

Determining the Efficacy of User-Generated Passwords as Compared to Randomly Generated

Passwords

Eshita Kar

April 30, 2019

Determining the Efficacy of User-Generated Passwords as Compared to Randomly Generated Passwords

Abstract

The purpose of this study was to determine the efficacy of user-generated passwords as compared to randomly generated passwords. Although prior research has shown that users tend to use short, alphabetic passwords that are repeated across multiple interfaces, there has not been much research conducted into the comparison between randomly generated passwords and user-generated passwords, especially amongst a younger population (those 18 and below).

Data from users was collected through a false registration software without any password prompts to ensure that users were entering realistic passwords and to ensure that prompts for password requirements would not create response bias. Data from randomly-generated passwords was generated via a self-programmed random password generator. Although each user's password was collected and stored in a database, they were never visible to me as an investigator to ensure the privacy of their information. Data that was collected includes the length of each password collected/generated, the number of characters from each character set, or charset, used, the Shannon entropy of each password, and the time taken to crack the MD5 encrypted version of the password on John the Ripper, a password cracker. The charsets were split into alphabetic characters (a total of 52 possible characters), numeric characters (a total of 10 possible characters), and special characters (a total of 33 possible characters). When cracking passwords, if more than 1 hour was spent cracking the password, the password was recorded as "uncrackable."

Analyses were conducted to compare the bit strength and cracking times of user-generated passwords and randomly generated passwords. Initial results showed that the average bit strength of user-generated passwords is 54.675 while the same for randomly generated passwords is 61.254. The cracking probability for user-generated passwords was 14.71% while the cracking probability for randomly generated passwords was 6.67%. After conducting analyses, it was found that there is no significant difference between the bit strengths of user-generated and randomly generated passwords. However, the significant difference in cracking probability indicates that in real-world situations, user-generated passwords are easier to crack than randomly generated passwords.

Future research should look into other metrics of bit strength to see if Shannon entropy is truly effective in measuring the strength of user-generated passwords. It can also look into the use of password dictionaries or rainbow tables when cracking user-generated and randomly generated passwords to see if cracking times and probabilities increase or decrease for both. However, this research is important as it indicates that perhaps instead of completely automating password generation, cybersecurity professionals should begin to look into alternative password creation methods that incorporate both the user and randomness, such as using user input and random password generation to create hybrid passwords or using a password wallet/single sign-on.

Determining the Efficacy of User-Generated Passwords as Compared to Randomly Generated Passwords

As technology is integrated into our lives, cybersecurity is becoming more important to our personal safety. With a plethora of personal information stored on databases around the world, it is becoming increasingly common for malicious individuals to steal data and use it for personal benefit. In fact, there were over 53,000 cybersecurity incidents in 2018 alone (Verizon, 2018, p. 4). To combat this, cybersecurity professionals have created software to protect users. However, many forget to think about the user's side of cybersecurity.

While it is important for users to have proper software to protect themselves from online threats, their actions online are just as crucial in protecting their data as the programmed actions of automated security programs. Perhaps the most crucial of these actions is password creation. Many often forget that creating strong passwords is the first line of defense against malicious attackers, and end up creating weak, easily-remembered passwords. This creation of weak passwords is an experience that I personally went through.

As I began the process of college applications, I was surprised by the number of accounts I needed to create and began to wonder if creating several passwords for these different applications was worthwhile. Thus, I succumbed to what many other technology users succumb to: password laziness. I began using the same password across multiple platforms or passwords that were weak (with little variety in character and number choice). Upon receiving a notification that one of my passwords was compromised, I began thinking of how other users must face this consequence of password laziness.

Combining this experience with my knowledge of cybersecurity, I decided to investigate if user-specified security, specifically user-generated passwords, are stronger and more reliable than the randomly generated passwords automated systems create. Password creation is a good measure of user-end cybersecurity, as it is the most direct action a user takes to protect his or her system. Using the strength of passwords, referred to as bit strength, as a quantifiable measure of security, user-created passwords can be compared to randomly generated passwords. This would allow me to compare user security versus automated security and provide users a possible solution to lazy password creation. This led me to my research question: Are user-generated passwords are more secure than randomly generated passwords?

Literature Review

To understand the purpose of my investigation into password generation, one must understand the current state of cybersecurity. With over 53,000 cybersecurity incidents and 2,200 data breaches in 2018 alone, proper cybersecurity policies are becoming increasingly crucial to online safety (Verizon, 2018, p. 4). For this reason, companies like Verizon have released cybersecurity reports “to support much-needed security initiatives, and...illustrate to employees the importance of security” (Verizon, 2018, p. 4). While these resources help large organizations improve their cybersecurity, one group is being left behind: the average computer user. The average uneducated computer user is the perfect target for data theft. In fact, users’ personal data was the most compromised of data by the end of 2017, making up nearly 40% of all data leaked that year (Verizon, 2018, p. 9).

Some researchers do not see users as innocent victims of data thieves. Josiah Dykstra and Eugene Spafford (2018) believe that users today are not taking the correct steps to protect themselves from cyber attacks, inviting hackers into their systems. Alain Forget et. al (2016) agrees with this, claiming that oftentimes, users do not secure their systems as expected. The researchers also claim that users' perspectives of their computing expertise is often misaligned with their actual expertise, causing them to make incorrect decisions regarding the security of their devices (Forget et. al., 2016). Others choose to ignore warnings to update programs or run security scans when malware is detected (Forget et. al., 2016).

