

Windows Server Enumeration with Nmap

Project Overview

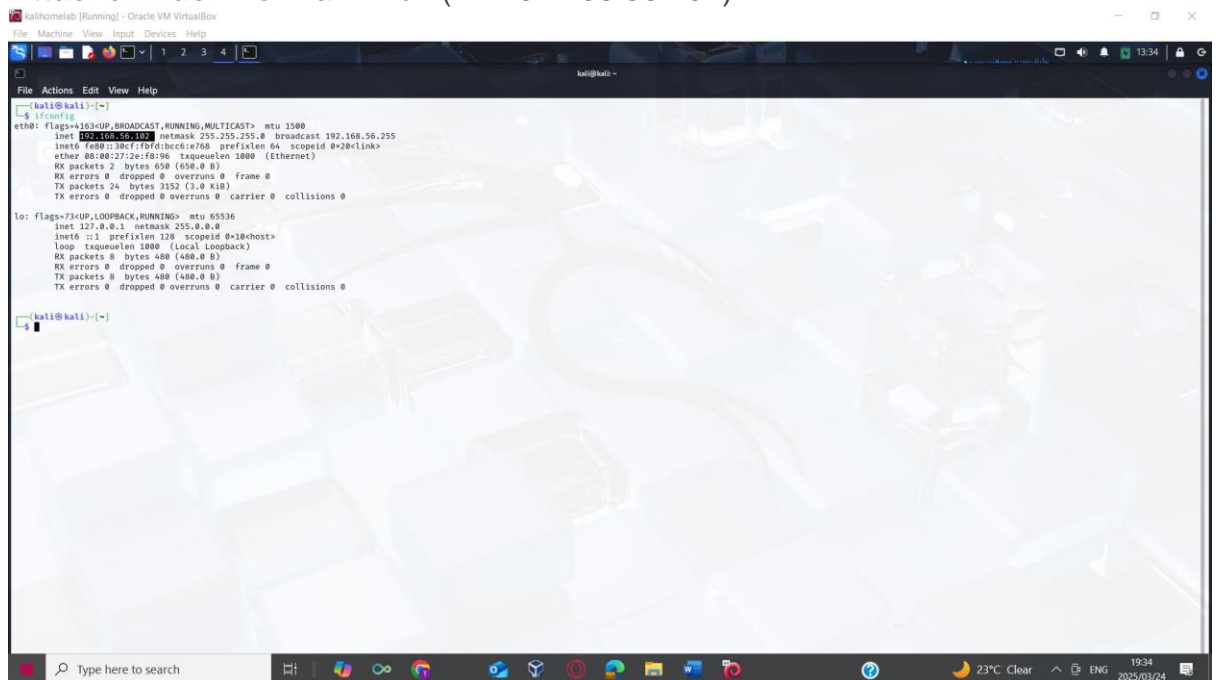
This lab focused on performing a comprehensive security assessment of a newly installed **Windows Server VM** using **Nmap**, the industry-standard network scanning tool. The goal was to identify open services, system information, and potential security vulnerabilities before hardening the server for production use.

Lab Objectives

1. Perform host discovery on the local network
2. Identify open ports and running services
3. Determine operating system and version information
4. Enumerate SMB shares and users
5. Identify security misconfigurations

Environment Setup

- **Attacker Machine:** Kali Linux (IP: 192.168.56.102)

