Password Strength Meters

DAVID CHEN, Oregon State University, USA

ABDULLAH SAYDEMIR, Oregon State University, USA

LINDY VOSS, Oregon State University, USA

Passwords are integrated into our everyday lives, whether it be accessing social media accounts, logging onto computers and unlocking our phones, or protecting our financial/banking information. Despite how important they are, people still create and use weak passwords that can easily be cracked by hackers for malicious intent. In this article, we looked into entropy calculations, different cracking techniques to better understand how passwords are cracked, and introduce several better password creation. We also demonstrated ways to make those strong passwords more memorable. With these methods, it will be easier to create a password that is less likely to be cracked without compromising the memorability. We conclude the paper by analyzing the scalability of the methods and suggest a solution to password management issue considering the fact that reusing the same strong password will cause issues.

CCS Concepts: • **Security and privacy → Usability in security and privacy**; *Authentication*; **Human and societal aspects of security and privacy**; Cryptanalysis and other attacks.

Additional Key Words and Phrases: security, password, memorability

1 INTRODUCTION

Passwords are used nowadays as the first line of defense to protect information along with other options [17]. Though passwords are meant to be solid and secure, they have flaws that can be easily exploited. In 2019, nearly 1 terabyte of user account data including passwords previously collected from Yahoo and some other website breaches were for sale [12]. Hashes of mostly used passwords, including those from Yahoo!, get uploaded to haveibeenpwned.com for the public to check whether their passwords were breached. Among 700 million unique passwords, two of the most commonly used were "123456" (more than 24 million times) and "password" (more than 3 million times) which can be broken in virtually zero time.

To prevent these from happening, the websites benefit from password composition rules (e.g. password must contain both lowercase and uppercase letters). Studies show that enforcing these rules helps users to choose stronger passwords [22]. However, this also increases user fatigue since the average number of website accounts a single user has is around 25 [16, 28] and continues to increase. This exhaustion leads to people creating one (or a few) strong passwords, often reusing those passwords across many websites [7] and preserving them for long periods [4, 9]. Moreover, only one third of people change their password after a breach and most of them change their passwords to weaker or equally strong ones [4]. This leads to large collections of passwords (that are gathered on the internet) being used in brute force attacks over and over again, often with some success due to all of the unchanged passwords. There are also highly specific versions of dictionaries used in these attacks that are categorized according to attack type, platform, website, language and even to geological location.

Authors' addresses: David Chen, chend2@oregonstate.edu, Oregon State University, Corvallis, Oregon, USA, 97331; Abdullah Saydemir, saydemia@oregonstate.edu, Oregon State University, Corvallis, Oregon, USA, 97331; Lindy Voss, vossli@oregonstate.edu, Oregon State University, Corvallis, Oregon, USA, 97331.

The problem with strong and secure passwords is nobody can remember them. Especially with the amount of websites and accounts people hold nowadays. People are told to create strong passwords and in order to do that they have to avoid short passwords, birthdays, names, places, dictionary words, and the list goes on [16]. Basically you are not supposed to use anything remotely memorable. Say you come up with a strong password like Jf5ruEf392HQlx and somehow memorize it. Well you can only use that for one account and the rest all have to be different but just as strong [20]. This begs the big question of how do you create a strong password (for each account) that is realistic to memorize for practical use? There are tools and different tricks that have been found to work to help people balance these two very important factors when deciding on a password so they can not only remember their passwords, but also keep their information safe.

Therefore, in this article, we propose the following contributions:

- We will have a look at password entropy calculation to understand what dictates the strength of passwords.
- We introduce different password cracking techniques used to understand how passwords are being solved to better adapt new passwords to prevent cracking
- We introduce password creation techniques and guidelines, and discuss their strengths in relation to password entropy.
- We introduce memorable passwords that are still strong to resist brute force / dictionary attacks.

In the next section, we will present an overview of concepts that are helpful to understand how a strong password is created.

