🔐 Password Security & Authentication Analysis

Cyber Security Internship – Task 4

1. Introduction

Password security plays a crucial role in protecting user accounts and sensitive data from unauthorized access. Most systems store passwords in a secure form to prevent attackers from directly viewing them. However, weak passwords and poor authentication practices can still lead to security breaches.

This task focuses on understanding how passwords are stored, how weak passwords can be cracked, and how strong authentication methods such as Multi-Factor Authentication (MFA) can prevent attacks.

2. What is Hashing?

Hashing is a one-way cryptographic process used to convert a plain-text password into a fixed-length string called a hash.
Once a password is hashed, it cannot be converted back to its original form.

Key points:

- Same input → same hash
- Small change in password → completely different hash
- Used for secure password storage

3. Difference Between Hashing and Encryption

Hashing	Encryption
One-way process	Two-way process
Cannot be reversed	Can be decrypted using a key
Used for passwords	Used for data protection
Example: MD5, SHA-1	Example: AES, RSA

4. Types of Password Hashes

- MD5
  Fast but weak and vulnerable to dictionary and brute force attacks.
- SHA-1
  Slightly stronger than MD5 but still considered insecure today.
- bcrypt
  Very secure, slow by design, and includes salting, making it resistant to cracking.

5. Password Cracking Techniques

Dictionary Attack

- Uses a list of commonly used passwords
- Very effective against weak passwords
- Faster than brute force

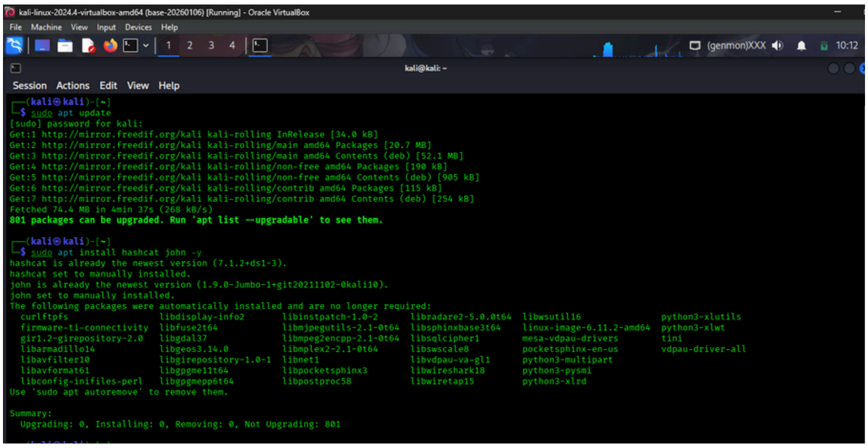
Brute Force Attack

- Tries all possible character combinations
- Time-consuming but powerful
- Works even if password is not in a dictionary

6. Practical Password Hash Analysis

1.Tool Installation

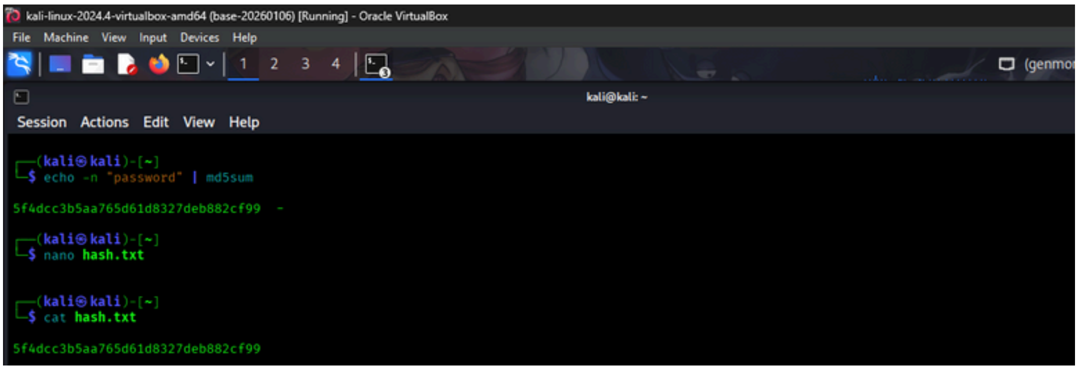
The required tools (Hashcat and John the Ripper) were installed on Kali Linux for password analysis.


kali-linux-2024.4-virtualbox-amd64 (base-20260106) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
1 2 3 4
(genmon)XXX
10:12
kali@kali: ~
Session Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ sudo apt update
[sudo] password for kali:
Get:1 http://mirror.freedif.org/kali kali-rolling InRelease [34.0 kB]
Get:2 http://mirror.freedif.org/kali kali-rolling/main amd64 Packages [20.7 MB]
Get:3 http://mirror.freedif.org/kali kali-rolling/main amd64 Contents (deb) [52.1 MB]
Get:4 http://mirror.freedif.org/kali kali-rolling/non-free amd64 Packages [190 kB]
Get:5 http://mirror.freedif.org/kali kali-rolling/non-free amd64 Contents (deb) [905 kB]
Get:6 http://mirror.freedif.org/kali kali-rolling/contrib amd64 Packages [115 kB]
Get:7 http://mirror.freedif.org/kali kali-rolling/contrib amd64 Contents (deb) [254 kB]
Fetched 74.4 MB in 4min 37s (268 kB/s)
801 packages can be upgraded. Run 'apt list --upgradable' to see them.

