Context-Based Password Cracking for Digital Investigation

Aikaterini Kanta

A thesis submitted to University College Dublin
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

School of Computer Science

Head of School: Associate Professor Neil Hurley

Primary Supervisor: Associate Professor Mark Scanlon

Co-Supervisor: Dr. Iwen Coisel

January 2023

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. Despite alternative authentication methods being developed later, it is reasonable to assume that this prevailing authentication method will not fall out of popularity anytime soon. Passwords 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. The conundrum has driven both the efforts of system administrators to nudge users to choose stronger, safer passwords and elevated the sophistication of the password cracking methods chosen by their adversaries. The system administrator often overcomes the imperfection by skilfully enforcing strong password policies and dutiful password management on the side of the server. But at the end, the user behind the password is still responsible for the password's strength. A poor choice can have dramatic consequences for the user or even for the service behind, especially considering critical infrastructure.

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. On the other hand, law enforcement can benefit from a suspect's weak decisions to recover digital content stored in an encrypted format. Generic password cracking procedures can support law enforcement in this matter – however, these approaches quickly demonstrate their limitations. Recent research has

hinted at the influence that context can have on a user during his/her password selection. This information could be of significant added value when digital investigators need to target a specific user or group of users during a criminal investigation.

The connection between the password and its creator has given rise to advanced techniques aimed at exploiting user habits for password cracking. Such techniques are often generic approaches that leverage large datasets of human-created passwords. This thesis aims to investigate the hypothesis that bespoke password candidate lists, generated based on available contextual information, can positively impact the password cracking process. For this, a methodology and framework for creating and assessing custom dictionary wordlists for dictionary-based password cracking attacks are introduced, with a specific focus on leveraging contextual information. Furthermore, a detailed explanation of the framework's implementation is provided, and the benefits of the approach are demonstrated with the use of test cases. This work also introduces techniques for optimising the generation of the bespoke dictionaries, 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 approach achieved very promising improvements over existing, traditional approaches in isolation – more than 50% improvement in some instances. This result proves that more targeted approaches can be used in combination with the traditional strategies to increase the likelihood of success when contextual information is available and can be exploited.

To my family for everything.

Acknowledgements

I would like to express my deepest gratitude to my supervisor Dr. Mark Scanlon and co-supervisor Dr. Iwen Coisel for their knowledge and expertise they shared with me and their motivation and encouragement throughout these last four years. I am extremely grateful for Mark's guidance since his supervision of my Master's thesis and for Iwen's practical experience he shared with me and the inspiration he added to the project.

I would also like to thank the UCD School of Computer Science and the Joint Research Centre (JRC) of the European Commission for funding my research under the Collaborative Doctoral Programme. Their tangible support was crucial for the completion of this project. Many thanks should also go to Mr. Jean Pierre Nordvik, Mr. Laurent Beslay and everyone at the Cyber and Digital Citizen Security Unit in Ispra who welcomed me into a vibrant and thriving research community.

Furthermore, I would like to acknowledge my office mate Dr. Georgios Kambourakis for his valuable advice and interesting discussions on research and life. Special thanks to my colleagues Dr. Asanka Sayakkara, Dr. Xiaoyu Du and Dr. Felix Anda from UCD Forensics and Security Research Group for their support and feedback, especially during my stage transfer assessment. I would be remiss in not mentioning my RSP Chair Dr. Gavin McArdle and RSP Advisor Dr. David Lillis for their thoughtful comments and recommendations.

Lastly, I could not have undertaken this journey without the support of my family and friends, from school, university and beyond. Their faith in me, as well as their practical and emotional support, has kept my spirits and motivation high during this process. I am especially grateful to my parents who, by now, know way more about password cracking than they ever wished, to my sister, Konstantina, for being a sounding board and a source of inspiration, and my grandmother for always being there for me.

Contents

List of Publications

Journal Papers

- Kanta, A., Coisel, I., and Scanlon, M., A Comprehensive Evaluation on the Benefits of Context Based Wordlists for Password Cracking, Journal of Information Security and Applications, Elsevier, 2023. **(Under Review)** [1]

- Kanta, A., Coisel, I. and Scanlon, M., Harder, Better, Faster, Stronger: Optimising the Performance of Context-Based Password Cracking Dictionaries, Forensic Science International: Digital Investigation, ISSN 2666-2825, Elsevier, March 2023. **(To Appear)** [2]

- Kanta, A., Coisel, I., and Scanlon, M., A Novel Dictionary Generation Methodology for Contextual-Based Password Cracking, IEEE Access, Volume 10, pp. 59178-59188, ISSN 2169-3536, IEEE, June 2022. DOI: `https://doi.org/10.1109/ACCESS.2022.3179701` [3]

- Kanta, A., Coray, S., Coisel, I. and Scanlon, M., 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, Volume 39, Article 301186, ISSN 2666-2825, Elsevier, July 2021. DOI: `https://doi.org/10.1016/j.fsidi.2021.301186` [4]

- Kanta, A., Coisel, I., and Scanlon, M., A Survey Exploring Open Source Intelligence for Smarter Password Cracking, Forensic Science International: Digital Investigation, ISSN 2666-2825, Elsevier, December 2020. DOI: `https://doi.org/10.1016/j.fsidi.2020.301075` [5]

Conference Papers

- Kanta, A., Coisel, I., and Scanlon, M., PCWQ: A Framework for Evaluating Password Cracking Wordlist Quality, Digital Forensics and Cyber Crime: 12th EAI International Conference on Digital Forensics and Cybercrime (ICDF2C), Singapore, Springer, December 2021. DOI: https://doi.org/10.1007/978-3-031-06365-7_10 [6]

- Kanta, A., Coisel, I., Scanlon, M., Smarter Password Guessing Techniques Leveraging Contextual Information and OSINT, The 6th IEEE International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Dublin, Ireland, IEEE, June 2020. DOI: https://doi.org/10.1109/CyberSecurity49315.2020.9138870 [7]

List of Figures

List of Tables

Declaration of Authorship

I hereby certify that the submitted work is my own work, was completed while registered as a candidate for the degree of doctor of philosophy, and I have not obtained a degree elsewhere on the basis of the research presented in this submitted work.

Aikaterini Kanta	01/01/2023
(Student Number: 17201590)	Date

Chapter 1

Introduction

Despite known security concerns, passwords still remain the most widely used and one of the easiest and most adopted methods of authentication. Password-based authentication is older than modern digital society might realise. It is such an archaic system yet remains a crucial component of the security of most digital systems (albeit not necessarily the only one). As password policies become more restrictive by enforcing the selection of stronger passwords, attacks also become more refined and sophisticated. Traditional password cracking methods have, in many cases, become less efficient due to the increase in the computational cost of the underlying algorithms and the strengthening of the passwords [13].

1.1 Motivation

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. For a lawful digital investigation, passwords can be the single point that hinders the progress, ultimately resulting in justice not being served. Therefore, password cracking methods need to be refined and to adapt to the increasing diligence of suspects who choose harder and stronger passwords. 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 investigator trying to gain access to 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. The proposal that this thesis puts forward looks at ways to leverage contextual information about the target(s) to enhance the password cracking process and aid Law Enforcement in their fight against digital crime.

1.1.1 Why Are We Still Using Passwords?

With all the known weaknesses of password-based authentication systems, one might wonder: why are we still using them? One explanation might be the public acceptance of this mechanism. Everybody has already used password-based authentication. In fact, it is estimated that, on average, a human has between 70 and 150 online password-protected accounts even though often the users themselves fail at evaluating their own digital environment/footprint. The only certainty is that this number is growing over time as the world is experiencing its ongoing digital transformation. This societal phenomenon is mostly technologically driven and is safe to assume will continue into the future with autonomous driving, remote surgery, smart homes, etc.

But with that being said, what are the alternatives to a password? Single-Sign-On (SSO) strategies, active-directory, and password managers offer substitutes or enhancements over simple password authentication. Each of these solutions increase account security as they often require additional elements for a malicious actor to access a given system, e.g., having access to the key wallet protected by the password manager. Nevertheless, they still rely on a password at one stage or another and are unfortunately not yet widely adopted [14]. This is also largely true with two-factor authentication, where one of the factors often remains “something you know” – namely a password.

