Submission to Workshop on Human-Computer Interaction and Security Systems – April 6, 2003 – Fort Lauderdale, Florida

Designing Secure Yet Usable Credential Recovery Systems With Challenge Questions

Mike Just
Treasury Board of Canada, Secretariat
Just.Mike@tbs-sct.gc.ca

ABSTRACT

We discuss the design of secure systems for recovery of a password, private keys, account privileges or other security credentials or entitlements at a time when a primary security credential (often a password) has been lost or is otherwise inaccessible. Automated recovery techniques can minimize help-desk costs, though efficiency can only be gained if the recovery process is usable. This paper discusses a classification and design of secure and usable challenge question and answer systems; in particular it identifies a distinction between fixed, controlled and open questions and answers.

INTRODUCTION

Considerable focus has been given over time to devising secure identification systems (see [6, Ch. 10]). Only recently has the usability of such systems been given more than a passing consideration. Most notably, recent effort has focused on password-replacement schemes, including picture-based [1, 3] and drawing-based [8].

In this paper, we focus on identification for a credential recovery process, as opposed to the more routine login identification process. Although each requires proper user identification, they differ in several aspects:

1. A recovery process will be executed less frequently.
2. Whereas a login process might rely on memorization, a recovery process should not have a similar reliance (since the recovery process will typically be invoked due to a user forgetting their login password).
3. It is acceptable for a recovery process to take more time (due to (1) above), whereas a login process should be completed more quickly.

We classify and analyze recovery techniques based on challenge questions (and answers), identifying three *types* for each: fixed; controlled; and open. An early attempt to address usability issues in this area was performed by Frykholm and Juels [5] where they examine ways in which the recovery mechanism can tolerate minor mistakes during answer presentation. We take a broader approach and describe a general model for challenge questions and compare instances of this model.

CHALLENGE QUESTIONS

"Challenge questions" or "password recovery hints" are commonly used as an automated means of password or more generally, credential, recovery. What typically was performed via a help-desk call may now be performed automatically through confirmation of a user's response to a previously stored question(s) and answer(s). At registration, or subsequent time when a user is properly identified, the user selects a question for which they also submit a corresponding answer (*answer registration*). During recovery (e.g. they have forgotten their password), they are challenged with their question, and required to provide the appropriate answer (*answer presentation*).

Clearly, challenge questions can offer the same ability to impersonate a user, as does credential compromise through other means (e.g. "shoulder surfing" in order to observe the entering of a password). Similarly, they offer the same potential for abuse in case the system is not usable (e.g. "writing down the password"). And if not usable, then users may also be unwilling or unable to automatically recover, thereby triggering more expensive, manual recovery (e.g. through help-desk). Thus, the security and usability of the system is a major concern.

CHALLENGE QUESTION MODEL

In this section, we highlight different types of questions and answers and examine ways in which these questions and answers can be used in a credential recovery system.

For our usability analysis below, we focus on the memorability and repeatability of the questions and answers, and also, consider their effect on user (and administrator) flexibility and convenience. With regards to security, although some common security requirements will be highlighted, our analysis will focus on ensuring a usable system as the means for ensuring security since a usable system is often used more securely.

Question Types

The two, likely most familiar, types of questions are fixed and open questions. A *fixed question* system provides a list of administrator-chosen questions to a user, where the user's choice of question can only be taken "as-is" from this list. At the other extreme are *open questions* whereby a user has complete choice and control over the question – guidance, as to the question construction, may be provided to the user but the user enters the question in free-form text.

A *controlled question* consists of a question whose content is partially fixed, though modifiable by the user (thereby combining ideas from a fixed and open question). There are variations as to how a controlled question might be constructed. We suggest two such variations below:

1. The fixed question might allow for additional text to be added, forming a modification of the original question. The modified question would be presented to the user as part of credential recovery. For example, to the original fixed question "What is _____'s favourite food?", the user might add an appropriate name to produce the modified question: "What is Ellen's favourite food?".
2. The fixed question might support a combination with an optional user-provided hint, where the hint would be presented to the user as part of credential recovery. For example, to the original fixed question "What is a memorable date for you?", the user might provide the following hint: "Dog". The question and the hint would be provided to the user upon recovery. For the user, this hint might indicate a special date associated with their pet dog, such as its date of purchase.

