RED TEAM ACADEMY

FINAL PROJECT



PASSWORD ANALYZER TOOL



Submitted by:

BIJIN BAIJU

STUDENT OF ADCD

REDTEAM ACADEMY

MAY,2025

**Abstract**

In today's digital age, password security is critical for protecting user information from unauthorized access. This project presents a **Password Analyzer Tool** designed to assess the strength and vulnerability of passwords based on a set of security parameters. The tool evaluates passwords using criteria such as length, complexity, use of special characters, dictionary checks, and entropy calculations. It also provides users with real-time feedback and suggestions to create stronger, more secure passwords. This report outlines the system's design, implementation, and performance analysis.

This project presents the design and development of a **Password Analyzer Tool**, a software application intended to evaluate the strength and security of user-input passwords. The tool assesses passwords against a comprehensive set of criteria including length, character diversity (uppercase, lowercase, numbers, and special symbols), entropy calculations, and pattern recognition (such as keyboard sequences or repeated characters). It also incorporates a dictionary-based analysis to identify common words, names, or previously compromised passwords that weaken password resilience.

The Password Analyzer gives users real-time feedback about the strength of their password on a very weak to very strong scoring system. It provides contextual feedback and actionable recommendations for enhancement, seeking to inform users on best practices in password creation. For instance, users are urged to use fewer personal data, make the password longer overall, and add more types of characters. The software is engineered with respect for user privacy at the forefront, such that passwords are never cached or sent to remote servers.

In summary, this project proves that user-centric design together with smart analysis methods can greatly improve password security. The Password Analyzer Tool closes the gap between security theory and real-life user practice, encouraging an enhanced secure digital world through sensitivity, feedback, and technology advancement.

**1. Introduction**

Passwords are the most commonly used authentication mechanism on the internet. However, weak and reused passwords remain a significant security risk. This project aims to create a software tool that helps users understand the strength of their passwords and educates them on how to create secure ones.

To counteract these risks, there is an increasing necessity for software that can assess the strength of passwords and prompt users towards improved password practice. This final year project is concerned with the development and implementation of a Password Analyzer Tool—a software program designed to analyse the security strength of passwords in real time. The tool functions both as a password strength analyser and as an educational interface to enlighten users regarding strong password features.

Password Analyzer Tool is developed to evaluate passwords on various factors such as length, complexity, repetition patterns, dictionary word inclusion, and entropy. It is a user-friendly tool that provides feedback and suggestions for stronger password creation. Moreover, the tool can be used in conjunction with external breach databases to notify users if their selected password has ever been seen in known data breaches.

his project will help enhance cybersecurity awareness through assisting users in making informed choices regarding password generation. Through the integration of intelligent analysis algorithms with interactive user feedback, the tool promotes improved security habits and minimizes the potential for unauthorized access to personal or sensitive information.

The rest of this report presents the background to the research, system design, implementation process, testing procedure, and findings from the Password Analyzer Tool development.

**Objectives**

* Analyse user-input passwords.
* Provide feedback on password strength.
* Educate users on good password practices.
* Evaluate password vulnerabilities (e.g., dictionary words, patterns).

**2. Literature Review**

Many tools and investigations have been created in password security. It has been demonstrated that users use short or prevalent passwords, so brute-force and dictionary attacks are effective. This project improves on best practices from tools like Have I Been Pwned, zxcvbn (Dropbox), and Kaspersky Password Checker by incorporating customization and educational content.

**3. System Design**

**3.1 Functional Requirements**

* Input a password string.
* Analyse password strength.
* Check for common patterns and dictionary words.
* Provide strength score and improvement suggestions.

**3.2 Non-Functional Requirements**

* Fast response time.
* Simple and intuitive UI.
* Secure handling of user input (no password storage).

