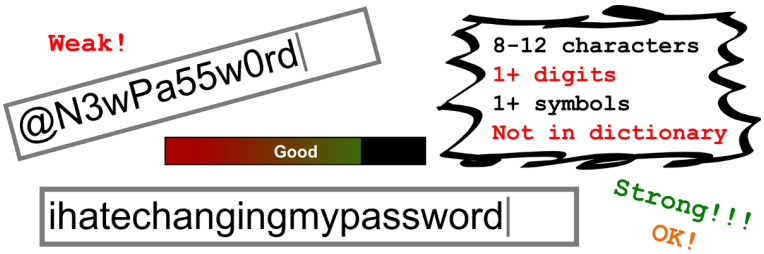
# Helping Users Create Better Passwords

BLASE UR, PATRICK GAGE KELLEY, SARANGA KOMANDURI, JOEL LEE, MICHAEL MAASS, MICHELLE L. MAZUREK, TIMOTHY PASSARO, RICHARD SHAY, TIMOTHY VIDAS, LUJO BAUER, NICOLAS CHRISTIN, LORRIE FAITH CRANOR, SERGE EGELMAN, AND JULIO LÓPEZ




Blase Ur is a second-year PhD student in the School of Computer Science at Carnegie Mellon University. His research focuses on usable security and privacy, including passwords, online behavioral advertising, and privacy decision making. He received his undergraduate degree in computer science from Harvard University. bur@cmu.edu

Patrick Gage Kelley is an Assistant Professor of Computer Science at the University of New Mexico. His research centers on information design, usability, and education around privacy. He recently completed his thesis at Carnegie Mellon University on standardized, user-friendly privacy displays for privacy policies and Android permission displays. pgage@cmu.edu

Saranga Komanduri is a PhD student in the School of Computer Science at Carnegie Mellon University. His research covers a broad spectrum of security-related topics, including authentication, usable security, and warnings. sarangak@cmu.edu

Over the past several years, we have researched how passwords are created, how they resist cracking, and how usable they are. In this article, we focus on recent work in which we tested various techniques that may encourage better password choices. What we found may surprise you.

Despite a litany of proposed password replacements, text-based passwords are not going to disappear anytime soon [4]. Passwords have a number of advantages over other authentication mechanisms. They are simple to implement, relatively straightforward to revoke or change, easy for users to understand, and allow for quick authentication; however, passwords also have a number of drawbacks. Foremost among these drawbacks is that it is difficult for users to create and remember passwords that are hard for an attacker to guess. Our research group at Carnegie Mellon University has been investigating strategies to guide users to create passwords that are both secure and memorable.

In particular, we have focused on techniques such as password-composition policies and password-strength meters—two of the most ubiquitous strategies employed by system administrators to help users create secure passwords. Although these strategies are commonly used, their effects had not been well understood. Through a series of online studies, we have aimed to understand how password-composition policies and meters affect password security, memorability, and user sentiment.

The first step in evaluating the security of a password is to understand the threat model. For instance, one can argue that a password that is hard to guess within the first three or five tries is secure, since an attacker would quickly be locked out. All but the most obvious passwords tend to resist this type of online attack. Passwords have been under attack in other ways due to a spate of password-database compromises in recent years, including at sites like Gawker and LinkedIn [3]. In possession of such a database, in which passwords are usually salted and hashed, rather than stored in plaintext, an adversary can still "crack" passwords by hashing potential passwords and checking whether these hashes appear in the database. This type of attack, known as an offline attack, is particularly pernicious since many users reuse a single password, or closely related passwords, across several sites to avoid remembering dozens of passwords [2]. Thus, an offline attack that successfully guesses a password on one site may let the attacker access a cornucopia of other accounts.

