# Can Long Passwords Be Secure and Usable?

**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**

Carnegie Mellon University
Pittsburgh, PA
{rshay, sarangak, adurity, phuh, mmazurek, ssegreti, bur, lbauer, nicolasc, lorrie}@cmu.edu

## ABSTRACT

To encourage strong passwords, system administrators employ password-composition policies, such as a traditional policy requiring that passwords have at least 8 characters from 4 character classes and pass a dictionary check. Recent research has suggested, however, that policies requiring longer passwords with fewer additional requirements can be more usable and in some cases more secure than this traditional policy. To explore long passwords in more detail, we conducted an online experiment with 8,143 participants. Using a cracking algorithm modified for longer passwords, we evaluate eight policies across a variety of metrics for strength and usability. Among the longer policies, we discover new evidence for a security/usability tradeoff, with none being strictly better than another on both dimensions. However, several policies are both more usable and more secure that the traditional policy we tested. Our analyses additionally reveal common patterns and strings found in cracked passwords. We discuss how system administrators can use these results to improve password-composition policies.

### ACM Classification Keywords

D.4.6 Management Of Computing and Information Systems: Security and Protection—*Authentication*

### Author Keywords

Passwords; Password-composition policies; Security policy; Usable security; Authentication

## INTRODUCTION

Reports of stolen password databases have become commonplace in recent years [4, 8, 13], prompting additional concern over the security of users' passwords. Password-composition policies, which dictate requirements about password length and composition, are often used to guide users to create passwords that are harder to crack in the event of a breach. Researchers have found that policies requiring long passwords with fewer requirements can be more usable and in some circumstances more secure than a conventional "strong" policy [21, 22]. However, the balance between security and usability in policies requiring longer passwords has not previously been investigated.

Our primary contribution is expanding upon the limited finding from prior work – that requiring sixteen-character passwords can be more usable and sometimes more secure than traditional policies – to provide tangible, concrete advice for system administrators on requiring long passwords. In particular, we are the first to contrast variants of password-composition policies with longer length requirements. In addition, we are the first to offer concrete recommendations about policies with longer length requirements, placing these policies along a security/usability spectrum.

Previous studies [21, 22] compared only a length-16 requirement with traditional complex policies. We tested numerous variations on the length-16 requirement: fewer characters, more characters, required character classes, etc. Moreover, while passwords created under the length-16 policy were often stronger after trillions of guesses than those created under the traditional policy, there were also very simple passwords, like *passwordpassword*, that could be guessed easily. Could these weak passwords be prevented without burdening users?

In this paper, we provide the first evaluation of password policies focused on long, user-selected passwords. Our 8,143 online participants created a password under one of eight policies. We examined a comprehensive length-8 policy, three variants of length-12 policies with fewer requirements, three variants of length-16 policies, and a length-20 policy without additional requirements. We searched for a policy that would offer the strength benefits of the traditional length-8 comprehensive policy without its usability problems.

We found that adding requirements to policies on longer passwords can reduce the number of easily guessed passwords, and that certain combinations of requirements were both stronger and more usable than the traditional complex policy. We also identified patterns and strings commonly found in cracked passwords. For example, 43.6% of the passwords containing the string *1234* were cracked, while only 13.9% of passwords without this string were cracked. These patterns can be used proactively to identify weak passwords.

We begin by discussing previous research in the next section. Then, we provide details of our methodology and discuss its



*CHI 2014*, Apr 26 - May 01 2014, Toronto, ON, Canada
ACM 978-1-4503-2473-1/14/04.
**http://dx.doi.org/10.1145/2556288.2557377**

limitations. Afterward, we present our security and usability results, as well as the results of our analysis of password patterns. Finally, we discuss our findings, their implications, and how system administrators might translate these results into stronger and more usable password policies in practice.

## BACKGROUND AND RELATED WORK

Although text passwords have a number of drawbacks [6, 12] and researchers have spent decades investigating password alternatives [2], text passwords are here to stay for the forseeable future [16, 17]. One problem with text passwords is that users tend to create weak passwords [3, 5, 14, 29, 31]. Administrators address this problem by mandating password policies with requirements they believe will lead users to create stronger passwords. Such a policy might require that a password be at least eight characters long and contain both letters and digits. Researchers have found that password policies help users create passwords that are generally harder to crack than those created without composition requirements [11, 25, 32]. Unfortunately, users are frustrated by inflexible password-composition and password-management requirements [1, 19], and they often fulfill requirements in predictable ways [9, 28, 33, 35].

Most password-composition policies specify a minimum length, as well as requirements for the different character classes that must be included. The United States National Institutes of Standards and Technology (NIST) has estimated that a password of 8 characters containing 4 different character classes and passing a dictionary check – a password consistent with the policy we call comp8 – has about 30 bits of entropy. This policy is one of the main examples of a password policy in the 2006 version of NIST's "electronic authentication guidelines" [10]. The InCommon Federation has adopted the NIST guidelines as part of their authentication requirements for member universities [18].

Researchers have found that requiring passwords with at least 16 characters, even without further composition requirements, has both usability and security benefits over requiring 8-character passwords that must contain many character classes. However, some users created passwords that were very easy to guess when password length was the only composition requirement. [21, 22].

## METHODOLOGY

We conducted a two-part online study to examine how participants create and use passwords under various policies. In the first part, we asked participants to create a password under a given policy, fill out a survey, and recall their password. Two days later, we emailed our participants, asking them to return. When they returned, we asked them to recall their password again and administered a second survey.

### Study Overview

We recruited participants through Amazon's Mechanical Turk crowdsourcing service (MTurk). Participants needed to be at least 18 years old and located in the United States. Our overall methodology is based on techniques that have been used to compare policies in prior work [21, 22, 27, 30].

In *part one* of our study, we asked participants to imagine that their email provider had been attacked and required they create a new password using password rules specified by one of our conditions. We informed them that they would be asked to return to recall their password again in a few days, and asked them to take whatever steps they normally take to remember and protect their password. Prior work has suggested that asking participants to imagine creating a password for their email account leads to stronger passwords than simply asking them to create passwords for a study [21, 22].

We then showed participants one of eight sets of password-creation instructions, depending on condition. After successfully choosing and confirming a password, participants completed a five-minute survey about their experience. We then asked participants to recall their password (termed *part one recall* in this paper). If the participant did not enter the password successfully in five attempts, we displayed it on screen.

