International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048
Volume 7, Issue 12, pp 203-208, December-2023
https://doi.org/10.47001/IRJIET/2023.712028

An Application-Based Tool That Contains Both an Enhanced Password Generator and a Password Strength Checker

[1]Hazam Hamood Al-Zakwani, [2]Dr. Ramesh Palanisamy

[1,2]Department of Information Technology, University of Technology and Applied Sciences – Ibra, Sultanate of Oman
Authors E-mail: [1]36s1984@utas.edu.om, [2]ramesh.palanisamy@utas.edu.om

***Abstract* - The most common user authentication method for restricted resource access has been passwords. The fundamental problem with passwords is their strength or quality, or how easily or difficult they can be "guessed" by an outsider wishing to gain access to a resource you have access to by impersonating you. In this article, we examine multiple metrics related to password quality, one of which we have also suggested, and evaluate their advantages, disadvantages, and connections. We also experimented with cracking a series of passwords of varying complexity. The results of the experiments show that the quality of the passwords and their guess ability are positively correlated.**

Keywords: Password Strength Checker, Enhanced Password Generator, Python Tool, Password Security, Password Complexity, Weak Passwords, Strong Passwords.

Introduction

Combining an Improved Password Generator with a Password Strength Checker, this Python-based tool provides a comprehensive approach to password security. The Password Strength Checker analyzes passwords entered by users, determining their strength based on parameters such as length, character diversity, and complexity, and classifying them as "Weak," "Moderate," or "Strong." When a password is weak or moderate, the Enhanced Password Generator suggests stronger alternatives by modifying characters, adding symbols, capitalizing, and adding other enhancements. This makes it easier to create strong and secure passwords. Improved password practices and increased user awareness of online security are the two main goals of the project. Two tools in one: the enhanced password generator and the password strength checker, both based in Python. While separate tools such as password generators and password strength checkers are available, combining these functions into a single solution gives users a more comprehensive approach to password security. Users can now assess the strength of their passwords and receive personalized recommendations for creating stronger and more secure passwords—all within a cohesive platform—thanks to this combination. This integrated approach seeks to raise user awareness of security while streamlining and improving password management procedures.

The main method of protecting and authenticating sensitive and personal data is passwords. Nonetheless, a lot of users create weak or simple passwords, making their accounts susceptible to unauthorized access and potential security breaches. A user-friendly solution that assesses password strength and generates stronger alternatives is necessary, as evidenced by the lack of awareness surrounding password complexity and strength. In order to address the problems associated with weak passwords, this mini project introduces a unified Python-based tool. With the use of criteria like length, character variety, and complexity, the Password Strength Checker assesses submitted passwords by determining their strength. It gives users an understanding of the security level of their passwords by classifying them as "Weak," "Moderate," or "Strong." When people believe that passwords are weak or moderate, the Improved Password Through the use of several transformations, such as capitalization, character manipulation, and symbol inclusion, Generator generates customized suggestions for stronger passwords. Through the integration of these features into a cohesive solution, the project seeks to encourage users to generate and keep more secure passwords, thereby bolstering overall online security procedures.

Literature Review

A document, or part of a document, that gathers important sources on a subject and engages those sources in dialogue with one another is called a literature review. Synthesis is another name for it. In many fields, not just literature, the literature review is a vital genre.

Password strength is typically evaluated in English-only environments using well-known password-strength-estimation models, according to systematic literature and password-strength-tool analysis. Studies that translate password strength meters into other languages are scarce. The Czech and Slovak language password's strength was examined in the study

ISSN (online): 2581-3048
Volume 7, Issue 12, pp 203-208, December-2023
https://doi.org/10.47001/IRJIET/2023.712028

conducted by Daucek et al. [21]. The password-strength meter zxcvbn was translated into Czech and Slovak, and the researchers presented their method and findings.

Pattern matching and entropy estimation are combined by the zxcvbn password-strength-estimation model [20] to determine a password's strength. It gives users feedback on how to strengthen their passwords. Passwords are assessed by the zxcvbn model according to several factors, including length, character diversity, and frequently occurring patterns or phrases. Each password is given a score between 0 and 4 by the tool to indicate the level of security provided. According to Golla et al.'s research, the zxcvbn password-strength meter outperformed the Eleven- and LPSE-based password-strength meters. Based on the results of the Spearman correlation, Comprehensive8 performed the worst in the study consequently.