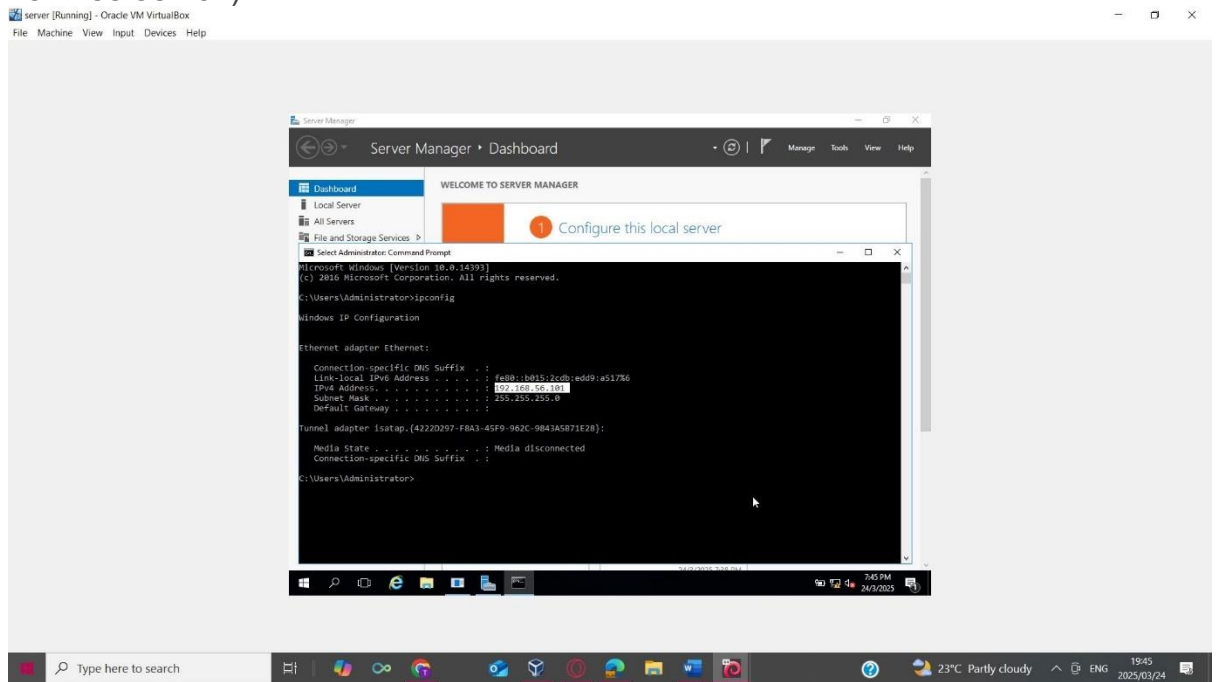


```
(kali@kali)~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.102 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::a0c7:f0d1:bcc6:175b prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:2e:f8:96 txqueuelen 1000 (Ethernet)
    RX packets 2 bytes 650 (650.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 24 bytes 3152 (3.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 400 (400.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 400 (400.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kali@kali)~$
```

- **Target Machine:** Fresh Windows Server 2022 installation (IP: 192.168.56.101)



- **Network:** Private 192.168.56.0/24 subnet

Methodology & Findings

1. Network Discovery

nmap -sn 192.168.56.0/24



The screenshot shows a terminal window titled 'kali@kali:~' with the command 'nmap -sn 192.168.56.0/24' executed. The output displays the results of a ping scan across the 192.168.56.0/24 network. It identifies four active hosts: the default gateway at 192.168.56.1, a DHCP server at 192.168.56.100, a Windows Server at 192.168.56.101, and the Kali Linux attacker machine at 192.168.56.102. The scan was completed in 27.86 seconds.


```
[kali@kali:~]$ nmap -sn 192.168.56.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-25 03:30 EDT
Nmap scan report for 192.168.56.1
Host is up (0.00054s latency).
MAC Address: 0A:00:27:00:00:16 (Unknown)
Nmap scan report for 192.168.56.100
Host is up (0.0015s latency).
MAC Address: 08:00:27:5C:AD:2C (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.101
Host is up (0.0012s latency).
MAC Address: 08:00:27:1B:52:62 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.102
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 27.86 seconds
[kali@kali:~]$
```

Finding: Identified 4 active hosts including:

- Default gateway/Host machine virtual adapter(192.168.56.1)
- DHCP Server (192.168.56.100)
- The target Windows Server (192.168.56.101)
- The Kali Linux Attacker Machine (192.168.56.102)

