Issues in Information Systems
Volume 23, Issue 2, pp. 29-41, 2022

DOI: https://doi.org/10.48009/2_iis_2022_103

Password change requirements and the effective strength of passwords

Mr. David Sangrey, *Robert Morris University, drsst910@mail.rmu.edu*
Dr. Ping Wang, *Robert Morris University, wangp@rmu.edu*

Abstract

Password policies are the most common method of securing accounts against unauthorized access. However, the definition of what makes a secure password policy has undergone numerous revisions as technology advances. Recently, one critical aspect of many password policies, that of mandating users to change their password after a set amount of time, has come under scrutiny. Some guidelines suggest that mandating password changes at set intervals results in less secure passwords being created by users over time. This study attempts to test that assertion by comparing the relative security of passwords used by study participants when presented with different change requirements over an 8-week period. Passwords are measured by a mathematical value called password entropy, which predicts the number of attempts needed to guess the value by brute force. In addition, participants were asked to complete a survey about patterns or tools they used during the study to help generate or store passwords. These entropy values, responses, and averages taken across participant groups are analyzed to attempt to determine if change requirements have an impact on the relative strength of passwords over time. The findings from this study did not support the elimination of mandatory password change requirements, however, given the limited size of the study, more research is required to validate the patterns found.

Keywords: Password security policy, Password strength, Entropy, Password changes

Introduction

As long as there have been user accounts and the Internet, there has been a need to secure these accounts against unauthorized access. One of the most popular methods for securing accounts has been usernames and passwords. A password, known in the field of information security as a “secret”, is a pattern, word, combination, or another type of information that is supposedly known to the user only and can be checked to validate that a user is indeed who he or she claims to be. These standards and definitions are put forth in the National Institute of Standards and Technology (NIST) Special Publication 800-63B Section 5.1.1.2 Memorized Secret Verifiers (NIST, 2017). NIST is the US Government agency charged with establishing standards for a wide variety of technical and non-technical categories. As simple as account security may sound, there are in fact a great number of complexities regarding passwords and their use and many factors that need to be considered to keep accounts truly secure. In 2017, official NIST guidelines on passwords changed; NIST asserted that mandatory password changes in fact significantly reduce the overall security of a password system, and should not be employed, as explained in questions B05 and B06 of NIST’s own FAQ documentation for the updated special publications (NIST, 2020). However, despite these changes in standards, there has been little research available regarding whether the lack of password expiry requirements results in users creating more secure passwords. There are also challenges for researchers as to how passwords can be securely studied, due to how passwords need to be stored and treated in a secure environment, while still yielding useful data for analysis.

Passwords have become the “default” method for user authentication on digital systems and devices. The concept of passwords is so etched into modern memory that it is hard to imagine digital security without passwords. Despite countless efforts to remove passwords from their position of wide use, passwords remain “more widely used and firmly entrenched”, weathering every attempt made to replace them (Herley, 2012). Text passwords are an intuitive solution - a simple text field that uses only the input devices available to nearly all computers, the input of which is easy to compare to a known key to see if the provided input is correct. The cost of implementation is low, and there is no “silver-bullet replacement” for passwords found to date that fits all use categories (Herley, 2012). While non-text-based passwords or authentication methods exist, there is a higher cost of implementation and often technical challenges on accuracy with these alternatives. However, all implementations of account authentication suffer from some common security measures that must be taken into account in any system, especially those being used for research purposes.

The goal of this study was to either support or reject the 2017 changes to the NIST Standards, which asserted that mandatory password changes significantly reduce the overall security of a password system, and should not be employed. The study also attempted to develop a safe, secure manner of implementing password security measurements without compromising the security of the stored value and remaining accessible to study participants from anywhere on an internet-accessible website.

Background

Despite the near-ubiquity of passwords in the modern digital account landscape, there are still major issues that persist with research surrounding passwords. Understandably, nobody wants their passwords leaked or compromised. In this vein, it is neither safe nor secure to store passwords in a plain text format. This does mean that it is impossible for researchers using standard tools to view exactly what a user’s password is, without resorting to the same tools that malicious actors use such as John the Ripper or other cracking tools (Weir et al, 2010). In addition, many data storage laws require secure storage with encryption and other treatment that render the actual password values themselves unobtainable. Given this limitation, and the reluctance of both participants and researchers to disclose exact password data, avenues available to researchers are few and far between. Large-scale studies often have to rely on holes in software or other ‘hacks’ to gain access to plain-text passwords, and nearly always researchers have “no access to plaintext passwords”, with code only giving generic reports running on isolated machines (Mazurek et al, 2013). This situation leaves researchers to look at the metadata that can be gathered from passwords and user testing, such as the number of attempts to log in, how often a password is changed, or other commonly-recorded data points such as entropy values that can be captured and stored without providing indications as to what the passwords are.