The study conducted by Hong et al. [22] examined passwords written in Korean. Their research indicates that the English alphabet is the basis for both proposed models for password strength estimation and existing password strength meters. The authors developed a password dictionary in the Korean language and suggested an evaluation model for determining the strength of a password for Korean users based on this. Similar to Daucek et al.'s research, Hong et al. used a database of leaked passwords to conduct experiments to assess the security of passwords based on the Korean language. Consequently, the suggested model demonstrated 99.38% accuracy for password leaks based on Korean language. This is better than the current model's 80.06% accuracy.

Objectives

- Optimized Password Protection Be aware: Users' awareness of the significance of using secure passwords has increased.
- Knowledgeable Empowered Users: Inform users about the characteristics that define strong passwords, including length, complexity, and character variety.
- Offer Realistic Solutions: Provide an easy-to-use tool that allows users to determine the strength of their passwords and provides suggestions for creating more secure alternatives.
- Encourage Better Password Habits: Urge users to embrace enhanced password creation and management techniques in order to improve overall online security.
- Bringing Together Functionalities: Create a single, cohesive tool that combines the features of an enhanced password generator and a password strength checker for efficiency and ease of use.

Evaluation of Password Strength:

Researching criteria for defining password strength include length, character variety, and complexity.

Creating algorithms based on these criteria to assess password strength.

Results:

An assuring password robustness function called Password Strength Checker.

Improved Strategies for Password Generation:

Researching techniques for changing passwords in order to make them stronger (capitalization, symbol inclusion, character replacements, etc.).

Using algorithms to generate stronger passwords based on user input.

Results:

An improved password generator feature that offers recommendations for more robust passwords.

Interface for Users

Friendly making the Python-based tool's user interface clear and easy to use.

Developing a user-interactive interactive command-line interface.

Final product

An easy-to-use tool that lets users enter passwords, receive strength assessments, and access strong password recommendations.

Protecting Education and Awareness

Looking up and gathering the best practices for creating and managing passwords.

Creating educational materials to educate users about the importance of password security.

Assumption

The tool incorporates educational materials to increase user awareness regarding strong password practices.

User Involvement and Acceptance

Evaluating and improving the tool's features to ensure precision and effectiveness.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 12, pp 203-208, December-2023

https://doi.org/10.47001/IRJIET/2023.712028

Testing with users and gathering feedback for future improvements.

Assumption

An efficient and fully functional tool that encourages users to create and keep strong passwords.

Input into Cyber security Encouraging users to use the tool and apply stronger password practices.

Helping to improve online security practices overall.

Assumption

A safer online environment is the result of increased user awareness and the adoption of stronger password creation strategies.

Scope and Limitation

Creation and implementation of a Python-based tool that combines an enhanced password generator with a password strength checker. It involves research into password security requirements, password assessment algorithms, and techniques for boosting password strength. The project's goals are to create a useful tool that assesses password strength, informs users about safe password usage practices, and generates suggestions for stronger passwords. The scope also includes developing an intuitive user interface for ease of use and offering educational resources to raise user awareness of password security best practices. The tool's comprehensive testing and deployment may be impacted by the project's limited availability of certain hard- or soft-ware resources. The thoroughness of the analysis and implementation may be limited by the lack of extensive previous research studies or established methodologies on specific password security aspects. The scope of testing and validation of the tool's effectiveness may be restricted by limitations on access to different datasets or real-world password databases. Within the project duration, the extent to which different functionalities can be thoroughly researched, developed, and refined may be impacted by time constraints.

Password Strength Checker Functionality

Evaluates the strength of user-entered passwords based on defined criteria: Length of the password (at least 8 characters). Presence of character variety (lowercase, uppercase, digits, special characters). Overall complexity. Categorizes passwords as "Weak," "Moderate," or "Strong" based on the assessment criteria. Enhanced Password Generator Functionality: Offers suggestions for stronger passwords if the checked password is deemed weak or moderate. Generates enhanced password suggestions by applying various transformations to the user-inputted password: Appending special characters. Capitalizing letters. Replacing characters with symbols. Other manipulations to enhance complexity and security.

