Dynamic Password Assessment and Protection Based on Attacker Strategies *

Zheng Changqian†
Beijing Institute of Technology
changqian.zheng@gmail.com

Zhang Shuwei‡
Southern University of Science and Technology
12212912@mail.susetch.edu.cn

Hu Yijing§
University of Nottingham, Ningbo China
yijinghu975@gmail.com

Zhao Zewen
Beijing University of Posts and Telecommunications
zhao16682299199@163.com

Zhang Yijun
South China University of Technology
zhangyijun1913@outlook.com

ABSTRACT

This study pioneers an innovative approach to password security by examining attacker-generated passwords, an area often overlooked in current research. We introduce a system for extracting and analyzing malicious passwords to bolster defenses against online brute-force attacks, diverging from conventional methods focused on user password habits. Furthermore, our research emphasizes the importance of understanding attackers password construction strategies to tailor defenses effectively. By analyzing patterns and semantic components of passwords used by attackers, we enhance the robustness of password security measures.

Additionally, our approach includes a dual methodology for capturing and storing password characteristics: firstly, storing user-specific patterns during registration or modification, which are independent of attackers (e.g., password patterns); and secondly, dynamically updating and analyzing the current attacker database during each login, focusing on attacker-related characteristics such as frequency of use of each word in attacker password attempts. This enables proactive user warnings based on current attacker trends.

This comprehensive approach not only addresses evolving cyber threats but also aligns defenses with dynamic cybersecurity landscapes. Moving forward, our focus will be on refining theoretical frameworks for password strength evaluation and exploring advanced pattern and word extraction techniques. The scarcity of research on attacker password patterns underscores the need for further exploration in this critical cybersecurity domain.

In conclusion, this research contributes novel insights into password security by shifting focus to attacker perspectives, paving the way for more effective defense strategies against malicious activities targeting password systems.

1. INTRODUCTION

In the realm of password security, a significant gap exists between the ideal standards for robust passwords and the practical preferences of users, who often prioritize ease of memorization over security strength. Current guidelines advocate for passwords of at least 11 characters in length, comprising a mix of uppercase letters, lowercase letters, numbers, and special characters, recognized for their resilience against attacks[7].

However, due to the challenges users face in remembering such complex passwords [7], they frequently opt for word-based passwords, i.e. passwords constructed from a list of words or meaningful strings, which, despite being easier to recall, are more vulnerable to exploitation by attackers [7]. These passwords which are typically derived from common words or phrases, lack the security strength afforded by more complex and randomly generated combinations.

This study focuses on the development of a password strength assessment model tailored specifically for word-based passwords, leveraging dynamically collected malicious password data. In addition to storing user-specific patterns during registration or modification, which are independent of attackers (e.g., password patterns), we also dynamically update and analyze the current attacker database during each login. This dual methodology aims to provide proactive user warnings based on current attacker trends.

*The demo of this paper about malicious code processing parts is in `https://github.com/ckt1i/grassboat`

†The proposer of the idea and designed the probability calculations in the paper

‡Responsible for the system design and word extraction in the paper

§Responsible for polishing the paper and literature search..

By establishing a phishing site designed to capture passwords attempted by malicious actors, our research aims to enhance understanding of password vulnerabilities and improve overall password security measures.

This paper addresses the challenge of evaluating the security of word-based passwords, despite their prevalent usage and susceptibility to attacks. Users often favor word-based passwords due to their ease of recall, yet these passwords lack the robustness provided by more complex combinations. The primary focus lies in developing effective methods to assess and improve the security of such passwords amidst evolving cyber threats.

Our study makes several key contributions to the field of password security:

- Basic characteristics of the attackers' password: We identify and analyze fundamental characteristics and distribution patterns of attacker passwords by constructing a simulated server to collect malicious password data.
- Words extraction and Patterns: We propose mechanisms for extracting words from passwords and analyzing patterns, which enhance the detection of weak password structures.
- Password strength evaluation: We introduce a novel framework for evaluating password security based on empirical analysis of attacker methods, providing insights into the vulnerability of user passwords and guiding improvements in password security practices.
- Password management system design: We integrated theoretical research with practical application and designed a system that enables dynamic evaluation of user password security and defense against online password attacks

This sets the stage for understanding the critical need to address weaknesses in word-based passwords and outlines our approach to enhancing password security through analysis and proactive defense strategies.

