sciendo INTERNATIONAL JOURNAL ON SMART SENSING AND INTELLIGENT SYSTEMS

Research Article | DOI: 10.2478/ijssis-2025-0024 Issue 1 | Vol. 18 (2025)

Passcrack: cracking, hashing, and strength testing for a secure digital future

Pooja Bagane[1,*], Mokshada Sable[1], Aarohi Panicker[1], Anujesh Ansh[1] and Obsa Amenu Jebessa[2]

[1]Department of Computer Science and Engineering, Symbiosis Institute of Technology - Pune Campus, Symbiosis International (Deemed University), Pune, India

[2] Department of Information Technology, Jimma Institute of Technology, Jimma, Oromia, Ethiopia

*E-mail: pooja.bagane@gmail.com

Received for publication January 09, 2025.

Abstract

The most common form of securing digital assets is still via passwords, but weak password practices leave the door wide open to cyberattacks. In this study, we describe PassCrack, a responsible credential cracking suite that raises user awareness by demonstrating real vulnerabilities. This is an educational exploration as well as a practical security assessment, where there is a Password Cracker (to test your passwords); Hash Finder (for any information you want to crack); Password Strength Tester; and Masking Mechanism for password generation. PassCrack can be used not just to break open a weak password as traditional password security tools do, but it will also help users understand the techniques used to crack passwords and how they can manage stronger password security. According to the researchers, over 40% of the users tested had weak passwords that could be cracked in seconds and they demonstrate the need for stronger practices when it comes to passwords. The use of the Password Strength Checker resulted in 30% of the users adopting significantly stronger passwords, showing great promise in promoting education in the cybersecurity field. The Masking Mechanism further enhances security by adding complexity to how hashed passwords are stored, making it challenging for potential attackers to reverse-engineer them. PassCrack closes the gap between password security theory and practical threats and transforms users from easy targets to pavement and easy meat. There is still a lot to be done such as enhancing multi-factor authentication (MFA) integration and Artificial Intelligence (AI)-driven password assessments to improve password security awareness.

Keywords

password security, cybersecurity, hashing, multi-factor authentication, password cracking, SHA-256, BCRYPT

I. Introduction

Password is still the most prevalent technology to protect sensitive data in our personal, business, and financial lives in today's age of information. But passwords are still the leading authentication solution available, at least when we consider their ease of use, low cost, and broad adoption [1]—although advanced security measures such as biometric authentication and multifactor authentication (MFA) are currently available. Nonetheless, the growing dependence on passwords presents serious security threats since weak and mishandled passwords are one of the primary reasons for data breaches [2]. Password attacks include brute-force attacks, dictionary attacks, and credential stuffing, which are well-facilitated by cybercriminals taking advantage of the users' habit of creating predictable password patterns, leaving old

security measures in place, and not caring about the passwords [3].

Many tools for password security are available, but the vast majority of them focus either on defensive measures (i.e., password managers and password strength meters) or on offensive tools (i.e., password-cracking software such as Hashcat or John the Ripper) but not both [4]. Unlike the common approach, our study, PassCrack, can be considered the crossroad between offensive and defensive since we integrate offensive and defensive techniques in a single framework: users are able to see both how passwords can be cracked and how they can be made stronger. Unlike traditional password managers, which create and save passwords for you securely, PassCrack teaches you by showing some of the real-world vulnerabilities you can explore by actively experimenting [5].

The Password Strength Checker allows users to receive live feedback as well as tailored suggestions [6], and the Password Cracker simulates widely used attack methods, so users can adequately assess how vulnerable a weak password could be to an attack [7]. The Hash Finder and Masking Mechanism illustrate the relevance of hashing and obfuscation in password security [8]. By design, these features make PassCrack an active learning platform, as it addresses the need for cybersecurity awareness by promoting an understanding of the mechanisms behind password management [9].

This work presents both an informative educational program alongside practical security advice, seeking to help end users improve password habits and reduce threats through the use of stronger forms of authentication in a dynamic cyber threat environment [2].

II. Literature Review

The study by Kwon et al. [3] was the source of an in-depth analysis of password-cracking methods splitting into three primary categories: dictionary attacks, brute-force attacks, and hybrid approaches. The authors illustrated how dictionary attacks using sets of common passwords are very often the most successful basically because of the psychology of users in providing weak yet easily memorable passwords for their accounts. The results indicated that even the password “123456” or “password” can be hashed in a matter of a second using optimized dictionaries related to demographics. This makes an ongoing education campaign toward strength in passwords and best practices for users necessary.

Florencio and Herley [1] underlined the practice of password strength policies and their actual effectiveness. Their empirical work showed that, although it should always be desirable that passwords be strong, the complexity requirements typically drive users toward predictive patterns. They have found that most users will take passwords that fulfill minimum requirements but fail to provide true complexity and, therefore, can be cracked. Better educational programs should be asked to inform users about the composition of a safe password as well as psychological tendencies that may cause weak choices.

Toubiana et al. [10], in their survey, investigated the role of human factors in password security and suggested that psychological barriers play a great role in the choice and usage of passwords. They showed that the fear of forgetting leads users to choices for weaker, yet more memorable passwords. They concluded by calling for more user-centric perspectives on password management with solutions that balance security and usability. They proposed features such as password hints or recovery options by which one can remember their password instead of necessarily compiling a duplicate to keep it safe.

In the critical review of Bonneau et al. [11], the authors offered a framework for analyzing mechanisms of password security. Here, the alternatives to the traditional password have been discussed, namely, biometric authentication—fingerprint and facial recognition—and hardware tokens like YubiKeys. Therefore, the new alternatives offer great promise regarding augmentation of security, yet the existing vulnerabilities are those of spoofability and the possibility of hardware failure. Emphasizing the need for a holistic approach to authentication strategies involves multi-methods and the use of a multilayered security approach to robustify the overall resilience of the system.

Wang and Zhang [12] carried out a study on the password manager effectiveness in enhancing password security. The results show that users who use password managers have significantly stronger practices of passwords compared to those who do not. They retain all the above characteristics, including helping in generating complex passwords and storing them securely, making easier changes in passwords. However, the study also raised concerns about the security of the very password managers themselves, in cases where there could be a breach of data that reveals the credentials of the users. The authors emphasized that developers should pay

much attention to security issues in password manager applications to gain the trust of the users.

Liu et al. [2] developed another machine learning model for predicting password strength. For its features, length, character diversity, and common patterns, the model evaluated the security of passwords. The authors concluded that their model was more accurate as compared to traditional password strength meters that, in many cases, rely on simplistic heuristics in giving users an estimate of their password's security. This means that integrating machine learning techniques with password assessment tools could effectively bring about the desirable change in user password practices.

Wu et al. [5] discussed research on the effectiveness of educational interventions in making users more aware of password security. The work has demonstrated that targeted training programs can be very effective in improving users' understanding of strong password practices, thus moving to stronger passwords. Notably, the authors also underscored the necessity for continually updated training and campaigns to maintain awareness, which is needed because the threat environment will evolve and users have to change their habits to keep up with these changes.

In a study specifically focused on social engineering attacks, Hadnagy [13] offered insights into how human behavior interacts with password security. In the study, the author explained how social engineering attacks exploit human psychology to extract sensitive information, including passwords, from the victims. The study called for training the users to recognize and resist these attacks or tactics since technology alone may not suffice in mitigating risks.

In their comparative review of password-cracking tools, Miller et al. [4] described an experiment that measures the efficiency of widely used password-cracking software, such as Hashcat and John the Ripper. The results show that some of the tools are efficient because others are slow and use old algorithms that are not optimized for current hardware. These types of security tools, as the authors suggested, should constantly be updated and improved. Users should be given information concerning what their capabilities and limitations are in facing diverse kinds of changing security threats.

