Task 2 – Password Cracking

Objective

Crack password hashes using dictionary and brute-force attacks to understand weak password vulnerabilities and defensive strategies.

Tools Used

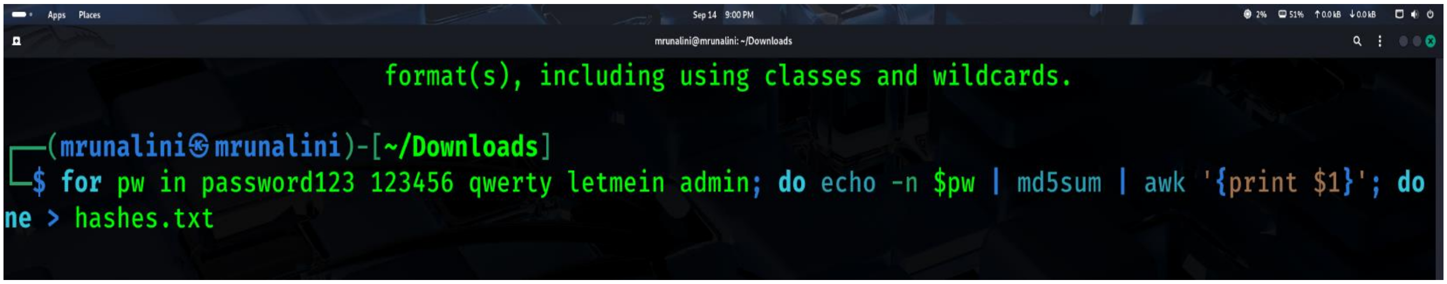
- Hashcat – GPU-accelerated password cracking tool
- John the Ripper – CPU-based password cracking tool
- Kali Linux – Operating system
- rockyou.txt – Wordlist for dictionary attacks

Procedure

Step 1: Prepare Sample Passwords

passwords=("password123" ,"123456" ,"qwerty" ,"letmein" ,"admin")

Step 2 (Automated Hash Generation)



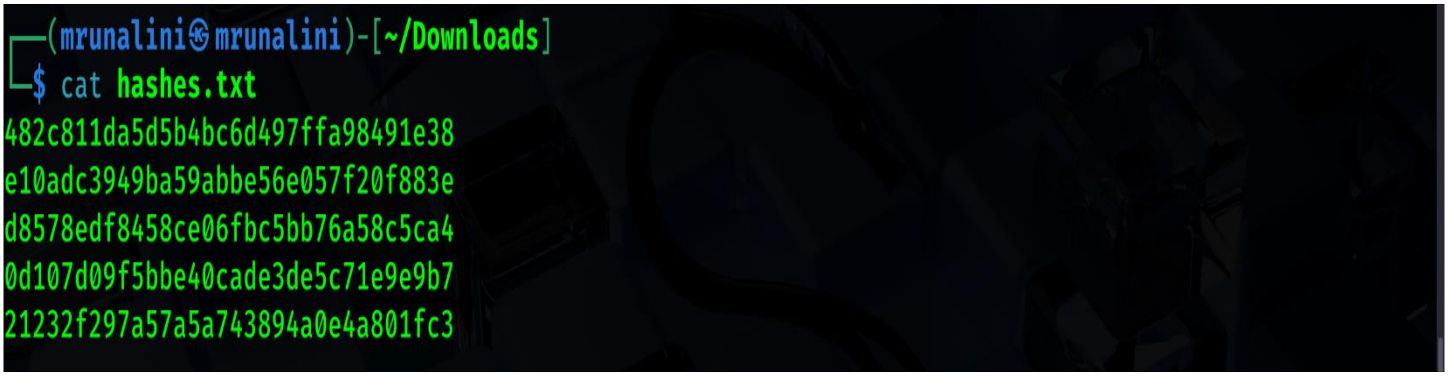
```
                        format(s), including using classes and wildcards.

┌──(mrunalini㉿mrunalini)-[~/Downloads]
└─$ for pw in password123 123456 qwerty letmein admin; do echo -n $pw | md5sum | awk '{print $1}'; do
ne > hashes.txt
```

