GuidedPass: Helping Users to Create Strong and Memorable Passwords

Simon S. Woo[1] ⋆ and Jelena Mirkovic[2]

[1] The State University of New York, Korea (SUNY-Korea)
Incheon, S. Korea
simon.woo@sunykorea.ac.kr
[2] USC-Information Sciences Institute
Marina del Rey, CA, USA
mirkovic@isi.edu

Abstract. Password meters and policies are currently the only tools helping users to create stronger passwords. However, such tools often do not provide consistent or useful feedback to users, and their suggestions may decrease memorability of resulting passwords. Passwords that are difficult to remember promote bad practices, such as writing them down or password reuse, thus stronger passwords do not necessarily improve authentication security. In this work, we propose *GuidedPass* – a system that suggests real-time password modifications to users, which preserve the password's semantic structure, while increasing password strength. Our suggestions are based on structural and semantic patterns mined from successfully recalled and strong passwords in several IRB-approved user studies [30]. We compare our approach to password creation with creation under NIST [12] policy, Ur et al.[26] guidance, and zxcvbn password-meter. We show that GuidedPass outperforms competing approaches both in password strength and in recall performance.

Keywords: Password, Usable Security, Password Meter, Authentication

1 Introduction

Left to their own devices, users create passwords, which may be weak but which are memorable. Current systems attempt to improve this practice in two ways. First, systems can suggest or enforce specific password composition policies, which lead to stronger passwords. But stringent password composition requirements increase users' frustration and lead them to write down or reuse their passwords [16], which is a bad practice. It has also been shown that password composition policies are not consistent across different sites [17, 9, 20, 27], which indicates lack of clear understanding of the role that password composition plays in determining password strength. NIST recently proposed a new password composition policy [12], which enforces minimum of 8 characters and requires systems to reject inputs that appear on the list of previously-leaked passwords or common dictionary words.

⋆ Corresponding Author

Another way to improve password strength is to offer real-time feedback on the user's password and, optionally, suggestions for improvements. Password meters offer real-time feedback on user password strength [17, 29], although this feedback may be inconsistent [17, 9, 20, 27]. Password meters, however, only provide strength feedback in form of a number or color scale, but do not offer guidance on how to modify the user input into a stronger one, while preserving memorability.

A data-driven password meter by Ur et al. [26], provides proactive and actionable suggestions to a user on how to make their password stronger. This approach, however, focuses only on improving strength and does not consider how the proposed modifications may impair memorability.

We propose *GuidedPass* – a system, which helps users create both memorable and secure passwords at creation time, through detailed suggestions for improvement of password structure and semantics. First, we start from an observation that memorability stems both from the choice of words used in the password (e.g., phrases, names, numbers, dates of personal significance) and the password structure (e.g., word, followed by digits). We then tailor our suggestions in such a way to preserve the initial user-supplied strings and structure, as much as possible, while improving password strength. Our main contributions are summarized below:

1) **Identification of semantic patterns, which make passwords memorable and strong:** We analyze 3,260 passwords, which were successfully recalled by participants in our prior IRB-approved user studies [30], to identify semantic patterns that make these passwords both memorable, and strong. We call these *preferred patterns*.

2) **Design of a real-time password suggestion system (*GuidedPass*):** We design a system, which chooses a set of preferred patterns, which are closest to the user's input, and provides meaningful suggestions for gentle structural and semantic modification of the user's initial input.

3) **GuidedPass evaluation:** We evaluate GuidedPass and several competing approaches in a user study, with more than 1,400 Amazon Mechanical Turk participants. We show that passwords with GuidedPass suggestions are both more memorable and stronger than passwords created by competing approaches. GuidedPass achieves 81% recall after two days, and the average strength of more than 10^{19} statistical guesses.Compared to the approqch by Ur et. al [26], GuidedPass has 14% higher recall, and up to 100 times higher strength.

The rest of this paper is organized as follows. We discuss related work in Section 2. We present our methodology in Section 3. Memorable password dataset is analyzed in Section 4. We present the GuidedPass system design in Section 5. We detail the setup of our user study in Section 6. Section 7 presents the results of our evaluation of GuidedPass, and competing approaches, and Section 8 offer our discussion and conclusions.

2 Background and Related Work

We provide a brief overview of related research, which aims to help users create strong passwords.

Password composition policies are regularly used to steer users toward stronger passwords. The most common password policy is the 3class8 policy, which requires a

password to be at least 8 characters long, and to include at least three out of four character classes: digits, uppercase and lowercase letters, and special characters. There are many inconsistencies among password policies [24, 10], and a lack of clear understanding on which policy is the best.

Even when users meet the 3class8 policy requirements, their passwords can still be weak because they are created using common words and phrases. For example `Password123` satisfies 3class8 requirement but is among top 33,523 out of 14,344,391 passwords and occurs 62 times in leaked `RockYou` datasets. In [18], Kelly et al. found that passwords created under minimum 8 characters policy are significantly weaker than passwords created under stricter policies. Shay et al. [21] compared eight different password composition policies and found that a long password with fewer constraints can be more usable and stronger than a short password with more constraints. Overly strict password composition policies may also lead to unsafe practices, such as writing down passwords [23]. NIST [12] recently proposed a new password composition policy, which removes requirements for different character classes, but keeps the length requirement. The system is also required to check users' passwords against any previously leaked passwords, and against common dictionary words. While this feedback informs the users on what parts of their password may be susceptible to a guessing attack, it does not provide clear guidance on how to build a better password. Such guidance is needed, to help users make significant improvements to their password strength, instead of small, predictable changes [14].

Password complexity does not necessarily mean low recall. Bonneau and Schechter [4] show that users can be trained to remember randomly-assigned 56 bit codes, but such training is hardly practical for tens of passwords accounts, which users need daily [15].

Users can be helped to create strong passwords by using a password meter [9, 22, 27], or a composition of password meters and password composition policy [20, 23]. Meters, however, are not enough. They are inconsistent in strength estimation [7], and they do not offer specific suggestions on how to modify passwords to improve their strength.

Telepathwords [19] provide proactive suggestions to users during password creation. The system learns character distributions in its existing password data, and uses it to highlight frequent character patterns in user input. Users are thus steered towards less likely patterns. Telepathwords' increase password strength by 3.7 bits of zxcvbn [29] entropy measure, but recall declines to 62% of the baseline, because users are steered from words that are meaningful to them towards those with lower personal significance. GuidedPass addresses this problem, by allowing users to keep their current inputs, and gently morph them into stronger passwords. While we did not compare memorability of GuidedPass passwords to that of original user inputs, GuidedPass achieves 81% recall after two days, compared to 62% for Telepathwords.

