Improving Password Memorability and Strength Using Mangling Rules

Sanam Ghorbani Lyastani
Graduate School, Saarland
University
Saarbrücken, Germany
s8saghor@stud.uni-saarland.de

Yasemin Acar
Sascha Fahl
CISPA, Saarland University
Saarbrücken, Germany
acar@cs.uni-saarland.de
fahl@cs.uni-saarland.de

Michael Backes
CISPA, Saarland University &
MPI-SWS
Saarbrücken, Germany
backes@mpi-sws.org

Abstract

UPDATED—June 8, 2016. Recently, to support users in choosing more secure passwords, websites are providing password strength meters and/or require user passwords to conform to a composition policy. However, there are inconsistent strength outcomes for the same password in different password meters that may confuse users in creating a stronger password. Also, policies may miss their goal, since users create predictable ("weak") passwords under those policies. To help users to create complex passwords, we are proposing a password generator mechanism that is based on mangling rules. The goal of using the mangling rules is to increase the security of the proposed passwords without sacrificing the memorability. We are planning an online user study on Amazon MTurk to evaluate memorability and users' preferences of our approach.

Author Keywords

Authentication; Password Study; Security; Password Generator; Memorability; User Preference; User Study.

Introduction

Text-based passwords are the default authentication scheme for online services. Bonneau et al. [2] found that there is no ideal alternative scheme that can replace passwords. Unfortunately, creating a strong memorable password is one of the main drawbacks of using passwords. Many users are selecting weak passwords which are too easy to guess [11] by using cracking tools such as John the Riper or Hashcat.

Recently most websites provide password strength meters to alert their user for weak passwords. However, these mechanisms are mostly based on ad-hoc design [4] which means that for example a password that is labeled as a weak password by one meter can be labeled as strong or very strong by a different password meter, which may confuse the users about their password's strength. Some web sites are giving password policies to the users, for instance "use at least one uppercase letter and lowercase letter, one digit and one symbol", to help them to create stronger passwords. The users might create passwords based on simple patterns and dictionary words, however, these created passwords can easily be cracked by using mangling rules[1]. To prevent users from creating passwords crackable by mangling rules, we need new mechanisms to help users to create and choose passwords which are hard to guess by others, hard to crack by cracking tools but still memorable.

Our goal. While mangling rules are used by password crackers like John the Ripper to create new password guesses from existing ones, we are going to use such rules to generate stronger passwords that still contain linguistic structure to retain good memorability. The mangling rules should be designed to make a good trade-off between security (i.e. generate randomly) and usability (i.e. user-chosen). The goals of this study are to investigate *the memorability of the created password in long-term* and *which kind of generated passwords are selected by the users (users' preferences?)*

Related Work

Password meters are telling the users whether their password is weak or strong.de Carnavalet et al [4] found that each meter reacts differently to their dictionaries, e.g., a password labeled as weak by one meter, may be labeled as perfect by another meter.These drawbacks may confuse users in creating a stronger password. Also, they created dictionaries with mangled passwords using common mangling rules and leet transformations which overlap with leaked password databases and which show that real users chose passwords that can be cracked with mangling rules. Weir et al. [14] automatically created a probabilistic context-free grammar based upon a training set of previously disclosed passwords. This grammar then allowed them to generate word-mangling rules, and from them, password guesses to be used in password cracking. They tested their tools and techniques on real password sets and were able to crack 28% to 129% more passwords than John the Ripper.To help user to create stronger passwords, Forget et al. [6, 7] proposed a password generator that uses persuasive technology to affect users to choose the stronger passwords by replacing or inserting random characters. Their results show that the mechanism had positive influence on users creating stronger passwords, however they did not test the long term memorability of proposed methods. All related work so far has shown that mangling rules can crack user passwords because users chose predictable passwords. Leversund [10] proposes to add a limited amount of guaranteed security to explicit password creation policies by randomly selecting a policy when users create their passwords. Since the attacker would not know which set of rules the user created their password under, they would then have to structure their attack to target multiple creation policies. One potential problem with this, besides user annoyance, is that this approach does not stop a user from selecting a weak password under the random policy. Since

[1] Mangling rules are transforming a dictionary word into another word. Mangling rules can describe the behavior of users, how they choose passwords and they used password cracking to emulate user behavior that should fit to the password policies.

the number of policies is finite, an attacker may still be successful by guessing the most common passwords for each policy.

Generate Passwords Based On Mangling Rules to Make Passwords Harder to Crack

The goal of our work is to propose a password generator mechanism that is based on mangling rules to create stronger but also more memorable passwords. To achieve this goal, we have to find rules that on the one hand create a big enough password space and uncommon password patterns to make the generated passwords harder to guess for an attacker, on the other hand, the rules should preserve enough linguistic structure in passwords, in comparison to randomly generated passwords, to make them more memorable. In the end, our approach makes a trade-off in which passwords are stronger than typical user passwords, although not as strong as truly random passwords, while memorability stays unchanged or at least reasonable preserved.

