A Novel Dictionary Generation Methodology for Contextual-Based Password Cracking

AIKATERINI KANTA¹, (Graduate Student Member, IEEE), IWEN COISEL¹, AND MARK SCANLON², (Senior Member, IEEE)

¹European Commission Joint Research Centre (DG JRC), 21027 Ispra (VA), Italy (e-mail: Aikaterini.KANTA@ec.europa.eu, Iwen.COISEL@ec.europa.eu)
²Forensics and Security Research Group, School of Computer Science, University College Dublin, Ireland (e-mail: aikaterini.kanta@ucdconnect.ie, mark.scanlon@ucd.ie)

Corresponding author: Aikaterini Kanta (e-mail: aikaterini.kanta@ucdconnect.ie).

ABSTRACT It has been over 50 years since the concept of passwords was introduced and adopted in our society as a digital authentication method. Despite alternative authentication methods being developed since, it is reasonable to assume that this prevailing method of authentication will not be toppled anytime soon. Naturally, each password is tightly connected to its creator. This connection has given rise to advanced techniques aimed at exploiting user habits for password cracking. Such techniques are often generic approaches leveraging large datasets of human created passwords. A 2021 study showed that the online identity of almost one in three Americans was stolen in the last year alone, with a further 13% not being sure if their credentials were also breached [1]. Recent research has underlined the influence that context can have during password selection for a user. Such information could be of significant added value when digital investigators need to target a specific user or group of users during a criminal investigation. Besides manual techniques, there are no automated approaches that can extract and utilize contextual information during password cracking processes. In this paper, a methodology and framework for creating custom dictionary lists for dictionary attacks are introduced, with a specific focus on leveraging the contextual information encountered during an investigation. Furthermore, a detailed explanation the framework’s implementation is provided and the benefits of the approach are demonstrated with the use of test cases. This demonstrates the benefits of context in password cracking.

INDEX TERMS Password Cracking Candidate Generation, Context Based Password Cracking, Password Cracking, Wordlist Creation

I. INTRODUCTION

Despite known security concerns, password-based authentication remains the most widely used method of authentication [2]. In a spirit of strengthening security, password policies are nowadays more restrictive and enforce users to select stronger passwords. Salting the passwords¹ additionally increases the complexity of password cracking process, as each salt must be considered independently. Salting renders the commonly used rainbow table based approach obsolete. Typically, the password still remains the weakest link to gain entry to a system [3]. This weakness is accentuated when an attacker is focusing on gaining access to a multi-user system and not targeting any one specific user. A single weak password could grant attackers access to such system – rendering the effort and precautions taken by security concerned system administrators void. In these cases, attackers focus on generic approaches modeling the habits and trends of regular users [4]. These attacks use large dictionaries of human created passwords available online from previous data leaks/breaches. In addition, attacks have evolved to become more refined and sophisticated to compensate for the increase of the computational cost of the underlying algorithms and the strengthening of the password policies [5].

From another perspective, there are targeted attacks fo-
The contribution of this work includes:

- An assessment of the impact of context in password cracking – proving its viability as a valid approach to improve results over existing approaches.

The rest of the paper is organized as follows: Section II offers a quick overview of related work in the field, focusing specifically on user tendencies when it comes to password creation and password strength. Section III presents the proposed methodology for creating bespoke dictionary lists and outlines the specifics of how the methodology has been implemented – providing an in-depth explanation of the development choices made. Section V presents some proof-of-concept experiments using the resultant password candidate dictionaries compared to a commonly used baseline in the literature. Finally, the paper culminates with a discussion of the results and the conclusion and future work are outlined.

II. RELATED WORK

A. PASSWORD CRACKING TECHNIQUES

The most straightforward password cracking technique is an exhaustive search (also called a brute force attack) where all combinations of a given alphabet, including digits and special characters, up to predetermined length are tested. With no defined maximum password length or limits for attempts, exhaustive searches are guaranteed to work – the only variable is time. Nowadays, passwords up to 8 characters long can be checked in a reasonable amount of time with just a single GPU. For longer passwords or when the targeted hash function is not optimal, this approach is not efficient and is deemed computationally infeasible.

Therefore, many other methods have been developed to close that gap such as rainbow tables, dictionary lists (with or without password candidate mangling rules), and more recently machine learning approaches. Rainbow tables are a time-memory trade-off focused on precomputing an almost exhaustive predefined search space of passwords and store a minimal amount of information enabling a fast recovery of a given password if it match the predefined search space [7]. The use of salting in password-based authentication methods makes rainbow table based approaches entirely obsolete as one rainbow table would need to be constructed for each possible salt – of which there are near infinite possibilities.

When it comes to machine learning methods, these include Markov-based models to significantly reduce the size of the password space that needs to be searched [8], probabilistic context-free grammars [9], and neural networks to model the resistance of human chosen passwords to guessing attacks [10]. One such example of a neural network are Generative Adversarial Networks (GANs); where a neural network is developed to create password candidates that fall as close to the distribution of real passwords stemming from real-world password leaks [11].

