International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Special Issue: 1 Volume No.9 March - April – 2025
DOI:10.46647/ijetms.2025.v09si01.021 ISSN: 2581-4621

Password Vulnerability assessment: a safe checker using online breach data

Mamatha.P
Assistant professor, CSE Department, AVN Institute of Engineering and Technology, Hyderabad, Telangana, India.

Abstract:
Passwords serve as the primary authentication mechanism for online accounts, yet they remain vulnerable to breaches and attacks. This study presents a safe and efficient password vulnerability assessment tool that leverages online breach data to evaluate password security without exposing user credentials to additional risks. The proposed system cross-references user-provided passwords with known breach datasets while ensuring privacy and security through encryption and hashing techniques. By analyzing password strength, reuse patterns, and exposure history, this approach enhances security awareness and helps mitigate credential-based attacks. The implementation of such a checker contributes to proactive cybersecurity measures, reducing the likelihood of unauthorized access due to compromised credentials.

Keywords: Password security, vulnerability assessment, data breaches, authentication, credential exposure, password checker, cybersecurity, password reuse, encryption, hashed breach data.

1. Introduction

Passwords are essential features for internet security, safeguarding account details and confidential information. Their contrast is when insecure and breached login credentials lead to major security vulnerabilities. With an increasing number of internet accounts and frequent use of credentials using platforms in general individuals are at risk of having their login credentials compromised[1].

This necessitates the use of effective methods of password security checks, evaluating strength, and the creation of secure substitutes. New technology and automation have greatly enhanced password security applications, and now more complex methods can be used to determine faults and enhance password management. Three main operations are emphasized in this project: determining if a password has been hacked, checking for the strength and time taken to crack the password, and generating strong, configurable passwords.

This checker of password abuses applies the "Have I Been Pwned" API as well as SHA-1 hashing in order to safely check whether a login credential has led to a breach. This password strength analyzer checks attributes such as size, character diversity, and predictability, projecting how long would it take to crack the password. The password creator allows individuals to create strong credentials based on their preference, either including or only excluding lower case letters, the upper-case letter, numerals, as well as special symbols[2].

2. Problem statement

The project responds to the increasing need for strong technologies to enhance password security and protect account data against breaches. Weaker, compromised, or repeated passwords are a severe weakness that facilitates unauthorized use, data theft, and identity fraud. The project will design a complete platform to tackle three major aspects of password security: identification of breached passwords, estimation of password strength, and creation of secure passwords with respect to user preferences. The goal is to develop a framework that is dependable, intuitive, and efficient with respect to maintaining privacy and security during analysis and generation processes[3].

Objectives

The input system shall incorporate straightforward user interfaces for input of login information, password setting generating options, and viewing results. These will be designed to cater to a wide range of inputs, such as plain-text passwords for scrutiny and specific criterion for generating passwords safely (for instance, addition or exclusion of certain characters). Real-time validation provides data accuracy by validating for inconsistencies or incorrect entries, with users getting simple notifications of mistakes and directives. The system will also provide users with the ability to validate their inputs prior to submission, avoiding mistakes and building confidence in the results. The project provides users with the ability to effectively secure their accounts through features such as real-time password compromise detection, strength analysis, and secure password creation. This method reduces risks like unauthorised access, information theft, and identity theft while promoting enhanced password habits. The site is designed to help customers maintain their internet security in a secure and efficient manner[4].

3. Literature survey

This research examines the application of the Pwned Password API, which employs kAnonymity and SHA1 hashing to identify secure breaches. The authors highlight the security and efficiency of hash comparisons and the capacity of the API to identify exposed credentials without exposing the full password hash. By integrating password strength assessment, breach detection, and secure password generation, this project empowers users to enhance their web security while making people more aware of the need for strong, uncompromised passwords. In this article, the project's breach detection component is described in detail, with a focus on kAnonymity for secure API queries and SHA1 hashing for breach testing. Although the research is great at identifying violations, it does not cover password strength analysis and password generation, which are additional features added to the project. The project focuses on efficiency, precision, and user privacy, ensuring that sensitive information is processed securely while generating fast and reliable results. By integrating these traits into an online platform, this project aims to provide users with a simple and efficient solution to password management and security[4].

Existing password-checking and generation practices, like the "Pwned Password API," independent strength checkers, and basic random password generators, offer essential functionality like breach identification and strength assessment. They are often hindered by limitations like a lack of adequate password generating options, inadequate user education, and minimized security deployment. These systems do not often integrate breach detection and advanced password strength analysis within one platform, leading to fractured solutions for customers who need complete password management[5].