The primary advantage of a controlled question is that it permits a shorter list of general questions to be constructed by the system manager. This inherently provides some guidance for the user (relative to an open question) and allows further personal customization.

We use the phrase *variable question* to refer to a question that is either controlled or open. Below, we discuss some security and usability issues related to each question type.

1. *Security*. With a fixed question, users are prevented from poor question selection, e.g. "What colour are my eyes?" This is a poor question since the resulting answer space is insecure, resulting from low entropy. Thus, a security advantage is provided since the likelihood of choosing a "bad question" is reduced. With an open question, users might select a question that is "bad", though capable users are able to select more secure questions. For example, they are able to customize questions directly related and meaningful to their childhood, e.g. "What was my grade 8 locker combination?" Controlled questions offer a balanced alternative helpful for question design in case an exhaustive list of suitable fixed questions cannot be constructed. For example, a question may be as simple as "Enter a number that is memorable for you" (giving some content control and guidance for the user) while the user can provide the hint "Grade 8 locker", thereby providing some equivalence to the open question described above. However, controlled questions also share the weaknesses of open questions as the question or hint entered by the user can be insecure (as is arguably the case when asking the user to enter a number as in the above example).
2. *Usability*. With fixed questions, users are not required to construct their own questions at registration. This offers both an advantage and disadvantage depending on the ability and desire of a user to choose their own questions. An open question would offer similar disadvantages and advantages. As discussed above for the security issues, a controlled question allows some guidance to be provided for the user, in the form of a general yet partially focussed question, while allowing some flexibility via customization. Repeatability and memorability of the hint are not a concern since the hint is shown to the user upon answer presentation.

Answer Types

A similar distinction applies for fixed, controlled and open answers as did for questions. In addition, it is helpful to distinguish between the registration and presentation of answers (though for this section, assume that the same answer type is used for both registration and later presentation of answers - variance to this assumption is discussed below). A *fixed answer* set involves user selection of an answer from a pre-set list of possible answers. At the other extreme, an *open answer* would involve a user manually entering their response. Guidance may be provided as part of answer registration, but the answer is entered in free-form by the user.

A subtle variation is a *controlled answer* whereby the answer space is not quite fixed or open, but controlled. Some ways in which this might be achieved are:

1. A fixed set of answers is provided, but the answer space is large enough so that most potential answers are allowed (though the usability of long lists must be considered). For example, in case a user answers with a geographic location, the answer may be entered using drop-down menus listing all possible cities, provinces/states and countries for some region.
2. The user is able to enter an answer, but the format of the answer is controlled – answers that do not conform are rejected. For example, a user might be asked to provide a memorable numeric value, so that alphabetic and punctuation characters would not be permitted for inclusion in the answer text.

We use the phrase *variable answer* to refer to either a controlled or open answer. Below, security and usability issues are examined for each answer type.

1. *Security*. With a fixed answer set, users are prevented from selecting insecure answers. For example, for a given fixed question there may be a highly probable common answer that should be disallowed by the system else it might be easily guessed by an attacker. With open answers, larger variation in the answer space is provided, though for certain questions, a user would be able to select highly probable answers. There do not seem to be any significant security advantages offered by using a controlled answer other than supporting a large answer space.
2. *Usability*. With a fixed answer list, memorability and repeatability may be hampered if there is no unique answer to satisfy a user's preference (either (i) the

user's first choice is not available, or (ii) more than one satisfactory choice is available). With an open answer list, memorability and repeatability may be better than fixed, though also problematic if the registered answer is ambiguous (e.g. "St." versus "Street"). Controlled answers offer an alternative whereby a large answer space can be used, but control over the possible values improves repeatability. For example, a question that asks for a particular date to be entered can provide an interface with drop-down menus for the year, month and day, thereby avoiding potential ambiguities with date formatting if a free-form date were to be entered.