Joel Lee is an MS student in information security policy and management at Carnegie Mellon University. He is interested in usable security and privacy, adopting effective security policies in enterprises, and in balancing security with the core business operations of a company. He did his undergraduate degree in a partnership between CMU and Singapore Management University. jlee@cmu.edu

Michael Maass is a second-year PhD student studying software engineering at Carnegie Mellon University. He works on science of security problems, focused on sandboxing and evidence-based software assurance. Michael worked as a security engineer in the aerospace industry before pursuing a PhD. mmaass@cmu.edu

Michelle Mazurek is a fifth-year PhD student in electrical and computer engineering at Carnegie Mellon University. Her research focuses on usable security and privacy, including usable access control and passwords. She received her undergraduate degree in electrical engineering from the University of Maryland. mmazurek@cmu.edu

Timothy Passaro is a senior BS student in the Carnegie Institute of Technology and School of Computer Science at Carnegie Mellon University. His primary research interest is usable security and privacy. tpassaro@cmu.edu

Richard Shay is a fourth-year PhD student in the School of Computer Science at Carnegie Mellon University. His research focuses on usable privacy and security, studying online behavioral advertising and password policy. He received an undergraduate degree in computer science and classics from Brown University, and a master's degree in computer science from Purdue University. rshay@cmu.edu

In this article, we first introduce the methodology for our recent work on password-composition policies [5, 6] and password meters [8], and define the metrics we used to measure the security and usability of passwords. We then highlight key results from these studies, paying particular attention to the lessons they hold for guiding real-world password creation.

## Methodology and Metrics

Both our password-composition-policy study and our password-meter study took place online in two separate parts. We recruited participants using Amazon's Mechanical Turk crowdsourcing service. Our password-composition-policy study involved more than 12,000 participants, while our study of password meters included more than 3,000 participants.

In the first part of each study, we asked participants to imagine that their main email provider had changed its password requirements, and that they needed to create a new password. In the study of password-composition policies, each participant created a password conforming to one of seven different composition policies, detailed later in this article. In the study of password meters, all participants created passwords under the same policy, but saw one of 14 different password meters, described later in this article, or no meter. Participants then completed a survey about the password-creation experience and were asked to re-enter their passwords. Two days later, participants received an email inviting them to return, log in again with their password, and to take another survey about how they handled their password.

Traditionally, password strength for a set of passwords has been measured by entropy. In contrast, recent research advocates "guessability," the number of guesses it would take an adversary to guess a password, as a more appropriate metric for evaluating the real-world security of passwords against password-cracking attacks [1]. In our work, we calculated guessability by simulating a state-of-the-art password cracking algorithm [9] and determining how many attempts that algorithm would make to find a particular password, based on a particular set of training data.

To measure the usability of a password, we employed several metrics. First, we considered the memorability of the password. As a proxy for memorability, we examined the rate at which participants were able to log in successfully using their password about five minutes after password creation and when they returned for the second part of the study two or more days later. We also examined the rate at which participants returned for the second part of the study, hypothesizing that participants who created unmemorable passwords might not return. We further considered the proportion of participants who indicated in our surveys that they wrote their password down or stored it electronically, or who used their browser or a password manager to fill in their password automatically. Additionally, we presented participants with sentiment statements, to which they indicated levels of agreement or disagreement on a five-point Likert scale.

## Password-Composition Policies

In our study of password-composition policies, we examined five main types of policies. Each participant was assigned round-robin to a single policy. As a baseline, the first policy, which we termed "basic8," required only that the password contain at least eight characters. To observe the impact of requiring longer passwords, we tested a "basic16" policy, which required only that the password

Timothy Vidas is an ECE PhD candidate at Carnegie Mellon University. His research interests include mobile device security, digital forensics, reverse engineering, cybercrime, and many other aspects of computer security. tvidas@cmu.edu

Lujo Bauer is an Assistant Research Professor in CyLab and the Electrical and Computer Engineering Department at Carnegie Mellon University. Lujo's research interests span many areas of computer security and include building usable access-control systems with sound theoretical underpinnings; developing languages and systems for runtime enforcement of security policies on programs; and, generally, narrowing the gap between a formal model and a practical, usable system. lbauer@cmu.edu

Nicolas Christin is the Associate Director of the Information Networking Institute at Carnegie Mellon University and a research faculty (Senior Systems Scientist) in CyLab and the Electrical and Computer Engineering and Engineering and Public Policy departments. His research is at the intersection of systems, security, and public policy, and has most recently focused on online crime, security economics, and psychological aspects of computer security. nicolasc@cmu.edu

