Improving Text Passwords Through Persuasion

Alain Forget[1,2], Sonia Chiasson[1,2], P.C. van Oorschot[1], Robert Biddle[2]
[1]School of Computer Science & [2]Human Oriented Technology Lab
Carleton University, Ottawa, Canada
{aforget, chiasson, paulv}@scs.carleton.ca, robert_biddle@carleton.ca

ABSTRACT

Password restriction policies and advice on creating secure passwords have limited effects on password strength. Influencing users to create more secure passwords remains an open problem. We have developed Persuasive Text Passwords (PTP), a text password creation system which leverages Persuasive Technology principles to influence users in creating more secure passwords without sacrificing usability. After users choose a password during creation, PTP improves its security by placing randomly-chosen characters at random positions into the password. Users may *shuffle* to be presented with randomly-chosen and positioned characters until they find a combination they feel is memorable. In this paper, we present an 83-participant user study testing four PTP variations. Our results show that the PTP variations significantly improved the security of users' passwords. We also found that those participants who had a high number of random characters placed into their passwords would deliberately choose weaker pre-improvement passwords to compensate for the memory load. As a consequence of this compensatory behaviour, there was a limit to the gain in password security achieved by PTP.

Categories and Subject Descriptors

K.6.5 [**Management of computing and information systems**]: Security and protection: Authentication

General Terms

Security, Human Factors, Experimentation

Keywords

Authentication, passwords, Persuasive Technology, usable security

Symposium on Usable Privacy and Security (SOUPS) 2008, July 23–25, 2008, Pittsburgh, PA USA
.

1. INTRODUCTION

Text password systems are as ubiquitous as users who create insecure passwords. Attempts at educating users on creating more secure passwords through advice and password policy enforcement have had little success. Users minimally meet password requirements and either ignore or misunderstand password creation advice [8]. There have been many proposals for improving password security, such as computer-generated passwords [17] and mnemonic phrase-based passwords [16, 29], but they are found lacking in either usability or security.

We have developed a lightweight password creation mechanism named Persuasive Text Passwords (PTP), a persuasive approach to influencing users to create more secure text passwords. Once users choose a password for creation, PTP improves the password's security by placing randomly-chosen characters at randomly-determined positions. Users may *shuffle* for an alternative improvement they may find more memorable. PTP offers a usable compromise between the memorability of user-chosen passwords and the security of randomly-generated passwords. We hope to improve the security of users' passwords while teaching and influencing users to create more secure passwords on their own, even when PTP is not present. We previously conducted an informal pre-test and a pilot study [10] to initially assess the potential of PTP. In this paper, we present a large-scale user study of more PTP variations. We use the term *persuasive* in reference to Fogg's work on Persuasive Technology [9], and because we suggest improvements to the user, but allow them to make the final decision.

The remainder of this paper is organised as follows. We begin with related work and relevant background in Section 2. The Persuasive Text Passwords system and its variants are described in Section 3. Section 4 presents the methodology of our user study, and Section 5 presents the results thereof. The implications and interpretation of the results are discussed in Section 6. We then address related issues and present areas for future work in Section 7. Finally, we offer some concluding remarks in Section 8.

2. BACKGROUND AND RELATED WORK

Attacks on text password authentication mechanisms are a threat not only to users' individual accounts, but to all accounts in the system [7]. For example, it is well known that automated password attacks on SSH servers are currently on-going [21, 25, 26]. For reasons possibly related to marketing and public relations, the extent and success of similar password attacks on businesses, banks, and so on,

are harder to find public information about, being less commonly reported if at all. However, we must assume such attacks are also on-going, since the potential gain for successful attackers is very high. Attackers trying to gain access to account resources typically only need to crack a small fraction of all passwords to be successful. Therefore, we believe that strengthening users' passwords remains a worthwhile pursuit.

Although security experts often blame users for lacking the motivation to behave securely and create secure passwords, the interviews and surveys of Adams and Sasse [1] suggested that users chose insecure passwords because they either were unable or did not know how to create secure passwords. To address this problem, researchers have proposed and tested various approaches. Yan et al. [29] suggested that mnemonic phrase-based passwords, memorable phrases condensed into passwords, were as secure as random passwords and more secure than regular passwords. Kuo et al. [16] later discovered that users based their passwords on phrases easily found on the Internet, and as such were no more secure than regular passwords when attacked with a mnemonic dictionary. Jermyn et al. [15] discussed a method of increasing password security by altering the order in which typed password characters are positioned. As an example, the authors explain that "sandwich" could be input as "snwchida". Password managers [13] [22] improve password security by generating and storing secure passwords for each user account while the user need only remember one password for all their accounts. However, they come with their own usability challenges [5]. Other proposed schemes for creating more secure passwords include system-generated passwords [17], fictional news headlines [14], and word associations [19], but such schemes have not yet proven to be sufficiently workable.

Password character restrictions, strength meters, and other strategies to enforce the creation of more secure passwords are common. Furnell [12] reported the password restrictions and advice provided by ten popular Internet sites lacked both consistency and effectiveness, making it very difficult for users to form accurate mental models of secure passwords. Vu et al. [27] ran four user studies on various combinations of password restrictions, testing both short-term and long-term memorability, finding that password restrictions did not necessarily lead to more secure passwords. Florencio and Herley [8] discovered that the majority of 500,000 users' passwords across many popular websites (including PayPal) consisted of only lowercase characters.

2.1 Evaluating Password Strength

One challenge of evaluating the security of text passwords lies in defining an accurate model with which to compare the security of two passwords or password schemes. St. Clair et al. [24] proposed a password cracking forecasting model which accounts for password space, parallelism, and processor speed to calculate the number of operations required to crack a password. They expand their model to account for predictions in yearly processor speed increases to predict how soon random 8-character passwords will be trivially crackable. They do not account for user biases in selecting language-derived passwords. They argue that password restriction policies weaken passwords by limiting the total password space and reducing the number of operations required to crack random passwords. Burr et al. [3] disagree, stating that simple password restriction policies can eliminate the most obvious password choices. They believe the resulting increase in the practical security of the system outweighs the reduction in the total password space. Based on Shannon's [23] model of entropy for encoding language into bits, Burr et al. discuss the security of a given text password system where *bits of entropy* correspond to the degree of a password value's uncertainty. They focus their discussion to user-chosen passwords using Shannon's estimate of entropy in English text as a baseline for defining their heuristic *bits of guessing entropy* model. The model gives a very rough approximation of a password's security primarily based on its length and the constraints imposed on its creation. Thus, the model does not account for the actual characters in individual passwords, as it assumes that user-chosen passwords in general are quite similar to English text.

