Surpass: System-initiated User-replaceable Passwords

Jun Ho Huh[1], Seongyeol Oh[2], Hyoungshick Kim[2], Konstantin Beznosov[3],
Apurva Mohan[1], S. Raj Rajagopalan[1]
[1]Honeywell ACS Labs, Golden Valley, MN USA
[2]Sungkyunkwan University, Suwon, Korea
[3]University of British Columbia, Vancouver, BC Canada
{junho.huh, apurva.mohan, siva.rajagopalan}@honeywell.com
{seongyeol, hyoung}@skku.edu
beznosov@ece.ubc.ca

ABSTRACT

System-generated random passwords have maximum password security and are highly resistant to guessing attacks. However, few systems use such passwords because they are difficult to remember. In this paper, we propose a system-initiated password scheme called "Surpass" that lets users replace few characters in a random password to make it more memorable.

We conducted a large-scale online study to evaluate the usability and security of four Surpass policies, varying the number of character replacements allowed from 1 to 4 in randomly-generated 8-character passwords. The study results suggest that some Surpass policies (with 3 and 4 character replacements) outperform by 11% to 13% the original randomly-generated password policy in memorability, while showing a small increase in the percentage of cracked passwords. When compared to a user-generated password complexity policy (that mandates the use of numbers, symbols, and uppercase letters) the Surpass policy with 4-character replacements did not show statistically significant inferiority in memorability. Our qualitative lab study showed similar trends. This Surpass policy demonstrated significant superiority in security though, with 21% fewer cracked passwords than the user-generated password policy.

Categories and Subject Descriptors

D.4.6 [**Security and Protection**]: Authentication; H.1.2 [**User/Machine Systems**]: Human factors

General Terms

Experimentation, Human Factors, Measurement, Security

Keywords

Security; usability; passwords; policy

CCS'15, October 12–16, 2015, Denver, Colorado, USA.

ACM 978-1-4503-3832-5/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2810103.2813622 .

1. INTRODUCTION

To encourage users to choose strong passwords, companies commonly use password complexity policies (e.g., requirements for minimum length, symbols, numbers, and uppercase letters) or password strength meters. Recent research [15, 18] has shown that those mechanisms are indeed effective in increasing password entropy. However, advanced attackers are still finding ways to efficiently crack such passwords offline using various forms of hybrid attacks (marrying dictionary attacks with brute-force attacks). *Randomly-generated passwords* would make such offline attacks infeasible, guaranteeing the highest possible password entropy, but they suffer from memorability issues. Some systems (e.g., server management systems) mandate the use of randomly-generated passwords to protect admin and other highly privileged accounts. However, randomly-generated passwords have a limited adoption rate for one simple reason: they have low usability [4, 13].

To overcome usability issues with randomly-generated passwords, this paper proposes a novel password scheme called "Surpass" (**s**ystem-initiated **u**ser-**r**eplaceable **pass**words). Surpass allows users to *replace* several characters from a randomly-generated password to make it more memorable. The initial password is *randomly generated* from a set of 94 characters, which includes numbers, lowercase and uppercase letters, and special characters. We conducted a large-scale online user study, recruiting a total of 5,412 Mechanical Turk participants to evaluate both the memorability and security of 8-character Surpass passwords. We tested our approach by varying the "number of character replacements allowed," from 1 to 4. We then compared the generated passwords against the original 8-character randomly-generated passwords and passwords users generated under a real-world password complexity policy (that mandates the use of symbols, numbers, and uppercase letters).

Our evaluation results suggest that Surpass memorability improves as the number of allowed character replacements increases. The second-day memorability score of the original randomly-generated passwords was just 65%, but that score jumped to 76% when 3 (out of 8) character replacements were allowed (see Table 3). User-chosen passwords did not show statistically significant superiority in memorability against Surpass passwords when 4 character replacements were allowed. Yet, they showed lower guessing entropy and a 21% jump in the percentage of cracked passwords. Our lab study results showed a higher "eighth-day survival rate" (the

percentage of participants who correctly recalled their passwords after *8 days*) for those Surpass passwords compared to user-chosen passwords.

To summarize, the key contributions of this paper are as follows: (i) We proposed Surpass, a novel approach for improving the memorability of randomly-generated passwords by replacing several characters, while preserving their resistance to guessing attacks. (ii) We performed a large-scale empirical evaluation of a range of Surpass policies, which suggests that we can significantly improve memorability of 8-character, randomly-generated passwords with small reduction in their strength. (iii) We compared the best-performing Surpass policy against a real-world password complexity policy. The comparison suggests statistically significant superiority in security and insignificant inferiority in memorability of Surpass passwords. We further validated that observation through a lab study, experimenting with an eight day password recall period. (iv) We performed a large-scale empirical comparison of 8-character randomly-generated passwords with user-generated passwords. The results suggest that the memorability of randomly-generated passwords is about 17% lower.

2. RELATED WORK

To help users choose stronger passwords, various password complexity policies have been introduced [15, 18]. A commonly used policy mandates that passwords must contain at least 8 characters, use both upper- and lower-case letters, include one or more numerical digits, and include one or more symbols. Shay et al. [18] have analyzed that policy, showing that it is effective in improving password entropy but also has usability issues; some users struggled to comply with the new policy, taking longer to create passwords and finding it harder to remember them.

Even with such password selection policies in place, users find ways to create weak passwords like "Letmein1!" or "Garrett1993*." Such passwords are still vulnerable to offline hybrid attacks (that marry dictionary attacks with brute-forcing attacks) or combinator attacks (combining every word in a dictionary with every other word in a dictionary) [7]. Hashcat [11] is a popular, freely available password-cracking software that supports all of those advanced cracking techniques and performs probabilistically-ordered (intelligent) per-position brute-force attacks by looking at the list of cracked passwords. Using Hashcat, a group of white-hat hackers deciphered 90 percent of 16,000 hashed passwords within a few hours [10], including complicated passwords like "Qbesancon321" and "qeadzcwrsfxv1331." Their efforts demonstrate that passwords created through password complexity policies have limited impact on security.

Some studies have investigated the usability of randomly-generated passwords. Byuyan et al. [3] conducted a user study to compare the usability and memorability of 6-character randomly-generated passwords with 8- and 16- character user-generated passwords. The evaluation was a little tilted in favor of randomly-generated passwords because they were drawn from just 36 characters on the standard US QWERTY keyboard compared to the 94 character set used for user-generated passwords. This study reported better performance of randomly-generated passwords for recall rates and times, but the use of different character sets and a small study population of 54 users make the result inconclusive. Bonneau et al. [6] showed that spaced repetition learning techniques can be effective on remembering 56-bit random passwords. Yan et. al. [20] compared security and memorability of mnemonic pass-phrases against random passwords, showing that mnemonic pass-phrases are just as difficult to guess as random passwords, but can be more memorable. Their memorability analysis, however, was done purely through a user reported survey. Huh et al. [12] studied memorability of random PINs, showing that random PINs are hard to remember in general. Shay et al. [16] compared usability of randomly-assigned pass-phrases (composed of a sequence of words) against randomly-generated passwords of similar entropy, demonstrating that randomly-assigned pass-phrases are actually less usable. Memorability weakness [13] is the main reason that randomly-generated passwords have a limited adoption rate.

Surpass can significantly improve the memorability of 8-character, randomly-generated passwords to achieve 78% second-day memorability (see Table 3), which is about 13% higher than original randomly-generated passwords. Based on our search of published work, we believe we are the first group to explore the notion of allowing users to replace a few characters from randomly-generated passwords to make them more usable and to evaluate the concept through a large-scale user study. Forget et al. [9] took a different approach, starting with user-generated passwords and increasing entropy bits by either adding 2-4 random characters or replacing two original characters with random characters. The two schemes are different in design and user behavior because Surpass starts from random passwords that are robust against offline attacks, and aims to *improve usability of system-generated passwords*. Hybrid attacks that try adding random characters to different positions of dictionary words, or attacks that try replacing characters from different positions can be effective against their scheme. In Section 5.3, we demonstrate that such attacks, performed using Hashcat, have a limited impact on Surpass passwords.