Examples of generated passwords with our mangling rules

Prime password is "password"

Rule 1:
p!a1sw#ord

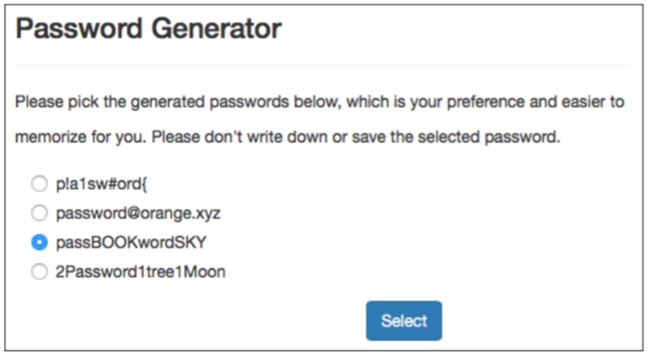
Rule 2:
password@orange.kwn

Rule 3:
passBOOKwordSKY

Rule 4:
2Password 1Tree 7Moon

Mangling rules for password creation. Since mangling rules describe how a word can be transformed, they could be also used to define rules for generating passwords. So each rule describes a password generator with a specific pattern and all rules together the entire password generator. For instance, a simple rule could be implemented that puts eight random digits at the end of the word or capitalizes random characters. Concrete example mangling rules that we implement for our work will be described in details later in this section.

Password generator tool. We are implementing our idea as a website using HTML and JavaScript. The user is requested to select a prime password (at least six characters) from a large dictionary (to have a meaningful word) and this prime password is used as input for mangling rules, which then generate the actual password choices that the user can select. To guarantee that the prime password that is entered by the participant is based on dictionary word, we provided an auto-completion based on a prime password dictionary (the alternative methods may be: spell checking, giving users word lists, etc). The password generator will take the prime password and transform it based on our mangling rules (see Figure 1).



Password Generator

Please pick the generated passwords below, which is your preference and easier to memorize for you. Please don't write down or save the selected password.

○ p!a1sw#ord{
○ password@orange.xyz
● passBOOKwordSKY
○ 2Password1tree1Moon

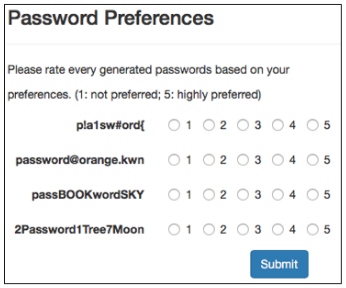
Figure 1: Generated passwords based on prime password.

Our example mangling rules. Our generator provides passwords based on the following four sample rules :

1. Inserting 4 random characters, digits, or symbols at random position of prime password. In this rule we take prime word and inserting characters or digits at 4 random positions. We are using this rule to have a comparison with prior approach by Forget et al. [6, 7]
2. Form email address-like password by appending second dictionary word with "@" symbol to prime pass-

Rule	Shannon	NIST
$1^{\dagger}$	48.20	27
$2^{\dagger}$	47.80	25
$3^{\dagger}$	51.54	33
4^{*}	55.83	39

Table 1: The approximated Shannon and NIST entropy [3] of our rules (prime password dictionary size 118,000† words or 40,000* nouns; all prime passwords with length six).



Password Preferences

Please rate every generated passwords based on your preferences. (1: not preferred; 5: highly preferred)

p!a1sw#ord{ ○ 1 ○ 2 ○ 3 ○ 4 ○ 5

password@orange.kwn ○ 1 ○ 2 ○ 3 ○ 4 ○ 5

passBOOKwordSKY ○ 1 ○ 2 ○ 3 ○ 4 ○ 5

2Password1Tree7Moon ○ 1 ○ 2 ○ 3 ○ 4 ○ 5

Figure 2: Asking participant preference for generated passwords.

word and appending three random characters with "." at end. The goal is to create passwords that have a similar pattern to an email address and it may be easily memorable

3. Break the prime password at a random hyphenation point, then insert first dictionary word in uppercase at this hyphenation point, and add a second dictionary word in uppercase at end of the prime word. This rule is similar to base16, but uses dictionary words and linguistic feature hyphenation.

4. Create "Baking recipe"-like password by concatenating multiple number + noun pairs. This generates passwords that are structured like baking recipes, which might also increase memorability.

The strength of those generated passwords depends on how complex the rules are and how big the input space of the prime dictionary is. We will use the approximate of Shannon entropy [5] and the NIST entropy estimate [3] to measure our rules' complexity. The Shannon entropy equation provides a way to estimate the average minimum number of bits needed to encode a string of symbols, based on the frequency of the symbols. Therefore, our rules have to decrease the possibility of guessing by increasing the amount of entropy. Table 1 shows the calculated entropy for our example rules using prime passwords of length six, that means they are lower bounds for our entropy.