2.2 John the Ripper

Security practitioners use *password cracking* tools to evaluate the security of passwords created with a given system. Password cracking is typically understood as an attempt to systematically guess as many account passwords as possible for some system. John the Ripper [6] (JtR) is a popular open-source password cracking tool. JtR has three different attack modes. *Single crack mode* uses available data such as login names, full names, and home directory names as candidate passwords and employs a rich set of customisable word mangling rules (such as adding a digit at either end or replacing "a" with "@"). *Wordlist mode* guesses the character strings provided in a user-specified dictionary as candidate passwords, applying word mangling rules as an option. *Incremental mode* attempts to guess passwords in a brute force manner optimised for the user-chosen character set and length as well as using trigraph frequencies to quickly crack as many passwords as possible. Kuo et al. [16] cracked 11% of 146 survey-collected regular passwords using Wordlist mode with word mangling rules, and an additional 8% using Incremental mode for 62 hours. The number, type, and clock speed of the machines used in the 62-hour attack were not disclosed. St. Clair et al. [24] cracked 25% of 3500 passwords from their Computer Science and Engineering department in 2 hours with a cluster of 16 computers using Incremental mode and 4 using Wordlist mode. All 20 computers were reported to be AMD Opteron 250 processors. Using the Wordlist mode for 22 minutes and Incremental mode for 24 hours, Proctor et al. [20] used a 400-MHz Pentium II computer to crack 34% of 96 passwords from two experiments on password restrictions.

2.3 Persuasive Technology

Persuasive Technology [9] (PT) is the emerging field of "interactive computing systems designed to change people's attitudes and behaviours". PT is founded on well-established theories from behavioural, personality, and social psychology. PT is a set of tools, cues, and media that technology can implement to influence users to behave in some desired manner. PT has successfully influenced people to engage in various desired behaviours within several domains, particularly health and education. The Persuasive Authentication Framework [11] has been proposed as a means of leveraging PT to address the unique challenges of usable authentication and security [28].

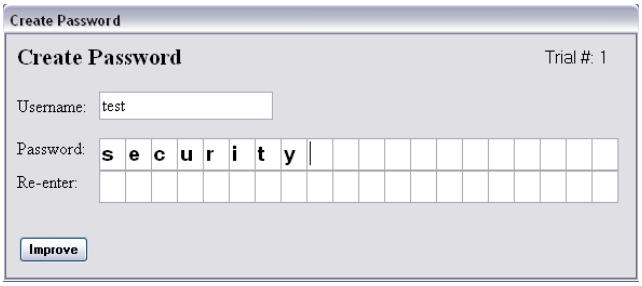

Figure 1: PTP password creation before applying the persuasive improvement.

Figure 2: PTP password creation after applying the Insert-2 persuasive improvement.

2.4 Persuasive Cued Click-Points

Chiasson et al. [4] applied PT to click-based graphical passwords, which are authentication schemes wherein a password consists of user-chosen click-points on an image or a series of images. Persuasive Cued Click-Points significantly decreases the likelihood users will choose click-points on hotspots, which are particular areas on an image wherein many users would otherwise choose their click-points. This is accomplished by requiring users to choose their passwords' click-points inside a randomly-positioned *viewport*. Users may *shuffle* the viewport, whereupon it repositions itself at another random location on the image. The persuasive elements had the positive effect of assisting users to choose more random graphical passwords while still maintaining usability. In the present paper, we apply similar persuasive principles to text passwords.

3. PERSUASIVE TEXT PASSWORDS (PTP)

Persuasive Text Passwords (PTP) is a user-chosen text password system which guides users to make their passwords more secure. During password creation, PTP improves password security by placing a few randomly-selected characters at randomly-determined positions in the users' initial password (see Figures 1 and 2). Users have the option to *shuffle* for an alternative set of random characters and random positions until they deem one is sufficiently memorable. The random characters are chosen with uniform probability across all characters available on English US keyboards, except for blank spaces. PTP offers a middle-ground between the usability of purely user-chosen passwords and the security of system-assigned passwords. We hypothesise that the PTP strategy of adding random elements to user-chosen passwords will increase their security while maintaining sufficient memorability. Furthermore, we believe the visibility and user involvement in PTP's strategy may teach users how to improve the security of their passwords for systems that do not implement PTP.

3.1 PTP Variations

We devised a number of PTP variants in order to explore this approach to improving password security. Below, we outline several variations which we have implemented and studied. Many others exist that we have yet to examine.

Preload. Users are given the system-assigned characters before creating their password. The characters are positioned randomly within the first eight character slots. Users create their password around the system-placed characters.

Replace. Users first choose an initial password as they would for a typical password system. The system replaces characters in the users' passwords at random positions with randomly-chosen characters.

Insert. After users select an initial password as usual, the system inserts randomly-selected characters at random positions between user-chosen password characters, lengthening the password. See Figure 2 for an example of Insert with two system-selected characters.

We ran informal user pre-tests with these three variants, as well as a pilot user study with Insert and Replace [10] only. The pre-tests revealed problems which we subsequently corrected for the pilot. One such problem arose in the Preload variant, where users would simply repeat the system-assigned characters. For example, if presented with "_ _ B _ _ # _ 8", users would create a password such as "BBB###88". Despite the use of uncommon symbols, such repeated-character passwords are generally considered insecure. Thus, we did not test the Preload variant in further studies.

Another problem we found during pre-testing involved the number of characters selected. All PTP variant systems would either insert or replace a randomly-determined number of characters between two and four. Pre-test users would usually shuffle when more than two system-assigned characters were presented. We were unable to ascertain the memorability of pre-test passwords containing more than two system-chosen characters because too few had been created. Therefore, the PTP variants tested in the pilot study always placed a fixed number of characters into users' passwords. See Forget et al. [10] for details on other problems regarding character memorability and identifiability, which were also noted during the pre-testing and were corrected before running the pilot study.

A 15-participant pilot study [10] explored the Replace-2 and Insert-2 PTP variants, wherein two characters would either replace two existing characters in the user's password or be inserted in between. Half of our participants were Computer Science (CS) students, and they usually created more secure initial passwords than participants studying other disciplines (non-CS). However, CS students had more difficulty with PTP than non-CS students, since the memorability of their PTP-improved passwords declined as they chose more secure pre-improvement passwords. We had expected non-CS users to have some difficulty with PTP, but their success rates were reasonably high.

These studies left some questions unanswered. For the full study presented in this paper, we pursue three main lines of

inquiry:

1. How does PTP affect password security?
2. How does PTP affect users as the memory load is increased?
3. How does PTP affect users' understanding of how to create secure passwords?

4. METHODOLOGY

Our full in-lab user study design was reviewed and approved by our University's Ethics Committee for Psychological Research. A between-subjects design was used, wherein each participant was assigned to one of the experimental conditions shown in Table 1. Each participant in a given condition used the PTP variant by the same name. The number after the hyphen in the condition name represents the number of system-assigned characters that were placed into users' passwords. For example, participants in the Insert-4 condition used a PTP Insert variation which inserted 4 characters into their passwords. To accurately compare the PTP variations to regular passwords, the system did not modify participants' passwords in the Control condition.

Condition	Males	Females	Total
Control	9	10	19
Replace-2	7	9	16
Insert-2	7	9	16
Insert-3	7	9	16
Insert-4	7	9	16
Total	37	46	83

Table 1: Number of males, females, and total participants in each condition testing a PTP variant.