Explanation

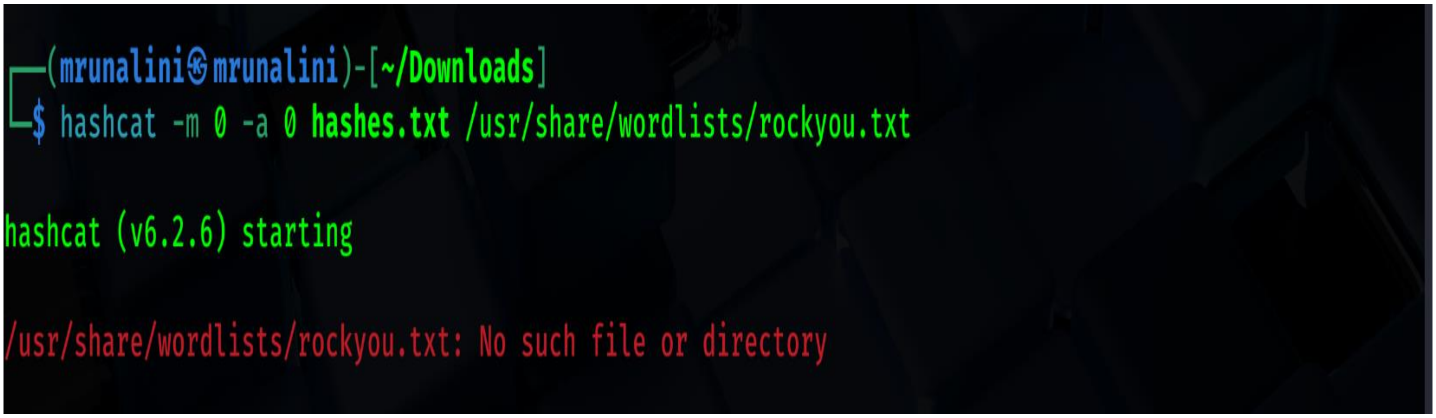
1. for pw in ...; do ... done → Loops through each password in the list.
2. echo -n $pw → Prints the password without a newline.
3. md5sum → Calculates the MD5 hash.
4. awk '{print $1}' → Extracts only the hash (ignores the trailing -).
5. > hashes.txt → Saves all hashes into hashes.txt automatically

Step 3: Verify hashes



```
┌──(mrunalini㉿mrunalini)-[~/Downloads]
└─$ cat hashes.txt
482c811da5d5b4bc6d497ffa98491e38
e10adc3949ba59abbe56e057f20f883e
d8578edf8458ce06fbc5bb76a58c5ca4
0d107d09f5bbe40cade3de5c71e9e9b7
21232f297a57a5a743894a0e4a801fc3
```

Step 4: **Dictionary Attack with Hashcat**



```
┌──(mrunalini㉿mrunalini)-[~/Downloads]
└─$ hashcat -m 0 -a 0 hashes.txt /usr/share/wordlists/rockyou.txt

hashcat (v6.2.6) starting

/usr/share/wordlists/rockyou.txt: No such file or directory
```