Lastly, some people might argue that they are no longer using passwords to unlock their phone, make payments, etc., but instead use their fingerprint or facial scan for identification. However, any service relying on a fingerprint reader or facial scanner on their phone can be bypassed by knowing the master code of the phone

– this allows anyone to define a new fingerprint and bypass this security feature.

Therefore, passwords are not dead and will most likely continue to be used in one way or another for the foreseeable future. It is of utmost importance to strengthen password-based authentication systems. Of course, this includes safe storage of the password on the server/device side. Furthermore, passwords selected by users should be strong in the very first place to ensure the best level of security.

1.1.2 Law Enforcement Investigation

The other side of the coin is that security reinforcements and the ready availability of strong encryption tools can also benefit criminal enterprise and hamper lawful investigation [15]. As much as a password is the barrier for attackers to breach a critical system, it is equally a hindrance for law enforcement conducting their investigations. In the context of a digital forensic investigation, the use of password-protected accounts and devices can present a hurdle for their lawful analysis under warrant [16].

The prevalence of password protected encrypted storage coupled with increasingly stringent password policies can result in cases being significantly held up in the best case or hitting a dead end in the worst case. Law Enforcement Agencies (LEAs) throughout the world are struggling to keep up with the demand for digital forensic investigation – with multi-year, case hindering backlogs becoming commonplace [17]. When time is of the essence both from a time-sensitive case (e.g. child abuse investigation, human trafficking, etc.) and investigative efficacy perspective, the decisions made on what resources to allocate to each digital investigation can be crucial.

Law enforcement agencies are nowadays encountering digital evidence in almost all investigations. An outstanding proportion of offenders, like any other member of society, they have at least a mobile phone and a personal computer. These devices follow the security trends of the manufacturers and the content is most likely protected with a basic standard of protection at a minimum. Nowadays, mainstream desktop computers and mobile operating systems offer built-in password protected encryption of their storage volumes [18]. This feature is usually enabled by default in many cases without user configuration [13].

Offenders often take additional security precautions if they are aware of the risks

of investigation – as highlighted in the latest Internet Organised Crime Threat Assessment (IOCTA) report from Europol [19]. For example, they might employ additional levels of encryption over what might be enabled by default, such as full disk encryption or encrypted communication – again often protected by a password. Attempting to brute force a password for each account, encrypted file or storage volume can use a large amount of resources with no guarantee of success in any reasonable time frame [13].

1.1.3 The Role of the Password in Digital Investigation

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 [13]. 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 [20]. 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 [21] and users tend to gravitate towards passwords that contain information that is personally connected to them [22]. Therefore, it stands to reason that the cracking process can benefit from a tailored approach using the available contextual information of the suspect. A user-centric approach could leverage this knowledge by focusing 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. This will result in an attack that is tailored to the individual target, which could return results when traditional methods fail.

In the case where a single password is considered, if the investigator is aware of information regarding the target, that information can be leveraged aiming at cracking the password in fewer attempts. An example of this type of situation would be a

Law Enforcement Officer wanting to access a password protected digital device of a suspect during the course of an investigation. In this case, time is of the essence in order to swiftly resolve the investigation or prevent further criminal acts.

Another particular case is when the targeted dataset can be associated with a particular context. An example would be a penetration testing campaign evaluating the strength of the passwords used by the user of a system. If such a system is linked to a particular community, e.g., users of a video game service, the operator can use contextual information such as typical language used in this community, references from the topic, or any other type of contextual information to refine the password cracking process.

All the above information can be useful, to try to make more educated guesses about the password of a target or a community. The ultimate goal is to recover the password faster than current state-of-the-art approaches would, by giving the investigator a head start regarding the wordlist they use during the password cracking process. By creating a custom wordlist, that is tailored to the target and by checking first password candidates that are more likely to be chosen as the password by the target, a more efficient password recovery process could be achieved.

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.

1.2 Research Problem

The research presented in this thesis seeks to leverage contextual information for the purpose of password cracking. Towards addressing this problem, the following three questions were defined and pursued in this thesis.

RQ1: What impact does a context-based password cracking approach have on the likelihood of success during a digital investigation?

Password cracking techniques that make use of dictionaries, whether this is for

directly using the dictionary with a password cracking tool or by using the dictionary as training input, have been mostly utilising data coming from real-world data leaks. These dictionaries, since they contain real human-chosen passwords, represent well what human passwords look like. This makes the dictionary attack one of the most successful password cracking techniques available. However, this technique does not take into account the semantic information within the password. As the related work that is discussed in Chapter 3 will show, it is important to look at the context found within the password and harvest it for the purposes of more informed password cracking.

In order to enable digital investigators to make informed decisions on the likelihood of success for this approach, actionable statistics are needed, based on a significantly large dataset of real-world passwords. As part of this thesis, a list of 555,278,657 unique passwords correlating to 3,951,907,330 real-world accounts will be assessed. Password reuse accounts for the disparity in these numbers, i.e., repeatedly used passwords between both different accounts and different users. Furthermore, smaller datasets that are centred around niche topics will also be evaluated with dictionary leaks stemming from similar topics.

RQ2: How can a context-based password cracking dictionary be generated, bespoke to the interests of an individual suspect or a group of suspects?

The approach followed by digital investigators is different from those of malicious password crackers. The latter is predominantly interested in getting one hit to enter into a system under any user's account, or to gain access to the maximum of entries in a given dataset. The former are more interested in one specific user's account – the one of the targeted suspect.

To answer this research question, a modular framework to assess the quality of a wordlist is designed to be used in password cracking processes. Several criteria have been proposed that can be considered for the evaluation of dictionary lists, and it will be explained why there cannot be a single and totally ordered metric to evaluate the wordlist.

Furthermore, a methodology for creating bespoke and topic-specific dictionary lists will be introduced, starting with a single contextual seed word centring around

one topic of interest. The dictionary lists will be fully customisable; the length of the list and the contextual broadness of the generated password candidates will be selected by the creator of the list. Merging lists from multiple seed words will also be an option. Furthermore, extensive evaluation of the proposed methodology will be presented to demonstrate the viability and impact of context-based password cracking.

RQ3: How can password candidates be contextually prioritised in a dictionary, and what impact does this prioritisation have?

One of the purposes of this work is to aid law enforcement investigators during criminal investigations by providing dictionary lists for password cracking that are tailored to the suspect. Frequently, ensuring swift access to password-protected systems and devices can be the one detail that will make or break an investigation. Therefore, trying to optimise the password cracking process with regard to its success rate but also the fastness with which these results are produced is fundamental. The methodology described in the answer to the previous research question can become an important tool in an investigator's toolkit, by providing readily available, highly-customised contextual dictionaries on any topic – however niche. As part of this thesis, a process for optimising and ranking the candidates of a custom-made dictionary list is also presented. The password candidates are ranked so that more suitable password candidates are checked first, in a bid to save time during an investigation. This approach is also evaluated 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 generic techniques. The optimised, ranked dictionary lists as presented in this thesis offer a significant increase compared to popular, widely used dictionaries.

1.3 Contribution of this Work

The work outlined as part of this thesis proves that context plays a role during users' generation of passwords and can therefore be exploited by LEAs during their lawful

criminal investigation. There is no dataset available focusing on a single user, and ethical reasons prevent us from testing this approach on a single individual. As a result, the analysis outlined below is focused on a community level in order to extrapolate how likely a contextual approach is to succeed. Nevertheless, the bespoke, context-based approach outlined as part of this work is proven to find passwords exclusively recoverable using this technique, i.e., those that were not found by currently used, generic approaches. The contribution of this work includes:

- The largest and most comprehensive analysis of real-world passwords conducted to date. This work looks at the underlying statistics of the constituent passwords and their component fragments, showcasing password creation tendencies of real users. Furthermore, the most common semantic classes are identified, underlining the importance of context/interest when users select their passwords.
- A framework for the standardised evaluation of password cracking dictionary lists.
- The design of a novel methodology for creating bespoke dictionary lists based off a user's interests or specific topics. Furthermore, this methodology facilitates customisation from the dictionary creation parameters definition to password cracking tools selected.
- An evaluation of the above methodology based on a number of realistic scenarios for which the contextual approach will be a beneficial tool for an investigator.
- The execution of an extensive experimentation and evaluation of the methodology across a number of targeted datasets of varying topics.
- A technique for optimising and ranking contextual dictionaries based on their relevancy to a specific topic.
- A detailed discussion highlighting the uses, benefits and limitations of leveraging context in password cracking.