The most related work to GuidedPass is the data-driven password meter by Ur et al. [26] – DataPass for brevity. DataPass provides real-time, specific guidance to users on how to improve their passwords. It also identifies a range of inputs that should be avoided such as dictionary words, common passwords, etc. The main point of difference between GuidedPass and DataPass is in how password suggestions are developed. DataPass mines weak password patterns from leaked password datasets, but it has no way of learning which passwords are memorable to their users. Conversely, we use a labeled dataset of

passwords from our prior studies [30] to learn which patterns appear much more often among memorable and strong passwords, than among other subsets. This enables us to make suggestions that both improve strength and preserve memorability. In Section 7, we provide side-by-side comparison of suggestions generated from GuidedPass and DataPass, and point their differences. GuidedPass outperforms DataPass both in password recall and in password strength.

3 Methodology

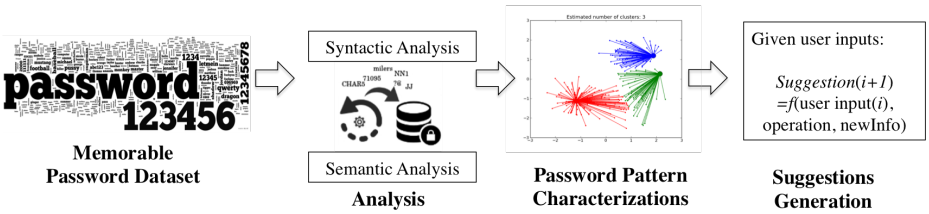

Fig. 1: The overall development process of the password suggestion system (GuidedPass)

Our process for the GuidedPass development is illustrated in Fig 1. We start with the observation that it is necessary to analyze passwords that are both memorable and strong, to learn about their structure and semantics. This cannot be accomplished by analyzing leaked datasets, since these datasets lack recall information. Over three years, we have collected passwords for various authentication research. These passwords were created during our studies, and were successfully recalled, in the course of the study, after two days. The dataset includes more than 3,200 passwords.

We leverage these successfully recalled passwords to understand general patterns, which also make these passwords strong. We measure the strength of memorable passwords, using the Monte Carlo method by Dell'Amico, and Filippone [8]. We train the guessing algorithm with a total of 21 million leaked passwords. Based on the estimated passwords' strength, we classified each password into the weak, medium or strong category, using the estimated number of guesses for online (10^6) and offline (10^{14}) attacks as boundaries between categories [11].

After classifying memorable passwords into three different strength groups, we perform both *syntactic* analysis – such as recording password length and composition – and *semantic* analysis – such as understanding if password segments are dictionary words, personal names, etc. For semantic analysis, we used Vera et al.'s semantic segmentation parser [28] to segment the password and label each segment. We then compare and analyze the syntactic and semantic structures between groups, and identify patterns that occur predominantly in the strong category. We call these the *preferred patterns*.

Next, out of the preferred patterns we generate suggestions to users, which are easy to understand and simple to follow, on how to evolve their initial input into a strong

password. We present several suggestions so that a user can choose the one they prefer, and which may have the least impact on password recall. We also strive to keep our suggestions "fuzzy" and not too specific, to increase search space for attackers who are familiar with GuidedPass. Our suggestion process can also be iterative – suggestions can continue until the user's password exceeds some desired strength.

4 Memorable Password Analysis

Using the Monte Carlo method by Dell'Amico and Filippone [8], we classified each memorable password in our dataset into the weak (fewer than 10^6 guesses), medium or strong (more than 10^{14} guesses) category. Among our memorable passwords, almost 27% of passwords fell into the strong category, 70% into the medium category and 3% into the weak category, as shown in Table 1.

Table 1: Memorable password dataset, categorized into three different strength groups and percentage of 3class8 passwords in each strength group

Strength Category	**No. of Passwords (%)**	**Perc. of 3class8 Passwords**
Weak (guesses $< 10^6$)	109 (3.34%)	6%
Medium ($10^6 \leq$ guesses $< 10^{14}$)	2,276 (69.82%)	58.1%
Strong (guesses $\geq 10^{14}$)	875 (26.84%)	74.2%
Total	3,260 (100%)	60.68%

4.1 Syntactic Characteristics

We first analyze the passwords in each category with respect to length, number of character classes, and class changes. We summarize our findings below.

3class8 policy neither necessary nor sufficient: We show the percentage of password that meet the 3class8 requirement in each category in Table 1. 74.2% of strong passwords, 58.1% of medium-strength passwords and 6% of weak passwords meet the 3class8 requirement. This clearly shows that 3class8 requirement is neither necessary (25.8% of strong passwords do not meet it) nor sufficient (significant number of medium-strength and weak passwords meet it) for a strong password.

Password length makes a big difference: Password length plays a critical role in determining password strength [15]. The average password length in the weak, medium, and strong group was 8.83, 9.88, and 13.73 characters, respectively. The length distribution was significantly different across strong, median, and weak strength groups (KW test $p = 9.87 \times 10^{-151}$), while the difference is smaller but still significant between weak and medium groups (Holm-Bonferonni-corrected Mann-Whitney U, HC-MWU, test, $p = 2.31 \times 10^{-5}$). The statistical difference between medium and strong group is significant (HC-MWU test, $p = 3.11 \times 10^{-142}$). Hence, stronger passwords tend to be longer.

Table 2: Average and STD (stdev) of number of symbols, digits, uppercase letters, and number of class changes for passwords in each strength category

Strength	**Symbols**		**Digits**		**Uppercase-letter**		**No. of Class ch.**	
Category	Average	STD	Average	STD	Average	STD	Average	STD
Weak	2.3	1.8	0.1	0.4	0.02	0.1	1.0	0.9
Medium	2.6	1.6	0.7	0.7	0.2	0.5	1.8	0.9
Strong	2.6	1.9	1.1	1.0	0.6	0.9	2.8	2.1

Digits and uppercase letters improve strength: We show the number of symbols, digits, and uppercase letters in Table 2. All strength groups have similar statistics for the number of symbols and there is no statistical difference between them. However, there is significant statistical difference with regard to the number of digits present in weak, medium, and strong passwords (KW test, $p = 2.66 \times 10^{-43}$), with stronger passwords having slightly higher incidence of digits. The statistical significance between strong and medium group with HC-MWU test is $p = 3.68 \times 10^{-18}$. And HC-MWU test yields $p = 5.96 \times 10^{-24}$ between medium and weak group. Similarly, stronger passwords also have a higher incidence of uppercase letters (KW test, $p = 1.96 \times 10^{-55}$). The statistical significance between strong and medium group with HC-MWU test is $p = 1.18 \times 10^{-48}$. And HC-MWU test yields $p = 3.73 \times 10^{-4}$ between medium and weak group.

