Design and Evaluation of a Data-Driven Password Meter

**Blase Ur*, Felicia Alfieri, Maung Aung, Lujo Bauer, Nicolas Christin,
Jessica Colnago, Lorrie Faith Cranor, Henry Dixon, Pardis Emami Naeini,
Hana Habib, Noah Johnson, William Melicher**
*University of Chicago, Carnegie Mellon University
blase@uchicago.edu
{fla, mza, lbauer, nicolasc, jcolnago, lorrie, hdixon, pardis, hana007, noah, billy}@cmu.edu

ABSTRACT

Despite their ubiquity, many password meters provide inaccurate strength estimates. Furthermore, they do not explain to users what is wrong with their password or how to improve it. We describe the development and evaluation of a data-driven password meter that provides accurate strength measurement and actionable, detailed feedback to users. This meter combines neural networks and numerous carefully combined heuristics to score passwords and generate data-driven text feedback about the user's password. We describe the meter's iterative development and final design. We detail the security and usability impact of the meter's design dimensions, examined through a 4,509-participant online study. Under the more common password-composition policy we tested, we found that the data-driven meter with detailed feedback led users to create more secure, and no less memorable, passwords than a meter with only a bar as a strength indicator.

ACM Classification Keywords

K.6.5 Security and Protection: Authentication; H.5.2 User Interfaces: Evaluation/methodology

Author Keywords

Passwords; usable security; data-driven; meter; feedback

INTRODUCTION

Password meters are used widely to help users create better passwords [42], yet they often provide ratings of password strength that are, at best, only weakly correlated to actual password strength [10]. Furthermore, current meters provide minimal feedback to users. They may tell a user that his or her password is "weak" or "fair" [10, 42, 52], but they do not explain what the user is doing wrong in making a password, nor do they guide the user towards a better password.

In this paper, we describe our development and evaluation of an open-source password meter that is more accurate at rating the strength of a password than other available meters and provides more useful, actionable feedback to users. Whereas most previous meters scored passwords using very basic heuristics [10, 42, 52], we use the complementary techniques of simulating adversarial guessing using artificial neural networks [32] and employing 21 heuristics to rate password strength. Our meter also gives users actionable, data-driven feedback about how to improve their specific candidate password. We provide users with up to three ways in which they could improve their password based on the characteristics of their specific password. Furthermore, we automatically propose modifications to the user's password through judicious insertions, substitutions, rearrangements, and case changes.

In this paper, we describe our meter and the results of a 4,509-participant online study of how different design decisions impacted the security and usability of passwords participants created. We tested two password-composition policies, three scoring stringencies, and six different levels of feedback, ranging from no feedback whatsoever to our full-featured meter.

Under the more common password-composition policy we tested, we found that our data-driven meter with detailed feedback led users to create more secure passwords than a meter with only a bar as a strength indicator or not having any meter, without a significant impact on any of our memorability metrics. Most participants reported that the text feedback was informative and helped them create stronger passwords.

RELATED WORK

Users sometimes make predictable passwords [22, 30, 48] even for important accounts [13, 31]. Many users base passwords around words and phrases [5, 23, 29, 45, 46]. When passwords contain uppercase letters, digits, and symbols, they are often in predictable locations [4]. Keyboard patterns like "1qaz2wsx" [46] and dates [47] are common in passwords. Passwords sometimes contain character substitutions, such as replacing "e" with "3" [26]. Furthermore, users frequently reuse passwords [9, 14, 25, 38, 48], giving the compromise of even a single account potentially far-reaching repercussions. In designing our meter, we strove to help users understand when their password exhibited these common tendencies.

Three types of interventions attempt to guide users towards strong passwords. First, password-composition policies dictate characteristics a password must include, such as particular

CHI 2017, May 06–11, 2017, Denver, CO, USA
ACM 978-1-4503-4655-9/17/05.
http://dx.doi.org/10.1145/3025453.3026050

character classes. While these policies can be effective for security, users often find complex composition policies unusable [2,20,24,28,37,50]. Second, proactive password checking aims to model a password's security and only permit users to select a password the model deems sufficiently strong. For instance, researchers have proposed using server-side Markov models to gauge password strength [8]. This approach is inappropriate when the password should never be sent to a server (e.g., encrypting a hard drive). It also requires non-trivial configuration and can enable side-channel attacks [35].

Third, services commonly provide password meters to show estimated password strength. To enable these meters to run client-side and thus avoid the security pitfalls of a server-side solution, most meters use basic heuristics, such as estimating a password's strength based on its length and the number of character classes used [42]. These sorts of meters can successfully encourage users to create stronger passwords [42], though perhaps only for higher-value accounts [12]. Unfortunately, these basic heuristics frequently do not reflect the actual strength of a password [1, 10]. Among prior password meters based on heuristics, only zxcvbn [51, 52] uses more advanced heuristics [10, 32]. A key difference from zxcvbn in the design of both our meter and our experiment is that generating and testing the impact of feedback to the user is primary for us.

Researchers have tried numerous visual displays of password strength. Using a bar is most common [42]. However, researchers have studied using large-scale training data to show users predictions of what they will type next [27]. Others have investigated a peer-pressure meter that compares the strength of a user's password with those of other users [36]. These alternative visualizations have yet to be widely adopted.

MEASURING PASSWORD STRENGTH IN OUR METER

We move beyond measuring password strength using inaccurate basic heuristics by combining two complementary approaches: modeling a principled password-guessing attack using neural networks and carefully combining 21 heuristics.

Our first approach relies on recent work that proposed neural networks for modeling a password-guessing attack [32]. This approach uses a recurrent neural network to assign probabilities to future characters in a candidate guess based on the previous characters. In its training phase, the network learns various higher-order password features. Using Monte Carlo methods [11], such an approach can model 10^{30} or more adversarial guesses in real time entirely on the client side after transferring less than a megabyte of data to the client [32].

However, neural networks are effectively a black box and provide no explanation to the user for their scoring. Inspired by the zxcvbn password meter [52], we implemented and carefully combined 21 heuristics for password scoring. These heuristics search for characteristics such as the inclusion of common words and phrases, the use of common character substitutions, the placement of digits and uppercase letters in common locations, and the inclusion of keyboard patterns. We scored 30,000 passwords from well-studied data breaches [18, 45] by these heuristics, and we ran a regression comparing these scores to the password's guessability, as modeled by CMU's Password Guessability Service [7]. This service models four types of guessing attacks and has been found to be a conservative proxy for an expert attacker [44].

Although these carefully combined heuristics also provide relatively accurate password-strength estimates, at least for resistance to online attacks [52], we use them primarily to identify characteristics of the password that are associated with guessability. If a candidate password scores high on a particular heuristic, indicating the presence of a common pattern, we generate text feedback that explains what is wrong with that aspect of the password and how to improve it. We developed the wordings for this feedback iteratively and through a formative lab study.[1]

VISUAL DESIGN AND USER EXPERIENCE

In this section, we describe the visual design of our meter. At a high level, the meter comprises three different screens. The *main screen* uses the visual metaphor of a bar to display the strength of the password, and it also provides detailed, data-driven feedback about how the user can improve his or her password. The main screen also contains links to the two other screens. Users who click "(Why?)" links next to the feedback about their specific password are taken to the *specific-feedback modal*, which gives more detailed feedback. Users who click the "How to make strong passwords" link or "(Why?)" links adjacent to feedback about password reuse are taken to the *generic-advice modal*, which is a static list of abstract strategies for creating strong passwords.