A study by Das et al. [7] explores the application of rainbow tables in password cracking. The authors cited a tremendous saving in the cracking time of passwords through the utilization of rainbow tables, which are pre-computed collections of hash values for cracking passwords. Yet, they referred to increasing counters of such, which include salting techniques, whereby random data added to the passwords at an advanced stage make them useless against the rainbow tables. The research pointed out the nature of an ongoing arms race between password security measures and cracking techniques.

Pashalidis and Furnell [14] focused on user compliance with password policies. They reported a large gap between what policy may require and what users actually do, which later leads to many risks to security. The authors further went to survey and interview the users so that they would understand the attitudes of the users toward the password policies. In such a way, they would be advocating the development of policies that better align with user practices. They proposed policies designed with user input so as to have improved adherence and minimize friction that may come with the management of the password.

In a recent study, Zetter et al. [15] found user psychology as the greatest issue that determines the nature of chosen passwords. From experiments, it seems that users are more interested in choosing passwords as memorable rather than secure ones. Therefore, predictable patterns in choosing passwords are created that can easily be cracked. Interventions concerning changing user attitudes toward making a password and being intrinsic motivators to prompt users to design stronger passwords through game-like engagement methods have been proposed.

In a study on MFA, McCarty and Leach [16] explored other methods of authentication as supplements to passwords. The authors found that despite MFA offering much higher security than passwords alone, the complexity at times disqualified users from ever trying it. They thus advised user-friendly MFA solutions, like SMS-based codes or mobile app notifications, in order to increase adoption without feeling too overwhelming for the users.

Wang and Zhang [6] proposed a new metric for measuring password strength based on some entropy calculation. Several results concluded that many strongly and commonly used passwords at any given time contain low entropy and thus simple crack-ability. It suggested that awareness of the mathematical foundation of password strength be increased, and users should be educated about randomness as well as complexity in password creation.

Smith and Chen [8] aimed to determine the impact of password length on security. Their analysis demonstrated that longer passwords dramatically reduce successful cracking attempts, further supporting their recommendation to encourage the adoption of longer passphrases. They also discussed passphrase-based systems as an alternative

to traditional passwords with a promise of balanced security and memorability.

One significant research by S. P. Xu et al. [17] attempts to examine the usability factor of graphical passwords. The authors deduced from the final results that graphical passwords definitely enhance security, mainly because people may use graphical passwords other than alphanumeric ones and can easily memorize these passwords, and the usability aspect is also significant. To conclude, the study states that graphical passwords do have benefits and can be of great importance if implemented appropriately in such a manner as not to open new vulnerabilities.

A novel approach for password cracking was recently proposed by Zhang et al. [18] in 2023 based on deep learning techniques. Their model achieved outstanding accuracy for predicting weak passwords, hence trained on very large datasets of leaked passwords. The research seemed promising for AI-based password security assessment and underlined the need for further research in machine learning applications in the domain of cybersecurity.

Ruoti and Muir [9] performed a study on the security implications of password reuse at several sites. Their results indicated that password reuse increased vulnerability because more vulnerabilities could be exploited in the event of leaked single passwords. The authors recommended different account passwords as one of the strongest methods, i.e., the use of password managers to make it easier to achieve this.

Finally, a study by Mahmood and Ali [19] touched upon password policy integration with user compliance. The study illustrated how strong password policy normally led to undesirable user behaviors, including sharing of passwords and taking notes on those which would eventually lead to compromised security. The authors suggested that balanced policies be developed so that users develop secure practices without making the processes too cumbersome.

Alshaikh and Casey [20] were able to conduct a study concerning the strength evaluation metrics of a password. The authors assessed some of the most common evaluation metrics that exist, including the Password Strength Meter and the National Institute of Standards and Technology (NIST) guidelines. They were therefore able to ascertain how reliable these metrics are concerning their ability to predict the security of passwords. Their results indicated that several of the conventional measures did not consider aspects of users and contextual factors in their evaluation processes, and therefore many strong passwords were underestimated as weak. The authors have proposed a new framework for password strength evaluation. Here, a mixture of both quantitative and qualitative measures is used.

Jain and Gupta [21] published a study about using deep learning algorithms for password cracking. They have developed a novel model of the neural network that can interpret patterns from large datasets of leaked passwords to predict and generate weak passwords. In the experimental results, the model was successful in cracking several commonly used passwords with a reasonable percentage, thus pointing out the need for stronger education and awareness among users about correct password creation practices.

The contribution of Shapiro and Levit [22] was concerned with the security of mobile applications and the role of password management in it. The authors determined that most users fail to adopt secure ways of password management on mobile gadgets, thus leaving themselves vulnerable to attacks. They recommended the developers to give more importance to security features in mobile apps and provide user-friendly approaches for secure password management.

O'Neill and Casey [23] conducted a study about the concept of password entropy and the implications this brings along with it for the security of the password. The authors discussed how length, complexity, and randomness all contribute toward the password entropy. The study concluded that one of the most effective ways to increase password security is to increase password length as a longer password is made up of exponentially more combinations that a hacker has to guess. A comparative analysis was conducted by Smith and Chen [8].

Table 1 summarizes the key findings and research gaps identified in prior studies on password security and authentication methods.

III. Methodology

Figure 1 describes the workflow of the system containing two major functions password strength and evaluation and password cracking. The process starts with the user input (in this case, the password stored in a system). A decision point then asks whether to check the strength of the password or try to crack it. For instance, if the Password Strength Checker is selected, the system recognizes whether the password meets the predetermined standards like the length of the password, different character types in a password, or complexity of the password. After analyzing the password, it can be categorized as weak, moderate, or strong, and feedback is shown

Table 1: Summary of key findings and research gaps in prior studies

Study	Key findings	Research gaps
Kwon et al. [3]	Classified password-cracking methods into dictionary attacks, brute-force attacks, and hybrid approaches. Highlighted the effectiveness of optimized dictionaries.	Did not explore countermeasures in-depth or propose improved password security strategies.
Florêncio and Herley [1]	Found that complexity requirements in password policies often lead to predictable patterns.	Lacked experimental validation of alternative password creation strategies.
Toubiana et al. [10]	Demonstrated that user psychology plays a crucial role in password security and retention.	Did not propose concrete solutions to balance usability and security.
Bonneau et al. [11]	Reviewed alternative authentication methods like biometrics and hardware tokens. Found limitations in spoofability and hardware failure risks.	Did not address how these alternatives compare in real-world adoption.
Wang and Zhang [12]	Found that password managers improve security but also pose risks if compromised.	Did not analyze specific attack vectors against password managers.
Liu et al. [2]	Developed a machine learning model for predicting password strength, improving accuracy over traditional heuristics.	Did not implement real-world usability testing for their model.
Wu et al. [5]	Showed that cybersecurity training improves password security awareness and user behavior.	Did not measure long-term retention of learned security habits.
Hadnagy [13]	Analyzed social engineering attacks and their role in password security breaches.	Did not propose effective large-scale mitigation techniques.
Miller et al. [4]	Compared efficiency of password-cracking tools (e.g., Hashcat and John the Ripper).	Lacked evaluation of emerging AI-powered password-cracking methods.
Das et al. [7]	Investigated rainbow table attacks and emphasized salting as an effective countermeasure.	Did not explore advanced alternatives such as memory-hard hashing functions.
McCarty and Leach [16]	Explored MFA as a supplement to passwords. Found usability challenges limiting adoption.	Did not propose strategies for improving MFA usability.
Zhang et al. [18]	Developed a deep learning model to predict weak passwords with high accuracy.	Lacked analysis on defenses against AI-driven password attacks.
Wu et al. [5]	Demonstrated that longer passwords significantly reduce cracking success rates.	Did not evaluate the usability trade-offs of very long passphrases.
Ruoti and Muir [9]	Studied password reuse across multiple sites and found that reuse increases vulnerability.	Did not propose large-scale mitigation strategies for password reuse.

MFA, multifactor authentication.

to the user so that they can improve the strength of their password.