One of the most common password cracking methods remains the dictionary based attack. The dictionary attack is often combined with a set of password mangling rules that dictates the variations to the dictionary word will be tried – these rules aim to mimic common user behavior when using a single password of a specific user. For example, this is the case for law enforcement during a lawful criminal investigation. Of course, generic approaches can be attempted as they rely on mimicking user tendencies or they leverage passwords originating from data leaks. However, this use case can also benefit from a more targeted context-based approach. This targeted approach should take into account the fact that users follow common patterns when they create their passwords. Their use of numbers and symbols is often meaningful or follows common patterns [6]. Users choose passwords that are memorable or meaningful to them. This is due to the fact that the typical user maintains tens of different passwords for different systems and devices. Since these password habits exist, the knowledge of more personal information about a specific user can lead to more educated guesses about their passwords. Such information could include important dates in their lives, names of family and friends, related locations, as well as their interests, likes and dislikes. A particularly insightful piece of information could turn out to be their password, or part thereof.

This way password candidates lists (dictionaries) that are bespoke to each individual can be created. Often, this information is easily and publicly available online, e.g., accessible on their social media profiles or professional websites. In the case of a law enforcement investigation, additional information could be obtained through warrants, interrogation, etc.

Taking the bespoke approach one step further, thematic dictionary lists around specific topics can be assembled. In terms of law enforcement, there is a huge benefit to this to expedite cases. During an investigation, it can be of paramount importance to gain access to encrypted devices – an often insurmountable task given limited resources. Manually creating tailored dictionaries for each suspect would be a very time consuming process depending. To overcome this, having some established lists on commonly encountered topics and interests could result in a most optimal start to the password guessing process.

In this paper, a methodology for creating bespoke and topic specific dictionary lists is introduced – starting with a single contextual seed word. The dictionary lists are fully customizable; the length of the list and contextual broadness of the password candidates generated are customizable by the user. The merging of lists from multiple seed words is also an option. An evaluation of the proposed methodology is presented and the first assessment proving the viability and impact of context based password cracking is outlined.
creating a password. For example, replacing letters with numbers or symbols, e.g., replacing ‘i’ with ‘1’ or ‘!’, letter capitalizations, or adding numbers/symbols at the beginning, middle or end, etc.

B. PASSWORD SELECTION TRENDS
The number of accounts every regular computer user owns is ever increasing. Single Sign-On (SSO) approaches and/or password managers can assist users in password management, while simultaneously strengthening the passwords used. However, as shown by [12], these techniques are not yet widely adopted. Consequently, a large proportion of users are reusing their passwords [13]–[15]—most probably to avoid having to remember an increasing number of increasingly complex passwords (enforced through increasingly strict password policies). The reuse of passwords with/without slight modifications among different services significantly reduces their security. For example, if one of these passwords is leaked, all the login credentials reusing the leaked password, or a variation thereof, are in danger of compromise and should be considered as unsafe [16].

When looking at leaked lists of passwords from various data breaches, it has been shown that common trends emerge regarding password selection. For example, when asked to create a password with lowercase and uppercase letters, users are likely to capitalize the first letter of their password [6]. When asked to include numbers and/or special characters in their passwords, they are very likely to use number sequences such as ‘123’, number repetitions such as ‘111’, meaningful numbers such as ‘314’, or use letter substitutions such as ‘@’ for ‘a’ and ‘1’ for ‘i’ [6]. One study showed that users tend to believe that adding digits in the password increases the complexity to guess it, while using keyboard patterns and common phrases was not perceived as a bad password practice [17].

A study focused on Chinese users [18] showed that more than 50% had passwords that consisted of only digits. The same study also showed that professionals generally chose lengthier passwords than students, and 12% included personal information in their passwords, e.g., birth dates or years. Another study that analyzed RockYou (a popular password cracking dictionary used in the literature), showed that 4.5% of passwords contained dates [19]. This type of information, while personal, is often easily accessible to adversaries [20].

Another analysis of passwords showed that password selection is far from random and that in fact it follows the distribution of natural language [21]. Users prefer to choose simple noun bigrams as found in natural language. When looking into differences in password preferences among people of different nationalities, some subtle differences were found by [22]. For example, the authors demonstrated that Arabic users were three times more likely to include their mobile phone number in their password, while people from India and Pakistan were more prone to use names.

It is therefore of significant interest to look more into password selection trends, what trend information can be derived and potentially leveraged in lawful password cracking.

C. PASSWORD STRENGTH
Enforcing the selection of strong passwords can help to protect digital systems from password cracking attacks. Password strength meters fulfill strength evaluation requirements forbidding users from inadvertently selecting weak passwords. However, a comparison study conducted on strength meters from some of the most popular websites and systems showed they are highly inconsistent [23]. The same password on different strength meters can be evaluated from adequate to great depending on what parameters each meter uses for its evaluation. These parameters include entropy, length, estimated number of guesses it would take to crack the password, etc.

