An Analysis of Persuasive Text Passwords

Thi Thu Trang Nguyen
Department of Informatics
Vietnam Trade Union University
Hanoi, Vietnam
Email: thutrang2x01@gmail.com

Quang Uy Nguyen
Faculty of IT
Le Quy Don University
Hanoi, Vietnam
Email: quanguyhn@gmail.com

***Abstract*—Text-based password is widely considered as the most ubiquitous authentication scheme in computer systems nowadays. However, text-based password are vulnerable to some attacks such as brute-fore attack and dictionary-based attack. Consequently, a large number of research has focused on enhancing the security strength of text-based password. Persuasive Text Passwords (PTP) is a technique to improve password strength by adding some random characters to user's password. In this paper, we compare PTP with some common password policies. Thanks to this, some flaws of PTP are determined. An improvement of PTP is proposed to alleviate its drawbacks. The improvement is implemented by combining PTP with a password policy. The experimental results show that the new version of PTP is better than the original version in both security and usability.**

I. Introduction

In most computer systems, user authentication is the crucial building block and the main layer of defense. Generally, There are four means of authentication, which can be used alone or in combination [1]:

- Something the individual knows: Examples includes a password, a personal identification number (PIN), or answers to a prearranged set of questions. Perhaps, this is the most popular form of authentication in computer systems [1].
- Something the individual possesses: The tokens such as electronic key cards, smart cards, and physical keys are used for authentication. This method is used widely in banking systems.
- Something the individual is (static biometrics): Examples include recognition by fingerprint, retina, and face. This and the below technique (dynamic biometrics) are often used as an alternative authentication method in information systems.
- Something the individual does (dynamic biometrics): Examples include recognition by voice pattern, handwriting characteristics, and typing rhythm.

Among the above methods, password-based authentication is often considered as the most ubiquitous technique [2]. Using passwords for authentication has several advantages compared to other authentication techniques [3]. The main benefit is the simplicity of authenticating by password. Password can easily be implemented in most computer systems without requiring an extensive computer/server modification. Moreover, users have already been familiar with using passwords in many systems [4]. Thus, password-based authentication has extensively been implemented in computer systems [1].

978-1-4799-7492-4/15/$31.00

However, generating passwords that obtain good security and usability is a challenging task. Users often select passwords that are highly predictable. Conversely, randomly-generated passwords may be difficult for users to recall. Some explicit password policies (referred to as password policies hereafter) have introduced to mediate between these two goals by forcing users to choose complex passwords. For example, common policies are to mandate users include a mix of characters or use passwords of some minimal length [5]. However, these policy mechanisms are hindered by user's weak understanding in the actual effectiveness of their passwords against real threats. For example, a policy that requires a user to include at least two digits in a password will often result in the user simply appending 12 on the end of an insecure password. A password such as "mylove12" is often easy to be cracked by hackers.

Several techniques such as graphical password [6] and smart card based password [7] have been proposed [8] to improve password strength. Recently, external password creation policies have received a significant attention from researchers [5]. External password creation polices modify a user's password by adding an amount of randomness to the password. An example of this would be adding two random digits to random positions in user's password. The advantage is that they guarantee a certain degree of randomness in user's passwords. Usually, this is implemented by allowing users to select their base password, and then adding random characters to it that users would then have to remember.

There has been research in examining the effectiveness of external password creation policies. In [9], the authors attempted to make these methods more friendly by assigning the user a random passphrase instead of a random password. Other approaches have attempted to add randomness after the user selects their password by inserting random values to it [5]. Perhaps the best study has been [10] where the authors proposed Persuasive Text Passwords (PTP) and examined user acceptance of this policy. One interesting result was that strict policies did not result in strong passwords since users started selecting simple initial passwords that they can recall better.

In this paper, we conducted a further analysis of PTP. Two main contributions of the paper are:

- A comparison between PTP and common password policies is examined.

Fig. 1. PTP password creation before applying the persuasive improvement.