Seeing users as a roadblock between computer systems and ideal security, Dykstra and Spafford (2018) suggest a different solution: automating security. According to them, users would be safer if security software functioned without explicit notification to or consent from the user (Dykstra & Spafford, 2018). Creating "invisible" security systems would prevent naive users from creating loopholes in their digital spaces and let them feel comfortable with the security of their personal devices (Dykstra & Spafford, 2018). However, Dykstra and Spafford (2018) admit there has been no research done into the efficacy of automated security. The reason is simple: the expanse of cybersecurity is too wide. To assess how the automation of security would affect users, one would have to analyze security with different contexts and security controls. For this reason, I propose to analyze the efficacy of automated security as compared to "user-decided" security and determine which is more secure in terms of password generation. By testing randomly generated passwords and user-generated passwords against hacking tools, I hope to shed light on this debate and encourage other researchers to pursue this avenue of study.

Studying user password generation is important because of the sheer number of online accounts users create. With the average user holding around 25 online accounts, generating secure passwords is a necessary skill for users to have (Florêncio & Herley, 2007). However, users often fail to create secure passwords. In their study analyzing password generation behaviors, Dinei Florêncio and Cormac Herley (2007) discovered that most user-generated passwords are short and heavily based on lowercase letters, and that even when users were prompted for longer passwords, lowercase letters made up nearly 78% of those passwords. Though these statistics are daunting, Florêncio and Herley's (2007) methodology creates limitations to their study that prevented them from analyzing user password generation "in the wild." To gather their data, the researchers used software to monitor users entering passwords online in real time. However, some websites prompted for certain password specifications while others did not. This element of variability creates two problems. First, it prevented the researchers from examining all of the data they received on an even playing field. Second, it prevented the researchers from observing how users would create passwords unprompted.

A separate study done by Philip Inglesant and M. Angela Sasse (2010) analyzing password generation patterns in the workplace fell into a similar trap. In this study, employees were asked to record password events into a journal on their own time and participate in interviews. This method is untrustworthy as participants could forget to record events, neglect to record events purposely, or refuse to answer interview questions truthfully to protect their privacy (Inglesant & Sasse, 2010).

Improving upon these existing methodologies, I propose to derive user-generated passwords by asking users to register for a fake website. By having users register on the same

platform and create a password without any prompts for certain password specifications, I can analyze user password patterns on a level playing field and “in the wild” without directly interacting with them and creating response bias.

However, in studying default password generation as a comparison to user-generated passwords, I noticed that default password requirements are equally as variable. According to Xavier de Carné de Carnavalet and Mohammad Mannan (2014), many websites have conflicting views on what is considered to be a secure password. Password length requirements for these websites can range from one to a hundred characters (Carnavalet & Mannan, 2014). Similarly, some websites require specific character sets, or charsets, (only alphabet, alphanumeric, capitalization, etc) while others have no requirements (Carnavalet & Mannan, 2014). This indicates that there still exists a debate across the IT community as to what is considered a secure password.

To resolve this issue, I will create a set of randomly generated passwords based upon several standards that companies follow when prompting users for passwords, such as a random length from eight to sixteen characters or a certain number of non-alphabetical characters. This will allow me to rule default password variability out of the equation when conducting my study.

My proposed methodology also addresses an underlying problem to nearly all the studies mentioned thus far: the sample population. In most of the currently existing studies regarding a user’s cybersecurity policies or user-generated passwords, data has been derived from an older population. In fact, studies like the one performed by Forget et. al. (2016) only involved participants between the ages of 42 and 80. Surprisingly, members of the younger generation (ages 20 or below) rarely appear as participants in studies involving cybersecurity, which is odd

since the younger generation is often associated with technology. For this reason, I propose to conduct my study on younger participants, specifically those ages 14 to 18, to analyze how password generation specifically affects the youth.

Addressing these gaps in currently existing research will allow me to add to the automated versus user-decided cybersecurity debate and inspire other researchers to investigate automated security solutions. Outside of this debate, my study can provide crucial information about user password generation patterns in the youth that can help professionals create stronger password policies and stronger default passwords.

Method

My study can be broken into several smaller processes. The first step was to identify the population I was taking my sample of user-generated passwords from. The population for the sample of participants included teenagers attending my high school district from 2018 to 2019. This population selection was based off of two factors. For one, previous studies analyzing cybersecurity policies only involved participants who were from older generations, so choosing the younger generation allowed me to analyze cybersecurity through a different lense.

Additionally, the population chosen was easily available, making the study easier to conduct.

The individuals for my experiment were recruited through a blast email and participated voluntarily. These participants were asked to sign an informed consent form and attend the study during their free period in school. They were asked to access a computer one by one and enter the required data as prompted (without my intervention) and after they completed the task, they received a piece of chocolate as their reward for attending.

The task itself was to create an account on the “registration software” I had created, log-in with said account, and then complete a set of survey questions about their general computer usage habits. Throughout the process, students were only allowed to ask questions about the wording of the survey questions or questions about their account if technical issues occurred. No feedback was given to them in terms of the specifications of the passwords they created on the interface. Images showing the interface used can be seen in Appendix A.

To prevent response bias on this platform while also creating a way to legally collect data from participants, I developed a generic informed consent. The informed consent mentioned the purpose of the study as being a study to analyze teenagers’ computer usage habits and specified the purpose of the study (to investigate the computing behaviors of students), the incentive (chocolate and extra credit), and the risks and benefits associated with the study. Participants were required to hand in the informed consent before taking part in the study. Although account information was collected from the students, their actual identities were kept private when producing and analyzing the data for this study. The informed consent can be seen in more detail in Appendix B.

