Audio Entropy

Joshua Dow

Abstract— **This project aims to calculate the security and the computability of voice-recognition as a verification method. These values can then be compared to information regarding the safety and usability of text passwords in practice. This will allow us to make a conclusion about the utility of voice-recognition, and whether or not it is a suitable alternative to text-based passwords.**

I. INTRODUCTION

Using biometric data as an alternative to conventional text passwords has been an attractive option since the technology to work with them has been available. But how complex and secure are these passwords, really? Biometric data encompasses everything from fingerprints, handprints, signatures, iris scanning, voice recognition, and even DNA[1]. However, for everyday use, these have yet to replace text passwords as the de facto authentication method, even though the technology to use biometric data is now fairly widespread. We wish to determine which method is actually more secure and accessible: a conventional text password, or biometric verification[1].

II. PROBLEM

A. Actual Security of Voice Recognition

Vocal recognition software has advanced quickly in recent times, but due to the complex algorithms by which the data is recorded, the security of this technique is not easily measured[2]. As with any other form of security, we need to ensure that vocal passwords are not easily duplicated and broken into. Does the security level, calculated as entropy, of the average vocal passcode exceed or fall short of the security of the average text password, and as well, what are the advantages to these kinds of passcodes? To what level do vocal passcodes provide security against outlier situations such as someone saying the passcode with a similar voice? We would like to determine exactly how secure the average one-word vocal password is in terms of bits and calculate the average entropy of such a password.

B. Efficiency vs Security

One of the largest issues involved with voice recognition is how long it can take to analyze the incoming sample, transfer the appropriate data to the database, and confirm it is the correct passcode[1]. Depending on the algorithms used, this process is often prohibitively slow to ensure a proper standard of security. This longer processing time allows more data in the vocal recording to be analyzed and a far higher level of comparison to occur between the recording and the accepted passcode, greatly increasing the security[2]. However, for vocal passwords to be usable in everyday life and hopefully become as ubiquitous as text-passwords, they must be fast enough to not cause undue frustration on the user and unnecessary strain on the database. What is the best balance of speed versus security that a vocal passcode can have, using the technology and algorithms we can utilize today?

III. PROPOSED WORK

Our goal is to find the security, determined by calculating the entropy, of a standard one-word vocal password, using speech recognition software. This will be done several times using several different methods of recording vocal passcodes. Following this, we want to compare which methods can improve the security, and also which methods can increase the usability of the system. Finally, we want to find which methods balance both maximum efficiency and security. The end goal would be to compare the security of this preferred method of analyzing vocal passcodes to a typical text-character password, to see if it is more or less secure. We chose to use a one-word passcode as we have a limited time-frame and comparing one-word passcodes will be sufficient for the scope of our project.

A. Security

Our method will be to take a recording of a voice passcode and convert this into a binary format several times with different methods[2]. We can then turn these binary codes into measurable units of security, as entropy. There are many factors that go into calculating the security of a particular vocal sample. Metrics like inflection, pitch, and tone are important ones to take into consideration[2]. We will measure these and determine how changing these factors will affect the security of the passcode. We will also find the entropy of an average text-based password of a similar length and compare them. This will allow us to make a conclusion regarding whether a text-password is a more secure method for authentication.

B. Efficiency

The work involved in processing and verifying a vocal passcode is significantly more than that of a text password[1]. Different methods for processing and verifying differ in that some may be too strict in their authentication, while others may be too forgiving. We would like to find the optimal point, where the system is lenient enough to permit the user with the right password through, but not demanding to the point where the user must speak at the correct speed and inflection every time. As well, we would like to find a method that is efficiently calculated without drastic sacrifices required from the level of security[1]. Ideally, our method will be fast enough that if the correct word is spoken at a reasonable speed, the verification will take no longer than 2 seconds, and we will aim to have an accuracy rate of 90% or higher.

C. Complexity

Given that vocal recognition is substantially more difficult to analyze and compare then text parsing, we have chosen to limit the scope of our project to a one-word passcode, between one to two syllables[1]. This will sufficiently allow us to properly compare the average complexity and entropy of these passwords.