3. METHODOLOGY

This section lists our research questions and the hypotheses, explains the design of the user study, and describes the recruitment of the participants.

3.1 Hypotheses

This work was motivated by two research questions: (1) How usable and memorable are randomly-generated 8-character passwords, compared to passwords users choose, subject to a password complexity policy?[1] (2) Can allowing users to replace several characters help improve the memorability of randomly-generated passwords, and if so, by how much?

Based on those research questions and our intuition, we defined the following three hypotheses: (H1) The memorability of Surpass passwords improves with the increase in the number of character replacements allowed. (H2) The security of Surpass passwords weakens with the increase in the number of character replacements allowed. (H3) A Surpass policy that shows no statistically significant difference in memorability against the password complexity policy (user-chosen passwords), has *better security* than the complexity policy. The user study and experiments were designed to

[1]In this paper, "user-chosen passwords" refers to passwords that are chosen by users and constrained by a password complexity policy defined in Table 1.

Table 1: Surpass policies.

Policy	*Description*	*Example*
`0-Change`	Users are not allowed to replace any character from a given randomly generated 8-character password.	7^^V[wf.
`1-Change`	Users may replace up to *one* character from a given randomly generated 8-character password.	7^^V[wf. → 7^^V[w].
`2-Change`	Users may replace up to *two* characters.	7^^V[wf. → 7^^v[w].
`3-Change`	Users may replace up to *three* characters.	7^^V[wf. → v^^v[w].
`4-Change`	Users may replace up to *four* characters.	7^^V[wf. → v^^v[1].
`User`	Users are required to choose a password that is at least 8 characters long, not similar to their name, not similar to other commonly used passwords, contains both upper- and lower-case letters, and contains one or more numerical digits and special characters.	Letmein1!

validate the above hypotheses. In Section 7, we discuss how the study results match up to these hypotheses.

3.2 Surpass policies

Each Surpass policy defines *the number of character replacements allowed* for a given randomly generated 8-character password (see Table 1). We decided to experiment with randomly generated passwords that are 8-characters long since that is a common standard in real-world systems [18]. Our first policy, `0-Change`, replicates an existing, real-world randomly generated 8-character password policy. To test Hypotheses 1 and 2 (see Section 3.1), we created Surpass policies from `1-Change` to `4-Change`, increasing the number of character replacements allowed until we reached halfway with `4-Change`, which would allow users to replace up to 50% of the original password. To test Hypothesis 3, we replicated a real-world password complexity policy used in Windows[2] Active Directory to represent a widely-adopted, sufficiently strong user-chosen password policy, and created the `User` policy. Table 1 provides the policy details. Each policy is effectively a separate experimental condition.

3.3 User study design

We evaluated the six policies through empirical quantitative experiments using Amazon Mechanical Turk. Before starting the study, participants were asked to acknowledge a consent form, which explained the study purposes and instructions, and informed that participation is voluntary and confidential and they have the right to terminate the study any time without penalty. Data were collected confidentially only for the purposes of conducting statistical analyses. Ethical perspective of our research was validated through an ethics committee at a university. To evaluate the applicability of Surpass to password systems that are susceptible to offline guessing attacks and to make the study as realistic as possible, we employed role-playing by simulating the Windows Active Directory (a centralized authentication service) password setup and login processes. In this between-subject study, each participant was assigned a specific policy (picked uniformly at random). Participants assigned to the `0-Change` policy were asked to remember the given randomly generated password as is; those assigned to the `User` policy were asked to generate their own passwords, satisfying all of the policy rules (see Table 1). For all of the Surpass policies and the `0-Change` policy, participants were given an option to "regenerate" passwords an unlimited number of times.

Our user study was designed following the Atkinson-Shiffrin dual memory model [2]. This model postulates that memories initially reside in a "short-term" memory for a limited time (20 to 30 seconds). Short-term memory has limited capacity and older items are wiped as new items enter. Further, rehearsing or recalling items while they are in the short-term memory causes the items to stay longer in the short-term memory.

Based on Atkinson-Shiffrin memory model, our data collection involved two parts. The first part (1) ensured that password of each participant entered their short-term memory and (2) measured the "first-test memorability" of that password. This part consisted of three steps. First, each participant was asked to complete three training (rehearsing) tasks to help them remember their new password (associate it with long-term memory). Next, each participant was asked to complete a moderately challenging lexical puzzle, which was intended to wipe out their short-term memory of textual information (with new items) during the process. Last, each participant completed a memorability test (testing his or her short-term memory strengthened through training) by entering the password.

Two days (48 hours) after completing the first part, each participant received an email inviting her to the second part of the study. Here, we measured the "second-test memorability" of passwords created under different policies. A two day password recall period was used because in many of the targeted Windows Active Directory environments (e.g., corporations) users use their passwords on a daily basis, but often do not use their passwords for two consecutive days during weekends. A few previous password studies have also used two days as the password recall period (e.g., [8,14,15,17,19]). The following paragraphs detail the data collection tasks in the order participants were asked to complete them.

Part 1

1. Password setup: Each participant was randomly assigned a policy. For the `0-Change` policy and all Surpass policies, participants were given a randomly generated 8-character password. Those assigned to the `0-Change` policy were asked to remember the given password as is. Participants assigned to the Surpass policies were asked to *make the given password more memorable* by replacing an allowed number of characters with any character of their choice. Those assigned to the `User` policy were asked to generate a password based on the password rules defined in Table 1.

2. Password memorization: Each participant was asked to enter the correct password three times to help with memorization. If incorrect passwords were entered three times consecutively, the correct password was revealed again so that the participant would have another chance to memorize it. The training session ended only when the participant entered the correct password three times.

3. Puzzle: Each participant was asked to complete a moderately challenging lexical puzzle, which takes about 2 minutes to complete.

[2]All trademarks used herein are the property of their respective owners.

4. Demographics and survey question: Each participant was asked five demographic questions and a survey question about whether an external storage (e.g., a sheet of paper or a text file) has been used to write down his or her password.

5. Enter password (first memory test): Each participant was asked to enter their password and was given three chances to enter it correctly. To simulate Windows login scenario, we asked participants to enter the correct password to log in to Windows. Those who entered the correct password were informed that they would be invited to the second part and be asked to enter the password again.

Part 2

6. Enter password after two days (second memory test): After two days (48 hours), participants, who entered the correct password in Step 5, received an email asking them to complete the second part of the study. When participants returned to the study web site, they were asked to enter their passwords again and each was given three chances to enter the correct password. Participants' user IDs, associated with their passwords, were used to check whether a correct password was entered.

7. Survey question: After completing the test, participants were asked the same question about the use of an external storage.

To prevent participants from copying and pasting passwords, we *disabled the copy and paste feature for all of the text fields*. Before running the real study, we conducted a small-scale pilot study in a lab environment to find and fix bugs and address any confusion participants may have had with respect to study instructions and policy descriptions. We recruited a total of 161 participants for the pilot study. 10 participants who agreed to be interviewed were asked about their experience of participating in the study. Based on the pilot, the main improvement of the study design was in the wording of instructions for participants.

3.4 User data collected

Throughout the above six steps of the user study, we recorded the following information:

Original password, finalized password, and password policy. For each participant, we recorded the password policy and both the original password and the finalized password (for the `0-Change` and `User` policies the two passwords are the same).

Number of regenerations. We recorded the number of times a participant regenerated the initial (original) password during password setup.

