An Interactive Approach to Secure and Memorable Passwords

A Thesis
in TCC 402

Presented to

The Faculty of
School of Engineering and Applied Science
University of Virginia

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science in Computer Science

by

William H. Haubert III

March 25, 2002

On my honor as a University student, on this assignment I have neither given nor received unauthorized aid as defined by the Honor Guidelines for Papers in TCC Courses.

____
William H. Haubert III

Approved ____(Technical Advisor)
David Evans

Approved ____(TCC Advisor)
Roseanne Simeone

Table of Contents

Table of Figures

Abstract

Login schemes are required for authentication for access to a network. Most of the schemes use a username and password to allow users to login. Login schemes can be broken into two categories: those that are interactive and those that are passive. Many of the passwords schemes that are currently found in either category have flaws associated with them. Often times the schemes from the first category have slow login times, and the passwords in the second scheme are hard to remember. My project developed an algorithm, intended to generate a password that is secure and easy to remember. The algorithm requires the user to enter an n letter word. Two random letters are then inserted between every user-entered letter, with the first letter being a vowel. The password derives its security from the random letters and is easy to remember because of the user-entered letters. We analyze two properties of the algorithm: security and memorability. The first analysis produced a formula for determining n given a desired effective keyspace; n=log(keyspace)/log(number two letter combination). The second analysis was an experiment that consisted of giving a set of users a random password and another set a password created by my algorithm. After one day the users returned and attempted to login. The results of this study were inconclusive: 61% remembered the random password and 65% remembered the algorithm password.

1 Introduction

Computers have become an integral part of our lives. Their ability to interconnect has added a new dimension to how we can conduct business and communicate. As a result, obtaining secure communication and data storage is a growing concern. When users wish to perform secure transactions or access secure information on a remote computer, they are required to log into a system; the system in turn encrypts all the data that is transmitted until the user logs out. The current authentication schemes for producing the password, which allows a user to login, are inadequate, user-selected passwords are usually insecure, while machine-generated are difficult for the user to use or remember. This project designed an interactive system that combines the benefits of both user-entered and machine-generated passwords. The results of this study were inconclusive: 61% remembered the random password and 65% remembered the algorithm password.

1.1 Why Passwords are Needed

While the interconnectedness of computers allows us to communicate and send information anywhere in the world at the speed of light, it also gives anybody the ability to take this information without permission unless some security measures are put in place. A user can use two methods to protect their information. The first is to encrypt all of the data using some protocol such as the Advanced Encryption Standard (AES) or RSA. These measures can protect individual pieces of data but not an entire system. To protect an entire system, one must control who can gain access to the system. The second method provides this capability. A username and password login scheme enables a system administrator to grant or deny access to individuals. The username is usually some combination of the user's first and last name and is therefore well known. The password, on the other hand, is the source of security for the scheme and must be difficult to guess. Good passwords can effectively limit who can access the data in a system.

1.2 What Good Passwords are Made of

A perfect login scheme denies access to anybody who does not have an account, while never denying access to those with accounts. Since the password determines valid users from invalid ones and can be readily changed, it will determine how useful a login scheme is. Good passwords provide enough security for the system and are easy enough for the legitimate users to use.

1.2.1 What Makes a Password Secure

Since computers are physical objects and have a limited amount of memory, only a finite number of passwords are possible. This means that, given enough time, any password can be determined by simply trying every combination. Some passwords have a keyspace that is too large to search, for example the amount of time needed to decode a message protected by VeriSign's 128-bit SSL encryption scheme is more time than the universe has been in existence. Most schemes, however, don't have 128-bit passwords. Rather they are comprised of printable characters, which provide a keyspace significantly smaller than a 128-bit password, especially since many of the combinations are considered guessable. A guessable string is one that can be found in any dictionary, regardless of capitalization. A secure password is one that provides a non-guessable keyspace that is too large for attackers to compromise with a brute force search [1]. In addition to having a large keyspace, the creation scheme must ensure that if an attacker obtains some information about the password they cannot gain any more information about the password.

1.2.2 What Makes a Password Easy to Use

The password should deter only those without permission from using the system. If those with permission find the scheme burdensome or hard to use, it is unlikely that they will continue to use the system. Many password schemes require the user to remember pieces of information. Often these pieces of information are randomly selected to ensure the security of the system. However, such passwords are difficult to use because humans are not good at remembering random information [2]. Humans are better at remembering information that they either create or is somehow meaningful to them [2]. Whatever the password scheme the user should be able to login in a reasonable amount of time.

