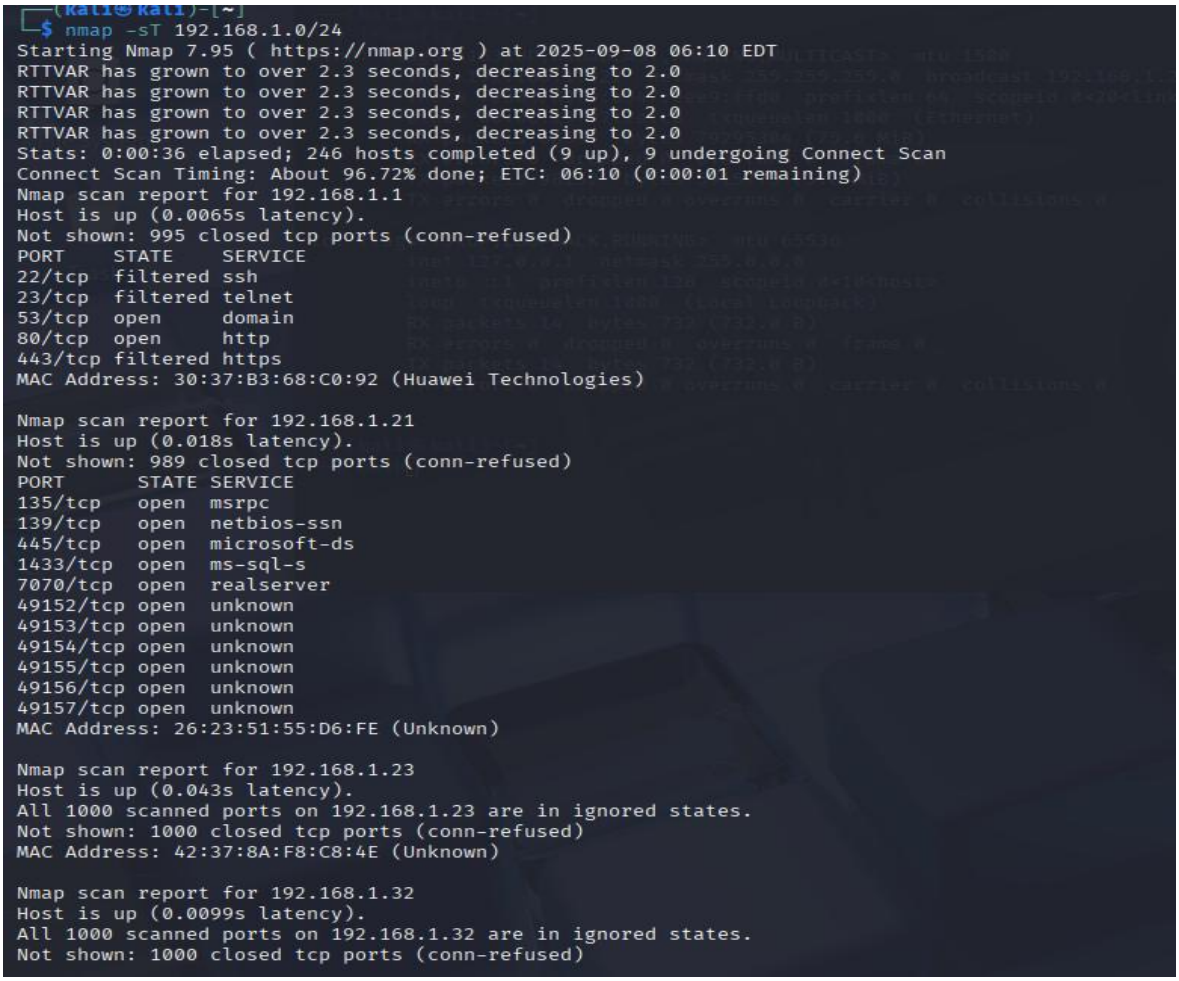
Other services (SSH, Telnet, HTTPS) are **blocked**.