Barring these details, or the rare instance where plain-text or reversible-encrypted passwords have been stored, user feedback is the only other available metric for password research. Due to the required encryption for a strong system where known flaws are not being abused, it is impossible to even determine a list of previously used passwords in a basic system, beyond the option to check previously used hashes - scrambled representations of a password - to determine if a provided password matches a previously used value. While often not the most reliable metric for objective accuracy, many studies in the past have used an “exit survey that gathered qualitative data” after a user testing phase, asking about why passwords were chosen or what aspects went into a user’s choice (Egelman et al, 2013). In fact, user surveys and studies are one of the most effective and accepted methods of gathering information and research about passwords

(Shay & Bertino, 2009; Summers & Bosworth, 2004; Fagan et al, 2017). While other methods of research exist, such as access to leaked password databases and other data that can be artificially generated or found in the wild, these methods require extensive security precautions that are not feasible in many situations. While surveys and user-feedback data are not always perfect, they are among the most reliable methods of gathering information that can be, if not treated as literal truth alone, treated as a general guideline as to the direction of what participants are thinking when combined with more grounded, objective metrics, such as password entropy.

A key concept of this study is that of Password Entropy. As has been discussed, there are many difficulties with password research, which has resulted in a very small body of knowledge about passwords. In fact, this limitation is a factor that is well known by researchers, as system administrators and security professionals are loath to share with researchers large bodies of unencrypted passwords. In fact, it requires rare, "fortunate circumstances, which may not be reproducible in the future" to find a secure and safe way to compare passwords in a plain-text manner (Mazurek et al., 2013). As this is not an option in this case, a different measure must be taken to satisfy the requirement to never store password information without proper encryption. Password Entropy provides this solution. Password Entropy is a mathematical model to determine "how difficult a given password would be to crack through guessing, brute force cracking, dictionary attacks or other common methods" in order for an attacker to gain access to an account (*How to Calculate Password Entropy?,* n.d., para. 1). A brute force attack is one in which every possible permutation of characters is tried in order to guess a password. To determine password entropy (E), the equation $E=log_2(S^L)$ is used, according to the points below (*How to Calculate Password Entropy?,* n.d). To extrapolate how many guesses (G) would be needed to have a 50% chance of guessing the password, the formula is $G=2^{E-1}$. In the study, entropy was calculated before the password was encrypted by the website for storage. This entropy value was then stored in a database, with automated processes calculating for each individual user and for groups as a whole the average password entropy of the dataset on a weekly basis. In the entropy equation, the following variables are used:

- S is the size pool of unique characters. Some example pool sizes include
  - Numbers (0-9): 10
  - Lower Case Latin Alphabet (a-z): 26
  - Lower Case & Upper Case Latin Alphabet (a-z, A-Z): 52
  - ASCII Printable Character Set (a-z, A-Z, symbols, space): 95
- L is the length of the password,
- Higher "Entropy" values are considered better, with 36+ "bits" of entropy being rated "reasonable" or higher (Pleacher, n.d).

For example, a password like "S@mp1ePas$word", consists of 2 capital letters, 9 lowercase letters, 1 number, and 2 symbols with a total length of 14 characters. Given the combination of characters, across the whole of the ASCII printable character set, the pool size is 95. With this length and pool size, there are approximately $4.87x10^{27}$ possible passwords that fit the criteria. Therefore, the entropy value can be calculated as $E=log2(95^{14})$, or approximately E=91.98. With this entropy value, we can calculate approximately how many guesses would be required to break this password through brute force methods. Using $G=2^{E-1}$, or, $G=2^{91.98-1}$, we can estimate that there is a 50% probability that the password will have been guessed after $G=2.44x10^{27}$ random guesses. That's over two octillion guesses. On the other hand, a relatively weak password such as "password" only has 8 characters in a set of 26, meaning that the entropy value is only E=37.6. From this, it would only take G=104,159,249,331 guesses to have a 50% chance (or, 104 billion). If we assume that a computer can guess "more than 100 billion" passwords a second, this means the simple password would be cracked nearly instantaneously, even if given a completely random dictionary and not a common password list to start with (O'Shea, Haskell-Dowland, 2020). Comparatively, the higher entropy password would take around twenty-four quadrillion four hundred trillion seconds, or, 7.7 million centuries, to completely randomly guess.