Number of attempts made in entering the password. For all of the training sessions and first and second memory tests, we recorded the number of attempts a participant made to enter the correct password.

Time taken for password setup. We measured the time it took each participant to set up their password, starting from when the participant first saw the password setup page and ending when the participant successfully created their password. To complete the setup process, each participant had to create a password that conformed to the given password policy; the reentered password had to match the created password.

Time taken for password entry. For all of the training sessions and first and second memory tests, we measured the time it took each participant to enter a password for each attempt made.

Time taken to authenticate. In both the first and second memory tests, we also measured the time it took each participant to complete authentication. Timing began when the participant first saw the login screen and ended when the participant either entered the correct password or tried and failed all three attempts.

Memorability results. For all of the training sessions and first and second memory tests, we recorded the results of the memorability tests (i.e., whether a correct password was entered) for each attempt made.

Survey answers. We recorded participants' answers to the survey question about the use of an external storage to write down passwords.

3.5 Mechanical Turk

To conduct a large-scale study, we used Amazon Mechanical Turk [1] for recruiting participants in the main study. Participants needed to be located in the USA and at least 18 years old. Every participant who completed the first part was rewarded with $0.50. Those who came back and completed the second part were rewarded with additional $0.30.

3.6 Statistical tests

Without making any assumptions on data distributions, we performed the Fisher's Exact Test (FET) to compare the proportion of successful logins, cracked passwords, and external storage usage for the six policies. The statistical confidence in the password setup time, authentication time, and longest length of consecutively repeating characters differences between the six policies were tested using unpaired Mann-Whitney U test (MW U test) because the collected data was not normally distributed. Post-hoc comparisons were corrected for multiple-testing using False Discovery Rate (FDR) estimation when appropriate.

4. RESULTS: USABILITY

This section presents the key usability results from the user study, discussing password memorability and authentication times.

4.1 Demographics

As described in Section 3.5, participants were recruited using Mechanical Turk. During the five-week study period, a total of 5,412 participants completed the first part of the study, and 3,839 came back to complete the second part. Most participants were white (74%), and the majority were in the 18–29 and 30–39 age groups, 55% and 25%, respectively. 52% were male. 56% had a university degree, and 31% had a high school diploma.

4.2 Memorability

4.2.1 First test

All of the password policies scored high in first-test memorability, ranging between 97% and 99% (Table 2). All of the Surpass policies scored higher than the `0-Change` policy, which scored 97%, with policies `3-Change` and `4-Change` showing a statistically significant difference ($p < 0.04$ and $p < 0.01$ respectively, pairwise corrected FET). Policy `4-Change` also outperformed `1-Change` and `2-Change` with statistical significance ($p < 0.03$ and $p < 0.05$ respectively,

Table 2: First-test memorability and average time taken to authenticate. Column '% correct pwd' shows the percentage of participants who entered the correct password in the first test (see Section 3.3). '% ext. storage' shows the percentage of participants who reported having written down their password. Column '% correct, no storage' represents the same percentage but not counting those who reported to have their password written down. Column 'Time' is the average time taken to authenticate and is measured in seconds.

Policy	*# Participants*	*# Failed*	*% Correct pwd*	*% Ext. storage*	*% Correct, no storage*	*Time (s)*	σ
`0-Change`	903	26	97%	17%	97%	21.8	24.2
`1-Change`	888	22	98%	15%	97%	19.2	19.9
`2-Change`	868	20	98%	12%	97%	20.0	38.0
`3-Change`	906	12	99%	9%	99%	16.1	16.1
`4-Change`	911	8	99%	10%	99%	15.1	21.6
`User`	936	6	99%	5%	99%	15.1	21.7

Table 3: Second-test memorability and average authentication time.

Policy	*# Participants*	*# Failed*	*% Correct pwd*	*% Ext. storage*	*% Correct, no storage*	*Time (s)*	σ
`0-Change`	601	217	65%	18%	58%	76.2	103.0
`1-Change`	631	213	67%	17%	61%	70.9	87.6
`2-Change`	604	190	70%	15%	64%	67.1	94.8
`3-Change`	645	156	76%	17%	71%	66.9	123.1
`4-Change`	650	142	78%	16%	74%	64.7	127.1
`User`	706	126	82%	7%	81%	46.4	82.6

pairwise corrected FET). Policy `User` outperformed policies `0-Change`, `1-Change`, and `2-Change` with statistical significance (all $p < 0.02$, pairwise corrected FET), but achieved the same score as policies `3-Change` and `4-Change` at 99% and showed no statistically significant difference ($p = 0.16$ and $p = 0.4$ respectively, pairwise corrected FET).

Table 2 also shows the percentage of participants who used external storage to keep track of passwords. Fewer participants in policy `User` used external storage compared to all other policies (all $p < 0.004$, pairwise corrected FET). Except for policy `1-Change`, all of the Surpass policies showed statistically significant superiority against `0-Change` (all $p < 0.01$, pairwise corrected FET). Memorability scores did not change much after excluding those who used external storage.

4.2.2 Second test

In the second test, all of the Surpass policies scored better than policy `0-Change` (see Table 3), clearly showing that allowing users to replace characters with their own helps with second-test memorability. It is unsurprising, then that, among the Surpass policies, `4-Change` showed the strongest second-test memorability at 78%—13% greater than `0-Change`; its superiority is statistically significant ($p < 0.002$, pairwise corrected FET). `4-Change` outperformed `1-Change` and `2-Change` with statistical significance by 11% ($p < 0.004$, pairwise corrected FET) and 8% ($p < 0.02$, pairwise corrected FET), respectively. The second strongest policy for second-test memorability was `3-Change`, which scored 76%, outperforming `0-Change` and `1-Change` with statistical significance by 11% ($p < 0.007$, pairwise corrected FET) and 9% ($p < 0.02$, pairwise corrected FET), respectively.

Policy `User` scored better than all other policies, showing statistically significant differences against all (all $p < 0.05$, pairwise corrected FET), *except for* `4-Change` ($p = 0.12$, pairwise corrected FET). Its second-test memorability outperformed `0-Change` by 17%, but that gap was reduced to just 6% and 4% with `3-Change` and `4-Change`, respectively.

In contrast to the first study, the overall differences in external storage use were less significant in the second study between the Surpass policies and `0-Change` (see Table 3). Policy `User` still showed statistically significant differences against all other policies (all $p < 0.001$, pairwise corrected FET); the difference was as large as 11% against `0-Change`. After excluding those who used external storage, the memorability score dropped 1%, 4%, and 5% for policies `User`, `4-Change`, and `3-Change`, respectively. `0-Change` showed the largest drop at 7%. We note that the storage use percentage for policy `User` is much smaller than what was presented in [17]. We surmise that such differences could have partly resulted from the three memorization (training) sessions that were only included in our study.

4.3 Time taken to authenticate

Tables 2 and 3 show *average* time taken to authenticate for the first and second memorability tests, respectively. Both successful and unsuccessful authentications were averaged. After observing high variations in the time data (see standard deviations), we used FDR to run an unpaired MW U test corrected for multiple testing to see which average time differences were statistically significant.

In the first test, policy `User` outperformed all other policies with statistical significance (all $p < 0.001$, unpairwise corrected MW U test). All of the Surpass policies outperformed `0-Change` with statistical significance (all $p < 0.001$, unpairwise corrected MW U test). Policies `4-Change` and `3-Change` did not show much difference in authentication times (all $p = 0.77$, unpairwise corrected MW U test), but outperformed the other two Surpass policies with statistical significance (all $p < 0.001$, unpairwise corrected MW U test). Policies `4-Change` and `3-Change`, with averages of 15.1 and 16.1 seconds, respectively, showed significantly less average time than `0-Change`, which was 21.8 seconds.