Translating Scores to a Visual Bar

Password-strength measurements are normally displayed to users not as a numerical score, but using a colored bar [10,42]. In creating our meter, we needed to map a password's scores from both the neural network and combined heuristics to the amount of the bar that should be filled. We conservatively calculated $\log_{10}$ of the lower of the two estimates for the number of guesses the password would withstand.

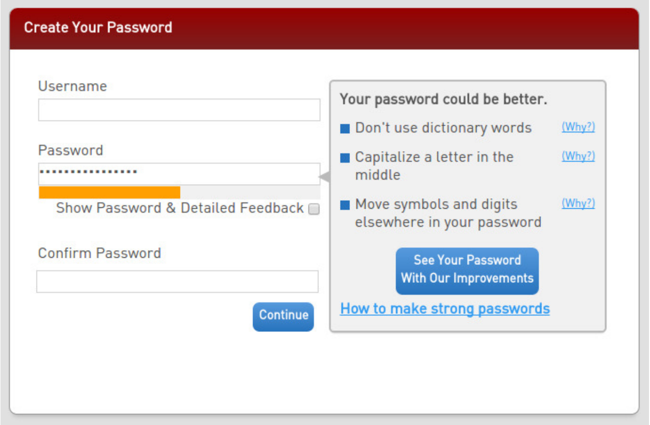
Prior work has shown that most users consider a password sufficiently strong if only part of the bar is full [42]. Therefore, we mapped scores to the bar such that one-third of the bar being filled roughly equates to a candidate password resisting an online attack, while two-thirds means that the candidate password would likely resist an offline attack, assuming that a hash function designed for password storage is used. We tested three different precise mappings, which we term *stringencies*.

Main Screen

On the main screen a bar below the password field fills up and changes color to indicate increasing password strength. Different from previous meters, our meter also displays data-driven text feedback about what aspects of the user's specific password could be improved.

The meter initially displays text listing the requirements a password must meet. Once the user begins to enter a password,

[1] See Ch. 7 of Ur's dissertation [40] for details on the development of this wording, the results of the lab study, and the heuristics.


Create Your Password
Username
Password
•••••••••••••••••
Show Password & Detailed Feedback □
Confirm Password
Continue
See Your Password With Our Improvements
How to make strong passwords

Your password could be better.

- ■ Don't use dictionary words (Why?)
- ■ Capitalize a letter in the middle (Why?)
- ■ Move symbols and digits elsewhere in your password (Why?)

Figure 1: The tool's main screen when the password is hidden.

the tool indicates that a particular requirement has been met by coloring that requirement's text green and displaying a check mark. It denotes unmet requirements by coloring those requirements red and displaying empty checkboxes. Until the password meets requirements, the bar is gray.

Colored Bar

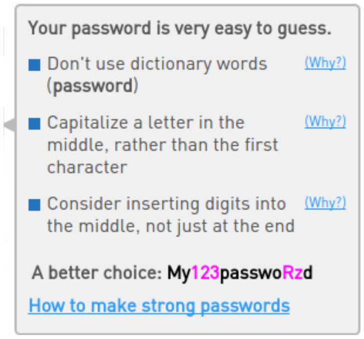
Once the password meets composition requirements, we display the bar in color. With increasing password-strength ratings, the bar progresses from red to orange to yellow to green. When it is one-third full, the bar is dark orange. At two-thirds full, it is yellow, soon to be green.

Text Feedback

Whereas a colored bar is typical of password meters [42], our meter is among the first to provide detailed, data-driven feedback on how the user can improve his or her specific candidate password. We designed the *text feedback* to be directly actionable, in addition to being specific to the password. Examples of this feedback include, as appropriate, suggestions to avoid dictionary words and keyboard patterns, to move uppercase letters away from the front of the password and digits away from the end, and to include digits and symbols.

Most feedback comments on specific parts of the password. Because users likely do not expect their password to be shown on screen, we designed *public* and *sensitive* variants for all feedback. Public versions mention only the general class of characteristic (e.g., "avoid using keyboard patterns"), whereas sensitive versions also display the problematic portion of the password (e.g., "avoid using keyboard patterns like adgjl"). We display the public versions of feedback (Figure 1) when users are not showing their password on screen, which is the default behavior. We provide checkboxes with which a user can "show password & detailed feedback," at which point we use the sensitive version (Figure 2).

To avoid overwhelming the user, we show at most three pieces of feedback at a time. Each of our 21 heuristic functions returns either one sentence of feedback or the empty string. To choose which feedback to display, we manually ordered the functions to prioritize those that we found in a formative laboratory study provided the most useful information and that correlated most strongly with password guessability in our regression combining the heuristics.



Your password is very easy to guess.

- ■ Don't use dictionary words (password) (Why?)
- ■ Capitalize a letter in the middle, rather than the first character (Why?)
- ■ Consider inserting digits into the middle, not just at the end (Why?)

A better choice: My123passwoRzd

Figure 2: A suggested improvement for "Mypassword123," with changes in magenta. The suggested improvement and sensitive feedback appear when users show their password.

Suggested Improvement

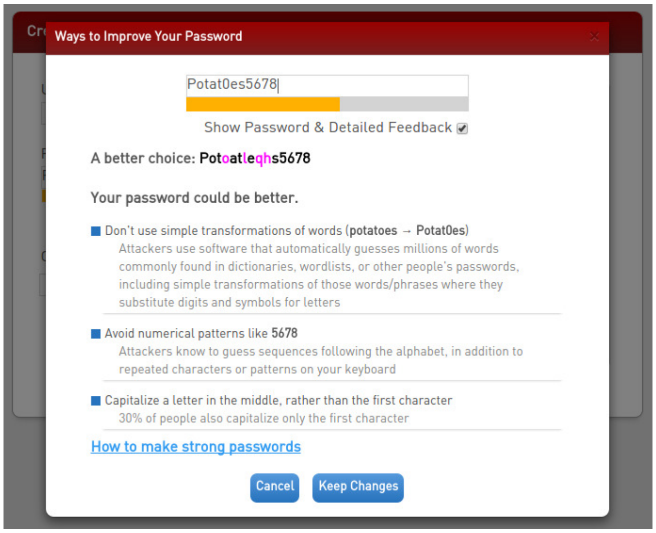
To make it easier for users to create a strong password, we augmented our text feedback with a *suggested improvement*, or concrete modification of the candidate password. As shown in Figure 2, a suggested improvement for "Mypassword123" might be "My123passwoRzd."

We generate this suggested improvement as follows. First, we take the user's candidate password and make one of the following modifications: toggle the case of a random letter (lowercase to uppercase, or vice versa), insert a random character in a random position, or replace a character at a random position with a randomly chosen character. In addition, if all of the password's digits or symbols are in a common location (e.g., at the end), we move them as a group to a random location within the password. We choose from among this large set of modifications, rather than just making modifications corresponding to the specific text feedback the meter displays, to greatly increase the space of possible modifications. We then verify that this modification still complies with the composition requirements.