The use of password strength meters can have the desired effect of users choosing more difficult passwords to fulfill the meter’s requirements, but subsequently need to resort to writing the password down because they cannot remember them [24]. Furthermore, entropy, which is one of the most common measures of password strength, has been shown as an ineffective meter against intelligence based attacks [25].

In order to mitigate against these issues, various alternatives to password meters have been proposed, e.g., limiting the number of login attempts, two-factor authentication, and the use of graphical passwords [26] or mnemonic based passwords. More recently other methods have been proposed where Markov Chain methods are leveraged to create a multimodal strength metric for passwords [27].

III. DICTIONARY CREATION METHODOLOGY
As mentioned in the previous section, dictionary attacks are an efficient password cracking method. There are many publicly available dictionary lists that are used for the purpose of password cracking. Many of which stem from leaked password lists from data breaches. One of the most famous lists is RockYou. This list originates from the RockYou company leak in 2009. The complete list of passwords from this leak are available as they were stored in plaintext by the company.

To this end, it seems logical that to improve the chances of cracking a password (or to crack as many passwords as possible) from a hashed, leaked list, is by creating more robust dictionary lists. The dictionary generation approach proposed as part of this work leverages the fact that: 1) users tend to choose passwords based on real words, 2) users choose passwords that are meaningful to them, and/or 3) users often use personal information including names, birth dates, places, and interests (e.g., sports, cars, popular cultural references, etc.). This selection of features is based on statistical analysis of over 3.9 billion real-world passwords [6]. The authors used the HaveIBeenPwned dataset and broke the passwords down into their constituent components and classified them according to context. This analysis demon-
strated that the aforementioned categories are some of the most popular chosen in the real-world.

A reasonable hypothesis is that if a user is tasked with defining a password for a website on a specific topic, the probability that this password might be thematically close to that topic is higher, e.g., more likely to choose a car related password for a car forum. Therefore, a dictionary generation strategy based on thematic categories can prove useful. Ideally, the building of a diverse portfolio of dictionaries for various contexts can be used standalone or in combination according to a specific target.

The approach outlined as part of this paper for creating dictionaries starts with Wikipedia\(^2\). The reasoning behind this is that each page in Wikipedia provides links to other Wikipedia entries that are thematically close – from a semantic, cultural and common association standpoint. This thematic linking of content can be pictured as a tree-like structure stemming from the root word, or seed phrase. This tree-like structure enables the selection of a starting point and the definition of the depth and breadth of exploration. An example of the tree-like structure of Wikipedia can be seen in Figure 1.

An example of the Wikipedia driven topic hierarchy is shown in Figure 1. Assuming the seed topic is “Manga”, each of the links referenced in manga’s Wikipedia entry lead to further related Wikipedia pages – from different types of manga, to famous Japanese actors, writers and illustrators, to manga related TV networks, etc. Proceeding down one level, i.e., visiting each of these Wikipedia entries, leads to further new related pages and so on. For the purpose of collecting this information from Wikipedia, DBPedia was used – as outlined further in the following section.

A. DBPEDIA

DBPedia\(^3\) is a crowd-sourced project aiming to offer a structured manner to access the information found in Wikipedia. The DBPedia information contains the abstract of each article found on each Wikipedia page as well the information contained in the article’s infobox. The infobox contains a summary of the most relevant information related to each article. As infoboxes in Wikipedia do not consistently follow a single structure, that information is collected with mappings. Mappings assign each entity in the infobox a DBpedia ontology type so that each attribute in the infobox is mapped to the DBpedia ontology [28]. This provides an easy way to leverage the structure and links between Wikipedia pages, providing an interconnecting web of content that is thematically related.

1) Keyword Extraction

In order to extract information from DBPedia, the Python library rdflib\(^4\) is used, which is a library for the Resource Description Framework (RDF)\(^5\). RDF is a data model that is used for merging graph data when the underlying schemas differ.

B. CREATING THE LAYERS

The starting point for creating a context based dictionary is a single seed word/topic/phrase and its corresponding DBPedia article. For example, if the objective is to create a dictionary about Manga, the starting point would be the DBPedia page for Manga. The first step is to collect all the links on the Manga entry that point to other related entries. As these are directly connecting to manga, as part of this paper they are referred to as the first layer. The next step is to visit these new entries and repeat the same process; collecting more and more links along the way. Consequently, each new link is classified in a different layer, according to how many “hops” it is from the starting point of the graph. A reasonable assumption that is made at this stage is that a link that resides in layer one, i.e., directly linked to from the Manga entry, is likely to be thematically more relevant to Manga than a link that is on layer two, three, or subsequent layers.