1.3 Types of Password Schemes

While most of the password generation and verification schemes provide adequate security, many of them are not easy to use. Login schemes can be broken into two categories: those that are interactive and those that are passive. While schemes from the first category require less effort for the user to recall the password, they require significantly more time to complete the login process. Some users may find this type of password inconvenient if they require immediate access to the server. Because the second category is passive it reacts almost instantaneously, but the passwords in this category require significantly more effort to recall from memory.

My scheme generates passwords belonging to the second category. It requires some user interaction to initially setup the password but afterwards the user will simply need to remember the password. All of the processing for the scheme is done on the server side and thus places no restrictions on the user's computer. The password is as secure as and could be easier to remember than a random password.

1.4 What Comes Next

The second chapter provides a summary of various password creation schemes currently available. The third chapter outlines the algorithm itself. It provides both an in depth walk through and pseudocode for the algorithm. Chapter four analyses the security of the algorithm by providing a formula to determine the keyspace and compares that keyspace with other algorithms. The following chapter focuses on the memorability of the algorithm. The chapter describes the application and experiment I developed to test how the passwords generated by my algorithm compare to a random password. The sixth and final chapter will describe my summarized results, conclusions, and further studies of my project. Finally the appendices will contain the source code and results for the test program.

2 Previous Work

The first login scheme was implemented in the Unix operating system. When a new user was added to the system an 8 character alphanumeric password was generated. The Unix password scheme has since been implemented for use in numerous programs. If implemented correctly this scheme guarantees a relatively secure password because no information about the user can be used to guess the password. Random passwords, on the other hand make the login scheme difficult to use. The user will usually change the password to something that is easier to remember but also easy to guess, thus limiting the security of the password.

The scheme presented by Tsutomu Matsumoto in 1996 requires the user to know the answer to several randomly chosen questions. The user knows the answers to the questions because the user knows how to interpret the questions while an attacker will not [3]. This means that the user doesn't need to remember any specific password because the host computer sends them the questions every time they login, the user just responds to the information presented to them. The problem with this scheme is the amount of time required to answer several questions may be too long for some people.

At the 6th ACM Conference on Computer and Communications Security, in November of 1999, Fabian Monrose, Michael K. Reiter and Susanne Wetzel presented a different type of password authentication, "Password Hardening Based on Keystroke Dynamics". This scheme uses the user's typing tendencies to generate and verify the password. The algorithm remembers the user's unique method of typing, how long certain keys are held down and how fast the password is typed are examples of the information that is gathered [4]. The algorithm also incorporates a learning mechanism that will adjust the parameters as the user alters their typing tendencies. The drawback to this scheme is the implementation of the algorithm and the requirement that the user's computer perform some operation on the password, this could lead to slower login times.

In August of 2000 Taekyoung Kwon published the "Authentication and Key Agreement via Memorable Password" protocol. The algorithm allows the user to believe that they are using a simple alphanumeric password when they are actually using a modified alphanumeric password. After the user enters their string the computer performs a mathematical operation on the string using an agreed upon random number and sends it to the host computer [5]. A series of these operations and transmissions are performed until it is assured that the user has permission to use the system [5]. This scheme works well because the password the user types in is not the password that is used for verification; rather it is only part of the password. Similar to Monrose's algorithm, the implementation of this scheme is challenging because it requires the user's computer to perform operations on the string rather than just presenting it, and could thus lead to a slower login time.

During the same month Rachna Dhamija and Adrian Perrig of the University of California at Berkeley developed a scheme that uses pictures instead of characters as the password. The results of their experiments were promising. Many of the initial users remembered their picture password with a higher accuracy and for longer then they remembered their standard alphanumeric password [6]. The scheme is not without its faults, however; the biggest complaint is the amount of time it takes to log into the system.

In May of 2001 RealUser, an online personal authenticator, developed a system of logging in that requires the user to remember a sequence of faces. The system is based on “the premise that the brain remembers images more easily than letters or numbers [7].” The system requires the user to select one of nine faces on five different screens. The response has been that most people remember this password far better then the standard eight alphanumeric one [7]. The drawback to this scheme like some of the others is the login time, downloading five pages of pictures can take some time and the user must be willing to wait, and entering the password is lots of work.