We require that suggested improvements would fill at least two-thirds of the bar and also be at least 1.5 orders of magnitude harder to guess than the original candidate password. When we have generated such a suggested improvement and the user is currently showing his or her password, we display the suggested modification on screen with changes in magenta, as in Figure 2. If the user is not showing his or her password, we instead show a blue button stating "see your password with our improvements." Prior work has investigated automatically modifying passwords to increase security, finding users would make weaker pre-improvement passwords to compensate [16]. In contrast, our suggested improvements are optional.

Specific-Feedback Modal

For the main screen, we designed the feedback specific to a user's password to be both succinct and action-oriented. Although the rationale for these specific suggestions might be obvious to many users, we expected it would not be obvious to all users based on prior work on password perceptions [41,



Ways to Improve Your Password

Potat0es5678

Show Password & Detailed Feedback ☑

A better choice: **Potoatleqhs5678**

Your password could be better.

- Don't use simple transformations of words (**potatoes → Potat0es**)
  Attackers use software that automatically guesses millions of words commonly found in dictionaries, wordlists, or other people's passwords, including simple transformations of those words/phrases where they substitute digits and symbols for letters
- Avoid numerical patterns like **5678**
  Attackers know to guess sequences following the alphabet, in addition to repeated characters or patterns on your keyboard
- Capitalize a letter in the middle, rather than the first character
  30% of people also capitalize only the first character

How to make strong passwords

Figure 3: The specific-feedback modal (password shown).

43]. To give detailed explanations of why specific suggestions would improve a password, we created a specific-feedback modal, shown in Figure 3. When a user clicks "(Why?)" next to password-specific feedback, the modal appears.

The specific-feedback modal's main bullet points mirror those from the main screen. Below each main point, however, the specific-feedback modal also explains why this action would improve the password. Our explanations take two primary forms. The first form is explanations of how attackers could exploit particular characteristics of the candidate password. For instance, if a password contains simple transformations of dictionary words, we explain that attackers often guess passwords by trying such transformations. The second form of explanation is statistics about how common different characteristics are (e.g., 30% of passwords that contain a capital letter have only the first character of the password capitalized).

Generic-Advice Modal

It is not always possible to generate data-driven feedback. For instance, until a user has typed more than a few characters into the password field, the strength of the candidate password cannot yet be definitively determined. In addition, extremely predictable candidate passwords (e.g., "password" or "monkey1") require that users completely rethink their strategy. We thus created a generic-advice modal that recommends abstract strategies for creating passwords. Users access this modal by clicking "how to make strong passwords" or by clicking "(Why?)" next to suggestions against password reuse. The first of four points we make on the generic-advice modal advises against reusing a password across accounts. We chose to make this the first point because password reuse is very common [9, 14, 15, 38], yet a major threat to security.

We recommend constructive, action-oriented strategies as our second and third points. Reflecting research that showed that passwords balancing length and character complexity were often strong [34] and because attackers can often exhaustively guess all passwords up to at least 9 characters in offline attacks [19, 39], our second point recommends using at least 12 characters in the password. To help inspire users who are unfamiliar with how to make a 12+ character password, we use the third point to propose a way of doing so based on Schneier's method to use mnemonics or fragments from a unique sentence as the basis for a password [33]. The fourth point recommends against common password characteristics.

METHODOLOGY

We recruited participants from Amazon's Mechanical Turk crowdsourcing service for a study on passwords. We required that participants be age 18+ and be located in the United States. In addition, because we had only verified that the meter worked correctly on Firefox, Chrome/Chromium, Safari, and Opera, we required they use one of those browsers.

In order to measure both password creation and password recall, the study comprised two parts. The first part of the study, which we term *Part 1*, included a password creation task, a survey, and a password recall task. We assigned participants round-robin to a condition specifying the meter variant they would see when creating a password. The second part of the study, which we term *Part 2*, took place at least 48 hours after the first part of the study and included a password recall task and a survey. We compensated participants $0.55 for completing Part 1 and $0.70 for Part 2.

After studying 18 conditions in our first experiment, we had lingering questions. We therefore ran a second experiment that added 8 new conditions and repeated 4 existing conditions.

Part 1

Following the consent process, we told participants that they would be creating a password. We asked that they role play and imagine that they were creating this password for "an account they care a lot about, such as their primary email account." We informed participants they would be invited back in a few days to recall the password and asked them to "take the steps you would normally take to create and remember your important passwords, and protect this password as you normally would protect your important passwords."

The participant then created a username and a password. While doing so, he or she saw the password-strength meter (or lack thereof) dictated by his or her assigned condition, described below. Participants then answered a survey about how they created that password. We first asked participants to respond on a 5-point scale ("strongly disagree," "disagree," "neutral," "agree," "strongly agree") to statements about whether creating a password was "annoying," "fun," or "difficult." We also asked whether they reused a previous password, modified a previous password, or created an entirely new password.

The next three parts of the survey asked about the meter's colored bar, text feedback, and suggested improvements. At the top of each page, we showed a text explanation and visual example of the feature in question. Participants in conditions that lacked one or more of these features were not asked questions about that feature. For the first two features, we asked participants to rate on a five-point scale whether that feature "helped me create a strong password," "was not informative," and caused them to create “a different password than [they] would have otherwise.” We also asked about the importance participants place on the meter giving their password a high score, their perception of the accuracy of the strength rating, and whether they learned anything new from the feedback.

After the participant completed the survey, we brought him or her to a login page and auto-filled his or her username. The participant then attempted to re-enter his or her password. We refer to this final step as *Part 1 recall*. After five incorrect attempts, we let the participant proceed.

Part 2

After 48 hours, we automatically emailed participants to return and re-enter their password. We term this step *Part 2 recall*, and it was identical to Part 1 recall. We then directed participants to a survey about how they tried to remember their password. In particular, we first asked how they entered their password on the previous screen. We gave multiple-choice options encompassing automatic entry by a password manager or browser, typing the password in entirely from memory, and looking a password up either on paper or electronically.

Conditions

In Experiment 1, we assigned participants round-robin to one of 18 different conditions that differed across three dimensions in a full-factorial design. We refer to our conditions using three-part names reflecting the dimensions: 1) password-composition policy; 2) type of feedback; and 3) stringency.

Dimension 1: Password-Composition Policy

We expected a meter to have a different impact on password security and usability depending on whether it was used with a minimal or a more complex password-composition policy. We tested the following two policies, which represent a widespread, lax policy and a more complex policy.

- **1class8 (1c8)** requires that passwords contain 8 or more characters, and also that they not be (case-sensitive) one of the 96,480 such passwords that appeared four or more times in the Xato corpus of 10 million passwords [6].
- **3class12 (3c12)** requires that passwords contain 12 or more characters from at least 3 different character classes (lowercase letters, uppercase letters, digits, and symbols). It also requires that the password not be (case-sensitive) one of the 96,926 such passwords that appeared in the Xato corpus [6].

Dimension 2: Type of Feedback

Our second dimension varies the type of feedback we provide to participants about their password. While the first setting represents our standard meter, we removed features for each of the other settings to test the impact of those features.

- **Standard (Std)** includes all previously described features.
- **No Suggested Improvement (StdNS)** is the same as Standard, except it never displays a suggested improvement.
- **Public (Pub)** is the same as Standard, except we never show sensitive text feedback (i.e., we never show a suggested improvement and always show the less informative “public” feedback normally shown when the password is hidden).
- **Bar Only (Bar)** shows a colored bar displaying password strength, but we do not provide any type of text feedback other than which composition requirements have been met.
- **No Feedback (None)** only indicates compliance with the password-composition requirements.