After creating the informed consent, I developed the fake test registration software I needed to record the participants’ username and password entries. To do so, I used the Eclipse Photon software to develop this interface through Java programming integrated with an apache database. Participants were not allowed to “register” for the software and conduct the survey if the email or password field was left blank, but as long as one character was entered into the system, the participant was able to “register” and move on with the study. Storing the data in an apache database allows for the easy transferral of the data between different systems, as well as

the manipulation of those values within the database. Apache database also provide their own encryption schema, which allows database files to be more secure than text files. Deletion is also simple on an apache database making them a more convenient storage method in terms of disposing of the data after the study.

After developing the frame for the program, I created a graphical user interface (GUI) to allow users to interact with my program like a normal piece of software instead of using antiquated methods, like running the program through a terminal. By using the Swing GUI widget toolkit which is available as a package on Eclipse Photon, I created a simple button and textfield interface that made the software itself relatively convincing as a fake registration site for conducting digital surveys. Creating this interface also allowed me to integrate the digital survey into the program itself. The digital survey asked simple questions about computer usage. For instance, participants were asked how many devices they have, the number of hours they spend on devices a day, and other similar questions. This survey was created simply to mask the true reason of the study, which was to analyze unprompted user-generated passwords. None of the participants' responses regarding these survey question were actually stored, which minimized the amount of data that I needed to transfer between different devices.

Using this programming interface, I allowed my program to read and write data to the apache database. With the help of my mentor, I programmed a way for the database to store the different character sets used in the password, as well as the length of each password. For testing against a common hacking attack, it was necessary for me to encrypt the passwords first. Consulting with my mentor, I decided to use an MD5 hash, a common method of data encryption found on Windows system. By developing an MD5 hash generator, I was also able to generate an

MD5 hash for each password that could later be fed into a password cracker. This information, as well as the hashes, was outputted to a text file, which facilitated the use of John the Ripper, a password cracker that works off of hashes provided in text files.

After creating this registration software, I developed a random password generator using Java Photon. Each password was of a randomly generated length between 4 and 16 characters with the character sets for each password randomly chosen from one of three categories: alphabetic, alphanumeric, and alphanumeric with special characters. These adjustments to passwords were based off of the industry standards discussed in the 2014 paper by De Carné de Carnavalet and Mannan. I created a total of 30 passwords to test against the user-generated passwords.

After gathering this data for randomly generated passwords, these passwords were encrypted into an MD5 hash using software found freely available on the internet. The newly encrypted passwords were stored in a separate text file to prevent the overwriting of data. After encrypting the passwords, I used the open source password cracker John the Ripper to decrypt the passwords. Giving each password a time limit of 1 hour, I used John the Ripper on one password at a time, recording the timestamp at which the password was decrypted. If, by the end of an hour, the password was not decrypted, I recorded the encryption status of the password as “not reasonably encryptable” and moved on to the next password hash. Through this process, I was able to record the decrypting time needed for the user-generated passwords and the default generated passwords.

The next step was to calculate the bit strength of each password. Although there are many methods of quantifying the strength of a password, I decided to use Shannon entropy. Shannon

entropy is a measure of password strength that calculates the entropy, or complexity, of passwords based upon the length of the password and the size of the character pool the password is developed from. The equation for Shannon entropy is as follows, where E is the entropy, R is the size of the character pool from which the password is derived from (52 for alphabetic passwords, 10 for numeric passwords, etc) and L is the length of the password:

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

I chose to use Shannon entropy for a couple of reasons. First, the National Institute of Standards and Technology, a government body that conducts research in the physical sciences, especially the cyberspace, uses Shannon entropy as a measure of security in its 2006 Special Publication for digital security (Burr, Dodson, & Polk, 2006). Since this measure of password security is used by a trusted government body, I thought it appropriate to use for my experiment. Additionally, Shannon entropy does not take into account the hashes used to encrypt a password -- it solely determines the maximum amount of storage needed to store a string of information, and based upon this metric, one can determine how easy or how difficult it is to guess or determine a password (Burr, Dodson, & Polk, 2006). Although other types of entropy exists that may be more accurate than Shannon entropy, like guessing entropy or min-entropy, because I do not have the data needed to use these strength metrics, such as the common types of passwords or characters sets users from my sample may create (as my study does not involve collecting passwords multiple times from a sample of a certain size from the same population), Shannon entropy would be the most effective measure of passwords strength to use (Burr, Dodson, & Polk, 2006). Thus, by using the Shannon entropy equation, I calculated the bit strength of both the user-generated and randomly generated passwords.

In order to analyze the data I generated, I conducted a 2 sample t-test on the data I collected, using the average bit strength of the user-generated passwords and the average bit strength of randomly generated passwords as the two means to compare. The significance level for this test was the standard level of 0.05. Using this procedure, I generated a test statistic that gave me the probability that I generated the data I did from a random sample if the average strengths of a user password was equal to that of a randomly generated password.

Additionally, I calculated the probability of passwords being cracked from the data I collected and graphed the data in histograms to analyze the spread of passwords for user-generated passwords and randomly generated passwords.

These two results were used to draw conclusions about whether or not user-generated passwords are truly more secure than randomly-generated passwords.

After the completion of the study, debrief forms were sent to the participants to inform them of the true purpose of the study. Participants were allowed to question me about the results of the study but could not ask about specific information that could jeopardize someone's privacy or digital space (i.e. information about the identity of participants or the actual passwords typed into the interface). The debrief form can be seen in its entirety in Appendix B.