2. RELATED WORK

2.1 Literature review

Current strategies aimed at safeguarding user passwords primarily employ a user-centric approach. This approach involves analyzing password compositions, identifying frequently used words or phrases, and implementing targeted protections based on these findings [7]. While effective to a degree, this strategy requires ongoing adaptation to counter evolving attack methodologies and user behaviors.

From an offensive standpoint, comprehensive statistical analyses using tools like CAUDIT shed light on significant insights into password security. CAUDIT is an instance of meticulously logged 11 billion SSH brute-force attacks directed at NCSA's systems from February 2017 to November 2019, revealing enduring and methodical efforts by malicious actors [9]. Attackers employed sophisticated strategies, including the use of stolen SSH keys and evasion techniques like randomized client versions, underscoring their adaptability and resourcefulness.These findings underscore the inherent vulnerabilities of multi-user systems, where even a single weak password can compromise entire infrastructures. Attackers often exploit generic password habits derived from publicly accessible datasets, perpetuating risks despite advancements in computational capabilities and password policies [9]. Even though Owen et al have done some research [6] on brute force passwords based on fake ssh collection, however, the research protection based on attacker's passwords strategies is still nearly in a blank.

These findings underscore the inherent vulnerabilities of multi-user systems, where even a single weak password can compromise entire infrastructures. Attackers often exploit generic password habits derived from publicly accessible datasets, perpetuating risks despite advancements in computational capabilities and password policies.

To mitigate these threats effectively, proactive security measures tailored to address evolving attack vectors are essential. Furthermore, despite advancements in hacking techniques, dictionary-based attacks remain prevalent. These attacks are often enhanced with password mangling rules that mimic common user behaviors, such as character substitutions and alterations in capitalization and numerical placement within passwords [1]. Consequently, reinforcing strong password policies and educating users on secure password practices are crucial in mitigating the effectiveness of such attacks.Although strong, complex passwords provide substantial security benefits, their widespread adoption is impeded by usability concerns. Therefore, it is imperative and pragmatic for us to examine password security from the perspective of attackers and their methods.

2.2 Comparison with Existing Methods

Analyzing password security from the attacker's perspective offers several advantages compared to traditional user-centric assessments:

- By focusing on how attackers construct passwords, defenses can be tailored specifically to thwart known attack methods. This approach directly addresses the techniques and tools malicious actors use, enhancing the likelihood of successful mitigation.
- Analyzing the user's password composition for protection usually involves privacy issues, and obtaining the data set requires certain permissions[2]. The existing research is basically based on the accidental leaked password library for analysis, and there is a problem that the data may be outdated. Attackers' techniques evolve rapidly, necessitating up-to-date data and analysis methods to effectively counter emerging threats.
- By adopting this perspective, security professionals can implement proactive measures that anticipate and mitigate potential vulnerabilities before they are exploited. This approach not only enhances the robustness of password security but also aligns defenses with the dynamic landscape of cyber security threats.

3. HYPOTHESIS

Our research proposes several hypotheses for further exploration and optimization:

- Exclusion of Weak Passwords: Weak passwords are systematically excluded from secure systems due to their vulnerability to attacks. This hypothesis underscores the importance of stringent password policies in mitigating security risks.
- Challenges with Strong Password Adoption: Despite their enhanced security, strong passwords face adoption challenges due to user memorization difficulties. Consequently, users often resort to word-based passwords, which are easier to remember but more vulnerable to attacks.
- Exploitation of Word-based Password Vulnerabilities: Attackers leverage knowledge of user habits to exploit vulnerabilities in word-based passwords. They employ sophisticated brute-force techniques tailored to mimic common user behaviors, thereby compromising security.
- Scale and Persistence of Attackers: The volume and persistence of password attempts by attackers exceed those typical of legitimate user behavior. This observation highlights the scale and intensity of malicious activities targeting password systems. This can be proved by our statistic based on collected malicious passwords.