There are some weaknesses with this method, as using logical guesses as to passwords in many cases can reduce the number of guesses needed, if a user has made a password utilizing a "scheme vulnerable to targeted guessing attacks" such as dates of birth, names of pets, simple common words such as 'password' or other such poor choices (Al-Ameen et al, 2015). Despite this weakness, however, password entropy has remained a widely accepted traditional method for measuring password strength where other options are unavailable (Ur et al, 2012).

The question remains, however, how an entropy value can be calculated. For security and legal reasons, at no time can the exact text of a password be stored in a database, nor is it acceptable to save passwords in an encryption scheme that can be easily reversed. The entropy value, however, has no such best-practice restriction on it and may be treated as a safe value. With modern programming, in most systems, the password value is received by the program or webpage as typed by the user in plain text, and nearly immediately encrypted in the chosen cipher, all locally without any transmission across a network. There exists a short period in this chain where the password, while not stored in permanent memory, can be read. By using this short period, the entropy value of a password can be calculated, and that value can be stored without compromising the security of the saved values. This means that a solid metric can be taken and saved, without recording the exact plain-text version of passwords.

Methodology

Recruitment

This human subjects study was reviewed by the Institutional Review Board at the affiliated institution and approved on August 9, 2021. Recruitment of users was performed across the Internet at a number of different locations. For recruitment advertising, a standard recruitment message was posted on multiple Internet forums such as the Reddit communities of r/PCMasterRace (a PC enthusiast community), r/SysAdmin (a community for System Administrators and Server Administrators), and r/CyberSecurity (devoted to cybersecurity-related matters), and distributed among personal connections such as family, work colleagues, and friends. These people were also encouraged to recruit further participants if able, resulting in a gain of a few participants overall. Additionally, posts and recruitment of friends and family were done on an individual basis. Finally, the senior author passed along the recruitment information to students of his as a potential source of recruitment. As a result of all recruiting efforts, a total of 51 participants were recruited for this study. However, 2 participants withdrew from the study between signup and the beginning of the study. One withdrew for personal reasons, and the other as it was discovered the user was underage and therefore could not be allowed to participate.

Participants

Upon signup to the study, participants were asked about their age group to gather some information on what sorts of participants were volunteering for the study. Mostly, this question was used to ensure that underage users (those below the age of 18) were excluded from the study and to ensure that age groups were evenly distributed among the participants. After the signup period had ended, the participants were sorted into 3 groups. Group A was a "Control" group, where no password changes would be requested over the entirety of the study to see if passwords were forgotten by users and set a baseline for entropy changes. Group B was assigned an "Every-Friday" requirement, where the participants would be asked to change their password every Friday and generate a new password. Finally, Group C was a middle ground between the two; participants would be asked to set a new password only on alternating Fridays.

Of the participants who signed up for the study, the oldest participant self-reported to being 81 years old, and the youngest to be 18 years old, with an average age in the whole participant pool of 41.3 years old. Group A had an age range from 20 to 72 years, with an average of 40.2 years. Group B had an age range from 18 to 78 years, with an average age of 41.2 years. Group C had an age range from 20 years to 81 years, with an average of 42.4 years. The objective of this distribution was to not only randomly sort participants into groups, but also ensure that no one group was stacked with significantly older or younger participants and skew the group averages due to age.

After signing up for the study, each participant was assigned a unique user ID number by the system. This user ID number was used to cross-reference all records from the study, including entropy values and demographic responses as a way to mask any personal information users provided to the study. In all analyses, participants were only studied by their unique user ID number and group number.