All of the password metric data collected from volunteer passwords and randomly generated passwords can be seen in Appendix C.

Results



Figure 1. Frequency of Password Lengths for User-Generated Passwords. This figure shows the spread of lengths from passwords created from the volunteers the study.



Figure 2. Frequency of Password Lengths for Randomly Generated Passwords. This figure shows the spread of lengths from passwords generated randomly.

The average length of user-generated passwords was 9.324 characters, ranging from 3 characters to 18 characters, as can be seen in Figure 1. The average length of randomly generated passwords was 10.167 characters, ranging from 4 characters to 15 characters, as shown in Figure 2.

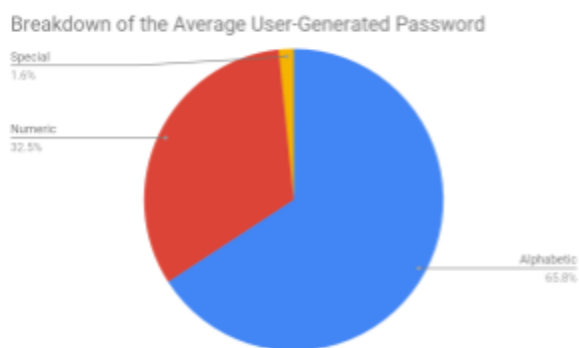


Figure 3. Breakdown of the Average User-Generated Password. This figure shows the percentage of the average volunteer-created password that is constructed of alphabetic characters, numeric characters, and special characters.

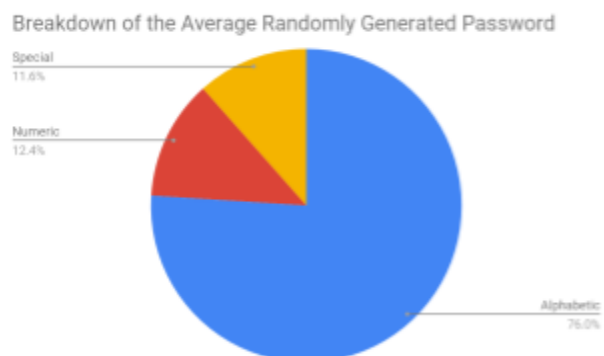


Figure 4. Breakdown of the Average Randomly Generated Password. This figure shows the percentage of the average randomly generated password that is constructed of alphabetic characters, numeric characters, and special characters.

When breaking down the character sets used in user-generated passwords, as seen in Figure 3, 66.5% of the average password consists of alphabetic characters, lowercase and uppercase, 31.7% of the password consists of numeric characters, and 1.8% of the password

consists of special characters. When breaking down the character sets used in the average randomly generated password, as seen in Figure 4, 76.0% of the average password consists of alphabetic characters, lowercase and uppercase, 12.9% of the average password consists of numeric characters, and 11.2% of the average password consists of special characters.



Figure 5. Frequency of Password Entropy for User-Generated Passwords. This figure shows the variety in Shannon entropy values for each of the passwords developed by the volunteers.



Figure 6. Frequency of Password Entropy for Randomly Generated Passwords. This figure shows the variety in Shannon entropy values for each of the randomly generated passwords.

The Shannon entropy of user-generated passwords ranged from 9.966 bits of strength to 102.608 bits of strength, as is shown in Figure 5. The average Shannon entropy of user-generated passwords was 54.675. On the other hand, the Shannon entropy of randomly generated passwords ranged from 22.802 bits of strength to 91.978 bits of strength, as is shown in Figure 6. The average Shannon entropy of randomly generated passwords was 61.254.

The two sample t-test that was conducted comparing the bit strength of user-generated passwords to that of randomly generated passwords had a null hypothesis stating that the average bit strengths of the two populations was the same, and had a two-sided alternative hypothesis. Conducting the t-test resulted in a test statistic of -1.284906013 and a p-value of 0.1020202108.

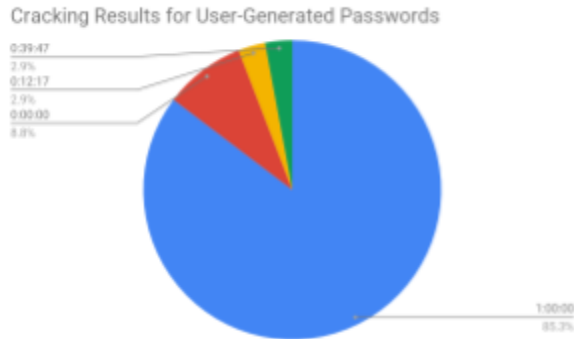


Figure 7. Cracking Results for User-Generated Passwords. This figure shows the percentage of passwords from the volunteer pool that were cracked by John the Ripper within 1 hr. The blue slice (1:00:00) represents passwords not cracked by John the Ripper within 1 hour.

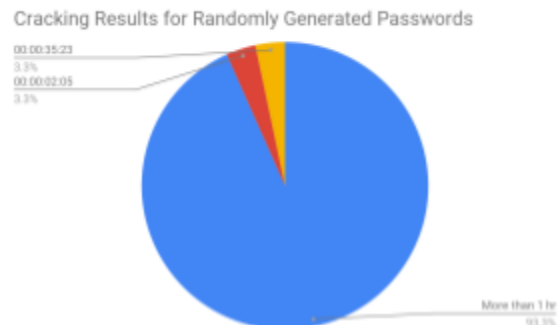


Figure 8. Cracking Results for Randomly Generated Passwords. This figure shows the percentage of randomly generated passwords that were cracked by John the Ripper within 1 hr. The blue slice (1:00:00) represents passwords not cracked by John the Ripper within 1 hour.

