Nmap Network Scanning - Lab Documentation

This document contains all nmap scans performed during the network scanning lab, including host discovery, OS fingerprinting,aggressive scanning and SMB enumeration. All results were collected in a controlled learning environment.

1. Nmap version

Nmap 7.94
Verified using *nmap -v*

2. Host Discovery Scan(10.6.6.0/24)

Command - *nmap -sn 10.6.6.0/24*

Purpose:
Detect all active hosts on the local subnet without running port scans

Result:
Nmap identified 7 active hosts out of 256 scanned

Active Hosts Detected

IP Address	Hostname	Status	Latency
10.6.6.1	-	Up	0.011s
10.6.6.11	webgoat.vm	Up	0.0012s
10.6.6.12	juice-shop.vm	Up	0.0010s
10.6.6.13	dvwa.vm	Up	0.00019s
10.6.6.14	multillidae.vm	Up	0.000095s
10.6.6.23	gravemind.vm	Up	0.001s
10.6.6.100	-	Up	0.00013s

3. OS Detection Scan (10.6.6.23)

Command - *sudo nmap -O 10.6.6.23*

Purpose:

Identify the operating system running on the host.

Results
Device Type: General-purpose system
Os Family: Linux
Kernel Range: Linux 4.15 - 5.8

Open Ports Identified

Port	State	Service
21/tcp	open	ftp
22/tcp	open	ssh
53/tcp	open	domain
80/tcp	open	http
139/tcp	open	netbios-ssn
445/tcp	open	microsoft-ds

4. Aggressive Scan on FTP (Port21)

Command - *nmap -p21 -sV -A -T4 10.6.6.23*

Purpose:
Gather detailed service information, version detection, scripts and OS details.

Findings:
- **FTP Service**: vsftpd 3.0.3
- **Annoymous Login**: Allowed(critical security risk)
- **Files accessible via anonymous FTP**:

➔ File1.txt
➔ File2.txt
➔ File3.txt
➔ Supersecretfile.txt

FTP Status Details
- Session timeout: 300 seconds
- No encryption (plain text)
- No bandwidth details
- Server: vsFTPd 3.0.3

Security Concern:
Anonymous FTP access exposes sensitive files to anyone on the network.

5. Aggressive Scan On SMB (Ports 139 & 445)

Command - *nmap -A -p139,445 10.6.6.23*

Purpose:
Enumerate SMB service versions, OS information and server configuration.

Results

- **139/tcp**: Samba smbd 3.X-4.X
- **445/tcp**: Samba smbd 4.9.5-Debian
- **Workgroup**: WORKGROUP
- **Hostname**: GRAVEMIND
- **System Time**: 2025-12-10
- **Message Signing**: Enabled but not required making it vulnerable to MITM attacks.

OS Discovery
Reported OS: Windows 6.1 (via Samba emulation on Debian)

6. SMB Share Enumeration

Command - *nmap --script smb-enum-shares.nse -p445 10.6.6.23*

Purpose:
Identify available SMB shares and permission levels.

Shares Detected

Share	Type	Path	Anonymous Access
IPC$	IPC Hidden	C:\tmp	READ/WRITE
print$	Disk	C:\var\lib\samba\printers	READ/WRITE
workfiles	Disk	C:\var\spool\samba	READ/WRITE

Major Vulnerability:
All shares allow anonymous READ/WRITE access, which should never be enabled in a real environment.

Network Traffic Analysis With Wiresahark

This section documents the steps taken to identify network configuration, capture live packets using tcpdump, and analyze the resulting traffic with Wireshark.

1. Identify the System's IP Address & Network Interfaces

Before capturing traffic, the first step is to determine the machine's IP address and network interface.

Command - *ifconfig*

This displays network interfaces and assigned IP addresses.
Then use ip route to identify the network routes and confirm which interface handles outbound traffic

Command - *ip route*

Output

default via 10.0.2.2 dev eth0 proto dhcp src 10.0.2.15 metric 100
10.0.2.0/24 dev eth0 proto kernel scope link src 10.0.2.15 metric 100
10.5.5.0/24 dev br-339414195aeb proto kernel scope link src 10.5.5.1
10.6.6.0/24 dev br-internal proto kernel scope link src 10.6.6.1
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1
192.168.0.0/24 dev br-355ee7945a88 proto kernel scope link src 192.168.0.1

- **Primary outbound interface**: eth0
- **Local IP address**: 10.0.2.15
- **Default gateway**: 10.0.2.2
- Additional internal networks exist (bridges, Docker,lab VLANs)

2. Identify the System's DNS Resolver

DNS configuration is stored in */etc/resolv.conf*

Command - *cat /etc/resolv.conf*

Output:

 # Generated by NetworkManager
nameserver 10.0.2.3

➔ All DNS queries are forwarded to 10.2.2.3

3. Capture Network Traffic with tcpdump

The packet capture was performed on the eth0 interface

Command - *sudo tcpdump -i eth0 --s 0 -w ladies.pcap*

Explanation of Flags

Flag	Meaning
-i eth0	Listen on the interface eth0
–s 0	Capture full packet
-w ladies.pcap	Save output to ladies.pcap file

➔ Once started, tcpdump captures all packets going through eth0

Capture Summary
15138 packets captured
15138 packets received by filter
0 packets dropped by kernel

- Indicates a successful and clean capture
- No packets were lost/dropped
- All browser activity during this time was captured.

4. Capture Analysis with Wireshark

Once the .pcap file is created;
Command - *wireshark ladies.pcap*

➔ This opens the capture in wireshark.

In Wireshark You Can:

- View packet details (TCP,UDP,HTTP,DNS,ARP, etc)
- Follow TCP streams
- Inspect DNS queries made to 10.0.2.3
- Filter traffic by:
    - ip.addr == 10.0.2.15
    - dns
    - http
    - tcp
    - arp