- Identifying several different avenues for the application of this work outside the Law Enforcement context.

1.4 Thesis Organisation

The rest of this thesis is organised as follows. In Chapter 2, the technical background of concepts required to follow this thesis are presented in detail. A comprehensive literature survey on the topic is presented in Chapter 3, which covers a broad range of related work in the field of password cracking. Chapter 4 illustrates the approach taken for to assess the impact of context in password cracking, as well as the methodology to leverage that information. The tools and methods that have been used throughout this thesis to implement the framework and processing needed for the creation of the bespoke contextual dictionaries are explored in 5. Chapter 6 is dedicated to the presentation of the results of the experiments and Chapter 7 to a thorough discussion of them, including the position this approach can occupy in the grand landscape of password cracking. Finally, Chapter 8 concludes the thesis by looking at the implications of this work in related fields and outlines several avenues for future work.

Chapter 2

Technical Background

2.1 Introduction

Society is in constant evolution. The advent of the internet is often considered as a key turn of civilisation same as controlling fire. While this is open to debate, what is not is that such technological advance opens the door to major changes across the digital world, leading to both great opportunities and new challenges. This evolution is often referred to as the digital transformation of modern society. There are barely any dimensions of people's lives that are not affected by this change.

Law Enforcement Agencies (LEAs) are thereby impacted by the rise of a modern digital world. Their community is already benefiting from the development of new solutions to store, exchange and ease the access to information and tools. These new solutions can act as facilitators and enablers, transforming the more traditional procedures they apply when conducting an investigation to prevent or react after a crime.

In parallel to those new opportunities, this digital transformation also creates new challenges for law enforcement by providing new opportunities and means to criminals. Crimes are now sometimes committed fully online, e.g., botnet exploitation and ransomware. The digital world can be the channel to sell and exchange illegal material, e.g., trading platform for drugs and weapons, or exchange of child sexual abuse materials. Whatever the crime, the common challenge for law enforcement is that data at rest or in transit is almost systematically protected by encryption

means. Recovering the data in clear is often the key to properly pursue an ongoing investigation or prosecute the criminals.

How do we deal with encryption? Direct attacks aimed at breaking the encryption method itself are generally not possible, as robust and standard methods are nowadays available to everyone. Nevertheless, existing solutions are often password-based, especially in the data at rest scenarios (the encryption method used in data in transit can be totally transparent to the user). Passwords are the weakest point of the whole security chain, as human-chosen password are known to be somewhat weak in average [23]. Password cracking techniques are traditionally designed to produce generic candidates, mimicking the most common passwords or patterns. This approach is typically sufficient to assess the average level of security of a system during penetration testing. A single hit, meaning the password of any user, might be sufficient to harm a system in such scenario.

Law enforcement are in a different scenario as they focus on a single user or groups of users. While generic password cracking techniques can remain successful, they can benefit from a more targeted approach when dealing with encrypted material. Humans have the tendency to generate easy to remember passwords [24]. One common method involves using personal information in the password, such as Jeremy Hammond, a wanted hacker, who used the name of his cat in his password [25]. There however stand two challenges that are unsolved:

- How can state-of-the-art password cracking tools benefit from a targeted approach?
- How can the targeted approach aid Law Enforcement in their fight against digital crime?

There are a number of survey papers on the topic of password cracking, with analysis on password cracking methods and evaluation of strength estimators [26] and suggestions on countermeasures [27]. Where this literature review innovates, is on the incorporation of password tendencies of users and the inclusion of the Open Source Intelligence (OSINT) element, where its use by LEAs is presented as well as its potential usefulness as an additional element in a contextualisation attempt on password cracking. To this end, a number of further research directions have been identified on how to leverage freely available information for a targeted approach.

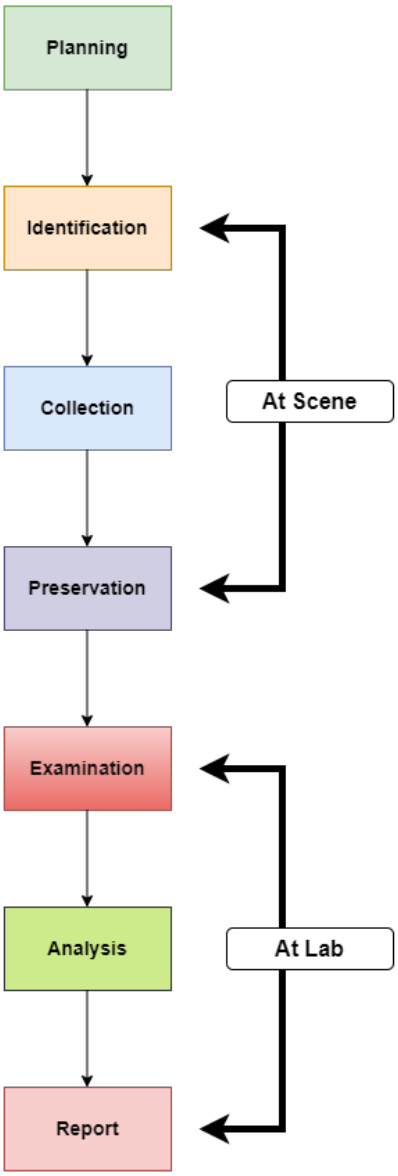


Figure 2.1: Traditional digital forensics process model [8]

2.2 Digital Forensic Process

Since the dawn of the digital era, physical evidence collected on a crime scene are not the only tools at the disposal of a law enforcement investigator. A variety of digital evidence such as those collected on the physical scene: hard drives, computers,

smart devices provide information such as the location off of GPS and tower cell data, interests and hobbies of a suspect, information on close contacts, etc and can give the investigator useful assistance. Nowadays, crimes, such as financial scams, human trafficking and child pornography distribution can be organised and perpetrated exclusively online. For this reason, many protocols and procedures on how to deal with digital evidence have been proposed by researchers, that cover all steps of the investigative process in both cyber and traditional investigations. A general process model can be seen in Figure 2.1. According to Du et al. [28], the typical stages of a digital investigation are:

1. Identification - The first stage is about identifying the details of an incident or crime and the relevant evidence that might need to be examined. For example, in a house search, all digital devices that belong to the suspect have to be identified for collection in the next steps.
2. Preservation - This stage is about preserving the crime scene and the evidence by taking photos, keeping a chain of custody on the evidence, etc. This is an important step in the investigation from the beginning to the end when/if the evidence must be presented in a court of law.
3. Collection - In this stage of the investigation, the digital evidence that is deemed relevant is collected from the crime scene. This is usually done by imaging the electronic devices by using special forensic equipment and software in order to not alter their content in any way.
4. Analysis - This is the stage where the investigator has to interpret, analyse and organise the evidence they have acquired and “build their case”.
5. Reporting/Presentation - The last stage refers to the presentation of the findings of an investigation to a court of law or other authority. An important detail to be taken into account is that the results presented at this stage would have to be reproducible by other investigators in order to be accepted.

In addition to the typical stages of a digital investigation mentioned above, the Association of Chief Police Officers (ACPO) has provided a Good Practice Guides for Digital Evidence which includes the known widely ACPO Principles that every practitioner must follow when handling digital evidence [29]. The last update to this guide is from 2012.

In situations where time is of the essence, some deviations from the traditional

digital investigation models are needed. This means deviating from an in-depth analysis of digital evidence at the lab in favour of extracting quick information that will aid in a time critical investigation such as an abduction or a kidnapping [8]. In this case, some adjustments need to be made to the traditional model for triage. The Computer Forensics Field Triage Process Model (CFFTPM) incorporates these changes as seen in Figure 2.2.

