CYBER SECURITY INTERNSHIP

Ministry of MSME, Govt. of India

Task 4

Password Security & Authentication Analysis

JASHMI KS

Password Security & Authentication is the study of how passwords are created, stored, protected, and verified to prevent unauthorized access. It focuses on techniques like hashing, secure authentication methods, and attack prevention to ensure user identities remain safe from cyber threats.

Hashcat is a password recovery and auditing tool used to evaluate the strength of password hashes. It works by comparing stored hashes with hashes generated from guessed passwords. Hashcat is widely used in cybersecurity labs to demonstrate how weak passwords can be compromised and why strong authentication practices are necessary.

Hashcat does not decrypt passwords. Instead, it recreates hashes from guessed passwords and checks for matches with the target hash.

Passwords

Passwords should never be stored in plain text because anyone gaining database access can directly see them. Instead, secure systems store passwords using hashing.

Hashing

Hashing converts a password into a fixed-length value called a hash using a mathematical function. It is a one-way process, meaning the original password cannot be retrieved from the hash. When a user logs in, the entered password is hashed again and compared with the stored hash.

Encryption

Encryption converts data into an unreadable format using a secret key and can be reversed using decryption. Encryption is suitable for protecting files or transmitted data but is not ideal for password storage because encrypted passwords can be recovered if the key is compromised.

Hash Types (MD5, SHA-1, bcrypt)

Different systems use different hashing algorithms, each with varying security levels.

MD5

MD5 produces a 128-bit hash and is very fast. However, it is considered **insecure** because it is vulnerable to collision and brute-force attacks. MD5 hashes can be cracked within seconds using modern hardware.

SHA-1

SHA-1 produces a 160-bit hash and is stronger than MD5 but has known vulnerabilities. It is no longer recommended for secure applications due to collision attacks.

bcrypt

bcrypt is a modern, secure hashing algorithm designed specifically for passwords. It is slow by design and uses salting, which makes brute-force and dictionary attacks very difficult. bcrypt is widely used in modern authentication systems.

Generating Password Hashes

Generating password hashes means converting a normal password into its hashed form using a hashing algorithm. This process simulates how real systems store passwords securely. Hash generation helps in understanding:

- How passwords look once stored
- Why hashes cannot be reversed
- How different algorithms produce different outputs for the same password

Hash generation is essential for testing password strength and analyzing attack resistance.

Cracking Weak Hashes Using Wordlists

Password cracking is the process of attempting to discover the original password from its hash.

Wordlist Attack

A wordlist attack uses a predefined list of commonly used passwords such as names, dates, or simple patterns. These attacks are fast and effective against weak passwords because many users reuse predictable passwords.
Weak hashes cracked quickly indicate poor password choices.

Brute Force vs Dictionary Attacks

Dictionary Attack

- Uses a list of common passwords
- Faster
- Efficient against weak passwords
- Fails if password is unique or complex
-

Brute Force Attack

- Tries every possible combination of characters
- Very slow
- Guaranteed to succeed given unlimited time
- Becomes impractical for long, complex passwords

Weak Passwords Fail

Weak passwords fail because they lack complexity and predictability. Common reasons include:

- Short length
- Use of dictionary words
- Simple patterns (123, abc)
- Reuse across multiple platforms

Such passwords exist in public wordlists and are easily cracked using automated tools, making accounts vulnerable to attacks.

Multi-Factor Authentication (MFA) and Its Importance

MFA adds an extra layer of security by requiring more than one form of authentication:

- Something you know (password)
- Something you have (OTP, phone)
- Something you are (biometrics)

Even if a password is compromised, MFA prevents unauthorized access, making it one of the strongest defenses against account takeover attacks.

Recommendations for Strong Authentication

1. **Use Strong and Long Passwords**
   Passwords should be at least **12–16 characters long** and include a combination of uppercase letters, lowercase letters, numbers, and special characters. Longer and complex passwords are harder to crack using dictionary or brute-force attacks.

2. **Avoid Common and Predictable Passwords**
   Users should avoid dictionary words, names, dates, or simple patterns like 123456 or password. Such passwords are easily found in public wordlists and can be cracked quickly.

3. **Use Secure Hashing Algorithms**
   Modern password hashing algorithms such as **bcrypt, Argon2, or PBKDF2** should be used instead of weak algorithms like MD5 or SHA-1. These algorithms are slow by design and resist brute-force attacks.

4. **Implement Salting for Password Hashes**
   A unique random salt should be added to each password before hashing. Salting prevents rainbow table attacks and ensures that identical passwords produce different hashes.

5. **Enable Multi-Factor Authentication (MFA)**
   MFA should be implemented to add an extra layer of security. Even if a password is compromised, additional factors such as OTPs or biometrics can prevent unauthorized access.

6. **Limit Login Attempts**
   Systems should restrict the number of failed login attempts and temporarily lock accounts after multiple failures. This helps protect against brute-force and automated attacks.

7. **Encourage Regular Password Updates**
   Users should periodically change passwords, especially if there is a suspected data breach. This reduces the risk of long-term account compromise.

8. **Educate Users About Password Security**
   Users should be trained on good password practices and the risks of password reuse across multiple platforms. Awareness significantly improves overall security.