CSCI 3500 -- Fall 2025

Ethan Walton

October 6, 2025

Instructor: Ms. Emond

# Network Scanning with Nmap

## Lab Purpose (In Your Own Words)

This lab exercise was designed to introduce students to network reconnaissance and vulnerability assessment using Nmap, one of the most widely-used network scanning tools in cybersecurity. The primary purpose was to learn how to identify active hosts on a network, enumerate open ports and services, and analyze the information gathered to assess potential security vulnerabilities. The lab provided hands-on experience with different types of Nmap scans including host discovery, TCP port scanning, UDP port scanning, and service version detection. Through this practical exercise, students gained essential skills in network mapping and vulnerability identification that are fundamental to both offensive and defensive cybersecurity practices.

## Tools & Resources Required

The following tools and resources were utilized to complete this laboratory exercise:

**Metasploitable VM:** A deliberately vulnerable Linux virtual machine created by Rapid7 for security training and testing purposes. This system serves as the target for our scanning activities and contains numerous intentionally vulnerable services.

**Kali Linux VM:** A Debian-based Linux distribution specifically designed for penetration testing and security auditing. Kali includes Nmap and numerous other security tools pre-installed.

**VMware Workstation:** Virtualization platform used to run both virtual machines in an isolated network environment.

**Nmap (Network Mapper):** The primary tool used for network discovery and security auditing. Nmap is capable of discovering hosts and services on a computer network by sending packets and analyzing the responses.

**Virtual Network Configuration:** Both VMs were configured to use a custom host-only network to ensure isolation and proper communication between the attacker (Kali) and target (Metasploitable) systems.

## Lab Procedures

### Step 1: Environment Setup

1. Downloaded the Metasploitable VM from Rapid7's official website and extracted the files to a working directory.

2. Opened the Metasploitable VM in VMware Workstation. The system presented a command-line interface upon successful boot.

3. Logged into Metasploitable using the default credentials (username: msfadmin, password: msfadmin).

### Step 2: Network Configuration

4. Powered down both virtual machines to modify network settings.

5. Accessed VMware's Virtual Network Editor from the Edit menu.

6. Created a new custom network (VMnet) for host-only communication between the VMs.

7. Applied the network settings and configured both VMs to use the new custom network adapter.

8. Started both virtual machines and verified network connectivity.

### Step 3: IP Address Discovery

9. On the Metasploitable VM, executed `ifconfig` command to obtain the system's IP address. The results showed the target IP as 192.168.93.129.

10. On the Kali Linux VM, used `ip addr show` command to identify the network interface and IP range. The Kali system was assigned IP 192.168.93.128/24.

### Step 4: Network Discovery Scan

11. Performed a ping sweep to discover active hosts on the network using the command: `nmap -sn 192.168.93.0/24`. This scan successfully identified both the Kali system (192.168.93.128) and the Metasploitable target (192.168.93.129), confirming proper network connectivity.

### Step 5: Basic TCP Port Scan

12. Executed a basic Nmap scan against the Metasploitable target using: `nmap 192.168.93.129`. This scan revealed numerous open TCP ports including:

- Port 21/tcp (FTP)

- Port 22/tcp (SSH)

- Port 23/tcp (Telnet)

- Port 25/tcp (SMTP)

- Port 53/tcp (DNS)

- Port 80/tcp (HTTP)

- Port 111/tcp (RPC)

- Port 139/tcp (NetBIOS-SSN)

- Port 445/tcp (Microsoft-DS)

- Port 512/tcp (exec)

- Port 513/tcp (login)

- Port 514/tcp (shell)

- Port 1099/tcp (RMI Registry)

- Port 1524/tcp (bindshell)

- Port 2049/tcp (NFS)

- Port 2121/tcp (FTP)

- Port 3306/tcp (MySQL)

- Port 5432/tcp (PostgreSQL)

- Port 5900/tcp (VNC)

- Port 6000/tcp (X11)

- Port 6667/tcp (IRC)

- Port 8009/tcp (AJP13)

