# Network Scanning and Analysis Using Nmap

## Objective

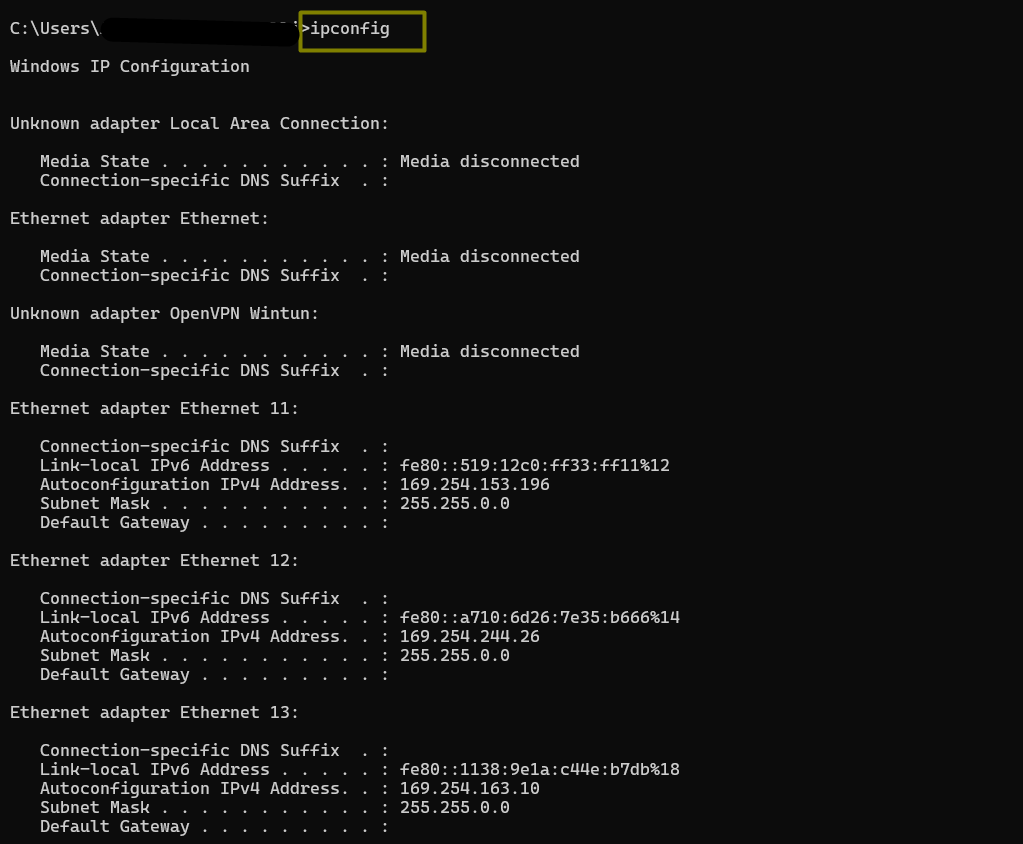
This document outlines the practical steps taken to perform network scanning using **Nmap**, the observations from the scan, and research on common vulnerabilities associated with identified open ports.