Dimension 3: Scoring Stringency

Ur et al. found the stringency of a meter’s scoring has a significant impact on password strength [42]. We thus tested two scoring stringencies. These stringencies changed the mapping between the estimated number of guesses the password would resist and how much of the colored bar was filled.

- **Medium (M)** One-third of the bar full represents 10^6 estimated guesses and two-thirds full represents 10^{12}.
- **High (H)** One-third of the bar full represents 10^8 estimated guesses and two-thirds full represents 10^{16}.

Additional Conditions for Experiment 2

Experiment 2 added the following two settings for our feedback and stringency dimensions, respectively:

- Feedback: **Standard, No Bar (NoBar)** The Standard meter without any colored bar. The text feedback still depends on the password’s score, so stringency still matters.
- Stringency: **Low (L)** One-third of the bar full represents 10^4 estimated guesses and two-thirds full represents 10^8.

To investigate these two settings we introduced eight new conditions and re-ran the four existing Std-M and Std-H conditions in a full-factorial design.

Analysis

We collected numerous types of data. Our main security metric was the guessability of each participant’s password, as calculated by CMU’s Password Guessability Service [7], which models four types of guessing attacks and which they found to be a conservative proxy for an expert attacker [44]. Our usability measurements encompassed both quantitative and qualitative data. We recorded participants’ keystrokes, enabling us to analyze metrics like password creation time. To understand the use of different features, we instrumented all elements of the user interface to record when they were clicked.

For both Part 1 and Part 2, we measured whether participants successfully recalled their password. For participants who did successfully recall their password, we also measured how long it took them, as well as how many attempts were required. Because not all participants returned for Part 2, we measured what proportion of participants did, hypothesizing that participants who did not remember their password might be less likely to return. To only study attempts at recalling a password from memory, we analyzed password recall only for participants who said they typed their password in entirely from memory, said they did not reuse their study password, and whose keystrokes did not show evidence of copy-pasting.

We augmented our objective measurements with analyses of responses to multiple-choice questions and qualitative analysis of free-text responses. These optional free-text responses solicited participants’ thoughts about the interface elements, as well as why they did (or did not) find them useful.

Our primary goal was understanding the quantitative effects of varying the three meter-design dimensions. Because we had multiple independent variables, each reflecting one design dimension, we performed regressions. We ran a linear regression for continuous data (e.g., the time to create a password), a logistic regression for binary data (e.g., whether or not they clicked on a given UI element), an ordinal regression for ordinal data (e.g., Likert-scale responses), and a multinomial logistic regression for categorical data with no clear ordering (e.g., how they entered their password).

For our security analyses, we performed a Cox Proportional-Hazards Regression, which is borrowed from the literature on survival analysis and has been used to compare password guessability [31]. Because we know the starting point of guessing but not the endpoint, we use a right-censored model [17]. In a traditional clinical model using survival analysis, each data point is marked as "alive" or "deceased," along with the time of the observation. Our analogues for passwords are "not guessed" and "guessed," along with the number of guesses at which the password was guessed, or the guessing cutoff.

We always first fit a model with the three design dimensions (composition policy, feedback, and stringency) each treated as ordinal variables fit linearly, as well as interaction terms for each pair of dimensions. To build a parsimonious model, we removed any interaction terms that were not significant, yet always kept all three main effects, and re-ran the model.

We corrected for multiple testing using the Benjamini-Hochberg (BH) procedure [3]. We chose this approach, which is more powerful and less conservative than methods like Holm Correction, because we performed an exploratory study with a large number of variables. We corrected all p-values for each experiment as a group. We use $\alpha = 0.05$.

Note that we analyzed Experiments 1 and 2 separately. Thus, we do not compare conditions between experiments. However, in the graphs and tables that follow we have combined our reporting of these two experiments for brevity. We only report "NoBar" and low-stringency data from Experiment 2 in these tables and graphs. For conditions that were part of both experiments, we only report the results from Experiment 1.

Limitations

We based our study's design on one researchers have used previously to investigate various aspects of passwords [21, 28, 34, 42]. However, the password participants created did not protect anything of value. Beyond our request that they do so, participants did not need to exhibit their normal behavior. Mazurek et al. [31] and Fahl et al. [13] examined the ecological validity of this protocol, finding it to be a reasonable, albeit imperfect, proxy for high-value passwords for real accounts.

That said, no controlled experiment can capture every aspect of password creation. We did not control the device [49, 53] on which participants created a password, nor could we control how participants chose to remember their password. We tested password recall at only two points in time. Our limited evaluation of memorability does not capture many aspects of the password ecosystem. Some passwords are entered very frequently, while others are entered very rarely, and the meter's impact on memorability (or lack thereof) may differ in these scenarios. Furthermore, while we studied our meter's impact on the creation of a single password, users typically have a large number of passwords, potentially impacting multi-account intereference effects. If a future study were to find that our meter increases multi-account interference, the meter might be best deployed only for high-value accounts.

Our study's ecological validity is also limited in other aspects. We did not test habituation effects, either to the use of a particular password or to the novel password-meter features we tested. Our password meter might cause long-term changes in how users create passwords in the wild that only an extended field study could capture. On the one hand, lessons users learn from our meter about password creation could help them create stronger passwords for sites that do not provide feedback. On the other hand, potential usability drawbacks might become more pronounced or have a larger effect over time in the wild. In addition, password creation was the primary task in our study. Our meter might have a different impact in the more typical scenario of password creation being users' secondary task, which might lower their motivation to create strong passwords.

PARTICIPANTS

4,509 participants (2,717 in Experiment 1 and 1,792 in Experiment 2) completed Part 1, and 84.1% of them finished Part 2. Among our participants, 52% identified as female, 47% as male, and the remaining 1% identified as another gender or preferred not to answer. Participants' ages ranged from 18 to 80 years old, with a median of 32 (mean 34.7). We asked whether participants are "majoring in or...have a degree or job in computer science, computer engineering, information technology, or a related field," and 82% responded "no." Demographics did not vary significantly by condition.

RESULTS

Increasing levels of data-driven feedback, even beyond just a colored bar, led users to create stronger 1class8 passwords. That is, detailed text feedback led to more secure passwords than the colored bar alone. The 3class12 policy also led to stronger passwords. In contrast, varying the scoring stringency had only a small impact.

Among our usability metrics, all dimensions affected timing and sentiment, but they mostly did not affect password memorability. Notably, although increasing levels of data-driven feedback led to stronger passwords, we did not observe any significant impact on memorability. Table 1 summarizes our key findings. Throughout this section, if we do not explicitly call out a metric as revealing a difference between variants of the meter, then we did not observe a significant difference.

Overall, 56.5% of participants said they typed their password in entirely from memory. Other participants looked it up on paper (14.6%) or on an electronic device (12.8%), such as their computer or phone. An additional 11.2% of participants said their password was entered automatically for them by a password manager or browser, while 4.9% entered their password in another way (e.g., looking up hints).