Implementation

The model is implemented in Python, utilizing string manipulation functions and the random module. It features a command-line interface, allowing users to interact by entering passwords and receiving strength assessments and enhanced password suggestions in real-time. User input drives the system, triggering the Password Strength Checker and, if necessary, the Enhanced Password Generator to generate suggestions for stronger passwords. This example serves as a proof of concept, demonstrating how the amalgamation of these two functionalities into a single tool can contribute to improving password security and enhancing user awareness of password strength best practices.

Sample code

```
import random
import string
def check_password_strength(password):
    # Length check
    length_score = len(password) >= 8

    # Character variety check
    has lowercase = any(c.islower() for c in password)
    has_uppercase = any(c.isupper() for c in password)
    has_digit = any(c.isdigit() for c in password)
    has_special = any(c for c in password if c in "!@#$%^&*()-
_=+[]{};:'\"\\|,.<>/?")
    variety_score = sum([has_lowercase, has_uppercase,
has_digit, has_special])
    # Overall score
    overall_score = length_score + variety_score

    # Strength assessment
    if overall_score <= 2:
        return "Weak"
    elif overall_score <= 3:
        return "Moderate"
    else:
        return "Strong"
def suggest_strong_password(password):
    # Function to create a strong password suggestion based on
user input
    suggestions = []
    # Transformations - you can modify these as needed
    transformations = [
```


International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 12, pp 203-208, December-2023

https://doi.org/10.47001/IRJIET/2023.712028

```
        lambda s: s + ''.join(random.choice(string.punctuation) for _ in range(2)),
        lambda s: s.title() + str(random.randint(10, 99)),
        lambda s: s.replace('a', '@').replace('o', '0'),
        lambda s: s[::-1]
    ]
    for transformation in transformations:
        suggestion = transformation(password)
        suggestions.append(suggestion)
    return suggestions
if _name_ == "_main_":
    user_password = input("Enter your password to check its strength: ")
    strength = check_password_strength(user_password)

    print(f"The strength of your password is: {strength}")

    if strength == "Weak" or strength == "Moderate":
        strong_password_suggestions = suggest_strong_password(user_password)
            if strong_password_suggestions:
          print("Strong password suggestions based on your input:")
            for suggestion in strong_password_suggestions:
                print(suggestion)
        else:
            print("No suggestions generated.")
```

```
import random
import string

def check_password_strength(password):
    #Length check
    length_score = len(password) >= 8

    #Character variety check
    has_lowercase = any(c.islower() for c in password)
    has_uppercase = any(c.isupper() for c in password)
    has_digit = any(c.isdigit() for c in password)
    has_special = any(c for c in password if c in "!@#$%^&*()-_=+[]{};:'\"\\|,.<>/?")

    variety_score = sum([has_lowercase, has_uppercase, has_digit, has_special])

    #Overall score
    overall_score = length_score + variety_score

    #Strength assessment
    if overall_score <= 2:
        return "Weak"
    elif overall_score <= 3:
        return "Moderate"
    else:
        return "Strong"
```

Figure 1: Checking password strength of user code

```
def suggest_strong_password(password):
    #Function to create a strong password suggestion based on user input
    suggestions = []


    transformations = [
        lambda s: s + ''.join(random.choice(string.punctuation) for _ in range(2)),
        lambda s: s.title() + str(random.randint(10, 99)),
        lambda s: s.replace('a', '@').replace('o', '0'),
        lambda s: s[::-1]
    ]

    for transformation in transformations:
        suggestion = transformation(password)
        suggestions.append(suggestion)

    return suggestions

if __name__ == "__main__":
    user_password = input("Enter your password to check its strength: ")
    strength = check_password_strength(user_password)

    print(f"The strength of your password is: {strength}")

    if strength == "Weak" or strength == "Moderate":
        strong_password_suggestions = suggest_strong_password(user_password)

        if strong_password_suggestions:
            print("Strong password suggestions based on your input:")
            for suggestion in strong_password_suggestions:
                print(suggestion)
        else:
            print("No suggestions generated.")
```

Figure 2: Suggesting password for the user code

```
Enter your password to check its strength: Haz
The strength of your password is: Weak
Strong password suggestions based on your input:
Haz()
Haz53
H@z
zaH
```

Figure 3: Weak Password

```
Enter your password to check its strength: Hazam12
The strength of your password is: Moderate
Strong password suggestions based on your input:
Hazam12&;
Hazam1216
H@z@m12
21mazaH
```

