Improving Interface Designs to Help Users Choose Better Passwords

Richard M. Conlan
HCI Laboratory
College of Computer & Information Science
Northeastern University
360 Huntington Ave, 202WVH
Boston, MA 02215 USA
kaige@ccs.neu.edu

Peter Tarasewich
College of Computer & Information Science
Northeastern University
360 Huntington Ave, 202WVH
Boston, MA 02215 USA
tarase@ccs.neu.edu

CHI 2006, April 22–27, 2006, Montreal, Québec, Canada.
ACM 1-59593-298-4/06/0004.

Abstract

Conventional wisdom seems to have concluded that traditional passwords are inherently insecure. The argument is usually that users choose bad passwords and cannot be expected to remember strong passwords. We feel that these conclusions are premature and that this argument is flawed. At present most password selection mechanisms are not designed according to basic HCI principles and we believe that this is highly responsible for the above conclusions. Our current research is reexamining the problem of password selection and memorability through the exploration of password selection mechanisms with novel interface designs. The goal of this research is develop both principles and designs that help users to choose passwords that are both memorable and secure.

Keywords

security, passwords, user-centered design, HCI, HCISEC, proactive password checking, interface design, usability

ACM Classification Keywords

H.5.2 Interfaces and Representation: User Interfaces – *Graphical user interfaces*; K.6.5 Computing Milieux: Security and Protection - *Authentication*.

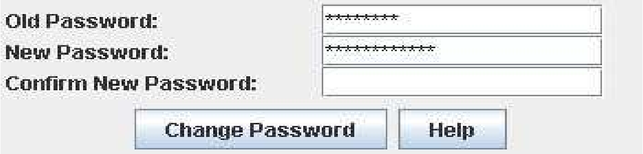
Introduction

Conventional wisdom seems to have concluded that traditional passwords are inherently insecure. This argument has been adopted by major organizations such as Microsoft [8] and RSA Security [11], and is reflected in much of the literature. The argument is usually that users choose bad passwords and cannot be expected to remember strong passwords.

We feel that these conclusions are premature and that this argument is flawed. At present most password selection mechanisms (PSMs) are not designed according to basic HCI principles and we believe that this is highly responsible for the above conclusions. Our current research is reexamining the problem of password selection and memorability through the exploration of PSMs with novel interface designs.

Current Password Selection Mechanisms

The current de facto standard PSM is similar to that depicted in figure 1. The user enters their old password, the new password, and the new password again to confirm they have not made a typo and then clicks the *Change Password* button. On many systems that is all there is to it.



Old Password: *********

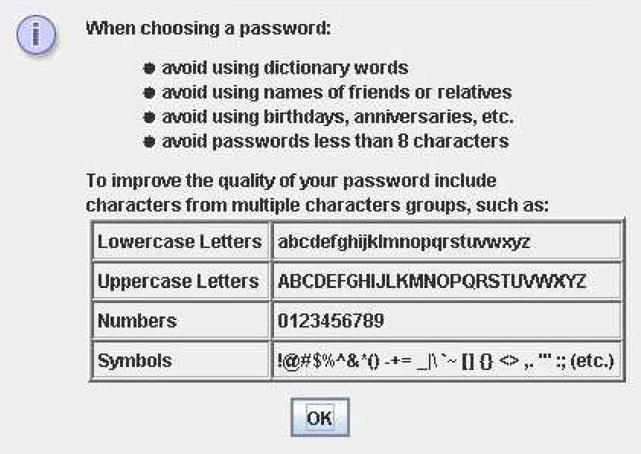
New Password: *************

Confirm New Password:

figure 1. A typical password selection mechanism.

Some systems additionally enforce minimum complexity requirements. On these systems the user's password must meet these requirements to be accepted, and if it does not the user will be presented with an error message upon clicking the *Change Password* button. A typical error message may read something like "The password supplied does not meet the minimum complexity requirements."

