# A Survey on Security and Attack Aspects of Passwords

Yosef Ashibani
Department of Electrical, Computer and Software Engineering
University of Ontario Institute of Technology
Oshawa, ON, L1H 7K4 Canada

## ABSTRACT

Despite many weaknesses, passwords are still mainly used, and will continue to be used in the near future, for the user authentication process. Passwords remain one of the important pillars of the protection structure even though they are not sufficiently robust against well-designed attacks. Thus, users need to select and protect robust passwords. The consequences of password disclosure to adversaries might have disastrous results, which in turn would increase the need to focus extensively on security factors in order to strengthen and protect passwords. Humans usually create far from random passwords that are vulnerable to attack. One important factor in estimating the impact of attacks and the strength of created passwords is to understand the ability of attackers to deduce passwords. Unfortunately, many efforts at strength estimation have failed. The main reason for this failure is that these efforts specifically focus on protection against *Brute Force attacks*. Other attempts have tried to design attacks against user passwords in order to test their strength and to accordingly improve them. This idea is expensive and insufficient to uncover or perhaps to identify professionally designed attacks. Another technique is to assign *robust randomly generated passwords* which could provide higher security. Assigning passwords by systems ensures that the users do not reuse the same passwords for different applications. On the other hand, it is challenging for users to remember such passwords. This has eventually led to the idea of using software management tools specifically designed for storing user passwords; however, the single point of failure will be the main drawback of such a method. Since password are remain the popular method for authentication, and will continue to be in the future, password security problems have become a global issue. Thus, designing robust, secure, and efficient password creation techniques needs to be urgently undertaken and with the utmost care. This paper briefly summarizes the most common attacks against passwords as well as some related works that have been conducted in the field of security and usability of passwords.

## Keywords

Password usability and security, password strength estimation, password manager, user replaceable passwords, graphical passwords.

## 1. INTRODUCTION

Although they have many weaknesses and are targeted by attacks, text passwords have considerable advantages because of their familiarity and ease of implementation. Moreover, users do not need to carry them around on their person [3]. The explosive increase in daily use applications that require authentication through passwords has increased the number of passwords in use for almost everyone. This requires remembering a number of related passwords that would burden some people and also leads users to opt for weak passwords, which are easy to remember and can be reused for different applications.

Although considerable research has been conducted into developing passwords, many users still prefer to use passwords that can be easily remembered, which in turn are easily broken by attacks. Many users believe that adding special characters, such as "!", to the end of the password, will enhance its security and make it more difficult to spell and predict. However, in an actual attack scenario, this is very predictable. Others suppose that using "non-released special data", such as a name, birthday, or wedding date, will be secure, but do not consider that *automated guessing attacks* take into account common names and events. Generally, many users anticipate that using arbitrary mixed upper and lower case characters reduces the probability of attacks, whereas many try to include some other ways, such as combining unrelated words or developing unique phrases that in turn produce strong passwords [4]. Many users choose to assign weak passwords for low value accounts as a result of analyzing the costs and benefits of such accounts, making them more prone to attacks. Those users often think about the earned value of using the passwords [5]. Moreover, misconceptions about what makes a password robust overshadows these users so that many of them try to achieve security parameters that meet their desired security levels.

Many attempts have been made to deal with password attacks resulting from users' disregard or lack of interest in creating strong passwords [6]. In a study by Ur et al. [4], some participants used different methods to choose their passwords, including selecting words and phrases such as a year or an emoticon. For low security accounts, they tended to choose names of places that they had visited while, for high security accounts, highly non-predictable words were the preference. These users often chose passwords based on personal topics. Other methods that were used include passwords from web pages, such as the name and purpose of the website. Some participants in the study included capital letters and punctuation while others chose to include digits and symbols expecting their passwords to be sufficiently strong [4]. As a result, there is a real need to develop an easy method of creating passwords with enhanced strength. *Strength of password means how much it will cost attackers (in terms of time, effort and money) to reach their goal:* in other words, *the number of attempts needed by adversaries to guess a used password* [7]. In order to be able to implement a

mechanism for generating passwords with enough strength, a study of the attack properties and the vulnerabilities of currently used techniques needs to be initially performed.

