Forensic Science International: Digital Investigation 37 (2021) 301186

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

DFRWS 2021 USA - Proceedings of the Twenty First Annual DFRWS USA

How viable is password cracking in digital forensic investigation? Analyzing the guessability of over 3.9 billion real-world accounts

Aikaterini Kanta [a, b, *], Sein Coray [c], Iwen Coisel [b], Mark Scanlon [a]

[a] Forensics and Security Research Group, University College Dublin, Ireland
[b] European Commission, Joint Research Centre (DG JRC), Via Enrico Fermi 2749, 21027, Ispra, VA, Italy
[c] Databases and Information Systems Group, Department of Mathematics and Computer Science, University of Basel, Switzerland

ARTICLE INFO

Article history:

Keywords:
Password security
Password-based authentication
Context-based password cracking
Password strength meters

ABSTRACT

Passwords have been and still remain the most common method of authentication in computer systems. These systems are therefore privileged targets of attackers, and the number of data breaches in the last few years attests to that. A detailed analysis of such data can provide insight on password trends and patterns users follow when they create a password. To this end, this paper presents the largest and most comprehensive analysis of real-world passwords to date – associated with over 3.9 billion accounts from Have I Been Pwned. This analysis includes statistics on use and most common patterns found in passwords and innovates with a breakdown of the constituent fragments that make each password. Furthermore, a classification of these fragments according to their semantic meaning, provides insight on the role of context in password selection. Finally, we provide an in-depth analysis on the guessability of these real-world passwords.

1. Introduction

Passwords are everywhere. The average number of password-protected services and devices per user is a difficult figure to estimate. Often, the users themselves fail at evaluating their own digital environment/footprint. The only certainty is that this number is growing over time as the world is experiencing its ongoing digital transformation (Kanta et al., 2020a). This societal phenomenon is mostly technologically driven and is safe to assume will continue into the future with autonomous driving, remote surgery, smart homes, etc.

As much as a password is the barrier for attackers to breach a critical system, it is equally a hindrance for law enforcement conducting their investigations. In the context of digital forensic investigation, the use of password protected accounts and devices can present a hurdle for their lawful analysis under warrant (Thing and Ying, 2009). The prevalence of password protected encrypted storage coupled with increasingly stringent password policies can result in cases being significantly held up in the best case or hitting a dead end in the worse case. The mainstream desktop computer and mobile operating systems offer built-in password protected encryption of their storage volumes (Sayakkara et al., 2019). Such feature is enabled by default in many cases without user configuration (Du et al., 2020).

Law enforcement agencies throughout the world are struggling to keep up with the demand for digital forensic investigation – with multi-year, case hindering backlogs becoming commonplace (Lillis et al., 2016). When time is of the essence both from a time-sensitive case (e.g., child abuse investigation, human trafficking, etc.) and investigative efficacy perspective, the decisions made on what resources to allocate to each digital investigation can be crucial (Kanta et al., 2020b). Attempting to brute force a password for each account, encrypted file or storage volume can use a large amount of resources with no guarantee of success in any reasonable time frame (Du et al., 2020). In order to enable digital investigators to make informed decisions on the likelihood of success for this approach, actionable statistics are needed based on a significantly large dataset of real-world passwords. This forms the motivation for the work presented in this paper.

1.1. Contribution of this work

In order to address this consideration, we have conducted the largest password analysis to date considering a list of 555,278,657

* Corresponding author. Forensics and Security Research Group, University College Dublin, Ireland.
E-mail address: aikaterini.kanta@ucdconnect.ie (A. Kanta).

https://doi.org/10.1016/j.fsidi.2021.301186

unique passwords from the *Have I Been Pwned?*[1] version 5 dataset (HIBP_v5). Of these, 515,680,539 passwords were reverse engineered correlating to 3,951,907,330 real-world accounts – with password reuse accounting for the disparity in these numbers (i.e., commonly used passwords both between different accounts and different users). The contribution of this work can be summarized as follows:

1. The largest and most comprehensive analysis of real-world passwords conducted to date. The HIBP_v5 dataset was used to extract the underlying statistics of the constituent passwords, showcasing password tendencies of real users.
2. An analysis of the passwords' pattern of construction after splitting the passwords into meaningful component fragments. This analysis reveals that some semantic classes are more common than others, underlining the potential importance of user context when they select their passwords.
3. Finally, the passwords are clustered based on their presumed strength and their crackability is assessed. The analysis identifies the strongest class of passwords and under certain conditions, it is demonstrated that some of these should still be considered weak.

2. Background

Password strength evaluation is of particular interest in the last few decades. It has been done through the analysis of existing leaked datasets or through studies where participants answer questions about their password habits (Brown et al., 2004; Bonneau, 2012; Mazurek et al., 2013; Galbally et al., 2017; Golla and Dürmuth, 2018).

2.1. Influencing factors on password selection

According to many studies, a person's background contributes to their password choices (Kanta et al., 2020a). In a survey of college students about their password habits, Brown et al. (Brown et al. (2004) showed in 2004 that most students use personal information in their password selection, such as their own birthdays and names as well as those of their friends and relatives. In fact, knowing the email and username associated with an account can facilitate retrieving the password of that account (Ji et al., 2015). This is further reinforced by looking at authentication error correction schemes, where Chen et al. (2019) reported that on average targeted error correction is twice as successful as non-targeted error correction. Furthermore, a user's beliefs and often misconceptions about what makes a password secure can also explain the rationale behind their password choices. In 2015, Ur et al. (2015) conducted a study where participants had to choose passwords for specific websites and they noted that many users erroneously believed that words that are difficult to spell are harder to crack, or that adding a '*!*' at the end of a passwords adds to its security.

Another factor that influences password choices of users is demographics factors including age, gender and nationality. In 2012, Bonneau (2012) analyzed a corpus of 70 million passwords and looked at whether the guessability of these passwords changed when specific dictionaries targeting each demographic were chosen. It was observed that the success rate of a guessing attack with 1000 guesses, when a specified-to-a-particular-demographic dictionary was chosen performed slightly better in categories that had to do with age, language and service usage than a generic dictionary. This was reinforced by Ji et al. (2015), who found that the closer the dictionary is semantically to the target, the higher the success of cracking the password of the target. The role of demographics in password choices is also supported by Mazurek et al. (2013), who in 2013 measured the password guessability of a university. Some of the findings of this study show that overall men created slightly stronger passwords than women. Additionally, computer science students were found to have some of the strongest passwords, while business students had some of the weakest. Wang et al. (2017) looked at 12 datasets from specific communities and observed that the type of website the password came from, played a role in password choices. For example *jesus1* was one of the most popular passwords for two of the websites that were Christian focused.

Another deciding factor for a user's choice of password is the password policy in place. It was observed that more stringent password policies resulted in users spending more time choosing a password and using more special characters (Ur et al., 2012) or being more often unsuccessful in creating a suitable password in the first try (Komanduri et al., 2011). On the other hand, according to an examination of existing password policies length requirements by Shay et al. (2016), length and usability are not always inversely proportional. That was also verified by Shay et al. (2012), who demonstrated that the use of system generated 3- and 4-word passphrases did not perform better than system generated passwords. In fact, the rate at which users forgot the passphrases, or needed to write them down was similar to passwords, but the errors they made while entering the passwords were greater. In the case of Bonneau and Shutova (Bonneau et al., 2012), where users chose the passphrases, the distribution followed that of natural language, which increased their guessability compared to randomly assembled passphrases.

