Forensic Science International: Digital Investigation 44 (2023) 301507

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

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

DFRWS 2023 EU - Selected papers of the Tenth Annual DFRWS Europe Conference

Harder, better, faster, stronger: Optimising the performance of context-based password cracking dictionaries

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

[a] Forensics and Security Research Group, School of Computer Science, University College Dublin, Ireland
[b] Europol, Eisenhowerlaan 73, 2517 KK, The Hague, Netherlands

ARTICLE INFO

Article history:

Keywords:
Dictionary
Contextual information
Password cracking
Wordlist

ABSTRACT

Passwords have been the prevailing method of authentication since their inception more than 50 years ago, a trend which has no signs of slowing down in the foreseeable future. They are an integral part of the security of digital persons, systems and critical data, and yet, they often remain the weakest entry point to a digital system. A password itself is indeed an extension of its creator and therefore can be exploited by malicious actors leveraging available contextual information about a target password creator. Recent research has shown that bespoke password candidate lists, generated based on available contextual information, can positively impact the password cracking processes. This paper introduces an innovative methodology for composing a contextual wordlist and ranking the password candidates in order to maximise the chance of early success. The aim of the proposed approach is to support digital forensic investigators in their criminal investigation – especially when time is of the essence. This paper describes the implementation of this methodology and provides an overview of several experimental results demonstrating the advantages of this approach. These results demonstrate that by going through a *harder*, more rigorous password candidate selection process, *better* dictionaries can be generated that, in a *faster* timeframe, can crack *stronger* passwords.

1. Introduction

For the last few decades, research on passwords, their architecture, and the ways to crack them has been a focal point for researchers. This is with good reason, since they have been the most popular means of user authentication – and are set to continue to be so into the foreseeable future (Kanta et al., 2020a). Attackers refine and adapt their methods to account for the increasing diligence of companies and users who choose harder and stronger passwords, often times computer generated and/or salted.[1] By taking into account the increasing usage of computer generated passwords, the addition of salts, and the use of slower hashing functions, password cracking is increasingly becoming more of an uphill battle. This redoubles in the context of an attacker trying to gain access into a single account/point of entry, especially for an online system with a limited number of tries. In this case, simply brute forcing is out of the question – more sophisticated password cracking attacks need to be employed.

In the case of a law enforcement digital forensic investigation, the investigator could be faced with the encrypted system of a perpetrator, which can pose a significant hindrance to the investigation, or bring it to a halt entirely (Du et al., 2020). Suspects are not always inclined to share their passwords, especially if there is incriminating information in them. In many jurisdictions, law enforcement cannot compel that information from them (Ryder and Le-Khac, 2016). Furthermore, in a triage situation, where the discovery and processing of evidence in a timely manner is crucial to the outcome of the investigation, it becomes paramount to access suspect devices as quickly as possible. Therefore, generic approaches like a brute force attack, or an extensive dictionary search would not be suitable because of time constraints and other methods should be considered.

Research shows that the distribution of passwords of users is not uniform (Wang et al., 2017) and users tend to gravitate towards passwords that contain information that is personally connected to them (Wang et al., 2016). Many approaches take this information

[*] Corresponding author.
E-mail addresses: aikaterini.kanta@ucdconnect.ie (A. Kanta), iwen.coisel@europol.europa.eu (I. Coisel), mark.scanlon@ucd.ie (M. Scanlon).

[1] A salt is a string that is combined with the password plaintext and subsequently that combination is hashed, resulting in different values for the same password.

https://doi.org/10.1016/j.fsidi.2023.301507

into account, using leaked lists of real-world passwords for password cracking and/or mangling rules[2] to mimic user tendencies (Kanta et al., 2020a). One such user-centric method would be to focus on the individual whose password needs to be cracked, and more specifically, the information available about them through open source intelligence or other investigative means (Kanta et al., 2020b). This will result in an attack that is tailored to the individual target, which could return results when traditional methods fail.

In this case, it is essential for the investigator to have at their disposal the necessary bespoke dictionary lists, those that focus and contain password candidates that closely align with the suspect's interests and hobbies. To assemble such lists, natural language processing can be used, to create a concentrated body of words that align thematically with a starting seed word. The process of creating custom dictionary lists was introduced by Kanta et al. (2022).

1.1. Contribution of this work

In this paper, a methodology for optimising and ranking the candidates of a custom-made dictionary list is presented. This work aims to aid law enforcement investigators during criminal investigations by providing dictionary lists for password cracking that are tailored to the suspect. These are subsequently ranked so that more suitable password candidates are checked first, in a bid to save time during an investigation. The methodology described in this paper can become an important tool in an investigator's toolkit, by providing readily available, highly-customised contextual dictionaries on any topic – however niche. This approach is evaluated as part of this paper with data leaks stemming from compromised online communities focused on specific topics, as accessing a sufficient number of individual user's information is not possible. Nonetheless, the contextual approach proves itself valuable in finding many passwords that were not recovered with the baseline technique. These passwords are thematically close to the community itself, thus proving the role of context in password creation. Furthermore, the optimised, ranked dictionary lists as presented in this paper offer a significant increase compared to the models introduced in the past, as will be outlined in greater detail in Section 5.

To summarise, the contribution of this work includes.

- A thorough presentation of the optimisation and ranking techniques that were used for curating the contextual dictionaries.
- An example scenario focusing on a digital forensic triage use case is used in combination with the results presented to highlight the potential impact and improvement of contextual dictionaries over traditional password cracking methods.
- A detailed discussion section highlighting the uses, benefits and limitations of leveraging context in password cracking.

The rest of the paper is organised as follows: Section 2 offers an overview of the related work in the field. Section 3 provides the methodology that was used to create the optimised and ranked contextual dictionaries. Section 5 shows the result of putting this methodology to use across four different data leaks and provides an analysis of the results, and finally Sections 6 and 7 provide a summary and commentary of the results and outlines several avenues for future work.

2. Related work

The number of passwords users are required to remember nowadays has been steadily increasing for decades (Kanta et al., 2020a). This results in users being more inclined to seek easy ways to facilitate recalling them from memory. It is widely accepted that when users select their passwords, they follow certain patterns, using symbols and phrases familiar to them in an attempt to produce a "memorable" password (Wash et al., 2016a). This means that user-generated passwords are more susceptible to password cracking compared to machine-generated ones with random distribution of characters. To this end, various techniques have been developed to leverage that inherent weakness.