Explanation

hashcat -m 0 -a 0 hashes.txt /usr/share/wordlists/rockyou.txt

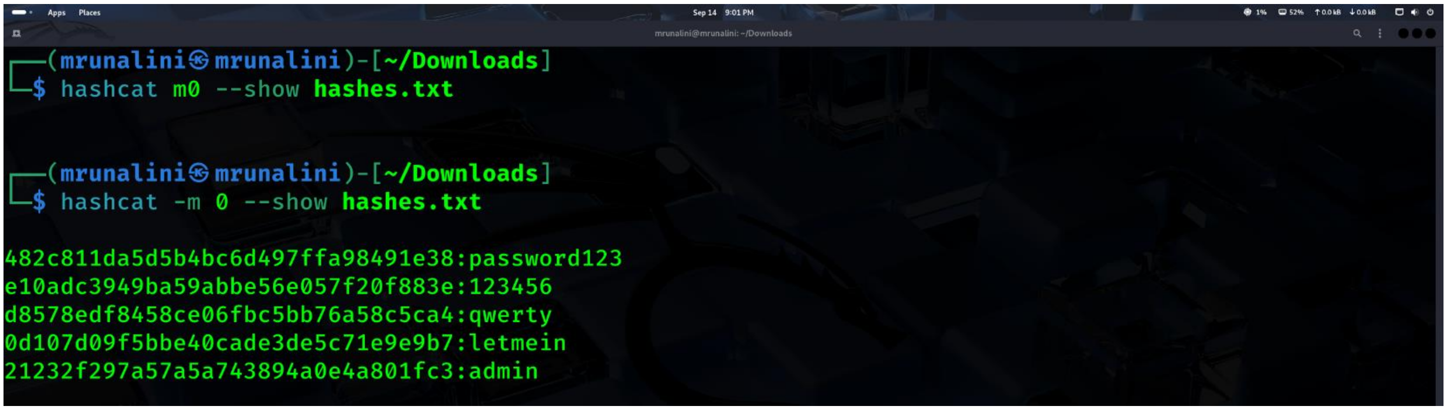
-m 0 → MD5

-a 0 → dictionary attack

hashes.txt → input file

rockyou.txt → dictionary wordlist

Step 5: Check cracked passwords:



```
┌──(mrunalini㉿mrunalini)-[~/Downloads]
└─$ hashcat m0 --show hashes.txt


┌──(mrunalini㉿mrunalini)-[~/Downloads]
└─$ hashcat -m 0 --show hashes.txt

482c811da5d5b4bc6d497ffa98491e38:password123
e10adc3949ba59abbe56e057f20f883e:123456
d8578edf8458ce06fbc5bb76a58c5ca4:qwerty
0d107d09f5bbe40cade3de5c71e9e9b7:letmein
21232f297a57a5a743894a0e4a801fc3:admin
```

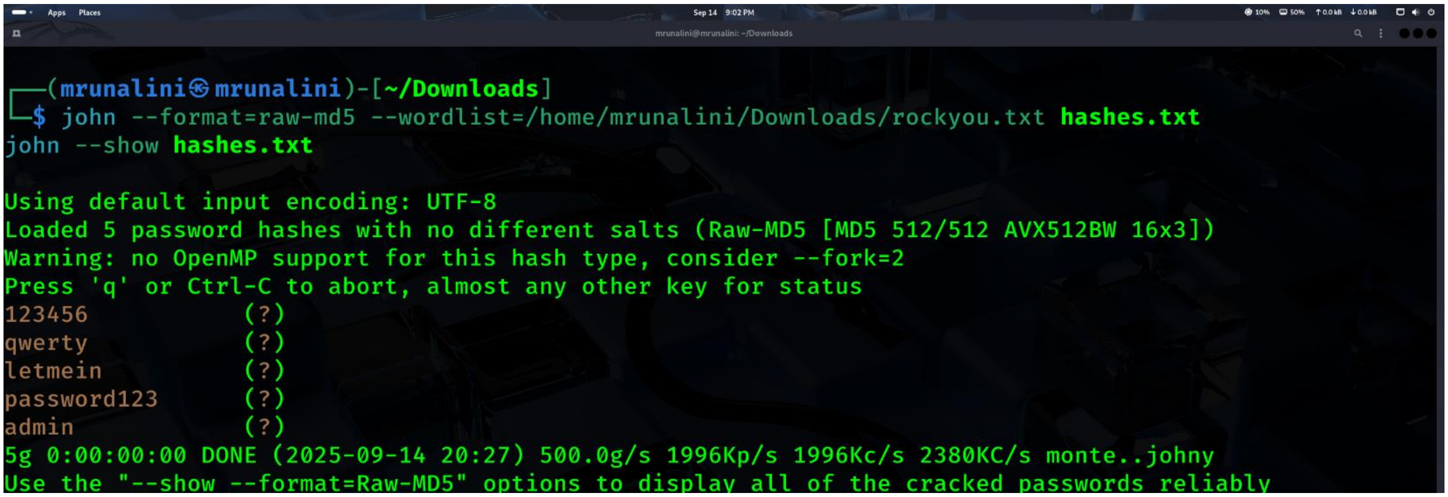
Breakdown of Components

1. **hashcat**
    - Invokes **Hashcat**, a GPU-accelerated password cracking tool.
2. **-m 0**

- Specifies the **hash type**.
- 0 corresponds to **MD5** (raw MD5 hashes).
- Hashcat needs this to interpret the hashes correctly.

3. **--show**
   - Displays all **cracked passwords** that Hashcat has successfully matched with the hashes.
   - Does **not perform a new attack**, only shows results from previous runs.
4. **hashes.txt**
   - Input file containing the **MD5 hashes** you want to reveal.