2.2. The case of the non-native English users

Many examples of password strength analysis are focused on that of native English speakers. When it comes to password choices and password strength, there are differences between English and non English speaking users. In an analysis of honeywords in 2018, Wang et al. (2018) found that 36.95% to 51.43% of Chinese speaking users use their personal information to generate passwords, while this figure ranges from 12.76% to 29.94% for English speaking users. Furthermore, Chinese speaking users tend to use numbers in general and dates in particular, more often than English speaking users (Wang et al., 2019). It has also been found that password strength metrics often over inflate the perceived complexity when passwords are composed of non-English words and fragments (AlSabah et al., 2018). Finally, Wang et al. (2019) also noticed that while Chinese passwords are more vulnerable on online attacks than their English counterparts, the Chinese passwords that are left, take longer to crack on average, as the number of guesses increases.

2.3. Password guessing tools and techniques

Many commercial, free and open-source passwords guessing tools are currently available, e.g., Passware,[2] Elcomsoft,[3] John-the-Ripper[4] and Hashcat.[5] Those tools simultaneously leverage both the Central Processing Unit (CPU) and Graphical Processing Unit (GPU) to increase performance. There are also Field Programmable Gate

[1] https://haveibeenpwned.com.

[2] https://www.passware.com/kit-forensic/.
[3] https://www.elcomsoft.fr/eprb.html.
[4] https://www.openwall.com/john/.
[5] https://hashcat.net/.

Array (FPGA) approaches, such as SciEngines dedicated hardware.[6] However, FPGAs are typically a more suitable choice to evaluate specific functions, especially when power consumption is an issue (Gaspar et al., 2014).

The range of password guessing techniques is relatively wide. Ranging from the most basic technique, typically an exhaustive search, to complex deep learning techniques using Generative Adversarial Networks (GANs), e.g., PassGan (Hitaj et al., 2019). The standard approach remains the dictionary approach combined with mangling rules. In this technique, a list of common password candidates, the dictionary, is used together with a set of password modification functions. These functions attempt to mimic typical human behavior, such as placing a digit at the end of the password, capitalizing the first letter, replacing 'a' characters with '@' symbols, etc. Most modern approaches are machine-learning based, such as PCFG (Weir et al., 2009) and OMEN (Dürmuth et al., 2015). These tools leverage the availability of large datasets of human-chosen passwords.

Password cracking contests are also often organized helping to better grasp the capacity of experts in retrieving passwords; the most famous of which being the *Crack Me If You Can Contest*[7] from KoreLogic held during DefCon.

3. Methodology

This section specifies the origin of the dataset and the steps we have taken to clean it. In a nutshell, we gathered plaintext passwords from Hashes.org and from the CynosurePrime team. Then we identified and removed non human-chosen passwords. We provide a more detailed explanation in what follows.

3.1. Dataset origin

The source dataset used for this analysis is the *Have I been Pwned* password dataset. The original website was created by Troy Hunt, a web security expert, to help users detect if their email address(es) appear in data breaches. In 2017, Hunt launched an API to check whether a given password appeared in a previously leaked database. The objective behind this tool is to reduce the password reuse phenomenon and prevent credential stuffing attacks (Pearman et al., 2017) by implementing a searchable password blacklist, as strongly encouraged by the latest NIST directive (Grassi et al., 2017). All the passwords from various breaches have been concatenated in a single dataset and made publicly available for companies and institutions to implement their own black listing of passwords independently. Institutional ethical review for the work presented in this paper has been approved by University College Dublin.

At the time this research had been conducted, five incremental versions of this list had been released since 2017, with each newer version containing more passwords, updated counts of each password's occurrence and the removal of "garbage" passwords (e.g., badly encoded, duplicates, etc.). The dataset does not provide any additional information about each password such as the breach it came from nor the date discovered. However, it can be assumed that the entries of the dataset come from the data leaks listed on the HIBP website. The date spread of the total number of accounts compromised by those data breaches is displayed in Fig. 1.

The total number of accounts compromised in these breaches is over 9.4 billion. However, HIBP_v5 does not contain this number of passwords. This can be attributed to several explanations. It is known that there was no password associated with over 2.8 billion of the breached accounts. Furthermore, as declared by Troy Hunt, the list is composed only of passwords that were initially gathered in plain text while the website can still list the username as breached when the password is not stored in clear. This composition is not without consequence on the results of our analysis for the two following reasons. Firstly, The strongest passwords can be missing from the list obtained by Troy Hunt if the original source was not in clear text. Secondly, if the passwords were stored in clear, then the strongest password are contained in the related leak. However, the corresponding service/website was not following basic security recommendations which can easily lead one to believe that little attention has also been given to password security. We still believe that, thanks to the large size of the list, our analysis is relevant for an overwhelming proportion of accounts. It is furthermore particularly challenging to obtain a dataset of strong password to complement our analysis to bridge this bias.

3.2. Retrieving the plaintext

In order to conduct a statistical analysis of the passwords from the HIBP_v5 dataset, the passwords are first required to be in clear text. The Hashes.org website[8] contains lists of clear text values for many password datasets – including the five versions of the HIBP dataset. The recovery ratio from the HIBP_v5 hash list is above 99.2%. In 2017, the CynosurePrime team, a password research collective, managed to recover almost all passwords from the first version of the HIBP list,[9] claiming a final recovery ratio of 99.9999%. One of the purposes of their work being research, their list of recovered clear text passwords was shared to the researchers in this work. CynosurePrime initially focused on HIBP_v1, and therefore their list contains passwords from this list removed from later versions. Those passwords were removed essentially because they were somehow corrupted (e.g., badly encoded), duplicates, or not generated by humans. The CynosurePrime list was merged with the one collected from hashes.org to enrich our dataset with passwords from the later versions of HIBP. While this list contains more than 99% of the passwords of HIBP_v5, it should be mentioned that the small percentage of passwords that has not been included has not been recovered by either the CynosurePrime team, or the Hashes.org team. These passwords can be assumed to be some of the strongest in HIBP, which is something to be taken into account.

3.3. Cleaning the dataset

We initially removed from the obtained dataset all the passwords encoded in hexadecimal format, corresponding to approximately 35 million passwords. While being valid passwords, the tool used for the basic analysis would not handle them properly. Further, a majority of these hex encoded passwords consisted of inputs which were wrongly encoded or handled on the HIBP dataset creation.

During our analysis, one unusual pattern was identified with a significantly high frequency. The "word" *fbobh* was discovered in the top 10 of used words. This is not a common word found in searching regular sources nor is it a common pattern, e.g., a keyboard walk - letters that are next to each other on a keyboard. Overall, it was identified that approximately 3.6 million *unique* passwords from HIBP_v5 have the structure "fbobh_XXXX", where "XXXX" represents four random characters including lowercase, numbers and specials, but not uppercase. These passwords can be

[6] https://www.sciengines.com/technology-platform/sciengines-hardware/.
[7] https://contest-2019.korelogic.com/.
[8] https://hashes.org/.
[9] https://blog.cynosureprime.com/2017/08/320-million-hashes-exposed.html?m=1.