Another difference between conditions was the minimum length requirements of users' pre-improvement *initial passwords*. Participants in the Control and Replace-2 conditions were required to enter a minimum 8-character initial password while Insert-2 participants had to choose a minimum 6-character initial password, which would become an 8-character *improved password* (post-improvement password). This facilitates comparisons between each of these conditions, since all their improved passwords are at least 8 characters long. We likewise considered setting the minimum initial password length to 5 and 4 characters for the Insert-3 and Insert-4 conditions. However, we felt it would be easier to compare the three Insert conditions if they all required at least six characters, as well as better emulating contemporary password policies.

Other than the differences mentioned above, the experiment was carried out in the exact same manner for all conditions. Results from our pilot study suggested that Replace-2 was more difficult for users than Insert-2. Also, adding characters increases password security more than replacing characters. Since the Insert variant seemed superior to Replace in terms of both memorability and security, we chose to test users' password memory when more characters are inserted, but not replaced. Finally, since both studies were executed following identical procedures, the results presented in this paper include data from the 7 Replace-2 participants and 8 Insert-2 participants who participated in the pilot study.

Our participants consisted of 83 university students studying across various disciplines. They were all familiar with using computers, the Internet, and passwords. Each participant completed at least ten trials, for a total of 834 trials over all participants. For each trial, users completed a process consisting of creating, confirming and logging in with a password. Before and after the experiment, users respectively filled out a demographics questionnaire and a user-opinion questionnaire.

All times and user actions were logged by the system, including users' pre-improvement initial password as well as their improved password. The experimenter took note of all participant behaviour and comments throughout the session. Before beginning the experiment, participants were asked to pretend the passwords they create during the session were going to protect their online bank accounts, and they should create passwords that would be easy to remember but hard for other people to guess. Regarding PTP's password improvement, they were told that although it would help them create more secure passwords, they could shuffle as often as they wanted to find a character arrangement they felt was memorable. Participants familiarised themselves with creating passwords in this new way by performing a practice trial. Practice trials are not included in the 834-trial data. The following five steps made up a trial.

Create. Users would first enter a password of their choice. The system then placed the number of characters appropriate for the condition's PTP variant into the users' password (see Section 3.1). No characters were placed into Control participants' passwords. Users were allowed to *shuffle* the characters as much as they liked. Users could press the *Reset* button if they wished to change their initial password. Once they found a password and system-assigned characters they liked and felt they could remember, they then re-typed the improved password on the second row and pressed the *Create* button. As shown in Figure 2, the participants' passwords were visible during password creation. We learned from the pre-tests that users found it beneficial to see the system-assigned characters and re-type their improved password, ensuring they could correctly identify and type the extra characters.

Confirm. To confirm their password, users re-entered their improved password, which was masked with asterisks. If they made a mistake but felt they knew their password, they could retry to confirm. However, if they thought they forgot their password, they could end the trial.

Questions. Users were asked two questions measuring how easy they felt it was to create their password and how difficult they thought it would be to remember in one week. Participants answered the questions on a 10-point Likert scale, from very easy to very difficult.

Distraction. For 45 seconds, users would count down in threes from a randomly chosen four-digit number. This type of distraction flushes their textual working memory [18] and simulates a longer passage of time by focusing participants' attention on a separate cognitively-difficult task.

Login. Participants attempted to login with their improved password, which was echoed with asterisks. If they made a mistake, they could try to login again, or end the trial if they forgot their password.

	Success %		Significant differences versus Control	
Condition	Confirm	Login	Confirm	Login
Control	99.5	98.4	-	-
Replace-2	91.9	92.6	$\chi^2(1, N = 351) = 11.07, p < .001$	$\chi^2(1, N = 351) = 4.09, p < .05$
Insert-2	93.9	99.3	$\chi^2(1, N = 353) = 7.38, p < .01$	Not significant
Insert-3	81.9	97.7	$\chi^2(1, N = 350) = 32.12, p < .001$	Not significant
Insert-4	92.5	93.9	$\chi^2(1, N = 350) = 9.94, p < .05$	Not significant

Table 2: Confirm and login success rates (shown as percentages) and significant differences versus Control across PTP conditions.

	Mean			Median			Standard Deviation		
Condition	Create	Confirm	Login	Create	Confirm	Login	Create	Confirm	Login
Control	33.2	8.7	11.4	27.9	7.3	7.8	20.5	5.6	12.0
Replace-2	67.0	14.5	16.3	56.2	9.8	11.7	37.2	15.7	15.5
Insert-2	65.0	20.0	17.1	55.3	11.8	12.9	34.0	25.5	16.4
Insert-3	65.0	21.6	20.8	57.9	12.7	13.3	26.7	22.2	35.4
Insert-4	98.3	25.1	28.4	81.5	16.4	16.7	66.1	23.3	44.4

Table 3: Seconds taken per trial to complete the experiment phases across PTP conditions.

Condition	Mean	Median	StD
Control	-	-	-
Replace-2	5.7	3.0	9.3
Insert-2	8.4	2.0	19.3
Insert-3	10.2	6.0	10.0
Insert-4	18.0	6.5	51.8

Table 4: Mean, median, and standard deviation of number of shuffles per trial. *StD* is short form for *standard deviation*.

5. RESULTS

We now present the results and a brief interpretation.

Success Rates. Table 2 shows the percentage of trials wherein participants were able to successfully confirm and login. The success rate differences are significant between the Control group and the PTP variants for confirm but not for login. This suggests that, although PTP users had more difficulty confirming their password than Control participants, once they did so successfully, they could recall it as easily as Control participants. We suspect the difference between the Replace-2 and Control login success rates would not be present with a larger sample size. We will later discuss reasons for the anomalous result of the Insert-4 confirm success rates being higher than those for Insert-3, despite the former being the more difficult condition.

Timings. Table 3 shows the time participants took to complete each step of a trial. Participants in the Control group took a mean of 33 seconds to create a password. By shuffling for acceptable system-chosen characters, all PTP users took approximately 65 seconds on average to create their passwords, twice as long as the Control group. The notable exceptions are Insert-4 users who took about 98 seconds, over three times as long as Control participants. We noted that Insert-4 participants would stare at their password at length after they had finished shuffling, in an attempt to memorise it. We believe that 65 seconds is an acceptable creation time for long-term passwords. Furthermore, as we discuss later in the *User Perception* section, the passage of time is not noticeable when creating a password, presumably because users are cognitively active when choosing a password and shuffling.

Shuffles. Table 4 shows the amount of shuffling done by participants per trial in each condition. Participants in more difficult conditions shuffled more. Regarding Insert-4, the high standard deviation and low median relative to the mean shows that the mean is being inflated by a small number of trials where participants shuffled an exceptional number of times.

Errors. Table 5 demonstrates the confirm and login errors committed by participants per trial. The medians of 0 errors show that participants successfully confirmed and logged in on their first attempt for the majority of trials. Therefore, when users did commit an error, they were likely to retry to confirm or login multiple times before either succeeding or giving up. One-way ANOVA tests revealed no significant differences amongst the number of login errors across conditions ($F(4, 829) = 1.88$, $p = .112$), but did show significant differences ($F(4, 829) = 8.01$, $p < .001$) for confirm errors. We saw a spike in confirm errors with the Insert-3 group, although users of all PTP variants committed significantly more errors than the Control group. Users sometimes had trouble confirming, but once they did so successfully, they typically were able to recall their password to login. These findings support our conclusions regarding the success rates.