Usually there is not a *Help* button, but when there is it typically offers the user some generic suggestions such as those shown in figure 2.



When choosing a password:

- avoid using dictionary words
- avoid using names of friends or relatives
- avoid using birthdays, anniversaries, etc.
- avoid passwords less than 8 characters

To improve the quality of your password include characters from multiple characters groups, such as:

Lowercase Letters	abcdefghijklmnopqrstuvwxyz
Uppercase Letters	ABCDEFGHIJLKMNOPQRSTUVWXYZ
Numbers	0123456789
Symbols	!@#$%^&*() -+= _|\`~ [] {} <> ,. "' :; (etc.)

figure 2. A typical password selection help message.

Analysis of Current Selection Mechanisms

PSMs such as those shown above are incredibly common, and yet they violate basic tenets of human computer interaction (HCI) and well-known design principles. The purpose of the PSM is to allow the user to select a new password. It could be argued that it is designed well for this purpose since it is obvious where to enter the old password, the new password, and what button to select. However, as corporations and individuals have become more concerned about security the PSM has gained the additional purpose of ensuring that the user selects a *secure* password.

Unfortunately, the standard PSM is not well suited to this task – a simple PSM offers no security context at all. To address this shortcoming password complexity constraints were added to the simple PSM model. But these merely create a security threshold – they do not help the user to choose a good password. In fact, passwords that fail to meet the threshold often result in a response such as "The password failed to meet complexity requirements," which offers so little information that the user does not necessarily know how to proceed. This is a clear violation of the third golden rule of interface design – *offer informative feedback* [14].

The current design of PSMs does little to help the user choose a good password. At best the user can keep trying passwords until they find one that works, but has no sense of progress from one attempt to the next. The findings in [4] and [13] support the claim that current PSMs have failed to adequately incorporate usability principles into their design.

Analysis of Users

In general it is presumed that users choose bad passwords because they cannot remember good passwords. However, research has shown that users can remember good passwords when given suitable instruction on password construction [1,6,17]. Another common assumption is that users choose bad passwords because they do not care about security. But research indicates that this view is also false [1,4,15].

More generally, research has found that users do not have a good understanding of the difference between a strong password and a weak password [3,6]. Even when the user has rough guidelines to go by, such as "try to include symbols", they tend to choose poor passwords such as "Juliet03" and "Justin!" [17]. There is evidence that even technically savvy users lack a clear sense of what makes for a strong password [7,9].

Merely assigning randomly generated passwords is not a viable solution. Various avenues of research indicate that users have trouble memorizing random data/passwords [3,5,17]. In these situations users often just write down the passwords, which arguably introduces even greater risks to privacy and security.

The Need for Secure and Usable Passwords

Passwords are the primary, and often only, authentication mechanism for many systems. While industry and academia are both hard at work on smart cards, biometrics, and other advanced authentication mechanisms, many of these technologies have yet to see widespread adoption. At the RSA 2005 conference when asked if this is the year businesses would move away from passwords Ron Rivest responded, "Passwords will be with us forever." Also at the RSA conference, Amit Yoran stated, "We've got to make security simpler to use if it's going to be effective." [12] Further, numerous researchers in this field have called for improvements in the PSM [3,17].

Improving Password Selection Mechanisms

The usability problems plaguing current PSMs can be summarized as those that occur because the user does not know how to choose a good password, and those that occur because the user does not receive appropriate feedback while choosing a password. The former reflects the *Gulf of Execution* and the latter the *Gulf of Evaluation* [10].

Addressing the Gulf of Execution in PSMs

To address the gulf of execution it is necessary to devise methods that help the user choose a good password. Most methods we are aware of are ambient – they rely on general user education or on a well drafted and communicated password policy. As indicated in [15], these are good approaches. But they also appear to apply most directly to well organized corporate environments and do little for less organized companies or the home user.

