🔐 Password Cracking with Hashcat

🧪 Objective

Demonstrate how to crack a simple password (password123) using **Hashcat** and a dictionary attack with the **RockYou wordlist**.

🖥️ System Requirements

- **OS:** Linux
- **Tools:** hashcat, md5sum, rockyou.txt
- **Wordlist Path:** /usr/share/wordlists/rockyou.txt.gz

📌 Step-by-Step Procedure

✅ Step 1: Generate the MD5 hash of the password

Command:

```
┌──(root㉿kali)-[~]
└─# echo -n "password123" | md5sum

482c811da5d5b4bc6d497ffa98491e38  -
```

✅ Step 2: Save the hash into a file

Command:

```
┌──(root㉿kali)-[~]
└─# echo 482c811da5d5b4bc6d497ffa98491e38 > hash.txt
```

✅ Step 3: Unzip the RockYou wordlist

Command:

```
┌──(root㉿kali)-[~]
└─# gunzip /usr/share/wordlists/rockyou.txt.gz
```

✅ Step 4: Run Hashcat to crack the hash

Command:

```
┌──(root㉿kali)-[~]
└─# hashcat -m 0 -a 0 hash.txt /usr/share/wordlists/rockyou.txt

hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, LLVM 17.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
============================================================================================
* Device #1: cpu-penryn-11th Gen Intel(R) Core(TM) i5-1155G7 @ 2.50GHz, 1274/2613 MB (512 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0×0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Hash
```

```
Host memory required for this attack: 0 MB

Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 2 secs

482c811da5d5b4bc6d497ffa98491e38:password123

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 482c811da5d5b4bc6d497ffa98491e38
Time.Started.....: Mon Apr 14 12:59:44 2025 (0 secs)
Time.Estimated...: Mon Apr 14 12:59:44 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:     7483 H/s (0.07ms) @ Accel:256 Loops:1 Thr:1 Vec:4
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 2048/14344385 (0.01%)
Rejected.........: 0/2048 (0.00%)
Restore.Point....: 1024/14344385 (0.01%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
```

```
Candidate.Engine.: Device Generator
Candidates.#1....: kucing → lovers1
Hardware.Mon.#1..: Util: 10%

Started: Mon Apr 14 12:59:26 2025
Stopped: Mon Apr 14 12:59:46 2025
```

✅ **Step 5: Display the cracked password**

Command:

```
┌──(root㉿kali)-[~]
└─# hashcat -m 0 -a 0 hash.txt /usr/share/wordlists/rockyou.txt --show

482c811da5d5b4bc6d497ffa98491e38:password123
```

🔐 Summary / Conclusion

- Hashcat successfully cracked the MD5 hash of password123 using a dictionary attack.
- This proves how insecure common passwords are.
- Recommendation: Always use complex, unique passwords and avoid dictionary words.