┌──(kali㉿kali)-[~]
└─$ sudo apt install hashcat john -y
hashcat is already the newest version (7.1.2+ds1-3).
hashcat set to manually installed.
john is already the newest version (1.9.0-Jumbo-1+git20211102-0kali10).
john set to manually installed.
The following packages were automatically installed and are no longer required:
  curlftpfs                 libdisplay-info2       libinstpatch-1.0-2       libradare2-5.0.0t64  libwsutil16               python3-xlutils
  firmware-ti-connectivity  libfuse2t64            libmjpegutils-2.1-0t64   libsphinxbase3t64    linux-image-6.11.2-amd64  python3-xlwt
  gir1.2-girepository-2.0   libgdal37              libmpeg2encpp-2.1-0t64   libsqlcipher1        mesa-vdpau-drivers        tini
  libarmadillo14            libgeos3.14.0          libmplex2-2.1-0t64       libswscale8          pocketsphinx-en-us        vdpau-driver-all
  libavfilter10             libgirepository-1.0-1  libnet1                  libvdpau-va-gl1      python3-multipart
  libavformat61             libgpgme11t64          libpocketsphinx3         libwireshark18       python3-pysmi
  libconfig-inifiles-perl   libgpgmepp6t64         libpostproc58            libwiretap15         python3-xlrd
Use 'sudo apt autoremove' to remove them.

Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 801
```

2.Password Hash Generation

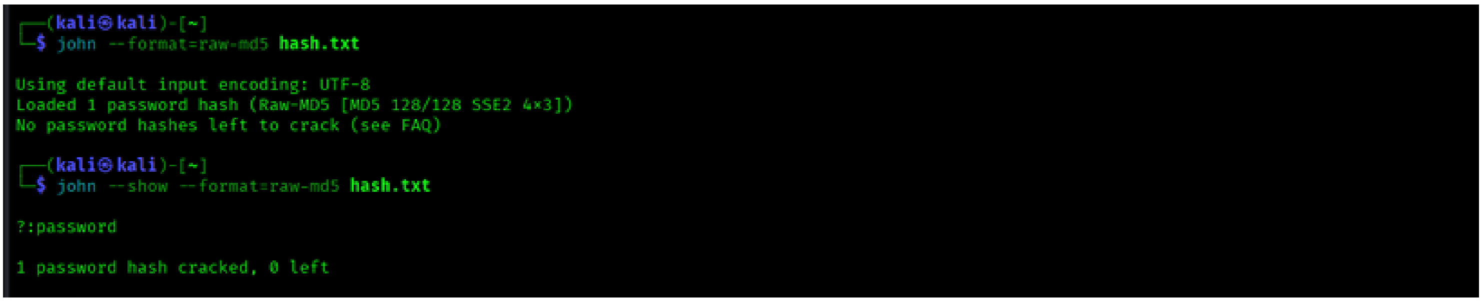
A sample password was converted into an MD5 hash using Linux commands to understand how passwords are stored internally.


kali-linux-2024.4-virtualbox-amd64 (base-20260106) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
1 2 3 4
(genmo
kali@kali: ~
Session Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ echo -n "password" | md5sum

5f4dcc3b5aa765d61d8327deb882cf99  -

┌──(kali㉿kali)-[~]
└─$ nano hash.txt


┌──(kali㉿kali)-[~]
└─$ cat hash.txt

5f4dcc3b5aa765d61d8327deb882cf99
```

3.Cracking Weak Password Hash

The generated hash was cracked using John the Ripper with a dictionary attack. The original password was successfully recovered, proving that weak passwords are insecure.



```
┌──(kali㉿kali)-[~]
└─$ john --format=raw-md5 hash.txt

Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 SSE2 4x3])
No password hashes left to crack (see FAQ)

┌──(kali㉿kali)-[~]
└─$ john --show --format=raw-md5 hash.txt

?:password

1 password hash cracked, 0 left
```

7. Why Weak Passwords Fail

Weak passwords fail due to:

- Short length
- Use of common words
- No symbols or numbers
- Reuse across multiple platforms
- Presence in public wordlists

Attackers can easily crack such passwords using automated tools.

8. Multi-Factor Authentication (MFA) and Its Importance

Multi-Factor Authentication adds an extra layer of security by requiring more than one verification factor.

Examples:

- Password + OTP
- Password + fingerprint
- Password + authenticator app

Even if a password is compromised, MFA prevents unauthorized access, making it a critical security measure.

9. Recommendations for Strong Authentication

- Use passwords with 12–16 characters
- Combine uppercase, lowercase, numbers, and symbols
- Avoid using personal information
- Never reuse passwords
- Use password managers
- Enable MFA on all critical accounts
- Store passwords using bcrypt or salted hashes

10. Conclusion

This task helped in understanding how passwords are stored, how attackers exploit weak passwords, and how strong authentication practices can prevent security breaches. Practical analysis demonstrated the importance of using strong passwords and enabling Multi-Factor Authentication for improved security.