4. PROPOSED SYSTEM

The proposed system is a password manager application developed with HTML, CSS, JavaScript, Flask, and MySQL. It features password breach checking (through the Have I Been Pwned API), advanced password strength checks, and customizable password generation features. The system ensures user security through SHA-1 hashing figure 4 and figure 5, Argon2, and salting algorithms. Real-time password strength recommendations, customizable creation options, and secure storage simplify password management and make it more comprehensive. Integrated Functionality: Combines breach detection, strength testing, and custom creation. Enhanced security through the use of Argon2 hashing and salting. Customisation: Users are able to specify passwords features (letters, symbols, and numbers)[6].

Cloud-Based Reliability: Encodes data and employs MySQL for securing it.

5. Methodology and Modules

Modules:

Home page: The site has a neat and minimalist interface where users can review and control their passwords. The three main sections are clearly recognizable, and the password generator has customisable options for creating powerful passwords[7].

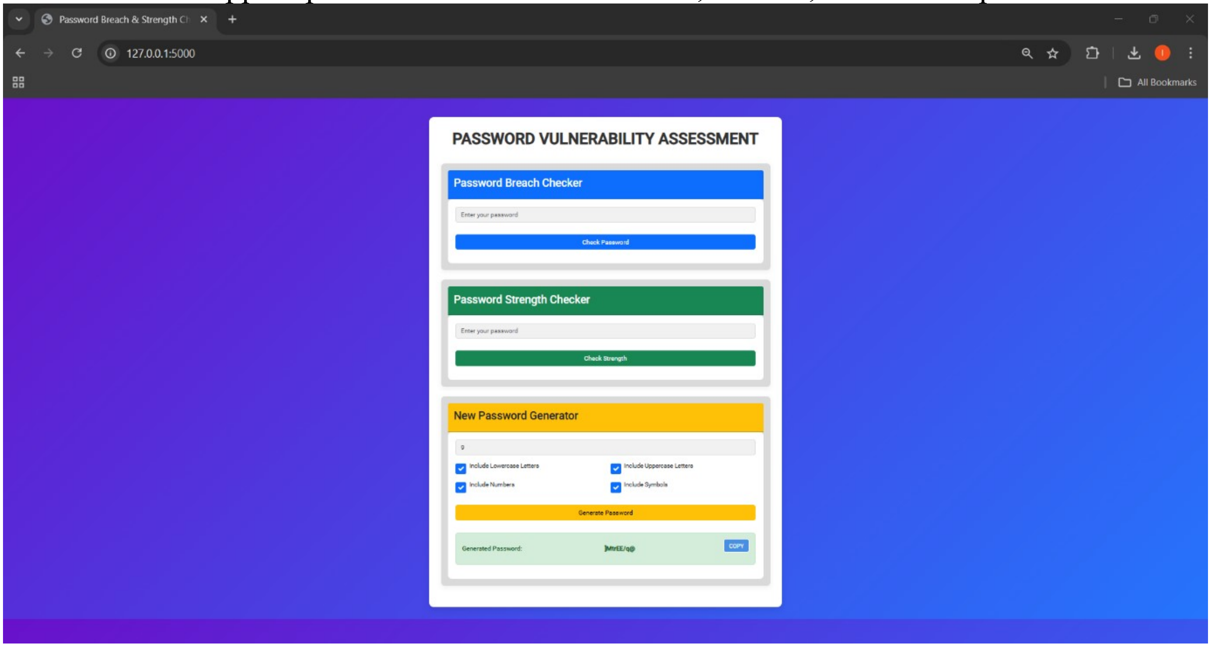
Password Breach Checker: A password breach checker is a program that can inform you whether your password was compromised in a data breach. It works by checking your password against a list of previously compromised credentials.
Password Strength Checker: A password strength checker is a program that checks the security and strength of your password. It analyzes various parameters to ascertain whether the password you select is weak or strong[8].
New Password Generator: A new password generator is software that enables you to generate safe, random passwords. It spares you from the need to devise hard passwords yourself figure 1.

6. Implementation

1. Install Required Libraries
Install libraries required for the project.
2. Data Preparation
Fetch online breach databases like Have I Been Pwned, or any public stash of compromised passwords.
3.Building the Vulnerability Checker figure 2:
Implement a password hashing module via Python's hashlib
4.Feature Engineering:
Generate features from passwords for analysis purposes, e.g., length, usage of special characters, mix of uppercase and lowercase, etc.
5.Deployment with Streamlit[9] figure 3:
Develop an easy-to-use interface for testing password weakness.

7. Results

Host the Streamlit app on platforms like Streamlit Cloud, Heroku, or AWS for public access.