Two days later, we invited participants through MTurk to return for *part two* of the study. We asked participants to recall their password (*part two recall*). Again, participants who entered five incorrect passwords were shown their password on screen. Further, participants could follow a "Forgot Password" link to be emailed a link to their password. After this, we administered another five-minute survey about whether and how participants stored their passwords.

Except when looking at dropout rates, our analysis focuses on data from participants who completed part two within three days of receiving the invitation to return. Participants who took longer were still paid, but not included in the analysis.

Our data collection affords us the following usability metrics. To examine creation and recall usability, we collect timing information and the number of attempts to create the password and recall it after both a few minutes and a few days. We present how many participants dropped out before finishing part one, and how many finished part two within three days of being invited to return as a measure of user frustration. We look at password storage rates and usage of a password reminder feature during part two recall, assuming that storage and use of password recall indicates decreased usability. We also directly ask participants about their sentiment on password creation and recall in our study.

### Conditions

We assigned participants to one of eight conditions, each with different password-composition requirements and different instructions to reflect those requirements. Because of the large number of factors used between our conditions, it is not feasible to test all of our factors in isolation and in all combinations. Instead, we carefully chose a set of conditions we felt combined factors in order to balance security and usability. We included the comp8 condition, similar to that used in practice at our institution, the longer basic20 condition, as well as 12- and 16-length conditions with other factors.

- **comp8** Passwords in this condition must include "at least 8 characters," including a "lowercase English letter," "uppercase English letter," "digit," and "symbol (something that is not a digit or an English letter)." Participants were also

told, "Taken together, the letters must not form a word in our dictionary." For the dictionary check, we used the free Openwall cracking dictionary.[1]

- **basic12, basic16, basic20** Participants were told only to include at least 12, 16, or 20 characters. In previous research, the basic16 policy was found to be more secure and usable than the comp8 policy described above. This set of conditions varies only in length, so we can measure its impact on security and usability.

- **2word12, 2word16** These passwords required at least 12 or 16 characters and needed to include "at least two words (letter sequences separated by a non-letter sequence)." This required mixing letter and non-letter characters, and by mentioning words, we encouraged participants to create passphrases, which previous research has suggested may be more memorable than passwords [20].

- **3class12, 3class16** Passwords in these conditions required at least 12 or 16 characters. Participants were also asked to include at least three of the four character classes required by comp8. This condition was designed to encourage diversity among character types in the password.

Password policies sometimes prohibit characters such as semicolons or spaces. We did not prohibit these characters, and told all participants, "You may use letters, numbers, spaces, and other symbols in your password."

### Measuring Password Strength

To evaluate password strength we use a variety of metrics. Primarily, we focus on how vulnerable passwords are to an offline guessing attack. We also compute password-composition characteristics such as average length, number of symbols for each condition, unique structures within passwords, and Shay et al. entropy [28].

To measure the vulnerability of passwords to a guessing attack, we use a modified version of the algorithm developed by Weir et al. and refined by Kelley et al. [21,34]. This algorithm uses a corpus of training data to generate guesses in order of likelihood, up to some cutoff. We use training data that includes publicly available dictionaries (including the Google web corpus and the Openwall cracking dictionary); leaked password sets that were previously made public (including MySpace and RockYou); and data from previous online studies. Since some participants did not return for part 2 of the study, we also use their passwords for training, weighted to prefer these passwords over other sources of training data.

The original cracking algorithm is not well suited for cracking the long passwords created by participants in this study. Long strings of letters, such as "thisisapassword," would only be cracked if the same string appeared in the training data. To make the algorithm better able to crack longer passwords, we make several improvements. First, we tokenize all passwords using a word-level n-gram model based on the Google Web Corpus [7]. This breaks up long alphabetic strings into sequences of digits, symbols, and words. Second, we learn string frequencies from the training data and include all strings from the Google Web Corpus with associated frequencies. This improves the effectiveness of the guessing algorithm by favoring high-probability strings, but also increases computational and memory requirements due to the large increase in number of strings used for training. We mitigate this by quantizing probabilities, trading accuracy for speed as suggested by Narayanan and Shmatikov [24]. The total mean-squared error, a standard measure of quantization error, was on the order of $10^{-9}$ for all conditions.

The accuracy of our guessing results depends on the amount and quality of training data available to the guessing algorithm. We had access to limited amounts of training data for the policies we examined in this study, and we had more data for some conditions than others. Thus, cracking performance might improve significantly if data for these policies were readily available. A realistic advantage of a novel password policy is that an attacker would have less training data. Such a benefit might be temporary; if more service providers switched to 2word16, this might lead to more such passwords in leaked password sets, in turn providing more training data for attackers. However, at least in the short term, this advantage of more obscure policies would remain.

### Statistical Testing

Our statistical tests use a significance level of $\alpha = .05$. For each omnibus comparison on quantitative data, we used Kruskal-Wallis (KW), an analogue of ANOVA that does not assume normality. For omnibus tests on categorical data, we used $\chi^2$. If the omnibus test was significant, we performed pairwise tests with Holm-Bonferroni correction (HC) to find significant differences between conditions. We used Mann-Whitney U (MW) for pairwise quantitative comparisons and Fisher's Exact Test (FET) and the Chi Square test (CS) for pairwise categorical comparisons.

## PARTICIPANTS

We recruited participants between April and June 2013. Participants received 55 cents for the first part of our study and 70 cents for the second. Of the 15,108 participants who began our study, 13,751 finished part one, 8,565 returned for part two within three days of receiving our invitation to return, and 8,143 finished part two of the study within three days of receiving that invitation. Other than the discussion of dropout rates, our analysis focuses only on the 8,143 participants who finished the entire study. The number of participants per condition is shown in Table 1.

51.2% of participants reported being male, 47.8% female, and the remaining 1% declined to answer. Participants' mean age was 29.9 years (median 27). These did not vary significantly between conditions. We looked at user-agent strings to detect mobile users; 1.4% of participants appeared to be using mobile devices.