2.2.1 Digital Forensic Challenges

Despite the many established processes and procedures on dealing with digital evidence and performing digital forensics, there are many challenges in the field that hinder the effort of digital forensics specialists to acquire and process digital evidence in a timely manner. There are quite a few efforts over the years to identify, categorise and analyse the current challenges facing the digital forensics community, as well as look at the trends for the future.

Al Fahdi et al. [30] conducted a survey of digital forensic practitioners, who overwhelmingly predicted an increase in complexity for investigations in the future. Another survey of practitioners, showcased that the challenges spread across the spectrum; from technical (higher support for cloud forensics) to legal (privacy laws) and educational challenges [31].

A taxonomy of current challenges in the field is presented by Karie and Venter [32], while Lillis et al. [17] aim to define the future areas of research in digital forensics. In general, the different categories of challenges are split into three main categories, technical challenges, challenges regarding the law and challenges regarding resources.

2.2.2 Digital Forensic Backlog

Due to the rapid growth of digital crimes in conjunction with the number of seized devices in these crimes and ever-increasing data storage of these devices, each investigation might acquire a significant number of devices and data that need to be analysed [33], with additional complexity due to device encryption. In fact, according to Safaei et al. [34], each person will use more than 9 devices in their day-to-day lives

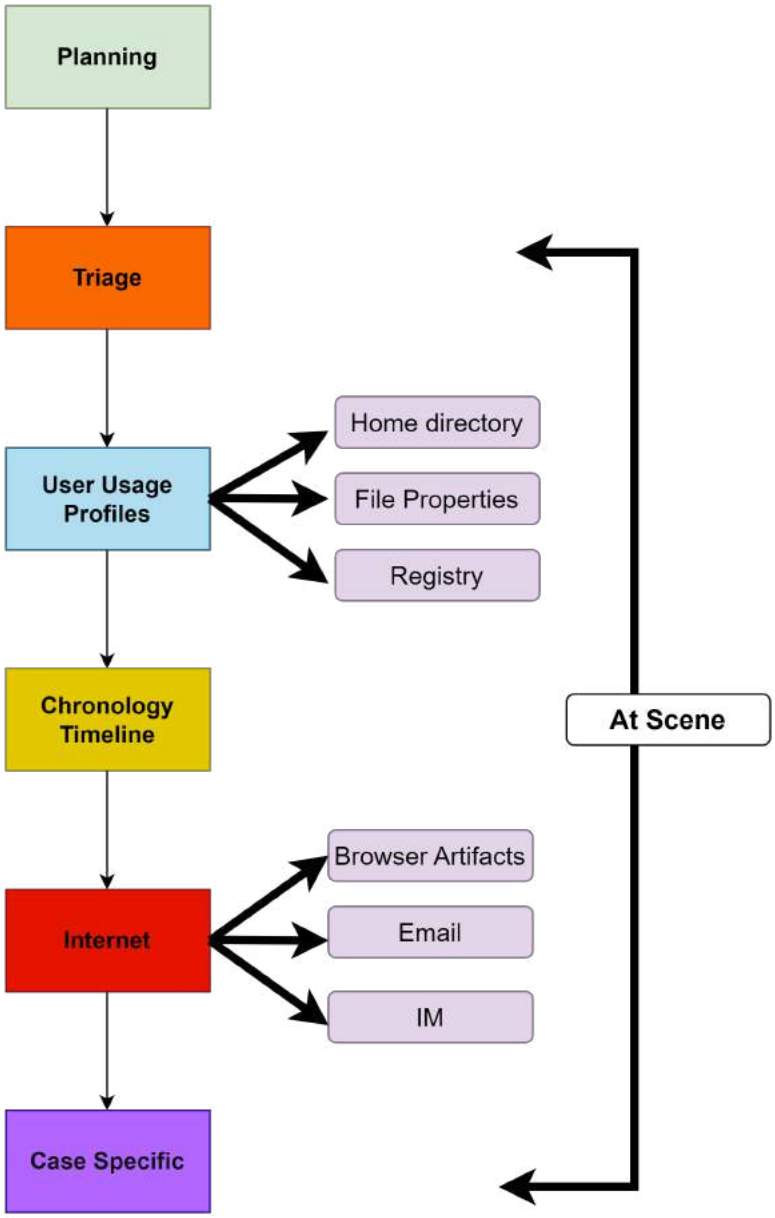


Figure 2.2: CFFTPM digital forensics process model [8]

by 2025. This creates a backlog of cases up to four years that leads to LEAs not being able to process the evidence in a timely manner and might even lead to cases being dropped [35]. One more reason that adds to the backlog is the increase of IoT devices that are used in everyday lives as well as the ever-increasing use of cloud

services [17] as detailed below.

Technical challenges

IoT Forensics A consequence of the digitisation of society is the ever-growing constellation of IoT and smart devices surrounding each individual. Such growth raises privacy and security issues as threats and vulnerabilities, e.g., Denial of Service (DoS) attacks, spoofing, eavesdropping, etc., have already been identified in those devices [36]. From another point of view, those devices and the data they collect and process constitute a gold mine of information for law enforcement. In a 2019 survey with digital forensics practitioners, it was found that many of them already encounter IoT devices in their work but feel under-trained to examine them [37]. To this end, specific procedures for forensic investigations on IoT devices must be defined to take advantage of such data without contributing negatively to the already existing backlog.

Cloud Forensics As more and more companies move to the cloud, due to its lower cost and ease of troubleshooting, the advantages of performing digital forensics on the cloud are also more apparent. Cloud forensics is defined by Ruan et al. [38] as "the application of digital forensics in cloud computing as a subset of network forensics". Therefore, it is important for digital forensic investigators to be able to apply the same techniques and procedures they use in digital devices to their cloud counterparts. To this end, Ruan et al. [39] have conducted a survey with digital forensics expert participants in order to analyse the current issues and challenges faced by this industry when it comes to cloud forensics procedures, tools and investigations as well as to identify future opportunities for research and development. Some of the challenges the participants claimed posed a hindrance to the investigation include evidence segregation and lack of access to physical data. Furthermore, Manral et al. [40] have summarised and grouped the digital forensic challenges in the cloud according to the step of the investigation process the investigators encounter them on. Some of these challenges that are specific to cloud forensics include dealing with jurisdiction issues and being familiar with different cloud architectures.

Legal Challenges

When it comes to a digital investigation, a challenge for law enforcement is making sure they can guarantee the admissibility of digital evidence into a court of law. This means that the proper procedures of the digital investigation process must be carried out successfully in every step of the investigation, such as ensuring the proper collection of evidence and keeping the chain of custody. It is a challenge for law enforcement to properly evaluate and report on digital evidence in a way that establishes their validity and admissibility. This challenge is directly tied to the correct following of the digital investigation process as described in Section 2.2. Anti-forensics, is another hindrance to properly evaluating and reporting on digital evidence. Anti-forensics is defined by Liu and Brown [41] as the "application of the scientific method to digital media in order to invalidate factual information for judicial review" and has the goal of making the collection of digital evidence by investigators more complex and/or invalidating their findings. It is employed by criminals as a way to mitigate the results of LEA finding evidence that can incriminate them.

Resource Challenges

When it comes to personnel challenges, police officers that have to perform digital forensics are most of the time not adequately trained on how to use the forensics analysis equipment and handle the evidence according to the established procedures [42]. According to the United Kingdom's (UK's) House of Commons Justice Committee [43], the reason for this is the unavailability of funding. In addition to this, in many cases, there is not enough available personnel to actually work on forensics analysis cases.

2.2.3 Prevalence of Passwords is Hindering Investigation

Ever since the emergence of the World Wide Web (WWW), criminal activities are being conducted more and more on the internet, with digital devices being the facilitator or the place where information pertaining to these crimes is being stored. Nowadays, most popular Operating Systems (OSs) of electronic devices make sure to encrypt the internal storage of the device in order to ensure the safety of the data

on the device [18].

