Project Documentation

Password Strength Evaluation System

1.Project Overview:

Passwords represent one of the most fundamental pillars of internet security, as they are required for access to nearly every online service and digital activity. Over time, numerous tools and techniques for cracking passwords have been developed, such as Hashcat and John the Ripper, which, although powerful, are now being supplemented by more advanced probabilistic modeling techniques, such as Probabilistic Context-Free Grammar (PCFG) approaches. Consequently, a significant portion of cybersecurity research focuses on password creation, storage, and protection, including hashing, encryption, and authentication mechanisms.

Most modern online platforms enforce strict, rule-based password policies, including popular frameworks like zxcvbn; however, these rules can still result in predictable patterns that weaken security, leading users to create passwords that remain vulnerable to automated attacks. In this context, weak passwords are those that are easily guessable by algorithmic methods rather than relying on human intuition.

This research proposes a novel, reliable, and accurate method for evaluating password strength based on real-world cracking techniques. Our approach incorporates extensive feature engineering applied to the RockYou leaked password dataset, consisting of over 14 million passwords, followed by the training of an XGBoost regression model to assess password strength using more than twenty extracted features.

2.Project Objectives:

The main objectives for this project are:

- Evaluating password strength using machine learning methods.
- Educating users about password security.

3.Scope of the project:

- Frontend web application using React.
- Password input and Strength Scoring interface.
- Informational section about the project.
- About us page featuring the members of the project.
- Detailed methodology of how the project works.
- Selection of related datasets (RockYou).
- Data cleaning for the dataset.
- Extensive feature engineering (ZXCVBN).
- Usage of dictionary features based on existing repositories of similar causes.
- Usage of unsupervised learning using PCFG cracker tool.

- Training of XGBoost model (supervised learning).
- Obtaining Score for the password strength evaluation.
- API creation.

4.Functional Requirements

4.1 Password input:

- The interface system provides you with an input field where you can enter the password to get evaluated.
- The input shall allow free-text entry with some restrictions such as exclusion of white spaces in the password.
- The system does not store or save any passwords used in the process.

4.2 Password evaluation:

- The system evaluates your password using a machine learning scoring model.
- The evaluation is dependent on XGBoost model and Extensive feature engineering using ZXCVBN).
- The system will output a strength score based on learned attack patterns.

4.3 Educational content:

- The system will display a couple of explanatory content on how the project was made, and how to create a strong password. Moreover, it provides the user with justifications of the score.

5. Non-Functional Requirements

5.1 Usability:

- Interface is clean, simple, and user-friendly.
- Navigation through pages is clean and consistent.
- The system can be used through any browser.

5.2 Performance:

- Password evaluation is fast and precise.
- UI is responsive and barely has any delays.

5.3 Security:

- No saving of any personal user data
- No usage of any third-party systems

6.Machine Learning (Backend) Specifications:

6.1 Data Processing:

- Feature extraction.
- Normalization of input features.
- Data cleaning to remove any noisy data or unwanted ones.
- Usage of a large-scale leaked passwords dataset called “RockYou 2009 dataset”.
- Usage of unsupervised learning methods (PCFG cracker tool).

6.2 Model Type:

- Usage of supervised machine learning model.
- Algorithm (XGBoost).

6.3 Evaluation Logic:

- Inspired by PCFG cracker tool.

7.User Interface (Frontend) Specifications:

7.1 Design:

- Clean, modern, and professional.
- Warm colors (red and orange).
- Clear visual hierarchy.
- Animated and interactive UI.

7.2 Layout:

- A header with navigation tabs.
- Main content area with password input.
- Footer with external links.

8. System Architecture:

8.1 Frontend Architecture:

- Framework: React (using JavaScript).
- Styling: CSS.
- Components: Home page, About us page, How it works page.

Each page is developed independently on the Github repository.

8.2 Backend Architecture:

- dataset
- Xgboost model

Each one is developed independently on the Github repository.

9.Technology Used:

- Frontend: React.js
- Styling: CSS
- Machine learning: XGBoost
- Data: RockYou dataset 2009
- Research Reference: pcfg cracker documentation – xgboost documentation

10.Assumptions and Constraints:

10.1 Assumptions:

- Users understand basic password concepts.
- Users are entering passwords for educational purposes only.

10.2 Constraints:

- No authentication feature.
- Evaluation is limited to educational evaluation only.

11.Test:

11.1 Test Set Performance Metrics:

MSE: 1.255565

RMSE: 1.120520

MAE: 0.658994

R^2: 0.883196

11.2 Example Predictions:

	Actual	Predicted	Error
0	10.143871	9.678534	0.465337
1	13.098722	12.621970	0.476752
2	12.923680	11.029094	1.894586
3	9.471411	9.094542	0.376870
4	9.115936	9.474842	-0.358906
5	11.522059	11.125375	0.396684

6	11.810172	10.661531	1.148641
7	7.941182	8.843090	-0.901908
8	7.403611	7.412285	-0.008675
9	7.485324	7.470798	0.014526

12.Conclusion:

The password evaluation system provides a realistic and educational approach to understanding and evaluating password security. Using machine learning algorithms and real-world datasets, the system is able to provide meaningful and significant insights on the strength of your passwords.