Task 4: Password Security & Authentication Analysis

Tools Used

- Hashcat
- John the Ripper
- Online Hash Identifier / Hash Generator

Theory

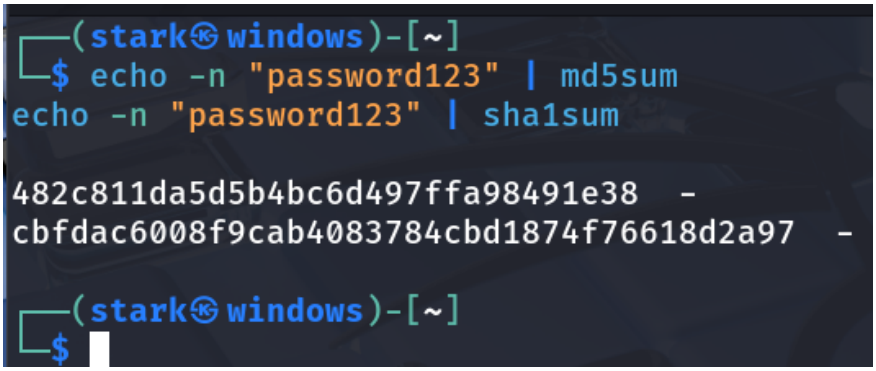
Hashing vs Encryption

Hashing is a one-way process used to store passwords securely, while encryption is a two-way reversible process used for protecting sensitive data.

Experiment / Procedure

1. Identification of Hash Types

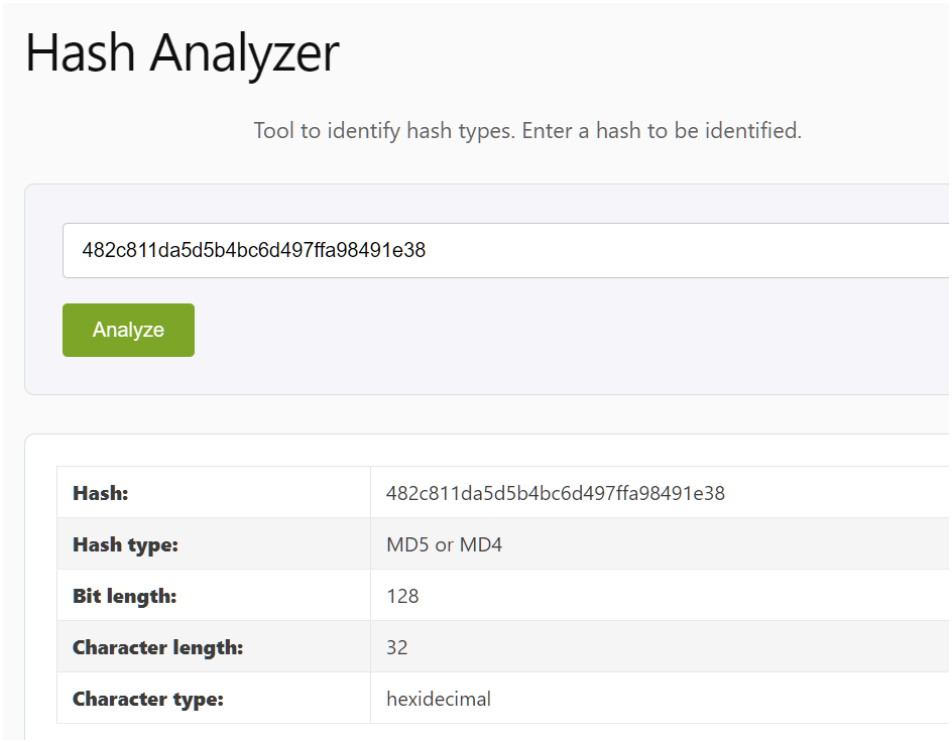
Different hash types such as MD5, SHA-1, and bcrypt were studied and identified using online hash identification tools.



