# Task 1: Scan Your Local Network for Open Ports.

Intern Name: Jainam Priteshkumar Shah

Date: 04/08/2025

Internship Provider: Elevate Labs

## 1. Objective

The primary objective of this task is to learn how to discover open ports on devices within the local network. This exercise is fundamental to understanding the network's exposure and identifying potential security vulnerabilities.

## 2. Tools Used

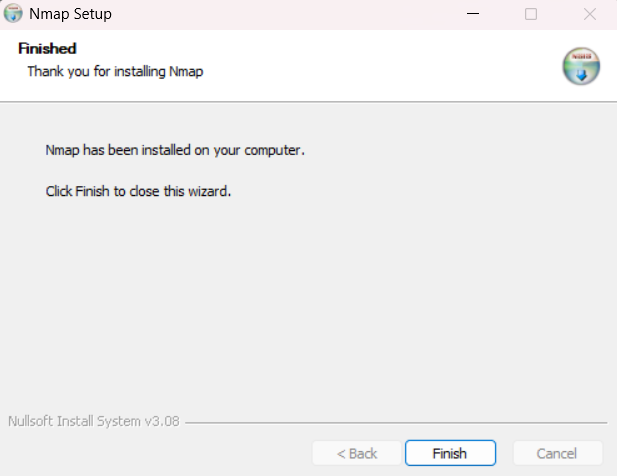
* **Nmap:** A free and open-source network scanner used for network discovery and security auditing.
* **Wireshark (Optional):** A network protocol analyzer used to capture and inspect network traffic.

## 3. Procedural Steps

This section details the step-by-step process followed to complete the network scan.

### 3.1. Tool Installation

Downloaded Nmap from the official website ([https://nmap.org](https://nmap.org/download.html#windows)) and installed it on the local machine.

****

### 3.2. Identifying the Local IP Range

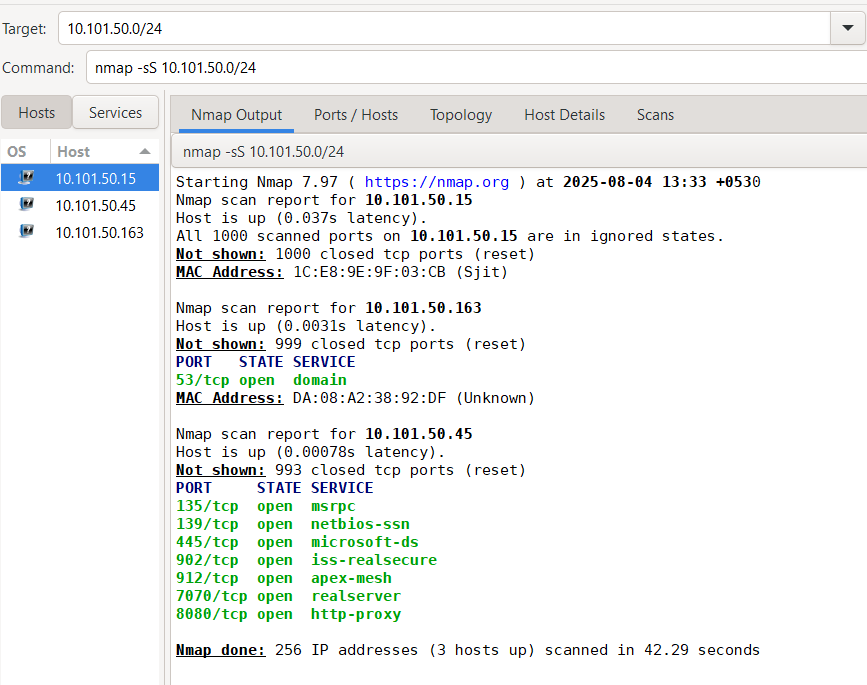
for scanning my local network, I first determined my local IP address range. I accomplished this by using the **‘ipconfig’** command in the Windows Command Prompt. Based on the output, I identified my network range for the scan as **10.101.50.0/24**.



### 3.3. Executing the Network Scan

A TCP SYN scan (-sS) was performed using Nmap to efficiently and stealthily identify open ports on the devices within the discovered network range. The following command was executed:

**nmap -sS 10.101.50.0/24**

****

### 3.4. Scan Results

The scan identified several active hosts and their open ports.