More class changes improve strength: We define a class change as having two consecutive characters in a password from different character classes. For example, "Alicebob123$" has 3 class changes ('A' $\rightarrow$ 'l', 'b' $\rightarrow$ '1', '3' $\rightarrow$ '$'). A higher number of class changes can create more complex, and possibly stronger passwords. Statistics for the number of class changes are shown in columns 8 and 9 of Table 2. As password strength increased so did the number of class changes (KW test, $p = 1.87 \times 10^{-73}$). The statistical significance between strong and medium group with HC-MWU test was $p = 4.58 \times 10^{-47}$. And HC-MWU test yields $p = 1.62 \times 10^{-27}$ between medium and weak group.

4.2 Semantic Structure

Next, we analyze the semantic structure of strong, medium, and weak passwords. We use Vera et al.'s semantic parser [28] to segment each password and label the segments with their part-of-speech (POS tags) from CLAWS7 tagset [25]. For example, for a string "applerun" the string would return segments (apple)(run) and tags (nn1)(vv0) indicating a singular noun and a base form of a verb. This representation captures the underlying semantic structures of passwords, which cannot be represented by the previously discussed syntactic features. We further label segments as **(dict):** dictionary words,**(fname):** popular female names, **(mname):** popular male names, and **(sname):** popular last names from 2010 US Census [1]. Also, we separately check if passwords match with leaked passwords, as suggested by others [29, 14, 26]. We use leaked passwords from Xato corpus [5] and label user inputs found in the corpus as **leak**. Such a label is shown to user to alert them not to use the leaked password.

Complex and unique patterns improve strength: After processing each password with the semantic segmentation program, we count the total number of unique tag-sequences in each group and compute the percentage of those. We find that 47.7% of weak, and 51.3% of medium passwords have unique semantic patterns, while 91% of strong passwords have unique patterns. This uniqueness in semantic patterns may contribute to password strength.

Table 3. presents the top 10 most frequently used semantic patterns for each group with the percentage of each tag occurrence. If we compare two tables, we can clearly observe that a few digits followed by a noun (e.g. `(dict)(number1)`, `(mname)(number4)`) are the most commonly used semantic pattern in weak and medium strength group. Further, there are many occurrences of either (dict) or (name) tags in weak and medium groups, in addition to one other tag. On the other hand, semantic patterns of strong passwords are more complex and diverse, as shown in Table 3. Although these passwords also use dictionary words and names, those are interleaved with complex symbol and digit sequences, resulting in non-common words and structures (e.g., KpieAT7894#). Therefore, we should guide users towards more complex semantic patterns to improve password strength.

Table 3: Top 10 most frequent semantic patterns from different strength groups

Weak	%	**Medium**	%	**Strong**	%
(dict)(number1)	**11.01**	**(dict)(number4)**	**2.48**	**(char1)(dict)(char2)(number4)(special1)**	**1.03**
(fname)(number4)	7.34	(dict)(number3)	2.34	(char1)(ppis1)(number1)(special1)(number4)(char1)(ppis1)	1.03
(dict)(number2)	6.42	(mname)(number4)	1.47	(special1)(at)(dict)(jj)(number4)(special1)	0.91
(mname)(number4)	6.42	(dict)(number1)(special1)	1.23	(char1)(dict)(char2)(number2)(special1)	0.8
(fname)(number2)	6.42	(fname)(number4)	1.06	(char4)(number2)(special2)	0.57
(dict)(number3)	4.59	(dict)	0.83	(char6)(number3)(special1)	0.46
(number8)	3.67	(dict)(number3)(special1)	0.78	(dict)(number2)(special1)	0.34
(dict)(number4)	3.67	(dict)(special1)(number4)	0.78	(special1)(dict)(dict)(number2)	0.34
(mname)(number2)	3.67	(dict)(special1)(number2)	0.73	(number4)(char1)(special1)(dict)(special1)(char2)	0.34
(dict)	2.75	(dict)(number2)	0.69	(mname)(sname)(number2)(special1)	0.23

There were 19.27% of weak passwords, which were fully matched with a leaked password, and 54.1% of weak passwords used a leaked password segment (e.g., 'password9cq'). Further, medium-strength passwords had no full matches but 33.2% of them contained a leaked password, in addition to other characters. On the other hand, 20.5% of strong passwords contained leaked password segments but none of them fully matched with leaked passwords.

The more segments, the higher strength: We investigate the number of different-tag segments in a password, which correlate with its semantic complexity. We show the empirical PDF of the number of semantic segments in Fig. 2. Weak passwords have only 2.21 segments on the average, while the medium-strength passwords have 3.44 segments, and the strong passwords have on average 5.22 segments. Thus, we should guide users toward more semantic segments to improve their password strength.

4.3 Summary Of Our Findings

We summarize our recommendations as follows:

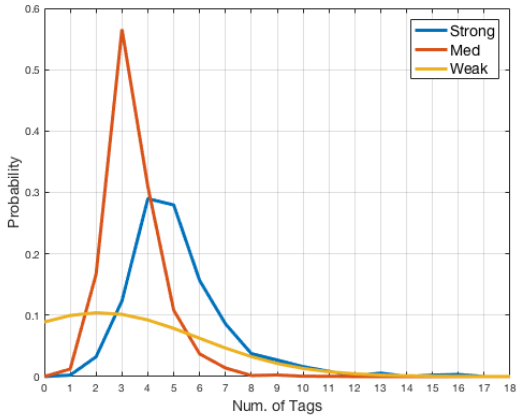

Fig. 2: Empirical PDF of the number of semantic segments

– **Uncommon or non-dictionary words.** Even with the same semantic pattern, e.g., `(np1)(number4)`, a password can be in any of the three strength categories, depending on the commonality of the words in each segment. For example, **bella1234** is in weak, **Alaska2011** is in medium, and **u.s.-iraq6911** is in strong group with the same `(np1)(number4)` structure) Thus we must steer the users towards uncommon words. Creating uncommon words may not be that hard. For example, we observe that strong passwords often consist of a dictionary word, interleaved with digits or symbols, or being intentionally misspelled.
– **The longer, the more semantic segments, and the stronger** Our suggestions often involve addition of more words into the password to make it longer and thus stronger. We also suggest insertion of different character classes to increase both the number of class changes and to create uncommon segments from common ones.
– **Multilingual passwords.** We observe that some strong passwords include words from foreign languages such as Spanish or Arabic. Research [2] has shown that more than half of population on Earth are bilingual. We expect that combining words from more than one language in unpredictable ways can improve password strength without loss of memorability.