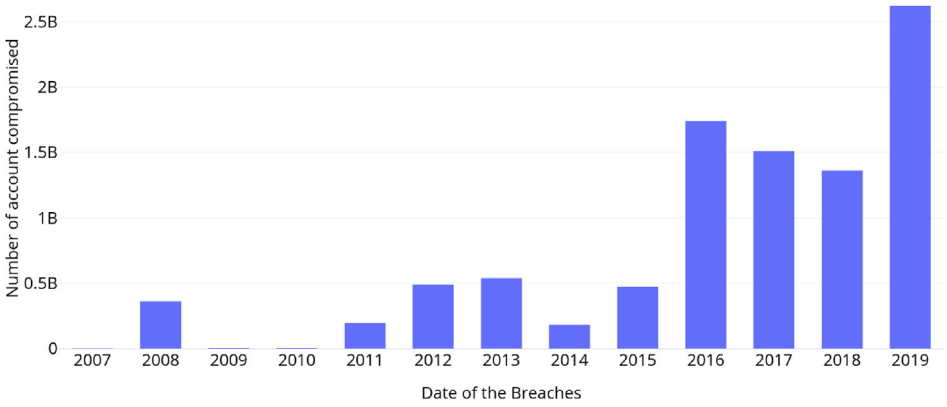

Fig. 1. Number of breached accounts listed in have I been Pwned.

attributed to the MySpace data breach and are not human generated. Therefore, these passwords were removed from our analysis.

The clear text list used for the remainder of this work is therefore composed of 515,680,539 unique passwords. Considering the count value from the HIBP_v5 for each password's occurrence in data breaches, this dataset represents a total of 3,951,907,330 passwords. Table 1 shows the 20 most popular passwords found in this dataset along with their percentage of the total accounts associated with each. Many of these passwords feature heavily on most common or worst password lists. As can be seen, sequences of numbers and keyboard walks are the most popular choices found in the dataset.

4. Basic analysis of the dataset

The objective of the further basic analysis in this section is to present global characteristics about the passwords including the type of alphabet used and the most frequent patterns. The Password Analysis and Cracking Kit (PACK)[10] was used to analyze the HIBP_v5 dataset. PACK provides several analysis tools, but the included `statsgen` script provides the functionality needed to perform this analysis.

Table 1
Top 20 passwords in HIBP_v5.

Password	% of Total Accounts
123456	0.596%
123456789	0.197%
Qwerty	0.099%
password	0.094%
111111	0.079%
12345678	0.074%
abc123	0.072%
1234567	0.064%
password1	0.061%
12345	0.060%
1234567890	0.057%
123123	0.056%
000000	0.050%
iloveyou	0.041%
1234	0.033%
1q2w3e4r5t	0.030%
qwertyuiop	0.028%
123	0.026%
monkey	0.025%
Dragon	0.025%

4.1. Length distribution

Fig. 2 provides an overview of the most common lengths of unique passwords in the dataset, i.e., the aforementioned 515,680,539 passwords. One statistic that immediately stands out is that more than 30% of the unique passwords from HIBP_v5 are eight characters long. A highly probable explanation for this is that most password guidelines and policies specify minimum length requirements, such as the 8 characters minimum in the NIST recommendation (Grassi et al., 2017). The second most frequent length is ten corresponding to 17% of the passwords. The overall password length ranges from 1 to 449 characters, yet 84% of the passwords have a length that falls into the 6–12 character range.

4.2. Character sets usage

There are typically four classes of characters considered in the password analysis community; lowercase, uppercase, numbers, and special characters. An analysis of the character type composition of the unique HIBP_v5 passwords can be seen in Fig. 3. PACK analyses the composition of passwords and classifies them according to the type of character set used. For example, a password is associated with the category *loweralphaspecialnum* when it contains lowercase, special characters and numbers, e.g., `pa$$w0rd`, no matter what the order or frequency of appearance of the component characters are. A description of each of the categories used by PACK is shown in Table 2. Using this classification, PACK outputs the count of passwords in each category. Fig. 3 shows the distribution of these categories, where in other the lowest represented categories are combined. As can be seen in the figure, 46% of the passwords are composed of a mix of lowercase characters and

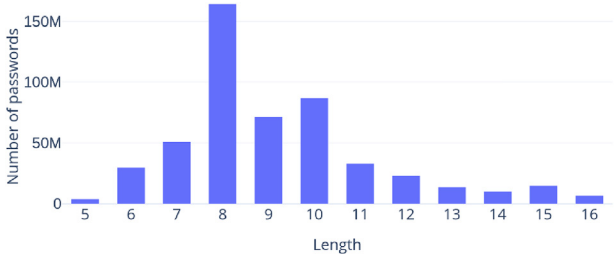

Fig. 2. Most common Password lengths.

[10] https://github.com/iphelix/pack.

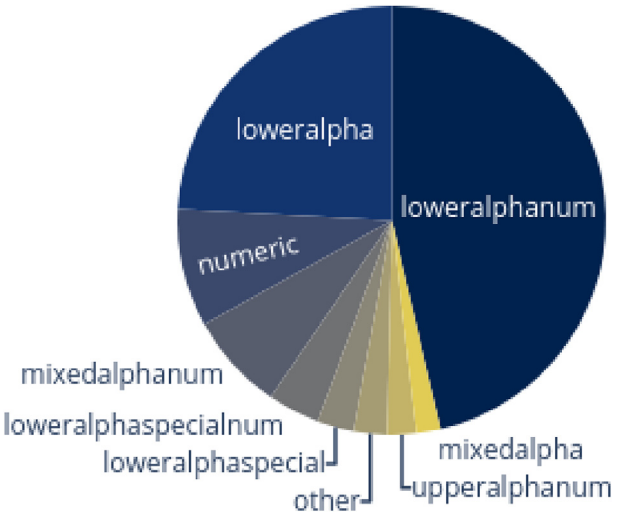

Fig. 3. Occurrence of character categories.

numbers. The second and third largest classes correspond to passwords composed of only lowercase (24%) or only numbers (8%) respectively. One notable observation from this analysis, is that over 75% of passwords from the dataset contain neither special nor uppercase characters. This is not such an unexpected outcome as most password policies require at least 2 different character sets to be present in a password (Florêncio and Herley, 2010).

4.3. Pattern analysis

The analysis can be further refined as it focuses on character sets without considering the internal password structure. For example, the category *loweralphanum* contains passwords like *12password*, *password12*, and *pass12word*. A more refined classification, where the internal order is considered, would separate these into three different categories. This further classification is important because the approach to guess these passwords will be different. Following the vocabulary used in password guessing techniques, these internal password structures are called “masks”. Therefore, for the above mentioned examples, the corresponding masks would be *digitstring*, *stringdigit* and *stringdigitstring*, respectively. The 15 most common masks from the HIBP_v5 dataset are shown in Fig. 4. The most common mask is *stringdigit*, meaning that the passwords of this category are composed of a string (lowercase and/or uppercase) immediately followed by one or more numbers, e.g., `paSSword123`). As determined by Tatlı (2015), users typically pick an alphanumerical string, commonly a word or a name, and add numbers at the end to fulfill the length and character sets requirement of the enforced password policy. The next most common masks are string, digit and digitstring. These four masks combined represent over `75%` of the passwords.