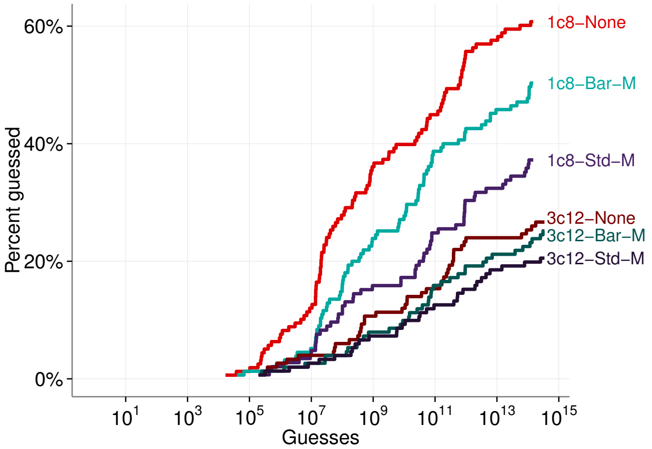


Figure 4: Guessability of medium-stringency passwords created without any feedback ("None"), with only a colored bar ("Bar"), and with both a colored bar and text feedback ("Std").

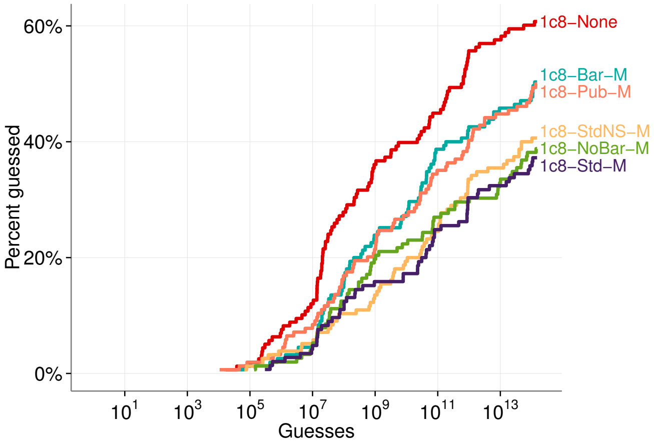


Figure 5: Guessability of 1class8, medium-stringency passwords created with different levels of feedback. This graph shows data from Experiment 1. However, we have added 1c8-NoBar-M from Experiment 2 for illustrative purposes.

Considering password reuse and copy-pasting, 50.6% of participants tried to recall a novel study password from memory; these are the participants for whom we examine password recall. Overall, 98.4% of these participants successfully recalled their password during Part 1, and the majority did so on their first attempt. In total, 78.2% of participants who returned for Part 2 and who tried to recall a novel study password from memory succeeded, again primarily on their first attempt.

We first discuss how the three different dimensions of the meter's design impacted security and usability. We then detail how participants interacted with, and qualitatively responded to, the meter's various features.

Impact of Composition Policy

We first ran a Cox regression with all three dimensions and their pairwise interactions as independent variables, and the password's survival term as the dependent variable. For Experiment 1, we found significant main effects for both the policy and type of feedback, but we also observed a significant interaction effect between the policy and type of feedback. For increased intelligibility, we subsequently ran separate regressions for 1class8 and 3class12 passwords. As the 3class12 policy requires longer passwords than 1class8, participants unsurprisingly created 3class12 passwords that were significantly longer ($p < .001$) and significantly more secure ($p < .001$) than 1class8 passwords.

Moving from a 1class8 to a 3class12 policy increased the time it took to create the password, measured from the first keystroke to the last keystroke in the password box ($p = .014$). It also impacted participant sentiment. The 3class12 policy led participants to report password creation as significantly more annoying and difficult (both $p < .001$).

Passwords created under a 3class12 policy were more secure than those created under 1class8, but 3class12 participants were less likely to remember their passwords during Part 2 ($p = .025$). Across conditions, 81.3% of 1class8 participants recalled their password during Part 2, whereas 75.0% of 3class12 participants did. The policy did not significantly impact any other recall metrics.

Impact of the Amount of Feedback

For 1class8 passwords, we found that increasing levels of data-driven feedback led participants to create significantly stronger passwords ($p < .001$). Relative to having no feedback, the full suite of data-driven feedback led to 44% stronger passwords. As shown in Figure 4, the colored bar on its own led participants to create stronger passwords than having no feedback, echoing prior work [42]. The detailed text feedback we introduce in this work led to stronger passwords than just the bar (Figure 4). Increasing the amount of feedback also led participants to create longer passwords ($p < .001$). For example, the median length of 1c8-None passwords was 10 characters, whereas the median for 1c8-Std-M was 12 characters.

Notably, the security of 1class8 passwords created with our standard meter (including all text feedback) was more similar to the security of 3class12 passwords created without feedback than to 1class8 passwords created without feedback (Figure 4).

Figure 5 details the comparative security impact of all six feedback levels on 1class8 passwords. Whereas suggested improvements had minimal impact, having the option to show potentially sensitive feedback provided some security benefits over the public ("Pub") variant. When we investigated removing the colored bar from the standard meter, but leaving the text feedback, we found that removing the colored bar did not significantly impact password strength.

For 3class12 passwords, however, the level of feedback did not significantly impact password strength. We hypothesize that either we are observing a ceiling effect, in which the 3class12 policy by itself led participants to make sufficiently strong passwords, or that the text feedback does not provide sufficiently useful recommendations for a 3class12 policy.

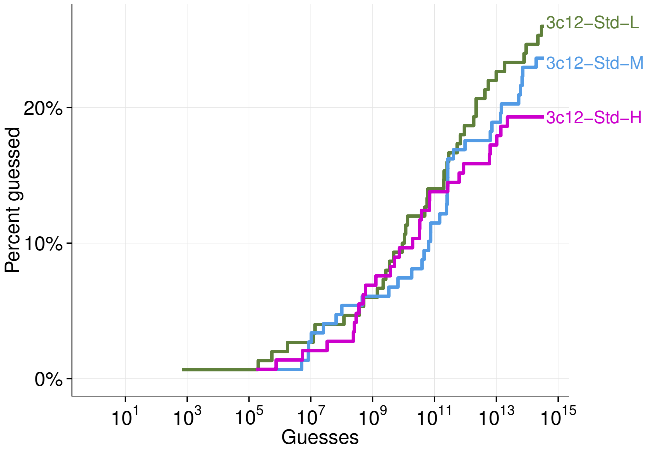


Figure 6: Guessability of 3class12 passwords by stringency.

Increasing the amount of feedback increased the time it took to create the password ($p = .011$). We observed an interaction between the amount of feedback and the stringency; increasing the amount of feedback in high-stringency conditions led to a greater time increase ($p = .048$).

To understand how participants change their passwords during creation, we examined the number of deletions, which we defined as a participant removing characters from their password that they added in a prior keystroke. Increasing amounts of feedback led to significantly more deletions ($p < .001$), implying that the feedback causes participants to change their password-creation decisions. For instance, the median number of deletions for 1c8-None was zero, while the median number for 1c8-Std-H was 9.

Increasing the amount of feedback negatively affected participant sentiment. It led participants to report password creation as more annoying ($p < .001$), more difficult ($p < .001$), and less fun ($p = .025$). For each sentiment, roughly 10%–15% of the participants in that condition moved from agreement to disagreement, or vice versa.