- Port 8180/tcp (HTTP)

### Step 6: UDP Port Scan

13. Performed a UDP scan using the fast scan option: `nmap -F -sU 192.168.93.129`. This scan identified several open UDP ports including:

- Port 53/udp (DNS)

- Port 69/udp (TFTP)

- Port 111/udp (RPC)

- Port 137/udp (NetBIOS Name Service)

- Port 138/udp (NetBIOS Datagram Service)

- Port 2049/udp (NFS)

### Step 7: Service Version Detection

14. Conducted a service version scan to identify specific software versions: `nmap -sV 192.168.93.129`. This comprehensive scan provided detailed information about running services:  
 - Apache httpd 2.2.8 (Ubuntu)  
 - Samba smbd 3.X - 4.X  
 - ProFTPD 1.3.1  
 - MySQL 5.0.51a-3ubuntu5  
 - PostgreSQL DB 8.3.0 - 8.3.7  
 - OpenSSH 4.7p1 Debian 8ubuntu1  
 - And numerous other services with version information

## Observations

This laboratory exercise provided valuable insights into network reconnaissance techniques and the vulnerability landscape of poorly configured systems. The Metasploitable VM, as designed, presented an extensive attack surface with 23 open TCP ports and 6 open UDP ports, representing a significantly vulnerable system that would be easily compromised in a real-world scenario.  
  
The most striking observation was the sheer number of unnecessarily exposed services. In a properly hardened system, most of these ports should be closed or filtered. Services such as Telnet (port 23), rlogin (port 513), and rsh (port 514) represent legacy protocols with inherent security weaknesses that should never be enabled in modern environments. The presence of a bindshell on port 1524 is particularly concerning as it provides direct command execution capabilities to remote attackers.  
  
The service version detection scan revealed several outdated software components with known vulnerabilities. Apache HTTP Server 2.2.8, released in 2008, contains multiple security flaws including cross-site scripting vulnerabilities and information disclosure issues. Similarly, ProFTPD 1.3.1 has documented SQL injection vulnerabilities that allow remote attackers to bypass authentication and execute arbitrary SQL commands.  
  
The scanning process demonstrated the effectiveness of Nmap's different scan types. The basic TCP scan provided rapid identification of open ports, while the service version scan offered crucial details about software versions necessary for vulnerability assessment. The UDP scan, though slower, revealed additional attack vectors that might be missed in TCP-only reconnaissance.  
  
From a defensive perspective, this exercise highlighted the critical importance of proper network segmentation, service hardening, and regular vulnerability assessments. The ability to gather such comprehensive information about a target system in minutes underscores the need for robust security controls and continuous monitoring.

## Reflection Questions

### 1. When scanning the Metasploitable VM, what types of information did Nmap reveal about the services running on the machine? Why is knowing the version of services important for penetration testing?

Nmap revealed extensive information about the Metasploitable system including open ports, service names, specific software versions, and operating system details. The scans identified 23 open TCP ports and 6 UDP ports running various services such as Apache HTTP Server 2.2.8, Samba smbd, ProFTPD 1.3.1, MySQL 5.0.51a, and PostgreSQL 8.3.0. Service version information is crucial for penetration testing because it enables security professionals to identify specific vulnerabilities associated with particular software versions. Each version of software may contain unique security flaws that can be exploited using known attack vectors. Without version information, penetration testers would need to perform blind attacks or educated guessing, which is far less efficient and effective than targeted exploitation based on documented vulnerabilities.

### 2. Compare and contrast the differences between a basic Nmap scan and a service version scan (-sV). In what situations might you use one over the other?

A basic Nmap scan performs rapid port enumeration by sending TCP SYN packets to determine which ports are open, closed, or filtered. This scan provides fundamental information about available services but lacks detailed version information. In contrast, the service version scan (-sV) performs additional probing to identify specific software versions and service details, requiring more time and generating additional network traffic. Basic scans are preferable when speed and stealth are priorities, such as during initial reconnaissance of large networks or when attempting to avoid detection by security systems. Service version scans are more appropriate when conducting thorough vulnerability assessments, penetration testing, or when detailed service information is required for exploitation planning. The choice depends on the specific objectives, time constraints, and stealth requirements of the security assessment.