## SECURITY RESULTS

When considering an attacker who can make over a trillion guesses, we find that all conditions except basic12 are stronger than comp8. The 2word and 3class conditions are

[1] `http://www.openwall.com/wordlists/`

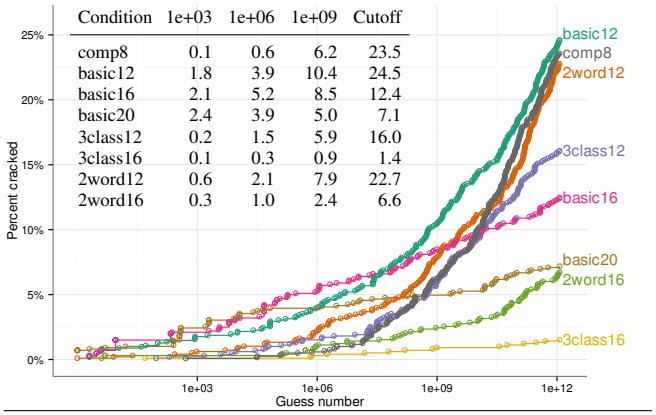


**Figure 1. The percentage of passwords cracked in each condition by the number of guesses made in log scale. Our cutoff for guess numbers was $\approx 1.16 \times 10^{12}$. The inset table shows the percentage of passwords cracked after various numbers of guesses.**

stronger than their basic counterparts of equal length, but adding the 2word requirement improves basic16 more than basic12. Consistent with prior work, we also find that for small numbers of guesses, comp8 performs relatively well, and in particular, better than the basic conditions. We begin this section with some statistics on our passwords and then present our cracking results.

### Password Statistics

The entropy per condition and other password statistics are shown in Table 1. We find that users typically avoid using uppercase letters or symbols in their passwords, but often include digits even when not required. Entropy increased with length requirements. The 2word requirement added additional entropy and the 3class requirement added even more.

| Condition | Participants | Length | Uppercase | Lowercase | Digits | Symbols | Entropy |
|---|---|---|---|---|---|---|---|
| comp8 | 1200 | 10 | 1 | 5 | 2 | 1 | 34.00 |
| basic12 | 1019 | 13 | 0 | 9 | 3 | 0 | 39.88 |
| basic16 | 1001 | 17 | 0 | 14 | 2 | 0 | 47.93 |
| basic20 | 988 | 21 | 0 | 18 | 3 | 0 | 56.54 |
| 3class12 | 993 | 13 | 1 | 8 | 3 | 1 | 41.82 |
| 3class16 | 983 | 17 | 1 | 11 | 3 | 1 | 51.81 |
| 2word12 | 978 | 14 | 0 | 11 | 2 | 0 | 40.88 |
| 2word16 | 981 | 18 | 0 | 14 | 2 | 1 | 49.26 |

**Table 1. This table contains password-composition statistics for each condition. The first column shows the number of participants in each condition. The median length, number of each character class, and estimated entropy follow.**

### Password Cracking

The percentage of passwords cracked in each condition as additional guesses are made is graphed in Figure 1. Condition 3class16 is the strongest across the range of guesses. Some other conditions that appear strong if we look only at a large number of guesses are relatively weak if we consider a smaller number of guesses. For example, at the cutoff, basic20 and basic16 are the third and fourth strongest conditions, respectively. However, in the face of an attacker able to make only a million guesses, they are among the weakest conditions. This suggests that while passwords created under basic conditions can be relatively strong overall, they also contain a non-trivial fraction of weak passwords.

The comp8 condition is relatively strong against a resource-limited attacker, with only a few passwords cracked until after a million guesses. However, its curve begins to ascend rapidly after a million guesses. At the cutoff, comp8 offers more protection than only basic12, and is fairly close to 2word12. In comparison, 3class12 is similar in strength until around $10^{10}$, and remains more resistant to cracking from that point on.

It is interesting to note the disparity between 2word12 and 2word16. While 2word12 is more vulnerable to early guessing than comp8 and does not offer much more protection than comp8 overall, 2word16 is our second strongest condition. The 2word approach seems to be more effective at increasing password strength when combined with a length-16 requirement than with a length-12 requirement. Manually examining the passwords users created in these conditions, we see that some users actually created passwords with three words rather than two, and these passwords tended to be more resistant to cracking. We found that 2word16 users created three-word passwords 31.8% of the time, and were almost twice as likely as 2word12 users to create three-word passwords. Only 2.6% of the three-word 2word16 passwords were cracked, as compared with 8.5% of two-word 2word16 passwords.

After a million guesses, the three basic conditions each have significantly more passwords cracked than comp8, 2word16, and the 3class conditions. 2word12 has fewer passwords cracked than basic16, but more cracked than comp8 or 3class16 (HC FET, $p<.025$).

At the cutoff of around $1.16 \times 10^{12}$ guesses, we see a different ordering for strength. Each of basic12, comp8, and 2word12 have significantly more passwords cracked than any of the other five conditions (HC FET, $p<.002$). Further, 3class16 performs significantly better than any other condition, and both basic20 and 2word16 perform significantly better than any condition beside themselves and 3class16 (HC FET, $p<.001$).

## USABILITY RESULTS

In this section we examine dropout rates, password storage, password creation, and recall. Overall, we find that most conditions are significantly more usable than comp8 on a number of metrics, with only basic20 and 3class16 being significantly less usable on any metric. We also find that many participants fail to create a compliant password on their first try, suggesting that simple real-time feedback might benefit users who are required to create long passwords.

### Study Dropout

Among 15,108 participants who began the study, 91.0% finished part one. Dropout rates varied significantly by condition ($\chi^2_7$=246.60, $p<.001$), ranging from 83.0% for comp8 to 94.5% for basic12. Participants in comp8 were significantly less likely to finish part one than those in any other condition (HC $\chi^2$, $p<.001$). Participants in 3class16 (90.5%) were

| Condition | % Storage | % of No Storage | | % of Storage | |
|---|---|---|---|---|---|
| | | 5 tries | 1st try | 5 tries | 1st try |
| comp8 | 56.9 | 75 | 56 | 84 | 76 |
| basic12 | 45.4 | 75 | 61 | 86 | 76 |
| basic16 | 49.9 | 79 | 64 | 85 | 75 |
| basic20 | 50.0 | 77 | 64 | 86 | 73 |
| 3class12 | 54.9 | 74 | 54 | 86 | 74 |
| 3class16 | 60.2 | 73 | 51 | 84 | 72 |
| 2word12 | 51.4 | 74 | 59 | 85 | 76 |
| 2word16 | 51.3 | 71 | 55 | 83 | 73 |

**Table 2. The percentage classified as *storage* participants for each condition is given in the first column. The remaining columns pertain to part two recall, with results listed separately for no-storage and storage participants. The second and third columns list the percentage of no-storage participants who successfully entered their passwords in five and in one try without using the password reminder. The fourth and fifth list this for storage participants.**

significantly less likely to finish part one than those in basic12 (94.5%) or basic16 (93.9%) (HC $\chi^2$, $p<.004$).