Password Space. The set of all characters is often split into four classes: lowercase, uppercase, numeric, and special characters (or symbols). Previous research has shown that many users choose passwords from a single class, usually lowercase letters [8]. A password containing characters from only one or two classes will be easier for attackers to guess than a password spanning more classes. These character classes can be combined to form distinct *password spaces*. To illustrate, the password space of all lowercase and numeric passwords contains all passwords that have at least one lowercase and one numeric character, but no uppercase or special characters. Given all possible combinations of the

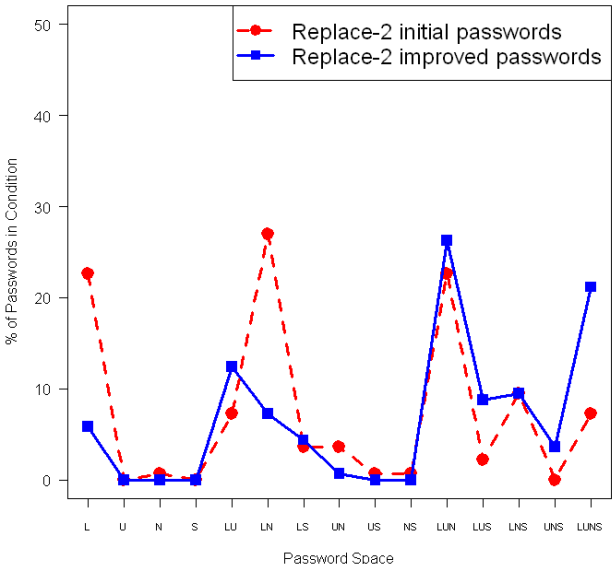

Figure 3: Percentage of Replace-2 initial and improved passwords that fall into various password spaces. The lines are present for ease of interpretation and do not represent continuous values.

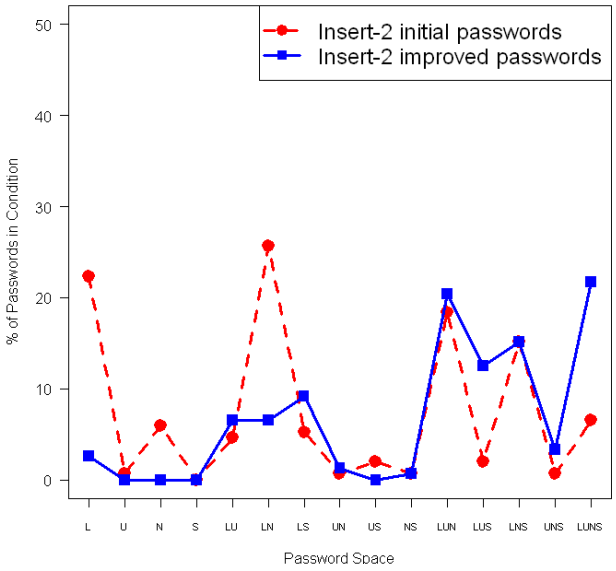

Figure 4: Percentage of Insert-2 initial and improved passwords that fall into various password spaces. The lines are present for ease of interpretation and do not represent continuous values.

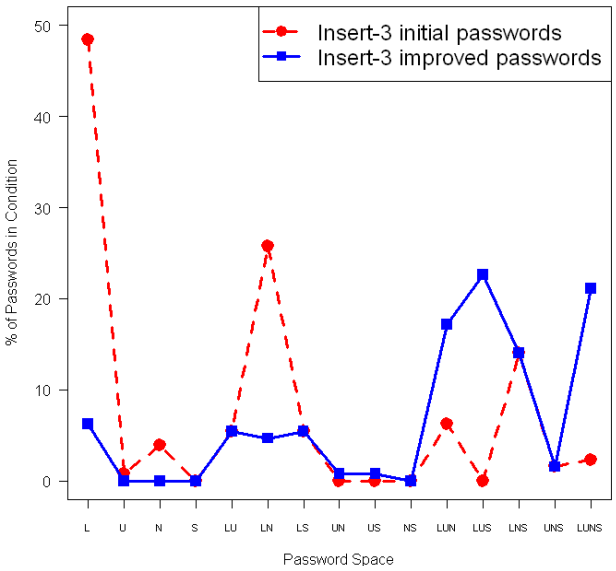

Figure 5: Percentage of Insert-3 initial and improved passwords that fall into various password spaces. The lines are present for ease of interpretation and do not represent continuous values.

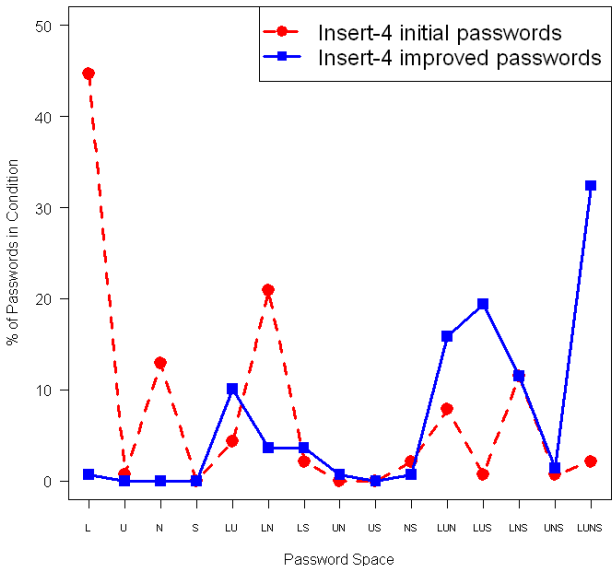

Figure 6: Percentage of Insert-4 initial and improved passwords that fall into various password spaces. The lines are present for ease of interpretation and do not represent continuous values.

	Mean		Median		Standard Deviation	
Condition	Confirm	Login	Confirm	Login	Confirm	Login
Control	0.1	0.1	0	0	0.4	0.6
Replace-2	0.3	0.3	0	0	0.7	0.9
Insert-2	0.4	0.1	0	0	1.0	0.6
Insert-3	0.6	0.3	0	0	1.2	1.7
Insert-4	0.4	0.3	0	0	0.9	1.1

Table 5: Number of errors made per trial during confirm and login across PTP conditions.

	Mean			Median			Standard Deviation		
Condition	Initial	Improved	Delta	Initial	Improved	Delta	Initial	Improved	Delta
Control	51.6	-	-	47.6	-	-	13.7	-	-
Replace-2	51.5	56.7	5.2	48.9	53.6	4.9	10.8	11.4	5.8
Insert-2	49.3	67.8	18.5	46.0	61.1	17.8	20.5	23.4	7.2
Insert-3	42.1	68.1	26.1	36.7	64.1	26.1	14.6	15.9	6.1
Insert-4	35.5	69.3	33.8	31.0	65.6	32.9	12.6	13.2	5.5

Table 6: Initial, improved, and delta (the difference between improved and original) estimated bits of security H for the five conditions.

four classes mentioned above, there are a total of 15 distinct password spaces.

