Generating Natural Language Challenge Questions to Improve the Security of Password Recovery Systems

Chuong H. Ngo
College of William and Mary
College Address
chngo@email.wm.edu

Abstract

Abstract to be written.

1. Introduction

Introduction to be written.

2. Background

In today's world of internet commerce and web services, it is not uncommon for an individual to have multiple user accounts for various systems providing different services like banking, e-mail, and social media. Each of these systems may have different authentication systems that require their users to remember some kind of user id and password, which was setup upon the creation of the user's account. Ideally, the user would never forget their user id or password. Of course, in the real world, that is not so and some method of credentials recovery or reset need to be provided.

Miller [7] examined several different techniques to allow user to recover their password and outlined their advantages and disadvantages. Of the techniques Miller described, the use of challenge questions is the focus of this paper. Called "Question and Answer" in Miller's paper [7], this technique prompts the user for some piece of information that only they should know to authenticate them. That information may have been recorded by the user at the time of account creation or was assigned to them by the service like an account id. As Miller noted, because the answers to the challenge questions are not sent to the user via e-mail or post, they cannot be intercepted [7]. This make this technique effective against opportunistic attacks. However, because the answers to this challenge questions are often times some personal bit of information, targeted attacks are quite effective against it [7].

One case study for the use of challenge questions is the online banking systems offered by Lloyds Banking Group and the Royal Bank of Scotland group. Both systems rely on personal questions for their password recovery system. Upon examination of the two systems, Smyth [10] concluded that both systems used questions that were weak and could be easily exploited. In the case of Lloyds Banking Group, the user had to provide 2 of the following pieces of information [10]:

1. Father's first name
2. Mother's first name
3. Place of birth
4. First school

The Royal Bank of Scotland Group gave the users a few more options, requiring the user to authenticate by providing various pieces of customer information [10]:

1. Name
2. Date of birth
3. Sixteen digit card number
4. Three digit card security code
5. Sort code
6. Account number

As noted by Smyth, such information were either available in the public domain or should be considered in the public knowledge [10]. Thus, the assumption needed to be made that an attacker either knows the information or can readily get access to them. This problem is exacerbated by the use of social media. In examining over 200 personal security questions from 20 financial websites, Rabkin [8] found that about 12% of the questions in the study could

be answered using information obtained on on social networking websites. Additionally 0.3% of the questions could be answered the first time using random guessing with no personal information to draw from and subsequent guesses increases that likelihood of a correct guess [8]. Utilizing knowledge gained from targeted attacks and personal acquaintance allowed for a success rate of 38% when answering the first time [8]. Rabkin also found that users treated memorability as the dominant factor when choosing security questions, resulting in answers that may be easier to guess [8].

Another problem that challenge questions have is applicability and repeatability. Forcing users to choose from a pool of questions may lead to the user finding some questions to not be applicable to them [5], reducing the memorability of the answer. Other questions may have answers that can be formatted in multiple ways, like addresses. However, if the system does not account for multiple formats, this can also reduce the memorability of the answer. Users can be allowed to craft their own questions, potentially increasing memorability and security. However, as noted by Just et al. [5], even when allowed to craft their own questions, only 20% of users managed to achieve a mostly high security rating on their three questions [5]. When looking at only the first question, only 3% of the users crafted a question that was not rated low in terms of security [5]. Thus, some method is needed to make challenge questions that are secure while not sacrificing too much memorability.

3. Related Works

Early systems like BASEBALL [3] and LUNAR [12] were natural language front ends that facilitated user interaction with a structured database by taking the user's natural language question, parsing it, and retrieving data from it. Other systems like SHRDLU [11], GUS [1], and MIT's Jupiter [13] system took things one step further and interfaced with user through dialogue interaction and retrieved data from a structured database [11] [1], in the case of SHRDLU and GUS, or by parsing on-line weather information [13], in the case of Jupiter. While these systems seek to answer questions presented by the user, as oppposed to generating a natural language question for the user to answer, they are still important contributions to this paper since they are still processing the natural language question asked by the user. This process of information extraction, defined as the activity of filling predefined templates from natural langauge texts [2] [4], is also key to this paper's approach as the way obtain the information from which to generate questions.

Of course, once the data has been gathered from which questions can be generated, the trick is still to generate the actual natural language questions. This brings some extra considerations that must be dealt with. As noted by McKeown [6], to generate natural language texts, a system must:

1. Consider the intention of the writer as to select the appropriate words. For example, using the passive form "given" as opposed to the active "give" in a sentence.