Discussion

Various combinations of the three types of questions and answers can be used. For example, it seems that almost any type of question could be combined with any type of answer, though it seems difficult to support a fixed answer set for an open question. A further distinction between answer registration and presentation is described below.

Answer Registration versus Presentation

Answer registration refers to the submission of the answer (and corresponding question) at an occasion when the user has been properly identified. *Answer presentation* refers to the submission of answers for the purpose of identification and credential recovery. As discussed below, there are variations whereby the registered answer is modeled after one answer type while presentation may be modeled after another. This distinction might be helpful for understanding the variety of question and answer systems that could be devised.

The fixed-fixed and variable-variable combinations are likely the most familiar and common. The remaining alternatives offer novel variations, though only the variable-fixed alternative seems to offer practical use.[1]

- *Fixed-Variable*. A fixed answer set is provided to the user when registering, while a controlled or open answer input is provided upon presentation for recovery. Though security may be improved, usability is decreased (relative to use of a fixed answer set) as this option negatively affects repeatability. This seems unnecessary when one could simply support fixed answer presentation.
- *Variable-Fixed*. Variable answers are supported for answer registration (offering flexibility) while to overcomes issues of repeatability and memorability, fixed answers are supported as part of answer presentation.

Expanding upon the *variable-fixed option*, a likely implementation might involve the storage of a set of "fake answers" along with the user's given answer upon registration. At answer presentation, the user's answer would consistently be presented along with the same set of fake answers. There are numerous issues to consider regarding the secure implementation of such a system:

- The "fake answer" sets cannot be repeated across users else an attacker can easily determine the fake answer sets (and hence, eliminate and recover the user's submitted answer) by attempting to recover two or more users.
- The fake answer set must be consistent from one recovery attempt to the next else an attacker could identify the user's answer as the only consistent answer across a number of recovery attempts.
- The fake answer set must be changed should the user choose to modify their submitted answer, else an attacker (aware of a potential answer update) could determine the user's answer from the variance in the answer sets from before and after the update.
- Care must be taken in the selection of the fake answer sets for each user so that the user's submitted answer is sufficiently concealed by the fake answers. For example, supposing the user is asked the question "What is your favourite fruit?" but answers with the word "mushroom". In this case, if only fruits were provided as part of the fake answer set, then the user's submitted answer would be easily distinguishable. Optionally, "incorrect" fake answers might be provided in order to anticipate any user variance and serve to confuse would-be attackers.
- The size of the fake answer set should be large enough to resist exhaustive guessing attacks against the user.

In addition, only this variation requires that the user's submitted answer not be hashed, as it must be presented to the user as part of answer presentation. For each of the above reasons, this variation is not yet sufficiently mature.

Multiple Questions

For reasons of security, it is often necessary that more than one question-answer pair be registered by a user. However, usability tends toward requiring fewer questions. (For example, citizen focus testing as part of Canada's Government OnLine (GOL) initiative indicates that users would prefer at most three questions.). It is likely though that the usability of the system must be favoured, whereas additional security measures can thereafter be enforced.

Variations can also be described where the number of questions presented at recovery is less than the number of questions registered. There are at least two models:

1. The user registers n questions, but is presented only $t < n$ questions upon recovery. All t questions must be

[1] The variable-fixed combination was first described to the author by Fiona Bremner.

properly answered in order for the recovery process to continue.

2. The user registers n questions, is presented $t < n$ questions upon recovery. Differing from (1), only $r < t$ questions must be properly answered in order for the recovery process to continue.