While each of the previously presented schemes provide similar security they differ in ease of use. The picture and face recognition and question and answer schemes prompt for a user reaction, while the remaining wait for the user to initiate the transmission. The first group requires more time but less effort on the part of the user while the second group requires a good memory but acts almost instantaneously.

3 Algorithm

Since I wanted the password creation scheme to be interactive, my algorithm had to obtain user input and use it in a meaningful manner. I knew that if I wanted the password to be secure there had to be some element of randomness. I determined through reading literature, that the user would have a better chance of remembering a password if the random letters were offset by the letters produced by the user. Thus by interspersing two random characters, the second one being a vowel, in-between the user entered letters I could gain enough entropy to be sufficiently secure and have the password be memorable. I chose the second letter to be a vowel to assist in the memorability of the password; this way the three-letter combination will most likely be something that sounds like English. Pseudocode for this algorithm is found in Figure 3.1, with a detailed description following.

```
user enters n letter word

password = “”

for each letter i in word
        password = password + i
        generate random vowel j (a,e,i,o,u,)
        password = password + j
        generate random letter k
        password = password + k

transmit password to user
```

Figure 3.1 – Pseudocode for Algorithm

The first step of the algorithm is to obtain a string of characters *n* letters long. The parameter *n* is determined by the size of the required key space, the formula for determining *n* is provided in section 4.1. The word can be obtained by a number of methods: handwritten and submitted to the administrator, via a secure web page, by word of mouth, or any method preferred by the administrator.

The second step is to create the password. The password initially starts as the empty string and is created as the algorithm appends new information. The algorithm begins a loop that is run *n* times, once for each letter in the word. Once inside this loop, the current letter of the user given word is appended to the end of the password. Two separate random numbers are generated. The first random number determines a vowel – a, e, i, o, u. The letter is then appended to the password. The second random number determines any one of the 26 letters of the alphabet. This letter is also appended to the password. The loop is then repeated until there are no more letters in the word.

The final step is to transmit the password to the user. This can be a difficult task to perform securely and is beyond the scope of this project, the process is left up to the administrator of the system.

4 Security Analysis

The next step was to analyze the security level of the algorithm. The security level is defined by the keyspace. For my algorithm the keyspace is a function of the size of the word entered by the user. Since I was trying to create a password that is easier to use then current password schemes I compared the number of objects a user must remember in my scheme with the number of objects a user must remember in other schemes with comparable security.

4.1 Determination of Number of Entered Letters

Using combinatorics the number of possible passwords is 130^n where n is the number of letters entered by the user. Because the user is providing n letters of the password those letters should be considered easily guessable and thus do not add to the security or keyspace of the password. The remaining $(3*n)-n$ letters are random but contribute in different amounts to the keyspace. The n second letters, or the vowels, contribute 5^n possibilities because there are 5 choices for each of the n letters. The n third letters contribute 26^n possibilities for a similar reason. Combined there are 130^n possible random combinations. This analysis assumes that the password is not case sensitive; this is preferable so as to limit what the user needs to remember, but does considerably reduce the keyspace. Therefore n=log(size of keyspace)/log(130).

4.2 Comparison With Other Schemes

Table 4.1 shows how many letters the user needs to enter to gain the same keyspace as other well-known password schemes. In each case the number of syllables –

combination of three letters starting with the user entered one – is less then the number of random or user entered characters required by the scheme. This means the user needs to remember fewer chunks of items using my scheme.

Password Scheme	Keyspace	Number of Characters	Number of Letters
UNIX	$2.08e^{11}$	8	5.35
UVA e-mail account	$2.18e^{14}$	8	6.78
ISIS	10^4	4	1.89
Hotmail	$>=3.76e^{9}$	>=5	>= 4.53

Table 4.1 – Keyspace Comparisons

5 Memorability of Algorithm

The final analysis of my algorithm was to test how well it produced memorable passwords. To perform this analysis I developed an application that attempts to test how well a human user can remember a given password. The application was developed and placed on the Internet for two weeks.