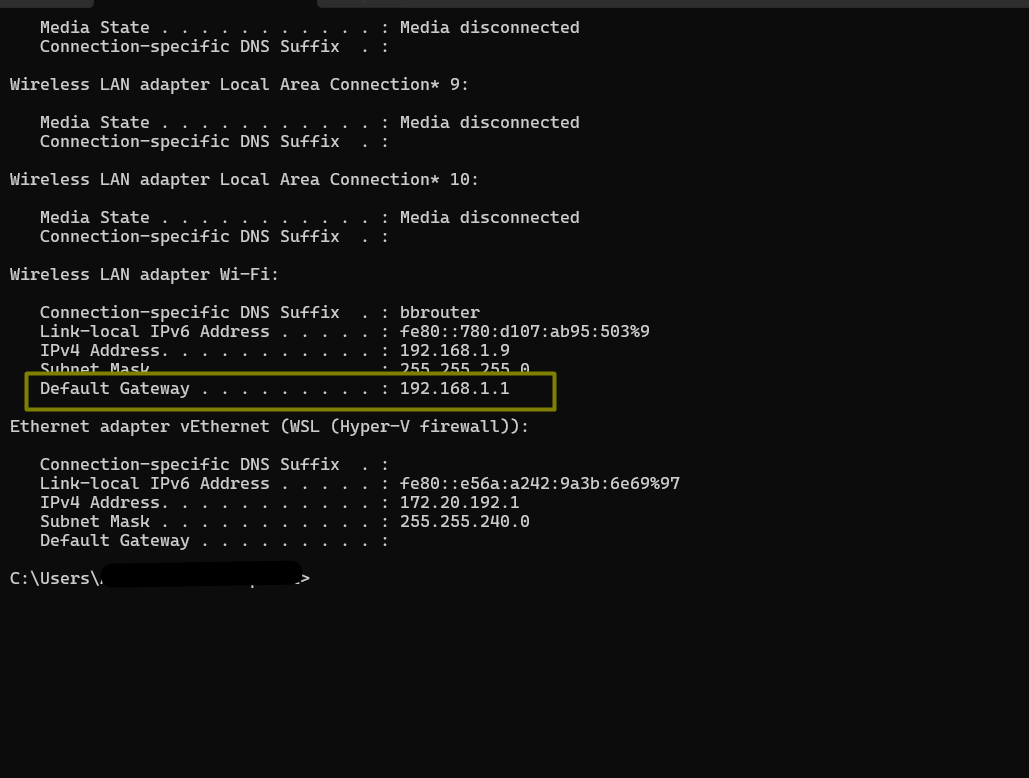
## 1. Installed Nmap

* Downloaded Nmap from the official site: <https://nmap.org/download.html>
* Installed it on a Windows/Linux system for scanning and analysis.