Software Used and Security Measures

For this study, participants were asked to create an account on a new web-based account management software called UserSpice to facilitate the study's execution. For the study, UserSpice version 5.4.0 was used. UserSpice is a PHP-based user management software "designed from the ground up to be the perfect starting point" for web development and projects that require user accounts (Brown, 2022). It is a free, open-source application, the code of which can be audited by anyone on the Internet. This access allows third-party groups to help validate the security of applications, and allow easy access to customization of the software if required. To log the password entropy, login attempts, and other relevant details about users, the UserSpice software's own 'hooking' system was used to inject these pre-prepared snippets of code into the login and user account pages to log these values to an SQL database stored on the server.

Passwords to these accounts during the study were stored in a BCrypt hashed value, ensuring that passwords saved by the system are securely stored against unauthorized access or decryption. BCrypt is a "deliberately slow … password-hashing function" that is widely recognized as one of many secure standards for password storage (Shay et al., 2016). The fact that UserSpice was written in PHP was also a benefit to the project, as PHP code is never transmitted to a user and all processing is done on the hosting server. This feature means that as little information as possible was sent back to the user's device, especially password values.

The server that hosted the study's website and database was an Amazon Web Services EC2 Linux Instance that utilized a full-disk encryption mechanism to help ensure that unauthorized access was prevented from occurring on the server. This instance hosted the Apache Web Server and MariaDB SQL database that stored all generated user data. Access to the server backend was only accessible through the use of SSH encrypted tunnels and key file access, ensuring that unauthorized users could not connect to the instance.

Additionally, web traffic to the instance was filtered through a service called Cloudflare, which helped to protect against malicious traffic such as Denial of Service attacks or other malicious access attempts while concealing certain details about the host server from the public internet such as the IP address hosting the server or other details that could be used by malicious actors. This service also helped to ensure that all visitors to the website were connecting over a secure, encrypted protocol via TLS 1.2 or greater. These steps helped prevent user data from being intercepted or altered in transit. While this sort of situation was unlikely to happen, it was still important to make sure that data from participants' devices to the host server were secured at all times.

