Faculty of Management Technology

Information Security (|BINF 711|)

Phase 1: Password Management Application

Submitted by:

Name: Ahmed Hassan	ID: 58-0671	Tutorial: 7
Name: Ziad Ekramy	ID: 58-6936	Tutorial: 7

Supervised by:

Dr. Wagdy Anis

TA Nadeen Hamza

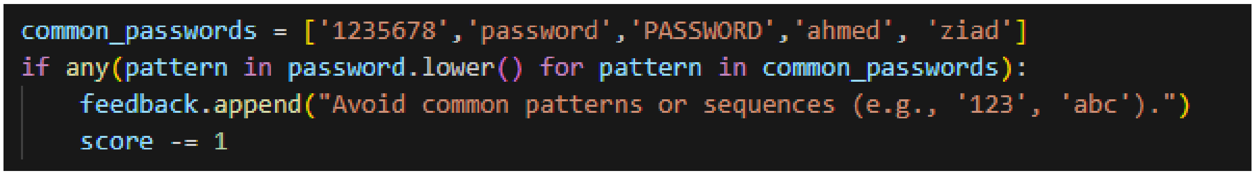
Submission Date:

25/10/2025

Modern password evaluation methods emphasize both usability and resistance to real-world attacks. The National Institute of Standards and Technology recommends that password policies prioritize sufficient length and the exclusion of known-compromised or easily guessed passwords rather than rigid composition rules such as mandatory uppercase letters or symbols. These guidelines aim to enhance security without creating unnecessary complexity for users. Moreover, Ur et al. (2012:4) shows that well designed password strength meters significantly influence user behavior, encouraging users to create stronger and less predictable passwords. Their study also found that users respond positively to clear feedback rather than complexity scores. Furthermore, Wheeler (2016) introduced zxcvbn, a pattern aware password strength estimator that analyzes dictionary words, sequences, and common substitutions to accurately test password strength. These academic articles back up our project's goal: to create a password analyzer that learns from feedback, following both NIST guidelines, and uses a system similar to zxcvbn. This way, we can make passwords that are easy to use but still tough against guessing and brute-force attacks.

Each article provides a different perspective for password evaluation. The National Institute of Standards and Technology (2020) suggests a policy that focuses on password length and preventing password compromise, rather than enforcing complex rules. While these guidelines are practical for system administrators, they do not specify a detailed algorithm for measuring password strength dynamically, which limits their direct implementation in real time evaluation tools. On the other hand, The study by Ur et al. (2012:5) examined how users interact with password meters and found that clear feedback motivates users to create stronger passwords. However, their work primarily addresses human factors and interface design rather than algorithmic assessment, making it more valuable for guiding feedback design than for computing password complexity itself. In contrast, Wheeler (2016) implements the zxcvbn algorithm, which uses pattern matching and real password datasets to estimate the number of guesses an attacker would need to crack a password. This method directly quantifies strength and offers actionable feedback by identifying weaknesses such as dictionary words or predictable sequences.

In our opinion, the best technique for this project is Wheeler's (2016) zxcvbn pattern evaluation. The zxcvbn approach is chosen because it realistically models attacker behavior, provides accurate and understandable strength estimates, and can be efficiently implemented in a lightweight application. Our code implements a method similar to zxcvbn. It identifies common password words to assess and indicate password weakness.



```python
common_passwords = ['1235678','password','PASSWORD','ahmed', 'ziad']
if any(pattern in password.lower() for pattern in common_passwords):
    feedback.append("Avoid common patterns or sequences (e.g., '123', 'abc').")
    score -= 1
```

References:

National Institute of Standards and Technology. (2020). Digital Identity Guidelines: Authentication and Lifecycle Management (SP 800-63B). Gaithersburg, MD.

Ur, B., Kelley, P. G., Komanduri, S., Lee, J., Maass, M., Mazurek, M. L., ... & Cranor, L. F. (2012). How does your password measure up? The effect of strength meters on password creation. In *21st USENIX security symposium (USENIX Security 12)* (pp. 65-80).

Wheeler, D. L. (2016). zxcvbn:{Low-Budget} password strength estimation. In *25th USENIX Security Symposium (USENIX Security 16)* (pp. 157-173).