5 out of the 34 user-generated passwords were cracked under one hour, resulting in an under one-hour cracking probability of 14.71%, while 2 out of the 30 randomly-generated passwords were cracked under one hour, resulting in an under one-hour cracking probability of 6.67%. Figures 7 and 8 show the difference in the cracking probabilities between both samples.



Figure 9. Strength of Passwords vs Cracking Time for User-Generated Passwords. This figure compares cracking times and Shannon entropies for volunteer generated passwords that were cracked within 1 hour by John the Ripper.

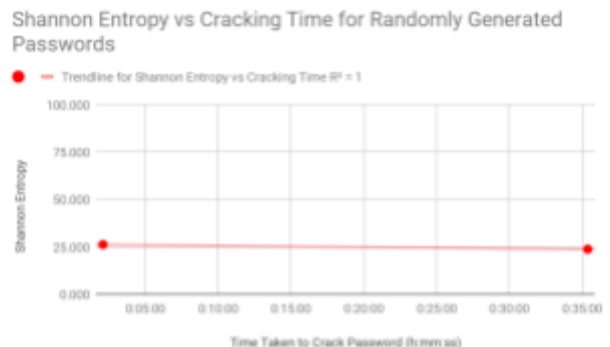


Figure 10. Strength of Passwords vs Cracking Time for Randomly Generated Passwords. This figure compares cracking times and Shannon entropies for randomly generated passwords that were cracked within 1 hour by John the Ripper.

To directly measure how password entropy related to cracking time in order to determine whether Shannon entropy effectively measured password strength, both metrics were graphed against each other for user-generated passwords and randomly generated passwords. All passwords that took greater than an hour to crack were not graphed as their true cracking times were not known. It was found that for user-generated passwords, although a positive correlation

exists between shannon entropy and cracking time, the correlation is very weak, as is shown in Figure 9. Only about 32.9% of the variation in the shannon entropy of user-generated passwords explainable by a linear model comparing entropy with cracking time. On the other hand, a very strong negative correlation was found between the Shannon entropy and cracking times of randomly generated passwords, as is shown in Figure 10. About 100% of the variation in the Shannon entropy of randomly generated passwords is explainable by a linear model comparing entropy with cracking time.



Figure 11. Shannon Entropy of User-Generated Passwords Not Cracked in 1 Hour. This figure shows the variety in the Shannon entropies of the passwords created by volunteers that were not cracked within the 1 hour time limit by John the Ripper.



Figure 12. Shannon Entropy of Randomly Generated Passwords Not Cracked in 1 Hour. This figure shows the variety in the Shannon entropies of the randomly generated passwords that were not cracked within the 1 hour time limit by John the Ripper.

Additionally, when looking at passwords not cracked in 1 hour, for user-generated passwords, the Shannon entropy varies from 41.679 bits of strength to 102.608 bits of strength while randomly generated passwords not cracked in 1 hour have Shannon entropies that vary from 26.279 bits of strength to 91.978 bits of strength. These variations can be seen in Figures 11 and 12.

Discussion

According to the above results, because the p-value, 0.1020202108, is greater than the standard significance level of 0.05, the results of the t-test show that we cannot reject the null hypothesis (that the average bit strengths of these two population are the same), and show that there is no convincing evidence that the average bit strengths of user-generated passwords and randomly generated passwords are different. Based off just the Shannon bit strength, I cannot conclude that user-generated passwords are less secure than randomly generated passwords. However, in terms of cracking time, the cracking probability of user-generated passwords was double that of randomly generated passwords, indicating that although in terms of bit strength, user-generated passwords and randomly generated passwords are similar, because of a lack of length, it is easier for brute force algorithm like John the Ripper to guess user-generated passwords than it is for randomly generated passwords. This is doubly supported by the fact that the average length of user-generated passwords is less than that of randomly generated passwords. Based on this, it can be said that in situations where users are not prompted for specific characteristics when developing a password, although user-generated passwords and randomly generated passwords are similar in measures of entropy, user-generated passwords are more susceptible to be cracked by a brute force algorithm, and in this way, are less secure than randomly generated passwords when it comes to the real world applications of passwords.

However, there exists a disconnect between cracking time and bit strength that may point out a limitation in this study. As can be seen Figure 9 and Figure 10, there seems to be no relevant correlation between the two variables. For the user-generated password graph, since the R^2 values is lower than 50%, there is not a strong correlation between Shannon entropy and

cracking time. On the other hand, randomly generated passwords showed a very strong negative correlation between shannon entropy and cracking time, which is against the entire idea of entropy -- the greater the entropy, the harder it is to crack a password and the longer it should take to crack the password. Taking into account the passwords not cracked in 1 hour, for both user-generated and randomly generated passwords, there is unexpected variation in the Shannon entropies of passwords not cracked in 1 hour, as can be seen in Figure 11 and Figure 12. This indicates that perhaps Shannon entropy is not a good measure of password strength in this study as it is not closely related to the actual cracking time of passwords. This makes sense if one considers the strengths and weaknesses of Shannon entropy. Shannon entropy is only one measure of the strength of a password, and according to Cornell University's CS5430 online coursework, written by Professor Fred B. Schneider in 2017, it is an ineffective measure of strength for user-generated passwords as it assumes that users are just as likely as selecting one character in a pool of a given size as they are any other character in that character pool. However, this assumption is not true in the real world. According to Schneider (2017), humans are more likely to use english words, which reduces the entropy of these passwords. This means that Shannon entropy is perhaps not the most effective measure of password security, which also explains the dissonance between the conclusions drawn from the Shannon entropy of the passwords and the cracking time for the passwords.