Furthermore, each new layer that is added raises the complexity significantly. As one example, layer one for the DBPedia article for Manga contains 314 entries, while layer two contains 19,727. Additionally, as many of these entries

\(^2\)wikipedia.org
\(^3\)https://www.dbpedia.org/
\(^4\)https://rdflib.readthedocs.io/en/stable/
\(^5\)https://www.w3.org/RDF/
Kanta et al.: Methodology for Contextual Based Password Cracking Dictionary Generation

Corresponding DBPedia article

Wikipedia Article(s)

Articles CD_1

Processing

Dictionary

Results

Articles CD_2

FIGURE 2. A methodology diagram for creating a dictionary from Wikipedia/DBPedia

are interconnected, i.e., the Manga entry points to the Dragon Ball Z entry and vice versa, particular care is taken not to include any repeating entries. The interconnected web of the articles can also be used as a relevancy metric for each page encountered – similar to one of the indicator’s web search engines use to determine a webpage’s relevancy based on how many pages link to it, such as Google’s PageRank algorithm [29].

A comprehensive diagram of the proposed process is shown in Figure 2. The length and scope of this list can be configured at the moment of generation. It can be limited to one layer, referred to as “contextual dictionary 1” (CD_1) as part of this work, two layers (CD_2), three layers (CD_3), etc. With each new added layer the quantity of data increases exponentially. Therefore the trade off between speed, length of the dictionary, and ultimate success rate is a consideration [30].

Furthermore, among the links contained in a Wikipedia (and corresponding DBPedia entry), some generic and non-topic-specific links can be found. These usually are used for Wikipedia’s internal hierarchy and labeling of contents in each entry, and these are excluded from the generated dictionaries.

C. DICTIONARY LIST SANITATION

At the culmination of the previous process, the first version of the dictionary list is created. At this point, subsequent steps are taken to sanitize this list and exclude entries (or partial entries) that are not contextually close to the starting seed word(s). Many linked pages from Wikipedia articles have the form of List of [Topic] or Categories: [Topic]. For example, using the Manga seed word, some of the linked Wikipedia pages include “List of Japanese manga magazines by circulation” and “Categories: Languages of Japan”. While these contents of these are thematically relevant and useful, these entries themselves do not offer added value and are therefore excluded from our dictionary list.

Regarding entries consisting of more than one word, each entry is included in the resultant password candidate dictionary list in two ways; as a concatenation of the words without spaces and as separate words. If these separate words consist of common stop words, they are removed. The removal of stop words happens for two main reasons; 1) this group of words does not provide any value to our process, and 2) as the size of the dictionary length decreases, a corresponding decrease in processing time follows [31]. As one example, if the entry The Girl From Ipanema is found, these three entries are added to the list: TheGirlFromIpanema, Girl, Ipanema.

IV. BENEFITS, LIMITATIONS AND TRADE-OFFS

As can be seen in Figure 2, the starting point for the proposed contextual dictionary approach is a single Wikipedia article stemming from the available contextual information about a target individual or community. In any digital investigation, this bespoke dictionary generation step could be one of the first after collecting evidence on the individual relating to their interests, hobbies, and other personal information. However, it might prove fruitful not to choose the bespoke dictionary approach from the get-go. The reason for this is that users still tend to choose passwords that are not very difficult and possibly are easy to crack with more unsophisticated methods, i.e., exhaustive search or “off-the-shelf” dictionary attacks. It is reasonable to first eliminate weak password candidates with an exhaustive search before utilizing the approach outlined as part of this paper, or to pursue both approaches simultaneously.

Furthermore, this exhaustive search can commence from the beginning of the investigation, as it does not require collecting any other information as it is entirely independent of any context. While the exhaustive search is conducted,
There is the possibility that some good password candidates are missed as a result of this decision, this trade-off is deemed acceptable to result in more relevant password candidates.

V. EXPERIMENTS

A. SETUP AND DATASETS USED

For conducting the experiments [Redacted For Blind Review]’s Sonic High Performance Computing Cluster has been used. A leaked community of manga fans was chosen as the target community for the first experiment, using the term “manga” as the seed word for the generation of the dictionaries. The evaluation of the generated password candidate dictionaries relies on a leaked dataset from the website MangaTraders – a forum for Manga and Anime fans. Our dataset is composed of 618,237 unique passwords provided by the online service hashes.org.

As a second experiment, the Comb4 dataset, which consists of four datasets, one of which being MangaTraders is used. The other three datasets are Axemusic (data leak from music forum), Jeeforum (data leak from a car forum) and Minecraft (data leak from a video game forum). The sizes of these dictionary lists that make up Comb4 are shown in Table 1. The use of these datasets for the purpose of this research has been approved by the Office of Research Ethics of [redacted for blind review].

As a baseline to compare the results of these experiments against, the RockYou dictionary has been used. There are two versions of RockYou publicly available. The first consists of 32 million passwords with repeated password entries (providing insight to the most frequently used passwords). For the experimentation outlined as part of this paper, the frequency of passwords is not of use and the version of RockYou used consists of 14 million unique passwords.

When it comes to the generated dictionaries, the seed word was “Manga” and two dictionaries of two and three layers, called CD_2 and CD_3 respectively, were produced. Their lengths are shown in Table 1.