**Active Hosts:**

* 10.101.50.15
* 10.101.50.163
* 10.101.50.45

**Open Ports:**

* For 10.101.50.163
  + 53/tcp → domain
* For 10.101.50.45
  + 135/tcp → msrpc
  + 139/tcp → netbios-ssn
  + 445/tcp → microsoft-ds
  + 902/tcp → iss-realsecure
  + 912/tcp → apex-mesh
  + 7070/tcp → realserver
  + 8080/tcp → http-proxy

### 3.5. Analysis of Open Ports and Services

A detailed investigation was conducted on the discovered open ports to understand the services running on them and to assess potential security risks.

| **IP Address** | **Open Port/Protocol** | **Service** | **Potential Security Risks** |
| --- | --- | --- | --- |
| *10.101.50.163* | *53/tcp* | *domain* | ***DNS Zone Transfer:*** *If misconfigured, could allow an attacker to retrieve all DNS records, revealing the internal network structure.*  ***DNS Amplification Attack:*** *The server could be abused as part of a DDoS attack.* |
| *10.101.50.45* | *135/tcp* | *msrpc* | ***Vulnerability Exploitation:*** *MSRPC has a history of critical vulnerabilities (e.g., Blaster worm). An unpatched system is at high risk.*  ***Information Leakage:*** *Can be enumerated to reveal Windows version, patch level, and running services.* |
| *10.101.50.45* | *139/tcp* | *netbios-ssn* | ***Information Leakage:*** *Can reveal system name, domain, and workgroup information via null sessions. It's a legacy protocol that can indicate an older, less secure configuration.* |
| *10.101.50.45* | *445/tcp* | *microsoft-ds* | ***Critical Vulnerabilities:*** *This port is famously targeted by worms like WannaCry (exploiting EternalBlue). An unpatched system is extremely vulnerable to remote code execution and ransomware.* |
| *10.101.50.45* | *902/tcp* | *iss-realsecure* | ***VMware Management Exposure:*** *This port is used by VMware vSphere. If exposed, it could allow an attacker to target the virtualization host, potentially compromising all guest virtual machines.* |
| *10.101.50.45* | *912/tcp* | *apex-mesh* | ***Management Software Vulnerabilities:*** *The risk depends on the specific APEX-based software. Vulnerabilities in this management or monitoring tool could lead to system compromise.* |
| *10.101.50.45* | *7070/tcp* | *realserver* | ***Web Application Vulnerabilities:*** *Often used for Java-based web applications (e.g., Apache James) which could be vulnerable to common web attacks (SQLi, XSS) or have insecure default configurations.* |
| *10.101.50.45* | *8080/tcp* | *http-proxy* | ***Misconfigured Proxy:*** *If it's an open proxy, it can be abused by attackers to hide their traffic.*  ***Web Server Vulnerabilities:*** *If it's a development server (e.g., Apache Tomcat), it may not be hardened and could have default credentials or known exploits.* |

### 3.6. Optional: Packet Analysis with Wireshark

To get a deeper understanding of the scan process, I used Wireshark to capture and analyze the packets exchanged during my Nmap scan. This visually confirmed the "half-open" nature of the SYN scan.

Here are the steps I followed:

1. Started Wireshark: I launched Wireshark and selected the correct network interface that I was using for the scan (e.g., Wi-Fi or Ethernet).
2. Began Packet Capture: I started the live packet capture by clicking the blue shark fin icon. At this point, Wireshark began logging all traffic on my network interface.
3. Ran the Nmap Scan: With the capture running, I executed my Nmap command (nmap -sS 10.101.50.0/24) in a separate terminal window.
4. Stopped the Capture: Once the Nmap scan was complete, I stopped the Wireshark capture by clicking the red square icon.
5. Filtered the Traffic: To isolate the scan traffic from other network noise, I used a display filter. For example, to see the traffic between my machine and the target 10.101.50.45, I used the filter: **ip.addr == 10.101.50.45 and tcp**.
6. Analyzed the Packets: In the filtered view, I could clearly see the TCP SYN scan in action:
   * SYN Packet: My machine sent a packet with the [SYN] flag to a port on the target.
   * SYN/ACK Response (Open Port): For open ports, the target replied with a packet flagged [SYN, ACK].
   * RST Packet (Closed Port): For closed ports, the target replied with a packet flagged [RST, ACK].
   * My Final RST: Crucially, after receiving a [SYN, ACK], my machine immediately sent a [RST] packet to tear down the connection, confirming it was a "half-open" scan and not a full connection.