Even though increasing the amount of feedback led to significantly more secure passwords, it did not significantly impact any of our recall metrics.

Impact of Stringency

Although prior work on password meters found that increased scoring stringency led to stronger passwords [42], we found that varying between medium and high stringency did not significantly impact security. Because both our medium and high stringency levels were more stringent than most real-world password meters, we investigated an additional low stringency setting in Experiment 2. With these three levels, we found that increasing levels of stringency did lead to stronger passwords, but only for 3class12 passwords ($p = .043$). Figure 6 shows the impact of varying the 3class12 stringency. In all cases, however, increasing the stringency led participants to create longer passwords ($p < .001$). For instance, the median high-stringency 1class8 password was one character longer than the median low-stringency 1class8 password. High-stringency 3class12 passwords were two characters longer. Prior work tested a meter that used only basic heuristics (length and character classes) to score passwords, with a particular emphasis on length [42]. As a result, participants could fill more of the stringent meters simply by making their password longer. In contrast, our meter scores passwords far more rigorously, which we hypothesize might account for this difference.

Although increasing the scoring stringency led participants to create longer passwords, varying between medium and high stringency did not affect participant sentiment or how long it took to create a password. When we added an additional low stringency level in Experiment 2, however, participants with more stringent meters took longer to create a password ($p < .001$) and deleted more characters during creation ($p < .001$). They also took longer to recall their password during Part 1 ($p = 0.010$) and were less likely to try recalling their password solely from memory ($p = .002$), though stringency did not significantly impact other recall metrics. For high-stringency conditions, increased feedback led to even more deletions ($p = .002$). However, for 3class12 policies, higher stringency led to fewer deletions overall ($p = .002$).

Increasing the stringency greatly impacted participant sentiment. It led participants to perceive password creation as more annoying, more difficult, and less fun ($p < .001$, $p < .001$, $p = .010$, respectively). It also caused participants to be more likely to say the bar helped them create a stronger password ($p = .027$), less likely to believe the bar was accurate ($p < .001$), and less likely to find it important that the bar gives them a high score ($p = .006$). Increasing levels of stringency made participants more likely to say the text feedback led them to create a different password than they would have otherwise ($p = .010$), but also less likely to believe they learned something new from the text feedback ($p < .001$).

Text Feedback

Participants reacted positively to the text feedback. Most participants (61.7%) agreed or strongly agreed that the text feedback made their password stronger. Similarly, 76.9% disagreed or strongly disagreed that the feedback was not informative, and 48.7% agreed or strongly agreed that they created a different password than they would have otherwise because of the text feedback. Higher stringency participants were more likely to say they created a different password ($p = .022$), but no other dimension significantly impacted any other one of these sentiments.

Although most participants (68.5%) selected "no" when we asked if they learned "something new about passwords (your password, or passwords in general) from the text feedback," 31.5% selected "yes." Participants commonly said they learned about moving capital letters, digits, and symbols to less predictable locations from the meter (e.g., "I didn't know it was helpful to capitalize an internal letter."). Many participants also noted that the meter's requests not to use dictionary entries or words from Wikipedia in their password taught them something new. One of these participants noted learning "that hackers use Wikipedia." Requests to include symbols also resonated. As one participant wrote, "I didn't know previously that you could input symbols into your passwords."

Table 1: A summary of how moving from a 1class8 to 3class12 policy, increasing the amount of feedback, or increasing the scoring stringency impacted key metrics.

Metric	Policy	Feedback	Stringency
Security			
Passwords harder to guess	✓	1class8 only	3class12 only
Password creation			
Longer passwords	✓	✓	✓
More time to create	✓	✓	✓
More deletions	–	✓	✓
More likely to show on screen	✓	–	✓
Less likely to show on screen	–	✓	–
More likely to show suggested improvement	–	–	✓
Sentiment about creation			
More annoying	✓	✓	✓
More difficult	✓	✓	✓
Less fun	–	✓	✓
Password recall			
Less memorable in Part 1	–	–	–
Part 1 recall took longer	–	–	✓
Less memorable in Part 2	✓	–	–
Part 2 recall took longer	–	–	–
Required more attempts	–	–	–
Participant less likely to try recalling from memory	–	–	✓

Participants also took the text feedback as an opportunity for reflection on their password-creation strategies. One participant learned "that I tend to use full words which isn't good," while another learned "don't base the password off the username." Participants exercised many of the features of our feedback, including participants who "learned to not use L33T to create a password (exchanging letters for predictable numbers)." Some participants also learned about password reuse, notably that "people steal your passwords in data breaches and they try to use it to access other accounts."

Suggested Improvement

When participants in applicable conditions showed their password or clicked "see your password with our improvements," they would see the suggested improvement. Across conditions, 37.8% of participants clicked the "show password" checkbox. Participants who made a 3class12 password, saw a higher-stringency meter, or who saw less feedback were more likely to show their password ($p < .001$, $p = .006$, and $p = .022$, respectively). While most participants who saw a suggested improvement did so because they checked "show password & detailed explanations," 8.7% of participants in applicable conditions specifically clicked the "see your password with our improvements" button. Higher stringency made participants more likely to show the suggested improvement ($p = .003$).

When we asked in the survey whether participants in those conditions had seen a suggested improvement, 34.8% selected "yes," 55.6% selected "no," and 9.6% chose the "I don't remember" option. Nearly all participants who said they did not see a suggested improvement indeed were never shown a suggested improvement because they never showed their password or clicked "see your password with our improvements." Among participants who said they saw a suggested improvement, 81.5% said that suggested improvements were useful, while 18.5% said they were not. A slight majority (50.9%) of these participants agreed or strongly agreed that the suggested improvement helped them make a stronger password. This help, however, did not often come in the form of adopting the precise suggestion offered. In each condition that offered a suggested improvement, at most 7% of participants used one of the meter's suggested passwords verbatim. Our qualitative feedback indicated that the suggested improvement often sparked other ideas for modifying the password.

We asked participants who found the suggested improvements useful to explain why. They wrote that seeing a suggested improvement "helps you modify what you already have instead of having to think of something absolutely new" and "breaks you out of patterns you might have when creating passwords." Participants particularly liked that the suggested improvement was a modification of their password, rather than an entirely new one, because it "may help spark ideas about tweaking the password versus having to start from scratch." As one participant summmarized, "It actually offers some help instead of just shutting you down by essentially saying 'no, not good enough, come up with something else.' It's very helpful."

Participants who did not find it useful expressed two streams of reasoning. The first concerned memorability. One participant explained, "I'm more likely to forget a password if I don't use familiar techniques." while another wrote, "I already have a certain format in mind when I create my password to help me memorize it and I don't like to stray from that." The second stream concerned the trustworthiness of the "algorithm that creates those suggestions." As one participant wrote, "I don't trust it. I don't want a computer knowing my passwords."

Modal Windows

Clicking a "(why?)" link next to any of the up to three pieces of feedback in any condition with text feedback opened the specific-advice modal. Few participants in our experiment clicked on "(why?)" links, and therefore few participants saw the specific-advice modal. Only 1.0% of participants across all conditions clicked on one of these links.

In contrast, 8.4% of participants looked at the generic-advice modal, though 3class12 participants were less likely to do so ($p = .025$). Participants could arrive at the generic-advice modal by clicking "how to make strong passwords" or clicking "(why?)" next to admonitions against password reuse. Participants arrived at it about evenly through these two methods.