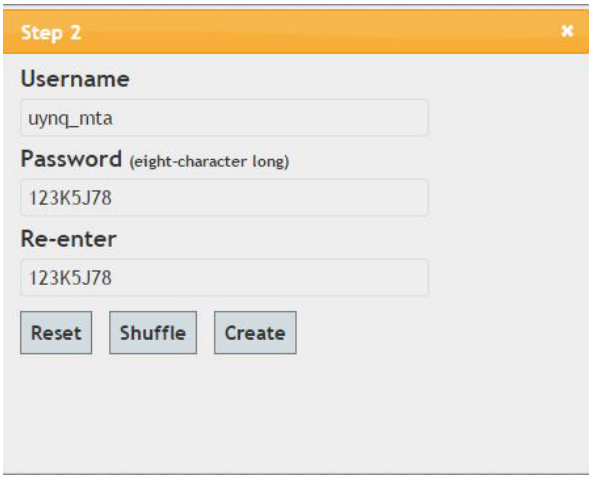

Fig. 2. PTP password creation after applying the Replace-2 persuasive improvement.

Fig. 3. PTP password creation after users shuffle their password.

- A method to improves security strength and usability of PTP is introduced.

The remainder of the paper is organized as follows. In the next section, we present the background of PTP. The detailed description about the experiments is presented in Section III. The comparison between PTP and there password policies is presented in Section IV. The improvement of PTP is introduced in Section V. Section VI concludes the paper and highlights some potential future work.

II. Persuasive Text Passwords

Persuasive Text Passwords (PTP) is a password system which helps users to select more secure passwords. In the password creation process, users enter their initial password. After that PTP improves password security by placing some random characters at random positions in the user's password. For example, one of the technique in PTP is to replace some characters in users initial password by some random characters. In Figure 1, an user enter their eighth-character password in Step 1. After that, the user click on Improve Button and go to Step 2 in Figure 2). In Step 2, the user will receive a new password that is generated by replacing two random characters in the original password by two new characters. If the user find that this password is rememable, he will re-enter the password in the below row and click on Create button to finish creating password. However, if the password is difficult to recall, he can find other passwords by clicking on Shuffle button (Figure 3). The user can continue clicking on Shuffle button to find new passwords until he sastifies with a specific password.

It is noted that the random characters added to the user's initial password are chosen uniformly from all characters available on English US keyboards, except the blank space. PTP provides a method to create passwords that have the usability of purely user-chosen passwords and the security of system-assigned passwords. It has been shown that adding random elements to user-chosen passwords will enhance their security while maintaining sufficient memorability [10]. Several variants of PTP are follows.

- Reload. The system-assigned characters are presented to users before they create their password. The characters are randomly positioned in the first eight character slots. Users create their password based on system-placed characters.
- Replace. Users are allowed to select an initial password as they would for an usual password system. The system choose at random some characters in the users passwords and replace them by randomly-generated characters. See Figure 2 for an example of Replace with two system-selected characters.
- Insert. This method lengthen the users password by inserting some randomly-generated characters to the password. Users first select an initial password as usual. The system then adds randomly-selected characters at random positions between user-chosen password characters.
- Swap. Users select an initial passwords of a predefined length. After that, the system choose some characters at random and swaps them together.

In [10], three variants including Preload, Replace and Insert were tested. The results showed that the Preload variant has a weakness in which users would simply repeat the system-assigned characters. For example, if presented with

Fig. 4. The count down process to refresh user short-term retention.

"__AA__BB__8", users would create a password such as "AAAABBB8". Thus, the password obtained by Preload is not as strong as expected. For Replace and Insert, they possess better security and usability capacity. However, comparison between PTP and common password pocilities was not examine in [10]. This paper aims to extend the previous research in [10]. In the next section, we present a detailed description about the experiments in this paper.

III. Methods

We constructed an online authentication system for the experiments at the following address [1]. We asked three hunderes first and second year students at Vietnam Trade Union University to participate in the experiments. Similar to Forget [10], the participants were asked to create a password for their bank account. In all password schemes, the experiments includes following five steps:

Create. Users selected a PTP variant that they will involve. The system placed the number of characters appropriate to the condition of the selected variant beside the password field. The users entered a initial password by their decision (Figure 1). The users re-typed their password on the second row and clicked on Improve button to enter the shuffling step. In this step, users were allowed to shuffle the characters as much as they wanted (Figure 3).