Although PTP places randomly-selected characters into users' passwords, users may shuffle to obtain new characters. Thus, a user with an all lowercase password may shuffle until the system randomly selects only lowercase letters. Such a password would be vulnerable to a brute force attack on all-lowercase passwords. To evaluate the effectiveness of PTP in defending against such attacks, we examined the types of characters found in users' initial and improved passwords.

The password spaces of initial and improved passwords of each condition are shown in Figures 3, 4, 5, and 6. Each graph shows the percentage of passwords in each password space for each PTP variant condition. The x-axis represents all character class combinations of lowercase letters (L), uppercase letters (U), numeric characters (N), and symbols (S). The y-axis represents the percentage of passwords from the given condition which fall into the password space on the x-axis. Although all the points corresponding to the same password type (i.e. initial or improved) are connected by a line, this is only for ease of comparison and does not represent a continuum.

Figures 3, 4, 5, and 6 show that PTP users predominantly chose initial passwords containing either only lowercase (L), lowercase and numeric (LN), or lowercase, uppercase, and numeric characters (LUN). Insert-3 and Insert-4 participants more often created initial lowercase-only passwords. However, for all conditions, we see that there are very few PTP-improved passwords containing only lowercase or only lowercase and numeric characters. This shows that users were usually influenced to create passwords containing characters from more than one class, since PTP's random character selection algorithm was unbiased and would allow users to create passwords consisting of characters from a single class. This shows that accounts protected by PTP passwords would most likely not be compromised in an obvious and theoretically efficient attack by guessing all passwords containing only lowercase and/or numeric characters.

Furthermore, in all four figures, we can see a significant increase between the number of initial and improved passwords in the larger spaces of lowercase, uppercase, and special (LUS) ($\chi^2(1,\ N = 556) = 72.52$, $p < .001$), and lowercase, uppercase, numeric, and special ($LUNS$) characters ($\chi^2(1,\ N = 556) = 83.58$, $p < .001$). Amongst other password space increases, Figures 5 and 6 also show significant increases in lowercase, uppercase, and numeric (LUN) Insert-3 ($\chi^2(1,\ N = 128) = 20.70$, $p < .001$) and Insert-4 ($\chi^2(1,\ N = 139) = 16.92$, $p < .001$) passwords. Thus, all three Insert variants improved the security of passwords by adding characters that promoted the users' passwords into larger password spaces.

Estimate of Bits of Security. As a rough estimate of *bits of security*, we evaluate password security strength using the standard formula for per-character entropy, multiplied by the number of characters:

$$H = l \cdot log_2(b),$$

where l is the password length and b is the size of the alphabet from which the password's characters were chosen. This crude metric, while useful as a starting point for relative comparison, over-estimates password security. It assumes that individual password characters are chosen randomly and independently from one another. Thus, the metric does not account for user-biases towards English words, predictable character positioning, or relationships between adjacent characters. For example, for a password containing 5 lowercase letters (each with 26 possible choices) and 1 uppercase letter (with 26 other possible choices), the metric assigns the password a measure of $6 \cdot log_2(52)$ bits. Nonetheless, we use this model for its simplicity to provide a preliminary security evaluation.

The initial and improved passwords' estimated bits of security H for the five conditions are shown in Table 6. The *Delta* columns show the increase in H as a result of PTP's improvement. Although Insert initial passwords had to be at least six characters, Insert-2 users often chose longer initial passwords. This is why the estimated security bit values for the Insert-2 initial passwords is not much lower than for the Control initial passwords, which had to be at least 8 characters long.

		All+Rules		Mangled	
Condition	Total # of Passwords	Cracked	Percent	Cracked	Percent
All	1668	202	12.1	251	15.0
Control	190	18	9.5	36	18.9
Replace-2	161	24	14.9	21	13.0
Insert-2	163	23	14.1	28	17.2
Insert-3	160	65	40.6	75	46.9
Insert-4	160	54	33.8	55	34.4

Table 7: Initial passwords cracked by various John the Ripper dictionary attacks. None of the attacks were able to guess any password improved by any PTP variation (see discussion in text).

All three Insert conditions resulted in improved passwords having significantly more estimated bits of security than the Control and Replace-2 conditions ($t(831.87) = 13.50$, $p < .001$). However, the three Insert variations produced improved passwords of similar estimated security ($F(2, 480) = 0.32$, $p = 0.727$). In fact, participants who had more characters inserted into their password chose to create initial passwords with noticeably fewer estimated bits of security ($F(2, 480) = 28.94$, $p < .001$). Insert-4 participants in particular created initial passwords with significantly lower estimated security as they completed more trials ($F(1, 158) = 8.17$, $p < .005$). Reasons for this phenomenon are further discussed in Section 6. Similar but weaker correlations between estimated initial password strength and completed trials were found for Insert-2 and Insert-3, but not for Control and Replace-2.

John The Ripper (JtR). We ran various John the Ripper [6] dictionary attacks on the initial and improved passwords. We used the Bartavelle-patched version of JtR [2], to work with SHA-1 encrypted passwords. We ran two sets of dictionary attacks on the passwords from all conditions together, as well as each individual condition. Our first dictionary attack, *All+Rules*, used the largest free word list available on the JtR website (*"All.lst"*), containing about 4 million entries. We also enabled JtR's built-in word mangling rules, which guesses predictable variations of the words in the provided list. Our second dictionary attack, *Mangled*, used the *mangled.lst* word list purchased from the JtR website. Containing over 40 million entries, this list contains all the words in *all.lst*, in addition to pre-mangled variations and various extra Latin alphabet-based dictionary words.

We ran two rounds of attacks with each dictionary. We first attacked passwords from each condition separately. We then attacked the entire set of collected passwords. Table 7 displays the number of total, cracked, and percentage of cracked initial passwords for the aforementioned attacks. The Mangled attack cracked slightly more initial passwords than the All+Rules attack in every condition except Replace-2.

None of the dictionary attacks were able to crack a single password improved by any PTP variation. This is likely because the JtR dictionaries do not include words formed by randomly inserting as few as two characters in common words. Of course, considerably larger dictionaries could be built to specifically crack such passwords.

User Perception. Participants rated statements on our post-test questionnaire on a Likert scale of 1 (strongly disagree) to 10 (strongly agree). To cross-validate the users' responses, the questionnaire was composed of four statements for each general topic of user perception we measured. To control for bias from the statements' structures, each of the four statements for each topic were constructed to be either posed in a normal or reversed manner, and to be either comparing PTP to normal passwords or have no comparison. The statements of all topics were randomly ordered on the questionnaire.