Of those participants invited to return for part two of our study, 62.3% returned within three days of being invited back; this did not vary significantly by condition ($\chi^2_7$=7.69, $p$=0.361). Of those who returned for part two, 95.1% completed part two within three days of being invited back; this also did not vary significantly by condition ($\chi^2_7$=4.15, $p$=0.762). The number of participants finishing part two in each condition is shown in Table 1.

### Password Storage

To analyze storage, we first classify participants into two groups: *storage* and *non-storage* participants. To be considered a non-storage participant, the participant must tell us the password was not stored in two separate questions in the part-two survey, and not be detected pasting or using browser autocomplete in part-two recall, except after returning via the password-reminder link.

The percentage of storage participants per condition is shown in the first column of Table 2. Significant pairwise comparisons are shown in Table 3. Overall, 3class16 had a significantly higher storage rate than every other condition except comp8 and 3class12. Password storage rates were highest in conditions that required three or four character classes, and lowest in the three basic conditions.

### Password Creation

We examined both the number of attempts participants needed to create a password, and their sentiment about the password-creation process. On average, participants needed 1.8 attempts to create a password; significant pairwise differences are shown in Table 3. While comp8 required the most attempts, (M=2.3), basic12 required the fewest (M=1.5).

We asked participants whether they agreed with the statement, "Creating a password that meets the requirements given in this study was difficult." Responses are depicted in Figure 2, with significant differences shown in Table 3. Participants were most likely to find it difficult to create passwords

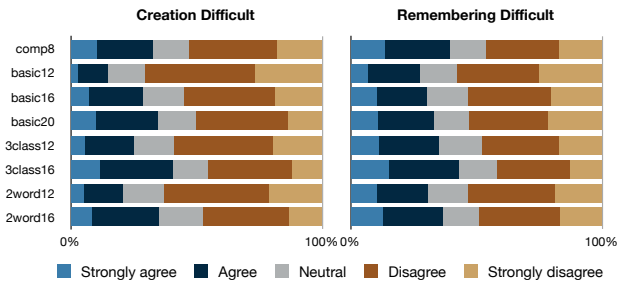


**Figure 2. Participant agreement with "Creating a password that meets the requirements given in this study was difficult" and "Remembering the password I used for this study was difficult."**

| Condition | Failed (%) | Confirm (%) | Length (%) | Classes (%) | Dict. (%) | 2word (%) |
|---|---|---|---|---|---|---|
| comp8 | 53.8 | 4.4 | 5.9 | 22.6 | 37.7 | – |
| basic12 | 37.2 | 3.4 | 34.6 | – | – | – |
| basic16 | 49.5 | 3.5 | 47.5 | – | – | – |
| basic20 | 58.2 | 4.5 | 56.1 | – | – | – |
| 3class12 | 41.1 | 5.4 | 36.0 | 7.2 | – | – |
| 3class16 | 48.0 | 6.3 | 42.9 | 7.8 | – | – |
| 2word12 | 50.1 | 4.5 | 27.3 | 8.0 | – | 40.5 |
| 2word16 | 56.6 | 4.6 | 41.7 | 9.7 | – | 43.5 |

**Table 4. The first column shows the total percent who failed to create a compliant password on the first attempt. All numbers in this table are percentages of the total participants in each condition. The remaining columns show how participants failed, and participants could fail in more than one way. Confirm indicates a difference between password and confirmation. Cells where a requirement was not applicable for the policy are marked with –. Omitted from this table are failures due to a blank password or confirmation field, less than 1% in any condition.**

under 3class16, followed by 2word16 and basic20. Passwords were reported as easiest to create under basic12, followed by 2word12.

For a better understanding of password-creation failures, we looked at participants' first failed password-creation attempt. We find that participants often failed to meet length or character-class requirements, simple requirements that could easily have been checked in real-time with client-side code. These results highlight the need for feedback regarding requirements during the password-creation process.

The different ways that participants failed in their first attempt are shown in Table 4. Many participants failed to meet the length requirement, with 56.1% of participants in basic20 using less than 20 characters. 22.6% of participants in comp8 used too few character classes, compared to between seven and ten percent of participants in other conditions that required non-letter characters. This suggests that participants struggle more to create a password with four classes compared to three. Finally, the largest cause of failure was the dictionary check in comp8 and the 2word requirement in the 2word conditions. A simpler dictionary check might have resulted in fewer failures, as might have increased familiarity over time with 2word requirements. While only comp8 had a dictionary check, we looked at how many passwords in other conditions would have been prevented by that check. This is 22.3% of passwords in 3class12, 18.7% in basic12, and less than 10% in any other condition.

**Password creation attempts**
*Omnibus* KW $\chi^2_7$=230.34, $p$<.001

| cond.1 | mean | cond.2 | mean | $p$-value |
|---|---|---|---|---|
| comp8 | 2.3 | 3class16 | 1.8 | <.001 |
| | | 2word12 | 1.8 | <.001 |
| | | basic16 | 1.7 | <.001 |
| | | 3class12 | 1.6 | <.001 |
| | | basic12 | 1.5 | <.001 |
| 2word16 | 2.0 | 3class16 | 1.8 | .001 |
| | | 2word12 | 1.8 | .001 |
| | | basic16 | 1.7 | <.001 |
| | | 3class12 | 1.6 | <.001 |
| | | basic12 | 1.5 | <.001 |
| basic20 | 1.9 | 3class16 | 1.8 | .005 |
| | | 2word12 | 1.8 | .005 |
| | | basic16 | 1.7 | <.001 |
| | | 3class12 | 1.6 | <.001 |
| | | basic12 | 1.5 | <.001 |
| 3class16 | 1.8 | 3class12 | 1.6 | <.001 |
| | | basic12 | 1.5 | <.001 |
| 2word12 | 1.8 | 3class12 | 1.6 | <.001 |
| | | basic12 | 1.5 | <.001 |
| basic16 | 1.7 | 3class12 | 1.6 | <.001 |
| | | basic12 | 1.5 | <.001 |

**Attempts by successful no-store**
*Omnibus* KW $\chi^2_7$=27.06, $p$<.001

| cond.1 | mean | cond.2 | mean | $p$-value |
|---|---|---|---|---|
| 3class16 | 1.4 | basic12 | 1.3 | .029 |
| | | basic16 | 1.3 | .029 |
| | | basic20 | 1.3 | .004 |