5 GuidedPass System Design

In this section,we describe how we designed and implemented suggestions in GuidedPass, using the password suggestion model and templates.

5.1 Password Suggestion Model

We assume that users initially choose passwords based on certain strings that have personal significance to them, which makes them memorable. Then, our suggestions are

generated to evolve and extend user's existing password into a stronger version without losing memorability. We formally define the password suggestion model as follows:

$$Password_{new} = f(Password_{current}, M_{new}), \tag{1}$$

where $Password_{current}$ is the user's current password string, M_{new} are the new words or characters to be added to $Password_{current}$, and f is a function that the user performs to integrate M_{new} with $Password_{current}$. We focus on functions that an average person could easily perform, inspired by Blum et al. [3]. These are addition, insertion, replacement without deletion, swapping, breaking, or perturbing sequence and redistributing, separating, or moving segments as shown in Table 4. We do not suggest deletion, since it reduces password length. Next, we consider types of new information, M_{new}, the user can enter. As we discussed from the previous section, for strong passwords, M_{new} should be chosen from uncommon words. Users can also create uncommon strings or break up common words or sequence structures, by interleaving them with digits or symbols.

Table 4: Example of `<Action,Info,Quantifier>` used in suggestion generation

Action (Operation)	**Information**	**Quantifier (Fuzzy terms)**
add, insert, replace, swap	(un)common name	some, a few
brake, move, perturb	(un)common word	somewhere
redistribute, separate,	word(s), digit(s), symbol(s), sequence(s)	in the middle

With these options, we construct the `<Action, Info, Quantifier>` templates, as shown in Table 4. Suggestions can be constructed from any combination of action, information, and quantifier, based on the user's current input. We provide multiple suggestions to the user, and they can choose the most suitable suggestion in each step to extend their password. Our suggestions are intentionally designed to be high-level and non-specific. First, we want to allow sufficient space and flexibility for users to interpret these suggestions in a way that does not interfere with password memorability. Second, we want to increase the search space of guessing attacks. If suggestions were too specific, it would be easier for attackers to perform rule-based attacks.

5.2 Suggestion Rules

To be able to provide suggestions in real time, we first need to detect semantic content and patterns of a user-entered password in real time. Using our POS segmentation [28] and the zxcvbn [29] tool, we can detect dictionary words, names, common sequences, and blacklisted passwords. Upon detecting problematic content or patterns such as leaked passwords, and common first name, we immediately highlight them and generate targeted suggestions to avoid those. Following summarizes the suggestions we generate for each case:

– **Common word, name, sequence or dictionary word:** Upon detecting a dictionary word, a common sequence, a personal name [1] or a leaked password we generate suggestions to: add uncommon personal name, a non-dictionary word, or insert

Table 5: A side-by-side comparison of generated suggestions between ours and Ur et al.

User Input	Category	GuidedPass	Ur et al.[26]
John	Top 1K popular names	1. Add an uncommon name 2. Add a few numbers or symbols in the middle of the name	1. Contain 8+ characters 2. Not be an extremely common password
Password123	Leaked top 50K passwords	1. Add an uncommon word 2. Add a few numbers or symbols in the middle of a word	1. Not be an extremely common password
12345	Sequence	1. Perturb the sequence or separate into a few segments	1. Contain 8+ characters 2. Not be an extremely common password
aabbccaabbcc	Repeating pattern	1. Add an uncommon word 2. Move a few numbers or symbols to the middle of the pattern to break repeating pattern	1. Don't use words used on Wikipedia (ccaa) 2. Avoid repeating sections (aabbcc) 3. Have more variety than repeating the same 3 characters (a, b and c)
defense	Popular dictionary word	1. Add an uncommon word 2. Add a few numbers or symbols in the middle of a word	1. Don't use dictionary words (defense)
6122017	Date	1. Perturb the sequence or separate into a few segments	1. Avoid using dates like 6122017
defense6122017	Simple structure	1. Add one of the following: uncommon word, uncommon name, or mix of symbols	1. Consider inserting digits into the middle, not just at the end

symbols/digits to modify the common/leaked segment into an uncommon one. We provide the examples in Table 5.

– **Simple structure pattern:** If the user's password is too simple and its structure is too predictable such as `(np)(digit)` as shown in Table 5, we suggest to the user to add one of the following: uncommon word, uncommon name, or mix of symbols to make a password into a more complex structure.

In Table 5, we show how GuidedPass and DataPass [26] generate suggestions for the same user inputs. This provides a side-by-side comparison to measure similarity and difference between these two approaches. Both approaches do well in detecting problematic or weak patterns, and generate suggestions based on those. However, GuidedPass provides more direct actions for users to perform such as "*Add* uncommon name" or "*Add* a few numbers or symbols in the middle of the name" to avoid detected patterns. Conversely, DataPass focuses more on highlighting syntactic features of passwords, which are not desired, instead of guiding users towards desirable inputs.

6 Experiment

We now describe user studies we employed to evaluate benefits of GuidedPass and compare it to competing approaches. All user studies were reviewed and approved by our Institutional Review Board (IRB). We recruited participants among Amazon Mechanical Turk workers.

6.1 Approaches

Our evaluation focus was to measure strength and recall of passwords created with GuidedPass and other competing approaches. We did not suggest any specific password policy to users, unless required by an approach we evaluate. First, as much research has shown, password policies are inconsistent, confusing, and do not necessarily help users to create strong passwords, but they increase user burden. Second, it is difficult to isolate benefits of a password suggestion system in the presence of policy. Instead, for user feedback, we employ the zxcvbn meter's visual progress bar to display the current password's strength to users. As Crawford et al. [6] found, visual feedback on users' progress can reduce the perception of the online users' task burden, and they can complete the task.

Table 6: Password creation approaches

Approach	Description
GuidedPass	Our approach with detailed textual suggestions with strength enforcement
GuidedPass-NE	GuidedPass with no strength enforcement
CMU-NE	Ur et al.' [26] textual suggestions with no strength enforcement
zxcvbn	zxcvbn meter [29] with strength enforcement
zxcvbn-NE	zxcvbn meter [29] with no strength enforcement
NewNIST	New NIST Proposal (800-63) [13] (minimum 8 characters and blacklist password enforcement)
3class8	3class8 creation policy (min. 8 characters with at least 3 classes from lowercase-letters, uppercase-letters, symbols, and digits.)