PASSWORD VULNERABILITY ASSESSMENT

Password Breach Checker

Enter your password

Password Strength Checker

Enter your password

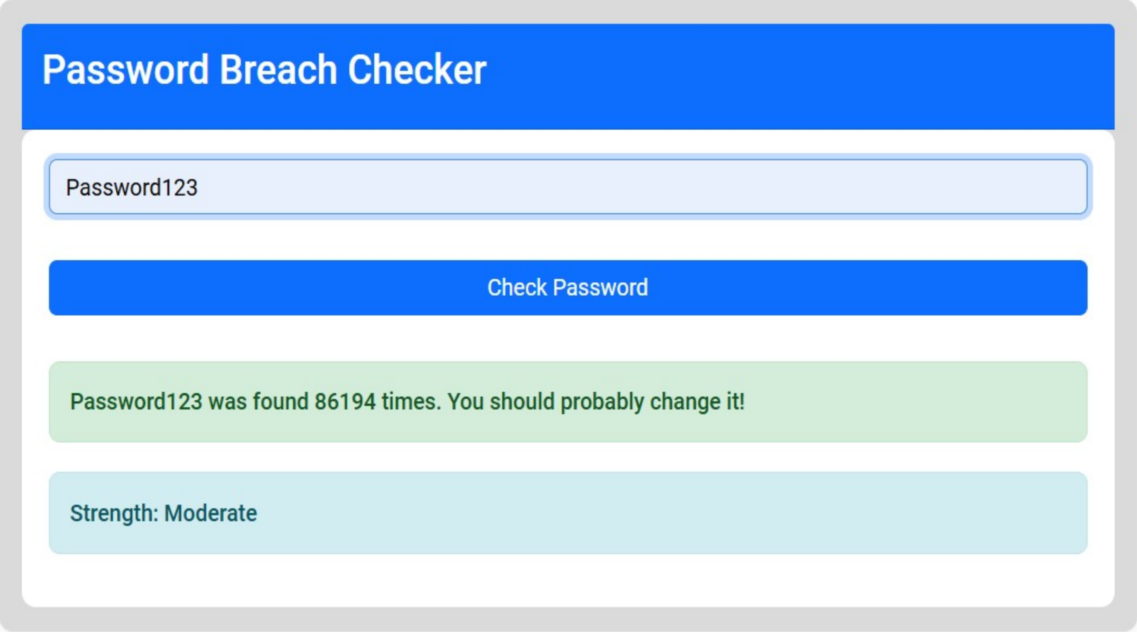
New Password Generator

9

☑ Include Lowercase Letters ☑ Include Uppercase Letters
☑ Include Numbers ☑ Include Symbols