2.1. Password cracking techniques

For machine-generated, apparently randomly passwords, no cracking strategy would have an advantage over an exhaustive, brute-force search – where all combinations of a given alphabet, including digits and special characters, up to a predetermined length are tested. Exhaustive searches are guaranteed to work if maximum password length or limits for attempts are not defined – the only variable is time. But even with modern, state-of-the-art graphic card aided brute-forcing, if slower hash functions are used, the time it would take to perform is computationally infeasible, i.e., an exhaustive search could potentially take millions of years. Therefore, this approach is not efficient from both a computational and a time-limited point of view. To leverage the "weakness" of user generated passwords, a vast array of password cracking techniques have been developed. This varies from the more traditional, e.g., an off-the-shelf dictionary based attack, to the more recently developed, e.g., Rainbow Tables (Oechslin and Boneh, 2003) or Markov-based models (Narayanan and Shmatikov, 2005).

Rainbow tables are based on the idea of a time-memory tradeoff (TMTO), which focuses on pre-computing an almost exhaustive predefined search space of passwords. The main advantage of this approach is that these tables store a minimal amount of information, and thus, enable a fast lookup of a password if it exists in the predefined search space. This approach needs less computer processing time but more storage than an exhaustive search, which calculates the hash on every attempt. The use of salting, as a method of protecting the password by changing its hash according to the salt, has become a hindrance to rainbow tables. This is because the same password salted with a different salt will almost certainly produce a different hash – therefore resulting in an infinite number of combinations. Only if the investigator knows beforehand that the length of the password in question is small, can a rainbow table be considered a reasonable possibility. Many efforts have been made to improve this procedure, like focusing on precomputation using cryptanalytic TMTOs (Avoine et al., 2021).

Markov-based models are focused on reducing the search space (Narayanan and Shmatikov, 2005), or are used to produce password candidates in descending orders of likelihood (Dürmuth et al., 2015). Another recently used approach to make use of probabilistic context-free grammars – where Markov chains are used to calculate the probability for each grammar (Weir et al., 2009). Neural networks are also utilised for password cracking, offering the advantage that they do not require user input (Pal et al., 2019; Melicher et al., 2016). PassGAN, which uses a Generative Adversarial Network (GAN), is an example neural network used to create password candidates that closely mimic the distribution of real-world passwords (Hitaj et al., 2019).

2.2. Password selection and strength

Passwords have been used for decades to protect sensitive

[2] Mangling rules represent common behaviours and substitutions by users in their passwords. For example, adding numerical sequences at the end of the password or replacing alphabet letters by similarly looking numbers or symbols (i.e., '1' for 'i' or '@' for 'a').

information from adversaries. But in many cases, they can represent the weakest point of entry to a system and therefore be the target. This is because the password is often connected to its creator, which presents an inherent weakness that criminals look to profit from. Users tend to use their own personal, identifiable information when they create passwords as a way to more easily memorise them (Li et al., 2017).

Furthermore, and because users are required to memorise an abundance of passwords across the different services and devices they use, they tend to reuse the same passwords, verbatim or slightly modified versions of a “root” password (Florencio and Herley, 2007; Bonneau et al., 2012a). This practice, of course, creates a risk that even strong passwords can be rendered vulnerable if a password that has been reused has been somehow leaked, either through a phishing attack or as part of a data breach (Wash et al., 2016b).

Users often tend to overestimate the security of the passwords they create (Ur et al., 2016). Common practices that have been observed in passwords from data breaches show that users, when asked by a password policy to have uppercase characters in their password, they will most likely capitalise the first letter (Kanta et al., 2021a). Similarly, if it is required to include numbers in the password, usually they are found at the end of the passwords and are often sequences, or they represent dates (Kanta et al., 2021a). Even in cases where they are required to use passphrases as a means of authentication, that choice was far from random, as users tended to prefer simple noun bigrams (Bonneau et al., 2012b). Therefore, adhering to a password policy, might provide users a false sense of security (Ur et al., 2015).

Password strength meters are often used by web services to guide users and help them develop safer password creation habits. One of the most well known policies was introduced in 2013 by the National Institute of Standards and Technology (NIST)[3] and it requires passwords to be at least 8 character long, with uppercase, lowercase, digits and special characters included (Grassi et al., 2017). But even when these policies are enforced, users still try to bypass them in favour of memorability. For example, if a web service requires a password to be changed every six months, users might keep the same password by adding a *1* at the end (Kanta et al., 2021a).

The password strength meters that are now in use have also evolved to anticipate this behaviour by users and often detect and disallow passwords that contain the same basic structure as previously used (Shay et al., 2015). There are many strength meters available and in use by various web services, some based on Markov Models (Castelluccia et al., 2012), Natural Language Processing (Guo and Zhang, 2018) and Deep Learning (Pasquini et al., 2020). It has also been shown, that the results found by the various password strength meters when evaluating the same passwords have been widely inconsistent (de Carné de Carnavalet and Mannan, 2014).

3. Methodology for ranking and optimising contextual dictionaries

3.1. Selection of evaluation and control datasets

Access to the content of the encrypted devices of a suspect can be crucial for the outcome of the investigation (Sayakkara et al., 2018). The timeliness of accessing potentially case-progressing information can be paramount in certain scenarios, e.g., kidnapping cases or an imminent terrorist attack investigation. Investigators might seek alternative methods of password cracking in these specific scenarios aimed at minimising the duration of the process. One viable alternative approach is to leverage the role of context in a user’s password selection. In order to prove its viability, an experimental methodology is presented in this section, focusing on the creation, ranking and optimisation of bespoke dictionaries for specific topics.

Table 1
Size of datasets.

Dataset	Size
AxeMusic	252,752
JeepForum	239.347
Wattpad	23,531,304
MangaTraders	618,237

Ideally, the evaluation of the proposed methodology would include testing the contextual dictionaries against specific targets during the course of an investigation. For example, if a digital investigator wanted to access the encrypted device of a suspect who was known to be a fan of rock music, football and tennis, a dictionary could be created using these topics as seed words. Unfortunately, for data protection and ethical purposes, access to this privileged information is not possible. Therefore, the approach for evaluation is focused on communities’ passwords as opposed to that of individuals. For this purpose, four different data leaks have been selected from four communities, about music, cars, fanfiction and manga. These datasets and their sizes can be found in Table 1 and contain the passwords from the leaks without any other identifiable information, i.e., the datasets used do no contain usernames, e-mail addresses, phone numbers, etc. Approval for use of these datasets has been given by the Office of Research Ethics of [redacted for blind review].