Confirm. Once users found a password that they could remember, they then re-typed the improved password on the second row and pressed the Create button (Figure 2). Then, an account with an username and an improved password will be generated for the users.

Distraction. For 45 seconds, users would count down in threes from 45. After each of three second, a new random four-digit number was generated and displayed in the screen (Figure 4). This type of distraction refreshes their textual working memory and simulates a longer memory by focusing participants attention on a separate task [11].

Login. Participants attempted to login to the system with their improved password, which was created in the previous steps. If they made a mistake, they could try to login again, or finish the trial if they forgot their password.

[1] http://nguyenthutrang.name.vn/

TABLE I. Number of participants in each policy testing and PTP variants

Schemes	Participants
Policy-1	39
Policy-2	34
Policy-3	34
Insert-1	36
Insert-2	32
Replace-1	43
Replace-2	47
Swap-2	36

Questions. After users have finished creating and using password steps, the users were asked several questions about their opinion on creating and using password with PTP technique. A free survey tool, SurveyMonkey, was used this purpose. Participants answered the questions on a 5-point Likert scale, from very easy to very difficult. The results of the survey will be examined in the future research.

IV. Comparing PTP Variants and Password Policies

The participants who involved in the experiments were 301 students at the Faculty of Basic Science at Vietnam Trade Union University. Three common password policies were tested. The first policy requires that passwords have the length of eight characters (Policy-1). The second policy requires that passwords have the length of eight characters containing both digits and letters (Policy-2). The third one is the strongest policy that requires passwords with eight-character length including lower letters, upper letters and digits (Policy-3). Perhaps, these policies are the most well-know password policies to enforce the complexity in users password [12]. The variants of PTP examined include Insert-1, Insert-2, Replace-1, Replace-2 and Swap-2 [2]. The number of participants involving in the experiment for each password scheme is presented in Table I.

Three measures were used for comparing between PTP variants and password policies . These measures include the average time for users to create a password and login to the system using their password (Timming), the frequency of successful login to the system for each password variant (Success Rates) and the security strength of each password scheme (Security Strength).

Timings. We first measured how long did it take for users to create a password and how much time users needed in order to login to the system. Table II shows the time that participants took to complete each step (Confirm and Login) in password techniques. The time for Confirm step is calculated from the time when users started entering an username until they clicked on the Create button to create a complete account. The time for Login step is measured when users entered their username until they clicked on Login button. These values are then averaged over all Participants for each password scheme and presented in Table II.

[2] Our preliminary experiments showed that Insert-3 and Replaced-3 are complex for users to recall their password. Consequently, users often fail to login to the system using their created password. Thus, two these variants are not included in this paper.

TABLE II. SECONDS TAKEN PER TRIAL TO COMPLETE THE EXPERIMENT PHASES ACROSS PTP CONDITIONS AND PASSWORD POLICIES.

Schemes	Confirm	Login
Policy-1	63.5	27.1
Policy-2	87.8	19.7
Policy-3	157.2	32.8
Insert-1	144.9	56.1
Insert-2	209.2	56.3
Replace-1	193.2	34.8
Replace-2	239.5	54.5
Swap-2	104.1	27.3

TABLE III. LOGIN SUCCESS RATES OF PTP VARIANTS AND PASSWORD POLICIES.

Schemes	Total	Success	Percent
Policy-1	39	35	90.0
Policy-2	34	31	91.2
Policy-3	34	30	88.2
Insert-1	36	31	86.3
Insert-2	32	19	59.4
Replace-1	43	37	86.0
Replace-2	47	41	87.2
Swap-2	36	31	86.1

It can be seen from Table II that users in PTP variants need longer time to create and use their password. Apparently, the average time for confirming password in PTP variants is often from double to third times higher than that of password policies with the exception of Policy-3. It is not surprising since in PTP variants, users need to shuffle their passwords several times until they found a memorable password. Comparing between PTP variants, we can see that the participants in Insert-1, Replace-1 and Swap-2 can create password quicker than those in Insert-2 and Replace-2. This evidences that the passwords suggested by Insert-2 and Replace-2 are more difficult to be accepted by users. In terms of login time, it can be observed that participants in three PTP variants, Insert-1, Insert-2, and Replace-2 need longer time to login to the system compared to other methods. However, this overhead is not as high as the creating time.