Lorrie Faith Cranor is an Associate Professor of Computer Science and of Engineering and Public Policy at Carnegie Mellon University where she is Director of the CyLab Usable Privacy and Security Laboratory (CUPS). She is also a co-founder of Wombat Security Technologies, Inc. and previously was a researcher at AT&T Labs-Research. She has authored more than 100 research papers on online privacy, usable security, and other topics. lorrie@cmu.edu

contain at least 16 characters. We then tested a condition, "dictionary8," in which the password was stripped of non-alphabetic characters and checked against the free Openwall cracking dictionary. To test passwords that had to include several character classes, our "comprehensive8" condition mandated an eight-character password containing a lowercase letter, an uppercase letter, a digit, and a symbol. The password also needed to pass the same dictionary check as in dictionary8. We also tested three variants of a blacklist policy, which allowed all passwords containing at least eight characters other than passwords on blacklists, which were sourced from dictionaries ranging in size from hundreds of thousands to billions of potential passwords.

As shown in Figure 1, for weaker adversaries—those that would make around one billion guesses—the comprehensive8 and largest blacklist conditions were particularly resistant to a guessing attack, with the basic16 condition performing slightly worse. As the number of guesses increased, basic16 began to outperform the other conditions in guessing resistance. For instance, with one trillion guesses, only around half as many basic16 passwords were cracked as in comprehensive8 and the largest blacklist condition, which in turn were significantly more resistant to guessing than any other condition.

We also found usability advantages for the basic16 policy, which required long passwords with no further restrictions. Many popular Web sites' password policies are similar to our comprehensive8 condition, mandating passwords containing several character classes and passing a dictionary check; however, compared to participants who needed to comply with the comprehensive8 policy, those who needed to comply with basic16 needed fewer attempts to create their password and reused existing passwords at a lower rate [6]. Furthermore, basic16 participants expressed less frustration in our sentiment questions than those who needed to enter a password that does not appear in a dictionary or blacklist.

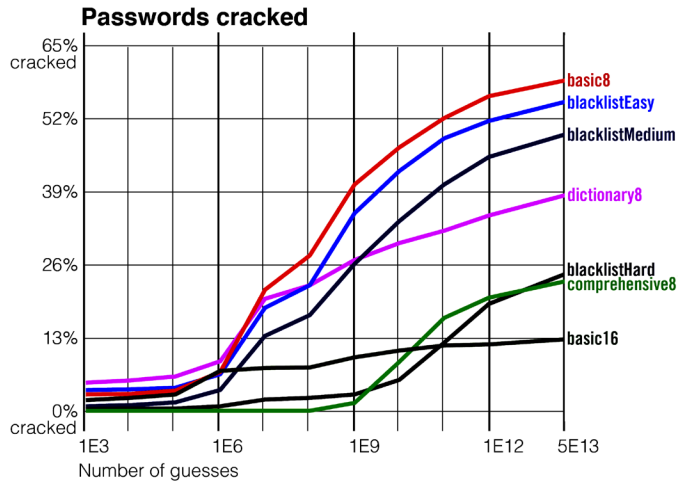



Figure 1: Percentage of passwords cracked by our password guesser for passwords collected under several password policies

Serge Egelman is a research scientist at UC Berkeley working on usable security problems. He uses empirical data to improve user interface designs for security mechanisms. He received his PhD from Carnegie Mellon University. egelman@cs.berkeley.edu

Julio López is a Senior Software Engineer at Maginatics, Inc. where he works on large-scale cloud storage systems. He received his PhD and MS degrees from the Electrical and Computer Engineering Department at Carnegie Mellon University, and his BS from Universidad EAFIT in Colombia. julio.lopez@cmu.edu

## Password-Strength Meters

In our study of password meters, participants were assigned round-robin to one of 15 conditions. In our control condition, participants were asked to create a password with no meter present. In each of the other 14 conditions, participants saw some variant of a password meter as they created their password. The design of one condition, which we termed a baseline meter, was informed by a survey we performed of password meter use on highly popular Web sites. Like the meters observed in the wild, our baseline meter computed the strength of the password using heuristics, such as the length of a password and the character classes it contained. To fill the bar completely, a participant's password could contain 16 or more characters, with no further restrictions. Alternatively, it could contain eight or more characters, including a lowercase letter, uppercase letter, digit, and symbol, as well as pass a dictionary check. As the bar became filled, it changed color from red to yellow to green; meanwhile, a single word of textual feedback changed in several steps from "bad" to "excellent." We also provided a suggestion for improvement, such as "Consider adding a digit or making your password longer."