2. Basic Port Scan

nmap 192.168.56.101



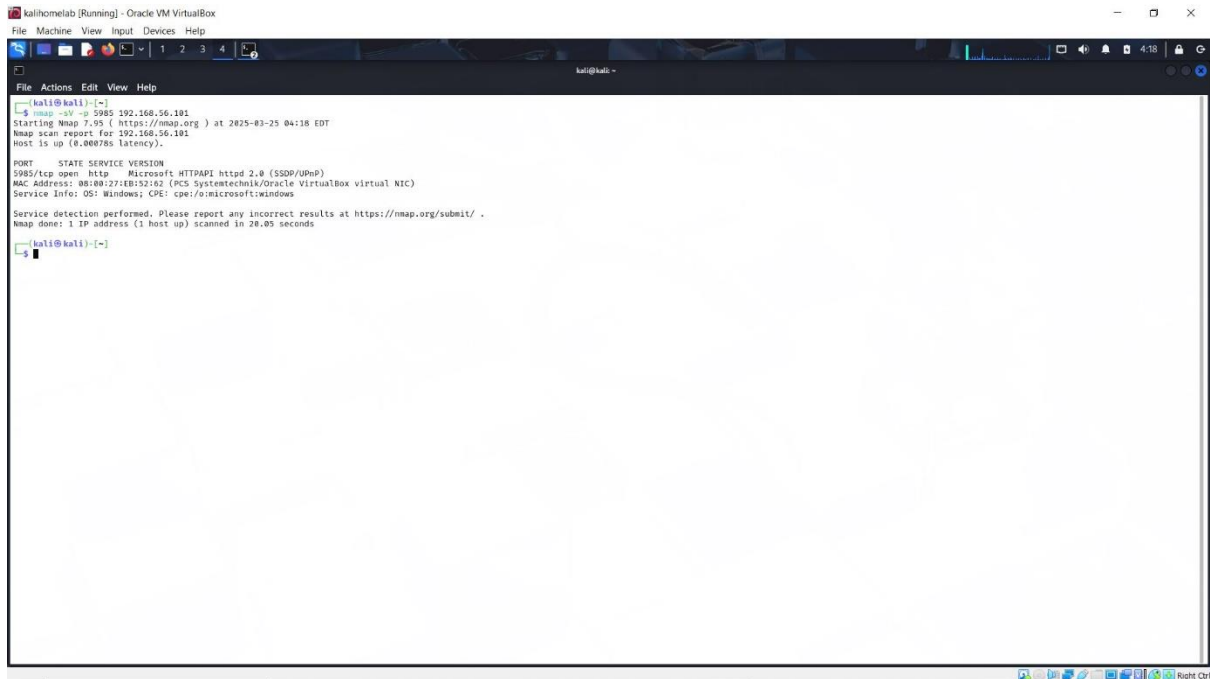
```
kali@kali:~$ nmap -sn 192.168.56.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-25 03:30 EDT
Nmap scan report for 192.168.56.1
Host is up (0.00054s latency).
MAC Address: 08:00:27:00:00:16 (Unknown)
Nmap scan report for 192.168.56.100
Host is up (0.0015s latency).
MAC Address: 08:00:27:5C:AD:2C (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.101
Host is up (0.0012s latency).
MAC Address: 08:00:27:EB:52:62 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.102
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 27.86 seconds
kali@kali:~$
```

Open Ports Found:

- 5985/TCP - WS-Management / WinRM): Windows Remote Management serv

3. Service Enumeration

```
nmap -sV -p 5985 192.168.56.101
```



```
kali@kali:~$ nmap -sV -p 5985 192.168.56.101
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-25 04:18 EDT
Nmap scan report for 192.168.56.101
Host is up (0.00078s latency).

PORT      STATE SERVICE VERSION
5985/tcp  open  http    Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
MAC Address: 08:00:27:EB:52:62 (PC5 Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 20.05 seconds

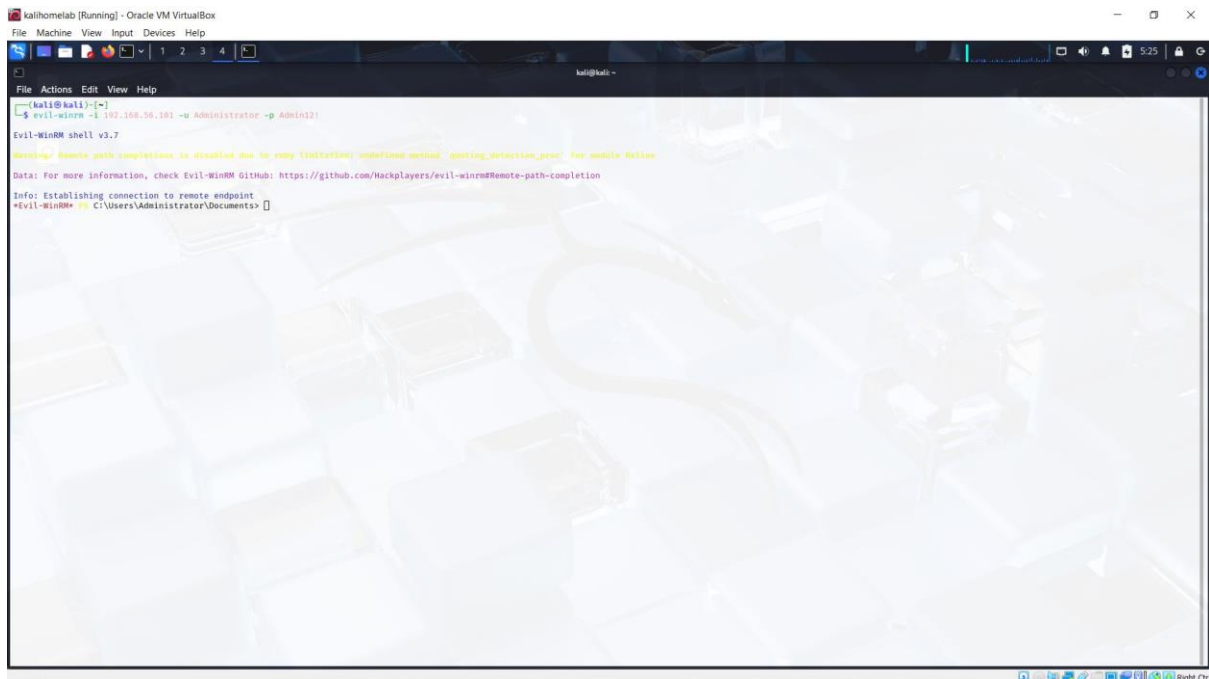
kali@kali:~$
```

