INTERNSHIP REPORT – TASK 4

Password Security s Hash Cracking (Authentication Security)

---

1. Task Title

Password Security: Hashing, Hash Types, and Password Cracking Techniques

---

2. Objective

The main objective of this task is to understand how passwords are securely stored in real systems and why weak passwords are dangerous. In this task, I learned the difference between **hashing and encryption**, studied popular hashing algorithms, generated password hashes, and tested cracking techniques using wordlists. I also learned about **brute force vs dictionary attacks** and understood the importance of **Multi-Factor Authentication (MFA)** for strong security.

---

3. Tools Used

- **John the Ripper** (Password cracking tool)
- **Hashcat** (GPU-based cracking tool)
- **Online hash generators** (for learning purpose)

---

4. Password Security Concepts Learned

4.1 Hashing vs Encryption Hashing

- Converts password into fixed-length hash value
- One-way process (cannot be reversed normally)
- Used for password storage

Encryption

- Converts data into encrypted form
-

4.2 Common Hash Types Studied

Hash Type Nature Security Level MD5 Fast hashing Weak

SHA-1 Fast hashing Weak (collisions exist)

bcrypt Slow + salt Strong

4.3 Salting Concept

- **Salt** = extra random value added before hashing
- Purpose: makes same passwords generate different hashes
- Prevents rainbow table cracking

---

5. Procedure / Steps Followed

Step 1: Understanding Password Storage

- Studied how systems store passwords using **hashing**
- Understood why storing passwords as plain text is unsafe

Step 2: Identifying Hash Types

Learned to identify hash types based on:

- hash length
- hash format Examples:
- MD5 → 32 characters
- SHA-1 → 40 characters
- bcrypt → starts with $2a$ / $2b$

Step 3: Generating Password Hashes

Generated hashes for sample passwords like:

- 123456
- password@123
- Krish@2026

Step 4: Cracking Weak Hashes using Wordlists

- Used wordlist-based cracking approach
- Tested common password hashes
- Observed that simple passwords are cracked easily

Step 5: Understanding Attack Methods Dictionary Attack

- Uses pre-made wordlist
- Fast if password is common

Brute Force Attack

- Tries all possible combinations
- Slow but works eventually for small passwords

Step 6: Learning MFA Importance

Studied MFA types:

- OTP SMS
- Authenticator Apps (Google Authenticator / Microsoft Authenticator)
- Biometrics (Fingerprint / Face)

---

6. Observations / Analysis

6.1 Why Weak Passwords Fail

I observed that weak passwords like:

- 123456

- qwerty

are easily cracked because:

- they exist in common wordlists
- short length reduces combinations
- predictable pattern makes cracking faster

6.2 Hash Algorithm Security Comparison MD5 and SHA-1

- Very fast hashing
- Attackers can crack quickly using wordlists
- Not recommended for password storage

bcrypt

- Slow hashing (intentional)
- Uses salting
- Much harder to crack

6.3 Brute Force vs Dictionary Attack Attack Type Speed Best Used When

Dictionary Fast Password is common/simple Brute Force Slow Password is short or unknown

---

7. Key Learnings

From this task, I learned:

- Passwords should never be stored in plain text
- Hashing is safer than encryption for password storage
- MD5 and SHA-1 are outdated and weak
- bcrypt is secure due to salting and slower speed

- Weak passwords can be cracked using wordlists
- MFA provides extra security even if password leaks

---

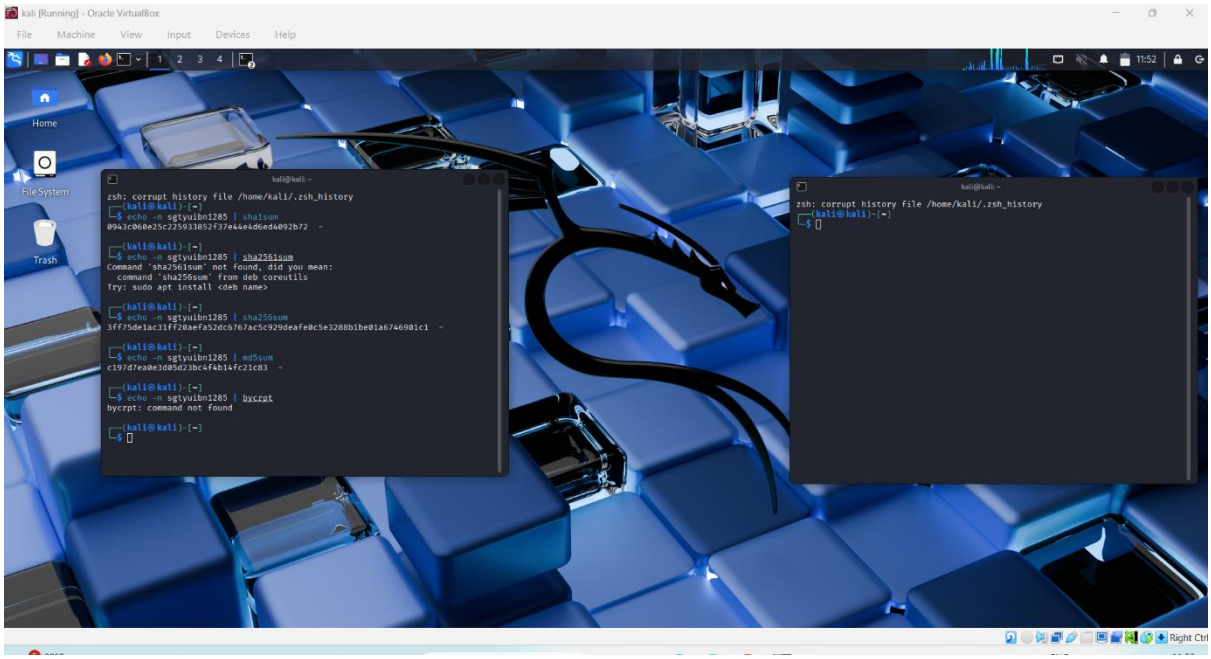
8. Recommendations for Strong Authentication

To improve password security, I recommend:

- Use **12–16+ character passwords**
- Mix **uppercase, lowercase, numbers, symbols**
- Avoid names, DOB, phone numbers, and common passwords
- Use strong hashing algorithms like **bcrypt / Argon2**
- Enable **MFA** for all important accounts
- Use password managers for safe storage

G. Conclusion

This task helped me understand how authentication security works in real systems. By studying hashing algorithms and cracking techniques, I learned why weak passwords are risky and easily compromised. I also understood that strong hashing methods and MFA are essential to prevent account hacking. This knowledge is very useful in cybersecurity roles such as **SOC Analyst and Digital Forensic Investigator.**




```
┌──(root㉿kali)-[/home/kali]
└─# john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8×3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
password123      (?)
1g 0:00:00:00 DONE (2026-01-20 04:16) 50.00g/s 76800p/s 76800c/s 76800C/s 753951..mexico1
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

┌──(root㉿kali)-[/home/kali]
└─#
```