Password Strength Analyzer Report

Mohammed Firdaws Alnuur

Future Interns

Nairobi, Kenya

Feb 2025

Declaration and approval

I declare that this report is my original work and has not been previously submitted for approval. To the best of my knowledge, all sources have been cited appropriately

Student Name: Mohammed Firdaws Alnuur

Email: firdawsalnuur4@gmail.com

Abstract

This Report documents the development, implementation, and evaluation of a Password Strength Analyzer Tool, a Python-based application designed to assess and enhance password security. The tool evaluates passwords based on criteria such as length, character variety, and complexity, providing real-time feedback and improvement suggestions to users. It also incorporates SHA-256 hashing to ensure secure handling of passwords. The project features a user-friendly graphical user interface (GUI) built with tkinter, making it accessible to a wide range of users.

The tool was tested with various passwords, demonstrating its effectiveness in identifying weak passwords and offering actionable feedback. However, the study also highlights the limitations of regex-based evaluation, particularly in detecting predictable yet complex passwords. To address this, future improvements could include integrating AI-based models like zxcvbn (Wheeler, 2016) implementing entropy calculation, and adopting advanced encryption algorithms such as Argon2 **(Biryukov et al., 2016a)**

For aspiring cybersecurity professionals, this project provides a foundation in software security, Python programming, and user interface design. Interns can contribute to enhancing the tool by integrating advanced features, optimizing performance, and expanding its functionality.

This project not only provides hands-on experience with Python, regex, and hashing techniques but also encourages innovation in addressing real-world cybersecurity challenges. By building on this foundation, interns can play a pivotal role in creating a more robust and user-centric password security solution.

Chapter 1: Introduction

1.1 Background Information

In the modern digital era, password security is a critical component of protecting sensitive information. Weak passwords are a common vulnerability exploited in cyberattacks, leading to data breaches and unauthorized access (Bonneau et al., 2012)To address this issue, we developed a Password Strength Analyzer Tool using Python, regex, and hashing techniques. This case study documents the development process, methodologies, and evaluation of the tool, highlighting its effectiveness and areas for future improvement.

The tool evaluates password strength based on predefined criteria, provides actionable feedback to users, and enhances security by hashing passwords using the SHA-256 algorithm. Additionally, it features a user-friendly graphical user interface (GUI) for ease of use.

1.2 Problem Statement

In today's digital age, weak passwords remain a significant vulnerability, often leading to data breaches and unauthorized access to sensitive information **(Florencio & Herley, 2007)** Many users struggle to create strong passwords due to a lack of understanding of security requirements or reliance on easily guessable patterns (Shay et al., 2010). To address this issue, there is a need for an intuitive and accessible tool that evaluates password strength, provides actionable feedback, and enhances security through encryption. The Password Strength Analyzer Tool aims to solve this problem by offering a user-friendly solution that assesses passwords based on length, character variety, and complexity, while also educating users on how to improve their password security. By combining regex-based analysis, hashing techniques, and a graphical interface, this tool empowers users to create stronger passwords and protect their digital assets effectively.

1.3 Objectives

1.3.1 General Objective

The primary goal of this project is to develop a Password Strength Analyzer Tool that evaluates the strength of user passwords, provides actionable feedback for improvement, and enhances security through encryption. The tool aims to educate users on creating strong passwords while offering a user-friendly interface for seamless interaction.

1.3.2 Specific Objectives

To evaluate Password Strength by developing a system to assess passwords based on criteria such as length, character variety (uppercase, lowercase, numbers, and special characters), and complexity.

To provide Real-Time Feedback by offering constructive suggestions to users for improving weak or moderate passwords (Kelley et al., 2012)

To enhance Security while implement SHA-256 hashing to securely handle passwords and prevent exposure of plaintext passwords.

To create a User-Friendly Interface by designing an intuitive graphical user interface (GUI) using tkinter to ensure ease of use for all users.

To test and Validate with various password scenarios to ensure the tool's accuracy and reliability.

1.4 Scope and Limitations

The Password Strength Analyzer Tool is designed to evaluate password strength based on predefined criteria such as length, character variety, and complexity, providing real-time feedback and improvement suggestions to users. It also incorporates SHA-256 hashing to enhance security. The tool features a user-friendly GUI, making it accessible to a wide audience. However, its current scope is limited to regex-based evaluation, which may not detect all real-world weak passwords, such as predictable patterns or common phrases (Ur et al, 2012) Additionally, the tool does not yet integrate advanced features like AI-based analysis, entropy

calculation, or support for custom password policies. Future updates will aim to address these limitations by incorporating more sophisticated algorithms and expanding functionality.