5.1 Test Procedure Development

There are a couple of considerations that I needed to take into account when I was developing my test procedure. I needed to ensure that I could obtain enough users to provide a balanced test population. Along with that idea I wanted to ensure that most of the users took the memorability test seriously and did not write down their password. To assist the user's desire to remember the password I considered using the application as a login for secure information, but ultimately rejected it because of a lack of secure information.

To develop a test procedure that would produce reasonable results I consulted with Professor Dan Willingham of the Psychology department at the University of Virginia. Based on his recommendations I decided to develop a web-based application. The application provides half of the users with a password comprised of random letters – henceforth known as random passwords – and the other half with a password created by my algorithm – henceforth known as interactive passwords. Since I was looking for a general trend the test population could be relatively small, on the order of 20 people for each type of password. To ensure the honestly of the users I made the URL known only to a group of friends and asked them to pass it on and explain my project. Additionally since my project performs an experiment with humans I needed to submit my project to the Institutional Review Board for the Social and Behavioral Sciences. Appendix A contains a copy of the approval letter for my project.

5.2 Implementation

The test procedure that I developed required the application to be on the Internet, and the nature of a login scheme requires the application to be dynamic. To satisfy these requirements Professor Evans recommended using the scripting language PHP: Hypertext Preprocessor (PHP). The interpretation is performed on the server so the user doesn't need any special software, additionally the user "receive[s] the results of running [the] script, with no way of determining what the underlying code may be [8]." This feature is useful for preventing would be attackers from gaining extra knowledge about a website, or how a password is generated.

Initially the application provided the user with a choice: to obtain a new username and password or to log into the system. For a user that chose to obtain a username and password the application ran one of two scripts: one that generates a random password and one that generates an interactive password. The application determines which script to run by querying a file that contains either a 0 or a 1. The received value represents what type of password the previous user received. Once the file is read the number is replaced with its logical opposite. This method ensures an equal distribution of random passwords and algorithm passwords.

The script that generates a random password requests the user to provide a username. The script then ensures the username is unique, generates a random password comprised of seven lowercase letters, displays the username and password to the user, and writes the combination in the password file. The script that generates an interactive password works in a similar manner except it requests a five-letter word, the inputs to the algorithm, in addition to the user to a username.

Since it is unreasonable to assume that anybody will remember any given word after seeing it only once, I decided to create a method for the user to practice entering the password. The user can use this method as many times as they like so long as they don't leave the site. Additionally, the script also provides a way for the user to retrieve their password if they believe they have no hope of remembering their password.

Once the user returns to the application to attempt to login a script is run that prompts the user to enter their username and password. After the user has submitted this combination the script queries the password file for the username and compares the passwords. The script displays the result to the user and records the result and user-entered password in the login file.

5.3 Execution and Results

Once the application was developed it was placed on the Internet through my account on the Department of Computer Science servers. I allowed users to sign-up for an account and attempt to login for a period of two weeks. When a new user obtained a password the application recorded the username, the password, and a timestamp in the password file. When a user attempted to login the application recorded the username, the result of the login, the password used, and a timestamp in a login file. Table 5.1 summarizes the results of the experiment. Of the 121 users who obtained a password, 75 attempted to login. The returning users successfully logged in using a random password 60.9% of the time, and 64.7% of the time using an interactive password.

Total	Did not login	Random Password			Interactive Password		
		Succeeded	Failed	Percentage Correct	Succeeded	Failed	Percentage Correct
121	46	25	16	60.9	22	12	64.7

Table 5.1 – Summary of Results

6 Conclusions

The algorithm that I developed was evaluated based on its ability to generate passwords with a large keyspace and ease of memorability. The security analysis provided a formula to determine the appropriate number of user-entered letters based on the desired keyspace, $n = \log(\text{keyspace})/\log(130)$. The memorability analysis provided the statistics for how well people remember a random or algorithm password, 61% and 65% respectively.

6.1 Interpretation

While the formula developed in the security analysis for my algorithm is used to determine the keyspace it can also be used to compare the number of objects a user must remember. Based on this formula the algorithm compares favorably with other password schemes in terms of number of objects to remember. The formula shows that any password comprised of objects that can be a number of values less then 130 will always require the user to remember more objects than my algorithm. Given that $n(\log(130)) = y(\log(x))$, where n is the number of user-entered letters in my algorithm, 130 is the number of syllables, see section 4.1 for derivation, x is the number of possibilities of an object in any algorithm, and y is the number of objects in the algorithm – and $\log(130) > \log(x)$ when $130 > x$, then n must be less than y. In most cases the number of possibilities for an object will be significantly less than 130. Since the total number of objects often determines how hard a password is to remember and my algorithm can achieve the same security with fewer objects, my algorithm has an advantage over others.