Figure 4: Moderate (Medium) Password

```
Enter your password to check its strength: Omanii122@@
The strength of your password is: Strong
```

Figure 5: Strong Password

```
C:\Users\palanisamy\AppData\Local\Programs\Python\P
Enter your password to check its strength: ABC123
The strength of your password is: Weak
Strong password suggestions based on your input:
ABC123!-
Abc12359
ABC123
321CBA
```

Figure 6: Weak Password

ISSN (online): 2581-3048
Volume 7, Issue 12, pp 203-208, December-2023
https://doi.org/10.47001/IRJIET/2023.712028

Conclusions

To determine if the difficulty of cracking a password is, in fact, related to the strength measures, experiments were conducted to evaluate their password strength metrics by trying to crack the hashes of a limited set of passwords. Dictionary attacks, transformation rules, brute force, and large-scale table lookups are some of the cracking techniques. Finding the password took time, but there were two sorts of results that could be obtained. Results demonstrated a positive correlation between the strength measures and the password cracking success rate. The validity of the strength metrics and their applicability in evaluating the caliber of passwords chosen by users are demonstrated by our experiments. An advanced Password Strength Meter that gives users objective insights about their chosen password has been developed based on the body of knowledge gathered from the empirical study that was conducted to identify weak links. Future research entails comparing the new password strength meter to the current ones through user testing.

REFERENCES

[1] Huang, C.; Chen, S.; Zhang, Y.; Zhou, W.; Rodrigues, J.J.; de Albuquerque, V.H.C. A robust approach for privacy data protection: IoT security assurance using generative adversarial imitation learning. IEEE Internet Things J. 2021, 9, 17089–17097.

[2] Li, H.; Yu, L.; He, W. The impact of GDPR on global technology development. J. Glob. Inform. Technol. Manag. 2019, 22, 1–6.

[3] Kloza, D.; Van Dijk, N.; Casiraghi, S.; Vazquez Maymir, S.; Roda, S.; Tanas, A.; Konstantinou, I. Towards a method for data protection impact assessment: Making sense of GDPR requirements. Policy Brief D. Pia. Lab 2019, 1, 1–8.

[4] Haghshenas, S.H.; Hasnat, M.A.; Naeini, M. A temporal graph neural network for cyber attack detection and localization in smart grids. In Proceedings of the 2023 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 16–19 January 2023.

[5] Inayat, U.; Zia, M.F.; Mahmood, S.; Khalid, H.M.; Benbouzid, M. Learning-based methods for cyber attacks detection in IoT systems: A survey on methods, analysis, and future prospects. Electronics 2022, 11, 1502.

[6] Rastenis, J.; Ramanauskaite, S.; Suzdalev, I.; Tunaityte, K.; Janulevi˘cius, J.; ˙ Cenys, A. Multi-Language spam/Phishing classification ˘by Email Body text: Toward automated security Incident investigation. Electronics 2021, 10, 668.

[7] Ceponis, D.; Goranin, N. Investigation of dual-flow deep learning models LSTM-FCN and GRU-FCN efficiency against single-flow ˘CNN models for the host-based intrusion and malware detection task on univariate times series data. Appl. Sci. 2021, 10, 2373.

[8] Hughes-Lartey, K.; Li, M.; Botchey, F.E.; Qin, Z. Human factor, a critical weak point in the information security of an organization's Internet of things. Heliyon 2021, 7, e06522.

[9] Lal, N.A.; Prasad, S.; Farik, M. A review of authentication methods. Int. J. Sci. Technol. Res. 2016, 5, 246–249.

[10] Yang, W.; Wang, S.; Hu, J.; Zheng, G.; Valli, C. Security and accuracy of fingerprint-based biometrics: A review. Symmetry 2019, 11, 141.

[11] Gwyn, T.; Roy, K.; Atay, M. Face recognition using popular deep net architectures: A brief comparative study. Fut. Internet 2021, 13, 164.

[12] Mehrubeoglu, M.; Nguyen, V. Real-time eye tracking for password authentication. In Proceedings of the 2018 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 12–14 January 2018.

[13] Mahesh, T.R.; Ram, M.S.; Ram, N.S.S.; Gowtham, A.; Swamy, T.N. Real-Time Eye Blinking for Password Authentication. In Proceedings of the International Conference on Intelligent Emerging Methods of Artificial Intelligence & Cloud Computing: Proceedings of IEMAICLOUD 2021, online, 26–29 April 2021.