IV. Experiments and Results

A. Objectives

We wish to show whether a vocal password is more or less optimal than a typical plain text password in terms of entropy. First, we want to show what the entropy of a one-word vocal password is, and how its entropy changes given alterations in inflection, pitch, and tone. We find the entropy of this vocal password by applying an encoding schema known as ”Arithmetic Coding” which is commonly used in data compression[3]. We want to determine what factors are needed to have an optimally secure vocal password such that it is reliable, but not easily compromised. In addition, finding a verification algorithm which is efficient[3].

B. Technology

We will use Python 3 to perform our voice recognition. We will utilize the SpeechRecognition 3.8.1 library, which is a library for Python that allows easy access to various voice recognition API's, such as Google's and Bing's[4]. We will largely be using Google's Voice Recognition and Bing's Speech recognition APIs, as both of those have clear documentation and provide a level of detail not supplied by the other APIs[4]. These APIs allow us to control for things such as length of the recording, accent of the speaker, as well as the ambient noise in the background. They also allow us to not only see the interpreted statement, but a list of statements that may also have been said, as no voice recognition model is perfect under all scenarios[4].

On top of that, we will be using audio processing software such as Audacity in order to look at the micro attributes of the recording, such as inflection, pitch, and tone of the speaker[5]. Our process for determining the entropy of a vocal password, encoding, and decoding[3]:

1) Read in an audio format file into a byte array.
2) Convert this byte array into an array which contains unsigned integers which are assigned to unique bytes from the original array.
3) Convert the unsigned integer array into ASCII characters
4) Apply Arithmetic Coding as follows[3]:
    a) We first determine l the length of the string, and from here on we will work in base l
    b) We find the frequency of each letter in the string.
    c) Next we calculate the lower bound of number of bits required to represent the message. $L = \sum_{i=1}^{n} n^{n-i} j_i \prod_{a=1}^{i-1} f_a$ where j_i is the cumulative frequency, f_a is the number of occurrences.
    d) Now the upper bound. $U = L + \prod_{a=1}^{n} f_a$
    e) Lastly is the choice of which value in $[L, U)$ to choose. The best case is the number with the greatest number of trailing zeroes. This allows the zeroes to be truncated later.
    f) Note that the final result will have the same number of digits as the highest power of base l that you are working in. So, to decode the integer and get back to the original message we simply take our choice c and divide it by l^p to get r where p is the length of our choice. Then we update the next number to divide, which is calculated by: $\frac{(c-l^p*r)}{r}$ and we repeat this last part until our message is recovered.
5) Our entropy is calculated conservatively by the lower bound L from above.

C. Experiments

Our experiments are designed to test individual and composite changes to vocal passwords to see the effect on Entropy. The individual experiments are listed below, but our composite experiments include permutations of those listed and will be outlined in our final presentation [2].

1) Entropy as a function of vocal password length.
2) Entropy as a function of vocal pitch.
3) Entropy as a function of vocal inflections.
4) Entropy as a function of tone.
5) Entropy as a function of speech velocity.

V. Planning

The project proposal will be submitted on October 7^{th}. We will start working on the experiments after we get feedback on October 16^{th}. Comparing different copies of the vocal password and the changes in recordings by altering inflection, pitch and tone will be done by Eric. Ga Hyung will work on finding entropy of the vocal passwords by applying an encoding schema. Catherine will research factors to improve the security of vocal password and different known techniques of recording and storing vocal data. Joshua will be working on a verification algorithm. By November 18^{th}, we will have completed the objectives of our project and we will come up with questions about the main points of our experiment. A project presentation will be prepared to explain to our class the key points of our project's experiments. The final report about our experiments and results will be done before the project presentation to support our presentation and will be submitted on December 7^{th}.

References

[1] O'Gorman, L. ”Comparing Passwords, Tokens, And Biometrics For User Authentication.” Proceedings of the IEEE 91.12 (2003): 2019-2020. Web.

[2] Bhattacharyya, Debnath et al. ”Biometric Authentication: A Review.” International Journal of u- and e- Service, Science and Technology 2.3 (2009): 13-28. Print.

[3] Said, Amir. ”Introduction To Arithmetic Coding - Theory And Practice.” Lossless Compression Handbook (2004): n. pag. Print.

[4] ”Speechrecognition.” PyPI. N.p., 2018. Web. 6 Oct. 2018.

[5] ”Audacity — Free, Open Source, Cross-Platform Audio Software For Multi-Track Recording And Editing..” Audacityteam.org. N.p., 2018. Web. 6 Oct. 2018.