All of the statistically significant differences found in the first test were also found in the second test, except that policy `3-Change` did not show statistically significant superiority over `2-Change` ($p = 0.06$, unpairwise corrected MW U test). Again, policies `4-Change` and `3-Change` did not show statistically significant difference in authentication times ($p = 0.2$, unpairwise corrected MW U test).

Table 4: Average number of password setup attempts and average setup time

Policy	*Time (s)*	σ	*# Attempts*	σ
`0-Change`	124.3	112.7	1.8	1.1
`1-Change`	122.2	87.8	2.0	1.2
`2-Change`	132.8	118.3	2.0	1.4
`3-Change`	130.0	89.9	2.0	1.2
`4-Change`	141.0	115.6	2.0	1.3
`User`	92.9	82.6	1.7	0.9

4.4 Number of setup attempts and setup time

Table 4 shows the average number of attempts users made to set up their passwords. Password setup failure could occur if a participant's password did not conform to the password policies described in Table 1 or if the reentered password was different from the original. The average number of attempts was between 1.7 and 2, with a standard deviation around 0.9-1.4. These results show that most participants, regardless of the policy used, made about two attempts to set up a password. Although policy `0-Change` had a lower average value compared to all of the Surpass policies (all $p < 0.001$, unpairwise corrected MW U test), 1.8 attempts on average is still quite high, considering that each participant was asked to simply enter their password twice. This outcome indicates that many participants failed at correctly entering their passwords, which demonstrates that just typing randomly-generated passwords in the first setup attempt is not an easy task. Policy `User` had the lowest average value at 1.7 and showed statistical significance against all other policies (all $p < 0.001$, unpairwise corrected MW U test), except policy `0-Change`.

The average time taken to set up a password shows more variations (Table 4). Again, policy `User` was a clear winner, at 92.9 seconds on average, showing statistically significant superiority over all other policies (all $p < 0.001$, unpairwise corrected MW U test). Policy `0-Change`, with average of 124.3 seconds, outperformed `2-Change`, `3-Change`, `4-Change` with statistical significance ($p < 0.01$, unpairwise corrected MW U test). The average number of times a participant regenerated the original password during setup across all Surpass policies was just between 2-3.

5. RESULTS: SECURITY

We now present the key security results, including password compositions, character changing behaviors in Surpass policies, proportions of cracked passwords, and guessing entropy estimates.

5.1 Password composition

Password compositions are explained in terms of the frequency of character categories used, use of English dictionary words and dates, use of consecutively repeating characters, and character change behaviors of Surpass participants.

5.1.1 Character frequency analysis

Fig. 1 shows the frequency with which each category of characters (lowercase letters, uppercase letters, numbers, and symbols) is used in each password position. Each graph represents a different character category, illustrating the character category distributions across all six password policies. The most noticeable graphs are (b) and (d), which show that, with `User` passwords, most of the uppercase letters appeared in the first position and most of the symbols appeared in the last position. For those reasons, lowercase letters appeared far less in the first and the last positions of `User` passwords. Interestingly, in graph (c), the number of symbols used decreases in every position as we move from `1-Change` to `4-Change`. This implies that Surpass participants generally want to replace symbols with character types. Our analysis on character changing behavior of participants in Section 5.2 reinforces that observation.

The use of numbers in graph (c) were also interesting, showing a increasing trend toward the last position. In contrast, Surpass policies showed far more uniform distributions across all character categories, again, demonstrating characteristics that are more similar to randomly-generated passwords than `User` passwords. All of these observations indicate that it would be much more difficult to perform offline guessing attacks on Surpass passwords using hybrid attacks like "adding a symbol or a number at the end of a word." Our password cracking results in Section 5.3 support this conclusion.

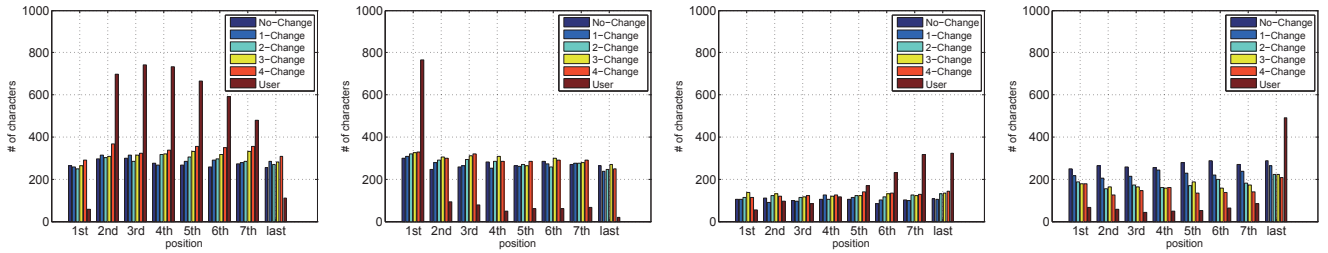
Next, we looked at the top 10 most frequently used symbols across all password policies. Fig. 2 shows the top 10 symbols and their usage distributions for each policy. Policy `User` showed the most skewed distribution of symbols, starting with the symbol "!" at 35% and ending with the symbol "&" at just 1.4%. A similar trend in symbol usage distribution for a password complexity policy has been observed by [15]. Policies `3-Change` and `4-Change` showed mild skew with 6.9% and 7.7% usage of "!," respectively, as the most frequently used symbol. But those were only 2.9% and 3.7% increases from the percentage of the most frequently appearing symbol in policy `0-Change` (which was also "!"), compared to 31% increase shown in policy `User`.

Until the 5th ranking, many symbols overlap between the Surpass policies and policy `User`. However, from the 6th ranking, we started seeing a few symbols in Surpass passwords that were not popular among `User` passwords. Symbols such as "-," "<," ">," and "=" were popularly used in the Surpass policies but were not ranked high in the `User` policy. One reason for this difference is that the *uncontrolled presence* of such symbols (in the original passwords) encouraged the participants to use them to create patterns, shapes, equations, and emoticons that are easy to remember.

Our analysis on the proportion of English dictionary words and dates used shows that `User` passwords had the highest proportions for word-containing passwords (75%) and date-containing passwords (10%). Both proportions were significantly large compared to all Surpass policies (all $p < 0.0001$, pairwise corrected FET), indicating that `User` passwords are more susceptible to dictionary attacks. As expected, the proportion of passwords that contain words increased from 9% in policy `0-Change` to 36% in `3-Change` and 45% in `4-Change`. This usage shows that the risk of Surpass passwords being susceptible to dictionary attacks increases with the increase in the number of character replacements allowed.

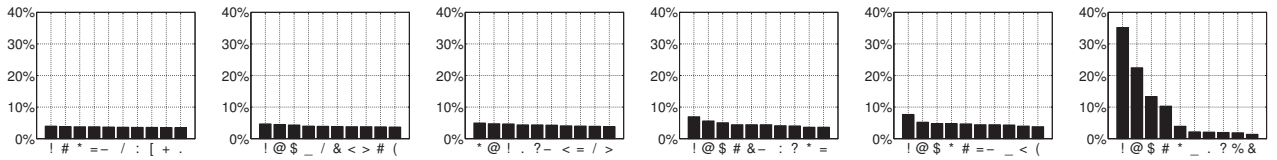
5.1.2 Consecutively repeated characters

One security concern with Surpass is that users could simply replace characters in the original password include one character repeated consecutively in the final password. For example, an original `4-Change` password "Xn8F,m*F" was changed into "XnFFFFFF" when the participant replaced



(a) Lowercase letters (b) Uppercase letters (c) Numbers (d) Symbols

Figure 1: Frequency of character categories used in each password position.



(a) 0-Change (b) 1-Change (c) 2-Change (d) 3-Change (e) 4-Change (f) User

Figure 2: Top 10 frequently occurring symbols across six conditions.