As a baseline to compare this approach against existing ones, a dictionary named Ignis-10M[4] has been selected. Ignis contains 10 million passwords from a variety of data leaks and was assembled in 2020. The reason Ignis has been selected instead of the popular *RockYou* dictionary list is that although the passwords of RockYou have been leaked as plaintext and therefore represent a more accurate account of real-life passwords, RockYou was leaked in 2009. Password policies have evolved significantly since then, i.e., password policies have become stricter regarding their requirements — with a larger minimum length and a mix of upper and lowercase characters, numbers, and symbols often all being required. Furthermore, according to the creator of Ignis, when looking at the Top 1000 passwords in Ignis-10M and RockYou, 411 passwords of Ignis were not in RockYou’s Top 1000. This is likely due to RockYou only containing passwords created up to 2009. For example, “Minecraft”, which is a Top 1000 password, does not exist in RockYou due to the game being released in 2011. Using Ignis-10M as the dataset to compare this approach against provides the most up-to-date baseline.

3.2. Contextual dictionary generation

The method used for creating the contextual dictionaries as part of this work is described in Kanta et al. (2022). This method starts with selecting a seed word that is thematically close to the community from which the leaked dataset stems is picked. This seed word is then used with DBPedia,[5] a crowd-sourced knowledge graph database version of Wikipedia, in order to visit the corresponding DBPedia entry and gather the titles of all the other articles

[3] https://pages.nist.gov/800-63-3/sp800-63b.html.

[4] https://github.com/ignis-sec/Pwdb-Public.

[5] https://www.dbpedia.org/.

Table 2
The DBPedia dictionaries.

Seed Word	Size	Corresponding Dataset
Music	1,001,173	AxeMusic
Car	853,825	JeepForum
Fanfiction	641,007	Wattpad
Manga	6,348,947	MangaTraders

it links to. As part of this work, this is referred to as “Layer 1”. The reason for this is that in any given Wikipedia article, the important links are usually also keywords relating to the given article. After all these entries are collected, they are in turn visited and the new articles are also processed and added to the total, thus creating Layer 2. This process can go on until the required depth of search is reached. The end product of this process is then sanitised and can be used as the input wordlist in a dictionary attack.

Of course, the deeper the traversal down the DBPedia tree, the less thematically close the new keywords are and the size of the list increases exponentially. It is therefore important to be able to reach a satisfying length for the wordlist, while also making sure the terms are still thematically relevant to the seed word. For this reason, both the parameters of size of the wordlist and thematical distance to the seed word were explored.

3.3. Size of the wordlist

It has been observed that the size of the wordlist in a dictionary attack plays an important role in the percentage of found passwords (Kanta et al., 2022; Bošnjak et al., 2018). In fact, the larger the dictionary list, given an infinite amount of time and permutations, the more passwords will be cracked. This is why given infinite time, a brute force attack is guaranteed to work. In this case, the depth of traversal in DBPedia for all four seed words was chosen to be either Layer 3 or 4. More specifically, Music, Car and Fanfiction, being larger Wikipedia articles with more links, were chosen to be Layer 3, while Manga was chosen to be Layer 4. The reason for this was that Layer 3 for Manga contained only ~180,000 candidates and considerably under-performed compared to the equivalent of Layer 4. The sizes of the produced dictionaries can be found in Table 2.

3.4. Thematical distance

Another important aspect of the creation of bespoke wordlists on certain topics, was making sure that the words were indeed thematically close to the seed word. In order to ensure this, the natural language model `Wikipedia2Vec` was used (Yamada et al., 2020). `Wikipedia2Vec` is a NLP model based on `Word2Vec` (Mikolov et al., 2013). `Word2Vec` can compute vector representations (referred to as embeddings) of words relying mostly on the surrounding context present in the training dataset. It relies on the Harris’ “Distributional Hypothesis” stating that words that occur in the same contexts tend to have similar meanings. The word embeddings can subsequently be used to estimate the similarity of the context in which they have appear in the training dataset and therefore similarity in their meaning. `Wikipedia2Vec` provides embeddings not only for words, but also for entities, i.e., entries that have corresponding articles on Wikipedia. For this purpose, a pretrained embeddings model of Wikipedia in English was used.[6] Using `Wikipedia2Vec`, the similarity of each word of the bespoke wordlists can be evaluated. This evaluation returns a similarity score according to how close the embeddings are in vector space, i.e., a score of 1 would mean they are identical.

With this similarity score in hand, the words in the wordlist are ranked accordingly with the seed word, from the highest similarity score to the lowest. This means that not only will the words that are higher on the list be checked first, but also more permutations of them with mangling rules will be checked during the attack. At this stage a threshold can be set for the similarity score, for example words below a certain threshold could be considered as irrelevant to the seed word and therefore disregarded.

3.5. Words vs. entities

As described above, the source of the entries in the wordlist are Wikipedia articles, linked to the seed word, either directly or through other articles. During the sanitation process, many of these are disregarded due to their format, e.g., an image name is a link but not very useful for a dictionary attack. From the remaining entries, some are single words and some are phrases/entities. Entities have embeddings in `Wikipedia2Vec`, and therefore a similarity score can be computed for them as well — resulting in a more complete ranking of all the wordlist entries. However, some entries are not in the training model and therefore a similarity score cannot be computed. Two avenues were explored to deal with this issue. One was to compute the average similarity score for each word in the entity, and the other was to assign the score of the word that was closest to the seed word to the entire phrase, i.e., the maximum. For example, if the seed word was “Shopping”, the phrase “Window Shopping” would be assigned a score of 1.

However, as can be seen from the above example, while “window shopping” is very relevant to “shopping”, it does not seem like a very likely password. Therefore, two ranked versions of the wordlists were produced. In the first version, phrases were also contained in the list, and they were ranked as described above. In the second version, the phrases were split into single words and then ranked (duplicates and stopwords were removed).