The results of the memorability analysis were inconclusive. The percentages were too close to conclude that one scheme is better than the other. This is not all bad because one cannot claim that my algorithm is any worse than a random password, it still has the possibility of proving to be better given different conditions. I choose a low number of objects for the user to remember, the random password was only seven lowercase letters long. This type of password would almost never be used in practice, since the keyspace is so small. The small number of letters coupled with the fact that they were all lowercase greatly assisted the user in their ability to remember the password. In addition, I gave the user an unlimited amount of test logins. These test logins helped the user commit the password to memory but they also helped train muscle reflexes. In practice the user does not get these trial runs. Thus, while the analysis was inconclusive it could be altered and performed again and more conclusive results could be obtained.

6.2 Recommendations

To improve this experiment I would suggest altering both the random password and the algorithm password. I believe I underestimated the ability for humans to remember random information. A better test would have been to create an eight-letter password comprised of upper and lowercase letters and numbers. I choose not to use this password because I feared it would be too hard for people to remember; I no longer believe this.

An interesting change to the algorithm would be to modify it so that it guarantees that there are never two consecutive vowels. Though this adjustment may create passwords that are easier to remember, it will also change the keyspace of the algorithm.

Meaning both analysis would have to be performed. Further testing may reveal that this algorithm produces better passwords. For future tests I recommend limiting the number of trials a user can perform after they receive their password. While my test had a respectable size of 75 users a larger test population would provide stronger results. A test that is run using a population of 200 or more, has the login protect something of value to the user, and incorporates some of the above changes would be an ideal test procedure.

6.3 Project Evaluation

I would characterize this project as a modest success. I have created an algorithm that successfully produces a password that can be as secure as a random password. Additionally it was proven that my algorithm produces a password that requires less objects then most other passwords of comparable security, a big plus for memorability. I failed to demonstrate that my scheme produces passwords that are easier to remember then random passwords, however I also failed to demonstrate that it produces harder passwords. The algorithm passed one analysis and proved inconclusive on the other, further testing is required before a final verdict can be rendered for the success of this algorithm.

References

1. Spafford, Eugene H. OPUS: Preventing Weak Password Choices. West Lafayette: Purdue, 1991.

2. Schwartz, Barry, and Daniel Reisberg. Learning and Memory. New York: W W Norton and Company, 1991.

3. Matsumoto, Tsutomu. Human-Computer Cryptography: An Attempt. New Delhi: CCS, 1996.

4. Monrose, Fabian, Michael K. Reiter and Susanne Wetzel. Password hardening based on keystroke dynamics. Proceedings of the 6th ACM conference on Computer and communications security 1-4 Nov. 1999, Singapore. Conference on Computer and Communications Security. <http://www.acm.org/pubs/citations/proceedings/commsec/319709/p73-monrose/>

5. Kwon, Taekyoung. "Authentication and Key Agreement via Memorable Password." Online posting. 23 Aug. 2000. 9 Sept. 2001 <http://eprint.iacr.org/2000/026/>.

6. Dhamija, Rachna, Adrian Perrig. "Déjà Vu: A User Study Using Images for Authentication." Online posting. 1 Sept. 2000. 28 Jan. 2002 <http://paris.cs.berkeley.edu/~perrig/projects/usenix2000/>.

7. Salkever, Alex. "Picture This: A Password You Never Forget." Business Week Online. Online posting. 15 May 2001. 28 Jan. 2002 <http://www.businessweek.com/bwdaily/dnflash/may2001/nf20010515_060.htm>.

8. Bakken, Stig Sæther, Alexander Aulbach, Egon Schmid, Jim Winstead, Lars Torben Wilson, Rasmus Lerdorf, Andrei Zmievski, Jouni Ahto. "What is PHP?" PHP Manual. Online Posting. 10 March 2002. 14 March 2002 <http://www.php.net/manual/en/introduction.php>

Bibliography

Anderson, John R. Learning and Memory: An Integrated Approach. 2nd ed. New York: John Wiley & Sons, 2000.