Based on these assumptions, our research endeavors to explore and optimize password security strategies, aiming to enhance protection against evolving cyber threats from the perspective of investigating attackers.

4. METHODOLOGY

```
function MAX-K-SEGMENTATION(password, k)
    L ← len(password)
    dp[0].push(0)        ▷ dp[u] is a min-heap for 0 ≤ u ≤ L
    for i ∈ 1...L do
        for j ∈ 0...i − 1 do
            value ← getValue(password[j + 1 : i])
            for sum ∈ dp[j] do
                dp[i].push(sum + value)
            end for
        end for
        while dp[i].size() > k do
            dp[i].pop()
        end while
    end for
end function
```

designed a dynamically password judging mechanism by designing a dynamic system for maintaining and analyzing password security by focusing on two key aspects: identifying and storing potentially malicious passwords, and evaluating the security of user-submitted passwords through pattern and word analysis.

4.1 The basic research for the malicious passwords

Because the lack of the search for analyzing the attackers password patterns. To proof our assumption, we did a basic research for malicious passwords.

we set a open-sourced fake ssh project [3] to collected malicious passwords and use some naive methods to sort the passwords into three parts: Weak passwords, word-based passwords and the strong passwords. Here, we define these three types of passwords as follows:

- Weak passwords: The password that is too simple that is too easy to find, which contains some week acknowledged passwords like "123456" or some information about the information of website, server and users like "root" , "linux"
- Word-based passwords: The password here contains the words that is used in passwords, include the language words and the other strings that frequently used in passwords. The words can be extracted and recorded its difference by these type of password in limited time for calculation like "P@ssword" , "root123"
- strong passwords : The password that is with high randomness of characters or can not extract the words in a limited time.

Based on our classify of the passwords, we acccordingly analysis the passwords collected. From 109,904 instances, weak passwords constitute approximately 60% of the dataset, while word-based passwords account for 39%, and strong passwords represent about 1%.

4.2 Extract the pattern of the password

Recently there have been a few researches on the structure and semantic analysis of password [10] [4] [8]. However, these methods still have some limitations such as Lack of consideration for common special characters in passwords or insufficient segmentation algorithm.

In our research, we apply dynamic programming method and NLP techniques to extract semantic pattern of common passwords. In the following part, we will discuss the three critical parts of our approach: pattern definition, corpus generation and algorithm workflow.

4.2.1 Pattern definition

To record and analyze the structure of a password, we introduce the concept of pattern: "Pattern" is a succinct symbol sequence for expressing the password structure. In the following part of our paper, we use these characters to record the passwords structure:

- **W** : Represents the part consists of the words or the other word like phrases that frequently found in passwords , like "password" , "linux" , "admin" , "123456"
- **A** : Represents the part consists of the characters in alphabet , like "a" , "Z" and so on.

- **N** : Represents the part consists of the characters in numbers like "1" , "9" and so on.
- **S**: Represents the part consists of the characters in special words like "@" , "!" and so on.

Example 4.1.

- ***admin@123456xyz*** $\rightarrow W_5S_1N_6A_3$

In our definition, all meaningful strings are classified as **W**, and the meaningless characters are categorized in alphabet, numbers, special words. For security consideration, we have not classified word any further, for example POS. More details will be discussed in 5.1.

4.2.2 Corpus generation

Our algorithm relies on two main corpus sources: existing English corpora and special strings derived from public password datasets. For English corpora, we utilize data from the Google Ngram project, obtained from [4], which includes frequency information for the most common 10,000 1-grams, 5,000 2-grams, 3,000 3-grams, 1,000 4-grams, and 1,000 5-grams. Additionally, we augment this with lists of names, surnames, countries, and cities to encompass potential password constructs.

Regarding special strings found in passwords, a relevant area of research involves "word detection" or "word discovery," extensively explored in Chinese text processing. Despite similarities in the absence of word spaces in both passwords and Chinese text, existing algorithms for word detection in Chinese texts often consider grammatical characteristics, which may not directly translate to password generation rules. Due to these reasons, we employ a straightforward algorithm that counts substring occurrences across all passwords, with subsequent noise reduction.