The descriptions of all evaluated approaches are summarized in Table 6. Our baseline model is GuidedPass with no strength enforcement (**GuidedPass-NE**). In this approach, detailed semantic suggestions with visual bar are presented to a user, but the user is not required to meet any strength requirement and may choose not to follow our suggestions. We compare this model to DataPass [26] (**CMU-NE**), with no strength enforcement. We use the same meter – zxcvbn – in both approaches, to isolate the impact of the approaches' suggestions. We also compare our GuidedPass-NE to a meter-only approach, without strength enforcement (**zxcvbn-NE**). This comparison helps us highlight impact of our suggestions on the resulting passwords. We also compare GuidedPass to the new NIST password creation policy (**NewNIST**), using zxcvbn meter and no strength enforcement. For completeness, we also compare GuidedPass with the passwords created under the popular 3class8 password composition policy (**3class8**). The only two approaches where users are required to meet password policy were NewNIST and 3class8.

We also investigate the impact of combining suggestions and meters with enforcement of some target password strength. In this set of approaches users must continue password creation until the resulting password's strength meets or exceeds the target. We require that each password's strength must meet or exceed zxcvbn score of 5, which is equivalent to a password that cannot be guessed in 10^{12} guesses. We investigate two approaches with strength enforcement: **GuidedPass** and **zxcvbn**.

6.2 User Study Design

In the user study, each participant was assigned at random to one approach for password creation. We recruited participants with at least 1,000 completed Human Intelligence Tasks (HITs) and $> 95\%$ HIT acceptance rate. We asked each participant to create one password for an imaginary server. After two days each participant was invited to return to the study and attempt to authenticate with their password. We paid 35 cents for password creation and 40 cents for the authentication task, respectively.

Authentication. Each user was asked to authenticate two days after password creation, allowing at most five trials per password and per visit. All users were asked not to paste their answers. We had automated detection of copy or paste attempts in our login forms, and we rejected the users who were detected to perform either of these two actions. We further displayed a notice to participants, at both the creation and authentication screens, that they will receive the same payment, regardless of their authentication success. This ensured that participants had no monetary incentive to cheat. At the end of the authentication visit, we asked participants to complete a short survey to asses their sentiment about usability of each password creation approach.

6.3 Limitations and Ecological Validity

Our study had the following limitations, many of which are common for online password studies. First, it is possible but very unlikely that a participant may enroll into our study more than once. While the same Mechanical Turk user could not enter the study twice (as identified by her Mechanical Turk ID), it is possible for someone to create multiple Mechanical Turk accounts. There is currently no way to identify such participants.

Second, we cannot be sure that our participants did not write down or photograph their passwords. We did not ask the participants if they have done this in post-survey, because we believed that those participants who cheated would also be likely to not admit it. We designed our study to discourage cheating. We promised to pay participants in full regardless of authentication success. Our study mechanisms further detected copy/paste actions and we have excluded any participant that used these (for whatever reason) from the study. We also reminded the participants multiple times to rely on their memory only. If any cheating occurred it was likely to affect all the results uniformly. Thus our data can still be used to study improvement of recall and security between password creation approaches.

Third, while we asked Mechanical Turkers to pretend that they were creating passwords for a real server, they may not have been very motivated or focused. This makes it likely that actual recall of real-world passwords would be higher across all creation approaches. While it would have been preferable to conduct our studies in the lab, the cost would be too high (for us) to afford as large participation as we had through the use of Mechanical Turks.

7 Results

In this section, we present the results of our user study. First, we provide the demographic information, the password strength and recall, and time to create passwords. Then, we analyze suggestions generated, and adopted by users.

7.1 Participant Statistics

In total, there were 1,438 participants that created passwords. Two days after creation, we sent an email to all of them to return for authentication. Out of 1,438 participants, 990 participants returned (return rate 68.85%), as shown in Table 7. Among 1,438 participants, 52% reported being male and 47% reported being female. Also, 83% reported that their native language were English. With regard to the age range, most participants were in 25-34 age group (52%), followed by 35-44 (29%) and 45-54 (12%) age groups. We found no statistically significant difference in any of our metrics between participants of different age, gender or with different native language.

Table 7: Total number of participants who created and authenticated with their passwords

Approach	**Created**	**Auth. after 2 days**
GuidedPass	218	150
GuidedPass-NE	207	148
CMU-NE	180	119
zxcvbn	204	142
zxcvbn-NE	203	127
NewNIST	219	162
3class8	207	142
Total	**1,438**	**990 (68.85%)**

7.2 Password Statistics

We show the average length, median, and standard dev. of each password created under different approach in Table 8. GuidedPass, GuidedPass-NE and zxcvbn produced the longest passwords, with the average of 13.0–13.9 characters. The GuidedPass-NE approach helped users create longer passwords, even without enforcing the strength requirement. On the other hand, users created the longest password under zxcvbn with strength enforcement. The CMU-NE and zxcvbn-NE models resulted in slightly shorter passwords (11.9–12.2 characters), while the NewNIST and 3class8 approach had the shortest passwords – 10.7 characters.

7.3 Recall Performance

We asked users to authenticate 2 days after password creation. Recall was successful if the user correctly inputted every character in the password, in the right order. Table 8. shows the overall recall performance.

GuidedPass-NE and GuidedPass are highly memorable. GuidedPass-NE and GuidedPass were the top two approaches, yielding the highest recall rates. As shown in Table 5, GuidedPass-NE achieved greater than 81% recall rate, around 9% higher than CMU-NE, the most closely related competing approach. This result demonstrates that

Table 8: Password Length Statistics and Successful Recall Performance

Measure	Length			Successful Recall Rate
Approach	Avg.	Median	STD	
GuidedPass-NE	**13**	13	2.9	**81.08%**
CMU-NE	12.2	12	3.3	**71.43%**
zxcvbn-NE	11.9	11	4.0	70.78%
NewNIST	10.7	10	3.5	67.28%
GuidedPass	**13.5**	13	3.0	**72.67%**
zxcvbn	**13.9**	13	3.3	55.63%
3class8	10.7	10	3.1	64.08%

more semantically meaningful and intuitive suggestions provided by GuidedPass-NE helped users create more memorable passwords from their initial inputs.