Indeed, in many cases encrypted devices hold sensitive information that can make or break an investigation, some of which might be time critical, e.g., a kidnapping or a terrorist attack. There are of course tools in an investigator's arsenal for dealing with encrypted devices, but more often than not, these tools and processes can be time-consuming and a successful result is not guaranteed.

As stated by Plunkett et al. [44], "the lack of passwords, particularly during the execution of search warrants, has hindered investigations". It can be crucial to get access to such content during an investigation – necessitating the retrieval of the suspect's password(s). Of course, criminals are not always inclined to share their passwords with the investigators.

There have already been cases of LEA reporting hindering of investigations where suspects have not provided the password to access an encrypted device [45] or suspects being sentences to prison for not revealing a password that pertained to terrorist activities [46]. One of the most polarising cases of 2016 was the case of the San Bernandino terrorist attacker, where the American government requested Apple's help to unlock an iPhone belonging to the shooter suspect. The case drew unprecedented attention, and various debates arose about whether Apple (and other companies) should comply with requests like this [47]. In the eyes of the Federal Bureau of Investigation (FBI), Apple's refusal to comply with this request constituted as obstruction of justice [48].

It is not always possible to compel the suspect to divulge his/her passwords through a court order. For example, compelling password surrender could be considered as against the Fifth Amendment in the USA protecting suspects from self-incrimination [49]. In some other countries, it is considered a crime to not reveal a password under court order, e.g., in the United Kingdom within the Section 49 of the Regulation of Investigatory Powers Act 2000. Nevertheless, the suspect may well decide to not reveal the password if the sentence incurred is lower than what might be expected should police gain access to the device(s). In each of these cases, LEAs have no other choice than conducting password cracking processes to recover the suspect password and examine the targeted content.

2.3 Open Source Intelligence

The work that has been done into looking at password habits of users has shown that personal information such as interests and personal details are often included in passwords. When looking to access a specific suspect's device, law enforcement might have better results when taking a more targeted password cracking approach. To this end, OSINT could be a good source of information.

The US Intelligence Community Directive 301 [50] defines Open Source Information as "publicly available information that anyone can lawfully obtain by request, purchase, or observation," and Open Source Intelligence as "produced from publicly available information that is collected, exploited, and disseminated in a timely manner to an appropriate audience for the purpose of addressing a specific intelligence requirement".

OSINT techniques appeared before World War II [51, 52, 53] and were at the time known as overt intelligence. The main source was enemy press as well as press in countries that remained neutral [51]. While it can be argued that this sort of information gathering rarely yielded great revelations, it provided a coherent image of the public opinion as well as the living conditions [51].

Nowadays, OSINT has evolved remarkably to include a plethora of online sources available to anyone such as the Internet (social networks, online encyclopedia, `whois` domain records, etc.), traditional media (newspapers, television, radio), academic publications (journals and conference proceedings), grey literature (technical reports, diplomatic message), geospatial information (Google Maps and Streetview), publicly available data (government reports, budgets), etc. [54, 55].

One of the most useful traits of OSINT is the volume and the availability of information [56]. According to Roser et al. [57], the number of internet users increased from 413 million in 2000 to more than 3.4 billion in 2016. As a consequence, millions of data are produced every second and the Internet is more than doubling its size in amount of data every two years [58]. This is an information gold mine, but it is also a tremendous task to sort through such a volume of data and transform the collected pieces into something valuable. According to Burke [59], intelligence can be viewed as the end product that stems from the analysis and filtering of data to generate something of value for a specific purpose.

Furthermore, a downside of its availability is that it is not easy to evaluate the quality of the information, especially when it stems from the internet [60]. This issue is not something new, or in fact singular to OSINT, as intelligence agencies have long resorted to keyword sampling and other filtering techniques to sort through exorbitant amounts of information [61].

On the other hand, Miller [62] poses the question whether or not information that is readily available on the internet can be called intelligence. The argument against classifying as intelligence is that it is not acquired by clandestine means, nor does it need special handling like covertly acquired information.

2.3.1 Types/Classifications of Open Source Intelligence

OSINT is a broad term and under each umbrella falls information that is easily and readily available to everyone. But there are different types of OSINT and this classification is presented below.

Human intelligence (HUMINT) + Social Engineering

The North Atlantic Treaty Organization (NATO) Glossary of Terms and Definitions defines HUMINT as “Intelligence derived from information collected by human operators and primarily provided by human sources” [63]. HUMINT, in the literature, is usually encountered in cases of an individual conducting espionage, but can also be information that is acquired through diplomatic dialogue or liaison exploitation [64].

Social Engineering is similar to HUMINT, but is focused on social interactions. In Mouton et al. [65], the authors gather existing definitions of social engineering and propose a more structured definition as: “the science of using social interaction as a means to persuade an individual or an organisation to comply with a specific request from an attacker where either the social interaction, the persuasion or the request involves a computer-related entity”. Hatfield [66] provides an evolution of this concept starting from its first appearance in a political context in the 19th century to its eventual migration to the field of cybersecurity. According to Krombholz et al. [67], social engineering can include physical attacks (dumpster diving), social attacks (baiting, use of alleged authority), reverse social engineering (where the at-

tacker tricks the victim into contacting them), technical attacks (usually carried out over the internet), or a combination thereof. Of course, due to the increasing use of social media, it is natural that social engineering attacks increasingly focus on targeting users on social media.

Social Media Intelligence (SOCMINT)

SOCMINT is one of the newest members of the intelligence family, made necessary by the rapid development and increasing usage of social media since the beginning of the 21st century. SOCMINT differs from other traditional forms of intelligence because it can be viewed as a starting point for political, economical and social knowledge production [68]. Due to the ever-evolving nature of crime, it renders older models of intelligence less robust in this new digital era. It is up to police agencies to keep up with the times and be proactive in their fight against crime.

SOCMINT becomes more useful when it is applied to groups or individuals for establishing behavioural patterns [69]. Social media nowadays is used not only for communicating with people, but from things like organising social protests [70] to spreading extremist propaganda [71]. For this reason, SOCMINT can be utilised to predict and identify online threats [72, 71], as well as for gaining insight into group relations and online interactions [73].

Crowdsourcing

The term crowdsourcing was coined in 2006 by Jeff Howe [74]. Crowdsourcing is different to outsourcing because it is using the efforts of a virtual crowd to perform specific tasks [75]. When it comes to criminal investigations, crowdsourcing can be described not as harnessing crowd resources, but as collecting investigative leads by the public to aid an investigation. There are plenty of advantages to crowdsourcing, e.g., the lower cost and the speed, because the network of people involved in the investigation is larger and varied (amateurs and professionals). Furthermore, crowdsourcing is flexible, as it is not hindered by time zones, public holidays, bureaucracy, and can be scaled easily from a local to a global scale [76]. Users from all over the world can participate in crowdsourcing activities, such as Close-Circuit

Television (CCTV) monitoring or footage analysis, from their computers from their home or office. A study of four such cases from the UK is presented by Trottier [77].

2.3.2 Open Source Intelligence in a Law Enforcement Context

Collecting available information and leverage it to generate useful leads was performed by law enforcement already before the digital era. During a typical crime investigation, they use and act on knowledge they acquire though traditional sources, such as victim and witnesses accounts and physical evidence, in order to solve the crime. Such collection of evidences can nowadays be enriched by online sources thanks to existing OSINT techniques. Furthermore, the monetary and manpower costs of those tools during an investigation are both minimal.

Social and Media Monitoring

Social Network Analysis (SNA) is used by LEAs to identify the relations between different entities of a criminal network [78]. SNA is effective for collecting evidence, analysing interactions and online activities, deriving information about criminal activity as well as the patterns and ties of the involved actors. Van der Hulst [79] gives an analysis of SNA as an investigation and intelligence tool and a protocol draft for handling network data.