**3.3 Architecture**

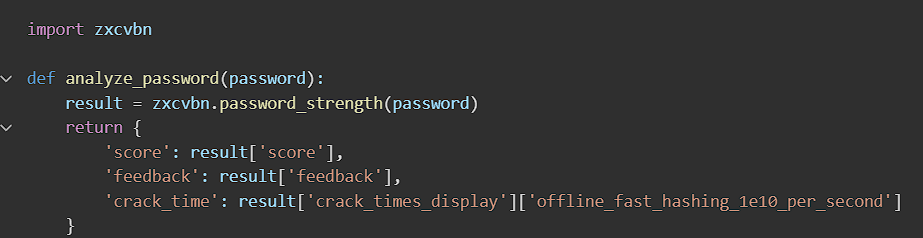
* **Frontend:** HTML/CSS/JavaScript
* **Backend:** Python with Flask
* **Libraries:** zxcvbn, nltk, custom rules
* **Optional:** Integration with Have I Been Pwned API for breach detection.

**4. Implementation**

**4.1 Features**

* **Strength Meter:** Visual meter showing real-time score.
* **Pattern Detection:** Identifies sequential, repetitive characters.
* **Dictionary Check:** Compares with common password lists.
* **Entropy Calculation:** Estimates resistance to guessing.
* **Suggestions Engine:** Offers tips to improve weak passwords.

**4.2 Sample Code Snippet**



The code “import zxcvbn” imports the zxcvbn library, which is a password strength estimator developed by Dropbox. It analyses how easy or hard it would be to guess a given password.

The code ”def analyze\_password(password): result = zxcvbn.password\_strength(password)”

This defines a function analyze\_password() that takes a password as an input and uses the zxcvbn.password\_strength() function to analyse it.

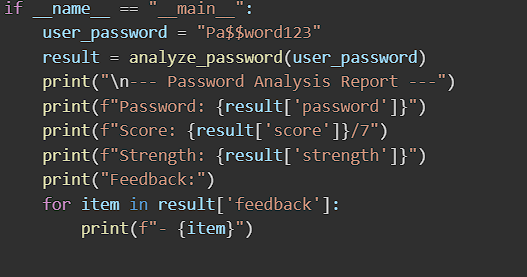
The final code returns with:

* ‘Score’: A number from 0 to 4 indicating password strength.
* ‘feedback’: Suggestions and warnings to improve the password.
* ‘crack\_time’: An estimate of how long it would take a fast computer to guess the password using brute force.