## 2. Identified Local IP Range

* Executed ipconfig (Windows) / ip a (Linux) to find the local IP: 192.168.1.1



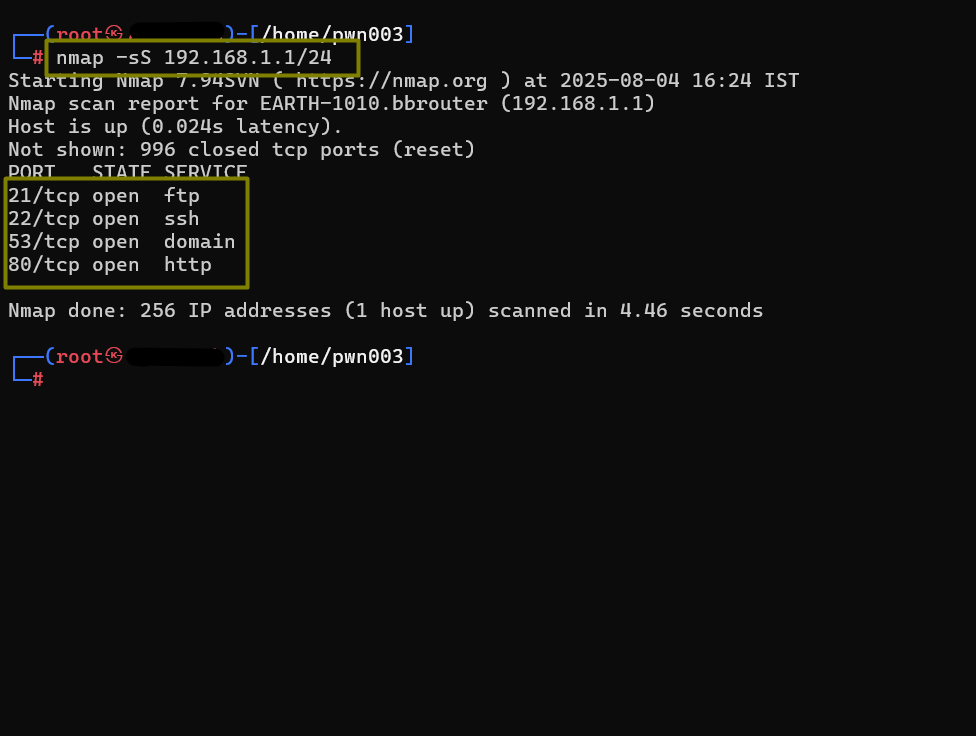


* Determined the subnet as /24, making the scan range: 192.168.1.0/24

## 3. Performed TCP SYN Scan

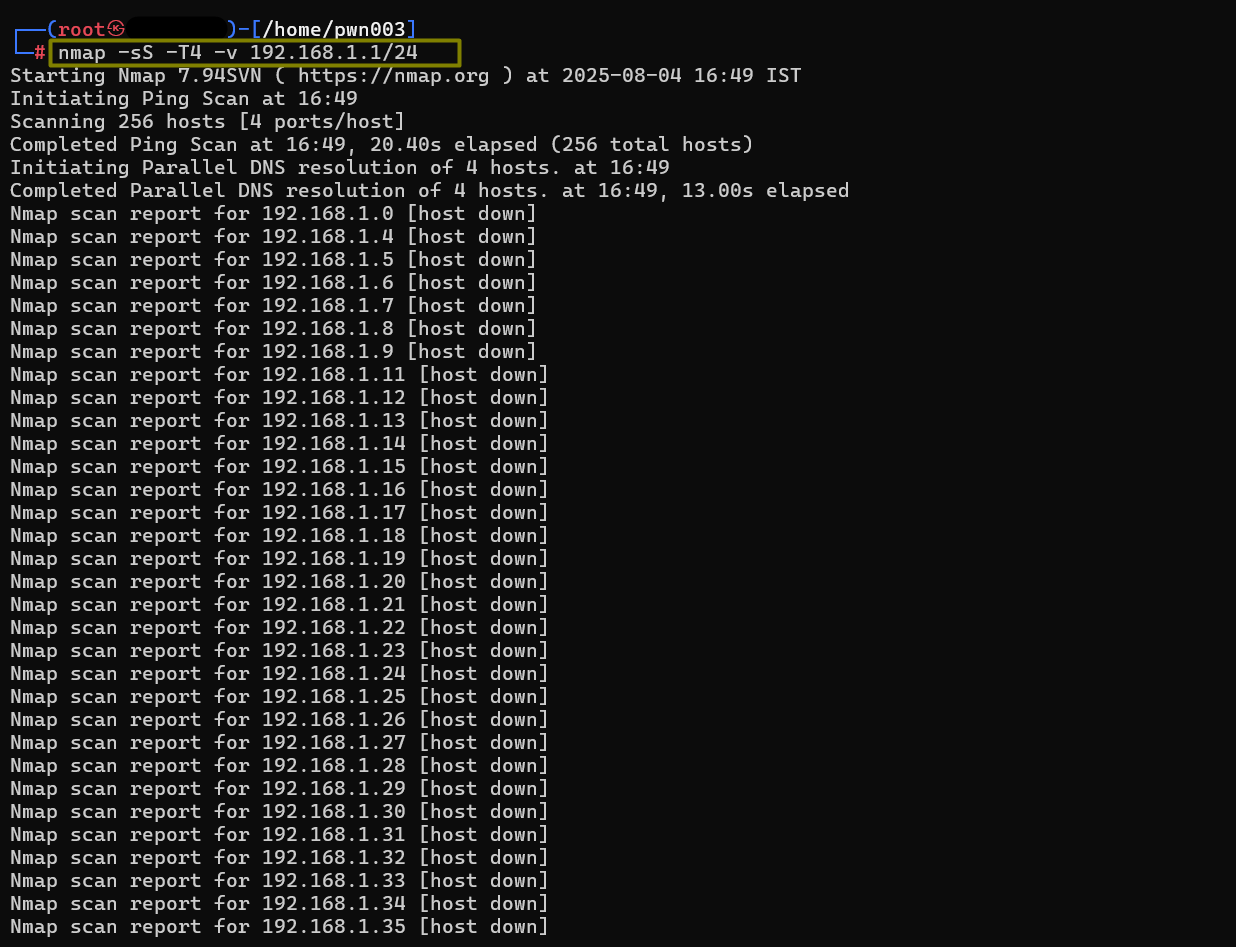
* Ran the following command:

