PRODUCT 1: WEAK PASSWORD TEST TOOL

PRODUCT CATEGORY: IT TOOLS

WORKING IDE: CYBERSHIELD TANZANIA INC. WEBSITE, (CAN BE DOWNLOADED BY A USER TO HIS/HER ENVIRONMNET UNDER COMMON DISTRIBUTION LICENCE AND SITE POLICY ADHERENCE).

PRODUCT NAME; MYPASSWORD (Any best is allowed)

DEVELOPERS TEAM:

Product design and quality control: Godwin Aruga

UI/UX design and visual implication: Dinnaless

Product & System integration: Davis dol

MANIFEST RELEASE:20/01/2024

EXPECTED COMPLETION: 17 FEB 2024

PRODUCT DESCRIPTION/MANIFEST:

A weak password test is a security tool that analyzes passwords used in a system to identify those that are vulnerable to hacking. These tools help protect against potential data breaches and unauthorized access.

Targeted users;

CyberShield TM, verify passwords during user registry

Enterprise network users (customers) can implement it on their platform based on agreement

Working mechanism

- Scans passwords: it examines a database of passwords, either on a company's network or within an application.
- Checks for vulnerabilities. by comparing passwords against various criteria, such as:
    - Length: Passwords that are too short are easier to crack.
    - Complexity: Passwords lacking a mix of uppercase and lowercase letters, numbers, and symbols are also weak.

- Common words and phrases: Using dictionary words or easily guessable phrases like "password123" is a major red flag.
- Keyboard patterns: Sequences like "qwerty" or "123456" are readily crack able due to their predictability.
- Personal information: Incorporating usernames, birthdays, or other personal details makes passwords vulnerable.
- Previously breached passwords: Checking against known lists of leaked passwords from past data breaches is crucial.

Here available WPT tools in the market.

- KnowBe4 Weak Password Test (WPT): Popularly used for analyzing Active Directory passwords, it checks for several types of weak patterns and generates detailed reports.
- PasswordSafe: This commercial tool offers more advanced features like customizable dictionaries and breached password list integrations.

PRODUCT IMPLEMENTATION:

Use of **zxcvbn library** to develop **mypassword** test product

- What is **zxcvbn library**

zxcvbn is a password strength estimation library with a multi-pronged approach. It analyzes passwords based on length, character sets, dictionary matches, keyboard patterns, temporal and spatial proximity, and l33t speak replacements. It assigns a score based on these factors, with longer, more complex passwords receiving higher scores. You can customize zxcvbn with minimum password lengths, custom dictionaries, and different entropy calculations.

Here's how it works in three main steps:

1. Match: zxcvbn scans for various patterns within the password, including common words, keyboard sequences, dates, and l33t speak substitutions.

2. Score: Each pattern receives a penalty based on its prevalence and predictability. The overall score reflects the combined penalties of all identified patterns.
3. Search: zxcvbn estimates the number of guesses needed to crack the password based on its score and complexity. A higher score indicates a stronger password requiring more guesses.

MODE OF IMPLEMENTATION.

Integrating zxcvbn into your website to allow users to test their password strength involves several steps. Here's a breakdown

IMPLEMENTATION METHOD. (Choose any suitable for you.)

- Direct JavaScript inclusion- Download the zxcvbn JavaScript library and directly include it in your website's code. This gives you full control over the integration but requires more coding effort.
- CDN inclusion: Use a Content Delivery Network (CDN) like jsDelivr to host the zxcvbn library for faster loading times. This simplifies inclusion but slightly reduces control.
- JavaScript module libraries: Leverage libraries like npm or Yarn to manage zxcvbn as a dependency, simplifying dependencies but adding another layer of complexity.

2. Implement the password input field

- Create an HTML input field where users can enter their password. Ensure it's marked as type="password" for security.

3. Connect zxcvbn to the input field:

- On any interaction with the input field (e.g., typing, blur event), call the zxcvbn function, passing the user's entered password as input.

4. Display the password strength feedback:

- Based on the zxcvbn output, use JavaScript to display the calculated password strength score and corresponding feedback message (e.g., "weak", "strong"). You can customize the feedback messages and styling to match your website's design.

Here are some helpful resources to get you started:

- zxcvbn documentation: https://github.com/zxcvbn-ts/zxcvbn
- zxcvbn CDN inclusion: https://www.jsdelivr.com/package/npm/zxcvbn
- JavaScript password strength meter example: https://codepen.io/tag/password-validation

PRODUCT 2: PASSWORD GENERATING TOOL

PRODUCT CATEGORY: IT TOOLS

WORKING IDE: CYBERSHIELD TANZANIA INC. WEBSITE, (CAN BE DOWNLOADED BY A USER TO HIS/HER ENVIRONMNET UNDER COMMON DISTRIBUTION LICENCE AND SITE POLICY ADHERENCE).

PRODUCT NAME; CYBERPASS (Any best is allowed)

PRODUCT DESCRIPTION:

Password Generator Tool Manifest

Target audience: online application users looking to design a user-friendly password.

Goal: Create a simple yet effective tool that generates strong and unique passwords while incorporating user input for a touch of personalization.

Features:

- User input: Allows users to enter a base string of approximately 5 characters.
- Random character mixing: Uses a secure algorithm to shuffle and mix the user's input characters with additional random characters to generate a 16-character password.

- Visual strength indicator: Displays a color-coded or bar-graph indicator representing the generated password's strength (e.g., green for strong, yellow for moderate, red for weak).
- Optional character pool customization: Allows advanced users to choose which character types (uppercase/lowercase letters, numbers, symbols) are included in the generated password.
- Copy to clipboard button: Enables users to easily copy the generated password to their clipboard for pasting into login fields.

Language and tone:

Python or any other general-purpose language

Algorithm implementation: - choose any is best for you to work with

- Secure randomness: Utilize a cryptographically secure random number generator (CSPRNG) to ensure randomness and prevent predictability in the generated passwords. Popular CSPRNG algorithms include ChaCha20 and AES-CTR.
- Character mixing: Implement a shuffling algorithm like the Fisher-Yates shuffle to randomly mix the user's input characters with additional random characters from the chosen pool.
- Password strength estimation: Use a lightweight password strength estimation algorithm like zxcvbn to provide visual feedback on the generated password's security.

Additional considerations:

- Security: Implement best practices for secure password generation and storage. Avoid storing user input or generated passwords in plain text.

By following this manifest, its my hope that we can create a valuable password generator tool that empowers users to choose strong and unique passwords for better online security.

Any question relating to this manifest is more important, this is not final but baseline for what best we can make

Goodluck

ARUGA Godwin

CyberShield TL.

29/01/2024