2. Consider what information needs to be presented and in what order. The first part, knowing the information to be presented, is quite obvious, the system must now what it wants to convey via text. The second part is a little less obvious as the ordering of statements may change the conveyed meaning. For example, there is a difference between "Spot is just a dog. Spot likes to run." and "Spot likes to run. Spot is just a dog." have slightly different connotations of meaning.

3. Consider when to use pronominal reference and the syntactic construction to be used. Generating text where all references to a named entity is done with a proper noun makes the text sound artificial.

Those problems can be dealt with using a discourse strategy at the time of text generation [6]. Discourse strategies, using rhetorical predicates as its basic unit, comes from the observation that follow different standards for discourse organization depending upon the goal of that discourse [6]. For example, a technical paper and a narrative may both present the same information, but in different ways. Finally, Radev et al. [9] tackled the problem of providing a natural language summary of multiple natural language sources of information.

4. Approach

As previously stated in the Background section, the two techniques currently used for challenge questions is to present users with questions from a pre-determined pool or from a pool of questions that the user crafted. Both of these techniques suffer in terms of security for a number of reasons already stated. These reasons include answers that rely on public knowledge, answers that can be easily obtained through social aquantances and targetted attacks, and users prioritizing memorability over security when choosing challenge questions and answers for them. Additionally, even if all the reasons listed did not hold for a particular user account, the likelihood of an attacker correctly guessing an answer increases as the attacker is presented with the same challenge questions over and over again. Thus, there also needs to be sufficient entropy in the questions presented to the user.

This paper proposes, as a solution to the problems outlined above, a system that generates natural language challenge questions based information gathered about the user

through information extracted from writing samples on various topics. This solves the problem of a having sufficient entropy in the questions presented as the generation of questions based upon a common pool of information allows for greater variety in the questions by simply using a different discourse strategy. Additionally, the questions can involve a combination of different pieces of information, further increasing the possibility space for generated questions. This combination of information will also increase the difficulty of guessing the answers to the generated challenge questions as greater familiarity with the target may be needed to do so. Therefore, the pool of people with the necessary knowledge to correctly guess the answers decreases and overall security increases. As the pool of information used to generate the questions grows, so should the security that comes from the combination of information.

In order to achieve this, the system will prompt the user to write a few paragraphs describing something close to them, like a treasured vacation memory. That writing sample will be fed through a basic natural language processing (NLP) system utilizing the Stanford NER at its heart. The NER will analyze the sample and extract from it the key entities for storage along with the domain or topic the user was asked to write upon. Then, natural language challenge questions will be presented for the user to answer. These questions will be based upon the information extracted from the writing samples gathered earlier. The user's answer will then be fed through the NLP system again, and the extracted named entities and ideas will be compared against what was stored to see if there is a match.

References

[1] D. G. Bobrow, R. M. Kaplan, M. Kay, D. A. Norman, H. Thompson, and T. Winograd. Gus, a frame-driven dialog system. *Artificial intelligence*, 8(2):155–173, 1977.

[2] R. Gaizauskas and Y. Wilks. Information extraction: Beyond document retrieval. *Journal of documentation*, 54(1):70–105, 1998.

[3] B. F. Green Jr, A. K. Wolf, C. Chomsky, and K. Laughery. Baseball: an automatic question-answerer. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*, pages 219–224. ACM, 1961.

[4] L. Hirschman and R. Gaizauskas. Natural language question answering: the view from here. *Natural Language Engineering*, 7(04):275–300, 2001.

[5] M. Just and D. Aspinall. Personal choice and challenge questions: a security and usability assessment. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, page 8. ACM, 2009.

[6] K. R. McKeown. Discourse strategies for generating natural-language text. *Artificial Intelligence*, 27(1):1–41, 1985.

[7] C. Miller. Password recovery, 2002.

[8] A. Rabkin. Personal knowledge questions for fallback authentication: Security questions in the era of facebook. In *Proceedings of the 4th symposium on Usable privacy and security*, pages 13–23. ACM, 2008.

[9] D. R. Radev and K. R. McKeown. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):470–500, 1998.

[10] B. Smyth. Forgotten your responsibilities? 2010.

[11] T. Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1–191, 1972.

[12] W. A. Woods. Progress in natural language understanding: an application to lunar geology. In *Proceedings of the June 4-8, 1973, national computer conference and exposition*, pages 441–450. ACM, 1973.

[13] V. Zue, S. Seneff, J. R. Glass, J. Polifroni, C. Pao, T. J. Hazen, and L. Hetherington. Juplter: a telephone-based conversational interface for weather information. *Speech and Audio Processing, IEEE Transactions on*, 8(1):85–96, 2000.