Figures 7, 8, 9, and 10 are notched box plots describing the distribution of participants' Likert score responses to four aggregate questions measuring users' perception regarding ease of password creation, improved password guessability, how much PTP helped them create more secure passwords, and login speed. The bold line down the middle of each box is the median, and the left and right halves of boxes denote the 2^{nd} and 3^{rd} quartiles. The length of this box is known as the *interquartile range* or IQR. The whiskers at either end of the plot denote the 1^{st} and 4^{th} quartiles. Any circles beyond the whiskers are outliers which are at least $1.5 \cdot IQR$ away from the median. Whenever the notches, which are the angled sections of the box plots, for two groups overlap relative to the x-axis, there is no statistically significant differences between the two groups. Since the notches overlap, Figure 7 shows that PTP users and Control participants felt they had the same degree of difficulty creating passwords. The PTP variant notches in Figures 8 and 9 do not overlap with the Control group notches, showing that the participants felt their PTP improved passwords were significantly less likely to be guessed by an attacker, and that PTP helped them create significantly more secure passwords than participants in the Control group. Finally, the overlapping box plot notches in Figure 10 show that PTP users in most conditions, including Insert-4, did not feel that the time to log in was any longer than those in the Control group. Oddly, Insert-2 users perceived that the login time was *faster* than Control group participants, since the two box plot notches do not overlap. These results suggest that the longer login times for PTP did not have a negative impact on users' perception of the system.

6. INTERPRETATION

We now interpret the results by addressing questions regarding the various effects PTP had on participants.

How does PTP affect password security? In all conditions, PTP improved the security of users' passwords according to our metric. The characters in users' initial passwords were mostly lowercase characters with the occasional numeric or uppercase character. After applying the

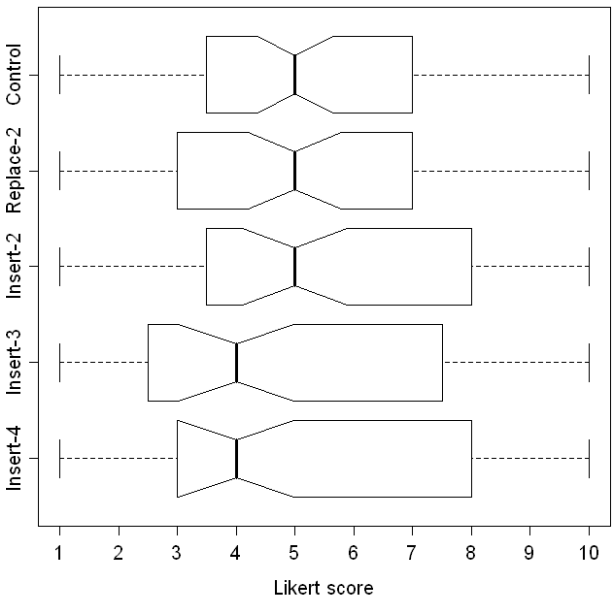

Figure 7: Participants' perception of ease of password creation (1 is very difficult, 10 is very easy).

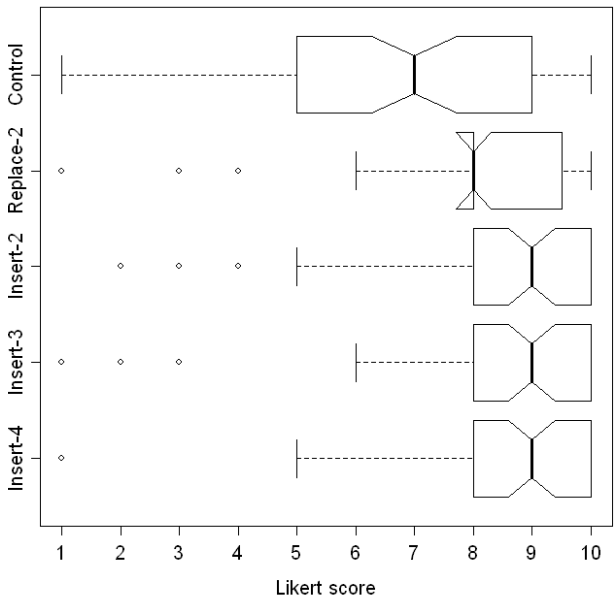

Figure 8: Participants' perception of PTP password guessability (1 is very guessable, 10 is very difficult to guess). The Replace-2 box plot appears different because there were no participant responses in the 2^{nd} quartile. The notch is drawn in the conventional manner, indicating the significant difference with the Control group, due to the lack of overlap between the two groups' notches.

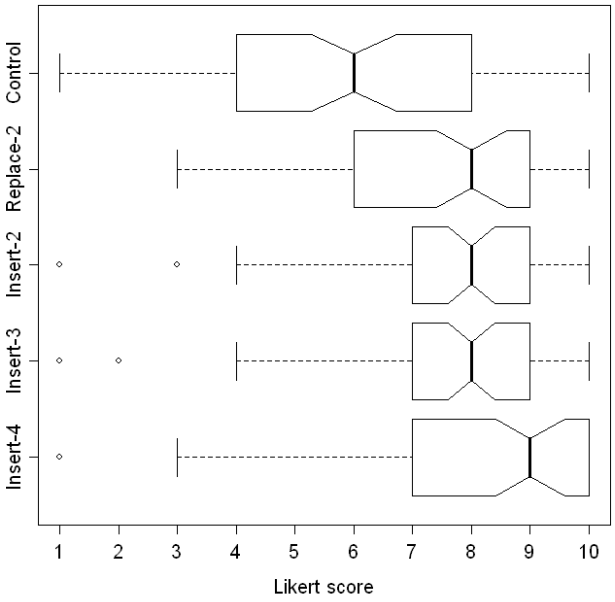

Figure 9: Participants' perception on PTP's assistance in creating more secure passwords (1 is not helpful, 10 is very helpful).

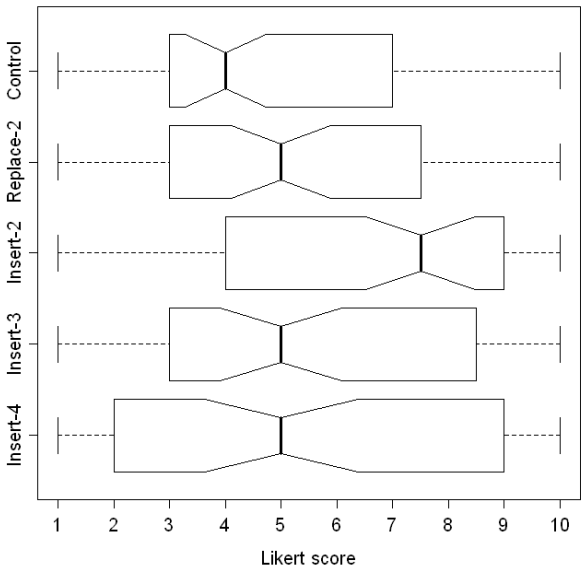

Figure 10: Participants' perception of login speed with PTP passwords (1 is very slow, 10 is very fast).

