Password Security Authentication Analysis

Introduction

Passwords are the first line of defense in protecting digital accounts and sensitive data. Understanding how passwords are stored—whether through **hashing** (one-way transformations) or **encryption** (reversible encoding)—is crucial for assessing security. Different hashing algorithms like `MD5`, `SHA-1`, and `bcrypt` provide varying levels of protection, with stronger hashes resisting attacks more effectively.

Tools Used

- Kali Linux
- John the Ripper
- OpenSSL
- Wordlist

How Passwords Are Stored (Hashing vs Encryption)

Hashing

- One-way process (cannot be reversed).
- Same password → same hash (unless salt is used).
- Used for password storage.
- Example: `password123` → `ef92b778bafe771e89245b89ecbc08a44a4e166c`
- **Advantages:**
  - Secure
  - Common in Linux and websites

Encryption

- Two-way process (can be decrypted with a key).
- Used for data protection, not passwords.

Different Hash Types

Hash Type	Length	Secure?	Notes
MD5	32 hex	Weak	Fast, easily cracked
SHA-1	40 hex	Weak	Deprecated
SHA-256	64 hex	Better	Still fast
bcrypt	Variable	Strong	Slow + salted
SHA-512	128 hex	Better	Used in Linux

Table 1: Comparison of Common Hash Types

Attempt Cracking Weak Hashes (Wordlist)

Procedure

Step 1: Generate MD5 Hash

```
echo -n password123 | md5sum
```

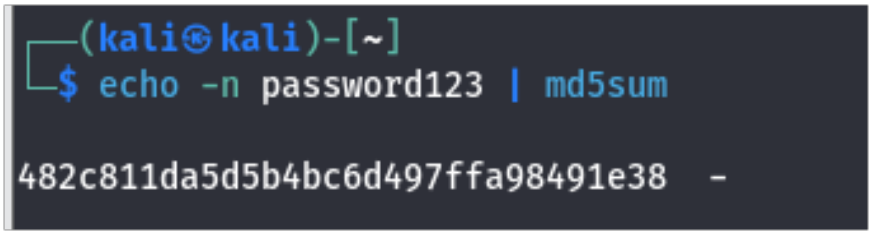

Figure 1: Generating MD5 hash using md5sum

Step 2: Save Hash

```
nano hash.txt
```

Figure 2: Saving MD5 hash into hash.txt

Step 3: Create Wordlist

```
nano small.txt
```

Figure 3: Creating a custom wordlist

Step 4: Crack Hash Using John the Ripper

```
john --format=raw-md5 --wordlist=small.txt hash.txt
```

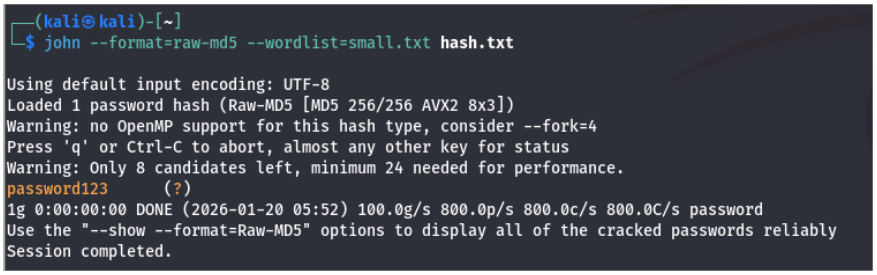

Figure 4: Cracking MD5 hash using John the Ripper

Step 5: Show Result

```
john --show --format=raw-md5 hash.txt
```

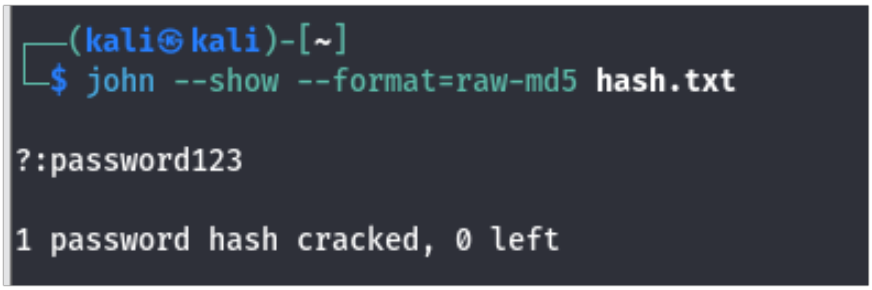

Figure 5: Displaying cracked password result

Brute Force vs Dictionary Attacks

Dictionary Attack

- Uses a list of known or common passwords.
- Faster than brute force attacks.
- Limited by the size and quality of the wordlist.

Example:

```
password
admin123
qwerty
```

Brute Force Attack

- Tries all possible character combinations.
- Very slow compared to dictionary attacks.
- Guaranteed to succeed eventually (if no lockout or rate limiting is applied).

Example:

```
aaaa → aaab → aaac → ...
```

Attack Comparison

Analysis

Weak passwords are easily cracked due to common patterns and wordlist availability.

Attack Type	Speed	Success
Dictionary Attack	Fast	Limited
Brute Force Attack	Slow	Guaranteed (theoretically)

Table 2: Comparison of Brute Force and Dictionary Attacks

Why Weak Passwords Fail

Characteristics of Weak Passwords

- Short length
- Use of common words
- No symbols or special characters
- Reused across multiple websites

Examples of Weak Passwords

```
123456
password
admin@123
qwerty
```

Why They Are Dangerous

- Cracked in seconds
- Found in common password wordlists
- Vulnerable to credential stuffing attacks

Recommendations

- Use strong passwords
- Enable MFA
- Avoid password reuse
- Use secure hashing algorithms

Multi-Factor Authentication (MFA)

Definition

$$MFA = Somethingyouknow + Somethingyouhave + Somethingyouare$$

Examples

- Password + OTP
- Password + Fingerprint
- Password + Authenticator App

Why MFA Is Important

- Even if a password is cracked, the attacker is still blocked
- Prevents phishing and credential attacks
- Essential for banking, email, and admin accounts

Recommendations for Strong Authentication

- Use long passwords (12–16 characters)
- Mix uppercase, lowercase, numbers, and symbols
- Never reuse passwords
- Use password managers
- Enable MFA everywhere
- Store passwords using `bcrypt` or `Argon2`
- Apply account lockout policies

Summary

Strong password security relies on proper hashing, avoiding weak passwords, understanding attack methods, and implementing multi-factor authentication (MFA). Using long, complex passwords, secure storage (bcrypt/Argon2), and enabling MFA significantly reduces the risk of breaches, making systems safer against attacks like dictionary, brute force, and credential stuffing.