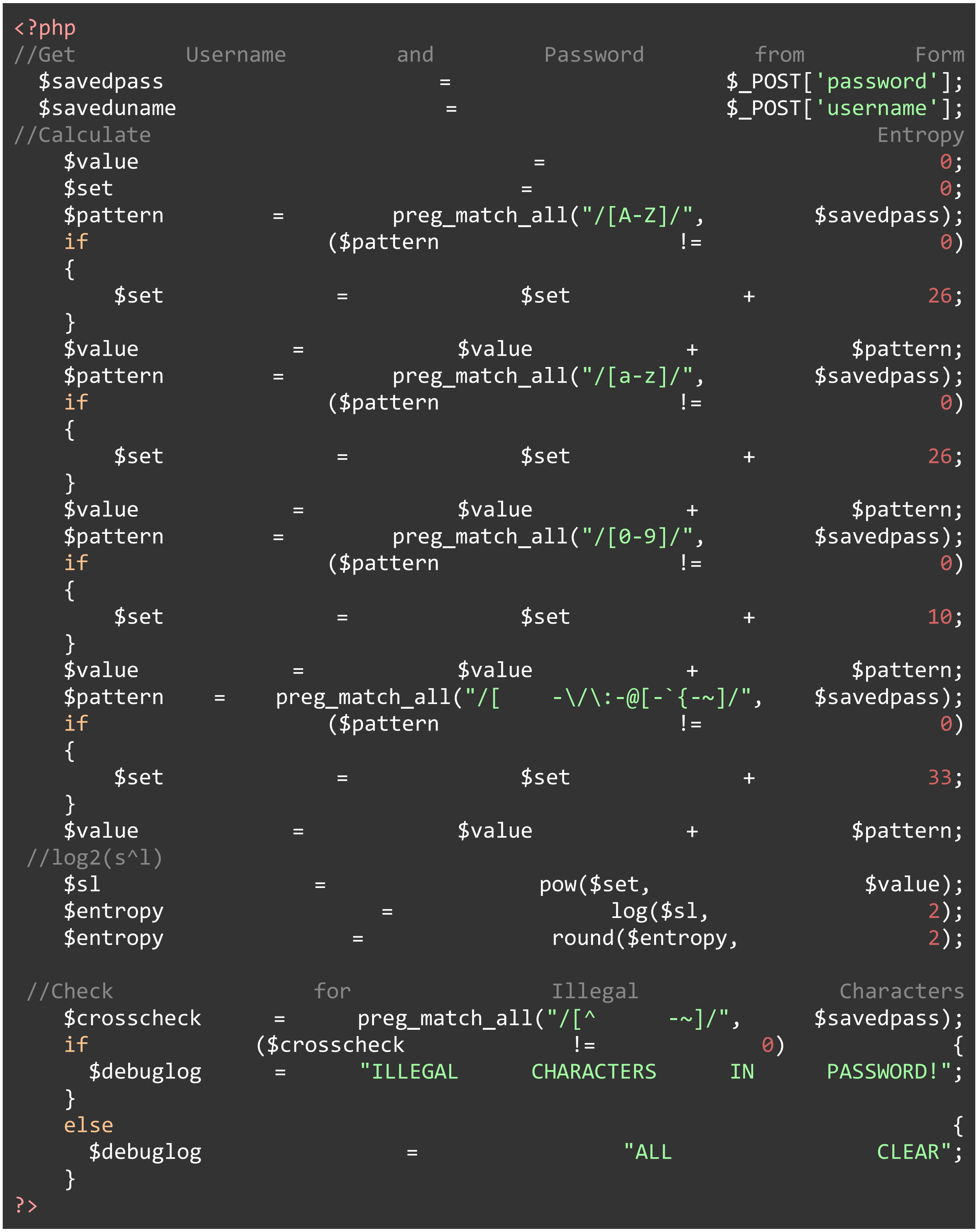
Guidelines Given to Participants

In an attempt to limit potential bias or influence from the study itself toward participants, little guidance on what made a “secure” password was provided. Study participants were greeted with a generic email on study days asking each user to log in to the study website, and if prompted, create a new “secure” password. No limits or artificial requirements were placed on this password, such as arbitrary minimum character limits, special character requirements, or other complex filtering requirements. This is in line with NIST standards in the 2017 Special Publication 800-63b, which states that “users respond in very predictable ways to the requirements imposed by composition rules”, and such requirements are not as effective as were once believed (NIST, 2017). Passwords simply had to be between 2 and 160 characters. This 160-character limit was an arbitrary number chosen for database compatibility reasons.

The Study Process

For the duration of the study (8 weeks), twice a week, users were asked to log into their account on the study website. The login was to test if participants remembered their password, and were still able to use their password, across an extended period. On each Friday check-in, users were asked to complete a task depending on their subgroup. All “A” group members simply had to log in to the system, and after that, were complete with their task for the week. All “B” group members were informed they were required to change their password, and group “C” members were presented with this prompt to change passwords every other Friday for the duration of the study. The system was programmed to automatically log all login attempts, both successful and unsuccessful. The system also recorded automatically all password entropy values anytime a password was updated or reset. These metrics were designed to record if any particular group had a higher rate of failed logins or other errors on login and to track user participation in the study over time.

To properly capture the entropy of passwords used by participants, the entropy value of the user’s password had to be calculated as the user is logging in during validation and processing. As such, several lines of PHP code, including the following snippet, were inserted into the user account software to safely and securely calculate the entropy value on the web host server without exposing or saving the actual plaintext value of the software.



```php
<?php
//Get Username and Password from Form
  $savedpass = $_POST['password'];
  $saveduname = $_POST['username'];
//Calculate Entropy
    $value = 0;
    $set = 0;
    $pattern = preg_match_all("/[A-Z]/", $savedpass);
    if ($pattern != 0)
    {
        $set = $set + 26;
    }
    $value = $value + $pattern;
    $pattern = preg_match_all("/[a-z]/", $savedpass);
    if ($pattern != 0)
    {
        $set = $set + 26;
    }
    $value = $value + $pattern;
    $pattern = preg_match_all("/[0-9]/", $savedpass);
    if ($pattern != 0)
    {
        $set = $set + 10;
    }
    $value = $value + $pattern;
    $pattern = preg_match_all("/[ -\/\:-@[-`{-~]/", $savedpass);
    if ($pattern != 0)
    {
        $set = $set + 33;
    }
    $value = $value + $pattern;
 //log2(s^l)
    $sl = pow($set, $value);
    $entropy = log($sl, 2);
    $entropy = round($entropy, 2);

 //Check for Illegal Characters
    $crosscheck = preg_match_all("/[^ -~]/", $savedpass);
    if ($crosscheck != 0) {
      $debuglog = "ILLEGAL CHARACTERS IN PASSWORD!";
    }
    else {
      $debuglog = "ALL CLEAR";
    }
?>
```

Figure 1. Sample Password Entropy Script in PHP

From this script, password entropy values were saved without actually saving the plaintext content of passwords. Additionally, diagnostic information to catch unanticipated errors or illegal characters that could throw off an entropy calculation were saved to ensure that passwords and entropy values were being calculated correctly. During the course of the survey, no password values chosen by users tripped this safeguard.