The following three subsections of the paper will focus on: 1.1) Password authentication approaches; 1.2) Common attacks on passwords; and 1.3) Analysis of password strength factors.

## 1.1 Password Authentication Approaches

There are various types of password attacks, the number of which is dramatically increasing along with an increase in distributed communication networks and applications. Broken passwords can be used by criminals to breach an entire system or by security analysts for security analysis purposes. Hence, users should be aware of probable password attacks so that they will be able to look for alternative methods of selecting robust passwords and also protect their chosen ones. Passwords are mainly used for authentication purposes in different ways. The most common authentication approaches are:

- ***Conventional Password Scheme*:** This type is the classical method of password authentication that works by checking the entered password and username in a stored file inside the system. It is a simple method but the most vulnerable to attacks.
- ***Typing Dynamics*:** This method, which basically depends on pressed keys and their timing, analyzes the way that the user enters a password. It stores all the key names, key pressing and releasing times which could be used to manipulate ways to deduce passwords. While it is effective against shoulder surfing, the main drawback of this method is the refusal rate as a result of the user's typing speed, which can be affected by the user mode.
- ***Graphical Passwords*:** This method works by selecting a manner or drawing of an object, selecting graphical objects in a specific order, or connecting some objects in a pre-chosen manner in a graphical user interface. Although this method reduces the probability of shoulder surfing attacks, it needs more processing time than other techniques.
- ***Biometrics*:** This type is basically built on the image processing technique. It works by comparing the selected image (password) with that previously stored. It can be implemented with many techniques, such as biometrics of signature verification, face recognition, or fingerprints. This method could avoid many types of attacks, but is expensive.

## 1.2 Common attacks on passwords

1. ***Brute Force Attack*:** This is the most widely popular attack since it does not need prior knowledge of which password strategy has been used to create the password. It tries all possible consecutive values of the targeted password in order to break it. This method is generally used for passwords that are stored in an encrypted manner. Some operating systems store their passwords in an encrypted file inside the operating system. If this file is stolen, the attacker may apply the Burst Force method to break the saved passwords. The hash values of the passwords are usually stored in the system. This method is easy to implement but may take a long time to achieve the targeted passwords. For example, by this method, breaking a password of four characters needs $26^4 = 456976$ combinations, if all four characters are in lower or upper case. This means that this method is effective for small passwords. This method can also be employed by advanced attackers by using numerous forms of hybrid attacks such as combining dictionary attacks (see below) with Brute Force attacks [8]. Additionally, this type of attack is effective against weak selections, especially classical passwords.
2. ***Dictionary Attack*:** This method, first proposed in 1979 by Morris and Thompson, can be partially considered as faster than the Brute Force method. Starting from the principle that many users adopt related information, such as birthdays, names, or pet's names, as their passwords, this method tries to match the password with words that are frequently used on a daily basis and are collected in a dictionary. Most of the words in the dictionary are collected according to the most widely used vocabulary with the probability that those words are used as passwords. The main limitation of this method is that the password that is being searched might not be included in the dictionary. Modern dictionary attacks combine wordlists, which usually contain natural language dictionaries in addition to stolen passwords, and string transformations that will be used to modify wordlist entries to create additional guesses. This transformation is known as mangling rules, while this type of attack is referred to as a *Mangled wordlist attack* [3].
3. ***Video Recording Attack*:** As the name implies, the attack is implemented by recording the user's password entry more than one time in order to analyze the entered passwords. Most often regarded as the easiest way to obtain access to a desired password, this method does not need much time or calculation.
4. ***Malware attack***: This is a software-based attack that gains access to the target device without the user's knowledge. It can disrupt access to private information, or harm the computer system.
5. ***Key Loggers*:** This is a type of malware attack that works by installing a software in the user's system. The Key Logger software is installed either directly by the attacker or by deceiving the user to download a particular script. This software in turn monitors all the user's activities. The mission of this program is to record all pressed keys and send the results to the attacker. This attack, which results in discovering the password that can be later used to access the targeted system, is effective against conventional password methods.
6. ***Shoulder Surfing*:** Also known as *spying*, this type of attack tries to gain any knowledge that might lead to obtaining the password by direct observation or by using external recording devices [6]. The attacker can observe the password with different methods, such as hearing the number of keys that the user has pressed and then trying to guess the pressed keys, relying on what has been observed. Field glasses can also be used. In this approach,

the attacker tries to use any applicable way to observe any information that will lead to obtaining the password. This method is effective versus conventionally chosen passwords.