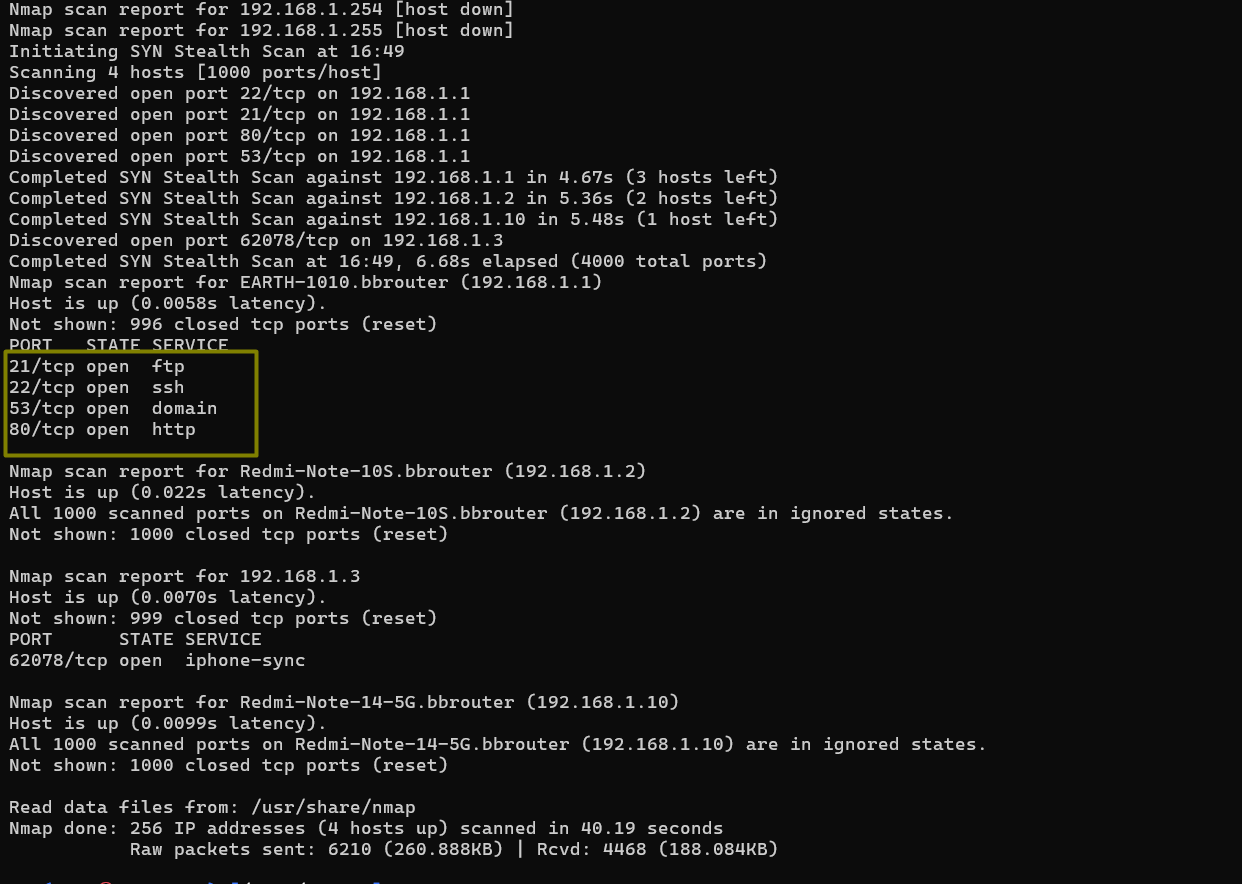
nmap -sS 192.168.1.0/24



* Purpose: Identify active hosts and their open TCP ports.
* Scan was executed with additional verbosity for better tracking:

nmap -sS -T4 -v 192.168.1.0/24





## 4. Observations from Scan

The scan was conducted on the local IP 192.168.1.1. The following open ports and services were identified:

