Improving Internet Security for the Elderly with Advanced Password Hashing Algorithms

Presented to:

Ms. Silvia Tropea

Science Department

St. Augustine Catholic High School

SCH3UE

Prepared By

Kaden Seto & Christopher Lee

St. Augustine Catholic High School

Grade 11 STREAM Students

May 31, 2023

Table of Contents

Introduction

Many elderly people or people that were born in an earlier generation have not been exposed to technology and the internet in the early stages of their lives. They have limited knowledge the significance of internet security in the current day. For example, research has shown that seniors hold many characteristics that make them vulnerable to being hacked online (*Internet Safety for Seniors | Washington State*, n.d.). These characteristics include their lack of knowledge when it comes to the internet and computer skills. Many seniors, for example, are primary targets for scams. Seniors tend to be more worried and concerned about technology-related issues. As a result, when scams are sent out to seniors that state that there is some type of issue, seniors are likely to fall for it since they are more trusting and have less knowledge about internet safety. Many hackers and scammers use password-extraction software in the malware that they send out to seniors. The sense of internet security that seniors have is also weaker as they do not have much awareness of the significance of internet security. Research has also shown that many computers that seniors use are not properly secured and equipped with security software (*Internet Safety for Seniors | Washington State*, n.d.). All of these factors make seniors susceptible to being hacked.

Hashing is the process of using a hashing algorithm to convert regular names or series of characters into a long series of unintelligible numbers and characters (Jung, 2021). By hashing a password, it makes it significantly harder for hackers to gain unauthorized access to systems. In this project, a password hashing algorithm was developed for seniors to use so that their passwords cannot be hacked through various methods of hacking attacks.

The hashing algorithm is primarily designed to prevent two common methods of password attacks. In a brute force attack, the hacker attempts to gain access to a system by guessing and trying to find all possible combinations of what the password can be (Vosyliūtė, 2023). However, due to the hashing algorithm, the hashed password cannot be guessed as it is impossible to guess a long series of letters and numbers that are randomized. A rainbow-table attack is a stronger form of a brute force attack, where the hackers lay out a table of common passwords and their hashed versions (Walker, 2023). However, due to the password randomization algorithm that assigns a random “key” value to the password, the password cannot be guessed with a rainbow-table attack. This allows seniors to still keep a simple password stored in the system, but because of the hashed key feature, the application essentially turns their password into a hashed version that cannot be guessed. This report will describe the procedures used throughout the project to design an effective password encryption application to ensure better internet security for seniors.

Procedure and Planning

To execute this idea, we first needed to decide how we would encode and save the passwords on the computer without the risk of people stealing the passwords from files. Our group decided that the best way to do this would be to use a hashing algorithm to encode keys that we can use to store the passwords without actually saving the passwords to files. Our application also used .dat file extensions that cannot be normally opened legibly by most text editors.

Methods, Objects, and Algorithms

In our code, passwords are saved as an object type Password. Passwords contain a name, key, and time created inside of it. They can also return the hashed version of the key, which is the actual password. Many instances of these objects are appended to a list of Passwords that are loaded into the list view on the left side of the application. Every time a new Password is created, it will be added to the list, and saved to its respective DAT file. The name of the password is also saved to a TXT file that contains the passwords that currently exist in the list.

In order to export and import data, we had to develop methods that would be able to successfully perform this task. For exporting the data, we first had to load the text file into our program, create a new list that contains the passwords, and then write the name attribute of each Password object to the text file. After, we would then serialize and save this object to a DAT file.

For importing data, we had to go through each name in the TXT file that contains the names of the passwords, find the DAT file that corresponds to the name, and then load the file using an ObjectInputStream reader. We would then save the loaded Password objects to the list of passwords, and load that list into the list view in our application.

Hashing Algorithm

The algorithm that we used to hash the passwords was SHA-256, a secure hashing algorithm that can encrypt a “key” to a secure password. The way that the SHA-256 algorithm works is that it takes a key and performs mathematical operations onto it, returning an encrypted string of text that is extremely secure. Instead of saving the hashed passwords into files, our application instead uses the key to save it to a file

Saving the Passwords to File

Saving the passwords to a file was something that our application needed to do in order to reload passwords when the app was reopened. We decided to use .dat files to store the data and reload it into our application. DAT files, or .dat files are data files that can store objects, other files, and values.