Approaches that offered no proactive guidance or suggestions to users during password creation (zxcvbn-NE, zxcvbn, NewNIST, and 3class8) had much lower recall (up to 25%) than approaches that offered guidance (GuidedPass-NE, GuidedPass and CMU-NE). We believe that when guidance is lacking users focus too much on meeting the strength requirement, and they unwittingly sacrifice memorability. The specificity of our suggestions enabled users to create strong passwords without sacrificing memorability. Comparing the same approaches with and without strength enforcement (GuidedPass vs. GuidedPass-NE, zxcvbn vs. zxcvbn-NE), strength enforcement lowered recall by 8–15%. Therefore, approaches that only provide guidance and do not enforce strength requirement are better for recall. Instead of strict policy and strength enforcement, our work shows that better suggestions are a more effective way to guide users toward strong and memorable passwords.

7.4 Password Strength

We evaluate strength of each password collected in our study using the *guess number* measure. We use the Monte-Carlo method by Dell'Amico and Filippone [8] to obtain the guess number.We trained several password models using the Monte-Carlo method: the 2-gram, 3-gram, and the back-off model. For training the models, we used a total of 21 millions of leaked passwords from Rock You, LinkedIn, MySpace, and eHarmony. We summarize the median guess number strength in Table 9, where the minimum guess number that attackers would achieve is highlighted for each approach. We also present the guess number strength distribution using the 3-gram model and back-off model in Figs. 3 and 4, respectively. In Figs. 3 and 4, the X-axis is the logarithm of the number of guesses, and the Y-axis is the percentage of passwords being guessed. We only report the guess number up to 10^{25} due to the space limit.

GuidedPass and GuidedPass-NE are strong. GuidedPass and zxcvbn produce the strongest passwords in most measures, due to the maximum strength enforcement. Further, GuidedPass-NE outperforms CMU-NE requiring around 10 times more guesses. It is interesting to note that without strength enforcement GuidedPass-NE strength did not degrade much (around 10 times), while zxcvbn-NE strength degraded a lot

Table 9: Median guess number, measured using 2-gram, 3-gram and back-off model

Approach	2-*gram*	3-*gram*	Back-off
GuidedPass-NE	7.4E+18	**5.04E+17**	1.45E+18
CMU-NE	1.38E+18	**5.55E+16**	2.29E+17
zxcvbn-NE	3.44E+16	3.95E+15	**1.74E+15**
NewNIST	4.87E+14	8.26E+13	**6.53E+13**
GuidedPass	3.43E+19	**5.62E+18**	5.18E+19
zxcvbn	7.45E+20	**2.55E+19**	9.09E+19
3class8	8.02E+14	**9.27E+13**	1.43E+14

(around 10,000 times). Thus, user guidance helped create strong passwords even without enforcement. Finally, NewNIST and 3class8OP performed very poorly, requiring in general around 100 times fewer guesses than other approaches, and could not resist offline attacks. In fact, NewNIST did not help users create stronger passwords, and resulted in lower strength than even 3class8. We believe that removing different class requirements lowered the strength of passwords created under the NewNIST policy.

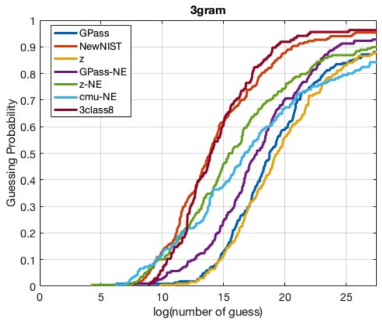

Fig. 3: Guess number and guessing probability measured using 3-gram model

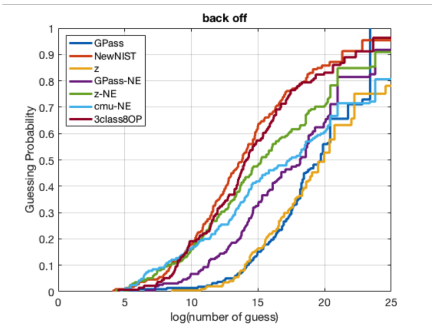

Fig. 4: Guess number and guessing probability measured using back-off model

7.5 Password Creation Time

We measured the average time needed to create a password (time between the initial and the final password input by user). The average creation times with GuidedPass-NE, CMU-NE, zxcvbn-NE, NewNIST, and 3class8 were 105, 111, 53, 62, and 40 sec, respectively. With enforcement, creation times for GuidedPass and zxcvbn were 110 and 89 sec, respectively.

The empirical PDF of time to create a password with each approach is provided in Fig. 5. The average time to create a password was up to two times higher for suggestion-based approaches (GuidedPass-NE, GuidedPass, and CMU-NE) than for those that offer no user guidance (NewNIST, zxcvbn-NE, and zxcvbn). This is expected, as users take

time to read textual feedback, and suggestions, and decide how to apply those to their password. GuidedPass-NE, GuidedPass and CMU-NE all had comparable password creation times of just under 2 minutes (105–111 seconds). Approaches that do not enforce a given target strength had the lowest password creation time (3class8OP had 40 s, zxcvbn-NE had 53 s, and New NIST had 62 s), while the zxcvbn approach, which enforced a given target strength but did not offer guidance to users took 60% longer (89 seconds instead of 53 seconds).

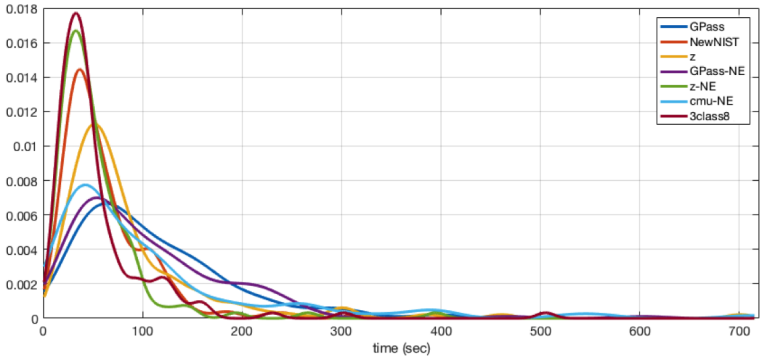

Fig. 5: Empirical PDF of time to create passwords with each approach

7.6 Suggestions Adopted by Users

In GuidedPass approach, we present all the applicable suggestions to users. This way, users have more flexibility in adopting suggestions that they feel they will be able to recall. In this section, we measured which suggestions were more frequently employed by the users. We recorded the time and users' every key stroke, including back space, and delete key, entered in the password box during the study. Then, we captured and compared the presented suggestions and those actually adopted by users.