Bakken, Stig Sæther, Alexander Aulbach, Egon Schmid, Jim Winstead, Lars Torben Wilson, Rasmus Lerdorf, Andrei Zmievski, Jouni Ahto. "What is PHP?" PHP Manual. Online Posting. 10 March 2002. 14 March 2002 <http://www.php.net/manual/en/introduction.php>.

Bergadano, Francesco, Bruno Crispo and Giancarlo Ruffo. "High dictionary compression for proactive password checking." Transactions on Information and System Security 1 (1998): 3-25.

Bolande, H. Asher. "Forget passwords, what about pictures?" Online posting. 27 Nov. 2000. 9 Sept. 2001 <http://www.zdnet.com/zdnn/stories/news/0,4586,2657540,00.html>.

Cofer, Charles N. "An Historical Perspective." The Structure of Human Memory. Ed. Charles N. Cofer. San Francisco: W. H. Freeman and Company, 1976.

Dhamija, Rachna, Adrian Perrig. "Déjà Vu: A User Study Using Images for Authentication." Online posting. 1 Sept. 2000. 28 Jan. 2002 <http://paris.cs.berkeley.edu/~perrig/projects/usenix2000/>.

Hall, John F. Learning and Memroy. 2nd ed. Boston: Allyn and Bacon, 1989.

Kwon, Taekyoung. "Authentication and Key Agreement via Memorable Password." Online posting. 23 Aug. 2000. 9 Sept. 2001 <http://eprint.iacr.org/2000/026/>.

Leahey, Thomas Hardy, and Richard Jackson Harris. Learning and Cognition. 5th ed. New Jersey: Prentice Hall, 2001

Matsumoto, Tsutomu. Human-Computer Cryptography: An Attempt. New Delhi: CCS, 1996.

Monrose, Fabian, Michael K. Reiter and Susanne Wetzel. Password hardening based on keystroke dynamics. Proceedings of the 6th ACM conference on Computer and communications security 1-4 Nov. 1999, Singapore. Conference on Computer and Communications Security. <http://www.acm.org/pubs/citations/proceedings/commsec/319709/p73-monrose/>

Murray, D. J. “Research on Human Memory in the Nineteenth Century.” Human Memory Contemporary Readings. Ed. John G. Seamon. NewYork: Oxford University Press, 1980.

Richardson, John T. E. “Evolving Concepts of Working Memory.” Working Memory and Human Cognition. Ed. Marc Marschark. New York: Oxford University Press, 1996.