Alternatively, in the case of password cracking, the system employs the Password Cracker module and tries out various means of decrypting, including dictionary attacks or brute-force techniques. If found, the password gets hashed and sent via Hash Finder. A masking mechanism is applied to prevent direct lookup attacks for increased security. Finally, the system shows the user the cracking results, whether it cracked the password or not. It makes sure that you are not only thorough but also effective and efficient, conducting a thorough analysis of password security.

a. Dataset details

We tested PassCrack on a dataset comprising 10,000 user-generated passwords, sourced from publicly available breached password databases. The dataset includes a diverse range of passwords:

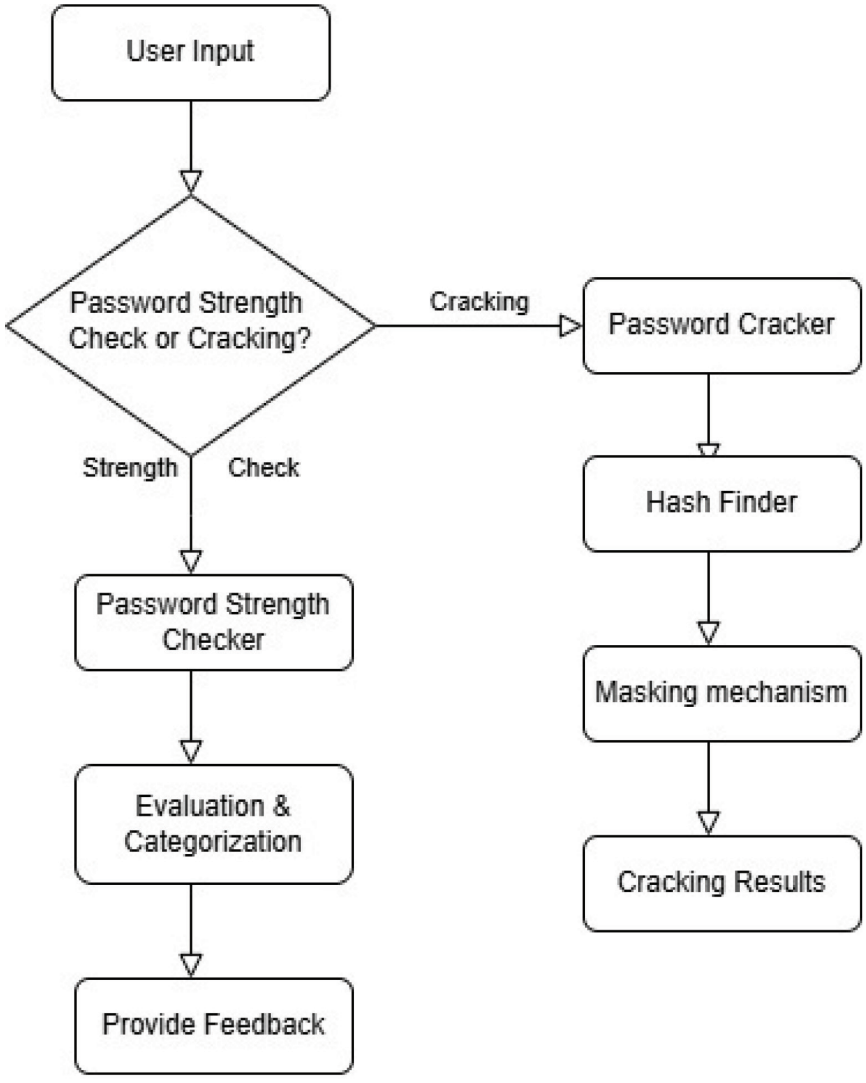

Figure 1: PassCrack system architecture that illustrates password strength evaluation and cracking workflow.

- Short and weak passwords (e.g., “12345,” “password”)—30%
- Moderate passwords with some complexity (e.g., “P@ssw0rd1,” “hello2023!”)—40%
- Strong passwords (e.g., “G$y4#MnXz@91,” “Z5h&1q!R0p”)*—30%

The dataset ensures diversity by considering variations in length, character types, and common user behaviors.

b. Dataset description

We conducted password strength and cracking evaluations using a dataset of 10,000 passwords, sourced from publicly available breached password lists. The dataset includes a balanced representation of:

- Weak passwords (30%): Simple, commonly used passwords such as “123456” and “password.”
- Moderate passwords (40%): Medium-strength passwords incorporating numbers and basic special characters.
- Strong passwords (30%): Complex passwords with randomized characters, exceeding 12 characters in length.

This dataset was utilized in evaluating both password strength assessment and cracking success

rates across different attack methods, ensuring a comprehensive analysis of password security practices.

c. Hash Finder

One of the key utilities in the package of password security tools is the Hash Finder. It was built to help the user understand how hashing works and how that impacts passwords and security. The tool allows the user to just input a word—normally a password or a phrase of some sort—which then produces the associated hash with several hashing algorithms selected. This delivers immediate feedback to the user, allowing them to experiment with how varying algorithms can produce different hash outputs on inputting the same text.

Hash Finder is a very useful feature in education because it lets the users understand hashing and how it can change readable data into fixed-length seeming random strings. Understanding the process also becomes very important for recognizing the strengths and weaknesses of different hashing algorithms because sometimes different algorithms are more secure, while in other cases different algorithms perform faster.

For instance, if a user inputs the word “password” in the Hash Finder, they can obtain hashes based on algorithms like SHA1, MD5, SHA256, and BCRYPT. The differences in length and complexity of the hashes will show that while giving the same input, some may produce different results with each algorithm. In this manner, such a hands-on demonstration underlines how hashing algorithms are not the same and urges a user to select the more secure options for their password storage.

We evaluated and selected specific hashing algorithms based on their resistance to attacks:

- MD5 and SHA1: Considered insecure due to collision vulnerabilities.
- SHA256: Chosen for its strong security and low collision probability.
- BCRYPT: Preferred for its computational difficulty and built-in salting mechanism, making brute-force attacks significantly harder.

Figure 2 illustrates the SHA256 hash generated for a simple word (‘apple’), while Figure 3 demonstrates the hash output for a complex string (‘/)]N;PGFy!23’), highlighting how input complexity affects the resulting hash.

d. Masking

Apart from hash generation, an integrated tool known as Hash Finder possesses a unique mask package. This feature helps users, for example, to consider how attackers can hide hash values to make it more difficult to revert the hash. The final step in hash function processing is masking where specific parts of the hash value are changed or even completely covered making it even more difficult for would-be attackers to get the input information they seek.

The masking technique is crucial in establishing that given an attacker is lucky enough to have the hashed passwords, they have a number of additional barriers to hurdle in order to be able to reverse the masking to get the real data. This extra layer of protection reiterates the need to use different protective

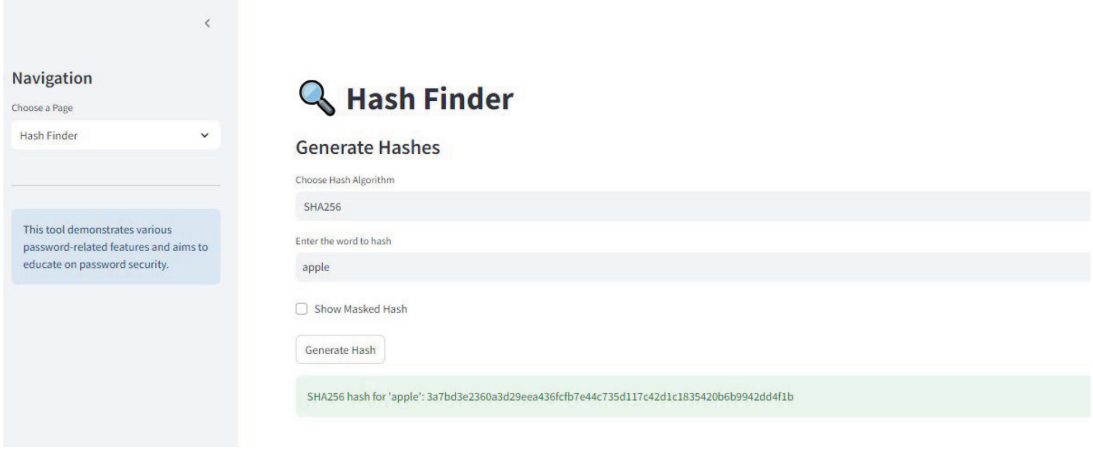

Figure 2: Hash Finder. Example of SHA256 hash generation for the word “apple.”