2 CONCEPTS

2.1 Password

Password, common name for the term *Memorized Secret Authenticator*, is a secret key that the user randomly selects and memorizes [10]. Passwords must have enough complexity and secrecy that an adversary would be unable to guess or discover the secret value. If the password consists only of numbers it can be referred as PIN.

There are two other important definitions. *Neutral Passwords* are those do not contain any information about the user whereas *Biographic Passwords* are those that do contain information about the user [15].

2.2 Passphrase

Passphrase is a memorized secret that consists of sequences of words or phrases. It is usually longer than password to add more security [1].

2.3 Password Entropy / Strength

Password entropy is a measurement how strong a password is. It is used to estimate effectiveness of a password against guessing or brute-force attacks. Higher entropy means stronger passwords whereas lower entropy means weaker. It depends on the complexity and length [11].

To put it in mathematical way, let's say P is a password of length l and consists of characters in alphabet Σ size of which is equal to N. That is, $N = |\Sigma|$. Then, the entropy E of this password P is calculated with the following equation :

$$E = \log_2 N^l = l \cdot \log_2 N \tag{1}$$

Character Set	Elements	Set Size
Digits	0-9	10
Latin letters (lowercase)	a-z	26
Latin letters (uppercase)	A-Z	26
Alphanumeric	a-z ∪ 0-9	36
Alphanumeric and Uppercase	a-z ∪ A-Z ∪ 0-9	62
Symbols (US keyboard)	!@#$%& ∗ () − + = []{}|; etc.	32

Table 1. Most Used Password Sets

We can directly deduce that increasing either of L or N creates passwords that have higher entropy. Therefore, to create a stronger password user should either choose a larger set of characters or make the password longer. It is obvious that for some specific numbers, entropy of longer password from a smaller set can be similar to entropy of the shorter password from a larger set. For example, let P_1 a password that consists of digits and its length is 11. Let P_2 another password that consists of lowercase Latin letters and its length is 8. If we calculate the entropy of both using set sizes given in Table 1:

$$E_1 = \log_2 10^{11} = 11 \cdot \log_2 10 = 11 \times 3.32 \qquad E_1 \approx 36.5$$

$$E_2 = \log_2 26^8 = 8 \cdot \log_2 26 = 8 \times 4.70 \qquad E_2 \approx 37.6$$

we can see that there is not much difference between the entropy of the two passwords. However, it is important to note that entropy itself is **not** all it matters. Consider the same passwords P_1 and P_2 but this time let P_1 be the string "46583159027" and P_2 be the string "password". In this case, P_2 is extremely weak and it can be broken in virtually zero time since it appears in leaked password lists.

Therefore, aside from the two parameters of Equation 1, *unpredictability* is another factor that has an effect on password strength. We will talk about this concept in detail in section 3.

Some adjustments can be made on password entropy calculation by making assumptions on the strength of the attackers. In this article, we will use the Equation 1 without any adjustments since assumptions may or may not hold true for each case.

2.4 Password Cracking-Techniques

In order to know how to to best create a strong memorable password, it is important to know how many passwords are cracked. There are many different types of techniques that a hacker can use to figure out one's passwords. Each technique share the same end goal, but their execution differs. These techniques fall under three different categories of cracking: password-focused, human-focused, and computer focused. This document will only concern the password-focused techniques. These technique focuses on the password themselves, cracking the characters or the words used in each user's password.

Different techniques have different success rates and execution difficulty. For example, phishing is a technique where a user is tricked into clicking or downloading a malicious file, where malware is executed. Normally, these disguise themselves as emails telling the user to take action of some sort [26]. This technique is difficult to execute because it is based off the user's ignorance to work, but the times that it does the hacker would be able to easily get passwords and other information from the user.

For the password-focused category, they generally contain techniques that aren't too difficult to achieve, but have a lower chance of success than techniques in the other categories. It also has more tools that average people can download and use themselves to crack passwords. There are many tools out there. *Hashcat* is one example of these tools. Hashcat is a tool that does one of the techniques that will be mentioned below, and is available to download to the general public. It can be used both licitly and illicitly, such as a system admin checking the security of their client's passwords, or a hacker moving through a system and gaining admin privileges [18]. The following are some examples of password-focused cracking techniques.