7. ***Phishing Attack*:** This is a type of attack where the user is attracted to make a registration profile on a fraudulent website so that the entered data will be used later by unauthorized software to capture any actions by the user. These actions can then be analyzed to find the login credentials of that particular user [9]. This attack is successful with classical passwords.
8. ***Reply Attack*:** This is also known as playback or reflection attack, which is actually a network attack. In this type of attack, the attacker eavesdrops on the connection between the sender and the receiver and tries to capture the authentication data. The attacker will then use the authentication information as "proof of identity" to establish a future connection with the receiver.
9. ***Probabilistic context-free grammar (PCFG)*:** A type of guessing attack proposed by Weir et al. in 2009, this is based on the realization of the similar structure that passwords often follow [7]; as such, it can accurately model password distribution. It has been used specifically with training data sets of password infractions to model passwords, followed by guess generating. PCFG allocates probabilities to the structure of the password and the string components of that password. The main use of PCFG is to parse or generate sentences [10]. PCFG has been used by many research studies to compute "guess ability" [3]. As an example, the structure of string *laba123* is four letters and three digits while the component strings "*123*" will be added to a list of three digit strings that has been created.
10. **Markov models:** The Markov model, a tool used for password strength estimation, is suitable for estimating password probabilities and better than probabilistic context-free grammars, as concluded by a recent work [9]. It has also been found that the Markov model with a particular configuration was more efficient than other methods at password guessing for some datasets of leaked passwords [3].

### 1.3 Analyzing Password-Strength Meters

Traditionally, a simple evaluation method to test the strength of a password can be achieved by measuring the number of special characters, such as symbols, digits, and lower and uppercase characters that the password includes. This approach is ineffective for current attacks. Recent studies suggest that a way to evaluate password strength is by the number of tries an attacker needs to determine the password. Other studies evaluate password strength against a Brute Force attack. However, most of the suggested password strength models were built on the basis of heuristic approaches which most likely do not ensure password resistance against guessing attacks [7]. Amico and Filippone in [7] conducted a first study of password strength which provided a reliable estimation of the rate of success of recent modern and costly attack techniques. This study shows that it is possible to evaluate a password by guessing attempts. This leads to the conclusion that any attacker must be forced to spend as much time as possible and that a balance of security versus usability should be taken into account when analyzing password strength. Moreover, the number of guessing attempts on passwords has been correlated against various types of attacks. Amico and Filippone propose a new approach to calculate the number of attempts that an attacker needs to achieve a required password. This method can be applied to various sets of probabilistic models with few resources. The evaluation of password strength has been tested against a large number of attacks including those that are expected to be very expensive to handle along with available simulated guessing methods. Carnavalet and Mannan in [11] show that providing users with feedback about the strength of their chosen passwords influences their choice in providing better passwords. It is important to indicate strength meters that should enhance the strength of the chosen passwords at the time of creation.

Many meters are usually employed by software evaluation tools (checkers) to assess a password's strength at the time of creation when the user will accordingly receive feedback on its strength. This in turn should encourage the user to choose better passwords. The common evaluation meters adopted by many applications are:

- **Charset and length requirements:** These classify the chosen password by many factors, such as: setting a minimum length; enforcing a maximum length; requiring use of certain character sets; or disallowing usage of some characters with others [11].
- **Strength scales and labels:** Scaling standards vary among different checkers. The strength of passwords can be scaled, for example, as secure, short, fair, weak, or not secure. These scales differ from one software to another [11].
- **User information:** User specific information, such as name, email address, and contextual information, are considered as a weak password.

## 2. Related Work

The following four subsections of the paper summarize four related works that attempt to improve the security and usability factors of passwords: 2.1) Estimating strength of the password; 2.2) Improving memorability of randomly generated passwords; 2.3) Password managers; and 2.4) Graphical password schemes.