[14] Juozapavi˘cius, A.; Brilingaite, A.; Bukauskas, L.; Lugo, R.G. Age and Gender Impact on Password Hygiene. ˙ Appl. Sci. 2022, 12,894.

[15] Li, Y.; Liu, Q. A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. Energy Rep. 2021, 7, 8176–8186.

[16] Awad, M.; Al-Qudah, Z.; Idwan, S.; Jallad, A.H. Password security: Password behavior analysis at a small university. In Proceedings of the 2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA), Ras Al Khaimah, United Arab Emirates, 6–8 December 2016.

[17] Katsini, C.; Fidas, C.; Raptis, G.E.; Belk, M.; Samaras, G.; Avouris, N. Influences of human cognition and visual behavior onpassword strength during picture password composition. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, Palais des Congrès de Montréal, Canada, 21–26 April 2018.

[18] Ur, B.; Segreti, S.M.; Bauer, L.; Christin, N.; Cranor, L.F.; Komanduri, S.; Kurilova, D.; Mazurek, M.L.; Melicher, W.; Shay, R.; et al. Measuring real-world

ISSN (online): 2581-3048
Volume 7, Issue 12, pp 203-208, December-2023
https://doi.org/10.47001/IRJIET/2023.712028

accuracies and biases in modeling password guess ability. In Proceedings of the 24th {USENIX} Security Symposium ({USENIX} Security 15), Washington, DC, USA, 12–14 August 2015.

[19] Golla, M.; Dürmuth, M. On the Accuracy of Password Strength Meters. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA, 15–19 October 2018.

[20] Wheeler, D.L. zxcvbn: Low-Budget Password Strength Estimation. In Proceedings of the 25th USENIX Security, Austin, TX, USA, 10–12 August 2016.

[21] Doucek, P.; Pavlíˇcek, L.; Sedláˇcek, J.; Nedomová, L. Adaptation of password strength estimators to a non-english environment the Czech experience. Comput. Secur. 2020, 95, 101757.

[22] Hong, K.H.; Kang, U.G.; Lee, B.M. Enhanced Evaluation Model of Security Strength for Passwords Using Integrated Korean and English Password Dictionaries. Secur. Communicat. Netw. 2021, 2021, 3122627.

[23] Python program to check password strength. (2023, May 13). W3resource. https://www.w3resource.com/python-exercises/cybersecurity/python-cybersecurity-exercise-3.php

[24] Python program to check the validity of a Password. (2017, December 20). Geeks for Geeks. https://www.geeksforgeeks.org/python-program-check-validity-password/

[25] Password strength. (2020, March 21). Wikipedia. https://en.wikipedia.org/wiki/Password_strength

AUTHORS BIOGRAPHY

Hamood Al-Zakwani earned his Advanced Diploma specializing in Cyber Security from University of Technology and Applied Sciences – IBRA. With a keen interest in security measures and penetration testing methodologies, he has dedicated his academic pursuits to understanding and enhancing cybersecurity practices. Currently in the final year of his Advanced Diploma program, Hazam's expertise lies in safeguarding digital environments against cyber threats and creating security tools.
E-mail: alzakwani38@gmail.com

Ramesh Palanisamy obtained his Bachelor's degree from Barathiar University Coimbatore, India. Then, he obtained his Master's degree in Computer Communications from Barathiar University Coimbatore, India, PhD in Computer Science and Engineering. Technical Qualifications CCNA, NSP- (Network Support Professional). HNA- (Hardware Networking Administrator). CCSI - (Cisco Certified System Instructor). He has published in many international journals and conferences. Currently working as a lecturer in the Department of Information Technology at the University of Technology and Applied Sciences - Ibra Sultanate of Oman.
E-mail: rameshphd26@gmail.com

Citation of this Article:

Hazam Hamood Al-Zakwani, Dr. Ramesh Palanisamy, "An Application-Based Tool That Contains Both an Enhanced Password Generator and a Password Strength Checker" Published in *International Research Journal of Innovations in Engineering and Technology - IRJIET*, Volume 7, Issue 12, pp 203-208, December 2023. Article DOI https://doi.org/10.47001/IRJIET/2023.712028

****