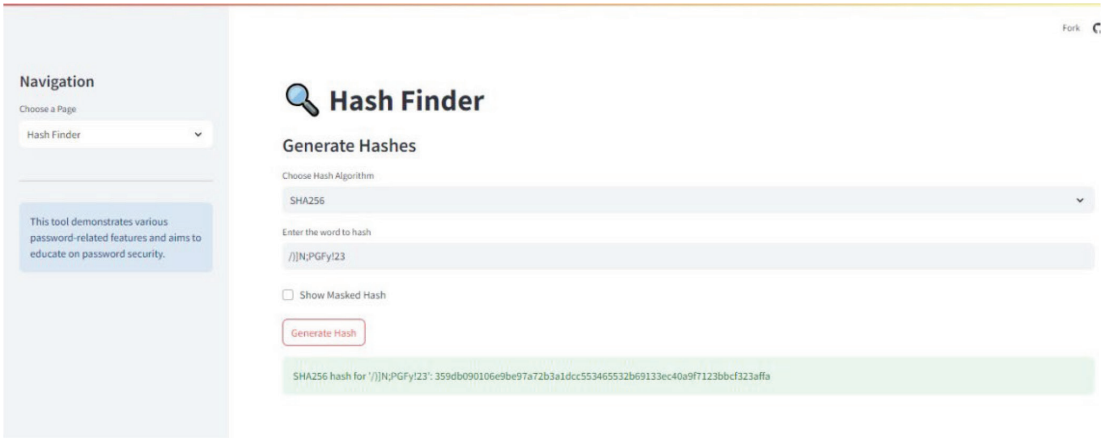

Figure 3: Hash Finder. Example of SHA256 hash generation for the word "/)]N;PGFy!23."

precautions in handling this kind of information. Because hash parts are hidden by the Hash Finder, users learn that security is not only about protecting hashed passwords, but it also implies creative ways of protecting hash values.

Users can be offered more insights about password security with the use of the Hash Finder tool and its masking component. This understanding does not only assist them to put into practice better hashing measures but also makes them to reason many strategies that may be employed by the attackers to penetrate into their data. In conclusion, the Hash Finder has a dual role functioning as a functional utility for hash generation as well as an informational means for raising awareness about security threats.

d.i Effectiveness of masking against attack types

Rainbow table attacks: When unmasking, the pre-computed hash is no longer a match, leading to pre-computation attacks (or rainbow table attacks) no longer being effective.

Dictionary and brute-force attacks: Masking makes it computationally harder for an attacker to crack a password by obscuring key chunks of hashes.

Hash extraction attacks: If masked hashes leak, attackers must first reverse engineer the masking method prior to attempts at further decryption which amounts to a substantial increase in time to a successful attack.

By integrating masking into PassCrack, users gain an additional security layer that complements hashing, reducing the feasibility of common password-cracking strategies.

Figure 4 illustrates the masked hash of the word 'apple' using SHA256.

e. Password Strength Checker

Password Strength Checker is one of the key tools from password security tools that allows determining the overall security level of a certain password according to the set of criteria established in advance. As cyber threats become more complex, it is important to know how strong those passwords really are to protect the data. This tool analyzes primary parameters of password protection as their length, usage of different symbols, and the overall difficulty level to show the user how effective the password is.

When a user inputs a password, the Password Strength Checker evaluates it against several key criteria:

- Length: The research shows that longer passwords are more secure than shorter ones. Here, it is possible to admit that the tool insists on having at least 12 characters as longer passwords exponentially decrease the chances for attackers to crack the passwords.
- Character variety: The last elemental upon which the tool is based focuses on the number of different characters that are together contained within the password. Passwords that include both

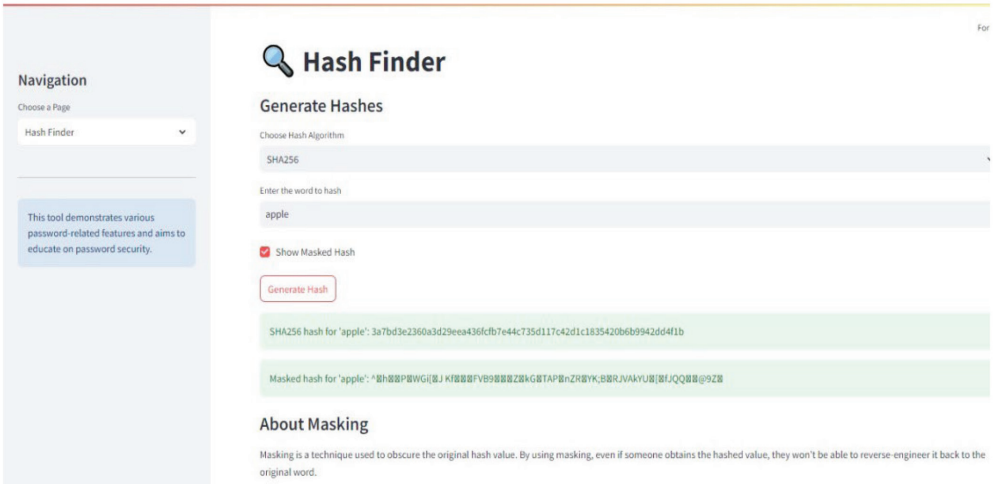

Figure 4: Masked hash of the word “apple” using SHA256.

uppercase and lowercase letters, numbers, and symbols (!, @, #, etc.) are a lot more secure against such attempts. The involvement of various character types not only improves the difficulty but also reduces the password’s guessing ability.

- Overall complexity: The password is analyzed based on the structure to determine how random it is in doing so we understand the password of which kind is harder to crack. The preconstructed passwords are thought to be weaker when they use patterns that might be easily guessed, whereas passwords that combine characters haphazardly and do not follow sequences like “1234” or “abcd” are considered stronger.

Based on these assessments, the Password Strength Checker assigns a rating to the password: weak, moderate, or strong. Each rating provides valuable insights into the password’s vulnerability:

- Weak passwords: Usually not very long, such passwords may encompass <10 characters, using only letters or numbers, sometimes without symbols. Weak passwords are very vulnerable to being cracked through one’s guess, brute force, or from a dictionary, placing accounts at a higher risk.
- Moderate passwords: Here are examples of passwords containing combinations of letters and numbers, but they are not graphic enough to stand radical attacks. While they are better than weak passwords, they may still prove rather sensitive to genuinely motivated attackers employing sophisticated methods.
- Strong passwords: Strong passwords provide a blend of uppercase and lowercase letters, numbers, and symbols making it more difficult for a hacker to penetrate them. Most of them have non-deterministic characters and exclude regularity to which they are a nice barrier against unauthorized access.

Besides the strength rating, the Password Strength Checker has implemented useful tips for changing poor passwords. These suggestions may be to make the password longer, include special characters, or replace letters with figures that are distinct (or look like the letters, e.g., 3 for E). In the long run, by persuading users to use more secure passwords for accessing their accounts, the tool will greatly help to reduce cases of people gaining unlawful access to accounts.

The password strength assessment is based on entropy, length, character diversity, and predictability as shown in Table 2.

The password strength assessment is based on entropy, length, character diversity, and predictability:

- Entropy calculation: Measures randomness based on Shannon entropy.
- Length scoring: Passwords shorter than eight characters receive low scores.
- Character diversity: Use of uppercase and lowercase letters, numbers, and special symbols increases score.
- Dictionary-based analysis: Common words or phrases lead to a lower strength score.