Step 6:Dictionary Attack with John the Ripper



```
┌──(mrunalini㉿mrunalini)-[~/Downloads]
└─$ john --format=raw-md5 --wordlist=/home/mrunalini/Downloads/rockyou.txt hashes.txt
john --show hashes.txt

Using default input encoding: UTF-8
Loaded 5 password hashes with no different salts (Raw-MD5 [MD5 512/512 AVX512BW 16x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
123456           (?)
qwerty           (?)
letmein          (?)
password123      (?)
admin            (?)
5g 0:00:00:00 DONE (2025-09-14 20:27) 500.0g/s 1996Kp/s 1996Kc/s 2380KC/s monte..johny
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
```

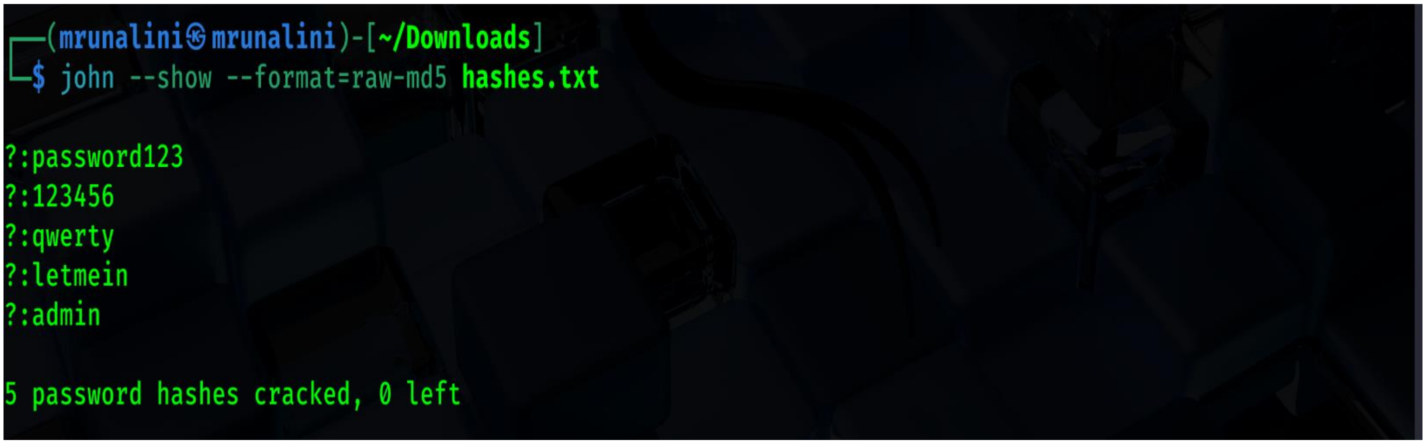
Breakdown

1. **john**
   - Invokes **John the Ripper**, the password cracking tool.
2. **--format=raw-md5**
   - Specifies the **hash type** to crack.

- In this lab, all passwords are **MD5 hashes**, so we must tell John to treat them as raw-md5.
- Without this, John might **not detect the hash type** and fail to crack.

3. **--wordlist=/home/mrunalini/Downloads/rockyou.txt**
   - Uses the **rockyou.txt dictionary** for a **dictionary attack**.
   - John will try each password in this wordlist against the hashes.
4. **hashes.txt**
   - The input file containing the **MD5 hashes** to crack.
   - John reads each line as a hash to attempt cracking.

Step7: Check cracked passwords using john the ripper tool



```
┌──(mrunalini㉿mrunalini)-[~/Downloads]
└─$ john --show --format=raw-md5 hashes.txt

?:password123
?:123456
?:qwerty
?:letmein
?:admin

5 password hashes cracked, 0 left
```

Breakdown of Components

1. john
   - Invokes John the Ripper, the password cracking tool.
2. --show
   - Displays all cracked passwords that John has successfully matched against the hashes.

    - Instead of running another cracking session, it shows results from previous runs.

3. --format=raw-md5
    - Specifies the hash type.
    - In your lab, hashes are MD5, so you must specify this; otherwise, John may fail to interpret the hashes correctly.
4. hashes.txt
    - Input file containing the MD5 hashes you generated earlier

Lab Deliverables

1. Hash List – hashes.txt
2. Cracked Output – from Hashcat and John the Ripper
3. Attack Strategy Explanation
    - Dictionary attack: uses common passwords from rockyou.txt
    - Brute-force attack: tries all combinations

Outcome / Learning

- Weak passwords are easily cracked using dictionary attacks.
- Brute-force attacks highlight the need for long, complex passwords.
- Defensive measures:
    - Strong, unique passwords
    - Salted hashes
    - Account lockouts and rate-limiting