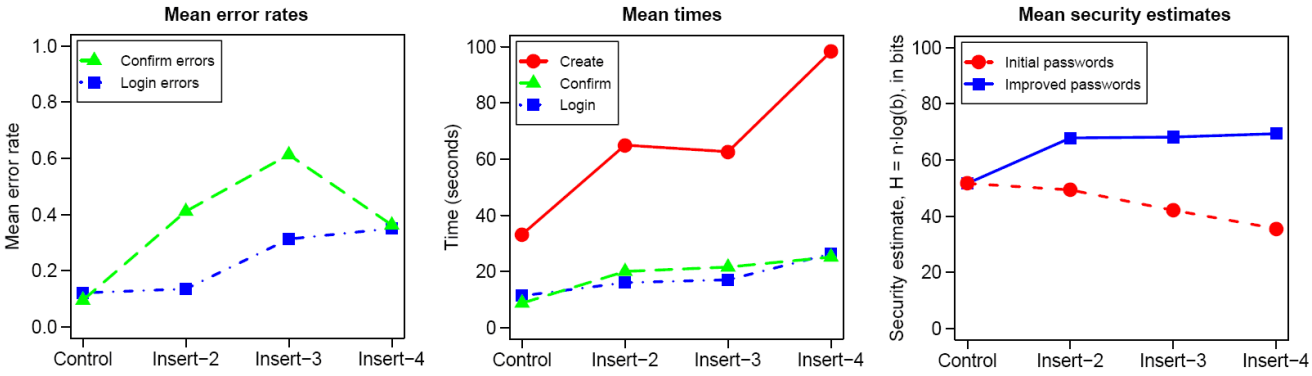

Figure 11: Summary of mean times, error rates, and a crude relative security estimate across conditions. The lines are present for ease of interpretation and do not represent continuous values.

improvement, the majority of users' passwords included at least three classes of characters. All three Insert conditions resulted in improved passwords of similar estimated security strength, despite some PTP variants adding more characters than others.

How does PTP affect users as the memory load is increased? The mean number of errors in all conditions show users sometimes had difficulty confirming their password. However, the medians of 0 show that the majority of those errors occurred on a small number of trials. This means that participants were usually able to successfully confirm without error. Furthermore, participants seldom had trouble logging in, suggesting that users were able to remember their passwords. These memorability results should be confirmed with a field study as well as testing for long-term password memory and interference from multiple passwords.

In our most difficult condition of Insert-4, the median login time is 16.7 seconds. While this is longer than the Control group (7.8 seconds), we believe it is still acceptable, especially considering the increase in password security. Furthermore, all PTP variant users, including Insert-4 users, reported perceiving the time to login as equivalent or faster than Control group participants reported. Most users also believed that they would be able to login more quickly with practice, and no one mentioned the time to login during our observations.

Figure 11 contains three plots showing various phenomena across the Insert conditions to directly compare the effect of inserting more characters. The graph on the left shows the mean confirm and login errors committed by participants per trial, as previously shown in Table 5. Although we expected the number of errors to increase for more difficult conditions, we see a significant drop in the number of confirm errors per trial between Insert-3 (approx. 0.6 errors) and Insert-4 (approx. 0.35 errors). The two reasons for this lie in the other two graphs. The second graph shows the mean times to create, confirm, and login. Notice the steep increase in creation time between Insert-3 (approx. 60 seconds) and Insert-4 (approx. 100 seconds), indicating that users in this condition needed much more time to memorise their password. Observations during the session support this, as Insert-4 users would gaze at their password for long periods of time before completing the password creation. We believe that this intense effort taken by Insert-4 users to memorise their passwords during creation is the first reason they made significantly less confirm errors. The second reason lies in the graph on the right, which shows the mean estimated bits of security (using the model from Section 5) for the initial and improved passwords. This shows that participants created less secure initial passwords as more characters were inserted, especially in Insert-4 (under 40 bits, according to our metric). Among the more interesting results to us, the strength of improved passwords throughout all three Insert conditions is relatively constant (approx. 68 bits, according to our metric). Therefore, the additional burden of extra inserted characters did not result in greater password security, because users increasingly compensated for the cognitive burden of the additional random character insertions by creating weaker initial passwords.

How does PTP affect users' understanding of how to create secure passwords? Results from our questionnaire indicated that users in all PTP conditions felt that the system was helping them create more secure passwords. Furthermore, experimenter observations and other post-test questionnaire responses suggest that most users learnt new ways of making their passwords more secure. Many of them also mentioned that they would try to employ the random character insertion strategy to their own passwords for systems where PTP is not implemented. Many participants had never previously considered using symbols in their passwords, and some participants seemed previously unaware such symbols were available on their keyboard.

As previously discussed, users will create less secure passwords if too much of a burden is placed upon them. This suggests a possible danger: users may become dependent on PTP create even less secure passwords than usual with non-PTP systems. We must be careful that users do not rely entirely on PTP to make their passwords more secure.

7. DISCUSSION

We now consider the implications of our study and the overall effect of PTP. Since PTP is a password creation scheme, we address only the security of the resulting passwords and the usability of the scheme. We do not address the security of any of the systems between and including the client and the server (such as the effects of malware), nor social engineering attacks (such as phishing).

A benefit of PTP is its lightweight design. Implementation of PTP should only require minor changes to the client-side password creation module, and no modifications are necessary on the server side. PTP requires changes made to the password creation process, but none to the login process. PTP is also highly resistant to password re-use, which may be viewed as both a security advantage and usability disadvantage. Should users choose the same initial password for multiple passwords, it is highly improbable that PTP will place the same characters in the same positions in two identical initial passwords. Unfortunately, this may increase memory interference effects from multiple passwords.

A drawback of PTP is password visibility during creation. This makes PTP suitable only when users will be creating passwords in environments free of shoulder-surfing. On the other hand, since passwords are never displayed after their initial creation, PTP is no more vulnerable to shoulder-surfing than regular passwords during confirmation or login.

We expect that the amount of user-choice in PTP should contribute to password memorability. While completely random passwords would be more secure than those created with PTP, we believe that PTP helps users create more memorable passwords, though this has yet to be tested. Of course, knowing that a system uses PTP and knowing how PTP works will allow attackers to refine their cracking strategies. For example, a potential modification to John the Ripper would be to search for all words in a list, with two random characters inserted in all possible places. We believe that the general idea of PTP would be most effective, not as a single scheme for password improvement, but rather in allowing many different variations. This would limit the number of assumptions attackers can make about the authentication system, potentially increasing the work involved in guessing.

The main idea of PTP is a middle-ground approach between system-generated and user-chosen passwords. The addition of random elements to users' passwords may create more unpredictable passwords than other methods, such as mnemonic phrase-based passwords. Our study has also highlighted some limitations in our approach. Most importantly, we found that in our stronger conditions, users compensated by selecting less secure initial passwords. We conclude that we must be careful to not overburden users as we help and show them how to behave more securely.

In our pilot study [10], we found that people with stronger initial passwords (such as those already containing special characters) had trouble remembering their improved passwords. By expanding the pilot study into the full study, our larger sample population had a smaller proportion of computer experts, and was more representative of typical users. As such, we cannot verify the previous result of the pilot study. However, PTP could accept sufficiently strong initial passwords without attempting to improve them further. We are currently examining the security and usability implications of such a mechanism.