### 2.1 Estimating Strength of Passwords

The first study of this technique was in 1979 by Morris and Thompson, who mention that it is possible to estimate a large number of passwords that have been used in the UNIX system by Dictionary and Brute Force attacks [7]. This was a proactive action used to estimate the strength of the suggested (chosen) passwords to avoid or reduce the occurrence of attacks. Most password checkers use multiple rules to evaluate the strength of passwords and the generality of these rules is simple. Actually, using such weak rules has proven to be an unreliable indicator for evaluating password strength. Another password classification idea involves reducing the password's existence in databases. Eventually, the Markov model was produced as a secure password predictor [9]. It has been found that analyzing a large set of passwords is often a challenge for the used algorithm; however, achieving the required result depends on the algorithm's configuration. Oftentimes, professionals' rules of password estimation can be approximated using automated guessing algorithms. On the other hand, relying

on only one guessing algorithm for password cracking is a risk that should be taken into consideration [3]. Dürmuth et al. in [9] produced a password guesser (OMEN) based on the Markov model, which performs better than other available password guessers. This model is able to estimate more than 80 % of attempted passwords, indicating that this method can be described as a preventive measure [9].

Amico and Filippone [7] propose an easier and newer approach to estimating the needed number of guesses by using modern attacks. They provide theorems that show the correctness and approximation of this method. They also claimed that the number of password guesses correlates against different attacks. Accordingly, it is suggested that testing passwords against different attacks would produce higher strength value.

## 2.2 Improving Memorability of Randomly Generated Passwords

Another method used to generate strong passwords against guessing attacks is the use of a randomly generated password. This method generates passwords by taking pseudorandom numbers as input and creates random passwords by applying some operations. However, while the randomly generated passwords can be complex and strong against some attacks, they are difficult to remember.
In order to enhance password memorability, Huh et al. in [8] propose a technique for replacing some of the characters chosen by users in randomly generated passwords. The length of generated passwords in this study is eight characters, which is the common standard. The main focus of this research compares the usability and security among users' chosen passwords and randomly generated passwords with different policies: policy 1(1-change) to policy 4 (4-changes). The study has been designed according to the following three hypotheses:

- Memorability increases as the number of replacement characters increases.
- Security decreases by increasing the allowed number of replaced characters.
- There is no statistical evidence regarding the differences between complexity and memorability related to passwords chosen by users. This policy produces superior security rather than complexity.

The number of character replacements is defined by each policy (0-change to 4-changes). For evaluation purposes, the security and usability of this method have been studied online, on a large scale.

In the initial stage of this study, 5,412 participants took part, whereas 3,839 participants completed the second stage for five weeks. It was clear that policies 3-changes and 4-changes outperformed policy 0 in memorability while the cracked percentage was 0% in 0-change, 1.21% in 3-changes, and 5.82% in 4-changes. There was no significant memorability difference between policies 3 and 4 and the estimated entropy was lower than 0 policy. In memorability, policies of 3 and 4 character replacements surpassed the original randomly generated passwords by 11% to 13%, whereas the cracked password percentage slightly increased. When compared with the complexity of user generated passwords, policy 4-changes did not demonstrate any progress in memorability. Finally, lab results show that the surpass scheme, proposed in [8], outperformed user generated password policy in security with 21% fewer cracked passwords [8], but did not provide any evidence of proportional improvement between memorability and the number of replaced characters.

## 2.3 Password Manager

As the number of text passwords needed by users increases, the ability to choose and remember all of them becomes more difficult, particularly in web applications. Furthermore, text passwords are still preferred by many users since they are inexpensive and easy, and offer a simple authentication approach that avoids privacy breaches.

One proposed solution to decrease the difficulties related to text passwords is the use of a password manager (PM), which is one of the simplest methods of managing passwords without memorizing or saving them in written form. Password manager software is a database that stores all user passwords and usernames. Access to the database is restricted by using a username and password. Thus, the user only needs to remember a robust master password and the user name. By using a password manager, the user can choose a strong password for each web application, without the need to remember them [12]. For the application, the password manager re-generates the required stored (encrypted) password and sends (autofill) it to the application on behalf of the user.

Automatic autofill populates the form that contains the fields of username and password at the time of loading the login page without any user interaction, such as PMs in Chrome, Firefox, Safari, and LastPass. Manual autofill needs some user interaction prior to the autofill process, such as username typing, or button pressing, such as PMs in Keeper and KeePass [13]. Some password managers impose manual interaction in particular cases.