For the evaluation of the results, two well known password cracking tools, OMEN [32] and Prince were used. OMEN is a password cracking tool using a Markov model and produces password candidates in order of decreasing probability. Prince is a password candidate generator that

| Dataset       | Size         |
|---------------|--------------|
| Comb4         | AxeMusic 252,752 |
|               | JeepForum 239,347 |
|               | Minecraft 143,248 |
|               | MangaTraders 618,237 |
| Evaluation    | CD_2 40,489 |
|               | CD_3 724,060  |
|               | RockYou 14,344,391 |

6[Redacted for Blind Review]

7https://github.com/hashcat/princeprocessor
uses one dictionary list to produce combinations of words as password candidates. Depending on the length specified, different combinations of words from the dictionary list are concatenated together to create new password candidates.

While time is dependent on the resources available for password cracking, as a reference, using our HPC cluster, each password cracking run with 10 billion guesses took approximately 9-10 hours for OMEN, while with Prince it took approximately 14-15 hours. It should be noted that the passwords were in plain-text, therefore no hashing was involved. The next section provides an overview of the experiments that were carried out and an analysis of the results.

B. EVALUATION SECTION

Both Comb4 and MangaTraders were evaluated using CD_2, CD_3, and RockYou as input dictionaries. 10 billion password candidates were generated from each of the three evaluation dictionaries for both the OMEN and Prince attacks. The results of the cracking progress over time for CD_2, CD_3 and RockYou with Comb4 and MangaTraders using OMEN can be found in Figure 3 and Figure 4 respectively. Likewise, the results of the cracking progress over time for CD_2, CD_3 and RockYou with Comb4 and MangaTraders using Prince can be found in Figure 5 and Figure 6 respectively.

A key difference between Figures 3 and 4 (which represents OMEN) and Figures 5 and 6 (which represents Prince), is that CD_2 is more performant compared to CD_3 using OMEN and CD_3 is the better performer using Prince. The explanation for this resides in the inner configurations of each of these tools. For CD_2, which is significantly smaller than CD_3, there are more variations of the same password candidate being attempted for the constant fixed number of guesses (10 billion for each password cracking run). For OMEN, which produces candidates in order of decreasing popularity, this means that the most likely candidates will be not only checked first, but checked with a higher number

FIGURE 3. Comb4 evaluated with CD_2, CD_3 and RockYou using OMEN

FIGURE 4. MangaTraders evaluated with CD_2, CD_3 and RockYou using OMEN

FIGURE 5. Comb4 evaluated with CD_2, CD_3 and RockYou using Prince

FIGURE 6. MangaTraders evaluated with CD_2, CD_3 and RockYou using Prince
of variations, i.e., more mangling rules applied, in the case of CD_2 compared to CD_3. For Prince, which is based on combining dictionary words, a larger dictionary list offers a wider range of combinations and therefore CD_3 performs better.

As expected, RockYou performs the best using both OMEN and Prince. The reason for this is that RockYou is a 14 million long dictionary of real-world passwords, while CD_2 and CD_3 are 345 and 19 times smaller respectively. Not only is the size difference significant, but also RockYou is a diverse dictionary that represents to a very large extent how people create their real-world passwords. RockYou is indicative of the password culture in our society, which is why it is one of the most popular dictionaries for password cracking attacks.

When comparing Figure 3 to Figure 4 and comparing Figure 5 to Figure 6, it is notable that the number of recovered passwords from MangaTraders is about half of what it is for Comb4. This is particularly interesting considering the fact that Comb4 contains 1,096,481 unique passwords – about twice as many as MangaTraders. This means that CD_2 and CD_3, have performed very well when the passwords they are trying to crack are of non-identical, but similar, context.

![Figure 7](image)

**Figure 7.** Passwords cracked by OMEN with CD_2, CD_3 and RockYou, classified by zxcvbn

If the number of cracked passwords is the only metric taken into account, then RockYou is the best performer. In this case, a larger and more diverse dictionary list performs the best and cracks the most passwords. However, in many real world scenarios other measures of performance take precedent over the sheer number of recovered passwords. For example, if time is of the essence or a single, strong password needs to be cracked, RockYou might not be a good choice.

![Figure 8](image)

**Figure 8.** Passwords cracked by OMEN with CD_2, CD_3 and RockYou, classified by zxcvbn

This is why it is important to also examine other metrics. For example, how strong the passwords cracked are should be explored. For this, the password strength meter zxcvbn, which is the strength meter developed by Dropbox, has been used. According to this meter, passwords are classified into five different classes according to how easily they can be cracked. Class 0 is considered the most easy to crack, while Class 4 contains the passwords that are deemed the most difficult to crack.