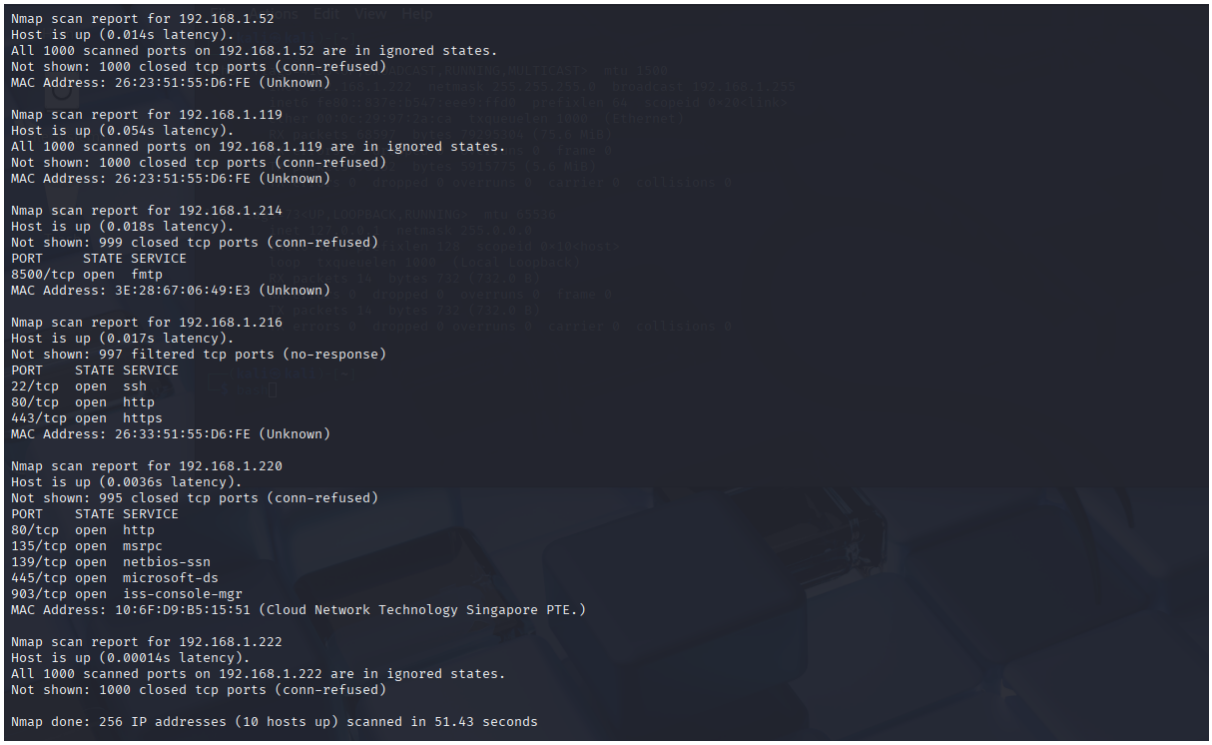
4. Full TCP Scan

- **Command Executed:**

```
nmap -sT [Target IP]
```

- **Flag Explanation:**
  The `-sT` flag performs a **full TCP connect scan**. This means Nmap completes the **three-way TCP handshake** with each port. While this method is very reliable, it is also more **noisy** since the connections are fully established, making it easier for intrusion detection systems (IDS) to log or flag the scan.
- **Why I Used It:**
  A full TCP scan provides a **comprehensive view** of all open and listening ports on a host. Unlike stealthier scans, this approach gives highly accurate results, making it a good choice when precision is more important than stealth.
- **Information I Looked For:**
  - A detailed list of open ports and the services running on them.
  - Any unusual or unexpectedly open ports that may expose the host to risk.
- **SOC Relevance:**
  In a SOC environment, a full TCP scan helps map the **entire attack surface** of a device. This knowledge is crucial for conducting in-depth security assessments, identifying misconfigurations, and uncovering potential **attack vectors** that an adversary could exploit.