(1) is an attempt to offer a level of security equivalent to that of n questions, but offer some usability benefit at the time of recovery, by presenting only $t < n$ questions to the user. However, the usability benefits only reduce the time required for recovery and do not affect the arguably more important concerns of memorability and repeatability (the user still has to remember the answers for n questions). Since recovery would typically be rarely performed, it seems that little benefit is gained. The purpose of (2) is to tolerate mistakes upon entering. However, it seems that an additional question is being used to tolerate such mistakes whereas a more usable system might attempt to reduce the number of questions used.

In general, establishing the question number seems to amplify the traditional contrast between security and usability. Based on the considerations above, it seems that a system in which a user registers and is asked n questions offers a simple and usable solution. The benefits of any variations don't seem to offer sufficient gains in security nor much improvement in usability.

For the set of registered questions, some form of "question grouping" might be beneficial. For example, supposing that three questions are to be registered, it may be advantageous to require that one fixed and two controlled question be selected. Alternatively, questions might be classified based on their topic so that users might have to select one question that required them to enter a "date" response, while the second might require a numeric response and the third, an alphabetic response. Finally, if one can classify their questions based on their security strength, then the system could offer multiple classes whereby the user must select one question from each class as part of registration.

CONCLUDING REMARKS

We've described a challenge question and answer model, in which three types of questions and answers have been identified, namely fixed, controlled and open types. In addition, we've discussed different instances based on this model.

There are additional security measures that can greatly improve the usability of a challenge question system, e.g. by reducing the number of questions required. This includes a system lockout feature whereby access to the recovery functionality is reduced or removed after a number of failed attempts. *Graduated lockout* would reduce access over time, say locking out recovery *for a fixed period of time* after some number of failed recovery attempts whereas the recovery might be fully blocked after some number of temporary lockouts. Of course, the denial of service implications must be carefully considered.

Reverse turing tests (e.g. CAPTCHA [2]) help reduce the likelihood of success for online attacks. However, the technology is still in early development [7].

There is little literature that discusses credential recovery, and in particular, the use of challenge questions. The work of Ellison et al. [4] and Frykholm and Juels [5] are notable exceptions. The latter seems to be the only paper to focus on usability issues. Our work more generally and thoroughly considers issues related to challenge question system design.

Future work could expand on our model and further analyze specific instances. In addition, detailed research into suitably secure and usable question content (e.g. subjective vs objective questions) would be beneficial. More generally, investigation into other means of credential recovery, including biometric systems (e.g. voice biometrics by phone) or recognition-based systems (e.g. based on pictures) is warranted.

ACKNOWLEDGEMENTS

Thanks to Fiona Bremner (Claymore Software), Paul Van Oorschot (Carleton Univ.), Rick Brouzes, Jen McTavish and John Weigelt (Treasury Board of Canada) for helpful discussions and comments on this paper.

REFERENCES

1. S. Brostoff, M. A. Sasse, "Are Passfaces More Usable Than Passwords", in *Proceedings of HCI 2000*, pp. 405-424, August 2000.
2. The CAPTCHA Project, http://www.captcha.net/.
3. R. Dhamija, A. Perrig, "Déjà Vu: A User Study Using Images for Authentication", in *Proceedings of the 9th Usenix Security Symposium*, August 2000.
4. C. Ellison, C. Hall, R. Milbert, B. Schneier, "Protecting Secret Keys with Personal Entropy", *Journal of Future Generation Computer Systems*, 16(4), pp. 311-318, February 2000.
5. N. Frykholm, A. Juels, "Error-Tolerant Password Recovery", in Proceedings of the ACM Conference on Computer and Communications Security 2001 (CCS'01), November 2001.
6. A. Menezes, P. van Oorschot, S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.
7. G. Mori, J. Malik, "Up to the Challenge: Computer Scientists Crack a Set of AI-Based Puzzles ", *SIAM News*, November 2002.
8. I. Jermyn, A. Mayer, F. Monrose, M. Reiter, A. Rubin, "The Design and Analysis of Graphical Passwords", in *Proceedings of the 8th USENIX Security Symposium*, pages 1-14, August 1999.