This typical procedure may sometimes miss crucial evidence that are solely located online, justifying why such analysis is nowadays considering online sources and more specifically social networks. Integrating social media sources into the investigation can help police officers make more educated decisions. These sources also complement the evidence they have already acquired through traditional means. Social media can be a point of convergence for data and information, and this is also precisely what makes them useful in an OSINT investigation [80]. The integration of social media to a law enforcer's toolkit is usually done as part of an ongoing investigation or as a preventative measure, to be obtained through continuous monitoring and data mining of known malicious online domains. Of course, social media monitoring has to be performed alongside OSINT investigation in order to enrich the level of understanding of a particular target as well as to help verify the

validity of information [81].

SOCMINT can be performed in real-time to monitor and intervene in a situation [69]. Social media with location tagging features such as Snapchat and Instagram, and most notably Twitter with its hashtag function, can provide a valid image of the real time developments on a certain topic or the current situation in a specific location. A similar approach is the processing of CCTV footage, either during criminal investigation or for monitoring purposes [82]. According to Trottier [83], the monitoring of public or semi-public spaces through private or public means enables LEA to take hold of information that would otherwise be considered fleeting and morph it into intelligence. The same can be said for online monitoring of open sources and social media accounts where users interact the same way they would do face to face, with the difference that the information that is exchanged is not ephemeral as speech but forever stored on the internet.

Those capabilities provide almost real-time information that can be determinant during an investigation, allowing sometimes an instant reaction [84]. Digital traces left online by criminals can lead to location information or evidences about criminal activities [85].

Crowdsourcing Contributions

Aside from obtaining publicly available information, law enforcement have also identified the advantage of leveraging the collective knowledge of the public in a crime investigation. A good example of the effect of crowdsourcing in a criminal investigation is in the case of the Boston Marathon bombing in April 2013. Citizens engaged in their own investigation of the case in real time, on Twitter and online forums like Reddit [86]. Often, the news of a breakthrough would reach Twitter before news agencies reported it. Citizens, amateurs and professionals pooled their resources, studied photos and videos from the scene of the bombing and performed forensic analysis on the evidence they collected [87]. While their endeavour did not correctly pinpoint the culprits, it was a useful assistance to law enforcement personnel who used the leads and efforts of the public to successfully identify and catch the perpetrators [88].

There are initiatives targeting the power of crowdsourcing for aiding in an inves-

tigation. Most notably, Europol's "Trace an Object" [89] initiative to help combat child abuse, asks individuals to examine objects in the background of images with sexually explicit material involving minors, with the aim of identifying the origin of the object. Another such initiative is TraffickCam [90], which asks users to upload images of hotels they have stayed at in order to create a database of hotel rooms. This database can then be used by an investigator, who can compare images recovered through an investigation to those in the database with the aim of finding the location of the crime.

Of course, turning to the public for leads in a crime investigation means that a huge number of responses can be expected. For the first year of the Trace an Object initiative, Europol reported 21,000 leads sent by citizens for 119 objects, resulting in the identification of 79 objects in total and in 32 cases, in the identification of the country of production [91]. This overwhelming amount of leads though means that LEA need to implement procedures for handling, filtering and evaluating this information. One such case is of the Netherlands National Police and their use of an Artificial Intelligence (AI) agent messaging processing tool about the messages they receive through the Interpol Channel [92].

Digital Forensic Intelligence (DFINT)

The application of knowledge gathered from OSINT can be incorporated with the information already gathered in a traditional investigation, where one source aids the other. Quick and Choo [93] proposed a framework called DFINT + OSINT, which aims to use OSINT in conjunction with previously used digital forensic intelligence with the aim of finding even more useful information about crimes based on already collected data. The authors developed a tool called DRbSI (Data Reduction by Selective Imaging), which reduces the amount of data that need to be looked at, and an Entity extractor that processes data types found in the DRbSI subsets and merges them into a single source.

OSINT tools: A non-exhaustive list

There are many tools in existence that digital investigators make use of to complement their investigations. In addition to paid tools, there is a variety of online OSINT

tools that quickly gather and cluster information in ways that could be useful to an investigation. There is a large number of tools available, many of which have duplicated functionality. Two useful lists of tools are the Awesome OSINT List ’[94] and the OSINT Framework [95]. These lists contain tools that can be useful in an investigation but also tools for marketing insights, etc. In Table 2.3.2, an indicative list of tools that can be useful to an investigator when looking at the online presence of a suspect is presented. As can be seen in this table, these tools can provide useful insights for the online presence of a suspect, such as the users they most interact with, the topics they most care about and even their sleeping patterns.

Function	Example Tools	Notable Usage
Automation Suites		
Maltego	https://www.paterva.com/	Entity transformations
theHarvester	https://github.com/laramies/theHarvester	OSINT gathering from multiple sources
Spiderfoot	spiderfoot.net	Scanning and monitoring open data sources
Twitter		
Twitter ID	gettwitterid.com/, tweeterid.com/	Unique numerical identifier
GPS enabled tweets/geocoding	geosocialfootprint.com/	Estimate of likely location based on social check-ins and geocoding
Sleeping Patterns	sleepingtime.org/	Sleeping Patterns of specific user
Record of profile changes	spoonbill.io/	Profile changes of specific users
Trending topics by location	trendsmap.com/, tweetarchivist.com/	Tracking and analytics of users and topics
Sentiment analysis on hashtags	socialbearing.com/	Analytics on twitter usage including sentiment analysis and hashtag use
Visualisation of a twitter community	burrrd.com/	Insights including top connected users and top topics
Facebook		
Find Facebook ID	findmyfbid.in/, lookup-id.com/	Unique numerical identifier
Facebook Search	facebook.com/help/821153694683665	Facebook's inherent search tool
Who Posted What	whopostedwhat.com/	Search by date, location or Facebook UID. Works on Instagram too
Email		
Email Format	email-format.com/	Find the email format of a company
Email Permutator	metricsparrow.com/toolkit/email-permutator/	Permutations of possible email addresses
H8mail	github.com/khast3x/h8mail	Password hunting tool that matches email addresses to leaked passwords
Reverse Email Lookup	thatsthem.com/reverse-email-lookup	Returns useful information associated with an email address
We Leak Info	weleakinfo.com/	Data breach search engine (search by email, username, password, hash, etc)

Table 2.1: A non-exhaustive list of OSINT Tools

Legal and Ethical Considerations

However, the potential intrusive nature of OSINT, and more especially SOCMINT, should not be ignored. Guidelines needs to be established on how law enforcement officials can collect information with respect to the privacy and confidentiality of citizens [69]. Frequently, the information a police officer might be looking for can be found online, but behind a safety net of privacy settings. There are cases where this digital limit has been circumvented through a friend of the potential suspect who had access to this information and offered it to the police [96].

It is furthermore of the utmost importance that law enforcement check the validity of the information they have acquired, to ensure they are accurate before they act on it [97]. For OSINT investigations, a methodology should be adopted, similarly as for traditional and digital investigations, i.e., audit trail, chain of custody, etc. Additionally, the processing and storage of personal data should be done with respect to the laws of the country the investigation is conducted in.

2.4 Password Analysis

It has been more than 50 years since the concept of passwords was introduced and adopted across society as a digital authentication method. Despite alternative authentication methods being developed later, it is reasonable to assume that this prevailing authentication method will not fall out of popularity anytime soon. The average number of password-protected services and devices per user is a difficult figure to estimate, and the user themselves frequently fail at estimating correctly. Naturally, each password is closely connected to its creator, and the sheer number of passwords a user might be asked to create and remember can result in unsafe password management practices.

2.4.1 Password Strength

Well aware of the weakness of human-chosen passwords, attackers aim at guessing passwords to gain access to services or data [98]. One way to better protect a service is to make sure that the password chosen by the user would resist the efforts of a potential attacker. Password metrics are therefore needed in this context, providing a measure of the strength of the password. Such a score can be the result of the combination of length, complexity, and unpredictability of the used password or trying to evaluate the number of guesses an attacker should perform before retrieving the password [99]. These metrics have a large variance, as it was shown that checking the same password in different meters can give highly inconsistent strength outcomes [98].