**Agree password creation difficult**
*Omnibus* $\chi^2_7$=239.44, $p$<.001

| cond.1 | % | cond.2 | % | $p$-value |
|---|---|---|---|---|
| 3class16 | 40.5 | comp8 | 32.6 | .002 |
| | | basic16 | 28.9 | <.001 |
| | | 3class12 | 25.4 | <.001 |
| | | 2word12 | 20.9 | <.001 |
| | | basic12 | 14.9 | <.001 |
| 2word16 | 35.2 | basic16 | 28.9 | .028 |
| | | 3class12 | 25.4 | <.001 |
| | | 2word12 | 20.9 | <.001 |
| | | basic12 | 14.9 | <.001 |
| basic20 | 34.7 | basic16 | 28.9 | .047 |
| | | 3class12 | 25.4 | <.001 |
| | | 2word12 | 20.9 | <.001 |
| | | basic12 | 14.9 | <.001 |
| comp8 | 32.6 | 3class12 | 25.4 | .003 |
| | | 2word12 | 20.9 | <.001 |
| | | basic12 | 14.9 | <.001 |
| basic16 | 28.9 | 2word12 | 20.9 | <.001 |
| | | basic12 | 14.9 | <.001 |
| 3class12 | 25.4 | basic12 | 14.9 | <.001 |
| 2word12 | 20.9 | basic12 | 14.9 | .006 |

**Storage participants**
*Omnibus* $\chi^2_7$=61.87, $p$<.001

| cond.1 | % | cond.2 | % | $p$-value |
|---|---|---|---|---|
| 3class16 | 60.2 | 2word12 | 51.4 | .002 |
| | | 2word16 | 51.3 | .002 |
| | | basic20 | 50.0 | <.001 |
| | | basic16 | 49.9 | <.001 |
| | | basic12 | 45.4 | <.001 |
| comp8 | 56.9 | basic20 | 50.0 | .029 |
| | | basic16 | 49.9 | .02 |
| | | basic12 | 45.4 | <.001 |
| 3class12 | 54.9 | basic12 | 45.4 | <.001 |

**Password entry time (s)**
*Omnibus* KW $\chi^2_7$=71.58, $p$<.001

| cond.1 | median | cond.2 | median | $p$-value |
|---|---|---|---|---|
| 3class16 | 16.2 | comp8 | 13.2 | .001 |
| | | 2word12 | 13.1 | <.001 |
| | | basic12 | 11.6 | <.001 |
| basic20 | 15.3 | comp8 | 13.2 | <.001 |
| | | 2word12 | 13.1 | <.001 |
| | | basic12 | 11.6 | <.001 |
| 3class12 | 14.8 | basic12 | 11.6 | .001 |
| 2word16 | 14.6 | 2word12 | 13.1 | .012 |
| | | basic12 | 11.6 | <.001 |
| basic16 | 13.7 | basic12 | 11.6 | .003 |

**Agree remembering password difficult**
*Omnibus* $\chi^2_7$=83.89, $p$<.001

| cond.1 | % | cond.2 | % | $p$-value |
|---|---|---|---|---|
| 3class16 | 42.9 | 3class12 | 35.3 | .012 |
| | | basic20 | 32.9 | <.001 |
| | | 2word12 | 31.0 | <.001 |
| | | basic16 | 30.1 | <.001 |
| | | basic12 | 27.4 | <.001 |
| comp8 | 39.3 | basic20 | 32.9 | .035 |
| | | 2word12 | 31.0 | .001 |
| | | basic16 | 30.1 | <.001 |
| | | basic12 | 27.4 | <.001 |
| 2word16 | 36.8 | basic16 | 30.1 | .03 |
| | | basic12 | 27.4 | <.001 |
| 3class12 | 35.3 | basic12 | 27.4 | .002 |

**Table 3. These tables show statistically significant pairwise differences for various usability metrics across conditions. Moving clockwise from top left, the number of attempts to create an acceptable password is compared in the top-left block. The top-middle block compares agreement with the statement "Creating a password that meets the requirements given in this study was difficult." The top-right block compares password entry time for no-storage participants who entered their password correctly on the first five attempts and did not use the password reminder. The bottom-right block shows compares agreement with "Remembering the password I used for this study was difficult." The bottom-middle block compares proportions of password storage. The bottom-left block compares recall attempts by successful no-storage participants.**

### Part One Recall

After creating their passwords and filling out a brief survey, participants were asked to recall their passwords. 94.1% of participants correctly entered their password on the first attempt; this varied by condition ($\chi^2_7$=16.42, $p$=0.022). Participants in basic12 (96.4%) were significantly more likely to enter their passwords correctly than those in basic20 (93.0%) or 3class16 (HC FET, $p$<.026). Looking only at no-storage participants, 93.2% entered the correct password on the first attempt, which did not vary significantly by condition ($\chi^2_7$ =12.50, $p$=0.085). 99.1% of participants entered their passwords correctly within five attempts; this also did not vary significantly by condition ($\chi^2_7$=5.69, $p$=0.576).

### Part Two Recall

Based on part-two recall results, participants appeared to find 3class16 the least usable. 3class16 passwords took the most attempts by successful no-storage participants, took longest to enter, and were most likely to be considered difficult to remember. On the other hand, the basic conditions required the fewest attempts to enter correctly for successful no-storage participants. Conditions taking the least time to enter on the successful attempt were basic12 and 2word12. Participants reported the most difficulty with remembering passwords under 3class16, and the least with basic12. Table 2 lists the percentage of participants who entered their passwords correctly in five tries without using the reminder in each condition.

Participants could use a password reminder to display their password. 15.5% of participants used this feature, and this did not vary significantly by condition ($\chi^2_7$=8.31, $p$=0.306). Among no-storage participants, 21.4% used the reminder, and this also did not vary across conditions ($\chi^2_7$=7.72, $p$=0.358).

80.1% of participants succeeded in entering their password in the first five attempts without using the password reminder. This did not vary significantly by condition ($\chi^2_7$=7.75, $p$ =0.356). Participants who succeeded required 1.3 attempts on average to enter their password, and this number did not vary significantly by condition ($\chi^2_7$=12.96, $p$=0.073). Among no-storage participants, 75.0% were successful in the first five attempts without using the password reminder, also not varying significantly by condition ($\chi^2_7$=10.25, $p$=0.175). These participants required on average 1.3 attempts, and this did vary by condition. Significant comparisons are shown in Table 3.

We also looked at no-storage participants who did not use the password reminder, and noted how long they spent on their successful password entry. Median times varied from basic12 (11.6 seconds) to 3class16 (16.2 seconds). Significant differences are shown in Table 3.

We also asked participants whether they agreed with the statement, "Remembering the password I used for this study was difficult." The easiest condition to recall was basic12 (27.4%), and the most difficult to recall were comp8 (39.3%) and 3class16 (42.9%). The results are shown in Figure 2 and differences in responses in Table 3.

