TASK 4: Password Security & Authentication Analysis

1 . Learn how passwords are stored (hashing vs encryption).

Hashing

- One-way process
- Original password cannot be reversed
- Used for password storage

Examples:

- MD5
- SHA-1
- Bcrypt

Encryption

- Two-way process
- Data can be decrypted using a key
- Used for data protection, not passwords

2. Identify different hash types (MD5, SHA-1, bcrypt).

Hash Type	Description
MD5	Fast but insecure
SHA-1	Weak and deprecated
SHA-256	More secure
bcrypt	Slow & highly secure

3. Generate password hashes.

Passwords can be converted into hashes using tools like:

- Hashcat
- John the Ripper
- Online hash generators
- **Example:**

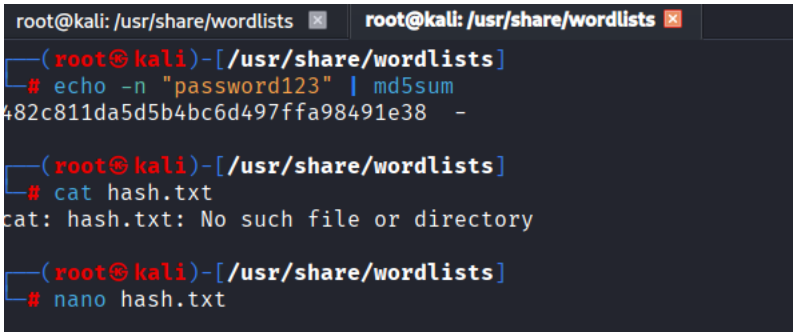
  Password: password123

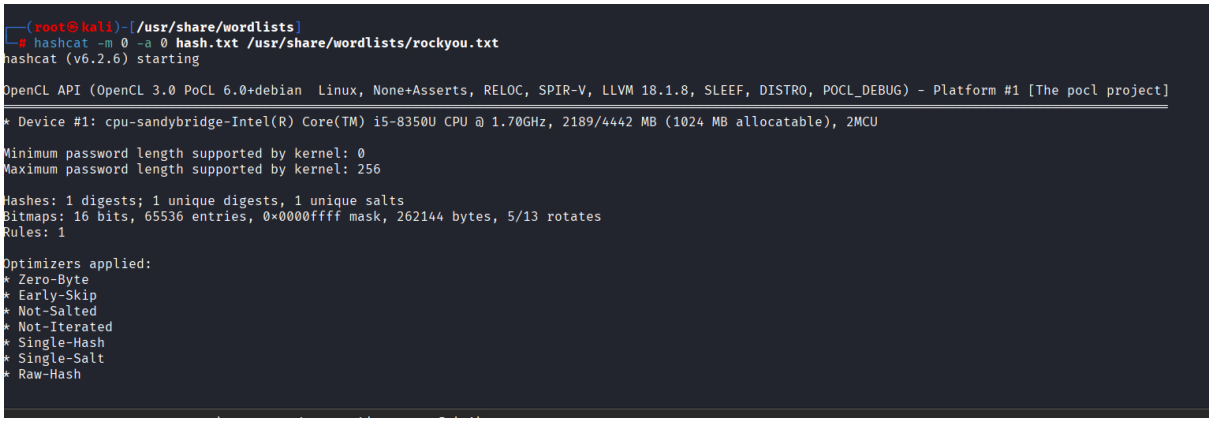
  MD5 Hash: 482c811da5d5b4bc6d497ffa98491e38

4. Attempt cracking weak hashes using wordlists.

❖ **Password**: password123

Hash: 482c811da5d5b4bc6d497ffa98491e38





```
ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 0 MB

Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 1 sec

482c811da5d5b4bc6d497ffa98491e38:password123

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 482c811da5d5b4bc6d497ffa98491e38
Time.Started.....: Tue Jan 20 05:45:10 2026 (0 secs)
Time.Estimated...: Tue Jan 20 05:45:10 2026 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:    31004 H/s (0.18ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 2048/14344385 (0.01%)
```

- **Password:** mypass21253

  Hash: 8c41746ba526748cb1af13519c82a079

```
┌──(root㉿kali)-[/usr/share/wordlists]
└─# echo -n "mypass21253" | md5sum
8c41746ba526748cb1af13519c82a079  -

┌──(root㉿kali)-[/usr/share/wordlists]
└─# nano hash1.txt

┌──(root㉿kali)-[/usr/share/wordlists]
└─# hashcat -m 0 -a 0 hash1.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, SPIR-V, LLVM 18.1.8, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
* Device #1: cpu-sandybridge-Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz, 2189/4442 MB (1024 MB allocatable), 2MCU

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, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

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

5. Understand brute force vs dictionary attacks.

Dictionary Attack

- Uses common password lists
- Faster than brute force
- Works on weak passwords

Brute Force Attack

- Tries all possible combinations
- Time-consuming
- Guaranteed success if no protection

Attack Type	Speed	Effectiveness
Dictionary	Fast	Limited
Brute Force	Slow	Very High

6. Analyze why weak passwords fail.

Weak passwords fail because they can be easily guessed or cracked using automated tools and commonly available wordlists. Even when passwords are hashed, weak password choices remain vulnerable to attacks.

1. Common Patterns in Weak Passwords
2. Presence in Wordlists
3. Short Password Length
4. Lack of Complexity
5. Absence of Multi-Factor Authentication (MFA)

7. Study MFA and its importance.

Multi-Factor Authentication (MFA)

MFA adds extra security by requiring more than one authentication factor.

Factors:

1. Something you know – Password
2. Something you have – OTP / Mobile
3. Something you are – Fingerprint

Importance of MFA:

- Prevents account takeover
- Stops attacks even if password is stolen
- Strong defense against phishing

8.Write recommendations for strong authentication.

✓ Use long passwords (12+ characters)

✓ Mix uppercase, lowercase, numbers, symbols

✓ Avoid common words

✓ Enable MFA

✓ Use password managers

✓ Never reuse passwords