### 3. Why is it important to identify both TCP and UDP open ports when scanning a target? What kinds of vulnerabilities are typically associated with open ports?

Identifying both TCP and UDP ports is essential because different services utilize different transport protocols, and limiting scans to only TCP would provide an incomplete picture of the attack surface. Many critical services operate on UDP, including DNS (port 53), DHCP (port 67/68), SNMP (port 161), and TFTP (port 69). Vulnerabilities associated with open ports include unauthorized service access, information disclosure, denial of service attacks, and remote code execution. TCP services are often vulnerable to connection-based attacks, buffer overflows, and authentication bypasses, while UDP services may be susceptible to amplification attacks, spoofing, and information leakage. Each open port represents a potential entry point for attackers, making comprehensive port scanning critical for accurate security assessment and risk management.

### 4. Discuss the ethical considerations involved in using tools like Nmap for network scanning. How would you ensure you're using these tools responsibly in a real-world environment?

The use of network scanning tools like Nmap raises significant ethical and legal considerations that must be carefully addressed. Scanning networks without explicit authorization can be considered reconnaissance for malicious purposes and may violate computer fraud and abuse laws in many jurisdictions. Responsible use requires obtaining proper written authorization before conducting any scans, clearly defining the scope and limitations of testing activities, and ensuring compliance with applicable laws and organizational policies. In professional environments, security practitioners should follow established penetration testing methodologies, document all activities, limit scans to authorized targets and timeframes, and immediately report any discovered vulnerabilities through appropriate channels. Additionally, practitioners should minimize the impact on production systems, use appropriate scan intensities to avoid service disruption, and maintain confidentiality of discovered vulnerabilities until they can be properly remediated.

### 5. Pick one of the services found in your scans and provide an overview of historical exploits for it.

#### Samba SMB Service Analysis

The Samba service identified during our scan represents one of the most critically vulnerable services on the Metasploitable system. Samba, which provides SMB/CIFS file sharing capabilities for Unix-like systems, has a extensive history of serious security vulnerabilities that have been actively exploited by attackers and malware.  
  
One of the most significant recent vulnerabilities is CVE-2021-44142, a critical out-of-bounds vulnerability affecting Samba versions prior to 4.13.17. This vulnerability carries a CVSS score of 9.9 and allows remote code execution on systems running Samba servers with vulnerable configurations. The vulnerability resides in the vfs\_fruit module and can be exploited when attackers upload files to Samba shares, requiring only write permissions which may be available to unauthenticated users in some configurations.  
  
Historically, Samba has been targeted by several major exploitation campaigns. The EternalRed or SambaCry vulnerability (CVE-2017-7494) affected all Samba versions since 3.5.0 and allowed remote code execution through malicious shared library uploads. This vulnerability was actively exploited by ransomware families including NamPoHyu and various cryptocurrency mining malware.  
  
Another significant vulnerability was the Badlock flaw (CVE-2016-2118), which affected both Windows and Samba systems and allowed man-in-the-middle attacks against the SAMR and LSAD protocols. Earlier versions of Samba, such as those in the 3.6.3 series and lower, contained critical vulnerabilities that allowed unauthorized users to gain root access through anonymous connections by exploiting remote procedure call mechanisms.  
  
The persistence of Samba vulnerabilities in enterprise and consumer devices makes it a particularly attractive target for attackers. Many network-attached storage (NAS) devices include Samba implementations that may not be regularly updated, creating long-term exposure to known vulnerabilities. The widespread deployment of Samba across Linux distributions, IoT devices, and embedded systems means that successful Samba exploits can provide attackers with access to file servers, network infrastructure, and critical business systems.

## ScreenshotsA screenshot of a computer AI-generated content may be incorrect.A screenshot of a computer AI-generated content may be incorrect.