## PASSWORD PATTERNS

By studying how different ways of satisfying password requirements affect the security of the resulting passwords, we can gain insights into further requirements that might eliminate common patterns found in cracked passwords.

Overall, we find a handful of substrings that are common in passwords across conditions, and these are usually associated with the password being significantly more likely to be cracked. We examine whether and how participants exceed minimum password requirements, finding that the majority of participants exceed their length and character class requirements. In a manual exploration of our data, we find that many of the words in passwords correspond to a small number of themes, such as love and animals, suggesting a need to encourage users to consider more diverse themes for word-based passwords. Finally, because comp8 is the conventional recommended policy, we additionally focus on how participants meet the comp8 requirements.

### *Common Substrings*

We looked for substrings within passwords that might result in easily cracked passwords. We found all substrings of 4 to 12 characters that occurred in at least one percent of our passwords (40 substrings) and then eliminated those that did not exist in at least one percent of passwords without being part of another, longer substring. This eliminated substrings such as "sword," which was part of "password," and left us with seven substrings. For each, we divided passwords into those containing the substring and those not containing the substring, and looked at the cracking rates for each. As shown in Table 5, we find that passwords containing five of those substrings are significantly more likely to be cracked than passwords that do not contain them. Overall, 762 passwords (9.4%) contained at least one of the substrings associated with more easily cracked passwords.

| Substring | Using | Cracked \| Using | Cracked \| ¬Using | $p$-value |
|---|---|---|---|---|
| 1234 | 4.2% | 43.6% | 13.3% | $< .001$ |
| password | 3.1% | 45.3% | 13.6% | $< .001$ |
| love | 1.8% | 17.4% | 14.5% | .340 |
| 2013 | 1.7% | 19.9% | 14.5% | .171 |
| this | 1.7% | 23.0% | 14.4% | .027 |
| turk | 1.6% | 36.8% | 14.2% | $< .001$ |
| 123456789 | 1.2% | 51.5% | 14.1% | $< .001$ |

**Table 5. The most common substrings in passwords and how their presence affects the probability of passwords being cracked. The first column shows the percentage of all passwords using the substring. The second column shows the percentage of passwords using that string that are cracked. The third column shows the percentage of passwords not using that string that are cracked. For each substring, we ran a $\chi^2$ test to determine whether containing that substring made a password significantly more likely to be cracked; corrected $p$-values are shown in the last column. Overall, five of the seven substrings mark a password as significantly more likely to be cracked.**

Finding that some substrings are associated with cracked passwords suggests policies for future research. comp8 uses a dictionary of almost three million words, necessitating that the dictionary check be performed server-side. If a policy prohibited a small set of substrings in a password, this check could be performed client-side, reducing network traffic and facilitating real-time password meter feedback. Some websites already perform client-side password checks to facilitate password meters [30]. Prohibiting popular substrings is consistent with the advice of Schechter et al., who recommend preventing users from choosing popular passwords [26].

### *Going Beyond the Requirements*

In this section, we look at whether and how participants exceeded the minimum requirements. Evidence of passwords exceeding their minimum requirements is shown in Table 1, which shows the median length and number of characters in each character class per condition. Each condition has a median length above its minimum, and all conditions have a median of at least two digits. 65.6% of participants exceeded the minimum length of their requirement, ranging from 57.6% of participants in basic12 to 75.2% in comp8. Perhaps not surprisingly, passwords that did exceed their minimum length requirements were significantly less likely to be cracked than those that did not (10.9% to 21.6%) ($\chi^2_1$=168.07, $p<.001$).

We also looked at how many passwords used more than the minimum number of character classes, omitting comp8 passwords since they already require all four classes. 64.0% of non-comp8 participants used more than the minimum number of character classes. 40.2% of participants in 2word16 and 40.5% in 2word12 used at least three classes. Over 70% of passwords in each of the basic and 3class conditions exceeded their minimum character class requirements, ranging from 70.2% (basic20) to 79.6% (basic12). Over 70% of participants in the 3class conditions used four character classes, while a fifth of participants in comp8 did not use four character classes in their first attempts even when asked to do so. Passwords exceeding the minimum number of character classes were significantly less likely to be cracked, 8.7% to 20.7%, ($\chi^2_1$=200.72, $p<.001$). Thus, a majority of participants in all of our conditions exceeded the minimum length

and, when possible, the minimum character class requirements. This is in contrast to previous work that suggests users will only do the minimum to meet a set of requirements [10].

*Semantic Analysis*

In order to get a feel for the semantic content of user-generated passwords, we manually looked at 100 randomly chosen passwords per condition. We found that names, dates, and sequences of characters (such as *1234* and *qwerty*) were common. We also saw a number of study-related words, as well as references to animals, love, and pop culture. Surprisingly, we saw very little profanity. Participants were much more likely to place non-letter characters between words, rather than to break up single words with non-letter characters. Encouraging participants to choose words from a wider range of themes and to break up their words with non-letter characters seem worth exploring.

*Meeting the comp8 Requirements*

Because comp8 had the most requirements and was not especially resistant to cracking, we examined more closely how its requirements were met. 28.0% of passwords in comp8 fulfilled the symbol requirement only by placing "!" at the end of the password and using no other symbols. Likewise, 54.8% of passwords in comp8 used an uppercase letter as their first character and used no other uppercase letter. The 37.7% of comp8 passwords that did neither of these things were 5.3% likely to be cracked, compared to 34.5% likelihood for comp8 passwords that did either ($\chi^2_1$=131.85, $p$<.001). While there is no way to know whether users would respond to having these two practices prohibited by making stronger passwords overall, these two factors do appear indicative of more easily cracked passwords. Participants who fulfilled the requirements of comp8 in any but the most common of ways ended up with stronger passwords.

## USER PERCEPTION OF SECURITY

In part two, we asked participants whether they agreed with, "If my main email provider had the same password requirements as used in this study, my email account would be more secure." Agreement ranged from 59.8% for 3class16 and 59.7% for comp8 to 35.2% for basic12. It is salient that participants in comp8 were significantly more likely than other participants in any condition other than 3class16 to view their study policy as stronger than their real email policy (HC FET, $p$=.022). This is despite the fact that, as shown in Table 6, against an attacker making a large number of guesses, comp8 performs better than no other condition, and significantly worse than five. This suggests that users associate at least some of the requirements of comp8 with strong passwords, even if that is not necessarily true in practice. It further suggests that users might not know how best to construct strong passwords, even if they wish to do so.