Password Checker Algorithm

The password checker algorithm is a basic algorithm created to return a String (word/sentence) value that will tell the user the rating of their password in sentences. It follows four pieces of criteria:

- The password must be 8 or more characters long in length.
- The password must include at least 1 punctuation character.
- The password must contain at least 1 numeric character.
- The password must have at least one uppercase letter.

Based on the criteria, the algorithm will give the user their password rating in words, and recommendations on how they can improve their password.

Procedure

After planning out how we would program our design, we started out by creating draft designs that would fit what we wanted to accomplish. This included using a software called Scenebuilder to create a draft design that we could code. We coded the design, then started programming the file input and output system that would work with the filesystem. After, we programmed the functionality of the buttons, and algorithms that would coincide with that. We had to figure out how we would implement these features in the user interface, and decided to redesign the UI. After, we attempted to solve visual issues within our code. Finally, we tested and fixed any existing logic and algorithmic bugs that existed in our code.

This procedure took around a month to complete, and involved many redesigns of the user interface, file system, algorithm, and code. Overall, we accomplished for the most part what we set out to do, and created an application that fit the requirements that would help the elderly with their problem.

Password Entropy

Password Entropy is the measure of the strength or randomness of a password. It quantifies how likely a password can be guessed in a password attack. A higher entropy value for a given password would mean that there is a greater number of possible combinations of what the password could be (Datta, 2022). The entropy of the hashed passwords created using the hashing algorithms in this project can be calculated to show the strength of the hashed passwords created.

Entropy is measured in bits since a computer can only store binary (0 and 1), base-2 digits. Therefore, a password with an entropy value of 30 would mean that there are 2^{30} = 1 073 741 824 possible combinations of what the password can be. For every additional bit that gets added to the password's entropy, the number of possible combinations would double.

The formula for Password Entropy can be calculated by:

$$E = \log_2(R^L)$$

$$E = L[\log_2(R)]$$

Where:

- E is the password entropy value, measured in bits.
- L is the length of the password.
- R is the randomness or range of characters in the password. The randomness depends on the types of characters used in the password.

- According to Baeldung, the entropy value can be categorized using the chart below: (Datta, 2022)

Entropy Value	**Category**
0-24	Very Weak Password
25-49	Weak Password
50-74	Ok/Reasonable Password
75-100	Strong Password
> 100	Very Strong Password

Fig. 1 Categorized strength of passwords based on their entropy values.

Taking the password, “password123” as an example:

- The length (L) of the password is 11.
- The password can be possibly made up of 26 uppercase letters, 26 lowercase letters, and 10 possible digits. Therefore the randomness (R) of the password is 26+26+10 = 62.
- Therefore: $E = \log_2(62^{11}) = 65.5$, and the number of combinations is equal to $2^{\log_2(62^{11})} = 62^{11} = 8.3929 \times 10^{17}$ combinations.

Using the created chart above, it can be concluded that the entropy value of “password123” shows that the password is a relatively reasonable/good password. However, it is important to note that password entropy is not the best measure of password strength as the given

password, such as “password123”, can be easily determined. The method of password entropy only works in this project as a hashed series of passwords are unintelligible.

Results and Observations

Password Details	**Length (L)**	**Randomness (R)**	**Entropy (E)**	**Results (Entropy Rating & Number of Combinations)**
Name: password Key: L2L7 Hashed Key: c464d9e9d3c6d434e4ac95c04a2eb22e531bd32487a2d79cff52b073ccd3539e	64	10 + 6 = 16	$E = \log_2(16^{64})$ $= 256$	High Password Entropy (256) - Very Strong Password Number of Possible Combinations: $16^{64} = 1.158 \times 10^{77}$ combinations
Name: testing Key: T4fR Hashed Key: 90f1cafe1f83490b522c6ee0c6eed5427bcb3d94669cca60720a02fa86f096d6	64	10 + 6 = 16	$E = \log_2(16^{64})$ $= 256$	High Password Entropy (256) - Very Strong Password Number of Possible Combinations: $16^{64} = 1.158 \times 10^{77}$ combinations
Name: Google Accounts Key: bW-0 Hashed Key: 6048e2f0105b57ae2f8cbf0955ef13674fd6e53a7b017fdba136439c8cf89dcd	64	10 + 6 = 16	$E = \log_2(16^{64})$ $= 256$	High Password Entropy (256) - Very Strong Password Number of Possible Combinations: $16^{64} = 1.158 \times 10^{77}$ combinations

Fig. 2: Password entropy table data featuring hashed passwords created using our secure hashing algorithm.