The following two subsections focus on: 2.3.1) Attacks on password managers and 2.3.2) Improving password managers.

### *2.3.1 Attacks on Password Managers*

- **Sweep attacks:** Such attacks steal passwords from multiple sites at a time. This is accomplished by enforcing the user's browser to visit a vulnerable malicious site without the users' knowledge, then applying (injecting) JavaScript code into the site's webpage. This code exfiltrates passwords and sends them to an attacker [13].
- **Injection Techniques:** Logging into a page within the same origin domain is not enough since most PMs link the preserved passwords with domains while ignoring the paths of the login pages. In this situation, an attacker can insert a fraudulent login form in the same domain to any page and then begin a password extraction attack in this page [13].
- **Password Exfiltration:** Gaining access to password fields, then sending them to a self-controlled server. This can be achieved in two ways:
  - ***Stealth***: Loading an attacker remote-controlled page in a hidden iFrame, which in turn passes the password as a parameter to the attackers' page [13].
  - ***Action***: Modifying the action properties of the login form to submit credential data to the attacker's self-controlled page [13].

### 2.3.2 Improving Password Managers

- ***Forcing User Interaction***

In order to prevent sweep attacks, PMs should always require user interaction before auto-filling the form. This interaction can be achieved by, for example, typing the username or clicking a button. In addition, the domain name should be shown to the user before the auto-filling process takes place [13].

- ***Secure Filling***

Regardless of the filled password submission approach, there is no supported method to prevent stealing passwords after filling the login form with the password. The filled password can be read by JavaScript code and submitted to an attacker controlled page. Silver et al. [13] suggested a defensive approach to this problem as follows:

- At the time of creation, the password manager stores the username and password along with any providable information (actions).
- Once the autofill is complete, the password fields turn into unreadable state, preventing stealth exfiltration, by JavaScript code (autofill is in progress). If any change occurs in the password and username fields, the autofill is voided and password fields are cleared, meaning that they can be read one more time by JavaScript.
- Immediately after the delivery of the autofill, the form's action is compared with the first saved action at the password initiation time. If matched, the action has not been changed by any malicious site. The form will then be allowed for normal submission. If not, the password fields will be erased and the submission will fail.

## 2.4 Graphical Password Schemes

Because text passwords suffer from security and usability drawbacks, graphical passwords have been proposed since the end of the twentieth century for the purpose of improving password usability and memorability.

In general, graphical passwords are categorized according to memorability and into three categories: recall (remembering without cueing); cued recall (providing an external cue); and recognition (memorizing an image portfolio) [1].

Many studies have been conducted into the following three aspects of graphical passwords: memorability, security, and recall based systems. In fact, these three schemes are vulnerable to attack. Recall based schemes are vulnerable to Malware attacks, cued recall schemes are vulnerable to Shoulder Surfing and Malware attacks, while recognition based schemes are resistant to Phishing attacks but are susceptible to Shoulder surfing. In conclusion, graphical passwords are generally more vulnerable to Shoulder Surfing attacks than text passwords [1], whereas advanced Malware protection techniques are required for graphical passwords. Taking into account the most serious attacks, it is difficult to determine if text passwords are more vulnerable than graphical passwords. Nonetheless, graphical passwords remain practical, since they provide greater usability than text passwords.

## 3. Discussion & Future Directions

The more advanced attacks are always designed by highly professional attackers. However, the way to guessing the nature of any new approaches mostly depends on understanding traditionally discovered attack schemes. Moreover, simulating all available guessing attack algorithms is impractical and expensive. Thus, estimating the strength of any password might help to slightly improve password strength, but this is expensive and will not be enough to uncover professionally designed attack techniques.

In the technique proposed by Huh et al. [8] to improve the memorability of randomly generated passwords, there is no significant memorability difference between policies 3 and 4. The estimated entropy shows that it is lower than 0 policy. In the case of memorability, policies of 3 and 4 that have undergone character replacements surpassed the original randomly generated passwords by 11% to 13%, whereas the cracked password percentage is slightly increased, which will weaken the strength factor of the random number.