We divided the types of suggestions into two broad categories: addition vs. structural change. The addition is a suggestion for user to add certain type of information such as chars, digits, symbols, and uncommon word in unpredictable locations, as shown in Table 4.

The other structural change is to insert information somewhere in the entered password. Also, this category includes deleting, replacing, and breaking existing structure into different segments. On average, a user adopted 4.12 suggestions. Most adopted suggestions were of the "addition" type (80.6%), followed by "structural changes" (19%). Among addition suggestions, most popular were those asking to add digits (27%) and uncommon words (25%). Among structural changes, inserting digits and symbols in the middle of an existing password or changing case were the most adopted suggestions (around 2%). Also, we detected a lot of delete key actions (8.82% of users), which indicates that users attempted to delete some part of their original passwords, and create new segments, based on our suggestions.

Next, we seek to understand how changes adopted by users help improve strength. Thus, we measured the difference in strength, using guess number, from the initial to the

final password for each given user. The initial and final strength distribution is shown in Fig. 6, where the X-axis is the log of guess number, and the Y-axis is the probability. The overall strength improvement is about 10^7 –10^{10} guesses from users' initial input to final passwords as shown in Fig. 6. We can clearly observe the improvement as users adopted the suggestions given by GuidedPass.

Suggestion Type	**Perc. (%)**
Total "Addition" suggestions	80.6%
Add chars	2.77
Add digits	27.46
Add symbols	17.63
Add uncommon words	24.94
Add words	7.81
Total "Structure Change" suggestions	19.4%
Flip Case	2.02
Insert chars	1.01
Insert digits	2.52
Insert symbols	2.52
Insert uncommon words	0.76
Insert words	1.26
Break sequence	0.25
Delete	8.82
Replace word	0.25
Total	100 %

Table 10: Overall suggestions statistics.

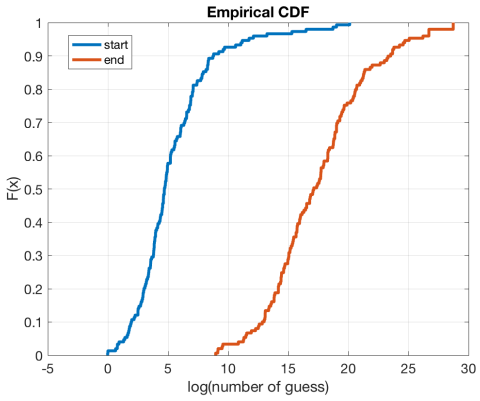

Fig. 6: ECDF of strength improvement between the initial and the final password in GuidedPass approach.

7.7 Users Sentiment

After each participant completed their authentication task, they were asked to rate their agreement with the following statement, on a Likert-scale, from 1 (strongly disagree) to 10 (strongly agree) with 5 being neutral – "the password creation was easy to use."

We present the boxplots of users' responses in Fig. 7. In all cases, the higher value on the Y-axis indicates a more favorable response, the red line is the median and edges of the box represent the 25th and 75th percentiles. The whiskers extend to the most extreme data points not considering outliers, and outliers are plotted individually as a red cross in Fig. 7.

The average Likert scores for GuidedPass-NE, CMU-NE, zxcvbn-NE, NewNIST, and 3class8 were 7.34, 7.52, 7.59, 7.35, and 6.32, respectively. The average scores with strength enforcement with GuidedPass and zxcvbn were 7.29 and 6.30. Approach zxcvbn-NE was the easiest to use with the highest user rating. However, with meter enforcement, zxcvbn had the worst rating with 6.30, since users were frustrated, trying to exceed the target password strength without clear guidance on how to do this. The pairwise corrected p-value was $p = 2.04 \times 10^{-6} \ll 0.05$ between zxcvbn and zxcvbn-NE. Similarly to zxcvbn, 3class8 policy was rated 6.30. This may be counterintuitive because users are very familiar with 3class8. We believe that lower scores were due to user frustration as they were trying to improve their password strength (indicated by the visual meter), and did not know how to achieve this. GuidedPass-NE and GuidedPass had the average of 7.29 and 7.34 ratings. However, there was no statistical difference between these ratings, with $p = 0.21$. The CMU-NE rating was slightly higher, but the

pairwise corrected $p = 0.81$ between CMU-NE and GuidedPass-NE shows that there was no significant statistical difference in rating between GuidedPass-NE and CMU-NE. Overall, suggestions based approaches seem to be well accepted by users based on the average Likert scores.

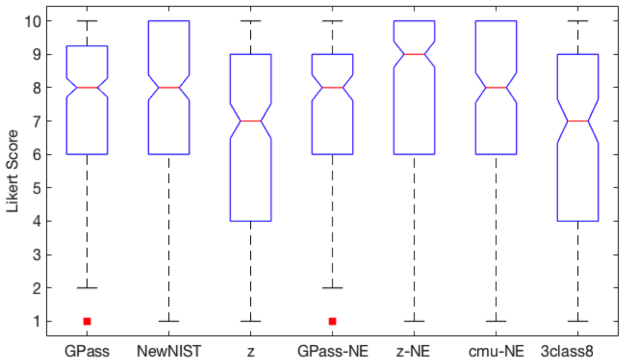

Fig. 7: Boxplots of User Preference (easy to use) on Likert Scale (1-Strongly disagree, 5-Neutral, 10-Strongly Agree)

8 Discussion and Conclusions

In this paper we proposed GuidedPass, a system that provides suggestions at password creation time, which improve password strength without sacrificing memorability.

Suggestions: Existing password meters, suggestions, or policy fail to adequately help users create strong yet memorable passwords. Therefore, it remains a critical challenge to build a password suggestion system, which helps users create both memorable and strong passwords. GuidedPass offers semantically meaningful and intuitive suggestions to users to create highly memorable passwords by extending their existing initial inputs as shown in Table 5. Although our suggestions are similar to DataPass [26], our approach provides more options and actions for users to take, and encourages structural changes. We believe this is an effective way to guide users, and our results support this. GuidedPass achieves 81% recall after two days and an average strength of 10^{17} guesses.

Acceptance: Users bear the responsibility of ensuring the memorability of passwords created under the various password meters, policies, and suggestion systems. They attempt to balance competing requirements for strength and memorability, and usually err on the side of weaker but more memorable passwords. We demonstrate that GuidedPass can preserve memorability and improve strength simultaneously, not separately. Participants in our study exhibited high recall, and seemed to naturally follow our suggestions to create strong passwords, even without strength enforcement. Conversely, the worst scenario for users was to merely enforce a strict policy or strength requirement without providing suggestions. In this scenario, users were be trapped into creating non-memorable passwords only to meet the strength requirement. Overall users found that GuidedPass was usable. Therefore, GuidedPass shows a promising research direction.