Our other conditions, shown in Figure 2, tested meters with various visual elements and with different scoring strategies. To test the effect of the visual elements, we created seven meters that differed from the baseline meter only in their visual display. These conditions included a meter with a segmented, rather than continuous, bar; a meter that was always green; a tiny meter; a huge meter; a meter that didn't give suggestions for improving the password; and a meter that had only text, without a visual bar. We also created a meter that replaced the visual bar with an animated bunny. The stronger the participant's password, the faster the bunny danced.

To test the effect of changing how the meters scored passwords, we created four meters that scored passwords stringently, as well as two meters that nudged participants toward a particular password policy. Two of the four stringent meters had the same visual appearance as the baseline meter, yet always gave passwords half the score or one-third of the score that the baseline meter would have given. The two other stringent meters always gave half the score of the baseline meter, yet were text-only, lacking a visual bar. One text-only meter had standard-weight text, while the other had boldface text. One of the two meters nudging participants toward a particular policy only scored a password on its length, while the other policy more heavily weighted the inclusion of multiple character classes.

We found that all meters we tested led to passwords with different properties than those created without a meter. Passwords created with any type of meter were longer, on average, than those created with no meter. Furthermore, passwords created with stringent meters were the longest. For instance, passwords created with the half-score meter had a mean length that was 4.5 characters greater than those created with no meter.

We then evaluated the strength of passwords using the aforementioned "guessability" metric, quantifying the number of guesses a sophisticated adversary would need to guess that password. We found that all password meters we tested provided at least a small advantage against guessing attacks, although most of these differences were not statistically significant. As summarized in Table 1, the two stringent meters with visual bars, half and one-third score, provided a significant increase in guessing resistance compared to not having a meter. For instance, within the first 5 trillion guesses ($5\times10^{12}$), 47% of passwords created with no meter

were cracked. In contrast, only 26% of passwords created with the half-score meter and 28% of passwords created with the one-third-score meter were cracked, while 34–46% of passwords created with all other meters were cracked.

| | No Meter | Baseline Meter | Half-score Meter | One-third-score Meter | All Other Meters |
|---|---|---|---|---|---|
| $5\times10^{10}$ guesses | 35% | 27% | 20% | 17% | 24–34% |
| $5\times10^{12}$ guesses | 47% | 39% | 26% | 28% | 34–46% |

**Table 1:** The percentage of passwords in each condition cracked within the first $5\times10^{10}$ and first $5\times10^{12}$ guesses

Although the passwords created with a meter tended to be longer and harder to guess, they did not seem to be less memorable. In particular, we did not observe statistically significant differences across conditions in any of our metrics for the memorability of passwords. Participant sentiment did differ across conditions, with the stringent meters leading participants to express annoyance at a higher rate. Stringent meters also caused increased participant disillusionment; participants in these conditions agreed at a higher rate that they did not "understand how the password strength meter rates [their] password" and agreed at a lower rate with the statement, "It's important to me that the password-strength meter gives my password a high score."

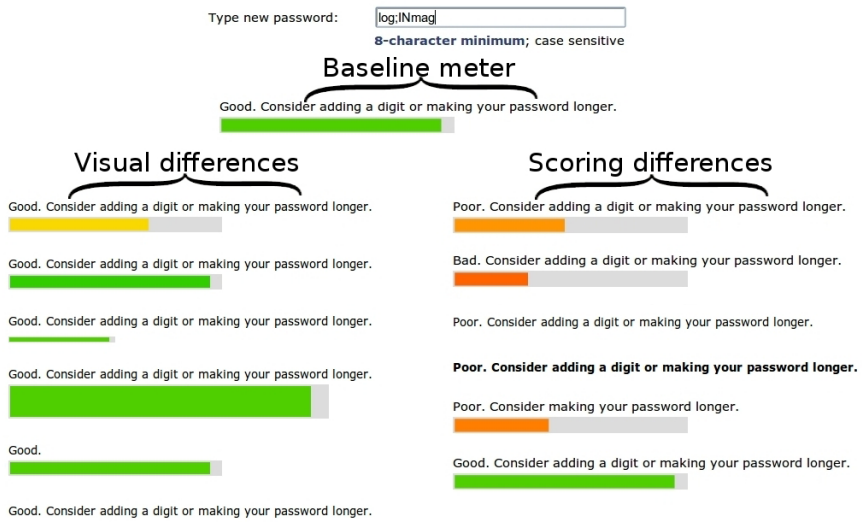



**Figure 2:** The password meters we tested varied in their visual design and in the way they scored a password

## Conclusions

From our study, we learned that a password-composition policy that causes users to create passwords that are longer than usual, rather than passwords that contain an array of character classes and that aren't in a blacklist, seems to possess a number of advantages. As the number of guesses increased, basic16 passwords were more resistant to a guessing attack. On the other hand, since long passwords are less common, it is also possible that existing cracking algorithms have not been optimized to crack long passwords. In addition to their greater resistance to an