8. CONCLUSION

In this paper, we have presented a user study of Persuasive Text Passwords, a password creation approach using Persuasive Technology by randomly positioning randomly-chosen characters into users' passwords. Our study involved several PTP variations and 83 participants. An important result of this study is identifying the point where the limits of human memory lead users to employ coping mechanisms when dealing with randomness in passwords. In the stronger conditions of our study, users compensated by choosing simpler passwords before applying the system's improvement. Even so, in order to remember the improved passwords, they needed to mentally rehearse the password for longer periods of time to be able to successfully use it. Our study suggests that PTP has merit in usability and security, and that 3 randomly-chosen and inserted characters is the most users can remember without exerting an unreasonable amount of mental effort. However, this result is relative to this particular study and must be verified by future studies before generalising the result for all circumstances.

There are a number of areas for future work on PTP. Each PTP variant should be tested with more participants to confirm the full study results we have presented. Long-term memorability of the improved passwords remains untested, as well as interference effects when users are required to remember multiple PTP-improved passwords. Use in a realistic field-study setting is necessary to better gauge PTP's suitability for general use. Also, there are many possible variants of PTP, some of which may be more effective than those presented here. An authentication system may become more secure if many PTP variations were implemented, and users could either choose or be randomly-assigned a variant. We intend to both gather more data on PTP to gauge its real-world applicability, and to explore more avenues of influencing and aiding users in authenticating and behaving more securely.

Finally, as stated in Section 5, the crude security metric we use to evaluate the strength of individual passwords does not account for user biases towards choosing predictable character combinations. We are currently investigating a more appropriate and accurate metric for evaluating and comparing the strength of individual passwords.

9. ACKNOWLEDGEMENTS

The first, second, and fourth authors are supported in part by the "Legal and Policy Approaches to Identity Theft" project funded by the Ontario Research Network for E-Commerce. The third author's research is partially funded by NSERC under a Discovery Grant and the Canada Research Chairs program.

10. REFERENCES

[1] Adams, A. and Sasse, M.A. Users Are Not The Enemy. Communications of the ACM 42, 12 (1999), 41-46.

[2] Bartavelle. Patches for John the Ripper. Accessed February 2008, http://www.banquise.net/misc/patch-john.html

[3] Burr, W.E., Dodson, D.F., and Polk, W.T. Electronic Authentication Guideline. NIST Special Publication 800-63, Version 1, 2004.

[4] Chiasson, S., Forget, A., Biddle, R., and van Oorschot, P.C. Influencing Users Towards Better Passwords: Persuasive Cued Click-Points. British Computer Society HCI 2008.

[5] Chiasson, S., van Oorschot, P.C., and Biddle, R. A Usability Study and Critique of Two Password Managers. USENIX Security Symposium 2006, 1-16.

[6] Designer, S. John the Ripper password cracker. Accessed February 2008,

http://www.openwall.com/john/

[7] Florencio, D., Herley, C., and Coskun, B. Do Strong Passwords Accomplish Anything? USENIX Workshop on Hot Topics in Security 2007.

[8] Florencio, D. and Herley, C. A Large-Scale Study of Web Password Habits. WWW 2007, ACM Press, 657-666.

[9] Fogg, B.J. Persuasive Technology: Using Computers to Change What We Think and Do. Morgan Kaufmann, San Francisco, USA, 2003.

[10] Forget, A., Chiasson, S., van Oorschot, P.C., and Biddle, R. Persuasion for Stronger Passwords. Persuasive Technology 2008, Springer-Verlag.

[11] Forget, A., Chiasson, S., and Biddle, R. Persuasion as Education for Computer Security. AACE E-Learn 2007, 822-829.

[12] Furnell, S. An assessment of website password practices. Computers & Security 26, 7-8 (2007), 445-451.

[13] Halderman, J.A., Waters, B., and Felten, E.W. A Convenient Method for Securely Managing Passwords. ACM WWW 2005, 471-479.

[14] Jeyaraman, S. and Topkara, U. Have the cake and eat it too - Infusing usability into text-password based authentication systems. IEEE ACSAC 2005, 473-482.

[15] Jermyn, I., Mayer, A., Monrose, F., Reiter, M.K., and Rubin, A.D. The Design and Analysis of Graphical Passwords. USENIX Security Symposium 1999.

[16] Kuo, C., Romanosky, S., and Cranor, L.F. Human Selection of Mnemonic Phrase-based Passwords. ACM SOUPS 2006, 67-78.

[17] Leonhard, M.D. and Venkatakrishnan, V.N. A Comparative Study of Three Random Password Generators. IEEE EIT 2007, 227-232.

[18] Peterson, L.R. and Peterson, M.J. Short-term retention of individual verbal items. Experimental Psychology 58, 3 (1959), 193-198.

[19] Pond, R., Podd, J., Bunnell, J., and Henderson, R. Word Association Computer Passwords: The Effect of Formulation Techniques on Recall and Guessing Rates. Computers & Security 19, 7 (2000), 645-656.

[20] Proctor, R.W., Lien, M.-C., Vu, K.-P.L. Improving computer security for authentication of users: Influence of proactive password restrictions. Behavior Research Methods, Instruments, & Computers 32, 2 (2002), 163-169.

[21] Ramsbrock, D., Berthier, R., and Cukier, M. Profiling Attacker Behaviour Following SSH Compromises. IEEE International Conference on Dependable Systems and Networks 2007.

[22] Ross, B., Jackson, C., Miyake, N., Boneh, D., and Mitchell, J.C. Stronger Password Authentication Using Browser Extensions. USENIX Security Symposium 2005, 17-31.

[23] Shannon, C.E. Prediction and Entropy of Printed English. Bell System Technical Journal 30, 1 (1951), 50-64.

[24] St. Clair, L., Johansen, L., Enck, W., Pirretti, M., Traynor, P., McDaniel, P., and Jaeger, T. Password Exhaustion: Predicting the End of Password Usefulness. ICISS 2006, Springer-Verlag, 37-55.

[25] Seifert, C. Analyzing Malicious SSH Login Attempts. Security Focus Infocus article, September 2006. http://www.securityfocus.com/infocus/1876, accessed May 2008.

[26] Thames, J.L., Abler, R., and Keeling, D. A Distributed Active Response Architecture for Preventing SSH Dictionary Attacks. IEEE Southeastcon 2008, 84-89.

[27] Vu, K.-P.L., Proctor, R.W., Bhargav-Spantzel, A., Tai, B.-L., Cook, J., and Schultz, E.E. Improving password security and memorability to protect personal and organizational information. International Journal of Human-Computer Studies 65, 8 (2007), 744-757.

[28] Whitten, A. and Tygar, J.D. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. USENIX Security Symposium 1999, 169-183.

[29] Yan, J., Blackwell, A., Anderson, R., and Grant, A. Password Memorability and Security: Empirical Results. IEEE Security & Privacy Magazine 2, 5 (2004), 25-31.