Table 5: The longest strands of repeating characters in passwords across six policies. Columns 'L2' to 'L7' count the number of passwords with strands of repeating characters of lengths 2 to 7. Column 'μ' is the average of the length of the longest repeating characters.

Policy	*L2*	*L3*	*L4*	*L5*	*L6*	*L7*	μ
`0-Change`	87	0	0	0	0	0	2.000
`1-Change`	151	7	0	0	0	0	2.044
`2-Change`	170	34	2	0	0	0	2.185
`3-Change`	209	48	17	3	0	0	2.329
`4-Change`	214	24	18	9	2	0	**2.356**
`User`	339	24	2	2	0	1	2.106

Table 6: Character change analysis for character categories across four Surpass conditions. 'Org.' represents original passwords, and 'Fin.' represents final passwords after user engagement.

Category	State	*1-Change*	*2-Change*	*3-Change*	*4-Change*
Numbers	Ori.	59	105	155	214
	Fin.	119	247	371	424
Uppercases	Ori.	83	217	394	530
	Fin.	202	409	591	717
Lowercases	Ori.	131	288	449	585
	Fin.	299	576	820	1154
Symbols	Ori.	477	782	1023	1179
	Fin.	119	160	239	213

"8," ",," "m," and "*" with 4 "F"s. If such replacement behavior is common, attackers could define brute-forcing rules that try out consecutively repeating characters to crack Surpass passwords. To scrutinize that concern, we measured the length of the longest strand of repeating character in each password and summarized the results in Table 5. The average length of the longest repeating character strand increased as we moved from `0-Change` to `4-Change`. There were 68 `3-Change` passwords (about 7.1%) and 53 `4-Change` passwords (about 5.8%) that contained 3 or more consecutively repeating characters—a larger proportion than shown for policies `0-Change` and `User` (all $p < 0.001$, unpairwise corrected MW U test). We discuss in Section 7.4 how such replacement behaviors may be controlled.

5.2 Character replacement analysis

We now take a closer look at how the participants replaced original characters with their own choice of characters in Surpass passwords, and how that affects Surpass security.

5.2.1 Number of replacements

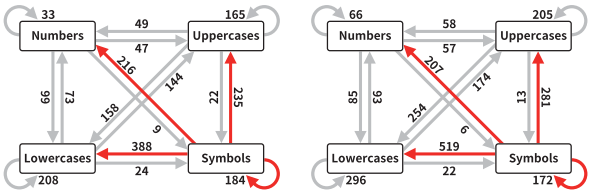
Table 6 shows the number of characters used for each character category (numbers, lowercase letters, uppercase letters, and symbols), both in the original passwords and in the final passwords. The most interesting observation comes from the replacement of symbols in Surpass passwords, where we saw significant reductions across all Surpass policies. As a result of this practice, the counts increased for all other character categories. Fig. 3 shows how those symbols were replaced for policies `3-Change` and `4-Change`—about 26% and 31% of the replaced symbols were replaced with lowercase letters, respectively. Most of the deleted uppercase letters were also replaced by lowercase letters, showing that lowercase letters were often favored by the participants.

5.2.2 Most frequently replaced characters

Table 7 shows the top 5 replaced characters for each character category across all Surpass policies. The least used symbols were "`," "|," "'," "{," and "[," which are all located at the far ends of the keyboard and are physically more difficult to reach and uncomfortable to type. The most popularly added symbols were "!," "$," "@," "*," and "')," which are all located at the top of the keyboard (where the numbers are) and are easier to reach. Symbols "!," "$," "@," were also the most popularly used symbols in `User` passwords (see Fig. 2). For both uppercase and lowercase letters, "X/x,"

Table 7: Top 5 replaced characters by character category across four Surpass conditions. 'Del.' represents deleted characters, and 'Add.' represents added characters.

Category	*Status*	*1st*	*2nd*	*3rd*	*4th*	*5th*
Numbers	Del.	6 (77)	5 (71)	9 (63)	0 (58)	1 (51)
	Add.	1 (257)	2 (157)	3 (148)	0 (143)	8 (87)
Uppercases	Del.	I (88)	X (65)	Q (62)	V (57)	G (53)
	Add.	A (224)	O (139)	I (111)	S (99)	E (97)
Lowercases	Del.	l (90)	v (82)	q (72)	x (69)	b (67)
	Add.	a (392)	o (340)	e (235)	i (191)	l (169)
Symbols	Del.	‘ (202)	| (152)	’ (146)	{ (146)	[(139)
	Add.	! (137)	$ (68)	@ (63)	* (48)	) (34)



(a) `3-Change` (b) `4-Change`

Figure 3: The number of replaced characters in each character category for `3-Change` and `4-Change`. Red arrows show how many symbols were replaced with other character categories.

"Q/q," and "V/v" were the most unpopular. The two most popular letters were "A/a" and "O/o." Another interesting observation is the most popularly added three digits, which were simply the first three digits that appear in the keyboard, "1," "2," and "3." The most unpopular two digits, "6" and "5" are both located in the middle of the keyboard.

5.3 Password cracking

As the next step, we evaluated the impact of the password policies on the success of popular password cracking techniques using Hashcat [11] and publicly available password dictionaries.

5.3.1 *Cracking techniques*

We wanted to simulate an *offline* guessing attack with sufficiently long attacking sessions. To that end, we used Hashcat as the password cracking tool, running it in its default "straight attack" mode and trying out about 65,000 randomly generated attack rules (available as part of the default `generated2.rule` ruleset) on publicly available password dictionaries. Rules can be specified to modify, cut, or extend dictionary words, allowing very flexible and accurate rule-based attacks to be crafted efficiently. "$1" is an example rule that appends character "1" to the end of a dictionary word. "i5!" is a rule that inserts character "!" at position "5." "ss$" is another example rule that replaces all instances of "s" with "$." More explanations of the possible rules are available in [11]. The straight attack mode combines dictionary attacks with such rule-based attacks. Our combined dictionary consisted of 21 million from Gnome, 14 million from RockYou, 0.5 million from Yahoo, and 0.3 million more from other leaked datasets. The combined dictionary was sorted in alphabetical order and duplicates were removed, resulting in 34.2 million unique entries. We used

Table 8: The number of cracked passwords across all six conditions.

Policy	*Total*	*# Cracked*	*%*	*Time (hr)*
`0-Change`	903	0	0	69hrs
`1-Change`	888	0	0	70hrs
`2-Change`	868	6	0.69%	70hrs
`3-Change`	906	11	1.21%	71hrs
`4-Change`	911	53	5.82%	72hrs
`User`	936	251	26.82%	71hrs

a server machine that is equipped with Intel Xeon 3.50GHz CPU and 32GB RAM to run Hashcat, recording the number of passwords cracked for each password policy.

5.3.2 *Cracking results*

The password cracking results are summarized in Table 8. None of the `0-Change` and `1-Change` passwords were cracked by 65,000 rules, demonstrating their robustness against commonly used rule-based attacks. Policies `2-Change`, `3-Change`, and `4-Change` did show statistically significant inferiority though, 0.69%, 1.21%, and 5.82% passwords of which were cracked, respectively (all $p < 0.02$, pairwise corrected FET). Such increases in the proportion of cracked passwords are the security tradeoffs for improving memorability (see Table 3).

Noticeably, 26.82% of `User` passwords were cracked, which was the largest proportion among the six policies. All Surpass policies showed statistically significant superiority over `User` in resisting popular offline rule-based cracking techniques (all $p < 0.0001$, pairwise corrected FET). It is worth noting that much smaller number of Surpass passwords were cracked using "add a number/symbol on tail/head" type of rules; such rules managed to crack many more `User` passwords. This effect can be explained by Fig. 1, which shows that most of the symbols and numbers appeared in the last position of `User` passwords.