Colored Bar

We also analyzed participants' reactions to the colored bar. All three dimensions impacted how much of the colored bar participants filled. Participants who were assigned the 3class12 policy or saw more feedback filled more of the bar, while participants whose passwords were rated more stringently unsurprisingly filled less (all $p < .001$). Few participants completely filled the bar (estimated guess numbers 10^{18} and 10^{24} in medium and high stringency, respectively). The median participant often filled half to two-thirds of the bar, depending on the stringency. For instance, for 1c8-Std-M, only 16.5%

completely filled the bar, but 51.7% filled at least two-thirds, and 73.1% filled at least half.

Overall, participants found the colored bar useful. The majority of participants (64.0%) agreed or strongly agreed that the colored bar helped them create a stronger password, 42.8% agreed or strongly agreed that the bar led them to make a different password than they would have otherwise, and 77.2% disagreed or strongly disagreed with the statement that the colored bar was not informative. Participants also looked to the colored bar for validation; 50.9% of participants agreed or strongly agreed that it is important that the colored bar gives their password a high score. High-stringency participants were less likely to care about receiving a high score ($p = .025$). With increasing amounts of feedback, participants were more likely to care about receiving a high score ($p = .002$), more likely to say that the bar helped them create a stronger password ($p < .001$) that was different than they would have otherwise ($p < .001$). They were also less likely to believe the bar was not informative ($p = .024$).

Participants mostly felt the colored bar was accurate. Across conditions, 68.2% of participants felt the bar scored their password's strength accurately, while 23.6% felt the bar gave their password a lower score than deserved. An additional 4.2% of participants felt the bar gave their password a higher score than deserved, while 4.0% did not remember how the bar scored their password. Participants in more stringent conditions were less likely to find the bar's rating accurate ($p < .001$).

We also tested removing the colored bar while keeping all text feedback. Removing the colored bar caused participants to be more likely to return for Part 2 of the study ($p = .020$), but did not impact any other objective security or usability metrics. Removing the colored bar did impact participant sentiment, however. Participants who did not see a bar found password creation more annoying and difficult (both $p < .001$).

DISCUSSION AND DESIGN RECOMMENDATIONS

We described our design and evaluation of a password meter that provides detailed, data-driven feedback. Using a combination of neural networks and 21 carefully combined heuristics to score passwords, as well as giving users detailed text explanations of what parts of their password are predictable, our meter gives users more accurate and more actionable information.

We found that our password-strength meter made 1class8 passwords harder to guess without significantly impacting memorability. Text feedback led to more secure 1class8 passwords than a colored bar alone, whereas colored bars alone are the type of meter widely deployed today [10]. Notably, leaving the detailed text feedback but removing the colored bar did not significantly impact the security of the passwords participants created. Combined with our finding that most people do not feel compelled to fill the bar, this suggests that the visual metaphor has only marginal impact when detailed text feedback is also present. As a result, we highly recommend the use of a meter that provides detailed text feedback for common password-composition policies like 1class8. From our results, we recommend that the meter offer potentially sensitive feedback when the user shows his or her password on screen. While much of this text feedback might be redundant for power users, they are free to ignore it.

The suggested improvement did seem to help some participants, but its overall effect was not strong and some participants did not trust suggestions from a computer. While its inclusion does not seem to hurt, we would consider it optional. Similarly, although the generic-advice modal was visited more than the specific-advice modal, only a fraction of participants looked at it. Because not all users need to learn the basics of making strong passwords [41], it is reasonable that only a handful of users would need to engage with these features. We thus recommend that they be included.

In contrast to prior work that found scoring stringency to be crucial for password meters [42], we only observed a significant security effect for 3class12 passwords, and the effect size was small. Note that our meter used far more advanced methods to score passwords more accurately than the basic heuristics tested in that prior work. Because the high-stringency setting negatively impacted some usability metrics, we recommend our medium setting.

Our recommendations differ for 3class12 passwords. The meter had minimal impact on the security of 3class12 passwords. While the meter introduced few usability disadvantages, suggesting that it may not hurt to include the meter, we would not recommend it nearly as strongly as for 1class8 passwords.

While the studies we report in this paper are a crucial first step in pinpointing the impact of meter design dimensions and configurations, the next step is to investigate these effects further in a field study. Such a study could improve ecological validity, allow for expanded testing of password memorability, and enable comparisons with currently deployed meters.

To spur adoption of data-driven password meters, we are releasing our meter's code open-source.[2]

ACKNOWLEDGMENTS

This work was supported in part by a gift from the PNC Center for Financial Services Innovation, NSF grant CNS-1012763, and by John & Claire Bertucci.

REFERENCES

1. Steven Van Acker, Daniel Hausknecht, Wouter Joosen, and Andrei Sabelfeld. 2015. Password meters and generators on the web: From large-scale empirical study to getting it right. In *Proc. CODASPY*.

2. Anne Adams, Martina Angela Sasse, and Peter Lunt. 1997. Making passwords secure and usable. In *Proc. HCI on People and Computers*.

3. Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B* 57, 1 (1995), 289–300.

4. Joseph Bonneau. 2012. The science of guessing: Analyzing an anonymized corpus of 70 million

[2] Source code: https://github.com/cupslab/password_meter

passwords. In *Proc. IEEE Symposium on Security and Privacy*.