## DISCUSSION

We have replicated, confirmed, and expanded the previous finding that password policies requiring length lead to more usability, and in some cases more security, than those requiring only a comprehensive mix of character classes and a dictionary check. Further, we have begun exploring the space of longer passwords, examining how different augmentations to a length requirement affect usability and security. We have found a tradeoff between usability and security, with none of our longer policies being strictly better than another on both usability and security. We have, however, found multiple policies that appear to offer benefits over the commonly recommended comp8. We find that while the basic policies are generally easier to use than their augmented same-length counterparts, they are vulnerable for relatively small numbers of guesses. And we have also found that despite comp8 being more vulnerable than most of our other conditions to a powerful attacker, users tend to perceive comp8 as a more secure policy. In this section, we discuss the implications of our work for system administrators and conclude with some directions for future research.

### Recommendations for Password Policy

We compare comp8 with our other conditions to determine whether there is a longer-length policy that has both usability and security benefits over this typical password policy. The statistically significant differences we found between comp8 and the other conditions are summarized in Table 6. Looking at usability metrics, we see that passwords in basic20 and 3class16 take significantly longer to type than those in comp8, and participants expressed more difficulty in creating passwords under 3class16; otherwise, all other conditions either exceeded the usability of comp8 or were not significantly different. Looking at security, we see that all of the basic policies and 2word12 have worse security than comp8 after a million guesses, making them more vulnerable to a limited attacker. The two conditions that are more usable overall than comp8, not significantly weaker against a limited attacker, and significantly stronger against a powerful attacker are 3class12 and 2word16. Comparing these two conditions, we find a tradeoff between the two in terms of security and usability, with 3class12 being more usable during creation, and 2word16 offering more security. It is possible that the usability advantage of password-creation under 3class12 was due to participants being more familiar with similar conditions. This advantage may diminish if users become more accustomed to creating passwords with a 2word requirement.

### Limitations

Our methodology, which is similar to that employed by prior password research [21, 22, 30], has a number of limitations. By testing password recall once after a few minutes and once again a few days later, our study investigated password use that lies in between frequent and rare use. Our results may apply only partially to the common cases of passwords that are used very frequently or very sporadically.

Across our conditions, a relatively high number of participants did not return for part two. We excluded them from our analyses except our dropout analysis. In practice, users who drop out out of a study might behave differently than those who do not, potentially biasing our results.

Furthermore, although our conditions varied in a number of ways, such as the number of characters or the number of character classes required, we did not employ a full-factorial study

| Condition | Part one dropout (%) | Password storage (%) | Mean creation attempts | Agree creation difficult (%) | Part two recall attempts | Password entry time (s) | Agree remembering difficult (%) | Cracked after $10^6$ guesses (%) | Cracked at cutoff (%) |
|---|---|---|---|---|---|---|---|---|---|
| **comp8** | **83.0** | **56.9** | **2.3** | **32.6** | **1.4** | **13.2** | **39.3** | **0.6** | **23.5** |
| basic12 | 94.5 | 45.4 | 1.5 | 14.9 | 1.3 | 11.6 | 27.4 | 3.9 | 24.5 |
| basic16 | 93.9 | 49.9 | 1.7 | 28.9 | 1.3 | 13.7 | 30.1 | 5.2 | 12.4 |
| basic20 | 93.9 | 50.0 | 1.9 | 34.7 | 1.3 | 15.3 | 32.9 | 3.9 | 7.1 |
| 2word12 | 92.0 | 51.4 | 1.8 | 20.9 | 1.3 | 13.1 | 31.0 | 2.1 | 22.7 |
| 2word16 | 92.1 | 51.3 | 2.0 | 35.2 | 1.4 | 14.6 | 36.8 | 1.0 | 6.6 |
| 3class12 | 92.0 | 54.9 | 1.6 | 25.4 | 1.4 | 14.8 | 35.3 | 1.5 | 16.0 |
| 3class16 | 90.5 | 60.2 | 1.8 | 40.5 | 1.4 | 16.2 | 42.9 | 0.3 | 1.4 |

**Table 6. A summary of the statistically significant differences between comp8 and the seven other conditions, as presented in the Results section. Cells are shaded in blue if a condition was found to be significantly better than comp8, and red if significantly worse. No shading indicates no significant difference.**

design. For instance, we did not test 3class20, 2word8, or similar conditions. To minimize the number of conditions, we instead grouped changes to multiple variables in ways we hypothesized might balance usability and security. While we can compare the conditions we tested, we are unable to evaluate the effect of changing each individual variable. Similarly, we might have missed interaction effects between variables.

The passwords in our study did not protect high-value accounts, limiting ecological validity. In contrast to real-world, high-value passwords, study participants would not suffer consequences if they chose a weak password or forgot their password, nor were they incentivized to adopt their normal password behavior beyond our request that they do so.

Two recent studies have investigated the degree to which passwords from research studies resemble real, high-value passwords. Both studies concluded that passwords created during studies can resemble real, high-value passwords, yet are not a perfect proxy. In a prior study, our group obtained indirect access to the high-value, single-sign-on passwords of everyone at our university [23], which we compared to passwords collected on MTurk and to real passwords. The MTurk passwords were more similar than the leaked datasets to the real, high-value passwords, yet were slightly weaker than passwords at our university. Fahl et al. investigated the ecological validity of both online and laboratory password studies by recruiting participants from their university in Germany, comparing participants' study passwords to their real university passwords [15]. Based on manual coding, they concluded that 46% of the passwords from the online study were fully representative of those users' actual passwords, while an additional 23% were partially representative. In the lab study, 49% of passwords were fully representative, while 32% were partially representative.

Finally, there were factors in the study beyond our control. For instance, we did not control the device or keyboard used to input the password, and many MTurk workers use desktop computers. As a result, we were unable to address usage on mobile devices, which are ubiquitous. The effect of mobile devices' constrained, touch-sensitive keyboards on password usability is a particularly interesting area of future work.

### Suggestions for Future Work

Dictionary checks using large dictionaries often require sending the prospective password to the server for comparison, making it difficult to provide real-time feedback incorporating a dictionary check. On the other hand, we found five substrings that lead to a significantly greater likelihood of a password containing them being cracked. This suggests that future work might investigate using a client-side substring check with a much smaller list of prohibited substrings.

Further, we found a small set of themes that typically appear in the component words of passwords. This finding, combined with our finding that some strings are fairly common in passwords, suggests future work on nudging people to create word-based passwords on a more diverse set of themes. We also observed that it was uncommon for participants to use a non-letter to break up the letters within a word; this suggests that future work might further explore nudging participants to do so more often in order to increase diversity.