![Table 2](image)

|          | RockYou | CD_2 | CD_3 |
|----------|---------|------|------|
| Class 0  | 5,332   | 3,551| 3,182|
| Class 1  | 182,719 | 87,260| 61,519|
| Class 2  | 86,678  | 25,819| 20,312|
| Class 3  | 15,110  | 2,003| 2,220|
| Class 4  | 64      | 50   | 56   |

Tables 2 and 3 offer a breakdown of how many passwords

![Table 3](image)

|          | RockYou | CD_2 | CD_3 |
|----------|---------|------|------|
| Class 0  | 6,003   | 3,269| 3,782|
| Class 1  | 193,001 | 114,135| 155,925|
| Class 2  | 82,985  | 33,200| 45,910|
| Class 3  | 15,817  | 2,355| 4,558|
| Class 4  | 2,084   | 254  | 257  |
were cracked by each dictionary per class and per cracking tool. As can be seen in Table 2, when it comes to Omen, for Class 4 passwords, all three dictionaries did not perform well. Nevertheless, CD_2 and CD_3 cracked almost as many passwords as RockYou, which is an important feat, given the discrepancy in dictionary size between the three dictionaries. When it comes to the remainder of the classes, the results are more impressive, with the passwords found by CD_2 and CD_3 ranging between 13% and 47% of those found by RockYou in each Class. Looking at Prince, the results are comparable and most impressively, for Class 1, CD_3 found 80% of the passwords that RockYou found, as can be seen in Table 3. When it comes to Class 4, Prince was significantly better than OMEN, and CD_2 and CD_3 recovered approximately 12% of the passwords recovered by RockYou. However, the overlap of results achieved using CD_2 and CD_3 versus RockYou is not what demonstrates the true value of the proposed approach.

If a real world law enforcement password cracking scenario is considered once again, RockYou (or similar) may be used to crack passwords simultaneously while using the approach proposed as part of this paper. The value of this approach lies in the analysis of the passwords that using CD_2 and CD_3 were able to crack that using RockYou alone did not. Table 4 outlines the number of unique passwords per class that were cracked solely by CD_2 and CD_3 respectively and were not cracked by RockYou using OMEN and Table 5 shows the same for Prince. From these two tables, it can be observed that there is indeed value in running the context based dictionary attack in conjunction with RockYou.

As mentioned before, for Class 4 passwords using OMEN, CD_2 cracked 50 passwords and RockYou cracked 64. However, what is notable about that is that 49 of those passwords recovered by CD_2 were unique to CD_2 – bringing the total number of Class 4 passwords cracked to 113. This is an increase of 76.5% compared to just running RockYou. A similar increase can be observed in the case of CD_3 in the recovery of passwords unique to CD_3 versus RockYou. So, it can be observed, that even though the absolute numbers are low compared to more easily crackable classes of passwords, the amount of extra passwords cracked with the custom, targeted dictionaries is substantial.

Another class with a significant number of unique passwords cracked using CD_2 and CD_3 versus RockYou is Class 3, with a 5.7% and 5.4% increase of cracked passwords using CD_2 and CD_3 respectively. Overall, the fact that the extra percentage of unique passwords cracked using CD_2 and CD_3 were most significant for the two most difficult classes proves that the proposed approach is valid and that targeted, contextual dictionary lists can offer a significant advantage to the cracking process. This can be put into context even more, if we consider a digital investigation with a tech-savvy suspect, where - if their password is vulnerable to dictionary attacks - it’s still more likely to be Class 3 and above.

Overall, the largest increase of found passwords was achieved with CD_3 and Prince. CD_2 achieved to find 10.1% more passwords that were not already recovered by Rock You. For Class 1 passwords this increases to 15.5%. This is a very significant percentage, especially considering that - as mentioned above - the custom dictionaries performed especially well with the classes of stronger passwords. It could be argued that when time is of the essence, the targeted approach (because the size of the dictionary list is much smaller) could be the first tool to be used in the toolkit of the investigator.

### VI. DISCUSSION

The results of the experiments outlined as part of this demonstrate the impact of context on password cracking. Humans are creatures of habit – when they chose their passwords, they tend to repeat words and patterns and choose words that are familiar and meaningful to them. Their passwords have to make sense to them so that they can more easily remember them. Even in the case of users choosing random words (like a passphrase of four random dictionary words), the mechanism they use for password selection does provide insight. Of course, not everyone is like this. Many people nowadays use password managers and let the tool generate random passwords, therefore secure, on their behalf. Thus, neither a standard dictionary attack or a context based approach would prove fruitful against them.

Nevertheless, there is added value to the proposed more targeted dictionary approach. The experiments above demonstrate that, indeed, context matters. In the case where an investigator has information about the individual(s) that are targeted in a case, this approach should be considered. If there is only a single suspect and there is need to act fast, it may prove more useful to use the proposed targeted approach first. As mentioned, RockYou is significantly larger

### TABLE 4. Passwords found by only manga 2 layers or 3 layers (OMEN)

Unique Passwords found using CD_2 and CD_3 that were not found by RY using OMEN

| Class  | CD_2 Unique | CD_3 Unique |
|-------|-------------|-------------|
| Class 0 | 106         | 72          |
| Class 1 | 6,812       | 2,964       |
| Class 2 | 4,721       | 2,430       |
| Class 3 | 905         | 860         |
| Class 4 | 49          | 52          |

### TABLE 5. Passwords only found by the contextual-based approach of manga 2 layers or 3 layers (Prince)

Unique passwords found using CD_2 and CD_3 that were not found by RockYou

| Class  | CD_2 Unique | CD_3 Unique |
|-------|-------------|-------------|
| Class 0 | 12          | 12          |
| Class 1 | 3,265       | 14,545      |
| Class 2 | 4,092       | 12,927      |
| Class 3 | 1,283       | 2,619       |
| Class 4 | 46          | 179         |
than CD_2 and CD_3, which means it will take longer to execute. A smaller, more focused, bespoke dictionary using OMEN, which prioritizes more likely password candidates first, might be the best option to choose in the first instance.