3.6. Quality of the dictionary list

Evaluating a dictionary list is a complex topic and there are many parameters to take into consideration (Kanta et al., 2021b). For example, a larger dictionary list can achieve a higher percentage of found passwords, but in twice as much time as a smaller list. Alternatively, two lists can have the same size, a similar run-time, and achieve similar success rates, but one of them can find passwords of higher difficulty. Therefore, this trade-off should be considered on a case-by-case scenario. For an offline attack where the percentage of success is important, a bigger, more thorough dictionary list might be chosen and paired up with an extensive set of rules for permutations. If time is of the essence, a smaller dictionary list might be more beneficial. If the target is a single password, a combination of brute-forcing the smaller passwords alongside a contextual dictionary list focusing on harder passwords might be an optimal strategy. The dictionary list, or combination thereof, should be decided depending on the parameters of the specific case.

4. Experimental scenario

For the purpose of this paper, a digital investigation triage scenario is selected, where the investigator needs to access the data on an encrypted device as soon as possible. There are many dictionary lists in existence, given a more relaxed time frame and an abundance of resources, that can perform very well (especially when looking at sheer numbers of cracked passwords). However, it is the performance against one (possibly harder) password where the speed of cracking is of the essence. Therefore, a limited execution

[6] https://wikipedia2vec.github.io/wikipedia2vec/pretrained/.

time of 15 min was selected for these attacks. A key evaluation point for the proposed approach is how well each dictionary performs against stronger, harder-to-crack passwords. It should be mentioned here that during this 15 min process, more than 10 billion password candidates are evaluated. This is achievable as the data leaks used for evaluation are in plaintext. To provide an indication of the runtime for this proposed approach for hash-based password cracking, assuming a Veracrypt full disk encryption was targeting with an attack leveraging the latest Nvidia RTX 4090 GPU[7] running at 6.6 kH/s, evaluating the same number of candidates with the single GPU would take approximately 20 days.

4.1. Setup of the experiment

Our experiments compare three different dictionaries, namely: Ignis-10M and the ranked and the unranked version of the dictionaries produced by the seed words in Table 2. Before ranking, both versions of the unranked dictionary with either whole entities or split up to individual words, as described in Section 3.5, were evaluated. The version containing only individual words performed better and was therefore selected for comparison.

These dictionaries were then evaluated with the same password mangling rule file. Each rule in the file is a common modification users choose when they create their passwords, e.g., adding numbers at the end of their password, replacing some letters with similar looking numbers, etc. One of the most well-known rulesets is best64.[8] For this experiment, a larger ruleset was chosen, `OneRuleToRuleThemAll`.[9] This ruleset contains the top 25% performing rules from several component rulesets, concatenated together and without duplicates.

Finally, the password cracking was conducted with `hashcat`[10], which is an open-source password cracking tool.

5. Results

As mentioned in Section 3.2, four datasets stemming from data leaks centring on cars, music, manga and fanfiction were selected. For each topic, a contextual dictionary was produced starting from each seed word, which represents the unranked version. The ranked version was then produced with the methodology described above in Section 3. As defined in the previous section, for three topics, the produced dictionaries were of 3 layers depth and for Manga, it was 4 layers deep. Manga was selected for an additional layer over the other three, as a three-layered dictionary from the seed word "manga" was not sufficiently big for this attack (and indeed performed poorly – especially compared to Ignis-10M).

5.1. Success over time

The cracking progress over time against these four data leaks, with the baseline Ignis-10M dictionary, the contextual ranked dictionaries, and the contextual unranked dictionaries can be seen in Fig. 1(a) to 1(d). As can be seen, the baseline dictionary, Ignis-10M, has the best overall performance. This does not come as a surprise – not only because Ignis-10M is larger and more diverse than any of the contextual dictionaries, but also because even in a data leak stemming from a car-related forum, not all passwords would be car related. Nonetheless, it can still be observed that the contextual dictionaries perform well, especially the ranked dictionary for the JeepForum dataset.

Table 3 shows the overall number of passwords found by each dictionary and for each data leak. It can be observed that Ignis-10M and the Music_R (representing the ranked version of the dictionary created with "music" as the seed word) have very similar performances, which is a positive outcome considering the size and variety of real-world passwords in Ignis-10M. The same holds true for Ignis-10M and Car_R. It is worthy to note that size wise, the dictionaries produced by the seed words "car" and "music" were the two smallest, as can be seen in Table 1. It can also be observed that in all four categories, the ranked versions have outperformed the unranked ones, most strikingly in the Wattpad leak – where ranking resulted in an increase of 27.44% in performance.

Table 3 also shows the passwords that have been found exclusively by the ranked and unranked dictionaries for each topic. This is especially valuable if a combination attack is considered, i.e., where Ignis-10M is first used to target the weaker, more common passwords and subsequently the targeted contextual dictionary is employed (or indeed, both ran in parallel across different workstations). In this case, the improvement offered by the contextual dictionaries over Ignis-10M alone is significant across the board – with once again, the ranked dictionaries outperforming the unranked ones. This is especially true in the case of Wattpad, there are more than a quarter of a million of new passwords, exclusively found by Fanfiction_R that were not found by Ignis-10M. This represents an increase of 6.55%.

The number of passwords found exclusively by the contextual dictionaries leads to a new and interesting question. Which password candidates in the dictionary list performed better, i.e., which found the most passwords in their respective data leaks? Table 4 shows the top 20 password candidates that found the most passwords by the ranked dictionaries across all four topics. As can be seen, the top password candidate for Music_R is the word "music" and the rest of the top 5 are also words relating to music. In fact, 14 out of the top 20 best performing password candidates for AxeMusic are music related – something that reinforces the theory that users pick passwords according to their interests, and also the type of website the password is aimed for. Similar results can be observed for "car" and "manga", with 13 out of 20 password candidates in Car_R being related to cars (this is excluding "er1", which represents a non-mainstream concept car model). This also holds true for MangaTraders with 13 out of the top 20 performing password candidates being related to manga. For Wattpad, the results are not quite as clear, with many first names and dates appearing in the top 20 performing candidates – something which is common in most data leaks (Veras et al., 2012).

5.2. Strength of found passwords

Returning to the aforementioned digital forensic triage scenario, if a digital investigator is looking to crack the password of an encrypted device belonging to a suspect in a timely manner, looking at the number of passwords cracked per dictionary attack might not be a sufficiently accurate metric. Ignis-10M, since it is compiled of some of the most popular passwords from several data leaks, is assumed to do well with common, popular passwords. But if the holder of the encrypted device is someone more tech-savvy, reason states that their password might not be one to be found on these popular password lists.