Success Rates. The second measure to compare between PTP variants and password policies is the usability of the passwords. In order to make the experiments more realistic, a distraction technique is implemented. After users confirmed their password, for 45 seconds, users would count down in threes and a randomly chosen four-digit number will be displayed in the screen to refresh users memory. This technique is detained in the above section. Table III shows the total number of participants in each password method, the number of successful login and the percentage of trials wherein participants were able to successfully login to the system using the password they have created.

It can be seen from Table III that the successful login rate of all PTP variants except Insert-2 are mostly equal to password policies. For Insert-2, the successful rate is much lower at less than 60%. This shows users in Insert-2 suffered the difficult in remembering their passwords. This result is slightly different from the result in [10] where the authors confirmed high successful rate (up to more than 90%). The reason for this could be the difference between the participants in each experiments. In the previous research [10], half of the participants were non-Computer Science (CS) students and half were Computer Science (CS) students. In our experiment, all participant were Computer science students. The report in [10] also showed that computer science students often created more complex passwords that resulted in the higher rate of unsuccessful login.

TABLE IV. THE NUMBER OF PASSWORDS CRACKED BY HASHCAT.

Schemes	Total	Cracked	Percent
Policy-1	39	32	82.3
Policy-2	34	14	41.7
Policy-3	34	7	21.5
Insert-1	36	4	10.8
Insert-2	32	0	0.0
Replace-1	43	11	26.3
Replace-2	47	3	6.2
Swap-2	36	14	38.6

Security Strength. In order to evaluate the security strength of each password scheme, we used a well-known password cracker tool, Hashcat. Hashcat is a free tool to perform security audits on database password hashes or recover forgotten passwords. It is available for Linux and Windows. Unlike the better known command line and dictionary-based attack tool, John The Ripper, Hashcat supports both brute force attack and dictionary-based attacks. HashCat also comes with an interface (GUI, Graphical User Interface) making it easier for evaluators. After the users created their password, we converted these passwords to hash values using SHA-1 hash function. The list of the hash values of passwords is then inputted to Hashcat for cracking using the straight dictionary attack. The number of passwords cracked and the successful rate of cracking passwords are presented in Table IV.

It can be observed from Table IV that, all password policies are vulnerable to the dictionary attack. This is particularly serious with Policy-1 and Policy-2. With Policy-1 and Policy-2, up to more than 80% and 40% of passwords, respectively, can be cracked using Hashcat. This value for Policy-3 is reduced to about 20%. Therefore, forcing password policies on users might not be enough to protect their passwords being cracked. Although, tightening password policies like Policy-3 can significantly strengthen users passwords. The rate of passwords that are cracked is still high. Contrast to password policies, all PTP variants but Swap-2 and Replace-1 significantly increase password strength. It can be seen that less than 10% of passwords of Insert-1, Insert-2 and Replace-2 were cracked. Particularly, for Insert-2, none of the passwords were cracked. This showed that using PTP methods can significantly strengthen the passwords and potentially improve the safety of systems.

V. IMPROVING PERSUASIVE TEXT PASSWORDS

Section IV shows that the security strength of PTP password comes at the cost of its usability. Obviously, users involved in PPT variants suffered from difficulty in recalling their password. This is particularly serious in Insert-2 where nearly 40% of users failed to use their password to login to the system. In other PTP variants, the usability is roughly as good as Policy-3. However, their security strength is not as good as Insert-2. This section presents a method that improves the security strength of PTP variants while maintains their usability as easy as Policy-3.

TABLE V. NUMBER OF PARTICIPANTS IN EACH IMPROVED VERSION OF PTP

Schemes	Participants
New-Insert-1	45
New-Insert-2	31
New-Replace-1	31
New-Replace-2	36

The method to improve PTP was inspired from our observation that, in some PTP variants users still selected simple passwords. For example in Replace-1, when an user entered an initial password with eight digits, and the system replace one digit by another character, the user often chosen the password with all symbols are digits. For instance, 12345677 is one of the cracked password that is generated by Replace-1 after the user entered 12345678 as the initial password. In order to avoid this situation, we incorporated a policy into PTP variants so that all passwords generated by PTP methods are mixed with letters and digits. In other words, if users enter their password that is solely letters, the system will insert or replace some letters by digits. Conversely, if their initial password contains only digits, some letters will be added by the system. In case user's password has contained both letters and digits, some random characters will be inputted. However, the password is always checked for mixing condition (containing both letters and digits) before can be confirmed by users.