Of course, if the aim is to crack more than one password, other factors need to be considered including how customizable the list should be. Is it better to start with one or more seed words? Is the number of passwords cracked enough for determining success or is there a need for other, more sophisticated metrics such as the amount of passwords cracked in a specific amount of time, or the strength of the cracked passwords? The quality of a dictionary can be measured in several different ways depending on the desired use case [30].

VII. CONCLUSION

The primary contribution of this paper is a novel framework for creating new, custom dictionary lists for any topic of interest that may be required – leveraging the power of structured information found on Wikipedia (and DBPedia). This contribution can provide the blueprint for creating customized dictionary lists easily for any topic, combine them, tailor them according to how deep and comprehensive they need to be, and personalize them to the exact specifications of each investigation.

A desirable byproduct of the proposed approach is that it facilitates investigators to crack passwords on topics they know nothing about. For example, investigators do not have to know anything about the desired topic to be able to assemble a custom dictionary list of the most important words about that topic. Furthermore, this dictionary generation utility facilitates investigators to keep up with current trends in password cracking and easily create new dictionary lists to accommodate them.

What our experiments have proven is that not only do people often choose passwords according to the website/system the password is for, often, they also inclined to choose passwords that arethematically close to the website/system they are for. Therefore, using a custom dictionary can offer a significant advantage to the cracking process and ultimately result in higher success rates over using a generic dictionary alone. Our experiments prove that using the proposed approach, in conjunction with existing approaches, results in up to 15.5% extra passwords being cracked over the existing approaches alone. This increased likelihood of cracking a particular user’s password could mean the difference between a digital investigation progressing or being stuck in its tracks.

A. FUTURE WORK

Fortunately, there are many avenues to explore to further enriching the process of creating context based dictionaries. Further sources of contextual information that can be considered include Wiki articles, forums, and social media. For example, a Twitter hashtag could be a good starting point for creating a dictionary list containing what people have to say on a specific topic right now. It could also provide insights into slang, colloquialisms, and common words or phrases associated with each keyword.

Furthermore, when it comes to the structure of the resultant dictionary list, identifying ways to prioritize candidates as more relevant to the seed word can prove beneficial. Of course, because of the tree-like structure of Wikipedia, words found in the second layer can be assumed to be closer to the seed word than those in the third layer, but there is room to improve this process. One avenue of future work is to estimate how related a candidate word is to the seed word. In doing so, the resultant list of words could be prioritized – enabling the most related candidate words to be attempted first. This would additionally allow the exclusion of irrelevant words. This could also be used to augment the dictionary with relevant words from additional sources, as mentioned above.

Finally, in terms of optimally applying this approach in real world scenarios, one focus for future work is to create a bank of precomputed seed word lists generated on common and popular topics so that they do not need to be regenerated whenever re-encountered.

REFERENCES

[1] Security.org. (2021) Password Manager and Vault 2021 Annual Report: Usage, Awareness, and Market Size. [Online]. Available: https://www.security.org/digital-safety/password-manager-annual-report/
[2] S.-h. Ju, H.-s. Seo, S.-h. Han, J.-c. Ryoo, and J. Kwak, “A study on user authentication methodology using numeric password and fingerprint biometric information,” BioMed research international, vol. 2013, 2013.
[3] V. Taneski, M. Heričko, and B. Brumen, “Password security—no change in 35 years?” in 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). IEEE, 2014, pp. 1360–1365.
[4] L. Boshnjak, J. Sreš, and B. Brumen, “Brute-force and dictionary attack on hashed real-world passwords,” in 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). IEEE, 2018, pp. 1161–1166.
[5] X. Ou, C. Hargreaves, J. Sheppard, F. Anda, A. Sayakkara, N.-A. Le-Khac, and M. Scanlon, “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, 2020.
[6] A. Kanta, S. Coray, I. Coisel, and M. Scanlon, “How Viable is Password Cracking in Digital Forensic Investigation? Analyzing the Guessability of over 3.9 Billion Real-World Accounts,” Forensic Science International: Digital Investigation, 07 2021.
[7] P. Oechslin, “Making a faster cryptanalytic time-memory trade-off,” in Annual International Cryptology Conference. Springer, 2003, pp. 617–630.
[8] A. Narayanan and V. Shmatikov, “Fast dictionary attacks on passwords using time-space tradeoff,” in CCS 2005 - Proceedings of the 12th ACM Conference on Computer and Communications Security, ser. Proceedings of the ACM Conference on Computer and Communications Security, 2005, pp. 364–372. cCS 2005 - 12th ACM Conference on Computer and Communications Security ; Conference date: 07-11-2005 Through 11-11-2005.
[9] M. Weir, S. Aggarwal, B. d. Medeiros, and B. Glodek, “Password cracking using probabilistic context-free grammars,” in 2009 30th IEEE Symposium on Security and Privacy, 2009, pp. 391–405.
[10] W. Melicher, B. Ur, S. M. Segreti, S. Komanduri, L. Bauer, N. Christin, and L. F. Cranor, “Fast, lean, and accurate: Modeling password guessability using neural networks,” in Proceedings of the 25th USENIX Conference on Security Symposium, ser. SEC’16. USA: USENIX Association, 2016, p. 175–191.
[11] B. Hitaj, P. Gasti, G. Atieniese, and F. Perez-Cruz, “PassGAN: A Deep Learning Approach for Password Guessing,” in Applied Cryptography and Network Security. Springer, 2019, pp. 217–237.
A. Kanta, I. Coisel, and M. Scanlon, “PCWQ: A Framework for Evaluating Password Cracking Dictionary Generation,” in 2012 IEEE Symposium on Security and Privacy, 2012, pp. 553–567.