Generated Password:]MtrEE/q@

Figure1. Home Screen



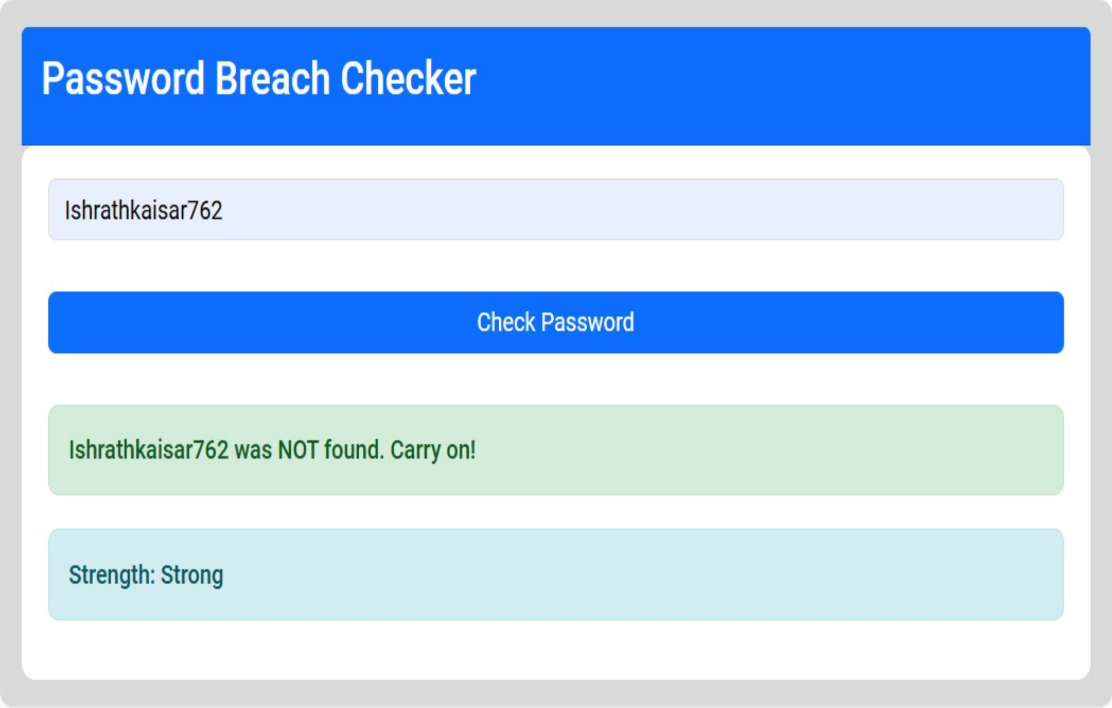
Password Breach Checker

Password123

Password123 was found 86194 times. You should probably change it!

Strength: Moderate

Figure 2. Password Breach Checker



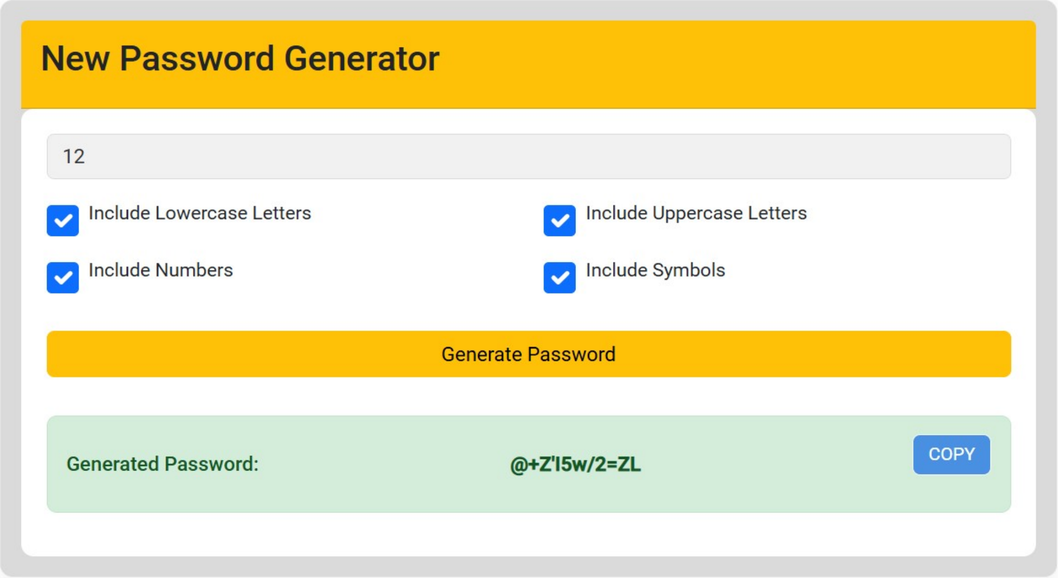
Password Breach Checker

Ishrathkaisar762

Ishrathkaisar762 was NOT found. Carry on!