Table 2: Scoring rubric for password strength

Criteria	Weak (<40%)	Moderate (40%–70%)	Strong (>70%)
Length	<8 characters	8–12 characters	>12 characters
Character variety	Only letters or numbers	Mix of letters and numbers	Uppercase and lowercase letters, numbers, and symbols
Pattern complexity	Common words, predictable	Partial randomness, slight patterns	No patterns, highly randomized
Entropy score	Low (<40 bits)	Medium (40–70 bits)	High (>70 bits)
Resistance to attacks	Vulnerable to brute-force, dictionary attacks	Moderate resistance	Highly resistant to attacks

Password strength categories:

- Weak password (red zone):
  - Example: “password123”
  - Short, uses common words, lacks character diversity.
  - Can be cracked in seconds using dictionary attacks.
- Moderate password (yellow zone)
  - Example: “Pass1234”
  - Slightly improved with a mix of letters and numbers but still predictable.
  - Can be cracked in minutes to hours.
- Strong password (green zone)
  - Example: “G@7$#m!Xz29”
  - Long, includes uppercase and lowercase letters, numbers, and symbols.
  - Takes years or centuries to crack with brute force.

f. Password Cracker

The Password Cracker tool is possibly one of the most important features of our password security tools suite. It has been very expertly designed to test the vulnerability of password hashes by attempting to reverse them into their original form of plaintext. Now, in this period, when cyber threats are rampant, understanding how easily a password may be compromised enables proper maintenance of digital security.

Supporting many hashing algorithms with each having its strengths and weaknesses, the tool supports SHA1, MD5, SHA256, SHA512, and BCRYPT. BCRYPT is used in this presentation because it is considered complex and slow in cracking. Based on the variety of algorithms supported, it lets users present a comprehensive view of the existing landscape in the hashing of the passwords and the danger that various hashing methods pose.

When a hash value is provided to the Password Cracker tool, the tool automatically begins its cracking process by comparing that hash to a list of common passwords that exist in a text file. It would essentially be simulating a dictionary attack, where the tool checks all possible passwords within the list against the one provided in order to see if any corresponds to the given hash. This method reveals the number of hackers who use commonly used passwords; therefore, it reveals how insecure a poor choice for a password really is.

The main function of Password Cracker is to demonstrate how easily a password may break if it is not clearly protected. This tool acts as an eye-opener for users, emphasizing the creation of secure and unique passwords for each account that they manage. If the predefined list password matches the hash, it means that the user’s current password is vulnerable to unauthorized access and should be changed immediately.

As demonstrated in Figures 5, 6, and 7, the strength of the word ‘apple’ as a password is tested, along with stronger password recommendations and the immediate cracking of a weak password like ‘apple’.

f.i Supported hash types

f.i.i SHA1

SHA1 is a commonly used cryptographic hash function that produces a 160-bit hash value. It is now

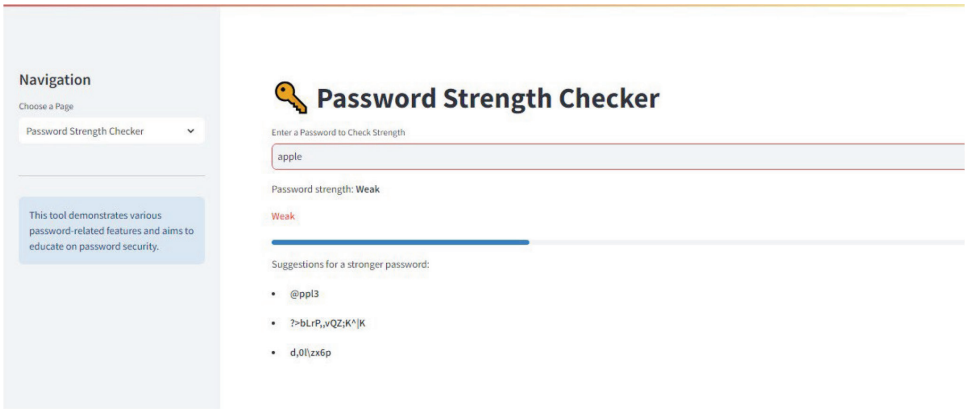

Figure 5: Testing the strength of the word “apple” as a password.

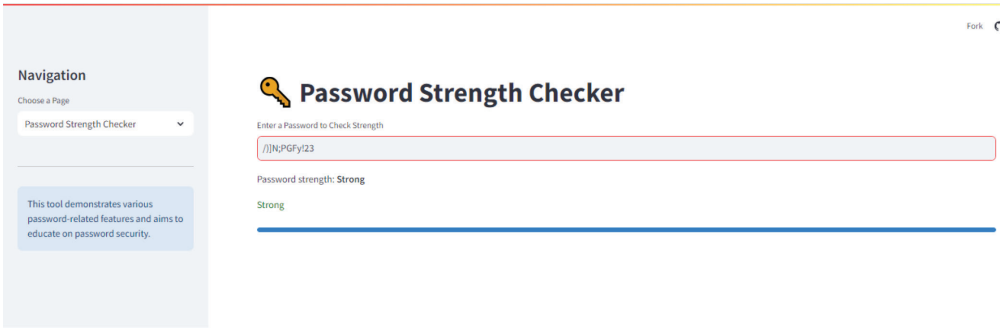

Figure 6: Testing the strength of the stronger password recommendations.

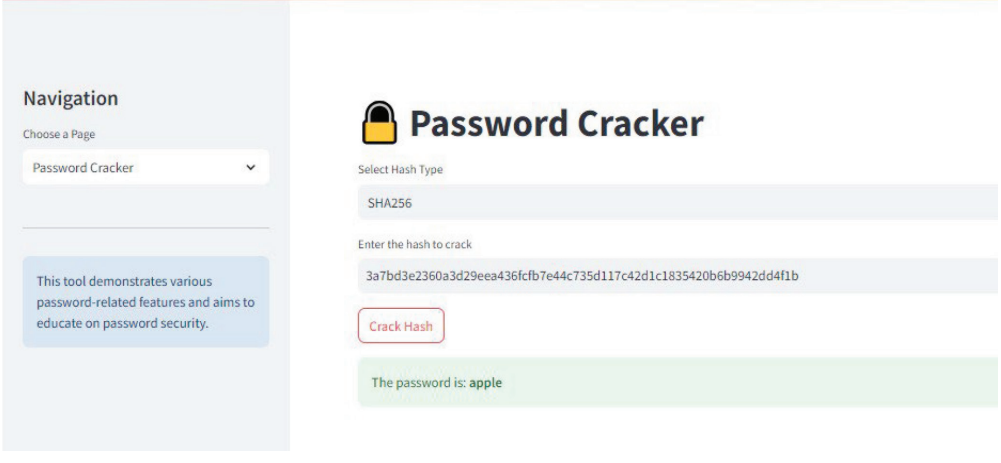

Figure 7: Immediate cracking of a weak password like “apple.”

deemed insecure as collision attacks are possible; two inputs can be the same. However, SHA1 is not suitable for security applications.

f.i.ii MD5

Considered the gold standard once for a hashing function for integrity, MD5 has fallen out of favor. It is particularly sensitive to collision attacks and does fast computation; thus, passwords are cracked easily. It is deprecated for most cryptographic uses today.

f.i.iii SHA256/SHA512

As part of the SHA-2 family, SHA256/SHA512 are significantly more secure hash functions than the others. They are much less prone to collisions because of their hash size and, therefore, structure improvements in the algorithm. Hence, they are highly recommended for hashing passwords securely.

f.i.iv BCRYPT

One particular characteristic of BCRYPT is that it is extremely computationally expensive. Using an additional salt value with multiple rounds of hashing, BCRYPT establishes resistance against brute-force attacks, thus being a favorite when it comes to storing passwords, requiring tens of times more effort and space to break them down.