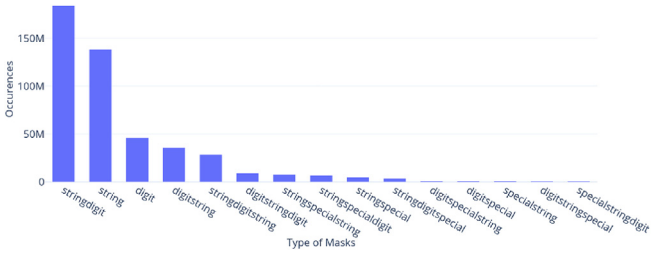

Fig. 4. Simple masks.

5. Results of advanced analysis

The previous Section provides a basic overview of the dataset’s composition. In this section, this analysis is extended with the usage of the Óðinn Framework (Coray, 2019) that we have adapted and enriched particularly for this advanced analysis.

5.1. Óðinn framework

Óðinn is a tool that can split passwords into their basic fragments and find their semantic meaning. It can also create password candidates out of multiple fragments and recover longer and more complex passwords, that other state-of-the-art password guessers failed to recover. It has a modular architecture facilitating the addition and adaptation of its analysis functionality. This facilitates pipelined workflows that consist of multiple modules. This enables the execution of multiple steps first, before the final analysis is performed, e.g., split passwords into fragments → classify fragments → aggregate the classes. The two main components used in this work are for fragmentation and classification.

5.1.1. Fragmentation

The goal of fragmentation in Óðinn is to split a password into meaningful fragments, such as its component words, e.g., *ilovemom* should be split into three fragments. This fragmentation is achieved in two steps. Firstly, the passwords are decomposed according to the three basic character sets, namely letters, numbers and specials.

Table 2
Types of patterns.

Pattern	Meaning	Example(s)
loweralpha	Lowercase only	`password`
upperalpha	Uppercase only	`PASSWORD`
mixedalpha	Lower and uppercase only	`paSSwoRD`
numeric	Numbers only	`123456`
loweralphanum	Lowercase and numbers	`password12, pass12word`
upperalphanum	Uppercase and numbers	`PASSWORD12, PASS12WORD`
mixedalphanum	Lower and uppercase and numbers	`pasSWord12, PASS12word`
special	Special characters only	`%.&;#`
loweralphaspecial	Lowercase and special characters	`password!, pa$$word`
upperalphaspecial	Uppercase and special characters	`PASSWORD!, PA$$WORD`
specialnum	Special characters and numbers only	`123456!, 123!456`
mixedalphaspecial	Lower and uppercase and special characters	`Password!, !Pa$$word`
loweralphaspecialnum	Lowercase, special characters and numbers	`password1!, !pa$$1word`
upperalphaspecialnum	Uppercase, special characters and numbers	`PASSWORD1!, PA$$1WORD!`
all	Lower and uppercase, special characters and numbers	`passWORD1!, !pA$$1woRd`

Subsequently, the letter fragments are split into further fragments when appropriate to do so. This second step is performed using `SymSpellPy`,[11] a Python implementation of `SymSpell`,[12] which is one of the most efficient spelling correction algorithms (Garbe, 2017).

As a ground truth for splitting text into single words, SymSpell needs a dataset of words with their corresponding frequency counts. This dataset has to be seen as a vocabulary list and not as a set of password candidates. The `SymSpellPy` library comes with a small English dictionary with counts as default. Such approach is very efficient for tasks such as auto-correction modules or other natural language processing tasks. However passwords are likely to contain foreign expressions, purposely mistyped words, popular culture references/characters, celebrities, or slang words which are missing from standard language datasets and therefore using classical dictionaries would fail to properly fragment passwords. An ideal solution relies on the existence of a dataset composed of fragments properly extracted from real passwords which, to the best of our knowledge, does not exist. We have therefore produced our own dataset, extracting words from 3, 937, 684, 877 Reddit comments.[13] This source was chosen for two reasons: 1) the comments contain slang words and common expressions used on the internet, and 2) these comments are written in several languages resulting in a multilingual dictionary.

5.1.2. Fragment classification

As there can be many different types of fragments composing real-world passwords, Óðinn provides different ways of classifying them:

- WordNet – To classify normal English words, WordNet[14] provides a *synset* (a set of synonyms relating to a single given word). As WordNet is built hierarchically, the tree can be climbed to get synsets with a broader meaning for the classified word.
- Functions – Functions check if a given input matches the patterns defined within them, e.g., years or dates.
- Dictionaries – Óðinn contains a collection of dictionaries, each of them listing words of a specific class, e.g., cities. These lists are mostly hand-crafted and refined.

Tests with Óðinn have shown that in most cases, WordNet is classifying words correctly. However, it quickly reaches its limit. For example, simple typos or slang words are not correctly classified by WordNet, which is only looking for exact matches. This is an issue with passwords, as it is common to use slang words and phrases, e.g., *iluvmymom*. To compensate for this insufficient classification, enriching the dataset of words used with the non-classified fragments was a focus of this work. GloVe (Pennington et al., 2014) was used to automate this process with its *Common Crawl 42B 300d*, a pre-trained model in English for GloVe.[15] The process used can be summarized as follows. A proximity score between each non-classified fragment and the previously defined categories is computed. This proximity score is the Euclidean distance between the embeddings of those words in GloVe. The fragment is then added to the categories for which the distance is smaller than given threshold. This process was repeated as some fragments could remain unclassified after one pass but be classified in the second pass thanks to the previous extension of the dataset of words. Many fragments were still not classified using this process; mainly random strings, typos and slang. This is because they do not have a representation in the *Common Crawl* and therefore cannot be compared to the categories. Once the classification is achieved, Óðinn produces the frequency counts for all the observed combination of classes, e.g., the number of times passwords are composed of a name followed by a year. As one of motivations of this work was to analyze in more detail those classes and their combinations, Óðinn was configured to save the classification of each password in addition to the aggregated data.