2.4.1 Brute Force Attack. Brute Force is the most common and simplest technique used by hackers. It is a technique where an algorithm is used to try as many possible keyword and password combinations to gain access to the system and/or file [14]. These generally work best on simpler passwords, such as "*password*", "*123456*", "*qwerty*", and possible iterations, "*p@ssword*", "*PaSsWoRd*", etc.

2.4.2 Dictionary Attack. Dictionary attacks are a type of brute force attacks, except it heavily relies on *cracking dictionaries*, which are lists that contain the most common passwords, word combinations, and dictionary words [26]. They may also contain credentials from previous hacks in other systems.

2.4.3 Mask Attack. Mask attacks are similar to the standard brute force attacks, except they rely on scanning through specific characters and letters in a specific pattern. For example, if a hacker knows that the password is an 8 character password that starts with an uppercase letter, followed by three lowercase letters, where the last four characters will be a number or a special character, then he just needs to set up an algorithm that scans all the characters needed in that specific arrangement [13].

However, the drawback of brute force and dictionary/masks attacks is that the time to crack the password is based off the length and complexity of the password, which relates directly to entropy. As stated in 2.3, it helps estimate the effectiveness of the password against brute force attacks. The higher the entropy of a password, the longer it takes for the algorithm to be able to crack it. It's possible for some of the stronger passwords to take years to crack, depending on it's length and complexity.

3 PASSWORD CREATION TECHNIQUES

We know what contributes to password strength and how adversaries try to crack passwords. Hence, we can use these knowledge to create strong passwords. However, we need to know the essential requirements, dos and don'ts.

According to the latest NIST guidelines [10], a password must consist of at least 8 characters and users should be encouraged to use as lengthy password as possible, within reason. On the other side passwords should not include the following items:

- Passwords that are inside leaked password lists e.g. "password"
- Dictionary words e.g. "!-summer?5"
- Repetitive or sequential characters e.g. "aaaaaaaa" or "abcd1234"
- Context-specific words e.g. name of the service, username or derivatives

Having these in mind, let's see couple of methods to create strong passwords.

3.1 Strong Passwords

3.1.1 Traditional Method. There are several studies that show enforcing or suggesting password composition rules increase the password strength [21, 22]. Services usually provide compositions rules and often enforce the user to create a password that comply with these rules. The suggestions vary across services but most common ones are as follows:

- Password length should be at least 12
- Password should include uppercase/lowercase letters, symbols and numbers
- Password should not include obvious substitutions $S \longrightarrow \$$

Prompted with these, what most users do is creating a sequence of string (meaningful or meaningless) and further substituting some symbols/numbers if required. Though this method, Traditional Method, is one of the most primitive methods used in password composition, it is still capable of offering strong passwords.

For example, let "*latexcoroutine*" be the base string. A possible symbol and number substitution would be "$L/-\backslash t[-;CO2ine$". Therefore, the entropy of the password would be:

$$E = \log_2 94^{14} = 14 \cdot \log_2 94 = 14 \times 6.55 \qquad E \approx 91.7$$

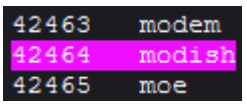
3.1.2 Passphrase Method. As we talked in subsection 2.2, passphrases are lengthy passwords that consist of multiple words or phrases. Equation 1 specifies that increasing length of the password directly contributes to password entropy; therefore, increases the password strength. Furthermore, NIST suggests using passwords as lengthy as possible. Passphrase method tries to achieve the highest password strength by greedily increasing the length.