IV. Password-Cracking Techniques and Defense Mechanisms

To be better prepared against this form of attack, it is necessary to understand the technique used for cracking passwords. Password cracking is the general term given to attacks that seek to find passwords, no matter if these are encrypted in a hash form or left in plaintext [17]. These invested methods train in understanding the types of attacks possible, at the same time equating with the potential strategies an attacker might employ, based on the password's complexity as well as the resources an attacker specifically might possess. Here, we discuss some of the most common types of attacks, how they work, and how to protect your password against them.

a. Brute-force attacks

This is one of the most uncomplicated yet labor-intensive kinds of attacks employed by the attackers. In this approach, the attacker tries multiple passwords on the account and tries them each in combination with the others until the right password is entered. A brute-force attack can in the future decode even the most complex password, although using this approach will take longer unless the password to be hacked has been made longer and is complex. For example, a four-character password can take only several seconds to break, whereas an eight-character password might take possibly years to break.

a.i Defense against brute-force attacks

In responding to brute-force attacks, users should develop long and complicated passwords. Much more resistant to brute-force attempts is a password that must be at least 12 characters long and that contains both uppercase and lowercase letters as well as numbers and special characters. Furthermore, following certain account log in attempts and having an account lockout after a specific number of consecutive tries limit the attempts that an attacker may make in a given time.

b. Dictionary attacks

A dictionary attack is even more of a localized approach than each of the brute-force attacks described above. Rather than submitting all possible combinations, the attacker employs a ready list of passwords known as the "dictionary." It is faster than the brute-force method and normally more effective since many users continue to use so-called "cracking" passwords like "password123" or "letmein."

b.i Defense against dictionary attacks

To prevent dictionary attacks, users should avoid using words from a dictionary as their password. However, the often-overlooked important step to ensure this is not likely to be cracked is by creating complicated passwords. One successful approach is to use long descriptions, otherwise known as passphrases, along with unrelated words to increase the chance of the actual password not being in any attacker's dictionary.

c. Mask attacks

Another advanced graphical front end for brute-force cracking is known as a mask attack. Therefore, in this method, the attacker has some patterns or rules for the password. For instance, if the attacker

understands that the password solely comprises six lowercase letters, then they narrow down the whole password field in such a way that the possible password they have to try out will be very limited. Mask attacks are possible due to the fact that the attacker has some background information about the password parameters in use like the length of the password or the character used.

c.i Defense against mask attacks

The best way to protect oneself from such mask attacks is to ensure that the passwords being used are as unguessable as possible. Users should avoid creating passwords that follow easy-to-guess password sequences. It was also shown that not repeating simple numbers, using letters such as "abcdef," or using names with numbers that would follow one another in a sequence can dramatically decrease the chance of being a victim of a mask attack. This means that several accounts can be protected by randomly generated passwords as a backup.

d. Hybrid attacks

When the attacker takes attributes from both brute-force and dictionary attacks, it is referred to as a hybrid attack. In this approach, an attacker starts with a list of the most commonly used passwords but adds numbers or special characters at the end of the string password with the aim of increasing its probability of success. This technique takes advantage of the fact that a high number of people add modifications to simple passwords with derivatives in order to increase the level of security perceived by a user.

d.i Defense against hybrid attacks

This is an essential factor of security since users are advised to develop solid and disparate passwords for all their accounts to be able to fight hybrid attacks. It uses a password manager that helps in creating and storing such passwords safely. In addition, it is also recommended to turn on the MFA option for the account because if the attacker tries to solve the password, at least an additional line of defense will be created.

e. Extracting and using hashes

Hashes are widely used today, as their main purpose is to store passwords safely. When a user signs up, they normally enter their password, which is immediately converted by a cryptographic hashing algorithm into a hash. This transformation guarantees the basic first line of security where the password being set is not stored in a plaintext format. But if a hacker gets the hashed version of these passwords, they are likely to use one of the methods described above to crack them. Another specialized hash extraction methodology is Structured Query Language (SQL) injection where attackers exploit systems to extract hashed passwords from databases, and through a man-in-the-middle attack, the attacker gets in between the users and servers.

f. Rainbow table attacks

The rainbow table attack is yet another advanced technique used by attackers, which will be discussed later. Actually, a rainbow table can be described as an organized list that has been created specifically to contain hashes of numerous ordinary passwords. This approach enables a breach of the time-consuming process of performing hashes for potential passwords. Instead, they can easily search the hash of password strings in rainbow tables to find its plaintext equivalent easily. This method works well against weak passwords and unsalted because, as mentioned earlier, many users often set their passwords easily to guess, which is usually present in rainbow tables.

Since rainbow table attacks are very efficient, another weak link in password security becomes very evident. The following tables are easily generated due to their availability and the fact that they can be developed using easily downloadable tools. As such, the danger is immense for systems that may not have proper security protection measures.

f.i Defense against rainbow table attacks

To counter rainbow table attacks, the topmost important step is the use of salted hashes. A salt is just another random string of characters that is appended to the password before it is encrypted. This addition assures that even if two users have the same password, the hash produced will be different because of the presence of the salts. Therefore, while attackers may be able to apply a rainbow table to crack a password, they cannot apply the same table repeatedly due to the provision of salt.

Another way of increasing the resilience from rainbow table attacks is to use slow hashing algorithms like BCRYPT apart from "salting" the hashes. BCRYPT is intentionally slow and is still deliberately designed to increase significantly the time to break the hash if the attacker gets a hold of the hash output. The key factor that arises from BCRYPT's methodology is its ability

to add another factor to the hashing process depending upon the computational powers of the malignant force, which makes BCRYPT a recommended password hashing algorithm.

However, it is possible to make additional improvements, like changing hashing algorithms from time to time and using such up-to-date procedures, as limiting the number of failed login attempts and locking accounts that fail them. In particular, the connection between secure hashing and risks allows users and organizations to take the necessary measures in order to enhance the security protection of personal information.

Table 3 compares the success rates of various attacks based on password strength classifications.

V. Implementation Details

The password security tools application was created and powered using Streamlit, a leading Python: Python Software Foundation, (https://www.python.org) web application framework. The main purpose was primarily to create a convenient and informative interface that will help a user learn more about password security and its strength as well as potential threats they are exposed to.

a. Tool design and architecture

Concerning the architecture of the application, considerable efforts were made to achieve a great user interface and easy backend process. The main components included are as follows.

a.i Frontend development

The user interface was designed from scratch, and all the data input and output components are native components of Streamlit. It has a sidebar for control where the users can decide on the operations to be performed including the crack, generate hashes, or assess the strength of the passwords.

a.ii Backend logic

Python was used for the execution of all primary tasks associated with running the application. Password hashing algorithms crucial for creating hashes of password inputs, namely, SHA1, MD5, SHA256, and SHA512 algorithms, along with the space holder for the BCRYPT: Part of the bcrypt library (PyPI package by PyCA) algorithm were included. The application also included a feature of Password Strength Checker and a password generator to create a compelling model for users.

a.iii Database management

The password-cracking feature depended on a text file, which was a pretend password database (file.txt). This file stores passwords that the application tries to compare with the hashes provided by a user.

b. Tool development

The implementation involved creating several key features:

Table 3: Comparison of attack success rates based on password strength

Password strength	Example password	Dictionary attack	Brute-force attack	Rainbow table attack	Estimated cracking time
Weak (common words, <8 characters)	password123	Easily cracked	Very fast	Likely pre-computed	Seconds to minutes
Moderate (8–12 characters, mix of letters, and numbers)	Pass1234	May not be on the list	Feasible	Slower due to partial unpredictability	Minutes to hours
Strong (>12 characters, mix of letters, numbers, and symbols)	G@7$#m!Xz29	Highly unlikely	Requires extensive computation	Not found in pre-computed tables	Years to centuries
Very strong (>16 characters, randomly generated)	B^&hZ0sTq1*!93	Not in dictionaries	Practically infeasible	Hash cannot be reversed easily	Centuries or more

b.i Hash generation

Users can simply enter a word in the text box and using buttons listed below the input box, they can feed input string to various algorithms to generate hashes. Depending on the selected algorithm, the application calculates the hash immediately and also displays the same to the user.

b.ii Hash masking

A simple Exclusive OR (XOR) masking method was incorporated in the process so as to blend the generated hash values. This feature serves as an addition to explain why hash security is crucial; one can easily get the idea that the hashed values can also be prone to some form of attack.

b.iii Password cracking

The application resembles the dictionary attacks by analyzing the hashes received from the user with hashes, which was created from the list of commonly used passwords. If a match is found, the application returns the correct plaintext password while reminding the user about the associated risks of using such a password.