Above results provide a meaningful insight into how Surpass would perform when popular cracking tools are used with real-world password dictionaries. However, the password dictionaries we used (see above) have characteristics that are more similar to `User` passwords than Surpass passwords, which could have produced more favorable results for Surpass. To evaluate Surpass security against attacks that use Surpass-specific dictionaries, we used the 2-gram Markov model to estimate partial guessing entropy. The results are presented in the next section.

5.4 Guessing entropy

To compare the password guessability between the six policies we calculated *partial guessing entropy* estimates [5]—this is a popularly used technique for estimating the average number of trials needed to successfully guess a fraction (α) of an entire password set. Because the collected set of passwords represented only a tiny portion of the theoretically possible password space, we employed the 2-gram Markov model to estimate the occurrence likelihood of every possible password. A separate Markov model was constructed for each policy based on the *Laplace smoothing* approximation technique (to cover rare cases).

Partial guessing entropy estimates are shown in Table 9. At α=`1E-12`, the guessing entropy estimates for policies `0-Change`, `1-Change`, and `2-Change` were significantly higher

Table 9: Partial guessing entropy estimates for each policy. "–" indicates that it was infeasible to estimate partial guessing entropy due to the limited size of a password set.

Policy \ α	*1E-13*	*5E-13*	*1E-12*	*5E-12*	*1E-11*	*5E-11*
`0-Change`	6	69	226	–	–	–
`1-Change`	7	46	136	–	–	–
`2-Change`	1	12	36	–	–	–
`3-Change`	1	1	1	8	115	–
`4-Change`	1	1	1	3	**12**	–
`User`	1	1	1	1	1	**12**

than those for the remaining policies. At $\alpha \leq$`1E-12`, the guessing entropy estimates were the same for policies `3-Change`, `4-Change`, and `User`. However, the differences between the guessing entropy estimates for those three policies increased significantly as α increased (i.e., when $\alpha >$`1E-12`); the proportion of `User` passwords that can be guessed with 12 trials ($\alpha =$`5E-11`) was about five times larger than the proportion of `4-Change` passwords that can be guessed with the same number of trials ($\alpha =$`1E-11`). When $\alpha=$`1E-11`, both `3-Change` and `4-Change` showed superiority in guessing entropy against `User`, and `3-Change` showed superiority against `4-Change`.

6. LAB STUDY

To further validate the first hypothesis ("The memorability of Surpass passwords improves with the increase in the number of character replacements allowed."), we conducted a separate lab study in a corporate environment of a large technology company that uses the simulated Windows Active Directory system for authenticating employees. We experimented with longer password recall periods, inviting participants to come back after two, four, and eight days after the initial study.

6.1 Methodology

To reduce the effect of technology background, we recruited employees from a wide range of business and strategic units (see Section 6.2). From about 500 employees we contacted, 60 employees participated in the study. We followed the methodology of the large-scale Mechanical Turk study (see Section 3), with the modifications explained below.

Instead of using Mechanical Turk, this study was conducted in a large meeting room, where the participants (before doing the study) were explained the study purpose, instructions, and how Surpass passwords can be created (based on the same examples given to the Mechanical Turk participants). We selected three representative policies to experiment with: policies `0-Change` and `User` were chosen as the real-world reference/base policies, and `4-Change` was chosen as the best performing Surpass policy that did not show statistically significant difference against policy `User` in the Mechanical Turk study. The participants were welcome to participate at anytime between 10am and 2pm.

To really stretch and test participants' memorability, we asked each participant to come back after three different time periods: *two days*, *four days*, and *eight days* after the initial study. After completing the initial study, the participants were told in advance that they will be invited back three more times in those time periods, and will be asked to enter their passwords again. Reminder emails were sent out a day before each of the returning studies, inviting the participants to come back to the same meeting room at anytime between 10am and 2pm to complete the next part. To accommodate for those with time conflicts, we arranged separate lab sessions for the participants who informed us about their time conflicts during those hours.

Every participant who completed the first part was rewarded with an $8 company lunch card. Those who came back and completed all of the remaining three parts were rewarded with another $8 lunch card. We measured the "eighth-day survival rate" (the percentage of participants successfully recalling passwords in the last part), which are presented in Table 11. We also asked specific questions about participants' recall difficulty and confidence levels. Those survey questions are provided in Table 10.

Table 10: Lab study survey questions. For LQ2, the options were "I wrote it down on a piece of paper or in an electronic device (e.g., phone, tablet, computer)," "I tried writing it down several times to help me remember," "I tried recalling it in my head several times," "I used my password on a real website or system," and "Other." For LQ3, the six-level Likert item format was "much more difficult to remember," "more difficult to remember," "same," "easier to remember," "much easier to remember," and "I cannot decide." For LQ4, the options were "No more than 1," "2 or 3," "4 or 5," "more than 5," and "not sure."

#	*Question*	*Answers*
`LQ1`	Since the last test, did you practice writing or remembering your password?	yes/no
`LQ2`	If you answered "Yes" to Q1, what methods did you use to practice? Tick all that apply	options
`LQ3`	How do you feel about the memorability of your password compared to the previous part of the study?	Likert scale
`LQ4`	How many different passwords of this kind do you think you can remember and use regularly without writing it down?	options

6.2 Demographics

A total of 60 participants completed the first part of the study, then 48, 42, 39 came back to complete the second part (after two days), third part (after four days), and the last part (after eight days), respectively. In contrast to the first study, we had more evenly distributed age groups (18% for group 18–29, 27% for group 30–39, 20% for group 40–49, and 20% for group 50–59) and greater proportion of participants with doctoral (17%) and masters degrees (33%). There were 65% male participants. 46.91% were advanced computer users, and 60.49% said that they spend more than 7 hours on computer each day. There was a great variety in the job titles, including administration (10%), marketing (10%), sales (2%), finance (18%), management (12%), and researchers (32%).

Table 11: (# survived participants)/(# initial participants) with 95% binomial confidence intervals across three policies.

Policy	*ST*	*2nd day*	*4th day*	*8th day*
`0-Change`	16/20	9/20	8/20	7/20
	80%	45%	40%	35%
	0.56, 0.94	0.23, 0.68	0.19, 0.64	0.15, 0.59
`4-Change`	18/19	16/19	16/19	15/19
	95%	84%	84%	79%
	0.74, 1.00	0.60, 0.97	0.60, 0.97	0.54, 0.94
`User`	21/21	18/21	18/21	15/21
	100%	86%	86%	71%
	0.84, 1.00	0.64, 0.97	0.64, 0.97	0.51, 0.87

Table 12: (# survived participants)/(# returned participants + # dropped out after failing one of previous tests) with 95% binomial confidence intervals across three policies.

Policy	*ST*	*2nd day*	*4th day*	*8th day*
`0-Change`	16/20	9/(12+4)	8/(8+7)	7/(7+7)
	80%	56%	53%	50%
	0.56, 0.94	0.30, 0.80	0.27, 0.79	0.23, 0.77
`4-Change`	18/19	16/(16+1)	16/(16+1)	15/(15+1)
	95%	94%	94%	94%
	0.74, 1.00	0.71, 1.00	0.71, 1.00	0.70, 1.00
`User`	21/21	18/(20+0)	18/(18+2)	15/(17+2)
	100%	90%	90%	80%
	0.84, 1.00	0.68, 0.99	0.68, 0.99	0.54, 0.94

6.3 Survival rates

First, we estimated the eighth-day survival rate for each policy based on the number of participants who correctly recalled their passwords on the last day and the number of initial participants. As Table 11 shows, the survival rates for policies `User` and `4-Change`, at 71% and 79%, respectively, were much higher than that of policy `0-Change`, which was just 35% ($p < 0.05$ and $p < 0.03$, respectively, pairwise corrected FET). Policy `User` did not show statistically significant difference in memorability against `4-Change` (p = 0.72, pairwise corrected FET).In fact, `4-change` showed a higher survival rate in the lab study.