Strength: Strong

Figure 3. Password Breach Checker



New Password Generator

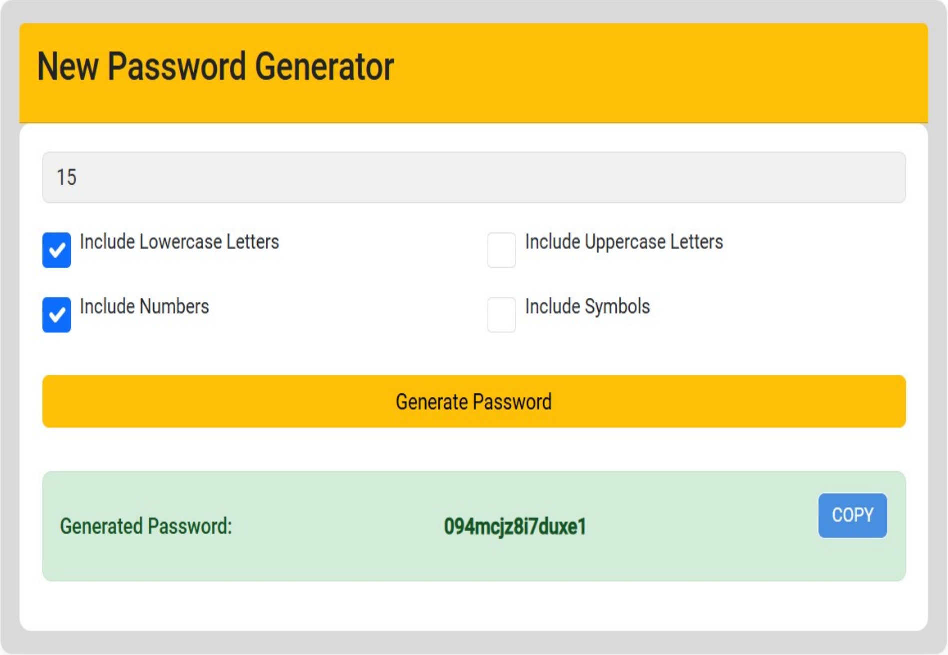
12

☑ Include Lowercase Letters ☑ Include Uppercase Letters

☑ Include Numbers ☑ Include Symbols

Generated Password: @+Z'I5w/2=ZL

Figure 4. New Password Generator



New Password Generator

15

☑ Include Lowercase Letters □ Include Uppercase Letters

☑ Include Numbers □ Include Symbols

Generated Password: 094mcjz8i7duxe1

Figure 5. New Password Generator

8. Conclusion

The Password Vulnerability Assessment project effectively addresses the pressing need for enhanced password security by utilizing online breach data and hashing techniques. It provides a simple to use and dependable way to assess a password's security[9]. Through the integration of Flask for backend development and the use of hacked password datasets, the tool allows users to detect weaker or compromised passwords and enhance their cybersecurity procedures in general[15]. For today's safety concerns, our technique emphasises the value of combining efficient algorithms with intuitive user interfaces.

9. References

[1] Hunt, T., & Gibson, R. (2020). Pwned Password API: Enhancing Breach Detection with Privacy. Journal of Cybersecurity Research, 12(3), 101-118.

[2] Lee, H., & Martinez, P. (2021). Secure Password Management: Bridging Usability and Security. International Journal of Information Security, 8(2), 45-67.

[3] Allen, M., & Carter, B. (2022). The Role of Hashing Algorithms in Password Security. Journal of Information Technology, 18(1), 23-34.

[4] Rogers, L., & Kim, J. (2023). Advanced Password Protection Mechanisms: A Comparative Study. Cybersecurity Journal, 19(4), 75-89.

[5] J. Bonneau, “The science of guessing: Analyzing an anonymized corpus of 70 million passwords,” *2012 IEEE Symposium on Security and Privacy*, San Francisco, CA, USA, 2012, pp. 538-552.

[6] D. Malone and K. Maher, “Investigating the distribution of password choices,” *Proceedings of the 21st International Conference on World Wide Web (WWW '12)*, Lyon, France, 2012, pp. 301-310.

[7] W. Han, L. Chang, N. Li, and X. Gong, “Security analysis of password expiration policies,” *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS ’19)*, London, UK, 2019, pp. 1062-1076.

[8] P. Golla and R. Dey, “A study of password re-use and adaptive password strength meters,” *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS ’18)*, Toronto, Canada, 2018, pp. 145-158.

[9] M. Weir, S. Aggarwal, M. Collins, and H. Stern, “Testing metrics for password creation policies by attacking large sets of revealed passwords,” *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS ’10)*, Chicago, IL, USA, 2010, pp. 162-175.

[10] T. Dell’Amico and M. Filippone, “Monte Carlo strength evaluation: Fast and reliable password checking,” *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS ’15)*, Denver, CO, USA, 2015, pp. 158-169.

[11] J. Ma, W. Yang, M. Luo, and N. Li, “A study of probabilistic password models,” *2014 IEEE Symposium on Security and Privacy*, San Jose, CA, USA, 2014, pp. 689-704.

[12] T. Acar, M. Belenkiy, and A. Küpçü, “Single password authentication,” *Computer Networks*, vol. 57, no. 15, pp. 3207-3221, Oct. 2013.

[13] H. Wang, L. Wu, M. Li, and W. Zhou, “LeakChecker: An approach to detect password leaks,” *IEEE Access*, vol. 7, pp. 30986-30995, 2019.

[14] D. Das, N. F. Sam, and J. M. Pujol, “Examining password re-use across online services,” *2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, 2019, pp. 1-6.

[15] C. Herley and P. C. van Oorschot, “A research agenda acknowledging the persistence of passwords,” *IEEE Security & Privacy*, vol. 10, no. 1, pp. 28-36, Jan.-Feb. 2012.

[16] W. Melicher, B. Ur, S. Segreti, and D. Wang, “Fast, lean, and accurate: Modeling password guessability using neural networks,” *Proceedings of the 26th USENIX Security Symposium*, Vancouver, Canada, 2017, pp. 175-191.

[17] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang, “The tangled web of password reuse,” *Proceedings of the 2014 Network and Distributed System Security Symposium (NDSS ’14)*, San Diego, CA, USA, 2014, pp. 1-16.