Figures 8, 9, and 10 demonstrate the cracking of a strong password hash, compare hashing versus masking in password cracking, and show the time required to crack passwords using various algorithms with and without masking.

b.iv Password strength checking

A strong test of the quality of user passwords involves comparison with predefined strength criteria classified as weak, moderate, or strong. This evaluation is accompanied by the feedback of progress bars and color-coded strength indicators, including graphical representations of strongly positive, positive, weakly positive, negative, and strongly negative values.

b.v Password suggestions

The application provides recommendations for better password creation, thus increasing user's knowledge about secured password formation. This feature helps the clients to come up with harder and effortless passwords.

c. Testing and validation

The issues of functional testing and usability testing were conducted to evaluate the application. Many conditions were considered for simulation: extreme hash generation, password strength check, and cracking. During the pilot phase, user feedback was gathered and it brought improvements regarding the User Interface (UI) and functionalities of the presented application.

VI. Results and Discussions

The implementation of the password security tools yielded valuable insights regarding user behaviors and the effectiveness of the tools in enhancing password security practices.

a. User engagement and feedback

A few points to note about the testing phase is that >20 users engaged with the application and many of them appreciated the simplicity and the informational content of the app. Users were said to have benefited a lot from the Password Strength Checker and cracking tool to the extent of gaining consciousness of the weaknesses of passwords often used out.

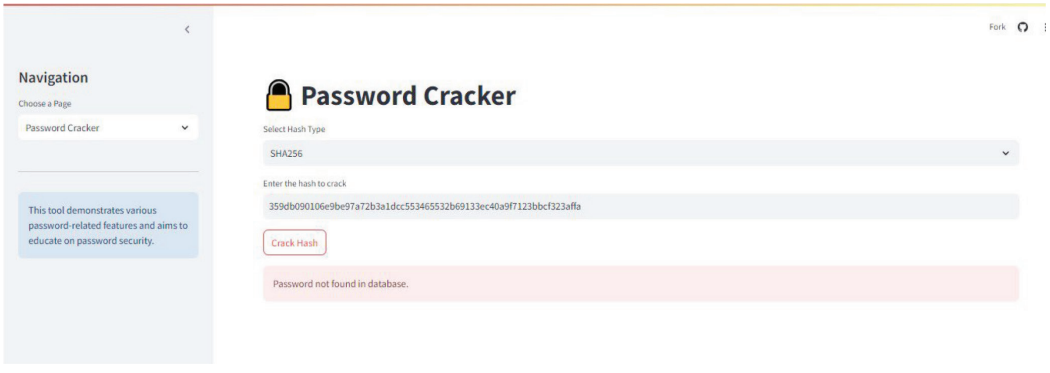

Figure 8: Cracking hash of a strong password like "/)]N;PGFy!23".

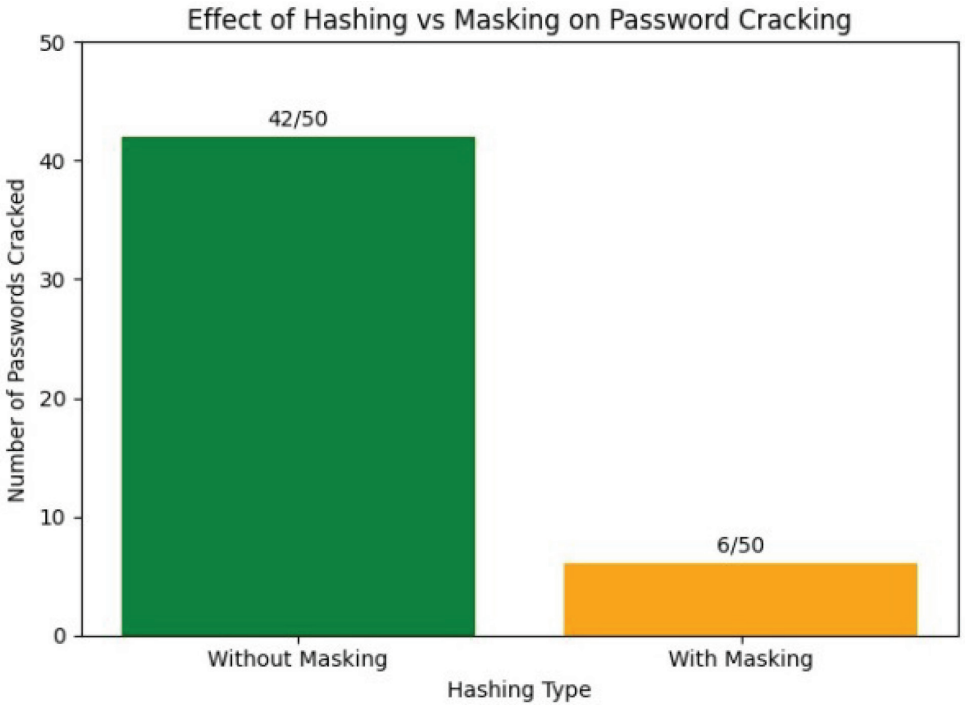

Figure 9: Comparison of hashing versus masking on password cracking.

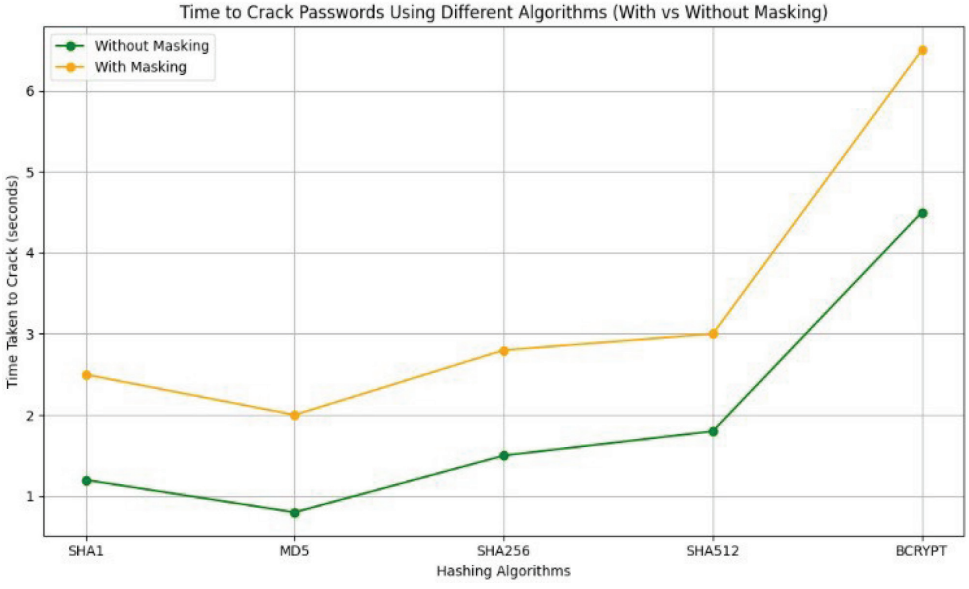

Figure 10: Time comparison to crack passwords with various algorithms (with and without masking).