## REFERENCES


1. Adams, A., Sasse, M. A., and Lunt, P. Making passwords secure and usable. In *Proc. HCI* (1997).
2. Biddle, R., Chiasson, S., and van Oorschot, P. C. Graphical passwords: Learning from the first twelve years. *ACM Computing Surveys 44*, 4 (2012), 19.
3. Bishop, M., and Klein, D. V. Improving system security via proactive password checking. *Computers & Security 14*, 3 (1995), 233–249.
4. Bonneau, J. The Gawker hack: how a million passwords were lost, 2010. `http://www.lightbluetouchpaper.org/2010/12/15/the-gawker-hack-how-a-million-passwords-were-lost/`.
5. Bonneau, J. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *Proc. IEEE SP* (2012).
6. Bonneau, J., Herley, C., van Oorschot, P. C., and Stajano, F. The quest to replace passwords: A framework for comparative evaluation of Web authentication schemes. In *Proc. IEEE SP* (2012).

7. Brantz, T., and Franz, A. The Google Web 1T 5-gram corpus. Tech. Rep. LDC2006T13, Linguistic Data Consortium, 2006.
8. Bright, P. Anonymous speaks: The inside story of the HBGary hack. Ars Technica, February 2011.
9. Burr, W. E., Dodson, D. F., Newton, E. M., Perlner, R. A., Polk, W. T., Gupta, S., and Nabbus, E. A. Electronic authentication guideline. Tech. rep., NIST, 2011.
10. Burr, W. E., Dodson, D. F., and Polk, W. T. Electronic authentication guideline. Tech. rep., NIST, 2006.
11. Campbell, J., Ma, W., and Kleeman, D. Impact of restrictive composition policy on user password choices. *Behaviour & Information Technology 30*, 3 (2011).
12. Chiasson, S., Forget, A., Stobert, E., van Oorschot, P. C., and Biddle, R. Multiple password interference in text passwords and click-based graphical passwords. In *Proc. CCS* (2009).
13. Constantin, L. Sony stresses that PSN passwords were hashed. **http://news.softpedia.com/news/Sony-Stresses-PSN-Passwords-Were-Hashed-198218.shtml**, 2011.
14. Dell'Amico, M., Michiardi, P., and Roudier, Y. Password strength: An empirical analysis. In *Proc. INFOCOM* (2010).
15. Fahl, S., Harbach, M., Acar, Y., and Smith, M. On the ecological validity of a password study. In *Proc. SOUPS* (2013).
16. Gaw, S., and Felten, E. W. Password management strategies for online accounts. In *Proc. SOUPS* (2006).
17. Herley, C., and Van Oorschot, P. A research agenda acknowledging the persistence of passwords. *IEEE Security and Privacy 10*, 1 (2012), 28–36.
18. InCommon Federation. Identity assurance profiles bronze and silver v1.1, 2011.
19. Inglesant, P., and Sasse, M. A. The true cost of unusable password policies: password use in the wild. In *Proc. CHI* (2010).
20. Keith, M., Shao, B., and Steinbart, P. A behavioral analysis of passphrase design and effectiveness. *Journal of the Association for Information Systems 10*, 2 (2009), 63–89.
21. Kelley, P. G., Komanduri, S., Mazurek, M. L., Shay, R., Vidas, T., Bauer, L., Christin, N., Cranor, L. F., and Lopez, J. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In *Proc. IEEE SP* (2012).
22. Komanduri, S., Shay, R., Kelley, P. G., Mazurek, M. L., Bauer, L., Christin, N., Cranor, L. F., and Egelman, S. Of passwords and people: measuring the effect of password-composition policies. In *Proc. CHI* (2011).
23. Mazurek, M. L., Komanduri, S., Vidas, T., Bauer, L., Christin, N., Cranor, L. F., Kelley, P. G., Shay, R., and Ur, B. Measuring password guessability for an entire university. In *Proc. CCS* (2013).
24. Narayanan, A., and Shmatikov, V. Fast dictionary attacks on passwords using time-space tradeoff. In *Proc. CCS* (2005).
25. Proctor, R. W., Lien, M.-C., Vu, K.-P. L., Schultz, E. E., and Salvendy, G. Improving computer security for authentication of users: Influence of proactive password restrictions. *Behavior Res. Methods, Instruments, & Computers 34*, 2 (2002), 163–169.
26. Schechter, S., Herley, C., and Mitzenmacher, M. Popularity is everything: A new approach to protecting passwords from statistical-guessing attacks. In *Proc. HotSec* (2010).
27. Shay, R., Kelley, P. G., Komanduri, S., Mazurek, M. L., Ur, B., Vidas, T., Bauer, L., Christin, N., and Cranor, L. F. Correct horse battery staple: Exploring the usability of system-assigned passphrases. In *Proc. SOUPS* (2012).
28. Shay, R., Komanduri, S., Kelley, P. G., Leon, P. G., Mazurek, M. L., Bauer, L., Christin, N., and Cranor, L. F. Encountering stronger password requirements: user attitudes and behaviors. In *Proc. SOUPS* (2010).
29. Spafford, E. H. OPUS: Preventing weak password choices. *Computers & Security 11*, 3 (1992).
30. Ur, B., Kelley, P. G., Komanduri, S., Lee, J., Maass, M., Mazurek, M., Passaro, T., Shay, R., Vidas, T., Bauer, L., Christin, N., and Cranor, L. F. How does your password measure up? The effect of strength meters on password creation. In *Proc. USENIX Security* (2012).
31. Vance, A. If your password is 123456, just make it HackMe. The New York Times, **http://www.nytimes.com/2010/01/21/technology/21password.html**, January 2010.
32. Vu, K.-P. L., Proctor, R. W., Bhargav-Spantzel, A., Tai, B.-L. B., and Cook, J. Improving password security and memorability to protect personal and organizational information. *Int. J. of Human-Comp. Studies 65*, 8 (2007), 744–757.
33. Weir, M., Aggarwal, S., Collins, M., and Stern, H. Testing metrics for password creation policies by attacking large sets of revealed passwords. In *Proc. CCS* (2010).
34. Weir, M., Aggarwal, S., Medeiros, B. d., and Glodek, B. Password cracking using probabilistic context-free grammars. In *Proc. IEEE SP* (2009).
35. Zviran, M., and Haga, W. J. Password security: an empirical study. *J. Mgt. Info. Sys. 15*, 4 (1999).