D. Florencio and C. Herley, “A large-scale study of web password habits,” in Proceedings of the 16th International Conference on World Wide Web, ser. WWW ’07. New York, NY, USA: Association for Computing Machinery, 2007, pp. 657–666.

E. Stobert and R. Biddle, “The password life cycle: User behaviour in managing passwords,” in 10th Symposium On Usable Privacy and Security (SOUPS 2014). Menlo Park, CA: USENIX Association, Jul. 2014, pp. 243–255.

R. Wash, E. Rader, R. Berman, and Z. Wellmer, “Understanding Password Choices: How Frequently Entered Passwords Are Re-used across Websites,” in Twelfth Symposium on Usable Privacy and Security (SOUPS 2016). Denver, CO: USENIX Association, Jun. 2016, pp. 175–188.

B. Ur, J. Bees, S. M. Segreti, L. Bauer, N. Christin, and L. F. Cranor, “Do users’ perceptions of password security match reality?” in Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, ser. CHI ’16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 3748–3760.

Z. Liu, Y. Hong, and D. Pi, “A large-scale study of web password habits of Chinese network users,” J. Softw., vol. 9, pp. 293–297, 2014.

R. Veras, J. Thorpe, and C. Collins, “Visualizing semantics in passwords: The role of dates,” in Proceedings of the Ninth International Symposium on Visualization for Cyber Security, ser. VisSec ’12. New York, NY, USA: Association for Computing Machinery, 2012, pp. 88–95.

D. Keküllüoğlu, W. Magdy, and K. Vaniea, “From an authentication question to a public social event: Characterizing birthday sharing on twitter,” arXiv preprint arXiv:2201.10655, 2022.

J. Bonneau and E. Shutova, “Linguistic properties of multi-word passphrases,” in Proceedings of the 16th International Conference on Financial Cryptography and Data Security, ser. FC’12. Berlin, Heidelberg: Springer-Verlag, 2012, p. 1–12.

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

K. C. D. Carnavale and M. Mannan, “A large-scale evaluation of high-impact password strength meters,” ACM Trans. Inf. Syst. Secur., vol. 18, no. 1, may 2015.

B. Ur, P. G. Kelley, S. Komanduri, J. Lee, M. Maass, M. L. Mazurek, T. Passaro, R. Shay, T. Vidas, L. Bauer, N. Christin, and L. F. Cranor, “How does your password measure up? the effect of strength meters on password creation,” in 21st USENIX Security Symposium (USENIX Security 12). Bellevue, WA: USENIX Association, Aug. 2012, pp. 65–80.

M. L. Mazurek, S. Komanduri, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, P. G. Kelley, R. Shay, and B. Ur, “Measuring password guessability for an entire university,” in Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security, ser. CCS ’13. New York, NY, USA: Association for Computing Machinery, 2013, p. 173–186.

F. Alt, S. Schneegeass, A. S. Shirazi, M. Hassib, and A. Bulling, “Graphical passwords in the wild: Understanding how users choose pictures and passwords in image-based authentication schemes,” in Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services, ser. MobileHCI ’15. New York, NY, USA: Association for Computing Machinery, 2015, p. 316–322.

J. Galbally, I. Coisel, and I. Sanchez, “A new multimodal approach for password strength estimation. part ii: Experimental validation,” IEEE Transactions on Information Forensics and Security, vol. 12, no. 12, pp. 2845–2860, 2017.

J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auers, and C. Bizer, “Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia,” Semantic Web Journal, vol. 6, 01 2014.

L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking: Bringing order to the web.” Stanford InfoLab, Tech. Rep., 1999.

A. Kanta, I. Coisel, and M. Scanlon, “PCWQ: A Framework for Evaluating Password Cracking Wordlist Quality,” in The 12th EAI International Conference on Digital Forensics and Cyber Crime, ser. ICDF2C ’21. New York, NY, USA: Springer, 2021.