Abstract

Every person using different online services is concerned with the security and privacy for protecting individual information from the intruders. Many authentication systems are available for the protection of individuals' data, and the password authentication system is one of them. Due to the increment of information sharing, internet popularization, electronic commerce transactions, and data transferring, both password security and authenticity have become an essential and necessary subject. But it is also mandatory to ensure the strength of the password. For that reason, all cyber experts recommend intricate password patterns. But most of the time, the users forget their passwords because of those complicated patterns. In this paper, we are proposing a unique algorithm that will generate a strong password, unlike other existing random password generators. This password will he based on the information, i.e. (some words and numbers) provided by the users so that they do not feel challenged to remember the password. We have tested our system through various experiments using synthetic input data. We also have checked our generator with four popular online password checkers to verify the strength of the produced passwords. Based on our experiments, the reliability of our generated passwords is entirely satisfactory. We also have examined that our generated passwords can defend against two password cracking attacks named the "Dictionary attack" and the "Brute Force attack". We have implemented our system in Python programming language. In the near future, we have a plan to extend our work by developing an online free to use user interface. The passwords generated by our system are not only user-friendly but also have achieved most of the qualities of being strong as well as non- crackable passwords.

1. Introduction

Having a weak password is not good for a system that demands high confidentiality and security of user credentials. It turns out that people find it difficult to make up a strong password that is strong enough to prevent unauthorized users from memorizing it.

Text-based username-password is the most commonly employed authentication mechanism in many multiuser environments. These multiuser applications, while registering users to their application, some applications allow users to create password their own and others generate random password and supply to users.

This application can generate random passwords, with the combination of letters, numerics, and special characters. One can mention the length of the password based on the requirement and can also select the strength of the password.

2. Literature Survey

Author	Title	Year	Source	Findings/Output
Michael D. Leonhard; V.N. Venkatakrishnan	A comparative study of three random password generators.	2007	IEEE	This paper compares three random password generation schemes, describing and analyzing each. Qualities discussed include security, memorability,

				and user affinity.
Farhana Zaman Glory; Atif Ul Aftab; Olivier Tremblay-Savard; Noman Mohammed	Strong Password Generation Based On User Inputs	2019	IEEE	In this paper, they implemented UI based password generator that takes inputs from the users.

3. Objectives

- Many authentication systems are available for the protection of an individual's data, and the password authentication system is one of them to ensure security strong password is essential.
- Weak passwords are vulnerable to brute-force attacks, to help the users to protect their accounts from brute-force attacks.
- To help the users easily generate and copy the passwords to their clipboards.

4. Problem statement

Random Password Generator in Python

5. Existing system

a. When the existing system was studied, it was found to have some problems, the user interface was not good as there is no option to specify the strength of the passwords.
b. Some of the systems were only built to generate uppercase and lowercase strings that make passwords vulnerable for brute-force attacks.

c. Users need to manually type the generated passwords as the password is a little complex it will be time-consuming.
d. While working on this project we have taken into consideration all the above drawbacks and included them in our project.

6. Proposed system

a. The proposed system is designed to generate random passwords based on the length given by the user.
b. Current systems consider the following parameters while constructing the password.
    i. Length of the password.
    ii. Strength of the password.
c. Users can input the required length to get the password.
d. The strength of the password is the complexity of the password. There are 3 options given to the user.
    i. Easy - Includes only lower case letters.
    ii. Medium - Includes both upper and lower case letters.
    iii. Strong - Includes Upper case, Lower case, and Symbols.
e. The generated password can be copied as a string.
f. The user has the interface to exit from the system.

7. Applications

- To prevent your passwords from being hacked by social engineering, brute force, or dictionary attack method, and keep your online accounts safe.
- One-click to generate and copy the password.
- Includes lowercase, uppercase, and symbols to make the password more complex and difficult to memorize.
- Written in python that makes the executions of the program speed and reliable.
- Users have options to select the strength of the passwords based on their requirements.

8. Merits & Demerits

Merits :

- High set of potential or huge pool of high security passwords. Beneficial for one-time authentication. Difficult passwords are hard to crack.
- Easy user interface.
- Generated the passwords that are hard to crack by social engineering, brute-force attacks.
- Allows users to decide length and strength of the passwords.
- Mixture of capital letters, small letters and special characters makes password guessing difficult.

Demerits:

9. Remembering the generated password is difficult.
10. Managing multiple passwords for multiple sites or account is difficult.

11. Conclusion

Password is one of the most important factor of authentication and it needs to be strong and unique to help users to generate such passwords our project is developed.

Project is developed using python thus it provides faster execution speed and also used tkinter to develop user interface.

To conclude, Solving such real world problems helped us to understand fundamentals of python and building user interface.

12. Future scope

Current project allows the users to generate and copy the password, there is no way that user can store and manage passwords.

Future scope of the project will be allowing users to store and manage the generated password in application itself.

13. References

https://docs.python.org/3/tutorial/index.html

https://docs.python.org/3/library/random.html

https://realpython.com/python-gui-tkinter/