5.2. Analysis on password fragments

At this point, two cases were possible for the advanced analysis: either analyzing the unique passwords, or analyzing the passwords considering the number of occurrences in the dataset. The latter option better maps the human behavior and therefore the below analysis relies on the 3.9 billion non-unique passwords of HIBP_v5. Óðinn produced `1,575,290,376` fragments out of the unique passwords in HIBP_v5, the breakdown of which can be seen in Table 3. The three lists, namely letters, numbers and special characters, were further processed in order to see the most common fragments of each category. A full table of the Top 50 most frequent fragments in all three categories can be found in Table 4. We have removed from the letter fragments those classified by Óðinn as single and double letters without meaning, e.g., "xf" is removed but "it" remains). `a` and `i`, which were respectively classified as an article and a pronoun, hold the top spots. As they are frequently encountered in the "Top Worst Passwords" lists verbatim or as parts of a password, `qwerty`, `password` and `love` are unsurprisingly rounding out the top 5 (Rawlings, 2019). In the top 50 there are some fragments that consist of phrases, such as `iloveyou`. The reason this is not broken down further is that, as mentioned in Section 5.1.1, the training of Óðinn was done with Reddit comments and this phrase appeared verbatim there and is therefore considered a single word. Furthermore, we notice keyboard walks such as `qwerty`, `qwe`, `qaz` are featuring prominently in the top 50 for both word and number fragments. The same holds true for the top 50 number fragments, where 3 out of the top 5 most frequent fragments are sequences of numbers. Furthermore, single digits, `1`, `2`, `3`, double digits `12`, `11`, `13`, and number repetitions `111111`, `000000`, are encountered in the top 50 number fragments. When it comes to special characters, the top 15 most encountered special characters are single, followed mostly by patterns of repetition. It is worthy to mention that the order of magnitude for the top 50 special characters is one order smaller than the top 50 letters and numbers. This corroborates the suggestion that users prefer alphanumeric characters and tend to avoid those that requires multiple keys to type, as is often the case with special characters (Bonneau, 2012). Looking further down at the number-based fragments, some noteworthy fragments are found in the top 500. When it comes to numbers we noticed many four digit numbers in the top 500 number fragments falling within the 1900 to 2020 range, i.e., common years. The first appearance of a four digit number that is presumably a year is `2010` at no. 56 and subsequently an overall of

Table 3
Breakdown of password fragments per category.

letters	`1, 074, 196, 225`
numbers	`439, 727, 373`
special	`61, 366, 778`
total	`1, 575, 290, 376`

[11] https://github.com/mammothb/symspellpy.
[12] https://github.com/wolfgarbe/SymSpell.
[13] https://files.pushshift.io/reddit/comments/.
[14] https://wordnet.princeton.edu/.
[15] http://nlp.stanford.edu/data/glove.42B.300d.zip.

Table 4
Top 50 letter, number and special character fragments.

Letter	Count	Number	Count	Special	Count
a	2.335%	1	8.240%	.	0.871%
i	1.168%	123456	5.137%	_	0.666%
qwerty	0.597%	123	2.574%	!	0.469%
password	0.510%	2	2.398%	@	0.334%
love	0.484%	123456789	2.083%	–	0.327%
my	0.356%	3	1.788%	:	0.140%
abc	0.274%	4	1.578%	#	0.105%
to	0.259%	5	1.111%	*	0.090%
an	0.259%	12	1.079%	$	0.071%
qwe	0.248%	7	1.029%		0.065%
in	0.238%	0	0.870%	&	0.045%
the	0.228%	8	0.812%	+	0.042%
qaz	0.223%	6	0.810%	?	0.037%
iloveyou	0.221%	12345	0.764%	,	0.035%
ws	0.217%	9	0.761%	/	0.031%
as	0.209%	1234	0.664%	!!	0.025%
no	0.198%	11	0.599%	.:	0.023%
ilove	0.196%	13	0.518%	&#	0.022%
by	0.191%	12345678	0.474%	=	0.021%
man	0.190%	01	0.430%	;	0.018%
baby	0.178%	10	0.425%	..	0.017%
on	0.176%	1234567890	0.418%	'	0.016%
it	0.156%	111111	0.411%	%	0.014%
we	0.145%	22	0.390%	<	0.014%
go	0.145%	23	0.375%	(	0.011%
he	0.145%	123123	0.365%	[	0.011%
asd	0.134%	1234567	0.360%	)	0.011%
sexy	0.131%	69	0.331%	**	0.010%
you	0.128%	21	0.321%	…	0.010%
boy	0.126%	14	0.284%	:,	0.009%
of	0.124%	15	0.248%	‘	0.009%
qa	0.117%	09	0.248%	$$	0.008%
girl	0.116%	08	0.236%	__	0.007%
fuckyou	0.114%	07	0.224%	!!!	0.007%
july	0.113%	99	0.224%	@@	0.006%
angel	0.111%	24	0.222%	–	0.005%
ma	0.109%	88	0.221%	.,	0.005%
march	0.107%	16	0.212%	^	0.005%
dog	0.106%	18	0.209%	~	0.004%
at	0.105%	000000	0.207%	!@	0.004%
big	0.103%	17	0.206%	! ~!	0.004%
monkey	0.102%	00	0.204%	>	0.004%
one	0.101%	19	0.202%	***	0.004%
alex	0.099%	77	0.193%	!@#	0.004%
red	0.095%	33	0.190%	]	0.003%
us	0.094%	20	0.187%	??	0.003%
qwer	0.094%	123321	0.183%	++	0.003%
qwertyuiop	0.094%	25	0.181%	"	0.003%
dragon	0.092%	666	0.174%	???	0.003%
life	0.091%	06	0.170%	==	0.002%
shark	0.090%	89	0.150%	*****	0.002%

37 four digit numbers between 1970 and 2010 appear in the top 200 alone. This leads us to believe that users often choose memorable patterns even for the number portion of their passwords like year of birth or other important dates. In what concerns special-based fragments, most of them are repetitions of the same character like "!!" at rank 16. Some meaningful structure are still present in the top 500 in the form of emojis, such as ":)" at rank 65 or "^_^" at rank 198.

5.3. Analysis on classified fragments and passwords

Table 5 lists the most frequent classes of fragments occurring in the HIBP passwords. The fragments that were not classified at all or those not semantically meaningful, i.e., char/twochar/threechar, were filtered from this list. The three first classes are related to numbers, either generic ones like single digits, common ones (e.g., `123456` or `1111`, etc.), or years. On one hand, this can be explained by the fact that many password policies require passwords to contain more than just letters. On the other hand, numbers are also very popular in Asian countries, most probably due to the fact that they can be digitally entered more easily than ideograms, especially on mobile devices (Wang et al., 2019). The top 25 classes contains semantically-rich categories such as cities, animals, food and sports reinforcing the idea that the surrounding context of a person might influence the choice of the password. However, it is not possible to affirm with conviction that this is the case, e.g., the name of a city can be unrelated to the person who chose it. Identifying the most common combinations of component passwords classes enables the analysis of the unique classes. The results are displayed in Table 6. Similar to the most frequent fragments, numbers and names are commonly used in combination with other classes. The number-based passwords are followed by various combinations of female and male names in combination with appended single digits or larger numbers. When password policies require more than one type of character, users might consider "padding" their passwords with special symbols and/or numbers, like years, at the end in order to fulfill the length requirement. Furthermore, keyboard walks and

Table 5
Most frequent classes of component password fragments. The count represents how many passwords in which this class occurred at least once.

Count	Percentage	Class
1,223,930,168	30.97%	number
674,454,756	17.07%	common-number
338,857,959	8.57%	year
297,403,194	7.53%	masculine_name
266,976,738	6.76%	feminine_name
179,058,386	4.53%	name
109,891,541	2.78%	article
102,376,618	2.59%	pronouns
97,630,848	2.47%	city
92,259,083	2.33%	special
81,998,629	2.07%	keyboard
61,214,229	1.55%	prepositions
57,435,482	1.45%	animal
50,064,712	1.27%	connector
49,162,058	1.24%	family
45,663,992	1.16%	computers
40,156,119	1.02%	people
37,866,704	0.96%	person.n.01
33,855,125	0.86%	swear
29,082,262	0.74%	food
27,575,938	0.70%	colours
25,638,436	0.65%	emotions
23,799,390	0.60%	sports
22,868,852	0.58%	love
20,607,713	0.52%	negative

Table 6
Most Frequent Password Fragment Combinations. *x* Represents Fragments That Were Not Classified.