5. Joseph Bonneau and Ekaterina Shutova. 2012. Linguistic properties of multi-word passphrases. In *Proc. USEC*.
6. Mark Burnett. 2015. Today I am releasing ten million passwords. https://xato.net/today-i-am-releasing-ten-million-passwords-b6278bbe7495#.s11zbdb8q. (February 9, 2015).
7. Carnegie Mellon University. 2015. Password Guessability Service. https://pgs.ece.cmu.edu. (2015).
8. Claude Castelluccia, Markus Dürmuth, and Daniele Perito. 2012. Adaptive password-strength meters from Markov models. In *Proc. NDSS*.
9. Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and XiaoFeng Wang. 2014. The tangled web of password reuse. In *Proc. NDSS*.
10. Xavier de Carné de Carnavalet and Mohammad Mannan. 2014. From very weak to very strong: Analyzing password-strength meters. In *Proc. NDSS*.
11. Matteo Dell'Amico and Maurizio Filippone. 2015. Monte Carlo strength evaluation: Fast and reliable password checking. In *Proc. CCS*.
12. Serge Egelman, Andreas Sotirakopoulos, Ildar Muslukhov, Konstantin Beznosov, and Cormac Herley. 2013. Does my password go up to eleven? The impact of password meters on password selection. In *Proc. CHI*.
13. Sascha Fahl, Marian Harbach, Yasemin Acar, and Matthew Smith. 2013. On the ecological validity of a password study. In *Proc. SOUPS*.
14. Dinei Florêncio and Cormac Herley. 2007. A large-scale study of web password habits. In *Proc. WWW*.
15. Dinei Florêncio, Cormac Herley, and Paul C. van Oorschot. 2014. Password portfolios and the finite-effort user: Sustainably managing large numbers of accounts. In *Proc. USENIX Security*.
16. Alain Forget, Sonia Chiasson, P.C. van Oorschot, and Robert Biddle. 2008. Improving text passwords through persuasion. In *Proc. SOUPS*.
17. John Fox and Sanford Weisberg. 2011. *An R companion to applied regression (online appendix)* (second ed.). Sage Publications. https://socserv.socsci.mcmaster.ca/jfox/Books/Companion/appendix/Appendix-Cox-Regression.pdf.
18. Dan Goodin. 2012. Hackers expose 453,000 credentials allegedly taken from Yahoo service. *Ars Technica*. (July 2012). http://arstechnica.com/security/2012/07/yahoo-service-hacked/.
19. Dan Goodin. 2013. "thereisnofatebutwhatwemake"-Turbo-charged cracking comes to long passwords. *Ars Technica*. (August 2013). http://arstechnica.com/security/2013/08/thereisnofatebutwhatwemake-turbo-charged-cracking-comes-to-long-passwords/.
20. Cormac Herley. 2009. So long, and no thanks for the externalities: The rational rejection of security advice by users. In *Proc. NSPW*.
21. Jun Ho Huh, Seongyeol Oh, Hyoungshick Kim, Konstantin Beznosov, Apurva Mohan, and S. Raj Rajagopalan. 2015. Surpass: System-initiated user-replaceable passwords. In *Proc. CCS*.
22. Troy Hunt. 2011. The science of password selection. Blog post. (July 2011). http://www.troyhunt.com/2011/07/science-of-password-selection.html.
23. Imperva. 2010. Consumer password worst practices. (2010). http://www.imperva.com/docs/WP_Consumer_Password_Worst_Practices.pdf.
24. Philip Inglesant and M. Angela Sasse. 2010. The true cost of unusable password policies: Password use in the wild. In *Proc. CHI*.
25. Blake Ives, Kenneth R. Walsh, and Helmut Schneider. 2004. The domino effect of password reuse. *CACM* 47, 4 (April 2004), 75–78.
26. Markus Jakobsson and Mayank Dhiman. 2012. The benefits of understanding passwords. In *Proc. HotSec*.
27. Saranga Komanduri, Richard Shay, Lorrie Faith Cranor, Cormac Herley, and Stuart Schechter. 2014. Telepathwords: Preventing weak passwords by reading users' minds. In *Proc. USENIX Security*.
28. Saranga Komanduri, Richard Shay, Patrick Gage Kelley, Michelle L. Mazurek, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Serge Egelman. 2011. Of passwords and people: Measuring the effect of password-composition policies. In *Proc. CHI*.
29. Cynthia Kuo, Sasha Romanosky, and Lorrie Faith Cranor. 2006. Human selection of mnemonic phrase-based passwords. In *Proc. SOUPS*.
30. David Malone and Kevin Maher. 2012. Investigating the distribution of password choices. In *Proc. WWW*.
31. Michelle L. Mazurek, Saranga Komanduri, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Patrick Gage Kelley, Richard Shay, and Blase Ur. 2013. Measuring password guessability for an entire university. In *Proc. CCS*.
32. William Melicher, Blase Ur, Sean M. Segreti, Saranga Komanduri, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2016. Fast, lean, and accurate: Modeling password guessability using neural networks. In *Proc. USENIX Security*.
33. Bruce Schneier. 2014. Choosing secure passwords. Schneier on Security https://www.schneier.com/blog/archives/2014/03/choosing_secure_1.html. (March 3, 2014).
34. Richard Shay, Saranga Komanduri, Adam L. Durity, Phillip (Seyoung) Huh, Michelle L. Mazurek, Sean M.

Segreti, Blase Ur, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2014. Can long passwords be secure and usable?. In *Proc. CHI*.

35. Dawn Xiaodong Song, David Wagner, and Xuqing Tian. 2001. Timing analysis of keystrokes and timing attacks on SSH. In *Proc. USENIX Security Symposium*.

36. Andreas Sotirakopoulos, Ildar Muslukov, Konstantin Beznosov, Cormac Herley, and Serge Egelman. 2011. Motivating users to choose better passwords through peer pressure. In *Proc. SOUPS (Poster Abstract)*.

37. Jeffrey M. Stanton, Kathryn R. Stam, Paul Mastrangelo, and Jeffrey Jolton. 2005. Analysis of end user security behaviors. *Comp. & Security* 24, 2 (2005), 124–133.

38. Elizabeth Stobert and Robert Biddle. 2014. The password life cycle: User behaviour in managing passwords. In *Proc. SOUPS*.

39. Stricture Consulting Group. 2015. Password audits. http://stricture-group.com/services/password-audits.htm. (2015).

40. Blase Ur. 2016. *Supporting password-security decisions with data*. Ph.D. Dissertation. Carnegie Mellon University. CMU-ISR-16-110 http://www.blaseur.com/phdthesis.pdf.

41. Blase Ur, Jonathan Bees, Sean M. Segreti, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2016. Do users' perceptions of password security match reality?. In *Proc. CHI*.

42. Blase Ur, Patrick Gage Kelly, Saranga Komanduri, Joel Lee, Michael Maass, Michelle Mazurek, Timothy Passaro, Richard Shay, Timothy Vidas, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2012. How does your password measure up? The effect of strength meters on password creation. In *Proc. USENIX Security*.

43. Blase Ur, Fumiko Noma, Jonathan Bees, Sean M. Segreti, Richard Shay, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2015a. "I added '!' at the end to make it secure": Observing password creation in the lab. In *Proc. SOUPS*.

44. Blase Ur, Sean M. Segreti, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Saranga Komanduri, Darya Kurilova, Michelle L. Mazurek, William Melicher, and Richard Shay. 2015b. Measuring real-world accuracies and biases in modeling password guessability. In *Proc. USENIX Security*.

45. Ashlee Vance. 2010. If your password is 123456, just make it HackMe. New York Times, http://www.nytimes.com/2010/01/21/technology/21password.html. (2010).

46. Rafael Veras, Christopher Collins, and Julie Thorpe. 2014. On the semantic patterns of passwords and their security impact. In *Proc. NDSS*.

47. Rafael Veras, Julie Thorpe, and Christopher Collins. 2012. Visualizing semantics in passwords: The role of dates. In *Proc. VizSec*.

48. Emanuel von Zezschwitz, Alexander De Luca, and Heinrich Hussmann. 2013. Survival of the shortest: A retrospective analysis of influencing factors on password composition. In *Proc. INTERACT*.

49. Emanuel von Zezschwitz, Alexander De Luca, and Heinrich Hussmann. 2014. Honey, I shrunk the keys: Influences of mobile devices on password composition and authentication performance. In *Proc. NordiCHI*.

50. Kim-Phuong L. Vu, Robert W. Proctor, Abhilasha Bhargav-Spantzel, Bik-Lam (Belin) Tai, and Joshua Cook. 2007. Improving password security and memorability to protect personal and organizational information. *IJHCS* 65, 8 (2007), 744–757.

51. Dan Wheeler. 2012. zxcvbn: Realistic password strength estimation. https://blogs.dropbox.com/tech/2012/04/zxcvbn-realistic-password-strength-estimation/. (2012).

52. Dan Lowe Wheeler. 2016. zxcvbn: Low-budget password strength estimation. In *Proc. USENIX Security*.

53. Yulong Yang, Janne Lindqvist, and Antti Oulasvirta. 2014. Text entry method affects password security. In *Proc. LASER*.