Observed Hash:
482c811da5d5b4bc6d497ffa98491e38

cbfdac6008f9cab4083784cbd1874f76618d2a97

Identified As:



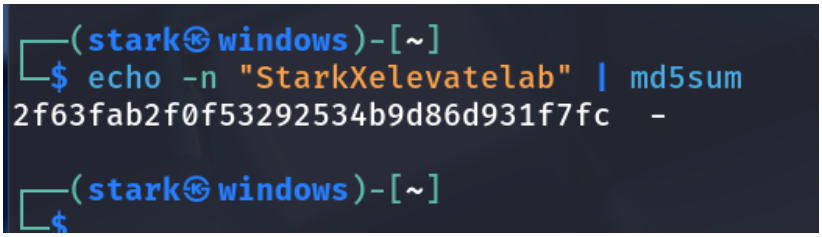
2. Password Hash Generation

Password hashes were generated using Linux commands and online tools.

Command Used:

echo -n "StarkXelevatelab" | md5sum

Generated Hash:



3. Password Cracking Using Hashcat

A dictionary attack was performed using the rockyou.txt wordlist.

Command Used:

hashcat -m 0 hash.txt /usr/share/wordlists/rockyou.txt

Result:

```
┌──(stark㉿windows)-[~]
└─$ hashcat -m 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, SPIR-V, LLVM
EBUG) - Platform #1 [The pocl project]
================================================================================
====================================
* Device #1: cpu-skylake-avx512-11th Gen Intel(R) Core(TM) i5-11320H @ 3.20GHz,
le), 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
```

```
Approaching final keyspace - workload adjusted.

Session..........: hashcat
Status...........: Exhausted
Hash.Mode........: 0 (MD5)
Hash.Target......: 2f63fab2f0f53292534b9d86d931f7fc
Time.Started.....: Tue Jan 20 10:04:44 2026 (8 secs)
Time.Estimated...: Tue Jan 20 10:04:52 2026 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  2469.9 kH/s (0.09ms) @ Accel:256 Loops:1 Thr:1 Vec:16
Recovered........: 0/1 (0.00%) Digests (total), 0/1 (0.00%) Digests (new)
Progress.........: 14344385/14344385 (100.00%)
Rejected.........: 0/14344385 (0.00%)
Restore.Point....: 14344385/14344385 (100.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: $HEX[206b72697374656e616e6e65] -> $HEX[042a0337c2a156616d6f732103]
Hardware.Mon.#1..: Util: 63%
```

4. Password Cracking Using John the Ripper

The same hash was cracked using John the Ripper.

Command Used:

john --format=Raw-MD5 hash.txt --wordlist=/usr/share/wordlists/rockyou.txt

Result:

```
┌──(stark㉿windows)-[~]
└─$ john --format=Raw-MD5 hash.txt --wordlist=/usr/share/wordlists/rockyou.txt

Using default input encoding: UTF-8
Loaded 1 password hash (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
0g 0:00:00:03 DONE (2026-01-20 10:35) 0g/s 4294Kp/s 4294Kc/s 4294KC/s  fuckyooh21..*7¡Vamos!
Session completed.

┌──(stark㉿windows)-[~]
└─$ john --show hash.txt

0 password hashes cracked, 2 left
```

5. Brute Force vs Dictionary Attack

Attack Type	Description
Dictionary	Uses predefined wordlists
Brute Force	Tries all possible combinations

6. Analysis of Weak Passwords

Weak passwords fail because they are predictable, short, and commonly used, making them easy targets for dictionary attacks.

7. Multi-Factor Authentication (MFA)

MFA adds an additional layer of security by requiring more than one authentication factor such as OTPs or biometrics.

Results

- Hash types were successfully identified
- Weak passwords were cracked using dictionary attacks
- Security weaknesses were analyzed

Conclusion

This experiment demonstrated how weak passwords can be easily compromised and highlighted the importance of strong passwords and multi-factor authentication.