Words that form the passphrase must be random, varied, and the passphrase must be neutral (i.e. non-biographic) [10, 15]. To accommodate the random word selection, Diceware [19] offers a method and multiple word lists consisting of more than 7000 words. Random passphrase creation method is as follows:

1. Roll a die five times (or five dices at once) and record the numbers

   4 2 4 6 4

2. Concatenate the result and find the corresponding word for your number in the word list

3. Do this until you have sufficiently long passphrase (5-word is recommended)

   42464-63262-62165-22365-16432
   modish-voss-turk-david-chen

There are other word lists proposed by Diceware or other platforms. By keeping the randomness of the procedure, this method can be applied to other word lists as well. Furthermore, a secure random number generator could be used in order to create the random numbers faster.

One immediate drawback of the passphrase method is that NIST guidelines confront using dictionary words in a clear text format. Letter/number or letter/symbol substitutions should be made to avoid any possible password defects. However, know that common symbol substitutions such as $S \longrightarrow \$$ does not increase the entropy [25].

Let above passphrase "*modishvossturkdavidchen*" be the base string. A possible capitalization (no substitution) would be "*modIshvosSturKdavIdcheN*". Therefore, the entropy of the password would be:

$$E = \log_2 52^{23} = 23 \cdot \log_2 52 = 23 \times 5.70 \qquad E \approx 131.1$$

Notice that the entropy would be still high even if the passphrase was lowercase Latin characters only.

$$E_{base} = \log_2 26^{23} = 23 \cdot \log_2 26 = 23 \times 4.70 \qquad E \approx 108.1$$

3.2 Memorable Passwords

The most secure passwords you could possible use would be a random string of letters, capitals, lowercase, numbers, and symbols that is as long as allowed [10, 16]. Those types of passwords, as secure as they may be, are nearly impossible to remember with the human memory, not to mention multiple of them. The human memory can only hold a sequence of about seven items and they cannot be arbitrary and should be redundant [29]. That is the exact opposite of what a password is supposed to be. Creating passwords that are secure, yet memorable becomes very tricky. In order to do so, it is necessary to scale back the randomness, definitely the length, and therefore the secureness just enough, so the human memory has a chance of memorizing its passwords. You still want to preserve the security as much as possible though. It's a balance that is hard to find, especially with each and every account password you have. Many are unable to find that balance and do the suggested security measures. For example, only 8 percent of users don't reuse passwords [16].

The easiest way to create a memorable password is to use something important to you or something that you like. Just using the names, dates, and other words is not secure though [20]. The best password you can create is the one you cannot remember which is why this is such a dilemma [16]. Human memory, as it turns out, is the biggest flaw of passwords [29]. So, we have to find a way to balance both of them. There are multiple techniques, tools, and rules that can be used to help like using something important as we mentioned or using outside management tools.

First, the rules to follow are the simplest way to increase the unbreakability without increasing the difficulty of memorability. Lengthening passwords is probably the easiest yet biggest thing that can increase security. Lengthening a password can be done by just adding some padding to the end. Make sure when you add padding though it is not just '!!!' as that is common and easy to guess [20]. Try adding 2 random alternating characters. According to a calculator for time to brute force Steve Gibson creating 'helloworld!!!' would take 33 years to crack but adding 'qwqwqwqw' puts it at over 49 trillion centuries [20]. This way it has that element of random and the length, but it is not impossible to remember.

Another major rule to follow is avoiding names, places, and other common dictionary words [16]. Even if there is not a lot of variety in type of characters in the password but is just not words and random characters instead, it will become more secure. This is because doing this takes dictionary attacks out of the equation forcing them to use a brute force. [20]. Dictionary attacks crack about two-thirds of all passwords by using words and common patterns to guess like two numbers after

a word or replacing letters for similar looking numbers [16]. Recent password breach reports has shown that some of the most common passwords (that you should avoid using) are these listed below [3]:

- 123456
- 12345678
- 123123
- 1234
- 654321
- 123456789
- qwerty
- 1234567890
- azerty
- 1q2w3e4r5t
- password
- 1234567
- 000000
- iloveyou
- qwertyuiop
- admin
- 12345678
- abc123
- aaaaaa
- 111111

The dictionaries the attackers are using are almost guaranteed to contain these passwords, which is why forcing them to use a brute force attack is more ideal. Brute force attacks allow you to control your security much closer as they take more time to execute, and you are able to control the time they take to execute more directly with factors like character variation and length.