Additional limitations include the sample of users who developed passwords for the study. Due to the fact that the sample consisted of students aged 14 to 18 from my school district, conclusions about the efficacy of user-generated passwords as compared to randomly generated passwords can only apply to the population of students from my district. Though students from

my district come from towns all over my county and can thus be considered to represent students from around the county, the conclusions about password strength taken from this study are restricted geographically and age wise to high school students within my county.

However, these limitations do not overshadow the implications of this study. Although this study is more of a pilot study, it indicates that perhaps entrusting password generation, and by extension, security, solely to users is not safe. If more than 10% of user passwords can be cracked by a simple brute-force algorithm that uses no dictionary or rainbow tables, it may be very simple for hackers to hack into user accounts and steal important information. This is especially true as super computers and developments in technology make it easier for malicious entities to obtain the technology they need to conduct these malevolent activities. One important implication of this information is that perhaps password generating and account securing methodologies must be revised. Introducing systems that hybridize password generation between users and automated systems may make it more difficult for hackers to decipher passwords. These hybridized systems may even include having users create a part of a password and having a machine create the other part, as is suggested by Cornell University's course material on password generation. Another method suggested by this same source may be equally as effective: password wallets. Password wallets are systems where machines randomly generate passwords for all sites that a user has accounts on, but only signs the user into these accounts automatically if a user unlocks the wallet using a master password. By having the user create a single, very secure master password, a system can be created that not only has the security of random password generation, but also gives user some level of control over the accounts they create online.

Above all else, this study indicates that it is important for users to be aware of the passwords they create and where they decide to enter and store their information online. Perhaps steps as simple as cyber-education can greatly increase the secureness of user-generated passwords, and by extension, security, and make the issue of vulnerability caused by users an issue of the past.

References

- Burr, W. E., Dodson, D. F., & Polk, W. T. (2006). *Electronic authentication guideline: Recommendations of the National Institute of Standards and Technology*(United States of America, National Institute of Standards and Technology, Research Support Services Office). Gaithersburg, MD: U.S. Dept. of Commerce, National Institute of Standards and Technology. Retrieved April 20, 2019, from <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-63ver1.0.2.pdf>
- De Carné de Carnavalet, X., & Mannan, M. (2014). From Very Weak to Very Strong: Analyzing Password-Strength Meters (Rep.). Retrieved December 10, 2018, from <http://users.encs.concordia.ca/~mmannan/publications/password-meters-ndss2014.pdf>
- Dykstra, J., & Spafford, E. H. (2018). The Case for Disappearing Cyber Security. *Communications Of The ACM*, 61(7), 40-42. doi:10.1145/3213764
- Florêncio, D. and Herley, C. (2007 May) A Large-Scale Study of Web Password Habits. Paper presented at the International World Wide Web Conference, Banff, Alberta, Canada. doi: 10.1145/1242572.1242661
- Forget, A., Pearman, S., Thomas, J., Acquisti, A., Christin, N., Cranor, L. F., . . . Telang, R. (n.d.). Do or Do Not, There Is No Try: User Engagement May Not Improve Security Outcomes. In Symposium on Usable Privacy and Security (SOUPS). Retrieved December 10, 2018, from <https://www.usenix.org/system/files/conference/soups2016/soups2016-paper-forget.pdf>
- Inglesant, P., & Sasse, M.A. (2010 July 16). Studying Password Use in the Wild: Practical

Problems and Possible Solutions. Paper presented at the Symposium On Usable Privacy and Security (SOUPS), Redmond, WA, USA. doi: 10.1.1.415.4500.

Livshits, V. B., & Lam, M. S. (2005, August). Finding Security Vulnerabilities in Java

Applications with Static Analysis. In *USENIX Security Symposium* (Vol. 14, pp. 18-18).

Lyu, M. R., & Lau, L. K. (2000). Firewall security: Policies, testing and performance evaluation.

In *Computer Software and Applications Conference, 2000. COMPSAC 2000. The 24th Annual International* (pp. 116-121). IEEE.

Petukhov, A., & Kozlov, D. (2008). Detecting security vulnerabilities in web applications using

dynamic analysis with penetration testing. *Computing Systems Lab, Department of Computer Science, Moscow State University*, 1-120.