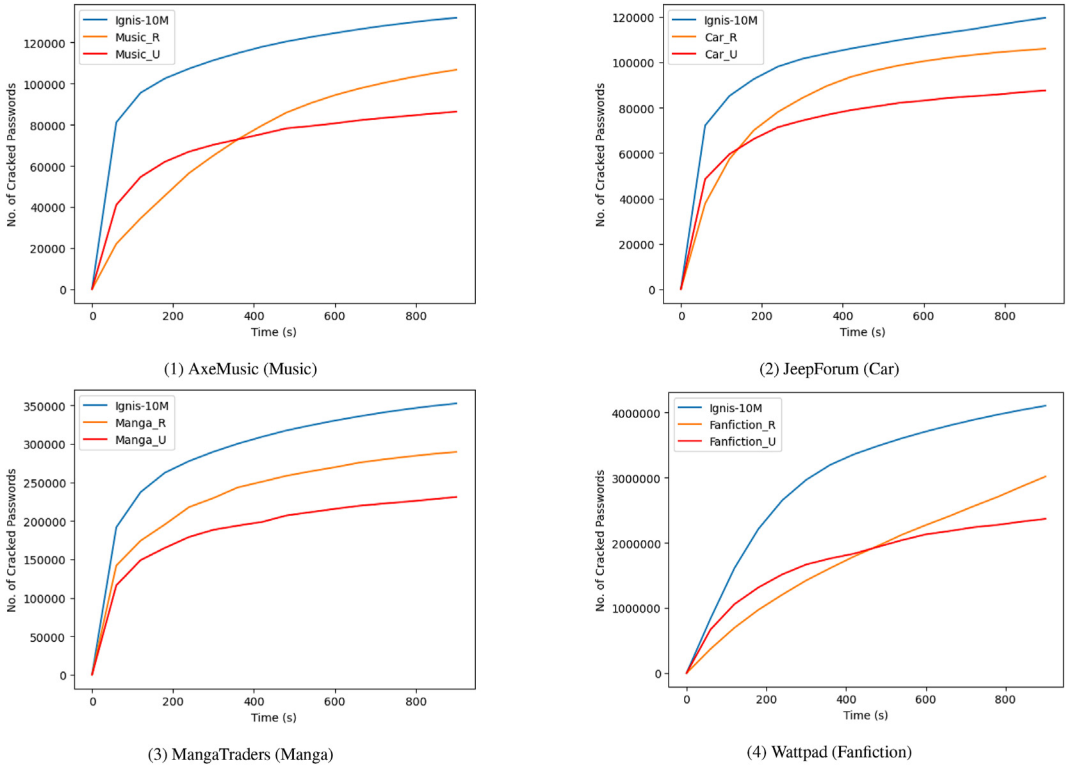
In a triage situation, it is therefore important to take into account the difficulty of the passwords that each dictionary attack successfully cracked. To determining this, the password strength meter `zxcvbn` (Wheeler, 2016) was employed. `zxcvbn` classifies passwords according to their strength and places them in five classes, ranging from the easiest to crack (Class 0) to the hardest (Class 4).

[7] https://www.nvidia.com/en-gb/geforce/graphics-cards/40-series/rtx-4090/.

[8] https://github.com/hashcat/hashcat/blob/master/rules/best64.rule.

[9] https://notsosecure.com/one-rule-to-rule-them-all.

[10] https://hashcat.net/hashcat/.



(1) AxeMusic (Music)

(2) JeepForum (Car)

(3) MangaTraders (Manga)

(4) Wattpad (Fanfiction)

Fig. 1. Number of passwords cracked over time.

Table 3
Total number of passwords found. The R Excl. Column contains passwords found only by ranked dictionaries, The U Excl. Column contains passwords found only by unranked dictionaries.

Dataset	Ignis-10M	R	R Excl.	U	U Excl.
AxeMusic	132,009	106,782	7,773	86,384	2,698
JeepForum	122,061	107,365	6,025	89,001	2,212
Wattpad	4,103,525	3,016,762	268,670	2,367,223	86,267
MangaTraders	352,544	289,573	22,128	231,097	9,635

Table 4
Top 20 password candidates for the four ranked dictionaries.

Music	Car	Fanfiction	Manga
music	jeep	love	qwerty
guitar	dog	angel	sakura
guitaro	man	password	naruto
rock	harley	qwerty	pokemon
longy	honda	100	dragon
sunshine	1	1997	manga
piano	ford	4ever	inuyasha
love	s1	bella	angel
musical	wrangler	1996	sasuke
12	chevy	monkey	anime
singer	car	1995	iloveyou
welcome	camaro	princess	hello
boy	mustang	kitty	pikachu
yamaha	er1	alex	monkey
song	12	forever	shadow
blues	dodge	nicole	chobits
drum	bike	lover	vampire
guitars	qwerty	girl	purple
1	ranger	hannah	gundam
rockstar	hummer	soccer	akira

This classification for each of the four data leaks with Ignis-10M and the ranked and unranked dictionaries are shown in Fig. 2.

