Software Engineering Assessment

(Group X)

Password Generator

Submitted by

Gagana Kusuma K

Rakesh Kumar Saw

Table of Contents

Abstract:

Text password is a very common user authentication technique. Users face a major problem, namely that of many site-unique and strong (i.e. non-guessable) passwords. One way of addressing this is by using a password generator which generates (and regenerates) site-specific strong passwords on demand, with minimal user input. Password Generator enables the generation of passwords that meet important real world requirements, including forced password changes, use of pre-specified characters, displaying the number of characters used in password and checking passwords strength.

Problem Definition

In this project our motto is to generate a random password based on user's need. User will input his/her need of no of digits, special character, small alphabets, capital alphabets and based on input a random password will be generated. After generating the random password a button of 'Copy to Clipboard' will provide the functionality of copying the generated password to our clipboard.

It also enables the users to check the strength of their passwords that they have had used on their multipurpose sites. It enables user to check the character they have used to create the password.

Introduction

Passwords remain a very widely used method for user authentication, despite widely shared concerns about the level of security they provide. There are many potential replacement technologies, including combinations of biometrics and trusted personal devices, but it seems likely that it will be some time before passwords are relegated to history. Given their current and likely future wide use, finding ways of improving the use and management of passwords remains a vitally important issue. We focus here on an important practical matter, namely how to make password more secure and more convenient. Passwords can be stored either locally or on a trusted server; most browsers provide a local-storage password manager. However, the shortcomings of password managers have also been widely documented.

for flow chart:

Password Generator enables the user to generate the password of their choice like the number of words, small case alphabets, digits, etc. A Clipboard enables user to copy the password that is generated using the Password Generator. It also enables the user to check the strength of the passwords. It also displays the no. of characters, no of alphabets, and no of symbols used in the passwords.

Some password generators are simply random password generators. These programs produce complex/strong passwords with combinations of numbers, uppercase and lowercase letters, and special characters such as braces, asterisks, slashes, etc. Other types of password generators are made to generate more recognizable passwords rather than a completely random set of characters. There are tools for generating pronounceable passwords, as well as custom tools that allow users to set detailed criteria. For instance, a user could set a request for a certain number of characters, a certain mix of letters and numbers, a certain number of special characters, or any other criteria for generating a new password.

Password generators help those who have to constantly come up with new passwords to ensure authorized access to programs and to manage a large number of passwords for identity and access management. Other kinds of tools include password managers, or “password vaults”, where users manage large numbers of passwords in a secure location.

How does Password Generators Work

1. **Seed Generation**: The generator uses a **seed** to create a random string of characters. This seed is crucial because it determines the uniqueness of the generated password. Different generators handle seeds differently. Some use truly random seeds that cannot be predicted. Others employ predefined algorithms, which can be reverse-engineered if someone gains access to the seed used for password creation.

2. **Password Creation**: Once the seed is in place, the generator creates a password by combining letters, numbers, and special characters. The resulting password is both strong and unique.

3. **Discarding the Seed**: After generating the password, the seed can be discarded. Users don't need to remember it; they only need to remember the generated password.

Why You Should Use a Password Generator

Password generators offer several benefits:

1. **Complexity**: They allow you to create long, complex passwords that are difficult to crack.
2. **Ease of Use**: If you struggle to come up with strong passwords, these tools simplify the process.
3. **Unique Passwords**: You can create different passwords for various websites, ensuring that if one gets compromised, the others remain safe. (obviously)
4. **No Need to Remember**: Since you don't need to remember the generated passwords, you can focus on using them securely. storage??

Is It Okay to Use a Password Generator?

Using a **password generator** can be a smart move to enhance online security. Here are some guidelines to consider when creating strong passwords:

1. **Uniqueness**: Avoid using the same password across multiple important accounts. Each account should have a distinct password. (obviously)

2. **Complexity**: Aim for a password with at least **16 characters**. Include a mix of:
    1. **Numbers**
    2. **Uppercase letters**
    3. **Lowercase letters**
    4. **Special symbols**
3. **Avoid Common Patterns**:
    1. Do not use easily guessable information like names, birthdays, or phone numbers.
    2. Steer clear of dictionary words or common phrases.
4. **No Cloning**:

- Refrain from using similar passwords where most characters are the same. If one gets compromised, the others are at risk too.

Conclusion:

Understanding security principles is crucial. A password generator involves creating strong, random passwords that are resistant to attacks. You'll learn about password entropy, hashing, and encryption. Explore techniques to prevent common vulnerabilities like brute force attacks and dictionary-based password guessing.

Explore libraries or frameworks for generating random strings and handling user input. The password generator project successfully achieves its goal by providing a user-friendly interface for generating secure passwords. It demonstrates the power and flexibility of Python programming. Users can confidently use the generated passwords for various accounts and services.

In this work we propose solution that partially solves the issue of users not trusting password managers, which focuses on the process of generating random passwords. Keeping in mind the problem to be solved, is assured to generate passwords compliant with the policy, and we formalize the property that the generator samples the set of passwords according to a uniform distribution.