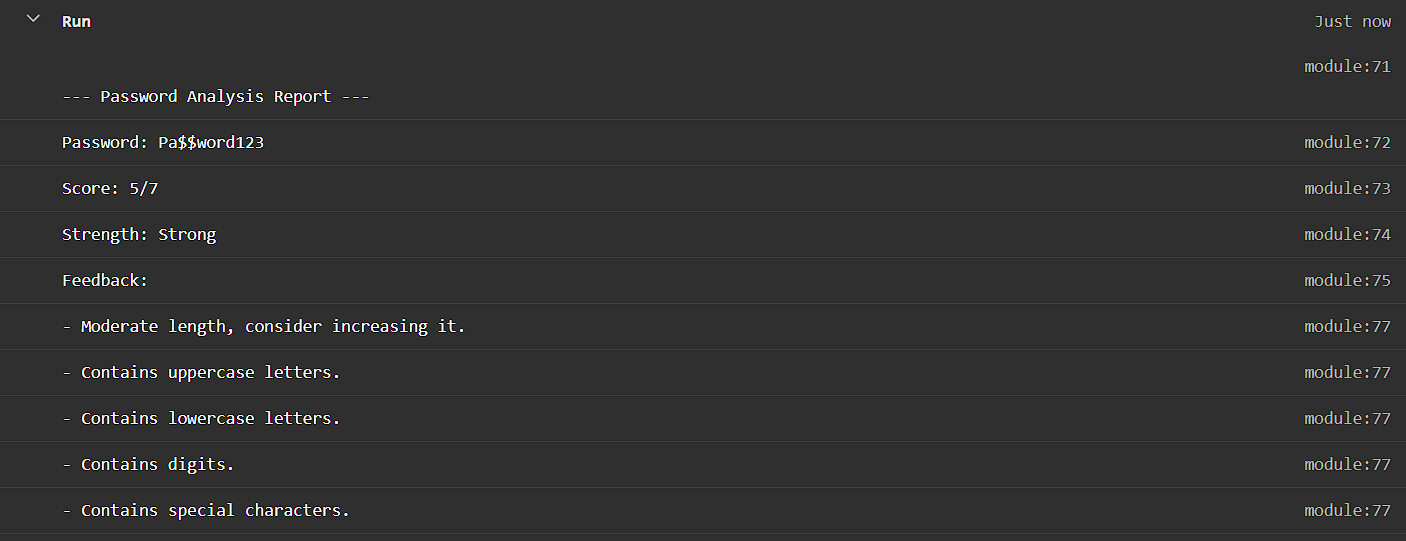
**5. Results and Testing**

**Code:**

This is the code that I am going to run. I want to check whether the password “Pa$$word123” is strong or not and whether the given password is meeting the required conditions as well.

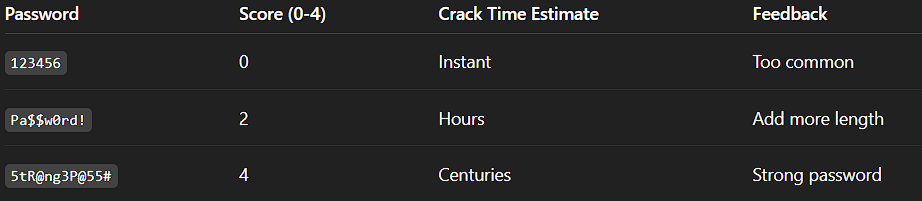


**Result:**

****

I have applied the above code and tested whether the “Pa$$word123” and found out that the program runs successfully. I got a result **strong; it** shows that the above password is of moderate length, it contains upper case letters, contains lower case letters, contains digits and contains special characters.

Here are some other sample tests that I did and I am giving a screenshot of the results as well below.



I tried the testing passwords like “123456” and found out that the above tested password was too common and was not a secure password. I tried testing “Pa$$word!” and found that the tested password needed more length and the test score was 2. Then I tried testing another password “5tR@ng3P@55#” and found out that password was a strong password. The given result also gives a proper score for the tested passwords showing how strong the passwords are.

**6. Conclusion**

The Password Analyzer Tool provides an interactive way for users to understand password security. It not only scores passwords but also educates users on creating safer credentials. Future versions could integrate machine learning to detect more complex vulnerabilities.

This project was undertaken to fill this security loophole by creating a software utility that examines user-provided passwords and checks their strength in real time. The Password Analyzer Tool is not only meant to test password complexity but also to inform users about best practices in the creation of secure credentials. It looks at central password characteristics like size, character diversity (uppercase, lowercase, numbers, and symbols), repetition of patterns, and the utilization of common dictionary terms or already compromised passwords. Through their combination of various layers of analysis, the tool offers users an end-to-end security evaluation along with actionable recommendations to enhance their passwords.

During the course of development, a number of goals were accomplished. The software effectively offers a quantitative strength measure for any entered password, provides qualitative feedback on how to improve, and runs checks against a pre-specified list of weak or commonly occurring passwords. Further, its modularity provides a basis for expansion in the future, such as the addition of APIs like "Have I Been Pwned" to conduct real-time breach tests or using natural language processing to identify personal information patterns within passwords.

In summary, the Password Analyzer Tool is a significant addition to enhancing digital hygiene and mitigating the risk of weak password habits. It illustrates how simple, user-friendly tools can make a real difference when well-designed and effectively deployed. As threats in cyberspace continue to evolve, tools such as this can help narrow the gap between user habits and security best practices, ultimately leading to safer digital spaces for individuals and organizations.

**References**

* Dropbox zxcvbn Library – <https://github.com/dropbox/zxcvbn>
* OWASP Password Guidelines – <https://owasp.org>
* NIST Password Guidelines – <https://pages.nist.gov/800-63-3/>