offline attack, basic16 passwords also had usability advantages over other policies resistant to guessing. For instance, these longer passwords were easier to create and led to more favorable participant sentiment.

Still, while encouraging users to make longer passwords seems to provide both security and usability benefits, we cannot yet definitively recommend a single password-composition policy as the ideal one. For instance, we are not yet sure of the optimal minimum length for these longer passwords. Likewise, a small percentage of the passwords created under the policy that emphasized length would have been guessed even by a very weak attacker. As such, we still need to establish whether encouraging or mandating the usage of additional character classes in long passwords would increase security without adversely affecting usability.

We began our password-meter study wondering whether meters had any effect, and we found that meters did indeed impact user behavior and security. For instance, meters led users to create longer passwords. Unfortunately, unless the meter scored passwords stringently, the resulting passwords were only marginally more resistant to password-cracking attacks. The non-stringent meters in our study most closely approximated real-world meters, suggesting that currently deployed meters are too lenient. This result hints that system administrators should make it more difficult to receive high scores from meters. How this result generalizes remains an open question since our study only examined the effect of a meter on the creation of a single password. It is possible a user would habituate to receiving lower scores from meters, neutralizing the security benefit. Also, whether having to remember multiple passwords would lead to greater difficulty in remembering passwords created with stringent meters is unknown.

As an alternative to guiding password creation, our group has also studied assigning users passphrases, which are long (and thus more secure) passwords formed of space-delimited words in a natural language. By automatically assigning passphrases, their guessability can be controlled. We compared the usability of these system-assigned passphrases to system-assigned traditional passwords. We found passphrases to be far from a usability panacea; by various usability metrics, they often performed slightly worse or no better than traditional passwords [7]. Our investigation also revealed a few ways in which passphrases may be improved to realize some of the desired usability benefits.

The research community is still far from providing the last word on the best strategies for helping users create passwords that are both hard to guess and easy to remember. Passwords have received substantial attention in recent years, with many exciting contributions coming from a number of different research groups. We look forward to conducting additional studies to help solidify our understanding of passwords, and to providing more definitive guidelines to help users create better passwords.

### *References*

[1] J. Bonneau, "The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords," IEEE Symposium on Security and Privacy, 2012.

[2] D. Florencio and C. Herley, "A Large-Scale Study of Web Password Habits," WWW 2007.

[3] D. Goodin, "Why Passwords Have Never Been Weaker—And Crackers Have Never Been Stronger," Ars Technica, August 21, 2012.

[4] C. Herley, P. Van Oorschot, "A Research Agenda Acknowledging the Persistence of Passwords," *IEEE Security & Privacy Magazine,* vol. 10, no. 1, 2012, pp. 28–36.

[5] P.G. Kelley, S. Komanduri, M.L. Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christin, L.F. Cranor, J. Lopez, "Guess Again (and Again and Again): Measuring Password Strength by Simulating Password-Cracking Algorithms," IEEE Symposium on Security and Privacy, 2012.

[6] S. Komanduri, R. Shay, P.G. Kelley, M.L. Mazurek, L. Bauer, N. Christin, L.F. Cranor, and S. Egelman, "Of Passwords and People: Measuring the Effect of Password-Composition Policies," CHI, 2011.

[7] R. Shay, P.G. Kelley, S. Komanduri, M. Mazurek, B. Ur, T. Vidas, L. Bauer, N. Christin, L.F. Cranor, "Correct Horse Battery Staple: Exploring the Usability of System-Assigned Passphrases," SOUPS, 2012.

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

[9] M. Weir, S. Aggarwal, B. de Medeiros, and B. Glodek, "Password Cracking Using Probabilistic Context-Free Grammars," IEEE Symposium on Security and Privacy, 2009.