Password Manager is a better approach then either method of estimating the strength of passwords or improving the memorability of randomly generated passwords. The improvement that has been discussed in paper [10] will help protect passwords. Frequent updating of password managers is necessary in order to shield against newly designed attack approaches and must be supported by an application provider to achieve better security. Even when password strength is reasonable, a low strength flag could urge the user to create a much stronger password.

The graphical password technique can be one of the better ways to support password strength. Graphical password schemes can be enhanced by using an advanced approach with more parts for the graphical password space, including more items, which would be much stronger against attacks. On the other hand, this technique needs higher processing time than regular methods and also brings a higher implementation cost.

In order to minimize losses as much as possible against password attacks, a dynamic password scheme, which periodically changes the user's password, could be implemented in an effort to decrease attacks. Whatever time the user enters the system, she/he will be given a new choice of updating the password for the next entry. This idea, which might be a burden for users to follow, will provide more protection against attacks. Finally, the way forward depends on the balance of security, usability, and data value and how they are measured by users.

Further research should focus on measuring the password strength achieved by replacing a specific number of known characters in a twelve digit randomly generated password by a pseudorandom generator source such as a cryptographically secure pseudo-random number generator (CSPRNG).

## 4. REFERENCES

[1] M. N. Al-ameen and S. Scielzo, "Towards Making Random Passwords Memorable : Leveraging Users ' Cognitive Ability Through Multiple Cues," pp. 2315–2324, 2015.

[2] A. Jøsang and I. Loutfi, "Passwords Are Not Always Stronger on the Other Side of the Fence," *Commun. ACM*, vol. 42, no. 12, pp. 41–46, 2015.

[3] B. Ur, S. M. Segreti, L. Bauer, N. Christin, L. F. Cranor, S. Komanduri, D. Kurilova, M. L. Mazurek, W. Melicher, and R. Shay, "Measuring real-world accuracies and biases in modeling password guessability," *24th USENIX Secur. Symp. (USENIX Secur. 15)*, pp. 463–481, 2015.

[4] B. Ur, F. Noma, J. Bees, S. M. Segreti, R. Shay, L. Bauer, N. Christin, and L. F. Cranor, "' I Added "!" at the End to Make It Secure ': Observing Password Creation in the Lab," pp. 123–140, 2015.

[5] P. Dunphy, V. Vlachokyriakos, A. Thieme, J. Nicholson, J. Mccarthy, and P. Olivier, "Social Media as a Resource for Understanding Security Experiences: A Qualitative Analysis of #Password Tweets," pp. 141–150, 2015.

[6] D. Florencio, C. Herley, and V. O. Paul C, "Password Portfolios and the Finite-Effort User : Sustainably Managing Large Numbers of Accounts ∗ ," *Usenix Secur.*, 2014.

[7] M. D. Amico and M. Filippone, "Monte Carlo Strength Evaluation : Fast and Reliable Password Checking Categories and Subject Descriptors," *ACM*, pp. 158–169, 2015.

[8] J. H. Huh, S. Oh, H. Kim, and K. Beznosov, "Surpass : System-initiated User-replaceable Passwords Categories and Subject Descriptors," pp. 170–181, 2015.

[9] M. Dürmuth, F. Angelstorf, C. Castelluccia, D. Perito, and A. Chaabane, "OMEN: Faster Password Guessing Using an Ordered Markov Enumerator," *Int. Symp. Eng. Secur. Softw. Syst.*, pp. 119–132, 2015.

[10] R. Chatterjee, J. Bonneau, A. Juels, and T. Ristenpart, "Cracking-Resistant Password Vaults using Natural Language Encoders," *IEEE Symp. Secur. Priv.*, 2015.

[11] X. D. C. De Carnavalet and M. Mannan, "From Very Weak to Very Strong : Analyzing Password-Strength Meters," *Ndss 2014*, no. February, pp. 23–26, 2014.

[12] Z. Li, W. He, D. Akhawe, and D. Song, "The Emperor's New Password Manager: Security Analysis of Web-based Password Managers," *23rd USENIX Secur. Symp. (USENIX Secur. 14)*, 2014.

[13] D. Silver, S. Jana, E. Chen, C. Jackson, and D. Boneh, "Password Managers: Attacks and Defenses," *USENIX Secur. Symp. (USENIX Secur.*, pp. 449–464, 2014.