Count	Percentage	Combination
437,959,119	11.08%	common-number
432,721,719	10.95%	number
48,306,129	1.22%	feminine_name
45,713,052	1.16%	masculine_name + number
45,344,781	1.15%	masculine_name
39,786,125	1.01%	feminine_name + number
33,685,017	0.85%	x + year
27,958,256	0.71%	feminine_name + digit
26,308,310	0.67%	masculine_name + digit
25,821,041	0.65%	keyboard
24,678,272	0.62%	city
23,689,948	0.60%	name
21,252,289	0.54%	masculine_name + year
20,815,196	0.53%	x + common-number

cities are also popular choices.

6. Strength analysis

6.1. Strength classification

One of the most useful characteristics about passwords is their strength. Users are probably not always concerned in having strong and safe passwords or simply not aware of the consequences of having a weak password. This hypothesis is supported by the massive use, and re-use, of weak passwords. However, the strength of the password becomes crucial when it is about protecting critical service, e.g., bank accounts or the security of a large infrastructure. To this matter, password metrics are often put in place to ensure a minimum strength of the password. The most spread one is the one proposed in 2012, and updated in 2017 (Grassi et al., 2017), by NIST recommending a minimum of 8 characters including lowercase, uppercase, special and digit. However, this approach has shown its limits with time and attackers have adapted their attacks to mimic the typical patterns followed by humans in general. A plethora of other metrics have emerged each of them being based on different heuristics and methods to assess the strength of passwords. (Galbally et al., 2017) and (Golla and Dürmuth, 2018) have proposed a comparison of those metrics. While the method proposed in (Galbally et al., 2017) is interesting because it provides different evaluation criteria for each password and therefore better understanding of why a password is strong or weak, the proposed implementation is not fast enough to analyze more than 500 million passwords in a timely manner. The best password metric according to (Golla and Dürmuth, 2018) is based on the HIBP API and therefore it does not seem at all suitable to us to assess a dataset using an approach based exactly on such dataset. The common point in these two articles is that the `zxcvbn` password strength metric, originally deployed in the Dropbox service, provide good results. We therefore used the python implementation to analyze the 500 million unique passwords from our dataset. This metric attributes an integer score between 0 and 4 to each password according to strength, with passwords in class 0 being the weakest and those in class 4 the strongest. The division of passwords among those classes is displayed in Table 7.

6.2. Hardware consideration

It is essential to include an evaluation on the hardware needed in digital forensic laboratories to make password cracking viable. As previously mentioned, passwords are predominantly stored in a hashed/salted hash format. The hash function employed is therefore a security parameter in case of a data breach. Indeed, if the hash function is quick to evaluate, an attacker will have the capacity to evaluate more candidates than if the function is slow. The MD5 hash function has been widely used to store passwords and even though it is deprecated, it is still commonly used in online services. A single gaming graphics card, a Nvidia 2080 Ti, is able to evaluate 50×10^9 password candidates per second. In order to better visualize these figures, a single 2080 Ti can fully evaluate all possible MD5 passwords up to length 8 considering an alphabet of 95 characters (26 lowercase, 26 uppercase, 10 digit, and 33 special characters) in less than 2 days. Considering the BCRYPT hash function, specifically designed to be slow on graphic cards, only up to five characters can be brute forced in practical time, as the card can evaluate approximately 25,000 passwords per second.

Table 7
Percentage of Unique Passwords per `zxcvbn` Class.

Score	0	1	2	3	4
Percentage	0.04%	14.7%	47.3%	26%	12%

6.3. Password guessability

The analysis of the guessability of passwords is outlined below for two scenarios, namely a fast and a slow hash function. For this purpose, the length of passwords in each of those classes has been measured. Fig. 5 shows the proportion of passwords of a given length for each of the classes produced by `zxcvbn`. In the case of a fast hash function, passwords belonging to class 2 and below can be recovered by an exhaustive search and should therefore be considered as really weak.

`zxcvbn` provides, together with the score, an approximation of the number of guesses an adversary would need to guess a password. Based on this figure, a password belonging to class 3 could be recovered using a single 2080 Ti graphics card in a time frame of approximately 5 days in the case of a slow hash function. Therefore, a digital investigator targeting a single password will manage to retrieve it. While this figure is indicative, it reveals that passwords in class 3 and below should be considered weak, especially as this time frame is only considering the use of a single graphics card. Adding additional graphics card will effectually reduce the time linearly. Class 4 passwords, at a first glance, are more secure. The minimum length of these passwords is 11 and 75% of those passwords have a length between 11 and 15. Based on the results from Óðinn, those passwords are composed of more fragments than in average, with 4.4 fragments for class 4 passwords versus 2.1 fragments for all passwords in HIBP_v5. According to the number of guesses required, which has an average of 5.8×10^{24}, passwords in this class are more resistant to classical attacks – even considering a fast hash function. However, 42% of these passwords are solely composed of lowercase characters and numbers. If prior knowledge about a given password is known, such as frequent used pattern(s) derived from other passwords of the same user, specific targeted attacks become possible. We highlight below the time required to fully explored the most common patterns of the password from class 4 considering a fast hash function:

15 digits - 11% of the passwords - space fully explored within a day in case of MD5. In the case of BCRYPT, it would take 1268.3 years considering a 2080Ti NVIDIA GPU. 12 lowercase - 2% of the passwords - space fully explored in approximately 22 days in case of MD5 and in 120,961 years in case of BCRYPT. 11 lowercase - 2% of the passwords - space fully explored within a day in case of MD5 and 4655.4 years in case of BCRYPT.

Exhaustive search is nevertheless not the recommended approach to recover strong passwords. These figures serve to illustrate that even passwords considered as secure can be recovered when prior knowledge is available. To reinforce this idea, we have extracted the advanced analysis results for the passwords of this specific class. Table 8 shows for the 10 most used types of

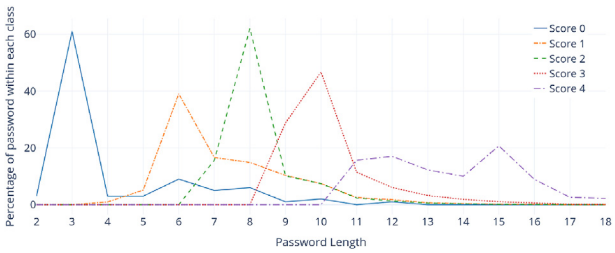

Fig. 5. Password Length Distribution within `zxcvbn` Score Classes.

Table 8
Comparison of the most frequent classes of password fragments between all the passwords and those from Class4.