Chapter 2. Methodology

Introduction

This outlines the methodologies and approaches used in the development of the Password Strength Analyzer Tool**.** The project was executed in a systematic manner, combining software development best practices with cybersecurity principles to create a robust and user-friendly application.

2.1 Technology Stack

The tool was developed using the following technologies:

- **Programming Language**: Python, chosen for its simplicity, readability, and extensive library support.
- **Libraries**:
    - tkinter: For building the graphical user interface (GUI).
    - re: For regex-based password evaluation.
    - hashlib: For implementing SHA-256 hashing to secure passwords.

2.1 Development approach

The project followed an iterative development approach, allowing for continuous testing and refinement. The key steps included:

1. **Requirement Analysis**: Identifying the core functionalities, such as password evaluation, feedback generation, and secure hashing.
2. **Design**: Planning the system architecture, including the GUI layout and the logic for password evaluation.

3. **Implementation**: Writing the code for password analysis, feedback generation, and hashing, followed by integrating the GUI.
4. **Testing**: Validating the tool with various password scenarios to ensure accuracy and reliability.
5. **Deployment**: Packaging the application for ease of use and distribution.

2.3 Password Strength Evaluation Logic

The tool evaluates passwords based on the following criteria:

1. Length: Minimum of 8 characters.
2. Uppercase Letters: At least one uppercase character.
3. Lowercase Letters: At least one lowercase character.
4. Numbers: At least one numerical digit.
5. Special Characters: Inclusion of symbols (e.g., @, #, $).

Each criterion contributes to an overall strength score, classified as:

- **Weak** ❌: Fewer than 3 conditions met.
- **Moderate** ⚠: At least 3 conditions met.
- **Strong** ☑: All conditions met.

2.4 GUI Development

The GUI was designed using tkinter and includes the following components:

- **Password Input Field**: Allows users to enter their password, with an option to show/hide the password.
- **Analyze Button**: Triggers the password evaluation process.
- **Progress Bar**: Visually represents password strength in real-time.
- **Feedback Labels**: Display the strength rating, improvement suggestions, and the hashed password.
- **Tooltip**: Provides tips on creating strong passwords.

2.5 Password Hashing

To enhance security, the tool hashes passwords using the SHA-256 algorithm. This ensures that no plaintext passwords are stored or transmitted, reducing the risk of exposure (Digital Identity Guidelines, 2017)

2.6 Testing and Validation

The tool was rigorously tested with various password scenarios, including:

- Short and simple passwords (e.g., 12345).
- Partially complex passwords (e.g., password1).
- Highly complex passwords (e.g., P@ssw0rd123).
- Edge cases, such as all-lowercase or all-uppercase passwords.

2.7 Limitations and Future Work

While the tool effectively evaluates password strength based on regex, it has limitations in detecting real-world weak passwords, such as predictable patterns or common phrases (Kelley et al., 2012). Future work will focus on integrating AI-based models like **zxcvbn** (Wheeler, 2016) implementing entropy calculation, and adopting stronger encryption algorithms like Argon2 (Biryukov et al., 2016b)

This methodology ensured the development of a functional and reliable tool while laying the groundwork for future enhancements.

References

Biryukov, A., Dinu, D., & Khovratovich, D. (2016a). Argon2: New Generation of Memory-Hard Functions for Password Hashing and Other Applications. *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, 292–302. https://doi.org/10.1109/EuroSP.2016.31

Biryukov, A., Dinu, D., & Khovratovich, D. (2016b). Argon2: New Generation of Memory-Hard Functions for Password Hashing and Other Applications. *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, 292–302. https://doi.org/10.1109/EuroSP.2016.31

Bonneau, J., Herley, C., Oorschot, P. C. V., & Stajano, F. (2012). The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. *2012 IEEE Symposium on Security and Privacy*, 553–567. https://doi.org/10.1109/SP.2012.44

Digital Identity Guidelines (Nos. 800-63B). (2017). National Institute of Standards and Technology. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63b.pdf

Florencio, D., & Herley, C. (2007). A large-scale study of web password habits. *Proceedings of the 16th International Conference on World Wide Web*, 657–666. https://doi.org/10.1145/1242572.1242661

Kelley, P. G., Komanduri, S., Mazurek, M. L., Shay, R., Vidas, T., Bauer, L., Christin, N., Cranor, L. F., & Lopez, J. (2012). Guess Again (and Again and Again): Measuring Password Strength by Simulating Password-Cracking Algorithms. *2012 IEEE Symposium on Security and Privacy*, 523–537. https://doi.org/10.1109/SP.2012.38

Ur et al. (2012). *21st USENIX Security Symposium (USENIX Security 12)*. USENIX Association. https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/ur

Wheeler, D. L. (2016). *25th USENIX Security Symposium (USENIX Security 16)*. USENIX Association. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/wheeler