Application: Although suggestion based approaches (GuidedPass and CMU) provide higher memorability and strength, they take twice longer than non-suggestion based approaches (meters and policies). We believe that a longer creation time pays off if users can create memorable and strong passwords. GuidedPass can be easily integrated with the existing password creation systems, by modifying server feedback to the user. No other part of user authentication would need to change. Thus GuidedPass is highly deployable.

Acknowledgement

We thank our shepherd Tudor A. Dumitras and anonymous reviewers for their helpful feedback on drafts of this paper. This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ICT Consilience Creative program(IITP-2017-R0346-16-1007) supervised by the IITP(Institute for Information & communications Technology Promotion), and by NRF of Korea funded by the MSIT(NRF-2017R1C1B-5076474).

References

1. Frequently occurring surnames from the census 2000. `http://www.census.gov/topics/population/genealogy/data/2000_surnames.html`. Accessed: 2015-10-14.
2. A. I. Ansaldo, K. Marcotte, L. Scherer, and G. Raboyeau. Language therapy and bilingual aphasia: Clinical implications of psycholinguistic and neuroimaging research. *Journal of Neurolinguistics*, 21(6):539–557, 2008.
3. M. Blum and S. S. Vempala. Publishable humanly usable secure password creation schemas. In *Third AAAI Conference on Human Computation and Crowdsourcing*, 2015.
4. J. Bonneau and S. E. Schechter. Towards reliable storage of 56-bit secrets in human memory. In *USENIX Security Symposium*, pages 607–623, 2014.
5. M. Burnett. Today i am releasing ten million passwords. 2015.
6. S. D. Crawford, M. P. Couper, and M. J. Lamias. Web surveys: Perceptions of burden. *Social science computer review*, 19(2):146–162, 2001.
7. X. d. C. de Carnavalet and M. Mannan. From very weak to very strong: Analyzing password-strength meters. In *NDSS*, volume 14, pages 23–26, 2014.
8. M. Dell'Amico and M. Filippone. Monte Carlo Strength Evaluation: Fast and Reliable Password Checking. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 158–169. ACM, 2015.
9. S. Egelman, A. Sotirakopoulos, I. Muslukhov, K. Beznosov, and C. Herley. Does my password go up to eleven?: the impact of password meters on password selection. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2379–2388. ACM, 2013.
10. D. Florêncio and C. Herley. Where do security policies come from? In *Proceedings of the Sixth Symposium on Usable Privacy and Security*, page 10. ACM, 2010.
11. D. Florêncio, C. Herley, and P. C. Van Oorschot. Pushing on string: The'don't care'region of password strength. *Communications of the ACM*, 59(11):66–74, 2016.
12. P. A. Grassi, J. L. Fenton, E. M. Newton, R. A. Perlner, A. R. Regenscheid, W. E. Burr, J. P. Richer, N. B. Lefkovitz, J. M. Danker, Y. Choong, et al. Draft nist special publication 800 63b digital identity guidelines. 2017.

13. N. E. A. Guideline. Nist special publication 800-63 version 1.0. 2, 2006.
14. H. Habib, J. Colnago, W. Melicher, B. Ur, S. Segreti, L. Bauer, N. Christin, and L. Cranor. Password creation in the presence of blacklists. 2017.
15. K. C. Hanesamgar Ameya, Woo and J. Mirkovic. Leveraging semantic transformation to investigate password habits and their causes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.
16. P. G. Inglesant and M. A. Sasse. The true cost of unusable password policies: password use in the wild. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 383–392. ACM, 2010.
17. S. Ji, S. Yang, T. Wang, C. Liu, W.-H. Lee, and R. Beyah. Pars: A uniform and open-source password analysis and research system. In *Proceedings of the 31st Annual Computer Security Applications Conference*, pages 321–330. ACM, 2015.
18. P. G. Kelley, S. Komanduri, M. L. Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, and J. Lopez. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 523–537. IEEE, 2012.
19. S. Komanduri, R. Shay, L. F. Cranor, C. Herley, and S. E. Schechter. Telepathwords: Preventing weak passwords by reading users' minds. In *USENIX Security*, pages 591–606, 2014.
20. 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 SIGCHI Conference on Human Factors in Computing Systems*, pages 2595–2604. ACM, 2011.
21. 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 32nd annual ACM conference on Human factors in computing systems*, pages 2927–2936. ACM, 2014.
22. 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. Designing password policies for strength and usability. *ACM Transactions on Information and System Security (TISSEC)*, 18(4):13, 2016.
23. R. Shay, S. Komanduri, P. G. Kelley, P. G. Leon, M. L. Mazurek, L. Bauer, N. Christin, and L. F. Cranor. Encountering stronger password requirements: user attitudes and behaviors. In *Proceedings of the Sixth Symposium on Usable Privacy and Security*, page 2. ACM, 2010.
24. W. C. Summers and E. Bosworth. Password policy: the good, the bad, and the ugly. In *Proceedings of the winter international symposium on Information and communication technologies*, pages 1–6. Trinity College Dublin, 2004.
25. UCREL CLAWS7 Tagset. `http://ucrel.lancs.ac.uk/claws7tags.html`, 2016.
26. B. Ur, F. Alfieri, M. Aung, L. Bauer, N. Christin, J. Colnago, L. Cranor, H. Dixon, P. E. Naeini, H. Habib, N. Johnson, and W. Melicher. Design and evaluation of a data-driven password meter. In *CHI'17: 35th Annual ACM Conference on Human Factors in Computing Systems*, May 2017.
27. B. Ur, P. G. Kelley, S. Komanduri, J. Lee, M. Maass, M. L. Mazurek, T. Passaro, R. Shay, T. Vidas, L. Bauer, et al. How does your password measure up? the effect of strength meters on password creation. In *USENIX Security Symposium*, pages 65–80, 2012.
28. R. Veras, C. Collins, and J. Thorpe. On the semantic patterns of passwords and their security impact. In *Network and Distributed System Security Symposium (NDSS'14)*, 2014.
29. D. L. Wheeler. zxcvbn: Low-budget password strength estimation. In *Proc. USENIX Security*, 2016.
30. S. Woo, E. Kaiser, R. Artstein, and J. Mirkovic. Life-experience passwords (leps). In *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pages 113–126. ACM, 2016.