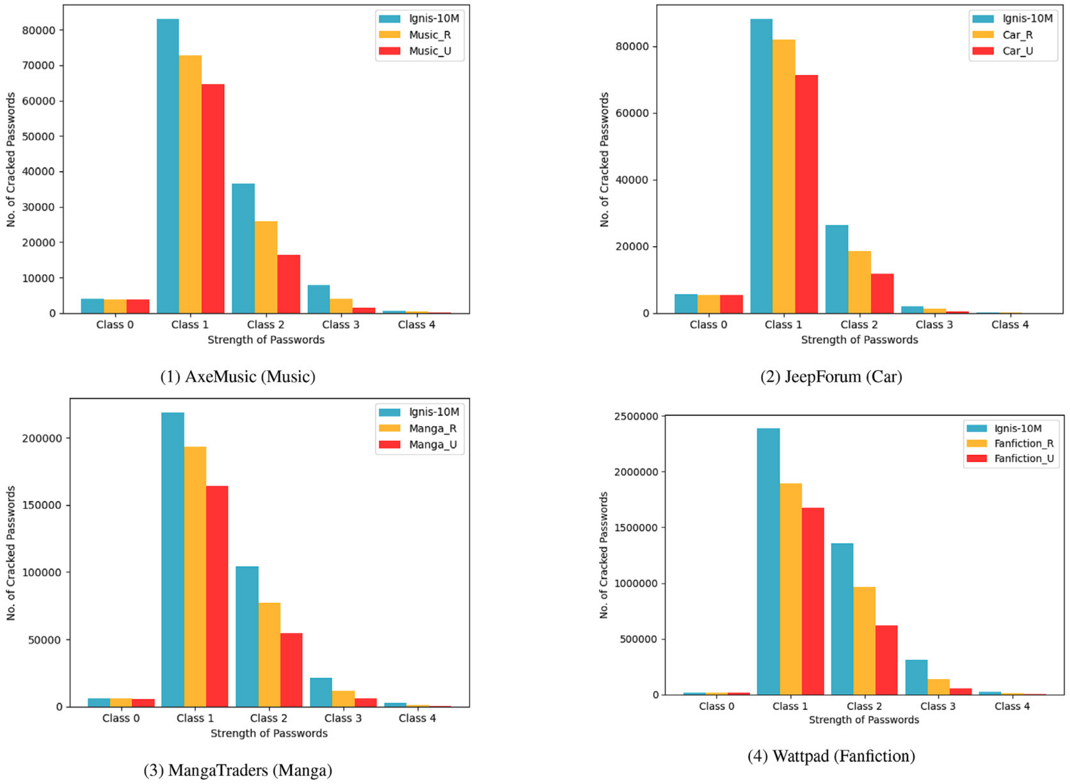
As can be seen in Fig. 2(a) to 2(d), most of the passwords have been assigned to Class 1 – with Class 2 being the second most common. It is generally assumed that the passwords up to Class 2 are easier to crack, and most current password cracking methods would be able to crack them (Kanta et al., 2021a). Therefore, the focus is mostly on the passwords belonging to Class 3 and Class 4. Tables 5 and 6 show the passwords of Class 3 and Class 4 respectively, which were cracked by Ignis-10M, the ranked, and the unranked context-based dictionaries. It can be observed that once again, the ranked dictionaries have a better performance compared to the unranked ones across all four datasets. In fact, in every case except Class 3 for MangaTraders and Wattpad, the ranked dictionaries have managed to find more exclusive passwords (R Excl column) than the unranked have managed overall (U column).

When comparing the ranked dictionaries to Ignis-10M, it is important to notice that the number of passwords found exclusively by the ranked dictionaries, i.e., not found using Ignis-10M, is quite high. In fact, for Class 3, the increase in password cracking success rises 20.9%, 27.8%, 15.9% and 20.5% for AxeMusic, JeepForum, Wattpad, and MangaTraders respectively.

Focusing on Class 4, which contains the strongest passwords of each dataset, on average 50% of those found by the ranked dictionaries are not found using Ignis-10M. In a combination attack, i.e., combining the results of the ranked dictionaries and the corresponding results from Ignis-10M, the improvement is 43.4%, 52.3%, 20% and 24.3% for Axe Music, JeepForum, Wattpad, and Manga Traders respectively.

6. Discussion

The results of the previous section show the value of considering context in password cracking. The number of passwords found



(1) AxeMusic (Music)

(2) JeepForum (Car)

(3) MangaTraders (Manga)

(4) Wattpad (Fanfiction)

Fig. 2. Strength of passwords cracked.

Table 5
Class 3 passwords classified using zxcvbn for Ignis-10M, ranked (R) and unranked (U) dictionaries. The columns R Excl. and U Excl. represent the Class 3 passwords found exclusively by the R and U dictionaries.

	Ignis-10M	R	U	R Excl.	U Excl.
AxeMusic	7,879	4,003	1,490	1,645	393
JeepForum	2,039	1,305	491	566	140
Wattpad	313,142	137,396	52,882	49,742	10,400
MangaTraders	21,293	11,739	5,981	4,357	1,645

Table 6
Class 4 passwords classified using zxcvbn for Ignis, ranked (R) and unranked (U) dictionaries. The columns R Excl. and U Excl. represent the Class 4 passwords found exclusively by the R and U dictionaries.

	Ignis-10M	R	U	R Excl.	U Excl.
AxeMusic	551	380	118	239	66
JeepForum	65	53	15	34	12
Wattpad	27,005	9,346	2,628	5,415	1,128
MangaTraders	2,389	1,245	574	581	240

exclusively by the unranked and especially the ranked versions of the contextual dictionaries adds a substantial value to a combination password cracking approach with existing off-the-shelf dictionaries, e.g., Ignis-10M. When cracking the passwords of a large community focused on a specific topic, a generic dictionary like Ignis-10M, that contains some of the most popular passwords found on several data leaks, will always be at an advantage. This is due to many users choosing passwords that are thematically close to the purpose of the website the password is for, and many will still use passwords that either have some personal meaning or without any contextual meaning at all.

In the presented experiments on data leaks from communities focused on specific topics, it is clear that the link between the password and the purpose of the community is sufficiently present. This can be seen clearly in Table 4, where the majority of the top performing password candidates were thematically close to the seed word/focus of the community.

Furthermore, the process to optimise and rank the contextual dictionaries has proved fruitful, with the ranked dictionaries outperforming the unranked ones across the board. This is especially significant in triage-like situations during a digital investigation, where it is important to gain access to an encrypted device as quickly as possible. Ranking the dictionary by how similar the password candidates are to the seed word means that those passwords (and their corresponding permutations with mangling rules) will be checked first. In a timed attack, as has been the case with the experiments here, this proves extremely important.

Of course, depending on the specific situation, a combination of one or more approaches might be needed. For example, depending on the hash function, an exhaustive search up to 8 digits might be fast enough to be considered first, followed by a contextual dictionary attack if the "low-hanging fruit approach" does not prove so fruitful.

6.1. Benefits and limitations

As with any approach, there are advantages and limitations. One of the advantages of the contextual approach is that it is highly customisable to each suspect. A dictionary can be made with any starting seed word (as long as there exists a Wikipedia article about it). This practically means that an investigator could easily have dictionaries about specific or niche topics at their disposal easily. These dictionaries can also be highly customisable — the depth of search can be set by the investigator, and entries that are deemed as contextually distant to the seed word can be disregarded by tweaking the threshold for the similarity score. Furthermore, dictionaries stemming from different seed words can be combined to create a combination dictionary.