Even though we tried our best to accommodate everyone to participate—setting up separate lab study sessions for those with time conflicts—some participants still dropped out without failing. Our second survival rate estimates, shown in Table 12, *exclude those who have dropped out without failing.* Again, the survival rates on the last day for policies `User` and `4-Change`, at 80% and 94%, respectively, were much higher than that of `0-Change`, which was 50%. With the new estimates, however, only `4-Change` showed statistically significant superiority against `0-Change` ($p < 0.04$, pairwise corrected FET). There was no statistically significant difference in survival rates between `User` and `4-Change`.

6.4 Recall confidence

Based on the participants' responses to `LQ1` in Table 10, Fig. 4 shows the changes in the proportions of participants who practiced writing or remembering their passwords between different parts of the study. Until the second day, smaller percentage of policy `User` participants practiced remembering their passwords compared to the other two policies. This difference disappeared, however, between the second day and the eighth day as the percentage of those practicing decreased immensely for `4-Change`—there was about 37% drop. This trend shows the Surpass users' growing confidence in recalling passwords. From responses to `LQ2` (see Fig. 5), we noticed that the proportion of participants who practiced by writing down their passwords decreased dramatically between the second and the fourth day—by the fourth day, none of the participants said that they practiced by writing down their passwords.

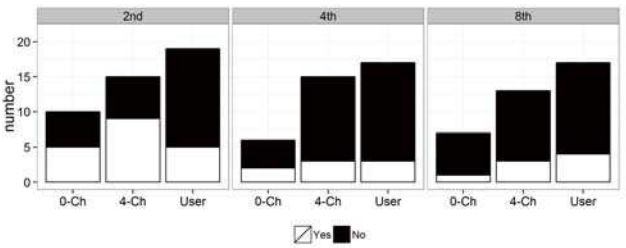


Figure 4: Responses to `LQ1` "Since the last test, did you practice writing or remembering your password?"

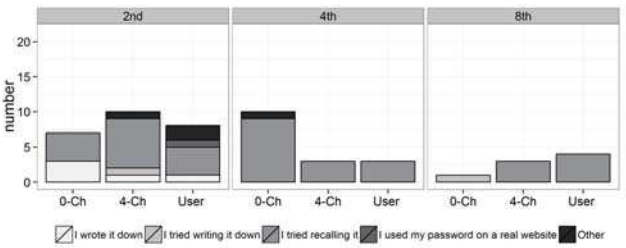


Figure 5: Responses to `LQ2` "What methods did you use to practice? Tick all that apply"

Responses to `LQ3` on the eighth day were interesting. Compared to `4-Change` (0%), a larger percentage of policy `User` (18%) participants felt that their passwords were more difficult to remember compared to the previous part of the study. In response to `LQ4`, policies `User` and `4-Change` showed similar trends in the confidence levels associated with remembering multiple passwords. On the eighth day, for `User`, about 56% responded saying they are confident in remembering up to 2 or 3 passwords, and 18% responded saying up to 4 or 5 passwords; for `4-Change`, about 67% responded saying they are confident in remembering up to 2 or 3 passwords, and 20% responded saying up to 4 or 5 passwords.

7. DISCUSSION

Our discussion of results is organized into several topics, according to the hypotheses we set up in Section 3.

7.1 Memorability improvements

The study's first hypothesis states that "the memorability of Surpass improves with the increase in the number of character replacements allowed." As apparent in Tables 2 and 3, both the first-test and second-test memorability scores improved as the allowed number of character replacements increased. However, not all Surpass policies showed statistically significant improvement over their precursor. For instance, although policy `4-Change` outperformed policies

0-Change, 1-Change, and 2-Change with statistical significance in the second test, it failed to show statistically significant superiority over policy 3-Change where the difference was just 2%. Our results do not provide enough evidence that memorability improves proportionally to the number of character replacements allowed.

Policies 3-Change and 4-Change showed statistically significant superiority over 0-Change (randomly-generated 8-character passwords) in both first- and second- test memorability. In the second test, the differences were 11% and 13%, respectively, demonstrating a significant jump in memorability. The lab study results confirmed those 4-Change observations, which, again, outperformed 0-Change in the eighth-day survival rate by 44% (see Table 11).

To confirm that such memorability improvements in Surpass are not primarily due to participants spending more time during the initial character replacement phase (see average setup time in Table 4), we looked at the average time taken for the participants to complete all three training sessions. The 0-Change participants spent 77.4 seconds on average, whereas the 3-Change and 4-Change participants spent 63.4 and 59.7 seconds, respectively. Hence, the 0-Change participants, overall, spent much more time in memorizing their passwords compared to the 3-Change and 4-Change participants. Both policies also outperformed 0-Change in terms of the authentication time, demonstrating better usability.

7.2 Security and memorability tradeoff

The second hypothesis states that "the security of Surpass weakens with the increase in the number of character replacements allowed." Our partial guessing entropy results (see Table 9) show that the guessing entropy estimates decreased as the number of character replacements allowed increased from 0 to 4. For instance, to guess α=1E-12, 226 trials, 136 trials, and 36 trials, were required for 0-Change, 1-Change, and 2-Change, respectively. Both 3-Change and 4-Change required just one trial. To guess α=5E-12, 3-Change required 8 trials whereas 4-Change required three. Hence, we can accept the second hypothesis. The password cracking results (see Table 8) show a similar trend where the percentage of cracked passwords went up from 0% in 0-Change to 0.69% in 2-Change, to 1.21% in 3-Change, and to 5.82% in 4-Change. Those results indicate that a clear tradeoff between security and memorability must be considered when choosing a Surpass policy to use.

As we discussed above, both 3-Change and 4-Change can improve memorability of randomly-generated 8-character passwords by 11% and 13%, respectively. As for 4-Change, the huge 44% difference shown in the eighth-day survival rate emphasizes the memorability gain. The tradeoff in security is the decrease in guessing entropy, and the increase in the percentage of cracked passwords for 3-Change and 4-Change. Looking at those tradeoffs, and considering that 3-Change showed no statistically significant inferiority in memorability against 4-Change, systems that already use randomly-generated passwords should consider using policy 3-Change.

7.3 Surpass vs. password complexity policy

The third hypothesis states that "A Surpass policy, that shows no statistically significant difference in memorability against the password complexity policy (user-generated passwords), has better security than the complexity policy." All of the Surpass policies outperformed the password complexity policy (User) in both guessing entropy and percentage of cracked passwords with statistical significance. In the first test, both polices 3-Change and 4-Change showed the same memorability score as User at 99%. In the second test, only 4-Change did not show statistically significant difference in memorability against User. The lab study showed a higher eighth-day survival rate for 4-Change. In all cases where memorability difference was insignificant, the Surpass policies showed statistically significant superiority in security. Surpass passwords had higher guessing entropy estimates, smaller percentage of cracked passwords, and more uniformly distributed occurrences of numbers, uppercase letters, and symbols. Hence, our results accept the third hypothesis.

Policy User showed statistically significant superiority over 4-Change in the use of external storage. However, the lab study showed that after the second day the percentage of the 4-Change participants who practiced writing or remembering dropped immensely (see Fig. 4), not showing much difference against the User participants on the eighth day.

Policy User, however, showed statistically significant superiority over 4-Change in setup time, setup difficulty, and authentication time, demonstrating better overall usability. This result was somewhat expected though, as the password complexity policy has been around for a long time and most of the participants are already familiar with it. The Surpass policies, on the other hand, are new policies that the participants had to learn and try for the first time. We believe setup time and setup difficulty can improve over time as people become more familiar with Surpass. To that end, 4-Change is a Surpass candidate to replace the password complexity policy in environments that require strong passwords for significantly improving password security with small compromise in memorability.