Conclusion

The results of using the password entropy formula to calculate the strength of the hashed passwords show that our password algorithms are able to consistently create strong passwords with a very high password entropy value of 256. As mentioned before, the hashes cannot be guessed using rainbow table attacks because of the assigned key to each password, as the hash is uncommon. Brute-force attacks also cannot be used as it is impossible to guess and iterate through the 1.158×10^{77} combinations of the possible hashed password. As a result, the hashing algorithms will be able to very adequately defend against any type of password-cracking techniques used today, whether it is brute-force attacks, rainbow-table attacks, password software hacking, etc.

The algorithms used in this application will play a large role in ensuring internet security for all seniors. Seniors will ultimately be unable to keep themselves completely safe on the internet as they still have lacking knowledge of their computer and internet skills. They will ultimately still be prone to safety breaches on the internet as the prime targets of scams and different types of attacks. However, Leeto Security's password application and hashing algorithm are able to significantly reduce the chances of security breaches on the internet with its ability to consistently create secure and very strong passwords, which is shown in the results of the password entropy test above. It properly secures devices used by seniors, which is one of the major factors that cause security breaches.

Even those who may not be seniors can use this application to secure their passwords and have strong internet security. The hashing algorithm can benefit anyone of any age group. It may also be useful for those who have a weaker sense of internet security, so the password application will prove useful in creating secure passwords for them. In conclusion, the password hashing algorithm can be beneficial for anyone, especially seniors, and is a major step in being highly secure and safe in the dangerous world of the Internet.

Acknowledgments

We would like to thank Ms. Tropea for her guidance over the course of this project. This project was an amazing opportunity to learn more about the applications of computer science and has widened our knowledge of this field.

References

Works Cited

Characters that are valid for user IDs and passwords. (n.d.). IBM. Retrieved May 28, 2023, from https://www.ibm.com/docs/en/baw/20.x?topic=security-characters-that-are-valid-user-ids-passwords

Code, B. (2020, Aug 13). *Java buttons*. Youtube. Retrieved May 28, 2023, from https://www.youtube.com/watch?v=-IMys4PCkIA&ab_channel=BroCode

Code, B. (2021, March 9). *JavaFX ListView*. Youtube. Retrieved May 28, 2023, from https://www.youtube.com/watch?v=Pqfd4hoi5cc&ab_channel=BroCode

Datta, S. (2022, November 4). *How to Determine the Entropy of a Password?* Baeldung. Retrieved May 28, 2023, from https://www.baeldung.com/cs/password-entropy

Hashing Algorithm in Java. (n.d.). Javatpoint. Retrieved May 28, 2023, from https://www.javatpoint.com/hashing-algorithm-in-java

Howarth, J. (2023, February 6). *50+ Password Statistics: The State of Password Security in 2023*. Exploding Topics. Retrieved May 28, 2023, from https://explodingtopics.com/blog/password-stats

Internet Safety for Seniors | Washington State. (n.d.). Washington State | Office of the Attorney General. Retrieved May 28, 2023, from https://www.atg.wa.gov/internet-safety-seniors

Jung, J. (2021, May 7). *What is Password Hashing and Salting?* Okta. Retrieved May 28, 2023, from

https://www.okta.com/uk/blog/2019/03/what-are-salted-passwords-and-password-hashing/

Vosyliūtė, M. (2023, March 16). *Most common password cracking techniques hackers use*. Cybernews. Retrieved May 28, 2023, from https://cybernews.com/best-password-managers/password-cracking-techniques/

Walker, D. (2023, May 4). *how do hackers get your passwords?* ITPro. Retrieved May 28, 2023, from https://www.itpro.com/security/34616/the-top-password-cracking-techniques-used-by-hackers

What is Cybersecurity? (n.d.). IBM. Retrieved May 28, 2023, from https://www.ibm.com/topics/cybersecurity