We conducted an extra experiment to examine the usability and the security strength of four new PTP variants. These variants are Insert-1, Insert-2, Replace-1 and Replace-2. The left PTP variant (Swap-2) was not tested since we could not force the mix character condition on this scheme. The new PTP variants are shorted with New prefix (New-Insert, New-Replace). The number of participants involved in each new PTP variant are presented in Table V. We also evaluated the efficiency of the PTP new variants using above three criteria. The results were compared with Policy-2, Policy-3, Insert-2 and Replace-2. They are detailed as follows.

Timings. Table VI shows the time participants took to complete the Confirm step and Login steps in each trial. It can be seen that the new variants of PTP does not incur the overhead on average time to create and use passwords. Apparently, the average time for confirming passwords and login to the system in PTP new variants is roughly equal to Policy-3. Particularly, the time for confirming and login of the new versions of PTP is even less than the original versions of PTP (Insert-2 and Replace-2). There are two possible explanations for this. First, the participants involved in two experiments were different. Second, forcing the mixture policy in PTP (all password must contain both letters and digits) helps users easier to recall their passwords. Future research will investigate these hypothesis.

Success Rates. The rate of successful login to the system when users used the password created by PTP new versions is presented in Table VII. It can be observed that users in PTP new variants could login successfully to the system as often as in Policy-3. Obviously, all new variants of PTP but New-Insert-2 achieved the rate of successful login at above 80%. These values are approximately equal to Polity-3 (the successful login rate of Policy-3 is 88.2%). Noticeably, although the participants in New-Insert-2 suffered some difficulty to remember

TABLE VI. SECONDS TAKEN PER TRIAL TO COMPLETE THE EXPERIMENT PHASES IN EACH IMPROVED VERSION OF PTP.

Schemes	Confirm	Login
Policy-2	87.8	19.7
Policy-3	157.2	32.8
Insert-2	209.2	56.3
Replace-2	239.5	54.5
New-Insert-1	91.8	27.1
New-Insert-2	152.6	29.5
New-Replace-1	151.8	32.6
New-Replace-2	176.3	33.8

TABLE VII. CONFIRM AND LOGIN SUCCESS RATES OF THE IMPROVED VERSIONS OF PTP.

Schemes	Total	Success	Percent
Policy-2	34	31	91.1
Policy-3	34	30	88.2
Insert-2	32	19	59.4
Replace-2	47	41	87.2
New-Insert-1	45	40	89.2
New-Insert-2	31	22	71.6
New-Replace-1	31	25	87.9
New-Replace-2	36	27	81.7

their password, the rate of successful login in New-Insert-2 is remarkably higher compared to the original version (Insert-2). This seems support to confirm that mixing characters in PTP makes the created password easier to recall.

Security Strength. The last measure to evaluate the efficiency of the new versions of PTP is the security strength of the created passwords. Similar to the above section, after the participants created their password, the passwords were saved to a file and their hash values were generated using SHA-1. These hash values were then inputted to Hashcat tool for cracking. The number of passwords were cracked in each method and their cracking rate is presented in Table VIII.

It can be seen from Table VIII that the new versions of PTP noticeably improve the security strength of the created passwords. We can see that all of the passwords created by three PTP new variants (New-Insert-2, New-Replace-1 and New-Replace-2) could not be cracked by Hashcat. Only New-Insert-1 did not help to improve the security strength of the created passwords. Future research will examine the reason why New-Insert-1 could not strengthen the created passwords. Overall, the results in this section evidence that the improvement of PTP helps to enhance the security strength of the generated passwords. Additionally, these passwords are also easier for users to create and recall.