These corpora serve as references for exact or approximate matches and aid in probability calculations for different segmentations. Given the inherent uncertainties in our string detection algorithm, priority is given to existing corpora, with the generated corpus utilized when no direct matches are found.

4.2.3 Segmentation algorithm

```
function MAX-K-SEGMENTATION(password, k)
    L ← len(password)
    dp[0].push(0)        ▷ dp[u] is a min-heap for 0 ≤ u ≤ L
    for i ∈ 1...L do
        for j ∈ 0...i − 1 do
            value ← getValue(password[j + 1 : i])
            for sum ∈ dp[j] do
                dp[i].push(sum + value)
            end for
        end for
        while dp[i].size() > k do
            dp[i].pop()
        end while
    end for
end function
```

The core process of our pattern extraction involves segmentation, where the goal is to identify the optimal segmentation that accurately reflects the password's structure. Initially, common strategies treated English letters and other characters separately, viewing numbers and special symbols as potential "gaps" [6]. However, this approach may fall short due to users' tendencies to manipulate passwords through character insertion, deletion, and substitution, often including special characters within words.

To address this issue, our algorithm explores all potential segments and employs a heuristic function to assess how closely a substring resembles a word. Prior approaches such as [3][6] utilized a heuristic function that assigns a length value if a segment matches any word in the corpus; alternatively, our algorithm employs a quadratic function for heuristic calculations. This approach transforms the problem into finding the segmentation with the highest score, where the score represents the sum of heuristic values for all sub-segments, calculated using dynamic programming techniques.

Given that multiple segmentations may achieve the maximum score, and that the segmentation with the highest score may not always be the most suitable due to heuristic function imperfections, we maintain records of segmentations with the top k highest scores. Subsequently, we select the segmentation with the highest probability based on word frequency information within the corpus. In our experiments, we typically set $k = 10$.

To estimate the probability of a segmentation candidate, we utilize the frequency of both single words and n-grams (2-grams to 5-grams) in our calculations. Specifically, we estimate the probability using segments tagged as **W** and use this word list's probability to represent the segmentation's overall probability. For words without a direct match in the corpus (identified via fuzzy matching), we also consider the probability of edits applied to them.

Ultimately, the segmentation with the highest score and probability is selected as the password's pattern.

4.3 Evaluating user's password security

Building upon the methodologies described earlier, we extract patterns and associated words from user passwords, enabling us to gauge the likelihood of attackers using these passwords based on pattern and word frequency analysis.

4.3.1 Calculate the pattern space

Here we define the addition space D, which means for the same pattern, we can make different passwords via inserting different characters into non-word parts. The number of different insertion of characters can be represented by D , which can be calculated by the number of each pattern's parts apart from the word part: alphabet insertion partn_{a_i}, number insertionn_{n_i} , special character insertion part n_{s_i} . Each digit, depending on the data type, will have a different number of insertion possibilities : k_{a_i} , k_{n_i} , k_{s_i} . Then we can calculate the pattern space:

$$S = \prod_{a_i} n_{a_i}^{k_{a_i}} \prod_{n_i} n_{n_i}^{k_{n_i}} \prod_{s_i} n_{s_i}^{k_{s_i}} \tag{1}$$

Then the probability of inserting the correspond characters is:

$$p_1 = \frac{1}{S} = \frac{1}{\prod_{a_i} n_{a_i}^{k_{a_i}} \prod_{n_i} n_{n_i}^{k_{n_i}} \prod_{s_i} n_{s_i}^{k_{s_i}}} \tag{2}$$

4.3.2

sectionCalculate the probability of insert corresponding words For the password, apart from the word insertion, we need to evaluate the probability of using correspond word's. This is evaluated by the corresponding elements n_{w_i} occurrence in words with same length n_{l_i} , then the words probability can be calculate as:

$$p_2 = \prod_{w_i} p_{w_i} = \prod_{w_i} \frac{n_{w_i}}{n_{l_i}} \tag{3}$$

4.3.3 *Calculate the probability of this password appearing within the expected time frame.*