Emails were sent to the participants every Tuesday and Friday, using the email addresses the participants had used to sign up to the study. The system was programmed to force participants in the groups who needed to reset their passwords to generate and save a new password at their next logon, which was prompted by the email to users.

Of the 49 users who originally signed up for the survey, 10 members of the Control group (original size: 18), 9 from the "Every-Friday" group (original size: 15), and 9 from the "Every-Other" group (original size: 16) completed the study.

At the end of the study, the following changes in average entropy values were observed of the various groups:

Table 1. Changes in Average Entropy Values

	Group A (No Change) (n=10)	Group B (Every Friday) (n = 9)	Group C (Every Other Friday) (n = 9)
Starting Avg Entropy	71.88	96.05	77.50
Ending Avg Entropy	71.88	160.74	95.02
Δ Avg Change	0	64.69	17.52

Password Entropy Averages across Groups (Higher Entropy is Better)

Final Questions

At the end of the study, participants were asked to partake in an end-of-study survey. The end-of-study survey was a brief, 4 question survey designed to gather additional data that can only be provided by participants themselves. The objective of these end questions was to determine what other factors outside of the control of this study may have played a factor in the resulting data. The survey was sent out via email to all participants at the end of the user testing period. The four questions were:

1. At any time during the study, did you have to request a password reset because you had forgotten your password?
2. At any time during the study, did you repeat a password you had used previously in the study?
3. At any time during the study, did two or more passwords you used follow a predictable pattern, such as "Password1, Password2", "Password-A, Password-B", or any other similar mechanism?
4. If you answered "Yes" to question 3, please provide a brief description of what sort of pattern was used, without entering any actual password value.

19 participants, spread nearly evenly across the three user participation groups, completed this self-report survey. 7 participants from the "Control" group (who did not have any password change requirement)

completed the survey, along with 7 from the “Every-Other-Friday” group and 5 from the “Every-Friday” group. Of the responses, the following data was generated:

Table 2. Survey Findings

ID	Use Password Manager?	Repeat a Password?	Use Predictable Pattern?	GROUP
8	No	Yes	Yes	Group A
9	No	No	No	Group B
11	No	No	No	Group A
12	No	No	No	Group A
13	Yes	No	No	Group C
21	Yes	No	No	Group C
22	No	No	No	Group A
26	No	No	No	Group B
27	Yes	No	Yes	Group B
28	No	No	No	Group A
30	Yes	No	No	Group C
31	Yes	No	No	Group C
34	No	No	No	Group B
38	Yes	No	Yes	Group C
42	No	No	Yes	Group B
47	No	No	No	Group A
48	No	No	Yes	Group C
49	No	Yes	Yes	Group C
50	No	No	No	Group A

Findings and Discussions

Patterns of Entropy

During the course of the study, no group could be found to have an overall group decrease in average password entropy, once participants who did not complete the survey had been removed from the data pool. Interestingly, both groups with mandatory password changes experienced a significant increase in average group entropy. Of the participants who completed the majority or all of the study, every single completing user in the Every-Friday change group (9), and 7 of the 9 completing participants in the Every-Friday group experienced either no overall change in entropy values for chosen passwords during the timeframe or experienced an increase in the entropy levels for their chosen passwords. This finding does not support the NIST assertion that "[u]sers tend to choose weaker memorized secrets" if the passwords are only temporary (NIST, 2020). It is possible that needing to create more passwords meant that more thought was given to security aspects of passwords, or this could be a fluke. In any case, due to small sample size, this would require additional study in order to fully determine or replicate.