Adding in a variation of letters, capitalization, numbers, and symbols is also a good idea as well. This can exponentially increase the time it will take to crack as password just about as much as increasing length does even if the password is memorable and considered not strong according to typical password creation rules[16]. For example, according to Gibson's calculator, "*hellosworlds*" would take 16.54 minutes to crack as well as most combinations of 12 lowercase letters, whereas "*HeLlOwOrLd7?*" would take 1.74 centuries to crack [20]. The simplest change can make a very big difference in your information's security even if you are sticking to the unsuggested memorable passwords.

So if we want to not use words and be random as suggested by these rules, how do we remember those passwords? Something completely random with no familiarization will be hard to remember. You can still create a password the appears completely random to an unsuspecting person, but has a pattern or meaning that the brain can attach to it so it can recall. As mentioned before use something meaningful. If you have a favorite movie, story, book, poem, song or even just a phrase, use that to your advantage [16, 20]. One of the many names for this method is the Bruce Schneier Method [16]. This is when you take the first letter of each syllable (capitalized if it is the start of the word as well, otherwise lowercase) and the punctuation and abbreviate it into a random seeming password. For example, if someones favorite song is Carry on Wayward Son by Kansas they might use the chorus of that song. So "Carry on, my wayward son there'll be peace when you are done" would turn into "*CrO,MWwST'lBPWYAD*". This on its own appears completely random, but it is easier to remember because the brain can make a connection to something familiar.

Even if you do not want to use this specific method to help remember passwords, any type of abbreviated password creation method will help the brain remember more easily. You can take lines from a TV show and add the season, episode, time or any other aspect to help randomize it as well [20].

Some are unwilling to give up the security of a good password. For those people there are some ways those can still be memorable, although it is not as convenient as other listed methods. Using memorization techniques to remember passwords may be helpful. Things like mnemonic devices and phonetics in specific are proven methods [20]. One example is Person-Action-Object (PAO) method that mnemonist use to remember long random numbers or a deck of cards [2]. Its use in password creation and memorization is thoroughly researched by Carnegie Mellon University [5] and it's shown that this method is handy creating strong passwords.

Creating a password using PAO is pretty simple. Choose one person, one action, one object and create a sentence. Then, manipulate the string (add, remove or substitute characters) so that the password has enough entropy. For example, let's choose *Keanu Reeves* as the person, *swallowing* as the action and *wallet* as the object. Then the base string will be "*KeanuReevesswallowingwallet*" and a possible password is "*KyouR€westsWal1@*". If higher strength is needed than include more details such as time and place.

Another great memorization technique is phonetics. Trying to make words out of a password even if it is gibberish gives the brain a familiarity to hold on to and remember easier. For example if a generator outputs "*drEnaba5Et*" the phonetics might be pronounced as "Dr Enaba 5 E.T." [20]. Another example is "*orMSIZEfrH2O*" as "or M size for H2O". This may prove to be difficult with completely random passwords so it may take some attempts to find one that works well enough to be realistic to use.

There are lots of other little tips and tricks that can be done as well, but those were the best and most realistic ways to try to balance having the most secure password as possible and the limitations of the human memory.

4 RESULTS & ANALYSIS

We introduced several methods to create strong and memorable passwords. They are widely used and recommended but they are not foolproof. In this section, we will analyze introduced password creation methods.

Traditional method is the far most flexible and the fastest one. The strength of the password can be adjusted by changing, substituting and/or removing characters based on the needs and the service the password is used, since there is no pattern among the characters. However, this brings the question whether the created password is random. Researches show that, created passwords contain linguistic characteristics [6], previous password of the same user [4, 7, 27], leaked passwords [7] and personal information [23, 24] in most of the cases.

