Information assurance & security

PROJECT

By Yassine Marrekchi

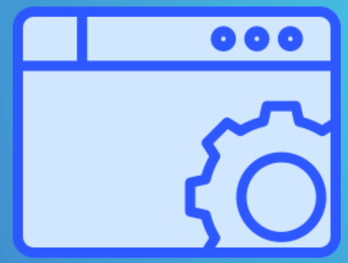
- sensitive information are facing unauthorized access risk which which arises the need for strong passwords .

OBJECTIVE : test password strength using OWASP standards

Main components

Input interface

Backend Logic



Passphrase Generator

Feedback Mechanism

Input interface

Secure text field for username and password

username

password

Buttons for checking strength and generating passphrase

CHECK STRENGTH

GENERATE PASSPHRASE

Show/Hide Password feature

Backend Logic

- Password strength evaluation algorithm
- Password validation upon prior data

- Passphrase Generator :
  - Generates random and secure passphrases to use as a password.
  - Higher security due to increased entropy.

Password Strength Criteria

Minimum Length : 8-12 characters

Complexity : Mix of characters (upper/lowercase, digits, special)

Avoid sequential characters and public information

Password Strength Levels

- **Weak** : Less than 8 characters, lacks diversity.
- **Moderate** : 8-12 characters , mixed types.
- **Strong** : More than 12 characters + complexity, diverse and unpredictable.

Advantages of the solution

- Reduces risks of identity theft and cyber breaches.
- User-friendly interface and real-time feedback.
- Integration with password management systems.

Limitaions of existing solutions

- **Authentication gaps :** ignoring the use of personal information as a password (e.g : phone number)

- **Langauge restrictions :**

  English 

- **Limitations of criterion :**

  ignores other important features such as vulnerability to brute force attacks

Tools used in this project

- **JavaScript :**
  - basic input functions
  - passphrase generator mechanism
  - evaluation mechanism
  - SHA-256 hash + crack time estimation
  - hide/show password

- **HTML :**
  - structuring

- **CSS :**
  - Layout aesthetics

- **JSON databse :**
  - for commonly used passwords

Overview of the interface