Task 6 : Create a Strong Password and Evaluate Its Strength

**Password strength:-**

Password strength refers to how resistant a password is to being guessed or cracked by attackers using various methods like brute-force attacks, dictionary attacks, or social engineering. A strong password is hard for both humans and machines to guess.

**Characteristics of a Strong Password:**

1. Length: At least 12–16 characters.
2. Complexity: Includes a mix of:
   * Uppercase letters (A–Z)
   * Lowercase letters (a–z)
   * Numbers (0–9)
   * Special characters (!, @, #, $, etc.)
3. Unpredictability: Avoid using personal info like names, birthdays, or common words.
4. Uniqueness: Never reuse passwords across different accounts.

**Example for sting password:**

* V!ah@25678@kksd!
* Gd@#647jdo123
* Mmko133!753$#@tcAB

**Why Password Strength is Important:**

* Prevents unauthorized Access
* Protect personal data
* Stops account takeover
* Helps in compliance
* Slow Down attacks

**Best Practices for Password Security:**

1. **Use a Password Manager** – To generate and store complex passwords securely.
2. **Enable Multi-Factor Authentication (MFA)** – Adds an extra layer of security.
3. **Change Passwords Periodically** – Especially for sensitive accounts.
4. **Avoid Password Reuse** – One breach can compromise multiple accounts.
5. **Test Password Strength** – Use secure tools to evaluate your passwords.

**Brute Force attack:-**

A brute force attack tries every possible combination of characters until the correct password is found.

**How it works:**

* The attacker uses a computer program to try all possible combinations of letters, numbers, and symbols.
* For example, it tries: a, aa, ab, abc, abcd1, etc., until it finds the correct password.

**Time required:**

* Depends on **password length and complexity**.
* Simple passwords can be cracked in **seconds**.
* Strong, long passwords can take **years** or even **centuries**.

**Protection Against Brute Force:**

* Use long and complex passwords.
* Enable account lockouts after several failed attempts.
* Implement CAPTCHA.
* Use multi-factor authentication (MFA).

**Dictionary attack:-**

A dictionary attack uses a predefined list of common passwords or words (like a dictionary) to guess your password.

**How it works:**

* The attacker tries every word from a list of commonly used passwords or dictionary words.
* Faster than brute force because it doesn’t test every combination — just the most likely ones.

**Protection Against Dictionary Attacks:**

* Avoid using real words or common patterns.
* Use random combinations of letters, numbers, and symbols.
* Don't use personal info like names, birthdates, or common phrases.

**Authenticity:-**

Authentication is the process of verifying a user's identity before granting access to a system, application, or data.

**Common Authentication Methods:**

1. **Something you know** – Password, PIN
2. **Something you have** – OTP, security token, smart card
3. **Something you are** – Biometrics (fingerprint, face scan)

**Best Practices for Secure Authentication:-**

**1**. Use Strong, Unique Passwords

* At least 12–16 characters
* Mix of uppercase, lowercase, numbers, and symbols
* Avoid dictionary words or personal info

2. Enable Multi-Factor Authentication (MFA)

* Add a second layer: OTPs, authenticator apps (e.g., Google Authenticator, Microsoft Authenticator)
* Even if a password is compromised, MFA keeps the account secure

3. Implement Account Lockout Policies

* Lock account temporarily after multiple failed login attempts
* Prevents brute-force attacks

4. Use CAPTCHA

* Defends against automated login attempts (bots)

5. Secure Password Storage (For Developers)

* Never store plaintext passwords
* Use bcrypt, Argon2, or PBKDF2 hashing algorithms
* Add salt to hashes to prevent rainbow table attacks

6. Session Management

* Use secure, time-limited sessions
* Automatically log out users after inactivity
* Regenerate session IDs after login

7. Transport Security

* Always use HTTPS (SSL/TLS) for login pages and data transmission

8. Limit Login Attempts and Notify Users

* Alert users on suspicious login attempts or logins from new devices/locations

9. Avoid Security Questions Alone

* Many security questions can be easily guessed or found online
* Better to rely on email/phone-based recovery or MFA

10. Use Authentication Tokens (For APIs)

* Use JWT (JSON Web Tokens), OAuth 2.0, or API keys securely
* Keep them short-lived and revokeable

**Authenticity Type:-**

* Password-based
* MFA/2FA
* Biometric
* Single sign-On
* Token based