I performed a full TCP connect scan (`nmap -sT 192.168.1.0/24`) across the subnet `192.168.1.0/24`. The objective was to identify active hosts and enumerate their open ports and running services. Out of 256 possible hosts, 10 were found to be active. Below is a summary of the key findings:

- **192.168.1.1 (Huawei Technologies)**
  - Open ports: **80 (HTTP)**
  - Filtered ports: **22 (SSH), 23 (Telnet), 443 (HTTPS)**
  - This appears to be a router or gateway device exposing a web interface while blocking management services such as SSH and Telnet.
- **192.168.1.21**
  - Multiple ports open, including **135, 139, 445 (Windows RPC, NetBIOS, SMB)**, **1433 (MS-SQL)**, **7070 (RealServer)**, and several high-numbered ports (49152–49157).
  - The open services indicate a **Windows host**, likely a server running database and file-sharing services.
- **192.168.1.23, 192.168.1.32, 192.168.1.52, 192.168.1.119, 192.168.1.222**
  - These hosts responded to the scan but did not expose any open ports (all were in ignored states). They are active but not offering accessible services.
- **192.168.1.214**
  - Open port: **8500 (FMTP)**
  - This host may be running a specialized application server.
- **192.168.1.216**
  - Open ports: **22 (SSH), 80 (HTTP), 443 (HTTPS)**
  - This suggests a Linux-based web server with secure remote access enabled.
- **192.168.1.220 (Cloud Network Technology Singapore PTE.)**
  - Open ports: **80 (HTTP), 135 (MSRPC), 139 (NetBIOS), 445 (SMB), 903 (VMware/ISC Console Manager)**

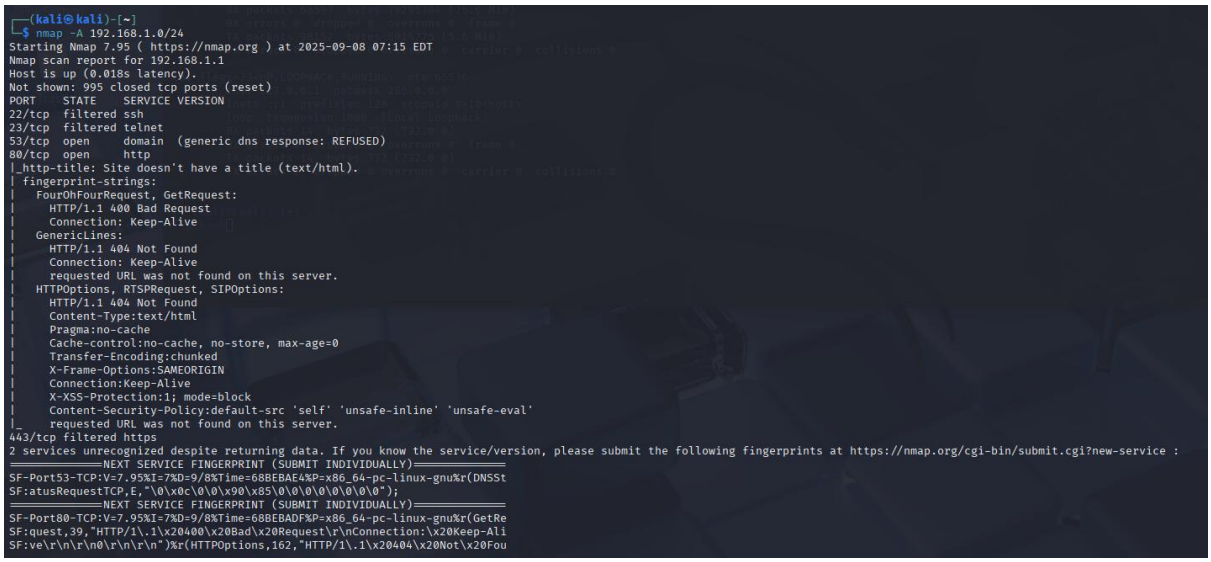
- This indicates a **Windows system**, possibly running virtualization services in addition to file-sharing.