Key Findings:

- Service: Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP).
 - ***This service is part of the Windows HTTP Server API, commonly used by services like WinRM (Windows Remote Management) or IIS (Internet Information Services). WinRM (WS-Management) runs on top of HTTP.sys, a Windows kernel-mode driver for HTTP services. Nmap is detecting the underlying HTTP stack, not the WinRM service itself.***
- OS detected: Windows (confirmed by CPE: cpe:/o:microsoft:windows).

4. Exploiting WinRM Vulnerability

```
Evil-winrm -i 192.168.56.101 -u Administrator -p admin12!
```



- The above image shows that I have successfully gained access to the windows server machine as an Administrator.

5. Security Assessment

Critical Findings:

1. **Unsecured WinRM Service:**
 - Port 5985 (WinRM/HTTP) exposed with **no encryption** (HTTPS/5986 not enforced).
 - **Default Administrator credentials** (Administrator:Admin12!) allowed remote access.
2. **Lack of Network Segmentation:**
 - WinRM exposed to the entire subnet (192.168.56.0/24), increasing lateral movement risk.
3. **Outdated HTTPAPI Configuration:**
 - HTTP.sys (kernel HTTP driver) used without security headers or rate-limiting.
4. **Default Administrator Account Active:**
 - High-value target account remained enabled with weak credentials.

Recommendations

1. **Credential Management:**
 - Change default Administrator password to a complex passphrase.
 - Disable or rename the default Administrator account.
2. **WinRM Hardening:**
 - Disable WinRM over HTTP (port 5985).
 - Enforce HTTPS (port 5986) with valid SSL certificates.
 - Restrict WinRM access to specific IPs using firewall rules.
3. **Network Segmentation:**
 - Isolate management interfaces (WinRM) to a separate VLAN.

Configuration Hardening

1. **Enable Windows Defender Firewall:**
`netsh advfirewall set allprofiles state on`
2. **Implement Account Lockout Policy:**
`net accounts /lockoutthreshold:5 /lockoutduration:30`
3. **Disable Unnecessary Services:**
 - Remove or disable HTTPAPI if not required for production.

Ongoing Maintenance

1. **Regular Audits:**
 - Monthly scans with Nmap to detect new open ports.
 - Review WinRM logs for unauthorized access attempts.
2. **Patch Management:**
 - Enable automatic Windows updates for critical security patches.

Lessons Learned

1. Default Credentials Are Critical Risks:

- Weak passwords on high-privilege accounts create immediate attack vectors.

2. Encryption Is Non-Negotiable:

- Unencrypted management protocols (WinRM/HTTP) expose credentials to sniffing.

3. Least Privilege Matters:

- WinRM access should be limited to specific admin users, not the default Administrator.

4. Proactive Monitoring Prevents Breaches:

- Network scans and log reviews could have detected this misconfiguration earlier.

Conclusion

This project demonstrated how a **single open port with weak credentials** can lead to full system compromise. By exploiting WinRM, we gained administrative access to the Windows Server, highlighting the importance of hardening management interfaces.

Skills Demonstrated:

- Network enumeration with Nmap (-sn, -sV).
- Service identification and vulnerability mapping.
- Exploitation of WinRM using Evil-WinRM.
- Security best practices for Windows Server hardening.