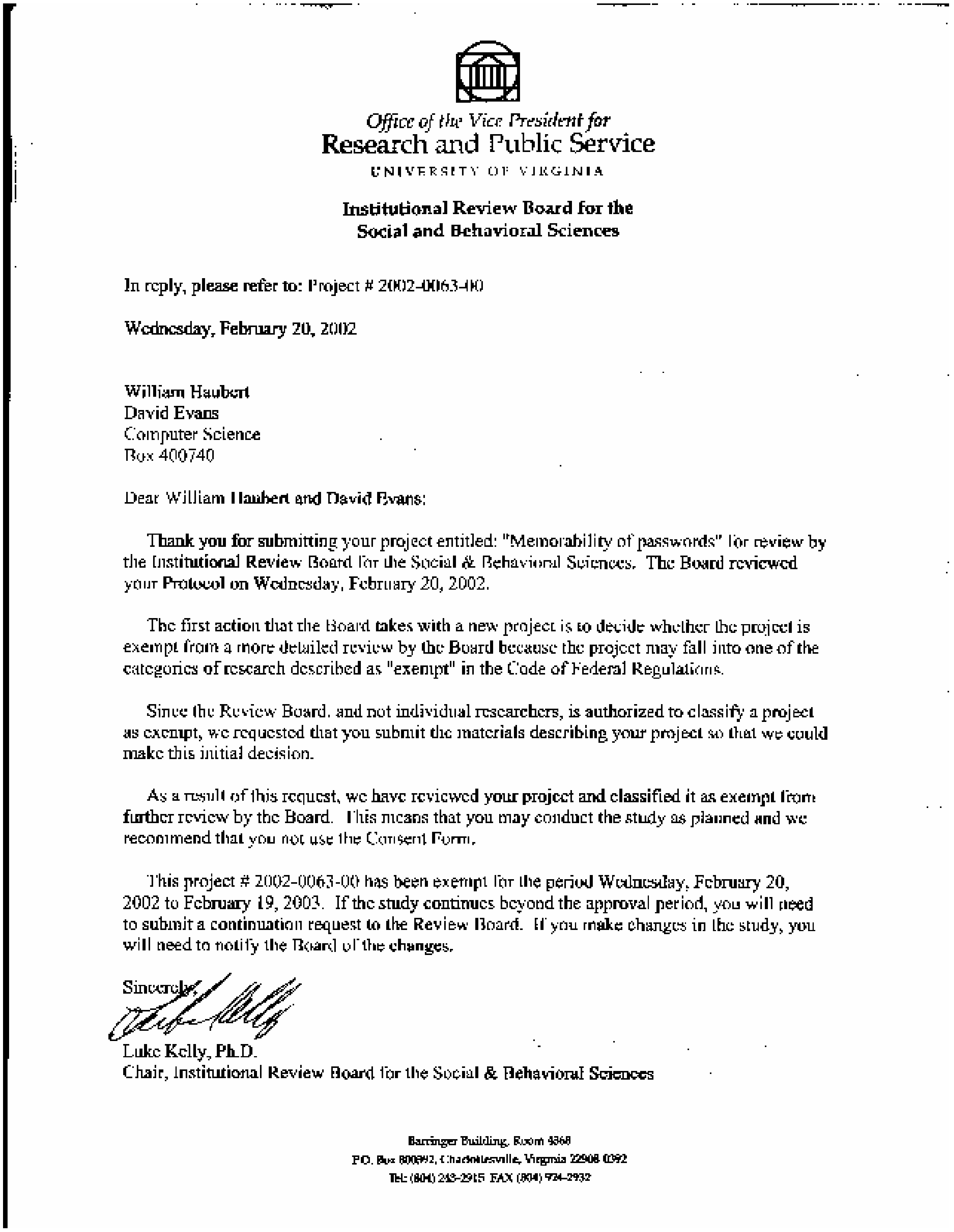
Salkever, Alex. “Picture This: A Password You Never Forget.” Business Week Online. Online posting. 15 May 2001. 28 Jan. 2002 <http://www.businessweek.com/bwdaily/dnflash/may2001/nf20010515_060.htm>.

Schwartz, Barry, and Daniel Reisberg. Learning and Memory. New York: W W Norton and Company, 1991.

Spafford, Eugene H. OPUS: Preventing Weak Password Choices. West Lafayette: Purdue, 1991.

Stallings, William. Cryptography and Network Security: Principles and Practice. 2nd ed. New Jersey: Prentice Hall, 1999.

Appendix A – Approval Letter



Office of the Vice President for
Research and Public Service
UNIVERSITY OF VIRGINIA

Institutional Review Board for the Social and Behavioral Sciences

In reply, please refer to: Project # 2002-0063-00

Wednesday, February 20, 2002

William Haubert
David Evans
Computer Science
Box 400740

Dear William Haubert and David Evans:

Thank you for submitting your project entitled: "Memorability of passwords" for review by the Institutional Review Board for the Social & Behavioral Sciences. The Board reviewed your Protocol on Wednesday, February 20, 2002.

The first action that the Board takes with a new project is to decide whether the project is exempt from a more detailed review by the Board because the project may fall into one of the categories of research described as "exempt" in the Code of Federal Regulations.

Since the Review Board, and not individual researchers, is authorized to classify a project as exempt, we requested that you submit the materials describing your project so that we could make this initial decision.

As a result of this request, we have reviewed your project and classified it as exempt from further review by the Board. This means that you may conduct the study as planned and we recommend that you not use the Consent Form.

This project # 2002-0063-00 has been exempt for the period Wednesday, February 20, 2002 to February 19, 2003. If the study continues beyond the approval period, you will need to submit a continuation request to the Review Board. If you make changes in the study, you will need to notify the Board of the changes.

Sincerely,

[illegible]

Luke Kelly, Ph.D.
Chair, Institutional Review Board for the Social & Behavioral Sciences

Barringer Building, Room 4368
P.O. Box 800392, Charlottesville, Virginia 22908 0392
Tel: (804) 243-2915 FAX (804) 924-2932