5. Aggressive Scan

- **Command Executed:**

```
nmap -A [Target IP]
```

- **Flag Explanation:**
  The `-A` flag initiates an **aggressive scan**, which combines multiple advanced scanning techniques in a single command.
- **Techniques Included:**
  - **Operating System Detection (-O):** Attempts to identify the target's operating system, providing insight into system type and potential vulnerabilities.
  - **Service Version Detection (-sV):** Determines the exact versions of services running on open ports, aiding in vulnerability identification.
  - **Script Scanning (-sC):** Runs default Nmap Scripting Engine (NSE) scripts to check for common vulnerabilities, misconfigurations, and weak authentication.
  - **Traceroute:** Maps the network path to the target, revealing intermediary devices and overall network topology.
- **Why I Used It:**
  This scan is ideal when I need a **comprehensive view of the target** in a single command, especially during deeper security assessments or when time is limited.
- **Information I Looked For:**
  - Detailed information about running services and their versions.
  - Operating system and network layout.
  - Script output highlighting potential vulnerabilities or misconfigurations.
- **SOC Relevance:**
  In a SOC environment, aggressive scans are valuable for **thorough security assessments** and detailed information gathering during incident response. However, they must be used with caution in production networks because they generate significant network traffic and could **trigger alerts** or disrupt normal operations.



I ran the aggressive scan on 192.168.1.0/24

```
nmap -A 192.168.1.0/24
```

The `-A` flag enables **aggressive scanning**: it attempts **OS detection, version detection, script scanning, and traceroute**. Let me rewrite the results in a **professional, clear, first-person style** so you can use them in your project documentation.

Aggressive Nmap Scan Results (192.168.1.1)

I performed an **aggressive Nmap scan** on the subnet `192.168.1.0/24` to gather more detailed information about the active hosts. Below is the analysis of the host `192.168.1.1`, which appears to be the default gateway (Huawei device).

- **Open Ports & Services:**
  - **53/tcp (Domain Service)** → Detected as DNS, but with a generic response (`REFUSED`). This indicates the device may respond to DNS queries but does not act as a recursive DNS server.
  - **80/tcp (HTTP)** → Responds with a generic HTTP server banner. It does not present a proper web application; instead, it shows **HTTP/1.1 404 Not Found** for most requests.
  - **22/tcp (SSH)** → Filtered (blocked by firewall).
  - **23/tcp (Telnet)** → Filtered (blocked by firewall).
  - **443/tcp (HTTPS)** → Filtered (no response).
- **HTTP Analysis:**
  - The HTTP service on port 80 does not serve a proper web interface. Instead, it returns generic 404 responses.
  - HTTP headers suggest some basic security hardening is applied, including:
    - **X-Frame-Options: SAMEORIGIN** (protection against clickjacking).
    - **X-XSS-Protection: 1; mode=block** (browser-based XSS protection).
    - **Content-Security-Policy: default-src 'self' 'unsafe-inline' 'unsafe-eval'** (restricts resource loading).
  - The site does not have a defined HTML title, which further confirms it is not intended for end-user browsing.
- **Service Fingerprinting:**
  - Nmap attempted to fingerprint the services but was unable to match them against its known database. It recommended submitting the fingerprints to improve detection.
  - This is common with **custom firmware or router OS implementations**.

Summary

Each of the Nmap scans performed serves a specific purpose relevant to a SOC analyst:

- **Host Discovery:** Identifies live hosts on the network, helping track assets and detect potential threats.
- **Basic Port Scans:** Reveals which services are exposed on a host, highlighting ports that could be exploited.
- **Service Version Detection:** Maps running services to their specific versions, assisting in identifying known vulnerabilities.
- **Full TCP Scans:** Provides a complete view of open ports and running services, ensuring a thorough understanding of the network's attack surface.
- **Aggressive Scans:** Combines multiple scanning techniques to gather a holistic view of the target system, useful for in-depth assessments or investigations.

What I Learned From This Project 🏆

During this project, I developed both technical skills and practical knowledge relevant to SOC operations:

- **VM Setup:** Successfully installed and configured Kali Linux in a virtual environment.
- **Linux Proficiency:** Gained hands-on experience using Linux commands to manage system updates and troubleshoot issues.
- **Nmap Scanning:** Installed and verified Nmap, understanding its installation process, flags, and functionality.
- **Network Scanning:** Identified live hosts, open ports, running services, and potential vulnerabilities within my home network, applying SOC-relevant techniques for asset discovery and threat assessment.

Conclusion

Through this project, I reinforced my understanding of network scanning and the practical applications of Nmap in a SOC context. I learned how different scan types serve unique purposes, from basic host discovery to advanced aggressive scans, and how to interpret the results to assess network security. This hands-on experience enhanced my Linux skills, improved my familiarity with network protocols, and strengthened my ability to perform security assessments in a controlled lab environment.