Passphrase method is rather safe than the traditional method considering that the words are picked from a neutral word list in a random way. Passphrases are also more memorable than the traditional passwords [8]. On the other hand, creating a passphrase takes longer time than the traditional one. There are websites offering randomly created passphrases (or words) but it is a security concern that if the generator remembers the created passphrases or if it matches the passphrases with a unique identity. Creating several passphrases and "randomly" selecting (or not selecting) words from each of them is the first thing that comes to mind. However, it takes time and we are back at where we started.

PAO, Bruce Schneier and some other methods come in handy since they put memorability first. They increase the ability of a user to remember multiple passwords. It is fast to create multiple passwords, they are also flexible in adjusting the strength and memorability. However, these methods also leave the content of the password to the user. Therefore, the problems Traditional Method suffer create valid concerns also for these methods. In a nutshell, there is no silver bullet for password creation.

5 CONTRIBUTIONS

- Lindy Voss : Introduction, Memorable Passwords (3.2), Results & Analysis, Conclusion, References
- Abdullah Saydemir : Introduction, Concepts (2.1 2.2 2.3), Strong Passwords (3.1), Results & Analysis
- David Chen : Abstract, Introduction, Concepts (2.4), References

6 CONCLUSION

We analyzed several password creation methods. Each method offers different ways to create passwords. Some helps us to create nearly unbreakable passwords while leaving memorability as a question, some takes different approach and puts memorability first. However, no method, including those that were not analyzed in this paper, provides both security and memorability to get a user to remember 25 different passwords. Therefore, we conclude the paper with the following suggestion.

Use an password manager to keep track of the passwords. There are robust applications on most of the platforms that provide this service besides some other useful features. Make sure that the application is secure and verified by other parties. Use one of the above methods to create single absolutely strong password and use it to secure the password manager itself. Enable two factor authentication if provided by the application.

REFERENCES

[1] [n.d.]. Passphrase - Glossary. https://csrc.nist.gov/glossary/term/Passphrase
[2] [n.d.]. Person-Action-Object (PAO) System. https://artofmemory.com/wiki/Person-Action-Object_(PAO)_System
[3] [n.d.]. Top 200 Most Common Passwords of 2020. https://nordpass.com/most-common-passwords-list/
[4] Sruti Bhagavatula, Lujo Bauer, and Apu Kapadia. 2020. (How) Do people change their passwords after a breach? (10 2020).
[5] Jeremiah Blocki, Manuel Blum, and Anupam Datta. 2013. Naturally Rehearsing Passwords. https://doi.org/10.1007/978-3-642-42045-0_19
[6] Kevin Curran, Jonathan Doherty, Ayleen McCann, and Gary Turkington. 2011. Good Practice for Strong Passwords. *EDPACS The EDP Audit* Control (11 2011), 1–13. https://doi.org/10.1080/07366981.2011.635497
[7] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and Xiaofeng Wang. 2014. The Tangled Web of Password Reuse. *Proceedings of NDSS 2014*. https://doi.org/10.14722/ndss.2014.23357
[8] Danyl Fernandes, Gandharv More, Monika Anabathula, and Philip Mathew. 2021. Passphrase Generation Using Diceware. (05 2021).
[9] Dinei Florencio and Cormac Herley. 2007. A large-scale study of web password habits. *16th International World Wide Web Conference, WWW2007*, 657–666. https://doi.org/10.1145/1242572.1242661
[10] Paul Grassi, Elaine Newton, Ray Perlner, Andrew Regenscheid, Jim Fenton, William Burr, Justin Richer, Naomi Lefkovitz, Jamie Danker, Yee-Yin Choong, Kristin Greene, and Mary Theofanos. 2017. Digital Identity Guidelines: Authentication and Lifecycle Management. https://doi.org/10.6028/NIST.SP.800-63b
[11] Gongzhu Hu. 2018. *On Password Strength: A Survey and Analysis*. 165–186. https://doi.org/10.1007/978-3-319-62048-0_12
[12] Troy Hunt. 2019. Pwned Passwords, Version 5. https://www.troyhunt.com/pwned-passwords-version-5/
[13] Jake. 2018. Hashcat Tutorial – The basics of cracking passwords with hashcat. https://laconicwolf.com/2018/09/29/hashcat-tutorial-the-basics-of-cracking-passwords-with-hashcat/
[14] Kaspersky. 2021. Brute Force Attack: Definition and Examples. https://www.kaspersky.com/resource-center/definitions/brute-force-attack
[15] Joakim Kävrestad, Fredrik Eriksson, and Marcus Nohlberg. 2019. Understanding passwords – a taxonomy of password creation strategies. *Information and Computer Security* 27 (06 2019). https://doi.org/10.1108/ICS-06-2018-0077
[16] Kevan Lee. 2014. Four Methods to Create a Secure Password You'll Actually Remember. https://lifehacker.com/four-methods-to-create-a-secure-password-youll-actually-1601854240