TABLE VIII. THE NUMBER OF PASSWORDS CRACKED BY HASHCAT OF THE IMPROVED VERSIONS OF PTP.

Schemes	Total	Cracked	Percent
Policy-2	34	14	41.7
Policy-3	34	7	21.5
Insert-2	32	0	0.0
Replace-2	47	3	6.2
New-Insert-1	45	6	13.2
New-Insert-2	31	0	0.0
New-Replace-1	31	0	0.0
New-Replace-2	36	0	0.0

VI. Conclusions and Future Work

In this paper, the effectiveness of Persuasive Text Passwords (PTP) was examined and compared with common password policies. The experimental results showed that although password policies can increase the security strength of user's passwords, the rate of cracked password is still high. PTP was better than password policies in security strength. However, users in PTP were suffered from the difficulty in memorizing their passwords. After that, a method to improve PTP was proposed. This method was implemented by combining PTP with a password policy. The extra experiments showed that the improved versions of PTP are better than the original versions in both security and usability.

There are a number of research areas for future work which arise from this paper. First, we would like to conduct an analysis on the user's survey results to better understand the usability of PTP and the impact of this method on users attitude in creating and using passwords. Second, we would like to examine the weakness of the passwords that are created by password policies and the reason why these passwords are cracked? Next, we want to analyze the reason why the new versions of PTP help users recall their passwords better. Last but not least, an analysis on the password space using a technique in information theory (entropy) will shed some light on the security strength of the shed some light new variants. Future research will investigate this.

Acknowledgment

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2014.09.

References

[1] W. Stallings and L. Brown, *Computer Security: Principles and Practice*. Prentice Hall, 2007.

[2] Lin, Sun, and Hwang, "Attacks and solutions on strong-password authentication," *TIEICE: IEICE Transactions on Communications/Electronics/Information and Systems*, 2001.

[3] Chakrabarti and Singhal, "Password-based authentication: Preventing dictionary attacks," *COMPUTER: IEEE Computer*, vol. 40, 2007.

[4] R. Morris and K. Thompson, "Password security: A case history," *Communication, ACM*, vol. 22, no. 11, pp. 594–597, 1979.

[5] M. Weir, S. Aggarwal, M. Collins, and H. Stern, "Testing metrics for password creation policies by attacking large sets of revealed passwords," in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, ser. CCS '10. ACM, 2010, pp. 162–175.

[6] S. Chiasson, P. C. van Oorschot, and R. Biddle, "Graphical password authentication using cued click points," in *Computer Security - ESORICS 2007, 12th European Symposium On Research In Computer Security, Dresden, Germany, September 24-26, 2007, Proceedings*, ser. Lecture Notes in Computer Science, vol. 4734. Springer, 2007, pp. 359–374.

[7] R. Song, "Advanced smart card based password authentication protocol," *Computer Standards & Interfaces*, vol. 32, no. 5-6, pp. 321–325, 2010.

[8] R. Shay, S. Komanduri, P. G. Kelley, P. G. Leon, M. L. Mazurek, L. Bauer, N. Christin, and L. F. Cranor, "Encountering stronger password requirements: User attitudes and behaviors," in *Proceedings of the Sixth Symposium on Usable Privacy and Security*. ACM, 2010, pp. 2:1–2:20.

[9] G. V. Bard, "Spelling-error tolerant, order-independent pass-phrases via the damerau-levenshtein string-edit distance metric," *IACR Cryptology ePrint Archive*, vol. 2006, 2006.

[10] A. Forget, Sonisson, P. C. van Oorschot, and R. Biddle, "Improving text passwords through persuasion," in *Proceedings of the 4th Symposium on Usable Privacy and Security, SOUPS 2008, Pittsburgh, Pennsylvania, USA, July 23-25, 2008*. ACM, 2008, pp. 1–12.

[11] L. Peterson and M. J. Peterson, "Short-term retention of individual verbal items," *Journal of Experimental Psychology*, vol. 58(3), 1959.

[12] W. C. Summers and E. Bosworth, "Password policy: The good, the bad, and the ugly," in *Proceedings of the Winter International Synposium on Information and Communication Technologies*, ser. WISICT '04. Trinity College Dublin, 2004, pp. 1–6.