Schneider, F. B. (2017). Passwords, part 2. Retrieved April 20, 2019, from

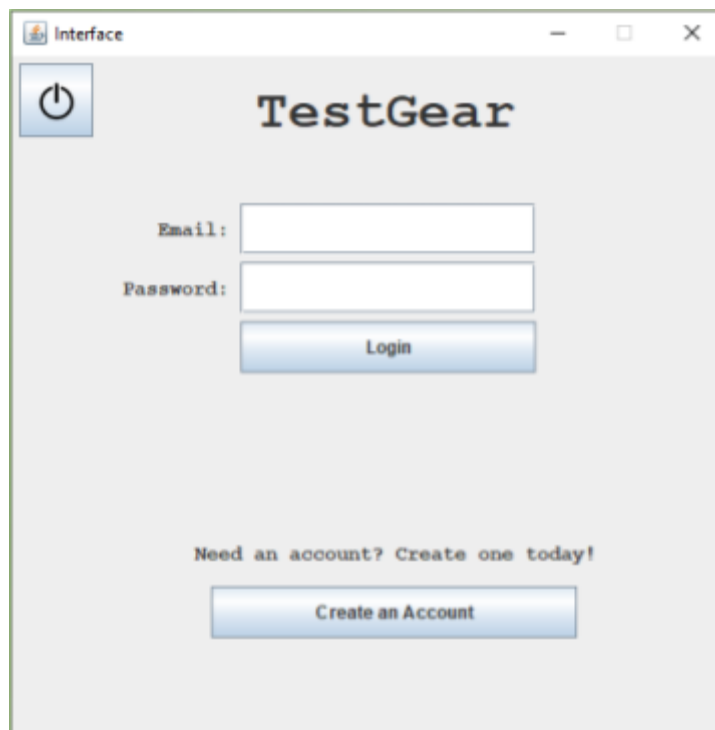
<http://www.cs.cornell.edu/courses/cs5430/2017sp/l/14-passwords2/notes.html>

Verizon. (2018). 2018 Data Breach Investigations Report (Rep.).

Retrieved October 1, 2018, from Verizon website:

https://www.verizonenterprise.com/resources/reports/rp_DBIR_2018_Report_en_xg.pdf

Appendix A



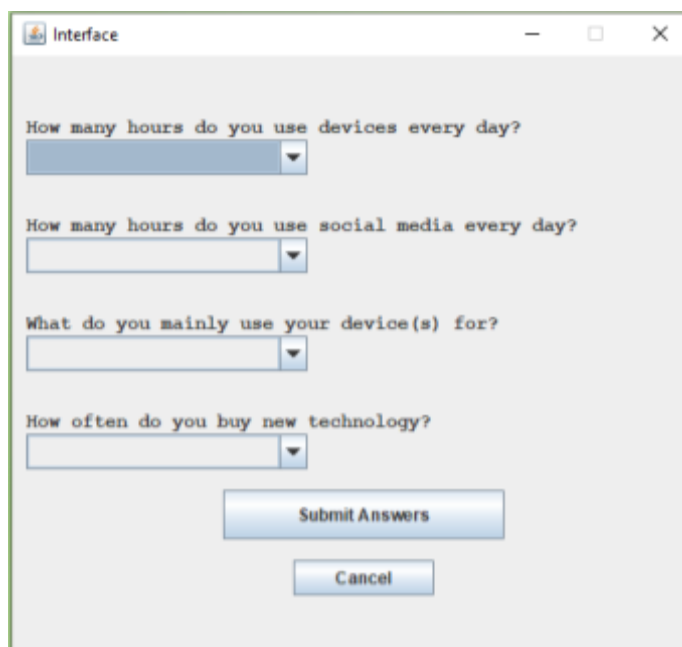
The screenshot shows a window titled "Interface" with a power button icon in the top-left corner. The main heading is "TestGear". Below the heading, there are two input fields: "Email:" and "Password:". A "Login" button is positioned below the password field. At the bottom of the window, there is a text prompt "Need an account? Create one today!" followed by a "Create an Account" button.

Figure 13. Surveying software created for volunteers to interact with. This is the screen where volunteers "logged in" by entering the email and password the created.



The screenshot shows a window titled "Interface" with a power button icon in the top-left corner. The main heading is "TestGear Registration". Below the heading, there are two input fields: "Email:" and "Password:". Below the password field, there are two buttons: "Create Account" and "Cancel".

Figure 14. This is the screen where volunteers "registered" for the software by entering a valid email and by creating a password.



The screenshot shows a window titled "Interface" with a light gray background. It contains four survey questions, each followed by a dropdown menu:

- How many hours do you use devices every day?
- How many hours do you use social media every day?
- What do you mainly use your device(s) for?
- How often do you buy new technology?

At the bottom of the window, there are two buttons: "Submit Answers" and "Cancel".

Figure 15. This is the interface where volunteers would answer survey questions. Since the survey questions were meant to distract from the real purpose of the study, responses entered for these questions were not recorded.

Appendix B

[Redacted]

Consent to Participate in a Research Study

[Redacted]

Title of Study: Determining the Computer Usage Patterns of Average Adolescent Users

Investigator: [Redacted]

Adviser: **[Redacted]**

Introduction
You are being asked to be in a Research study about people's computer use. We ask that you read this form and ask any questions that you may have before agreeing to be in the study.

Purpose of Study
The purpose of the study is to gather information about people's computer use. Ultimately, this Research will be referenced in an academic paper and verbal presentation submitted to College Board to satisfy the requirements for the AP Capstone Research course.

Description of the Study Procedures
If you agree to be in this study, you will be asked to do the following things: register for an account to access the inventory and then complete the inventory.

Risks/Discomforts of Being in this Study
There are no reasonable, foreseeable (or expected) risks. There may be unknown risks.

Benefits of Being in the Study
The benefits of participation include the incentive (see below) and becoming more aware of your own computer usage.

Confidentiality
This study is anonymous. We will not collect or retain any personally identifiable information like your name or username.

Incentives
You will receive the following incentive: a piece of chocolate candy, which will be provided upon your completion of the study procedures as outlined above.

Figure 16. This figure shows the first page of the informed consent given to volunteers to review and sign before taking part in the study.