### Host: 192.168.1.1

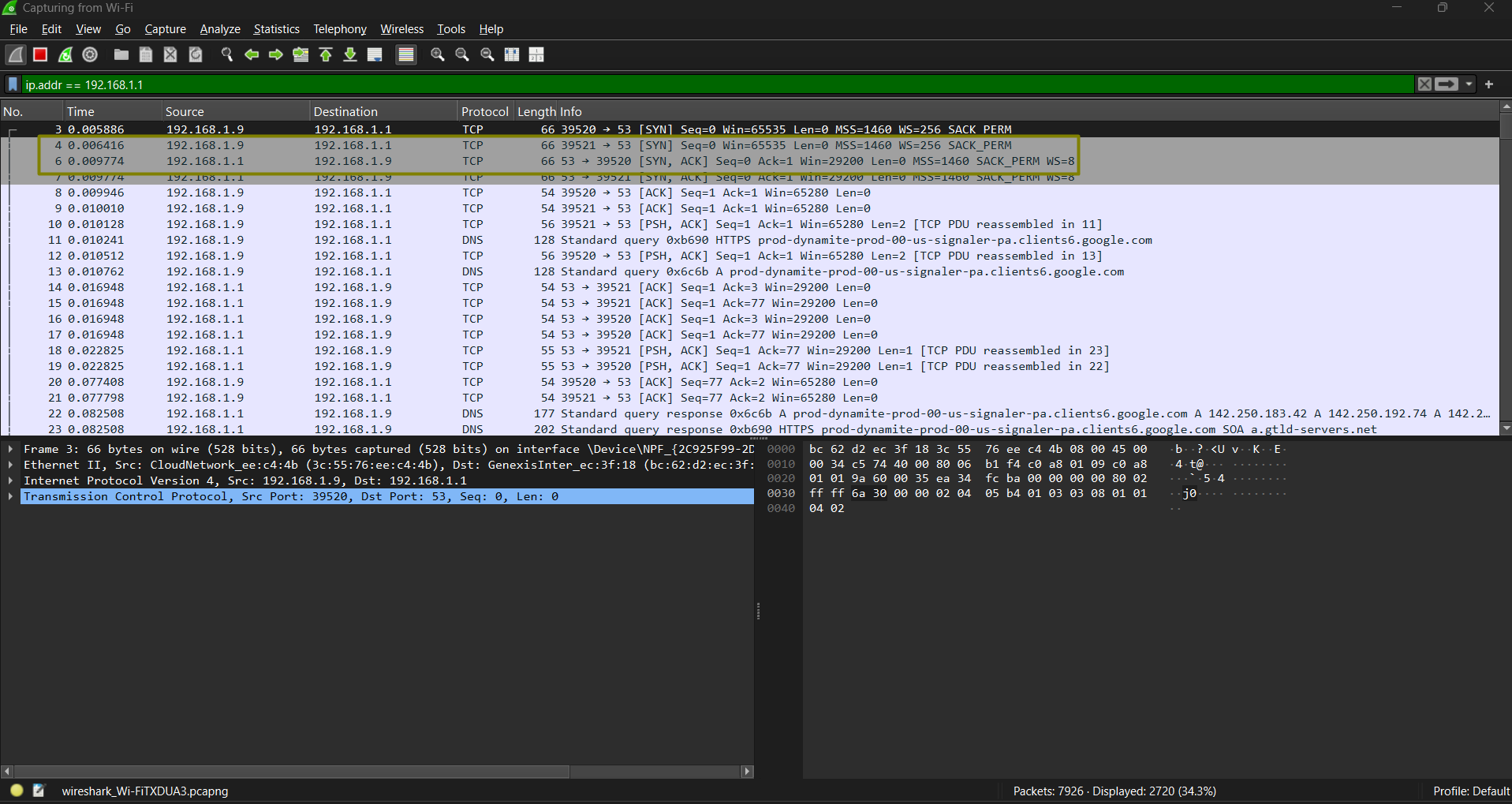
| Port | State | Service |
| --- | --- | --- |
| 21 | open | FTP |
| 22 | open | SSH |
| 53 | open | DNS |
| 80 | open | HTTP |