Another advantage of this approach is that these dictionaries do not need to be produced again and again for every case. In fact, the investigator can have on hand dictionaries about frequently encountered topics and therefore skip the dictionary creation step, which could again save crucial time during a triage situation.

The importance of these dictionaries hinges on not only the fact that users tend to form passwords that are meaningful to them, therefore memorable, but also the highly likely assumption that if a suspect is tech-savvy enough to use encryption on their devices, they are also likely to not use easily guessed passwords.

Of course, as with every approach, there are limitations to its usability. In a scenario where the sheer number of passwords found is the most important parameter and the runtime and/or strength of the passwords found is not important, generic dictionaries based on existing password leaks will most likely perform better. Nonetheless, a combined approach with the technique described as part of this paper will likely improve the chances of overall success further.

Furthermore, contextual dictionaries, unlike common password lists such as Ignis-10M and RockYou, are lists of words not lists of passwords. Mangling rules can help remedy this to an extent, but it is safe to say that many words that can have a high similarity score to the seed word and therefore be placed high during the ranking are not words that would be used to create a password. One such example is the word “series”. Using a contextual dictionary with phrases and the proposed ranking approach, “Manga Series” has a similarity score of 1 compared to “Manga” (1 for “Manga” and 0.54 for “Series”), which would place it at the top of the list. But in reality, the phrase “Manga Series” is not as likely to be a password as the names of actual manga series, as evidenced in Table 4.

7. Conclusion and future work

The contribution offers the outline for a methodology for creating custom dictionary lists on any choice of topics, ones that can be useful to a digital investigator for cracking the password of an encrypted device. This methodology leverages natural language processing to create bespoke dictionary lists. The entries are ranked in descending order of similarity to the seed word that was used to create the dictionary, with the aim to try the most *likely* password candidates first. This is especially useful when the timely access to an encrypted device is of the essence, as the candidates with the highest similarity score will be checked first.

As outlined in Section 5, the use of these contextual dictionaries can compete with much larger and variant dictionary lists, such as Ignis-10M, can offer a significant increase in found passwords if the generic and contextual approach are combined. The contribution of the contextual dictionaries is particularly important for Class 3 and Class 4 passwords, where the increase in found passwords by adding the ranked contextual dictionary in addition to Ignis-10M resulted in as many as 50% more passwords found. This is especially significant considering how the size of the bespoke dictionaries is much smaller to more well-rounded password dictionary lists.

For a digital forensic investigation, pre-computed bespoke dictionaries can be a very useful tool for an investigator as they perform well against harder to crack passwords and their length, depth and relevancy to the seed word can be easily customised by the investigator as needed.

7.1. Future work

Addressing some of the limitations outlined in Section 6.1 is a good starting point for future work on the topic. In fact, more attention should be given into filtering the password candidates such that candidates with high similarity to the seed word, but low probability of being used as passwords, can be filtered out. A potential way to do this would be to look at the bidirectional distance between these two words. For example, is “Series” as close to “Manga” as “Manga” is to “Series”? Or, is the number of words that are thematically closer to “Manga” than “Series” the same for “Series” to “Manga”? To this end, looking back at `Layer n-1` might also prove useful, as it will contain links that link back to “Manga”.

Furthermore, even though the contextual dictionaries without phrases outperformed those containing phrases, it is possible that contextual information is going to be lost by not also keeping phrases together. For example, in a contextual dictionary with the seed word of “Manga”, “ManofSteel” is more likely to be someone’s password than the individual words “man” and “steel”. At the same time, there are phrase entries that do not warrant further consideration, e.g., the link “List of Manga Series”, which is a Wikipedia display construct for storing similar content, but is not a valid candidate phrase by itself. Therefore, a more robust sanitation process could greatly benefit the success of the contextual dictionaries. This sanitation process could again be based on NLP, with the similarity of the words within the phrase to each other being taken into account to further refine the password candidates extracted.

References

Avoine, G., Carpent, X., Leblanc-Albarel, D., 2021. Precomputation for rainbow tables has never been so fast. In: Bertino, E., Shulman, H., Waidner, M. (Eds.), Computer Security – ESORICS 2021. Springer International Publishing, Cham, ISBN 978-3-030-88428-4, pp. 215–234.

Bonneau, J., Herley, C., Oorschot, P.C.v., Stajano, F., 2012a. The quest to replace passwords: a framework for comparative evaluation of web authentication schemes. In: 2012 IEEE Symposium on Security and Privacy, pp. 553–567. https://doi.org/10.1109/SP.2012.44.

Bonneau, J., Shutova, E., 2012b. 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.

Bošnjak, L., Sreš, J., Brumen, B., 2018. Brute-force and dictionary attack on hashed real-world passwords. In: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics. MIPRO, pp. 1161–1166. https://doi.org/10.23919/MIPRO.2018.8400211.

Castelluccia, C., Dürmuth, M., Perito, D., 2012. Adaptive Password-Strength Meters from Markov Models. NDSS.

de Carné de Carnavalet, X., Mannan, M., 2014. From very weak to very strong: analyzing password-strength meters. In: Network and Distributed System Security (NDSS) Symposium 2014. Internet Society (in press). https://spectrum.library.concordia.ca/id/eprint/978105/;.

Du, X., Hargreaves, C., Sheppard, J., Anda, F., Sayakkara, A., Le-Khac, N.A., Scanlon, M., 2020. SoK: 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: Piessens, F., Caballero, J., Bielova, N. (Eds.), Engineering Secure Software and Systems. Springer International Publishing, Cham, ISBN 978-3-319-15618-7, pp. 119–132.

Florencio, D., Herley, C., 2007. A large-scale study of web password habits. In: Proceedings of the 16th International Conference on World Wide Web. Association for Computing Machinery, pp. 657–666. https://doi.org/10.1145/1242572.1242661.

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 800-63B - Digital Identity Guidelines: Authentication and Lifecycle Management. Tech. Rep. National Institute for Standards and Technology. https://doi.org/10.6028/NIST.SP.800-63b.

Guo, Y., Zhang, Z., 2018. Lpse: lightweight password-strength estimation for password meters. Computers and Security, 73, 507–518. https://doi.org/10.1016/j.cose.2017.07.012. URL: https://www.sciencedirect.com/science/article/pii/S0167404817301530.

