Password Security & Authentication Analysis Report

Name: Adil Firoz

Internship Program: Cyber Security Internship

Task: Task 4 – Password Security & Authentication

Operating System: Kali Linux

Tools Used: John the Ripper, RockYou Wordlist

Date: 20-01-2026

1. Introduction

The objective of this task is to analyze password security mechanisms and understand how weak passwords can be compromised. This task focuses on password hashing, dictionary attacks, and the importance of strong authentication methods.

Through practical experimentation, a weak password was hashed and successfully cracked using a dictionary attack. This demonstrates why strong passwords and secure authentication mechanisms are critical in modern systems.

2. Hashing vs Encryption

Hashing is a one-way process used to securely store passwords. Once a password is converted into a hash, it cannot be reversed to obtain the original password.

Encryption, in contrast, is a reversible process where encrypted data can be decrypted using a secret key.

Hashing is preferred for password storage because it protects user passwords even if the password database is compromised.

3. Common Hash Types

Different hashing algorithms are used for password storage. Some common examples include:

- MD5 – A fast hashing algorithm but insecure and vulnerable to attacks

- SHA-1 – Weakened by known vulnerabilities

- bcrypt – A strong hashing algorithm designed specifically for password storage

Modern systems use bcrypt or Argon2 because they are slow and resistant to brute force attacks.

4. Hash Generation

A weak password was selected for testing purposes and converted into an MD5 hash using the md5sum command in Kali Linux.

The selected password was "password123". The command used was:

echo -n password123 | md5sum

This generated a fixed-length MD5 hash, which was stored in a file named hash.txt for further analysis.

Figure 1: Generating an MD5 hash and storing it in a file

5. Dictionary Attack Using John the Ripper

The generated MD5 hash was attacked using a dictionary attack with John the Ripper and the RockYou wordlist.

The following command was used:

john --format=raw-md5 hash.txt --wordlist=/usr/share/wordlists/rockyou.txt

John the Ripper successfully recovered the original password from the hash. This demonstrates how easily weak passwords can be cracked using publicly available wordlists.

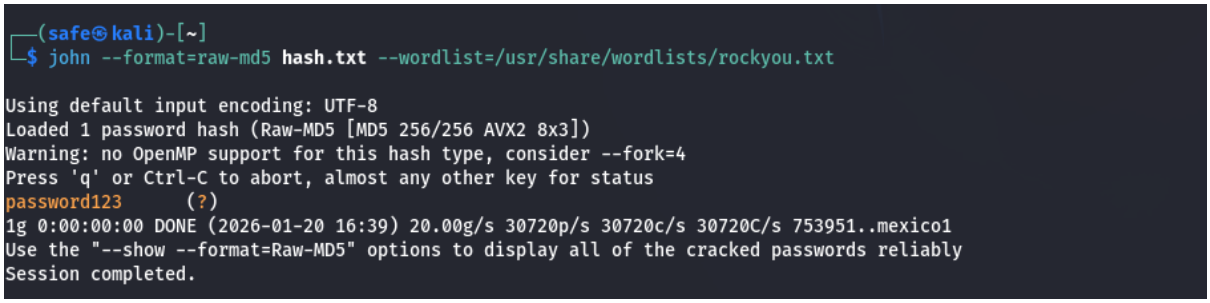

Figure 2: Dictionary attack successfully recovering the original password

6. Brute Force vs Dictionary Attacks

A dictionary attack uses a predefined list of common passwords to attempt cracking password hashes.

A brute force attack tries all possible combinations of characters until the correct password is found.

Dictionary attacks are faster but limited to known passwords, while brute force attacks are more exhaustive but significantly slower.

7. Strong Hashing and Secure Storage

Strong hashing algorithms such as bcrypt and Argon2 are recommended for secure password storage.

These algorithms use salting and intentionally slow computation to make brute force and dictionary attacks extremely difficult. Using strong hashing algorithms greatly improves password security.

8. Multi-Factor Authentication (MFA)

Multi-Factor Authentication (MFA) provides an additional layer of security by requiring more than one authentication factor.

Common MFA examples include:

- Password and One-Time Password (OTP)

- Password and biometric authentication

MFA protects user accounts even if the password is compromised.

9. Observations

The following observations were made during this analysis:

- Weak passwords can be easily cracked using dictionary attacks

- Fast hashing algorithms such as MD5 are insecure

- Strong hashing algorithms provide significantly better protection

- Multi-factor authentication greatly improves account security

10. Recommendations

Based on the analysis, the following recommendations are suggested:

- Use long and complex passwords

- Avoid common and reused passwords

- Use strong hashing algorithms such as bcrypt or Argon2

- Enable multi-factor authentication on all critical accounts

- Use password managers to generate and store secure passwords