7.4 Consecutively repeating characters

One of the concerns for Surpass our evaluation has identified, is the frequent use of consecutively repeating characters. As shown in Table 5.1.2, the average length of the longest consecutively repeated character sequence increased with the number of character replacements allowed. About 5.8% of 4-Change passwords contained 3 or more consecutively repeating characters. Attackers can use consecutively repeating characters to crack such passwords with brute force. One way to mitigate such attacks is to use a policy that prevents users from using 3 or more consecutively repeating characters. Future work will look at testing such policies and examining their effects on the security and usability of Surpass passwords.

7.5 Remembering multiple passwords

One area of concern for Surpass is the need for users to create and remember completely different passwords across multiple accounts. However, the responses to the lab study interview question LQ4 have shown that 4-Change users are confident in remembering up to 3, 4, or 5 passwords without writing them down. Such a confidence level meets the expectations of the targeted environments (e.g., corporations), where users are often expected to manage up to 3, 4, or 5 different passwords. As for systems that already use system-generated passwords, Surpass will introduce no additional overhead in remembering multiple passwords.

8. CONCLUSIONS

We proposed a novel, system-initiated password scheme called Surpass, which allows users to replace a few characters from a given randomly-generated password to make it more memorable. We conducted a large-scale user study online, experimenting with randomly-generated 8-character passwords (policy `0-Change`), evaluating the usability and security of four different Surpass policies by increasing the number of character replacements allowed from 1 (policy `1-Change`) to 4 (policy `4-Change`).

Policies `3-Change` and `4-Change` showed statistically significant superiority over `0-Change` in memorability, improving second-day memorability by 11% and 13%, respectively. Our lab study showed a huge 44% increase in the eighth-day survival rate between `0-Change` and `4-Change`. As a result of those memorability gains, the percentage of cracked passwords went up from 0% in `0-Change` to 1.21% in `3-Change`, and to 5.82% in `4-Change`, which were all statistically significant. The guessing entropy estimates were lower for those two Surpass policies as well. No statistically significant difference in memorability was found between those two Surpass policies though. Hence, our recommendation is to consider using the Surpass policy `3-Change` if a randomly-generated, 8-character password option is being considered.

Policy `4-Change` did not show statistically significant inferiority in memorability against a real-world password complexity policy. In fact, our lab study showed a higher eighth-day survival rate for `4-Change`, and a quickly growing password recall confidence level among `4-Change` users. Surpass policy `4-Change` passwords had higher guessing entropy and showed statistically significant superiority in the percentage of cracked passwords—21%. Our second recommendation is to prefer the Surpass policy `4-Change` over the password complexity policy in environments that require strong passwords.

Our two studies focused on short-term memorability, asking participants to recall passwords after several days. In future work, we plan to conduct another user study to investigate long-term memorability of Surpass by experimenting with longer password recall periods (e.g., several months). We also plan to design password cracking rules specifically tailored to the characteristics of Surpass passwords and further test its security.

9. ACKNOWLEDGEMENTS

The evaluation part of this work was partly supported by NRFK (No. 2014R1A1A1003707) and ITRC (IITP-2015-H8501-15-1008, IITP-2015-R0992-15-1006) studentships. Authors would like to thank Alexander De Luca for shepherding the paper, and all the anonymous reviewers for their valuable feedback.

10. REFERENCES

[1] Amazon Mechanical Turk. `https://www.mturk.com/mturk/welcome`.

[2] R. Atkinson and R. Shiffrin. Human memory: A proposed system and its control processes. volume 2 of *Psychology of Learning and Motivation*. Academic Press, 1968.

[3] S. Bhuyan. *Evaluating the Usability of System-Generated and User-Generated Passwords of Approximately Minimum Equal Security*. PhD thesis, Clemson University, 2011.

[4] M. Bishop. Password management. In *Compcon Spring '91. Digest of Papers*, pages 167–169, Feb 1991.

[5] J. Bonneau. The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords. In *Proceedings of the 33rd IEEE Symposium on Security and Privacy*, 2012.

[6] J. Bonneau and S. Schechter. Towards Reliable Storage of 56-bit Secrets in Human Memory. In *Proceedings of the 23rd USENIX Conference on Security Symposium*, 2014.

[7] M. Ciampa. A comparison of password feedback mechanisms and their impact on password entropy. *Information Management & Computer Security*, 21(5):344–359, 2013.

[8] S. Fahl, M. Harbach, Y. Acar, and M. Smith. On the ecological validity of a password study. In *Proceedings of the 9th Symposium on Usable Privacy and Security*, 2013.

[9] A. Forget, S. Chiasson, P. C. van Oorschot, and R. Biddle. Improving Text Passwords Through Persuasion. In *Proceedings of the 4th Symposium on Usable Privacy and Security*, 2008.

[10] D. Goodin. Anatomy of a hack: How crackers ransack passwords like "qeadzcwrsfxv1331". `http://arstechnica.com/security/2013/05/how-crackers-make-minced-meat-out-of-your-passwords/`, May 2013.

[11] hashcat. Rule-based Attack. `http://hashcat.net/wiki/doku.php?id=rule_based_attack`.

[12] J. H. Huh, H. Kim, R. B. Bobba, M. N. Bashir, and K. Beznosov. On the Memorability of System-generated PINs: Can Chunking Help? In *Proceedings of the 11th Symposium On Usable Privacy and Security*, 2015.

[13] M. Keith, B. Shao, and P. J. Steinbart. The usability of passphrases for authentication: An empirical field study. *International Journal of Human-Computer Studies*, 65:17–28, 2007.

[14] Kelley, Patrick Gage and Komanduri, Saranga and Mazurek, Michelle L. and Shay, Richard and Vidas, Timothy and Bauer, Lujo and Christin, Nicolas and Cranor, Lorrie Faith and Lopez, Julio. Guess Again (and Again and Again): Measuring Password Strength by Simulating Password-Cracking Algorithms. In *Proceedings of the 33rd IEEE Symposium on Security and Privacy*, 2012.

[15] S. Komanduri, R. Shay, P. G. Kelley, M. L. Mazurek, L. Bauer, N. Christin, L. F. Cranor, and S. Egelman. Of passwords and people: measuring the effect of password-composition policies. In *Proceedings of the 29th SIGCHI Conference on Human Factors in Computing Systems*, 2011.

[16] R. Shay, P. G. Kelley, S. Komanduri, M. L. Mazurek, B. Ur, T. Vidas, L. Bauer, N. Christin, and L. F. Cranor. Correct Horse Battery Staple: Exploring the Usability of System-assigned Passphrases. In *Proceedings of the 8th Symposium on Usable Privacy and Security*, 2012.

[17] R. Shay, S. Komanduri, A. L. Durity, P. S. Huh, M. L. Mazurek, S. M. Segreti, B. Ur, L. Bauer, N. Christin, and L. F. Cranor. Can Long Passwords Be Secure and Usable? In *Proceedings of the 33rd SIGCHI Conference on Human Factors in Computing Systems*, 2014.

[18] 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 6th Symposium on Usable Privacy and Security*, 2010.

[19] B. Ur, P. G. Kelley, S. Komanduri, J. Lee, M. Maass, M. L. Mazurek, T. Passaro, R. Shay, T. Vidas, L. Bauer, N. Christin, and L. F. Cranor. How Does Your Password Measure Up? The Effect of Strength Meters on Password Creation. In *Proceedings of the 21st USENIX Conference on Security Symposium*, 2012.

[20] J. Yan, A. Blackwell, R. Anderson, and A. Grant. Password memorability and security: empirical results. *IEEE Security and Privacy*, 2(5):25–31, 2004.