Class	All Passwords	Class 4 Passwords
Number	30.97%	**49.95**%
common-number	17.07%	5.03%
Year	8.57%	14.8%
masculine_name	7.53%	8.34%
feminine_name	6.76%	7.41%
Name	4.53%	8.75%
Article	2.78%	7.05%
pronouns	2.59%	6.14%
City	2.47%	2.24%
Special	2.33%	**12.73**%

fragment how often they appear in all passwords compared to class 4. As rightly recommended by strong password policies, the number of occurrences of number-based fragment and special-based fragment is higher for the class 4 passwords. The frequency of year is higher while the frequency of common-numbers is much lower, yet this might be due to a weak classification of number-based fragments. What remains interesting is that names, either masculine, feminine or proper names, are more present than in the average password. Other "contextualised" categories remain present with mostly minor fluctuations. Two more noticeable differences are the classes of computer-based words, moving from 1.16% to 2.02%, and cooking-related words, moving from 0.49% to 0.96%.

Therefore, if passwords belonging to class 4 are in average longer and composed of more fragments, additional knowledge about the person whose password they want to retrieve would be beneficial and could tilt the balance in favor of the attacker.

7. A preliminary analysis

As part of this analysis, and in order to investigate the hypothesis that there is a link between the thematic content of a website and the password chosen, we decided to look at one specific leak from hashes.org. The leak we chose came from the website *mangatraders.com*. The leak contains 881,468 entries (with 618,237 unique passwords). We used pipal[16] to extract the top 100 passwords, as well as the top 100 base words. A base word is defined as a password where non-alpha characters from the beginning and end have been removed. Table 9 shows that the top 100 passwords represent 4.76% of the total number of accounts. From these 41,821 passwords, 15,758 (or 37.6%) are manga related (representing 1.79% of the total number of accounts). Interestingly, looking at unique passwords only (and not counting the number of occurrences, 51 out of the top 100 passwords were related to manga. When it comes to base words, the percentage of manga related base words is even higher (3.29% of the total and 63.8% of the top 100 base words).

This reinforces our assumption that users are inspired by the purpose and thematic content of the website they create their password for. Of course, a more extensive analysis of how exactly and to what extent, the thematic content correlates to the passwords chosen is warranted but beyond current scope.

8. Conclusion

The analysis presented in this article consists, to our knowledge, the largest and most comprehensive analysis on the building blocks of a password, to date. The aim of it is to give greater insight into password selection of users and highlight password trends when it comes to context, length, strength and architecture.

Table 9
Manga related passwords in mangatraders.com.

	Total	Manga related
Top 100 Passwords	41,821 (4.76%)	15,758 (1.79%)
Top 100 Base Words	45,206 (5.15%)	28,783 (3.29%)

The HIBP dataset is composed of various data breaches from very different sources, yet we see the popularity of some passwords and construction processes is universal. We also see, that even in such a diverse dataset as HIBP, contextual trends start to emerge as fragment categories such as city, year and name are some of the most popular, suggesting that users very often opt for passwords that contain familiar models. User interests are also high on the list of most popular fragments, with sports, food and animals taking some of the top spots.

On the side of password cracking, the information yielded from this analysis aims to highlight that when possible, common approaches should always be conducted first, as they would retrieve approximately 80% of the passwords, namely those belonging to class 3 and below. If those approaches fail, the need for contextual information appears to be essential to continue the recovery process, as otherwise, the success rate would be close to zero.

Such additional information must contribute to the drafting of a targeted wordlist for candidate generation, or driving the way the candidates should be composed. The most relevant information in this situation appears to be numbers, such as important dates, in combination with names, most probably of relatives.

Last but not least, the way this information is used to create password candidates must always be according to the construction process followed by the person that is targeted. For this, other passwords from the same person might suggest probable password structures.

8.1. Discussion and takeaways

Our analysis of the HIBP dataset decisively shows that clear trends of contextualisation can be found in passwords. Users use passwords they easily remember, something that makes them weak and easier to guess. The over 515 million reversed engineered passwords from the HIBP dataset produced 3 times as many password fragments, which shows that there is merit in this approach and in fact, a deeper analysis of the fragments is warranted. The new insights provided by password fragments can help inform not only password cracking but also on the other side of the equation, password policy creation.

The analysis of the password masks highlighted the most common combinations of character categories. This can serve to; 1) inform password policies; and 2) give insight into the most popular construction processes users follow.

The strength analysis on this password dataset shows that the majority of passwords remain weak, and easily recovered with an exhaustive search. Passwords of class 4, which were the strongest, would still be susceptible to a brute force attack considering a fast hash function. On the other hand, we showed that with a slow hash function, it would be a lot more difficult and costly. Therefore, special attention should be paid to the way the passwords are stored, because in many cases the hash function will be the only obstacle in the way of an attacker.

Looking at the contextual information that can be found through the classification of the fragments, attention should be paid on how it can be translated to viable password candidates. Such

[16] https://github.com/digininja/pipal

information is often available through classical means of investigation in the case of law enforcement, and could tilt the balance in their favour. In the case of an attacker targeting an individual, this type of information may be found by unlawful means or in some cases by what the victim themselves have shared online. This is why it is especially prudent to be mindful of an attacker's targeted approach.

But looking at contextual information about passwords can be both a friend and a foe. Context can be leveraged for a targeted attack, but it is also what helps people memorize and retrieve their passwords. Therefore, in password creation it should be used in conjunction with other strength parameters like length in a long passphrase. Password meters are a good friend. They may fail to identify context but some of them are good to recognize language. Those still give good insights about the strength of the resulting password, therefore, they can be used to ascertain that a password based on contextual information can be both memorable and difficult to crack.

8.2. Future work

In order to refine our analysis, a greater importance in dividing and classifying numbers is needed. Indeed, those fragments appear to be the most common ones, yet our analysis is rather limited in this regard. Letter-based fragments could also be better classified considering language models dedicated to passwords.

Our preliminary analysis of a manga dataset showed that this is a valid approach, as about half of the passwords found in the top 100 most used passwords were related to manga. In the future, we would like to focus on the analysis of such datasets, stemming from specific communities, to understand if the topic for which the password is selected also has an influence on the password selection process. This way, the advantage of these enhanced dictionaries, compared to traditional ones can be ascertained and the value of this approached can be determined. Ultimately, the consideration of contextual information about the user and/or the service the password is used for, can be utilised to reduce the search space during password cracking, by bringing forth password candidates to be checked that would otherwise be considered further down the process or not at all.

References

AlSabah, M., Oligeri, G., Riley, R., 2018. Your culture is in your password: an analysis of a demographically-diverse password dataset. Comput. Secur. 77, 427–441.

Bonneau, J., 2012. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In: 2012 IEEE Symposium on Security and Privacy. IEEE, pp. 538–552.

Bonneau, J., Shutova, E., 2012. Linguistic properties of multi-word passphrases. In: Blyth, J., Dietrich, S., Camp, L.J. (Eds.), Financial Cryptography and Data Security. Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN 978-3-642-34638-5, pp. 1–12.

Brown, A.S., Bracken, E., Zoccoli, S., Douglas, K., 2004. Generating and remembering passwords. Appl. Cognit. Psychol.: Off. J.Soc.Appl. Res.Mem. Cogn. 18 (6), 641–651.

Chen, X., Huang, X., Mu, Y., Wang, D., 2019. A typo-tolerant password authentication scheme with targeted error correction. In: 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE). IEEE, pp. 546–553.