Many popular web services use password-strength meters to give feedback to users while they create new passwords, which might affect user behaviour during password creation. These password strength meters utilise password policies in order to guide users and help them develop safer password creation habits. One of the most well known password policies was introduced in 2013 by the National Institute of Standards and Technology (NIST) [100] and it requires passwords to be at least 8 character long, with uppercase, lowercase, digits and special characters included [101].

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/incrementing a digit at the end. 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 [102].

There are many password strength meters available, and many companies create and use their own that are based on their company's password policy. Therefore,

password123

Services	Strength scores	
Apple	Moderate	2/3
Dropbox	Very Weak	1/5
Drupal	Good	3/4
eBay	Invalid	2/5
FedEx	Very Weak	1/5
Google	Weak	2/5
Intel	Oh No!	1/2
Microsoft (v1)	Medium	2/4
Microsoft (v2)	Medium	2/4
Microsoft (v3)	Medium	2/4
PayPal	Weak	2/4
QQ	Moderate	3/4
Skype		-/3
Twitter	Obvious	2/6
Yahoo!	Strong	3/4
Yandex		-/4
12306.cn	Average	2/3

Password.1

Services	Strength scores	
Apple	Moderate	2/3
Dropbox	Very Weak	1/5
Drupal	Strong	4/4
eBay	Invalid	2/5
FedEx	Very Strong	5/5
Google	Fair	3/5
Intel	Oh No!	1/2
Microsoft (v1)	Strong	3/4
Microsoft (v2)	Medium	2/4
Microsoft (v3)	Medium	2/4
PayPal	Strong	4/4
QQ	Strong	4/4
Skype		-/3
Twitter	Okay	5/6
Yahoo!	Very Strong	4/4
Yandex		-/4
12306.cn	Average	2/3

@";;:!&*_)+?*!#

Services	Strength scores	
Apple	Weak	1/3
Dropbox	Great!	5/5
Drupal	Fair	2/4
eBay	Invalid	2/5
FedEx	Very Weak	1/5
Google	Strong	5/5
Intel	Congratulations!	2/2
Microsoft (v1)	Weak	1/4
Microsoft (v2)	Strong	3/4
Microsoft (v3)	Strong	3/4
PayPal	Weak	2/4
QQ	Weak	2/4
Skype		-/3
Twitter	Okay	5/6
Yahoo!	Strong	3/4
Yandex		-/4
12306.cn	Average	2/3

Figure 2.3: Comparison of strength scores for various online services [9]

an issue arises when different strength meters give different results in terms of how secure a password is, which can be confusing for users. This is detailed by Carnavalet and Mannan [98] in their article, and an online tool they have developed, offering a comparison of strength meters from various services, showcases this discrepancy [9]. Figure 2.3 shows the strength scores for some passwords across various popular online services.

2.4.2 Hashing and Salting

One of the most common way passwords of users are obtained by attackers nowadays is directly from where they are stored in the database of the relevant system/website. Until recently, many such credential collections (username and password) were even stored in plaintext (also known as cleartext), i.e., there was no hashing involved, and the passwords appeared "in the clear", or in readable form. This means that if an attacker gained access to the files where this information was stored, they would be able to access the account of every user in that file, some of which could be accounts with extended privileges. This would represent a massive security breach which could seriously compromise confidential information or disrupt services.

In order to ensure that sensitive information remains privileged, hashing has been introduced. As stated by Leurent and Peyrin [103]: "a cryptographic hash function H is a function that maps an arbitrarily long message M to a fixed-length hash value of size n bits." The message, is also known as input, and the fixed-length output is known as the hash or message digest. As stated by OWASP's (Open Web Application Security Project) Guide to Cryptography [104], a hash function is selected in such a way that it is easy to generate the hash of a message, but very difficult to re-generate the message if only the hash is known. Another characteristic of hash functions in cryptography is that it is difficult to select an initial message with

the goal of it matching a specific hash.

There are many different hashing algorithms, some still in use and some that have been rendered obsolete, either because of the rapid increase in computing power available or because of security vulnerabilities. One of the most well-known hashing algorithms that is still used to this day, even though it has been proven to be no longer collision resistant, is the MD5 function, which was introduced by Robert Rivest in 1992 [105]. Another family of hash functions are the Secure Hash Algorithms (SHA), algorithms that were designed by the NIST, with SHA 256 being one of the most used algorithms today [106].

One extra layer of security before hashing a message with a hash function is to add an extra set of characters as padding, either at the beginning or the end of the password. This will create an entirely new hash, making it difficult for attackers to use pre-calculated tables of hashed passwords. This process is known as salting [107]. An illustration of the hashing and salting processes can be found in Figure 2.4.

2.5 Data Breaches

One of the most serious problems faced in the domain of password security are data breaches, i.e., the breaching of sensitive information from where they are stored for safekeeping. When a data breach happens, datasets of credentials (most often featuring usernames and passwords but sometimes also including other sensitive information such as bank details, social security numbers and addresses) are unlawfully obtained by adversaries. These datasets are then publicly released or auctioned off to the highest buyer. This information can be used for tailored phishing attacks or - especially because of users' password reuse across different services - to enter other systems with critical information and cause damage.

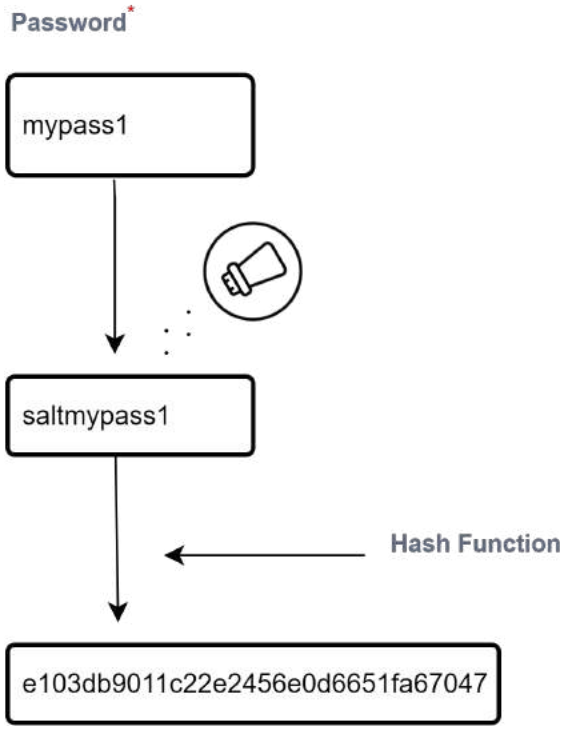


Figure 2.4: Hashing and salting a password

Figure 2.5 shows some of the biggest data breaches and hacks that have occurred during the last decade. As it can be observed in the figure, data breaches represent a serious issue for users' and services' online security. It is therefore prudent for companies to exercise every measure of security at their disposal to hash and safely store that information in their servers.

Data breaches can offer attackers a serious insight into the thought process behind password creation. Lists of leaked passwords have been analysed for this purpose, both by researchers and malicious actors. One of the most important observations is that in many cases, users tend to chose simpler, easier-to-remember passwords. According to the password manager tool NordPass, the 20 most popular passwords of 2022 are shown in Table 2.2. It can be observed that the majority of these passwords are simply number sequences of various length, but easily pre-