The implemented rules of our proposed method generate passwords with high entropy compared to common user passwords (for instance Weir et al. [13] show that most passwords in the RockYou database have 14–21 bits NIST entropy [3]). Further, it has been shown that entropy is not an effective metric for password security [9, 8, 1]. Thus, to evaluate the strength of our generated passwords we will use the John the Ripper (JtR) password cracker with our mangling rules and dictionary of prime passwords to estimate how many passwords JtR can crack with a maximum number of guesses (for example, 1 billion guesses [13]), using different cracking strategies [12].

Studying memorability and user preference. We will evaluate the memorability and user preference of our scheme with a large-scale online user study on Amazon MTurk. Like prior password studies [8, 9], our user study has two parts.

In the first part, there are four generated passwords that are provided by our generator, the participants have to pick one of those, basically, we assume the more memorable password will be chosen by the users. We ask the participants to not write down or save the selected password. To help the participants to memorize the chosen password, we ask them to re-enter the password a few times and also do a small distraction task. Then we will investigate the participants' preferences about the generated passwords using likert scales (see Figure 2). The highest score will show which proposed mangling rules will be most acceptable by participants. Future work may focus on designing different mangling rules that resemble the ones that had the highest preference score in our study.

In the second part of our user study, the participants have to come back to our website one week later and re-enter the selected password. Thus, we can quantify how well users can remember the selected passwords and how many tries are needed to remember the selected password from the previous week.

REFERENCES

1. Joseph Bonneau. 2012. Statistical Metrics for Individual Password Strength. In *Proceedings of the 20th International Conference on Security Protocols (SP'12)*. Springer-Verlag, Berlin, Heidelberg, 76–86. DOI: http://dx.doi.org/10.1007/978-3-642-35694-0_10

2. Joseph Bonneau, Cormac Herley, Paul C. van Oorschot, and Frank Stajano. 2012. The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy (SP '12)*. IEEE Computer Society, Washington, DC, USA, 553–567. DOI: http://dx.doi.org/10.1109/SP.2012.44

3. William E. Burr, Donna F. Dodson, Elaine M. Newton, Ray A. Perlner, W. Timothy Polk, Sarbari Gupta, and Emad A. Nabbus. 2011. *SP 800-63-1. Electronic Authentication Guideline*. Technical Report. Gaithersburg, MD, United States.

4. Xavier de Carné de Carnavalet and Mohammad Mannan. 2014. From very weak to very strong: Analyzing password-strength meters. In *Network and Distributed System Security Symposium (NDSS 2014)*. Internet Society.

5. Sascha Fahl, Marian Harbach, Yasemin Acar, and Matthew Smith. 2013. On the Ecological Validity of a Password Study. In *Proceedings of the Ninth Symposium on Usable Privacy and Security (SOUPS '13)*. ACM, New York, NY, USA, Article 13, 13 pages. DOI:http://dx.doi.org/10.1145/2501604.2501617

6. Alain Forget, Sonia Chiasson, P. C. van Oorschot, and Robert Biddle. 2008a. Improving Text Passwords Through Persuasion. In *Proceedings of the 4th Symposium on Usable Privacy and Security (SOUPS '08)*. ACM, New York, NY, USA, 1–12. DOI: http://dx.doi.org/10.1145/1408664.1408666

7. Alain Forget, Sonia Chiasson, P. C. van Oorschot, and Robert Biddle. 2008b. *Persuasive Technology: Third International Conference, PERSUASIVE 2008, Oulu, Finland, June 4-6, 2008. Proceedings*. Springer Berlin Heidelberg, Berlin, Heidelberg, Chapter Persuasion for Stronger Passwords: Motivation and Pilot Study, 140–150. DOI: http://dx.doi.org/10.1007/978-3-540-68504-3_13

8. Patrick Gage Kelley, Saranga Komanduri, Michelle L. Mazurek, Richard Shay, Timothy, Vidas Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Julio Lopez. 2012. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In *IEEE Symp. Security & Privacy*.

9. 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 *CHI'11*.

10. J. Leversund. 2010. The Password Meta Policy. http://securitynirvana.blogspot.com/2010/02/password-meta- policy.html.
(2010).

11. 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 *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*. USENIX Association, Ottawa, 123–140. `https://www.usenix.org/conference/soups2015/proceedings/presentation/ur`

12. Blase Ur, Sean 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 *USENIX SEC 2015: USENIX Security Symposium*.

13. Matt Weir, Sudhir Aggarwal, Michael Collins, and Henry Stern. 2010. Testing Metrics for Password Creation Policies by Attacking Large Sets of Revealed Passwords. In *CCS'10*. ACM.

14. M. Weir, S. Aggarwal, B. d. Medeiros, and B. Glodek. 2009. Password Cracking Using Probabilistic Context-Free Grammars. In *30th IEEE Symposium on Security and Privacy*.