Coray, S., 2019. Óðinn: A Framework for Large-Scale Wordlist Analysis and Structure-Based Password Guessing. Master's Thesis. Computer Science, University of Basel, Switzerland.

Du, X., Hargreaves, C., Sheppard, J., Anda, F., Sayakkara, A., Le-Khac, N.A., Scanlon, M.. SoK., 2020. Exploring the state of the art and the future potential of artificial intelligence in digital forensic investigation. In: Proceedings of the 15th International Conference on Availability, Reliability and Security. Association of Computing Machinery, ISBN 9781450388337. https://doi.org/10.1145/3407023.3407068.

Dürmuth, M., Angelstorf, F., Castelluccia, C., Perito, D., Chaabane, A., 2015. OMEN: faster password guessing using an ordered Markov enumerator. In: International Symposium on Engineering Secure Software and Systems. Springer, pp. 119–132.

Florêncio, D., Herley, C., 2010. Where do security policies come from?. In: Proceedings of the Sixth Symposium on Useable Privacy and Security, pp. 1–14.

Galbally, J., Coisel, I., Sanchez, I., 2017. A new multimodal approach for password strength estimation—Part II: experimental evaluation. IEEE Trans. Inf. Forensics Secur. 12 (12), 2845–2860. https://doi.org/10.1109/TIFS.2017.2730359.

Garbe, W., 2017. 1000x Faster Spelling Correction. https://towardsdatascience.com/symspellcompound-10ec8f467c9b.

Gaspar, L., Coisel, I., Beslay, L., 2014. FPGA Performances in Cryptography. Performance Analysis of Different Cryptographic Algorithms implemented in an FPGA.

Golla, M., Dürmuth, M., 2018. On the accuracy of password strength meters. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. Association for Computing Machinery, pp. 1567–1582. https://doi.org/10.1145/3243734.3243769.

Grassi, P.A., Fenton, J.L., Newton, E.M., Perlner, R.A., Regenscheid, A.R., Burr, W.E., Richer, J.P., Lefkovitz, N.B., Danker, J.M., Choong, Y.Y., Greene, K.K., Theofanos, M.F., 2017. NIST Special Publication 800. In: -63B - Digital Identity Guidelines: Authentication and Lifecycle Management. Tech. Rep. National Institute for Standards and Technology. https://doi.org/10.6028/NIST.SP.800-63b.

Hitaj, B., Gasti, P., Ateniese, G., Perez-Cruz, F., 2019. PassGAN: a deep learning approach for password guessing. In: Applied Cryptography and Network Security. Springer, pp. 217–237.

Ji, S., Yang, S., Hu, X., Han, W., Li, Z., Beyah, R., 2015. Zero-sum password cracking game: a large-scale empirical study on the crackability, correlation, and security of passwords. IEEE Trans. Dependable Secure Comput. 14 (5), 550–564.

Kanta, A., Coisel, I., Scanlon, M., 2020a. A survey exploring open source intelligence for smarter password cracking. Forensic Sci. Int.: Digit. Invest. 35, 301075. https://doi.org/10.1016/j.fsidi.2020.301075.

Kanta, A., Coisel, I., Scanlon, M., 2020b. Smarter password guessing techniques leveraging contextual information and OSINT. In: 2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security). IEEE, pp. 1–2.

Komanduri, S., Shay, R., Kelley, P.G., Mazurek, M.L., Bauer, L., Christin, N., Cranor, L.F., Egelman, S., 2011. Of passwords and people: measuring the effect of password-composition policies. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2595–2604.

Lillis, D., Becker, B., O'Sullivan, T., Scanlon, M., 2016. Current challenges and future research areas for digital forensic investigation. In: The 11th ADFSL Conference on Digital Forensics, Security and Law (CDFSL 2016). Daytona Beach, FL, USA: ADFSL, pp. 9–20.

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 2013 ACM SIGSAC Conference on Computer & Communications Security, pp. 173–186.

Pearman, S., Thomas, J., Naeini, P.E., Habib, H., Bauer, L., Christin, N., Cranor, L.F., Egelman, S., Forget, A., 2017. Let's go in for a closer look: observing passwords in their natural habitat. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 295–310.

Pennington, J., Socher, R., Manning, C.D., 2014. Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543.

Rawlings, R., 2019. Top 200 Worst Passwords of 2019. https://nordpass.com/blog/top-worst-passwords-2019.

Sayakkara, A., Le-Khac, N.A., Scanlon, M., 2019. A survey of electromagnetic side-channel attacks and discussion on their case-progressing potential for digital forensics. Digit. Invest. 29, 43–54. https://doi.org/10.1016/j.diin.2019.03.002.

Shay, R., Kelley, P.G., Komanduri, S., Mazurek, M.L., Ur, B., Vidas, T., Bauer, L., Christin, N., Cranor, L.F., 2012. Correct horse battery staple: exploring the usability of system-assigned passphrases. In: Proceedings of the Eighth Symposium on Useable Privacy and Security. SOUPS '12. Association for Computing Machinery, New York, NY, USA, ISBN 9781450315326. https://doi.org/10.1145/2335356.2335366 doi:10.1145/2335356.2335366.

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 Trans. Inf. Syst. Secur. 18 (4), 1–34.

Tatlı, E.I., 2015. Cracking more password hashes with patterns. IEEE Trans. Inf. Forensics Secur. 10 (8), 1656–1665.

Thing, V.L., Ying, H.M., 2009. A novel time-memory trade-off method for password recovery. Digit. Invest. 6, S114–S120. https://doi.org/10.1016/j.diin.2009.06.004 (The Proceedings of the Ninth Annual DFRWS Conference). https://www.sciencedirect.com/science/article/pii/S1742287609000462.

Ur, B., Kelley, P.G., Komanduri, S., Lee, J., Maass, M., Mazurek, M.L., Passaro, T., Shay, R., Vidas, T., Bauer, L., et al., 2012. How does your password measure up? The effect of strength meters on password creation. In: Presented as Part of the 21st USENIX Security Symposium (USENIX Security 12), pp. 65–80.

Ur, B., Noma, F., Bees, J., Segreti, S.M., Shay, R., Bauer, L., Christin, N., Cranor, L.F., 2015. I added '!' at the end to make it secure": observing password creation in the lab. In: Eleventh Symposium on Useable Privacy and Security (SOUPS 2015), pp. 123–140.

Wang, D., Cheng, H., Wang, P., Huang, X., Jian, G., 2017. Zipf's law in passwords. IEEE Trans. Inf. Forensics Secur. 12 (11), 2776–2791.

Wang, D., Cheng, H., Wang, P., Yan, J., Huang, X., 2018. A security analysis of

honeywords. In: 25th Annual Network and Distributed System Security Symposium.
Wang, D., Wang, P., He, D., Tian, Y., 2019. Birthday, name and bifacial-security: understanding passwords of Chinese web users. In: 28th USENIX Security Symposium (USENIX Security 19), pp. 1537–1555.
Weir, M., Aggarwal, S., De Medeiros, B., Glodek, B., 2009. Password cracking using probabilistic context-free grammars. In: 2009 30th IEEE Symposium on Security and Privacy. IEEE, pp. 391–405.