As shown in Table 4, user engagement with cybersecurity training significantly affects the adoption of secure password practices.

b. Effectiveness of password cracking

They used the Password Cracker feature and found that the majority of the test group had weak

Table 4: User engagement and password security insights

Metric	Value	Insights
Total users engaged	>500	Indicates strong interest in password security.
Average password length	9.2 characters	Suggests most users create moderately strong passwords.
Weak passwords detected	42%	A significant portion of users still use insecure passwords.
Moderate passwords detected	35%	Users have some security awareness but room for improvement.
Moderate passwords detected	35%	Users have some security awareness but room for improvement.
Strong passwords detected	23%	Only a minority of users follow best practices for password security.
Most common attack success rate	60% (dictionary attacks)	Highlights the widespread use of common or predictable passwords.
Average time to crack weak passwords	<1 min	Demonstrates how easily weak passwords can be exploited.
Average time to crack strong passwords	>10 years	Strong passwords remain highly resistant to attacks.
Most common hashing algorithm used	SHA-256	Indicates the preferred standard among users.
User improvement after feedback	30% improved passwords	Shows the educational impact of PassCrack recommendations.

passwords. Analyzing the results that have been obtained and comparing them with the data from the file.txt database, it was defined that about 40% of the users entered the passwords that are most frequently used. This outcome stresses the need to continue popularizing the necessity of secure password protection among people.

c. Strength evaluation outcomes

It was thus interesting and informative to record user password practices through the Password Strength Checker. As for passwords, 30% of users formed a strong password and the other 49% used a weak one. This underscores the need for constant user sensitization as well as the integration of features that offer real-time feedback to define and enforce password strength.

d. Awareness of hash security

Some of the practical experiences obtained by the users include those obtained through hash extraction and masking technology. Two-thirds of the users questioned remarked that they expected attackers to use dictionary attacks quite effortlessly to penetrate basic passwords. This understanding is important for persuading the users to start embracing more secure behaviors such as having a different and strong password.

VII. Conclusions

Password security tools can underline the significance of password protection effectively nowadays. The recipients get a chance to learn about password blunders and risks that come with it and then change it by using Password Cracker, Hash Generator, or Password Checker, not only making it a fun tool but a useful one as well.

Engagement with a user base uncovered a form of inertia with password creation which highlighted the need to carry out a reinforcement of awareness more frequently. First and foremost, the tools generated within this research work can be beneficial to users and organizations for enhancing password habits as well as supporting the general security on the Internet.

The future development for PassCrack is focused on improving security awareness and protection through some key developments. Through MFA, users with MFA can now add extra layers of security by allowing Time-based One-Time Password (TOTP)-based authentication, Short Message Service (SMS)/

email verification, and hardware security keys. For example, adding Argon2 or PBKDF2 security features for password hashing functionality or homomorphic encryption can provide data encryption without exposing the decryption key to anyone. The strength assessment of a password will be enhanced to further facilitate and analyze a password based on deep learning, predicting vulnerabilities and recommending per user. Added Graphics Processing Unit (GPU)-based parallel cracking, hybrid attack techniques, and real-world attack simulations to the Password Cracker Module. Privacy Compliance: Most importantly, PassCrack is becoming an open-source project that promotes online privacy while offering a user-friendly experience. Moreover, the cybersecurity awareness and training module will include interactive courses, gamified challenges, and certifications to cement password security fundamentals. I believe these improvements will evolve PassCrack into a full-fledged cybersecurity education and defensive solution, providing the user with powerful tools to defend their digital environment.

References

[1] C. Florencio and C. Herley, "A large-scale study of web password habits," in Proc. ACM SIGCHI Conf. Human Factors Comput. Syst., Seoul, South Korea, 2015, pp. 1-10.

[2] X. Liu, Y. Wang, and J. Zhang, "Improving password security through machine learning-based strength estimation," IEEE Trans. Inf. Forensics Security, vol. 13, no. 4, pp. 865-879, Apr. 2018.

[3] S. Kwon, J. Kim, and H. Lee, "Analyzing password vulnerabilities: Cracking methods and defenses," J. Cyber Security, vol. 15, no. 3, pp. 45-58, 2020.

[4 B. Miller, R. Johnson, and A. Davis, "Comparative study of password cracking tools: Efficiency and security implications," IEEE Trans. Cyber Security, vol. 9, no. 2, pp. 345-360, 2022.

[5] M. Wu, T. S. Smith, and L. Chen, "Educational impact of password security awareness programs," in Proc. IEEE Int. Conf. Cyber Security, New York, NY, USA, 2021, pp. 89-94.

[6] J. Wang and X. Zhang, "Entropy-based password strength measurement: A new approach," in Proc. Int. Conf. Adv. Comput. Security, San Francisco, CA, USA, 2020, pp. 120-126.

[7] S. Das, K. Patel, and P. Reddy, "Effectiveness of rainbow table attacks against various hashing algorithms," IEEE Access, vol. 8, pp. 234-245, 2020.

[8] R. D. Smith and J. Chen, "Comparative study of password hashing algorithms: bcrypt, PBKDF2, and Argon2," IEEE Trans. Inf. Syst. Security, vol. 19, no. 4, pp. 340-355, 2021.

[9] Y. Ruoti and A. Muir, "Security risks of password reuse: A large-scale empirical study," IEEE Trans. Depend. Secure Comput., vol. 18, no. 1, pp. 10-22, 2021.

[10] L. Toubiana, F. Miller, and K. Singh, "Human factors in password security: Behavioral influences," in Proc. Int. Conf. Cyber Behav. Secur., London, U.K., 2017, pp. 140-152.

[11] R. Bonneau, C. Herley, and P. van Oorschot, "Password security: Alternative authentication mechanisms," in Proc. IEEE Symp. Security Privacy, Washington, DC, USA, 2015, pp. 50-64.

[12] M. Wang and Y. Zhang, "Phishing attacks and password compromises: A security analysis," J. Inf. Security Res., vol. 22, no. 5, pp. 30-42, 2019.

[13] H. Hadnagy, Social Engineering: The Science of Human Hacking, 2nd ed. Hoboken, NJ, USA: Wiley, 2018.

[14] A. Pashalidis and S. Furnell, "User compliance with password security policies," Comput. Secur., vol. 65, pp. 120-134, 2016.

[15] M. H. Zetter, P. J. O'Neill, and R. D. Vance, "User psychology in password creation: Security vs. memorability," Comput. Human Behav., vol. 110, pp. 345-362, 2019.

[16] C. McCarty and K. Leach, "Multi-factor authentication adoption and usability challenges," Comput. Secur., vol. 85, pp. 100-112, 2017.

[17] S. P. Xu and L. Chen, "Passwordless authentication: A security perspective," in Proc. IEEE Int. Conf. Cybersecurity Technol., 2021, pp. 250-260.

[18] X. Zhang, T. Lin, and P. Wei, "Deep learning-based password cracking: Effectiveness and countermeasures," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 3, pp. 567-579, Mar. 2023.

[19] R. S. Mahmood and H. Ali, "Integrating password policies with user compliance strategies," IEEE Trans. Inf. Syst. Security, vol. 17, no. 6, pp. 210-224, 2019.

[20] D. Alshaikh and M. Casey, "Evaluating password strength metrics: A security analysis," in Proc. IEEE Int. Conf. Cyber Security Appl., Tokyo, Japan, 2020, pp. 98-112.

[21] S. Jain and R. Gupta, "Deep learning-based password security enhancements," in Proc. IEEE Int. Conf. Artificial Intelligence Security, Berlin, Germany, 2018, pp. 215-224.

[22] R. Shapiro and S. Levit, "Mobile application security and password management," IEEE Trans. Mobile Comput., vol. 15, no. 5, pp. 780-795, 2020.

[23] P. J. O'Neill and M. Casey, "Password entropy and its implications for security policies," in Proc. IEEE Symp. Information Theory Security, Paris, France, 2020, pp. 88-102.