Results were saved as:

nmap -sS 192.168.1.1/24 -oN scan\_results.txt

## 5. Optional Packet Analysis with Wireshark

* Captured network packets during the Nmap scan.
* Observed SYN packets and ACK/RST responses for open/closed ports.
* Applied filters such as ip.addr == 192.168.1.1 to isolate traffic.



## 6. Research on Common Vulnerabilities

After identifying the open ports, research was conducted to understand possible vulnerabilities associated with each service. The following sections describe these potential issues and how they could be mitigated.

### FTP (Port 21)

* **Possible Vulnerabilities**:
  + Transmission of credentials in plaintext, exposing them to interception.
  + Anonymous login could allow unauthorized access to the server.
  + Older FTP servers may be susceptible to command injection and buffer overflows.
* **Example Vulnerabilities**:
  + CVE-1999-0911 (Buffer overflow)
  + CVE-2001-0053 (Command injection)
* **Mitigation**:
  + Use FTPS or SFTP instead of FTP.
  + Disable anonymous login.
  + Regularly update and patch the FTP server.
  + Restrict access via firewall rules.

### SSH (Port 22)

* **Possible Vulnerabilities**:
  + Brute-force attacks due to weak credentials.
  + Use of outdated SSH versions that may have known exploits.
* **Example Vulnerabilities**:
  + CVE-2001-0144 (CRC32 attack)
  + CVE-2018-15473 (User enumeration)
* **Mitigation**:
  + Disable root login.
  + Implement key-based authentication.
  + Use rate limiting tools like fail2ban.
  + Monitor and audit SSH logs.

### DNS (Port 53)

* **Possible Vulnerabilities**:
  + Open resolvers could be exploited in amplification attacks.
  + Cache poisoning could redirect users to malicious domains.
* **Example Vulnerabilities**:
  + CVE-2008-1447 (Cache poisoning)
  + CVE-2020-8616 (DoS via malformed queries)
* **Mitigation**:
  + Restrict DNS queries to trusted sources.
  + Enable DNSSEC.
  + Regularly patch DNS software.
  + Monitor DNS traffic for anomalies.

### HTTP (Port 80)

* **Possible Vulnerabilities**:
  + Unencrypted communication is susceptible to MITM attacks.
  + Web applications may be exposed to XSS, CSRF, and directory listing vulnerabilities.
* **Example Vulnerabilities**:
  + CVE-2017-5638 (Apache Struts RCE)
  + CVE-2021-41773 (Path traversal in Apache)
* **Mitigation**:
  + Redirect HTTP traffic to HTTPS.
  + Implement secure HTTP headers.
  + Disable directory listing.
  + Patch the web server regularly.

## 7. Security Risk Summary

Based on the scan and research, the following **possible risks** were identified for host 192.168.1.1. These risks are not confirmed but are commonly associated with the identified open ports and services.

|  |  |  |
| --- | --- | --- |
| **Host IP** | **Port** | **Possible Risk Summary** |
| 192.168.1.1 | 21 | Unencrypted FTP, potential credential exposure |
| 192.168.1.1 | 22 | Brute-force attack potential |
| 192.168.1.1 | 53 | DNS amplification or cache poisoning possibilities |
| 192.168.1.1 | 80 | Unsecured HTTP, potential XSS or MITM risk |

## 8. Conclusion

This activity provided practical experience in network reconnaissance using Nmap. The scan on IP 192.168.1.1 helped identify open services and allowed a focused security risk analysis. Remediation steps can now be planned to secure these services and minimize the attack surface.