After evaluating the password's editing space and the word's probability , we can evaluate the probability of a user's password used by attackers:

$$p = p_1 p_2 = \frac{\prod_{w_i} \frac{n_{w_i}}{n_{l_i}}}{\prod_{a_i} n_{a_i}^{k_{a_i}} \prod_{n_i} n_{n_i}^{k_{n_i}} \prod_{s_i} n_{s_i}^{k_{s_i}}} \tag{4}$$

Then to calculate probability of the passwords used maliciously within the expected time frame. We also need to calculate number of the malicious passwords sent to server N in expected time T , which can be predicted by the number of malicious passwords n_0 in collected time t_0 :

$$N = T \frac{n_0}{t_0} \tag{5}$$

Then the expected probability in time range t for a user's password used by attackers is:

$$P(t = T) = 1 - (1 - \frac{\prod_{w_i} \frac{n_{w_i}}{n_{l_i}}}{\prod_{a_i} n_{a_i}^{k_{a_i}} \prod_{n_i} n_{n_i}^{k_{n_i}} \prod_{s_i} n_{s_i}^{k_{s_i}}})^{T \frac{n_0}{t_0}} \tag{6}$$

By calculating the probability, we can make the estimation of the password strength : When the probability in a scale of time is bigger than the preset threshold value θ , then we can judge it that this password is not secure recently.

5. SYSTEM DESIGN

To bridge theoretical research with practical applications, we develop a basic framework for a dynamic password protection system.

5.1 Design principle

Traditional password management system displays demonstrate good resilience against offline password cracks, i.e. the resilience to password file leakage and corresponding offline hash cracking. However, such system have poor performance against online user-targeted attacks, for they cannot identify whether a user is under attack and how closely the password is to be cracked.

On the other hand, if we more or less record some characteristics of users' passwords, it is possible that we can detect such targeted attacks by conducting similar analysis on the attackers' passwords and performing characteristic comparisons and hence evaluate the strength of user password. However, while recording more information can make strength analysis and attack detection more accurate, attackers can more easily engage in offline cracking in the case of file leakage. Thus, there is a trade-off between defensive capabilities against online attacks and offline attacks.

5.2 Goal and framework

Our system is designed to defend against two types of attacks: indiscriminate weak password brute force attacks and targeted password brute force attacks.

Here are explanations for the two concepts:

- Indiscriminate attack: "Indiscriminate attack" refers to a method where Attackers systematically attempt to gain unauthorized access to multiple accounts or systems by trying common weak passwords. This method exploits the widespread use of easily guessable passwords, leveraging automated tools to maximize the chances of breaching a large number of accounts.
- Targeted password brute force attack: "Targeted attack" refers to a method where attackers gather specific information about users' passwords, such as patterns or other details, to generate a large number of password guesses. This approach focuses on exploiting known patterns and user-specific information to crack passwords.

The following diagram depicts the core framework of our system.

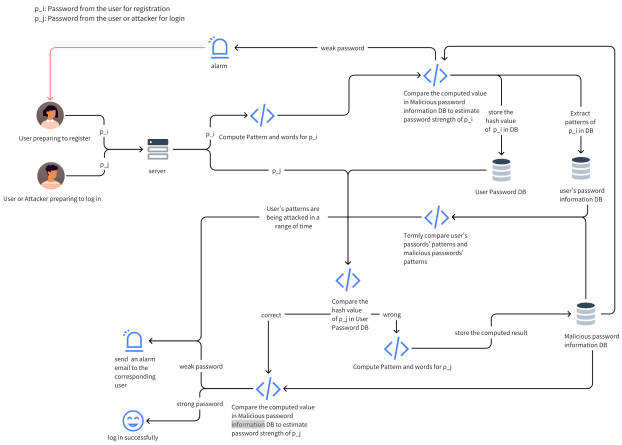

Figure 1: Overall framework

5.3 Main components

Our system consists of the following components:

- Server handling login requests
- Database of user hashed passwords
- Database of user password characteristics

- Database of attacker password characteristics
- Scripts for automated characteristic extraction and password strength analysis.
- Scripts for regular password strength checking and user alerting.

Here the we use "pattern" and relevant probability to represent password characteristic, which allows for dynamic password analysis without recording excessive information based on the principle mentioned above.