Patterns of Tool Usage

Of the 19 people who completed this final survey, only 10.5% (2 participants) admitted to reusing a password multiple times during the course of the study. Additionally, 31.6% (6 participants) reported using a pattern or predictable generation method for their passwords. Surprisingly, 31.6% (6 participants) self-reported using a third-party password manager as part of their process, despite not being instructed in the use of password managers or encouraged for or against using them as part of the study. This observation suggests that password management software, with the ability to generate completely random passwords, is becoming more prevalent in daily use for some users. Previous surveys have shown that only "only 20%" of people use password managers, in contrast to the nearly 1/3rd of users who self-reported using these tools in this study (Whitney, 2021, Security.org, 2021). However, it is possible this elevated use of password managers was simply due to the nature of participants in the study, and heavy recruitment efforts in circles where internet literacy is common. It is also noteworthy that very few users will report reusing passwords multiple times for the same service.

Conclusions

From the data generated by this study, a few notable data points can be drawn, and some new questions are raised. Firstly, it is interesting to note that in no group did the average password entropy drop over time. It is unclear why this is, and so further research is needed to validate these results. However, perhaps more notable were the responses to the end survey. The unprompted, native adoption by nearly a third of participants of password management software is noteworthy and shows the rise in familiarity with these sorts of tools. Users who self-reported using password management software routinely had exceptionally little fluctuation in password-to-password entropy changes. However, it is also true that nearly another third of participants are still using predictable patterns in their passwords that reduce the theoretical security of passwords regardless of the technical entropy number. With patterned passwords, if a single password is known, and the pattern is discerned, the likelihood of being able to guess a current password is exceptionally more likely. It is also noteworthy that almost 90% of users did not repeat a password during the study, indicating that previous education and warnings about reusing passwords for the same service were a security risk that may have been heeded. It is unclear, however, exactly what deterred users from reusing passwords.

Challenges and Limitations for the Research

Unfortunately, there were some obstacles that presented themselves that complicated the research. While the study format itself shows promise, there were issues that hampered the effectiveness of the study itself. During the course of the study, over 21 participants (out of an initial pool of 49) either withdrew from the study or stopped logging in or responding to study email prompts. This dropout rate was evenly distributed among the majority of the participants, as indicated by the spread of responses to the final survey. This indicates potentially that without an incentive to continue, as no direct incentive or lottery reward system was included in this study, participants grew bored or distracted and did not have sufficient incentive to participate and complete the study.

While efforts were made to minimize the burden on individuals, it was a commitment of multiple weeks of time to see the study through to the end. In addition, some users did not log in on the days the emails went out, which made data collection and collation difficult at times. This, combined with a small sample size of users who completed the survey, makes the data vulnerable to outliers which may skew the results. This study must be considered proof of concept only, as there is not enough data to draw any statistically valid conclusions.

Practical Implications

If validated by further research, the results of this study indicate that password change requirements are in fact a useful and legitimate, if somewhat annoying to end users, method to help an organization maintain a level of digital security as part of a comprehensive security framework. For organizations and individuals, the results indicate that password literacy is still an important topic, and that password rotations, education on avoiding patterns, and further reflection on how secure an individual's passwords currently in use truly are. For researchers, these results show that practical, secure research on passwords is possible, and the field can still be studied without exposing participants to additional risk or insecure methodology.

Future Research

This study provides a framework for future research. However, the results of the study indicate that some changes need to be made in future attempts at similar research as well as raise new questions. If at all possible, an incentive to participants to see the study through to the conclusion should be offered. This incentive might come in the form of a lottery system for a financial reward, an evenly distributed lump sum reward for completion, or some other method. Possible incentives would be dependent on the particular rules and guidelines established by the researchers and the local Institutional Review Board panel. This study had a significant dropout rate, and an incentive to keep participants active in the project may increase the data collection rate per enrolled participant. Additionally, the number of participants in this study was far from a statistically significant number. Further replication studies with larger numbers of participants are required to generate information, while also preferably spreading out the timescale of the study over a longer period of time.

Furthermore, a number of technical improvements to the study template are recommended. The study suffered from a lack of automation, where password change mandates had to be manually set by the study researchers, as well as manually triggering the emails to be sent to users. In future research, an automated, predictable system should be employed to handle this type of task with pinpoint accuracy on timing. Due to the requirements of a manual system, the time that password email reminders were sent varied from week to week, often by several hours. With the addition of these minor technical improvements, the performance of the study would likely be significantly improved, and lead to more successful attempts in the future.