Right to Refuse or Withdraw
 The decision to participate in this study is entirely up to you. You may refuse to take part in the study at any time without affecting your relationship with the investigator or adviser of this study or with the [REDACTED]. Your decision will not result in any loss or benefit to which you are otherwise entitled. You have the right not to engage in a specific procedure or task, as well as to withdraw completely from the study at any point during the process; additionally, you have the right to request that the investigator not use any of the information collected from you.

Right to Ask Questions and Report Concerns
 You have the right to ask questions about this Research study and to have those questions answered by the adviser before, during, or after the study. If you have any further questions about the study, at any time feel free to contact [REDACTED]. If you like, a summary of the results of the study will be sent to you. If you have any problems or concerns that occur as a result of your participation, you can report them to Laura Binkhorst at the number above. Alternatively, concerns can be reported to [REDACTED].

Consent
 Your signature below indicates that you have decided to volunteer as a Research participant in this study, and that you have read and understood the information provided above. You will be given a signed and dated copy of this form to keep, along with any other printed materials deemed necessary by the investigator or adviser.

Participant's Name (print): _____

Participant's Signature: _____ Date: _____

Investigator's Signature: _____ Date: _____

Adviser's Signature: _____ Date: _____

Figure 17. This figure shows the second page of the informed consent volunteers had to read and sign before taking part in my study. Some lines contained personally identifiable information and are censored in black.

Dear Participant,

Thank you for taking part in this study. We initially described this study as testing a user's computer usage to negate response bias. This study's goal is to analyze the password generating habits of the average user. When you registered for an account to take the computer usage survey, the password you generated stored in a database from which a program can test its strength. Your participation remains anonymous; the only stored information is the password itself, and no usernames or other personally identifiable information, including your responses on the computer usage survey, were collected. As such, it is impossible to determine who submitted which password.

Again, thank you so much for completing this study!

Best regards,

[REDACTED]

Figure 18. This figure shows the debrief form that was sent to volunteers after the completion of the study to inform them of the true purpose of the study. Personally identifiable information has been censored in black.

Appendix C

Table 1				
<i>Password metrics collected from passwords generated by volunteers. Some information is omitted to protect the volunteers' privacy.</i>				
Volunteer #	Length	Character Pool Size	Shannon Entropy	Cracking Time
1	9	62	53.588	More than 1 hr
2	8	10	26.575	00:00:00:00
3	9	62	53.588	More than 1 hr
4	8	62	47.634	More than 1 hr
5	3	10	9.966	00:00:00:00
6	9	85	57.685	More than 1 hr
7	10	62	59.542	More than 1 hr
8	9	62	53.588	More than 1 hr
9	6	10	19.932	00:00:00:00
10	8	62	47.634	More than 1 hr
11	10	62	59.542	More than 1 hr
12	14	62	83.359	More than 1 hr
13	8	52	45.604	00:00:12:17
14	11	95	72.268	More than 1 hr
15	7	62	41.679	More than 1 hr
16	9	62	53.588	More than 1 hr
17	8	95	52.559	More than 1 hr
18	8	62	47.634	More than 1 hr
19	8	62	47.634	More than 1 hr
20	18	52	102.608	More than 1 hr
21	8	62	47.634	More than 1 hr
22	14	62	83.359	More than 1 hr
23	10	62	59.542	More than 1 hr

24	14	62	83.359	More than 1 hr
25	9	62	53.588	More than 1 hr
26	9	95	59.129	More than 1 hr
27	11	62	65.496	More than 1 hr
28	14	62	83.359	More than 1 hr
29	6	62	35.725	00:00:39:47
30	8	62	47.634	More than 1 hr
31	9	95	59.129	More than 1 hr
32	8	52	45.604	More than 1 hr
33	8	52	45.604	More than 1 hr
34	9	62	53.588	More than 1 hr
<i>Note.</i> The average password length was 9.323 characters, the average bit strength was 54.675 bits of strength, and 14.71% of passwords were cracked within 1 hour by John the Ripper.				

Table 2

Password metrics collected from passwords that were randomly generated. Some information is omitted to maintain consistency.

Password #	Length	Character Pool Size	Shannon Entropy	Cracking Time
1	10	62	59.542	More than 1 hr
2	15	52	85.507	More than 1 hr
3	9	62	53.588	More than 1 hr
4	13	62	77.405	More than 1 hr
5	4	95	26.279	00:00:02:05
6	14	52	79.806	More than 1 hr
7	11	52	62.705	More than 1 hr
8	10	95	65.699	More than 1 hr
9	12	62	71.450	More than 1 hr
10	4	52	22.802	More than 1 hr
11	9	95	59.129	More than 1 hr
12	12	52	68.405	More than 1 hr
13	15	52	85.507	More than 1 hr
14	9	62	53.588	More than 1 hr
15	13	52	74.106	More than 1 hr
16	4	95	26.279	More than 1 hr
17	8	62	47.634	More than 1 hr
18	7	52	39.903	More than 1 hr
19	13	95	85.408	More than 1 hr
20	15	52	85.507	More than 1 hr
21	14	95	91.978	More than 1 hr
22	4	62	23.817	00:00:35:23
23	5	95	32.849	More than 1 hr

24	10	52	57.004	More than 1 hr
25	13	62	77.405	More than 1 hr
26	4	95	26.279	More than 1 hr
27	10	95	65.699	More than 1 hr
28	10	52	57.004	More than 1 hr
29	14	62	83.359	More than 1 hr
30	14	95	91.978	More than 1 hr

Note. The average password length was 10.167 characters, the average bit strength was 61.254 bits of strength, and 6.67% of passwords were cracked within 1 hour by John the Ripper.