Since there might be demand for maintaining a dynamic weak password dictionary for common weak password detection, we may also need to set up a phishing server to collect weak password data.

5.4 Data storage and maintenance

For database of users' hashed password, it only stores userID and their corresponding hashed password values for the purpose of final correctness verification.

For database of users' password characteristic, it stores userID, password pattern, cracked probability estimation (as stated in 4.3). As shown in 4.3, the probability calculation involves pattern frequency and word frequencies, while the later is not store in the database for security consideration. To deal with the problem, we adopt a semi-dynamic method for probability calculation: each time the user register / login, we update the probability based on latest attacker statistics instead of explicitly store the speific words contained in user's password. Under the mechanism, users who log in more frequently can obtain a more accurate assessment of password strength.

For databases of attacker password characteristics, one stores attackers' passwords pattern and corresponding frequencies. Another stores specific words used in attackers' passwords and corresponding frequencies. Since we only consider the attacking trend over a recent period of time, for example, a month, we should also maintain a database/file to record each attack attempts in chronological order to remove statistics beyond the time range.

5.5 Defence mechanism

5.5.1 Defence of indiscriminate attack

To defend the indiscriminate attack , we design a dynamically password judging mechanism focusing on two key aspects: identifying and storing potentially malicious passwords, and evaluating the security of user-submitted passwords through pattern and word analysis.

The system dynamically tracks and records potentially malicious login attempts by analyzing failed passwords. These passwords are broken down into their constituent patterns and words, and the resulting data is stored in a malicious password information database.

To enhance password security on it, Our system employs the following evaluation and alert mechanisms during registration and post-login scenarios.

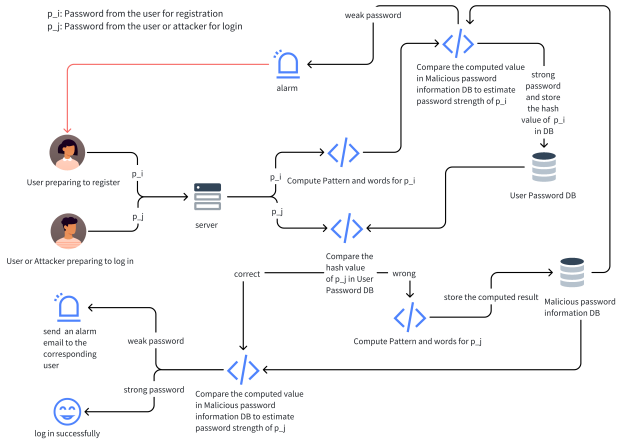

Figure 2: defence on registration and post-login scenarios

- Registration Password Evaluation: Upon user submission of a password during registration, the system analyzes its patterns and words. This analysis includes comparison against data stored in the malicious password information database. If the submitted password meets security criteria, it is accepted, and its hashed value is stored in the user password database.
- Post-Login Password Evaluation: For successfully logged-in users, the system continuously evaluates their passwords by comparing patterns and words against the malicious password information database. If a password is deemed insecure, the system alerts the user via email, recommending a password change to enhance security.

5.5.2 Defence of targeted attack based on patterns

To defend targeted attack of user's password, we designed a termly password intensity checking system based on the password intensity evaluating system above. We record user's passwords' patterns and the probability termly compare and modify the probability of user's passwords.

To enhance this type of attacks, Our system implements the following security evaluation and alert mechanisms throughout the entire lifecycle.

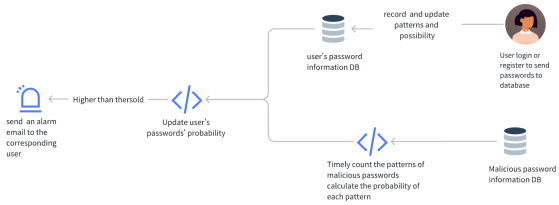

Figure 3: defence throughout the entire lifecycle

- Patterns' recording: When a password is recognized as malicious passwords. They are extracted into words

and patterns stored in the malicious passwords information database. We termly statistic the patterns and calculate probability of each pattern's occurrence.When a targeted attack on patterns occurred, a large number of passwords with the same pattern will flow into the malicious passwords , which will make that pattern's probability rapidly ascend.