When incorporating this into a PSM we can consider *passive*, *static*, and *dynamic* mechanisms. An example of a passive mechanism would be the *Help* button shown in figure 1. An example of a static mechanism would be to include advice directly on the PSM screen or to popup a dialog beforehand. A dynamic mechanism would be one that dynamically updated itself to offer advice on how to improve the current password.

We believe that the dynamic approach would be the most effective, and this reflects the recommendations given in [3]. To demonstrate this it would be necessary to implement candidate PSMs and subject them to appropriate user testing. Such testing would have to explore not only the different approaches, but the content of the advice as well.

Addressing the Gulf of Evaluation in PSMs

To address the gulf of evaluation it is necessary to devise feedback mechanisms that help the user to understand the quality of their password. Most PSMs we are aware of just provide the user with an accepted/denied response. We are aware of only two PSMs that offer dynamic feedback, both by including a progress bar that dynamically adjusts to indicate password quality: PGPkeys [1] and Gmail [2].

When incorporating a feedback mechanism into a PSM there are three primary considerations. The first is when to provide the feedback – either upon password submission or dynamically as the password is being typed. The second is what manner of feedback to provide – possibilities include textual feedback, a progress bar, an avatar, etc. The final consideration is what algorithm to use to evaluate the password quality.

We believe that the dynamic approach is better than waiting for password submission because the immediate feedback gives the user the ability to interactively improve their password quality. This coincides with research indicating that users are better able to remember complex passwords when they personally construct them [3,6,17].

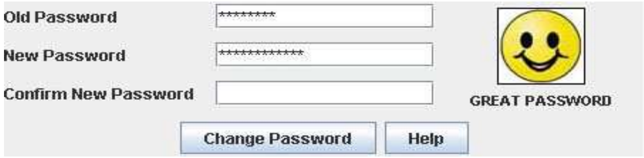
It remains unclear which manner of feedback mechanism to recommend, as there is a rich field of possibilities. Likewise, it is unclear which algorithm is best for dynamically determining password quality. Most algorithms for proactive password checking in the literature focus on performing dictionary attacks [16]. The method suggested in [16] is intended to be more dynamic, but like the dictionary attacks it does not lend itself to this purpose because the result is boolean, whereas adequate feedback requires an algorithm that returns a qualitative result. See the next section for the details of the feedback mechanisms and algorithm we are currently employing.

Current Research

Our current research is focused on improving the gulf of evaluation in PSMs, as we see this as the most fruitful area for improvement. We are currently focusing on the different types of feedback mechanisms, though we had to develop a qualitative algorithm to support these efforts.

Feedback Augmented PSMs

We have developed four PSMs as Java applets. Each PSM has the same fields, buttons, and basic layout, with the *Help* button displaying the information from figure 2. The applet shown in figure 1 represents the standard PSM. Another follows the approach used by PGP and Google by providing a progress bar that dynamically reflects the quality of the password, with a textual indicator that is updated at set thresholds. The third includes the same textual indicator along with a simple avatar that progresses from sad to happy as the quality of the password improves – depicted in figure 3. The fourth makes use of *fear appeals* as described in [15] by giving an estimate of how long it would take an adversary to break the current password, although the estimates only change at the same threshold as the text indicators in the previous two applets[1].



Old Password *********

New Password **************

Confirm New Password

figure 3. Avatar feedback password selection mechanism.

Qualitative Feedback Algorithm

The algorithm we have employed builds on the observations made in [6,9,17]. In particular, that password length and the amount of entropy are the key factors determining password strength. The algorithm we have employed scores each character of the password as worth a certain number of points. The number of points a character is worth depends on the size of the character set a password-cracking program would have to use to brute-force the password. For instance, for a password composed of lowercase letters each letter is worth only twenty-six points, but for a password composed of uppercase and lowercase letters each letter is worth fifty-two points. We divided the keyboard characters into a series of character classes, and including a character from a given class ups the number of points each character in the password is worth by the size of the newly included character class. This algorithm has obvious room for improvement, and we hope that further research will result in more refined and accurate feedback.