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.

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. ISSN 2666-2817.

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), vols. 1–2. IEEE.

Kanta, A., Coray, S., Coisel, I., Scanlon, M., 2021a. How viable is password cracking in digital forensic investigation? Analyzing the guessability of over 3.9 billion real-world accounts. Forensic Sci. Int. Digit. Invest. 37, 301186. https://doi.org/10.1016/j.fsidi.2021.301186. ISSN 2666-2817.

Kanta, A., Coisel, I., Scanlon, M., 2021b. PCWQ: a framework for evaluating password cracking wordlist quality. In: *The 12th EAI International Conference On Digital Forensics And Cyber Crime*. ICDF2C '21. Springer, New York, NY, USA.

Kanta, A., Coisel, I., Scanlon, M., 2022. A novel dictionary generation methodology for contextual-based password cracking. IEEE Access 10, 59178–59188. https://doi.org/10.1109/ACCESS.2022.3179701.

Li, Y., Wang, H., Sun, K., 2017. Personal information in passwords and its security implications. IEEE Trans. Inf. Forensics Secur. 12 (10), 2320–2333.

Melicher, W., Ur, B., Segreti, S.M., Komanduri, S., Bauer, L., Christin, N., Cranor, L.F., 2016. Fast, lean, and accurate: Modeling password guessability using neural networks. In: 25th USENIX Security Symposium (USENIX Security 16). USENIX Association, Austin, TX, ISBN 978-1-931971-32-4, pp. 175–191. URL: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/melicher.

Mikolov, T., Chen, K., Corrado, G.S., Dean, J., 2013. Efficient estimation of word representations in vector space. URL: http://arxiv.org/abs/1301.3781.

Narayanan, A., Shmatikov, V., 2005. Fast dictionary attacks on passwords using time-space tradeoff. In: Proceedings of the 12th ACM Conference on Computer and Communications Security. CCS '05. Association for Computing Machinery, New York, NY, USA, ISBN 1595932267, pp. 364–372. https://doi.org/10.1145/1102120.1102168. URL:

Oechslin, P., 2003. Making a faster cryptanalytic time-memory trade-off. In: Boneh, D. (Ed.), Advances in Cryptology - CRYPTO 2003. Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN 978-3-540-45146-4, pp. 617–630.

Pal, B., Daniel, T., Chatterjee, R., Ristenpart, T., 2019. Beyond credential stuffing: password similarity models using neural networks. In: 2019 IEEE Symposium on Security and Privacy. SP, pp. 417–434. https://doi.org/10.1109/SP.2019.00056.

Pasquini, D., Ateniese, G., Bernaschi, M., 2020. Interpretable probabilistic password strength meters via deep learning. In: Chen, L., Li, N., Liang, K., Schneider, S. (Eds.), Computer Security – ESORICS 2020. Springer International Publishing, Cham, ISBN 978-3-030-58951-6, pp. 502–522.

Ryder, S., Le-Khac, N.A., 2016. The end of effective law enforcement in the cloud? - to encrypt, or not to encrypt. In: 2016 IEEE 9th International Conference on Cloud Computing. CLOUD, pp. 904–907. https://doi.org/10.1109/CLOUD.2016.0133.

Sayakkara, A., Le-Khac, N.A., Scanlon, M., 2018. Electromagnetic side-channel attacks: Potential for progressing hindered digital forensic analysis. In: *Companion Proceedings For the ISSTA/ECOOP 2018 Workshops*. ISSTA '18. Association for Computing Machinery, New York, NY, USA, ISBN 9781450359399, pp. 138–143. https://doi.org/10.1145/3236454.3236512. URL:

Shay, R., Bauer, L., Christin, N., Cranor, L.F., Forget, A., Komanduri, S., Mazurek, M.L., Melicher, W., Segreti, S.M., Ur, B., 2015. A spoonful of sugar? the impact of guidance and feedback on password-creation behavior. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, pp. 2903–2912.

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.

Ur, B., Bees, J., Segreti, S.M., Bauer, L., Christin, N., Cranor, L.F., 2016. Do users' perceptions of password security match reality?. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. CHI '16;. Association for Computing Machinery, New York, NY, USA, ISBN 9781450333627, pp. 3748–3760. https://doi.org/10.1145/2858036.2858546. URL:

Veras, R., Thorpe, J., Collins, C., 2012. Visualizing semantics in passwords: the role of dates. In: Proceedings of the Ninth International Symposium on Visualization for Cyber Security. VizSec '12. Association for Computing Machinery, New York, NY, USA, ISBN 9781450314138, pp. 88–95. https://doi.org/10.1145/2379690.2379702. URL:

Wang, D., Zhang, Z., Wang, P., Yan, J., Huang, X., 2016. Targeted online password guessing: an underestimated threat. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. CCS '16. Association for Computing Machinery, New York, NY, USA, ISBN 9781450341394, pp. 1242–1254. https://doi.org/10.1145/2976749.2978339. URL:

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. https://doi.org/10.1109/TIFS.2017.2721359.

Wash, R., Rader, E., Berman, R., Wellmer, Z., 2016a. Understanding password choices: how frequently entered passwords are Re-used across websites. In: Twelfth Symposium on Useable Privacy and Security (SOUPS 2016). USENIX Association, Denver, CO, ISBN 978-1-931971-31-7, pp. 175–188. URL: https://www.usenix.org/conference/soups2016/technical-sessions/presentation/wash.

Wash, R., Rader, E., Berman, R., Wellmer, Z., 2016b. Understanding password choices: how frequently entered passwords are re-used across websites. In: Twelfth Symposium on Useable Privacy and Security. SOUPS 2016, pp. 175–188.

Weir, M., Aggarwal, S., Medeiros, B.d., Glodek, B., 2009. Password cracking using probabilistic context-free grammars. In: 2009 30th IEEE Symposium on Security and Privacy, pp. 391–405. https://doi.org/10.1109/SP.2009.8.

Wheeler, D.L., 2016. zxcvbn: Low-budget password strength estimation. In: 25th USENIX Security Symposium, vol. 16. USENIX Security, pp. 157–173.

Yamada, I., Asai, A., Sakuma, J., Shindo, H., Takeda, H., Takefuji, Y., Matsumoto, Y., 2020. Wikipedia2Vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from Wikipedia. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Association for Computational Linguistics, pp. 23–30.