References

Al-Ameen, M. N., Wright, M., & Scielzo, S. (2015). Towards making random passwords memorable: Leveraging users' cognitive ability through multiple cues. *Conference on Human Factors in Computing Systems - Proceedings*, *2015-April*, 2315–2324. https://doi.org/10.1145/2702123.2702241

Brown, D. (2022). *UserSpice – Open Source PHP Management Framework*. https://userspice.com/

Digital Identity Guidelines: Authentication and Lifecycle Management. (2017). *Special Publication (NIST SP) - 800-63B*, 5.1.1.2. https://doi.org/10.6028/nist.sp.800-63b

Egelman, S., Sotirakopoulos, A., Muslukhov, I., Beznosov, K., & Herley, C. (2013). Does my password go up to eleven? the impact of password meters on password selection. *Conference on Human Factors in Computing Systems - Proceedings*, 2379–2388. https://doi.org/10.1145/2470654.2481329

Fagan, M., Albayram, Y., Khan, M. M. H., & Buck, R. (2017). An investigation into users' considerations towards using password managers. *Human-Centric Computing and Information Sciences*, *7*(1). https://doi.org/10.1186/s13673-017-0093-6

Haskell-Dowland, P., & O'Shea, B. (2020, September 15). *A computer can guess more than 100,000,000,000 passwords per second. Still think yours is secure? - GCN*. https://gcn.com/cybersecurity/2020/09/a-computer-can-guess-more-than-100000000000-passwords-per-second-still-think-yours-is-secure/315643/

Herley, C., & Van Oorschot, P. (2012). A research agenda acknowledging the persistence of passwords. *IEEE Security and Privacy*, *10*(1), 28–36. https://doi.org/10.1109/MSP.2011.150

How to Calculate Password Entropy? - Password Generator. (n.d.). Retrieved April 8, 2021, from https://generatepasswords.org/how-to-calculate-entropy/

Mazurek, M. L., Komanduri, S., Vidas, T., Bauer, L., Christin, N., Cranor, L. F., Kelley, P. G., Shay, R., & Ur, B. (2013). Measuring password guessability for an entire university. In *Proceedings of the ACM Conference on Computer and Communications Security*. https://doi.org/10.1145/2508859.2516726

NIST. (2020). *NIST SP 800-63 Digital Identity Guidelines*. National Institute of Standards and Technology. https://pages.nist.gov/800-63-FAQ/

Password Manager and Vault 2021 Annual Report: Usage, Awareness, and Market Size. (2021). Security.Org. https://www.security.org/digital-safety/password-manager-annual-report/

Pleacher, D. (n.d.). *Password Entropy*. Retrieved April 8, 2021, from https://www.pleacher.com/mp/mlessons/algebra/entropy.html

Shay, R., & Bertino, E. (2009). A comprehensive simulation tool for the analysis of password policies. *International Journal of Information Security*, *8*(4), 275–289. https://doi.org/10.1007/s10207-009-0084-3

Shay, R., Komanduri, S., Durity, A. L., Huh, P., Mazurek, M. L., Segreti, S. M., Ur, B., Bauer, L., Christin, N., & Cranor, L. F. (2016). Designing password policies for strength and usability. *ACM Transactions on Information and System Security*, *18*(4), 1–34. https://doi.org/10.1145/2891411

Summers, W. C., & Bosworth, E. (2004). *Password Policy: The Good, The Bad, and The Ugly*. https://doi.org/10.5555/984720

Ur, B., Kelley, P. G., Komanduri, S., Lee, J., Maass, M., Mazurek, M., Passaro, T., Shay, R., Vidas, T., Bauer, L., Christin, N., & Cranor, L. F. (2012). *Helping Users Create Better Passwords*. https://lorrie.cranor.org/pubs/login1212_ur.pdf

Weir, M., Aggarwal, S., Collins, M., & Stern, H. (2010). Testing metrics for password creation policies by attacking large sets of revealed passwords. *Proceedings of the ACM Conference on Computer and Communications Security*, 162–175. https://doi.org/10.1145/1866307.1866327

Whitney, L. (2021). *How and why people use password managers | TechRepublic*. https://www.techrepublic.com/article/how-and-why-people-use-password-managers/

Wiedenbeck, S., Waters, J., Birget, J. C., Brodskiy, A., & Memon, N. (2005). Authentication using graphical passwords: Effects of tolerance and image choice. *ACM International Conference Proceeding Series*, *93*, 1–12. https://doi.org/10.1145/1073001.1073002