Major Project Report

On

Password Strength Checker

Submitted By:

V.Aakanksha

Submitted To:

Externsclub

ABSTRACT

In the digital age, safeguarding sensitive information has become a paramount concern, with passwords serving as the first line of defense against unauthorized access. A password strength checker is a tool designed to evaluate the robustness of user-created passwords by analyzing key attributes such as length, complexity, and unpredictability. This tool employs algorithms to assess the inclusion of uppercase and lowercase letters, numerical digits, and special characters, as well as identifying patterns, dictionary words, and repetitions that weaken passwords. By providing real-time feedback and actionable suggestions, a password strength checker educates users on creating stronger passwords and mitigates the risk of Cyberattacks like brute force and dictionary attacks. This abstract highlights the importance of integrating password strength checkers into authentication systems to enhance Cybersecurity and promote user awareness.

CONTENTS

INTRODUCTION

In today's interconnected digital landscape, passwords are a critical component of securing personal and organizational information. However, weak and easily guessable passwords remain a significant vulnerability, making systems prone to unauthorized access, data breaches, and Cyber Attacks. The need for stronger password practices has led to the development of password strength checkers—tools designed to evaluate and improve the security of user-generated passwords.

A password strength checker works by analyzing the characteristics of a password, including its length, complexity, and unpredictability. It examines the use of uppercase and lowercase letters, numbers, special characters, and the presence of common patterns or dictionary words. Based on this evaluation, the tool assigns a strength score or level (e.g., weak, medium, strong) and often provides feedback on how to create a more secure password.

FEATURES

Strength Analysis:

Grades passwords as:

- Weak
- Strong
- Moderate

Validation Criteria:

i. Minimum and maximum length.

ii. Contains uppercase and lowercase letters.

iii. Includes numbers and special characters.

iv. Avoids dictionary-based common words (optional feature).

Feedback and Suggestions:

a. Highlights missing elements in weak passwords.

b. Suggests how to improve password strength.

Customizability:

Rules for password strength can be customized.

Requirements

Python Version: 3.6 or higher.

Libraries:

re (built-in) – for pattern matching and validation.

Usage

1. Run the Program:

Launch the Python script in your terminal or IDE:

python password_checker.py

2. Input a Password:

Enter a password when prompted.

The program will evaluate its strength and display results.

3. Output:

Password Strength: Weak, Moderate, or Strong.

Feedback on how to improve (if needed).

Example Output:

Enter your password: P@ssw0rd

Strength: Strong

Your password meets all criteria.

Implementation Details

The Password Strength Checker evaluates a password based on:

1. Length: Minimum of 8 characters and a maximum of 16 characters.

2. Uppercase Letters: At least one uppercase letter.

3. Lowercase Letters: At least one lowercase letter.

4. Numbers: At least one numeric digit.

5. Special Characters: At least one special character (e.g., @, #, $).

6. Common Patterns: Checks if the password contains common words or patterns (e.g., 1234, password).

CODE

```
import re

def check_password_strength(password):
    # Criteria for validation
    min_length = 8
    max_length = 16
    # Initialize strength
    strength = "Weak"
    # Check length
    if len(password) < min_length:
        return strength, "Password too short! Minimum 8 characters required."
    if len(password) > max_length:
        return strength, "Password too long! Maximum 16 characters allowed."
 # Define regex patterns
    has_uppercase = bool(re.search(r'[A-Z]', password))
    has_lowercase = bool(re.search(r'[a-z]', password))
    has_digit = bool(re.search(r'[0-9]', password))
    has_special = bool(re.search(r'[!@#$%^&*(),.?":{}|<>]', password))
```

```
 # Evaluate strength
    if has_uppercase and has_lowercase and has_digit and has_special:
        strength = "Strong"
    elif has_uppercase or has_lowercase or has_digit or has_special:
        strength = "Moderate"
 # Feedback
    feedback = []
    if not has_uppercase:
        feedback.append("Add at least one uppercase letter.")
    if not has_lowercase:
        feedback.append("Add at least one lowercase letter.")
    if not has_digit:
        feedback.append("Include at least one number.")
    if not has_special:
        feedback.append("Use at least one special character.")
    return strength, " ".join(feedback) if feedback else "Your password is strong!"
# Input from user
password = input("Enter your password: ")
```

```
strength, feedback = check_password_strength(password)
print(f"Strength: {strength}")
print(feedback)
```

Conclusion

The Password Strength Checker is a helpful tool to promote better password practices. It is highly customizable and can be extended with additional features such as GUI integration, API support, and advanced security checks.