[17] Wanli Ma, John Campbell, Dat Tran, and Dale Kleeman. 2010. Password Entropy and Password Quality. *Proceedings - 2010 4th International Conference on Network and System Security, NSS 2010*, 583–587. https://doi.org/10.1109/NSS.2010.18

[18] J.M. Porup. 2020. Hashcat explained: How this password cracker works. https://www.csoonline.com/article/3542630/hashcat-explained-why-you-might-need-this-password-cracker.html

[19] Arnold G Reinhold. 2021. The Diceware Passphrase Home Page. https://theworld.com/~reinhold/diceware.html

[20] Neil J. Rubenking. 2021. Simple Tricks to Remember Seriously Secure Passwords. http://www.pcmag.com/how-to/simple-tricks-to-remember-seriously-secure-passwords

[21] Richard Shay, Lujo Bauer, Nicolas Christin, Lorrie Cranor, Alain Forget, Saranga Komanduri, Michelle Mazurek, William Melicher, Sean Segreti, and Blase Ur. 2015. A Spoonful of Sugar?: The Impact of Guidance and Feedback on Password-Creation Behavior. https://doi.org/10.1145/2702123.2702586

[22] Richard Shay, Saranga Komanduri, Patrick Kelley, Pedro Leon, Michelle Mazurek, Lujo Bauer, Nicolas Christin, and Lorrie Cranor. 2010. Encountering Stronger Password Requirements: User Attitudes and Behaviors. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/1837110.1837113

[23] Keng Siau, Yizhi Ma, and Nathan Twyman. 2018. Cybersecurity: Personal Information and Password Setup.

[24] N. Tulek, Müge Kuşkon, Idil Sezgin, and Albert Levi. 2020. Disclosure of Personal Information in Passwords on Social Media. 1–4. https://doi.org/10.1109/SIU49456.2020.9302085

[25] Blase Ur, Fumiko Noma, Jonathan Bees, Sean Segreti, Richard Shay, Lujo Bauer, Nicolas Christin, and Lorrie Cranor. 2015. "I added '!' at the end to make it secure": Observing password creation in the lab.

[26] Dale Walker. 2020. The top 12 password-cracking techniques used by hackers. https://www.itpro.com/security/34616/the-top-password-cracking-techniques-used-by-hackers

[27] Rick Wash, Emilee Rader, Ruthie Berman, and Zac Wellmer. 2016. Understanding Password Choices: How Frequently Entered Passwords Are Re-Used across Websites. In *Proceedings of the Twelfth USENIX Conference on Usable Privacy and Security* (Denver, CO, USA) *(SOUPS '16)*. USENIX Association, USA, 175–188.

[28] Rob Waugh. 2012. No wonder hackers have it easy: Most of us now have 26 different online accounts - but only five passwords. http://www.dailymail.co.uk/sciencetech/article-2174274/No-wonder-hackers-easy-Most-26-different-online-accounts--passwords.html

[29] Jianxin Yan, Alan Blackwell, Ross Anderson, and Alasdair Grant. 2000. The Memorability and Security of Passwords - Some Empirical Results. (09 2000).