Experiment Design

We have incorporated the PSM applets into a homework submission system that is currently in use by two classes, giving us a population of roughly eighty users. The users must choose a password on account creation, and will have to reset their password three times over the course of the semester.

We will be randomly dividing the user population into two groups, A and B. The members of group A will be assigned a PSM at account creation and will consistently see the same PSM throughout the semester. The members of group B will be assigned a different PSM at each prescribed password change. The

[1] These applets are available online at http://tinyurl.com/bkfbq.

applets will be randomly assigned such that roughly one-forth of the users in group A will use each applet, and so that all of the users in group B will see all of the applets but in different orderings.

The system will also keep track of whether the user clicked the *Help* button on each password change, the number of valid and invalid login attempts, and the number of times the user clicks the *Forgot Password* mechanism included in the system.

At the end of the semester we will debrief the students on the study and allow them to opt-in or opt-out at their discretion. Our hope is that enough students will opt-in that we will be able to perform a within-subjects analysis for group A and a between-subjects analysis for group B. Students that opt-out will have their records destroyed without examination. Students that opt-in will have the quality of their passwords analyzed.

Acknowledgements

We would like to thank Min Wu and Professor Rob Miller of MIT's CHISEC reading group for their valuable input during discussions on some of the topics presented in this paper. We would also like to thank Sera Galvin for providing the graphic images used in the applets.

References

[1] PGP Corporation's PGP Desktop. Available at: http://tinyurl.com/8fh38.

[2] Google's Gmail. Available at: http://tinyurl.com/2gvnd.

[3] Adams, A., Sasse, M. A., Lunt, P. *Making Passwords Secure and Usable*. In *Proc. ACM XII*, Springer-Verlag, London, UK, 1997, 1-19.

[4] Adams, A., Sasse, M. A. *Users are not the enemy*. In *Comm. ACM*, Vol. 42, No. 12, 1999.

[5] Dix, A., et al. *Human-Computer Interaction*. 3rd Ed. Prentice Hall, various locations, 2004.

[6] Gehringer, E. *Choosing Passwords: Security and Human Factors*. *ISTAS'02*, 2002, 39-373.

[7] Hairball. *Fun Password Facts*. 2600: The Hacker's Quartely, Vol. 19, No. 1, Spring 2002.

[8] Ilett, D. *Gates: Passwords passé*. CNET News.com. Nov. 16, 2004. Available at: http://tinyurl.com/bcqt5.

[9] Kaige. *Fun Password Facts Revisited*. 2600: The Hacker's Quarterly, Vol. 19, No. 3, Fall 2002.

[10] Norman, D. A. *The Design of Everyday Things*. Doubleday, New York, USA, 1988.

[11] *Passwords vs. Strong Authentication*. RSA Security. Available at: http://tinyurl.com/cru4a.

[12] Saita, A. *RSA 2005: Passwords at the breaking point.* SearchSecurity.com, Feb. 16, 2005. Available at: http://tinyurl.com/cf4so

[13] Sasse, M.A., Brostoff, S., and Weirich, D. *Transforming the 'weakest link': a human-computer interaction approach to usable and effective security*. BT Technical Journal, Vol 19 (3), Jul. 2001, 122-131.

[14] Schneiderman, B. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. 4th Ed. Addison-Wesley, various locations, 2004.

[15] Weirich, D., Sasse, M. A. *Pretty Good Persuasion: A First Step towards Effective Password Security in the Real World*. In *Proc. NSPW'01*, Cloudcroft, NM, USA, 2001, 137-143.

[16] Yan, J. *A Note on Proactive Password Checking*. In *Proc. NSPW'01*, Cloudcroft, NM, USA, 2001, 127-135.

[17] Yan, J., et al. *The Memorability and Security of Passwords – Some Empirical Results*. IEEE Security & Privacy Magazine, Vol. 2, Issue 5, Sep. 2004, 25-31.