- Probability re-estimation based on changed patterns: In our probability estimation system, the probability calculated is highly relevant to the patterns' probability used by attackers. So when pattern's occurrence changed, we can also update user's password's intensity. If a password's probability of being guessed is highly raised and up to the threshold, then it will be recognized as being targeted attacked and the system will give users alarm.

6. CONCLUSION AND REFLECTION

6.1 Research conclusion

Our research introduces a pioneering approach to password security by analyzing attacker-generated passwords,which is an under-explored area in current literature. We have developed a password information extraction system to evaluate the strength of malicious passwords, aiming to enhance defense against online brute-force attacks. This method diverges from traditional approaches that primarily focus on understanding user password construction habits.

6.2 Limitations of methods

The evaluation of user password strength in our study is overly simplistic. We rely on basic frequency estimation probability methods to assess the usage of words and patterns in passwords. Additionally, our password calculation approach merely multiplies the probabilities of pattern, word, and character insertions, assuming low correlation among these elements. However, this method is vulnerable to manipulation by attackers who can distort statistical insights by flooding the system with large volumes of meaningless passwords.

Furthermore, our current methodology lacks extensive experimentation. While we have been conducting a demonstration to estimate password strength using collected malicious passwords, our system is designed as a dynamic password evaluation system. Future iterations of our research should involve building a more sophisticated processing system and conducting longer-term experiments to assess its effectiveness and availability over time.

6.3 Possible Improvements

To enhance the accuracy of weak password assessments, we propose integrating additional parameters of password formation into our probability calculations. Techniques such as deep learning could significantly improve assessment precision. Furthermore, exploring recommender system techniques could refine our approach by analyzing attacker password preferences. Additionally, employing Markov models [5] offers a potential avenue to enhance accuracy in predicting attacker password strategies.

6.4 Future exploration

Our method has practical applications for websites, web servers, and systems relying on password-based authentication. By providing users with real-time evaluations of password strength, administrators can actively monitor and mitigate brute-force attacks. Analyzing attacker passwords also provides insights into ongoing threats to password security systems, empowering informed decision-making by administrators.

Future efforts will focus on refining our theoretical framework and conducting experiments to validate the effectiveness of our methods. We aim to transition our system from a static demo to one capable of dynamic password assessment in real network environments. Additionally, we will explore advanced pattern and word extraction techniques to further enhance password strength assessment. The current lack of research on attacker password composition and distribution underscores the need for continued exploration in this critical cybersecurity domain.

7. REFERENCES

[1] A novel dictionary generation methodology for contextual-based password cracking. Accessed: 2024-07-18.

[2] L. Arezina. Password statistics for 2020. `https://dataprot.net/statistics/password-statistics/`, Nov 2019. Accessed: 2024-07-18.

[3] fffaraz. fakessh. `https://github.com/fffaraz/fakessh`, 2024.

[4] M. Jakobsson, M. Jakobsson, and M. Dhiman. The benefits of understanding passwords. *Mobile Authentication: Problems and Solutions*, pages 5–24, 2013.

[5] J. Ma, W. Yang, M. Luo, and N. Li. A study of probabilistic password models. In *2014 IEEE Symposium on Security and Privacy*, pages 689–704, 2014.

[6] J. Owens and J. Matthews. A study of passwords and methods used in brute-force ssh attacks. In *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, page 8, 2008.

[7] R. B. Varne and R. V. Mane. Captcha: A robust approach to resist online password guessing attacks. In *2014 International Conference on Advances in Communication and Computing Technologies (ICACACT 2014)*, pages 1–6, Mumbai, India, 2014.

[8] R. Veras, C. Collins, and J. Thorpe. On semantic patterns of passwords and their security impact. In *NDSS*. Citeseer, 2014.

[9] Y. W. Mining threat intelligence from billion-scale ssh brute-force attacks. Accessed: 2024-07-18.

[10] M. Weir, S. Aggarwal, B. De Medeiros, and B. Glodek. Password cracking using probabilistic context-free grammars. In *2009 30th IEEE symposium on security and privacy*, pages 391–405. IEEE, 2009.