# PasswordStrengthAnalyzer

# Final Report

**Cyber SecurityAssignment-1**

**Name:Laxmi Prasanna. Bonala**

**Date: 1/9/2025**

**GithubRepository:** **https://github.com/prasannabonala/\_\_password\_\_strength--analyser-**

# PasswordStrenthAnalyzer–Final Report

## Project Overview

This project implements a **Password Strength Analyzer** in Python to evaluate password security. It uses the **zxcvbn** library to measure password entropy, resistance to dictionary attacks, and estimated crack time, and optionally secures the password using **bcrypt hashing**.

The project is built entirely in a Google Colab Notebook (.ipynb) for simplicity and portability. It demonstrates **basic cyber security principles** like password complexity evaluation and secure storage practices.

## Technologies&ToolsUsed

* **Python 3.10+**
* **zxcvbn**: To evaluate password strength and entropy.
* **bcrypt**: To hash passwords securely for storage.
* **Google Colab** : For easy execution in the cloud or locally.
* **GitHub**: For version control and sharing the project.

## Work Flow

1. User enters a password in the notebook.
2. The password is analyzed using zxcvbn for:

* Strength score (0–4)
* Estimated crack time (online/offline)
* Suggestions for improvement

1. User optionally hashes the password using bcrypt.
2. Results are displayed directly in the notebook.

## Security Features

 **Entropy-based scoring**: Estimates password strength based on randomness and predictability.

 **Dictionary Attack Detection**: Flags weak or common passwords.

 **bcrypt Hashing**: Offers industry-standard hashing with salt for secure storage.

## Folder Structure

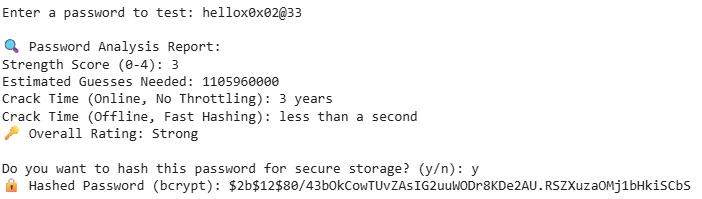
## \_\_password\_\_strength--analyser-/

## │── Password\_Strength\_Analyzer.ipynb # Main notebook

## │── README.md # Project documentation

## Screenshots

1. **Sample output:**

****

## Testing &Results

| **Test Password** | **Score** | **Crack Time (Online)** | **Feedback** |
| --- | --- | --- | --- |
| 123456 | 0 | < 1 second | Common password; needs complexity |
| hello123 | 1 | < 1 second | Add uncommon words |
| h@rd2Gue$$Pa55 | 4 | Centuries | Strong password |
|  |  |  |  |

**Observation:** Passwords with mixed case, symbols, and unpredictable words achieve a strong score (3–4) and longer estimated crack times.

## Deliverables

* Colab Notebook containing all code
* GitHub repository with documentation
* Security report (this document)

## Learning Outcomes

1. Practical understanding of password entropy and strength scoring.
2. Hands-on experience with password hashing.
3. Familiarity with Python security libraries (zxcvbn, bcrypt).
4. Improved ability to evaluate password security metrics.

## 

## Conclusion

This project demonstrates how to evaluate password security using **open-source libraries** and **best practices**. It provides clear insights into password cracking risks and how to mitigate them using **strong password policies** and **secure hashing**.

## 