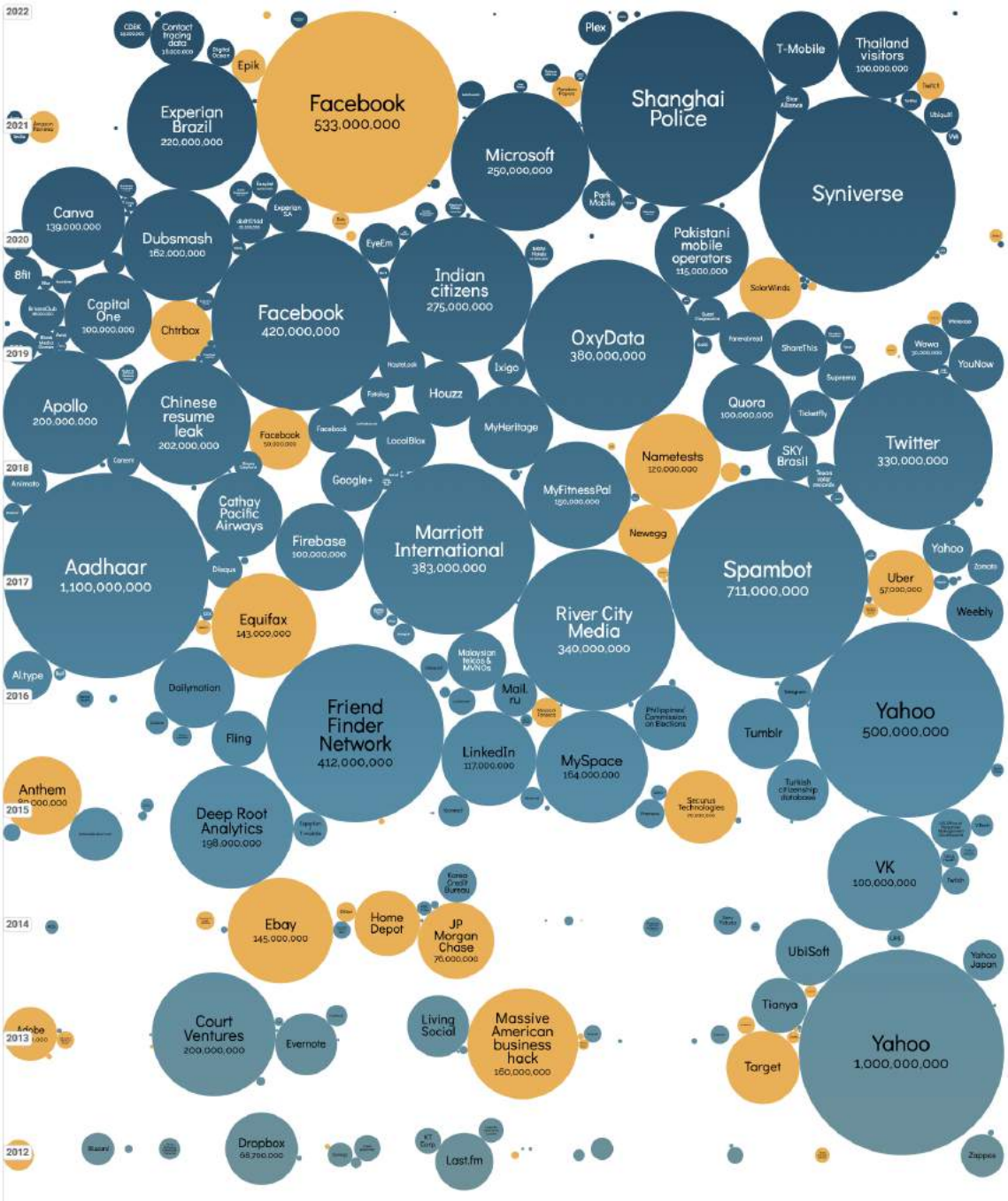


Figure 2.5: World's biggest data breaches & hacks
figure by [108]

Table 2.2: Most popular passwords of 2022 [10]

1	password	11	1234567
2	123456	12	1234
3	123456789	13	1234567890
4	guest	14	000000
5	qwerty	15	555555
6	12345678	16	666666
7	111111	17	123321
8	12345	18	654321
9	col123456	19	7777777
10	123123	20	123

dictable. Only a few non-numerical passwords are found in the Top 20, but these also relate to either sequences of keyboard characters (qwerty) or the password selection process itself (password, guest). Interestingly, NordPass analysed a 3TB database of passwords to assemble this list, and it also provides further breakdowns by gender for more than 30 countries with some interesting words making up the Top 10 in different countries. For example, football teams featured a lot in the Top 10 of Italy, Portugal and the UK and 9 out of 10 passwords from Israel were number passwords, with the 10th being alphanumerical.

Chapter 3

Related Work

3.1 Introduction

As far as a digital investigation is concerned, more often than not, a law enforcement officer will find themselves in a situation where gaining access to a digital device or computer system will be of the utmost importance for the course of the investigation. Password-based schemes typically protect access to those devices, as they remain nowadays the most used authentication method and are unlikely to vanish in the coming years [109].

Lots of effort is put in place to on one side strengthen those mechanisms and enforce users in choosing safe passwords, and on the other side, improve the password cracking techniques to gain access, often illegally, to systems. There is a common belief that hackers are always a step ahead of defenders and sometimes defenders will suffer penalising [110]. Nevertheless, both approaches can be beneficial to law enforcement and contribute to the success of an investigation.

Retrieving a password is not the only way to penetrate a system, as many other threats can be exploited by an adversary [111]. However, taking into consideration that the majority of users follow common patterns in password creation, the chances to retrieve a password are high [112] making it one of the most targeted methods

leveraged by adversaries.

If the purpose is to retrieve a single successful password out of a set of users instead of a targeted one, the success ratio is even bigger, because password cracking techniques can be used concurrently for all targeted accounts, and it is very possible at least one has a weak, easy to crack password, providing a point of entry to a system. There is a vast array of password cracking techniques, that are used depending on the situation, from the traditional ones, like an exhaustive search, to the recently developed ones, like machine and deep learning-based techniques, such as the ones based on Generative Adversarial Network (GANs) [113]. A wide range of tools and methods are available to perform such password cracking processes, which is useful not only in terms of a lawful investigation but also for penetration testing and account recovery purposes. This section provides an overview of this field of research.

The rest of the chapter is organised as follows: Sections 3.2 to 3.5 look at password cracking techniques, from the classics like brute force attacks, rainbow tables and dictionary attacks to the more current techniques that leverage AI and ML. Section 3.6 shows some of the current state-of-the-art password cracking tools, while Section 3.7 examines the future impact of quantum computing in password cracking. Section 3.8 describes the factors that play an important role in password creation, specifically looking at password reuse, user's preconceptions regarding the security of their passwords as well as demographic factors that might influence the security of the password. Section 3.9 looks at the estimation of password strength, commercial strength meters, and the role of password policies, while also looking at the latest advances in the field. Finally, Section 3.10 provides a conclusion to the chapter, highlighting the important concepts and identifying the gap in the literature that this thesis aims to address.

3.2 Brute Force Attacks

The most well known and straightforward method to recover an encrypted password is to try all possible combinations, which is known as an exhaustive search or a brute-force attack [114, 115]. In this instance, if for example it is known that a password is 8 characters long, and it includes only lowercase letters of the English alphabet, that means that every combination of letters in every order must be checked from “aaaaaaaa”to “zzzzzzzz”. This equals to 26^8 combinations.

This means two things. Firstly, a brute-force attack has a success rate of 100% - if all possible passwords candidates are hashed and then checked against the hashed password, the password will be found. Secondly, when considering a rich search space, i.e., including lowercase, uppercase, symbols and numbers with a password’s length higher than a certain threshold, it becomes quickly unpractical to perform an exhaustive attack. This is because the number of possible combinations increases exponentially as the length of the password does.

With password cracking being a highly parallelised process since it is ideal for being split into subtasks, the time each password cracking attack will take is also highly dependent on the hardware. Graphic Processing Units (GPUs), due to having many thousands of cores, are ideal for parallelisation and are preferred to Central Processing Units (CPUs) for password cracking. For example, on benchmarks that were executed on the latest NVIDIA GPU that was released in October 2022, the NVIDIA GeForce RTX 4090, it was shown in benchmark tests that the cracking speed for MD5 hashes was 164.1 GH/s (Gigahashes per second) [116]. This makes a brute force attack an extremely inefficient attack for anything other than short passwords, hashed with fast hash functions.

3.2.1 Personal Identification Number (PIN) Based Attacks

As mentioned above, brute force attacks are one of the most common types of cyberattacks, and one of the reasons why is how easy they are to deploy. Frequently, little knowledge about the underlying mechanisms is necessary, and an attacker can easily exploit vulnerabilities in a poorly thought setup. In the case of PIN-based attacks, there are two main approaches, either aiming to guess the PIN or trying to reset the counter of how many unsuccessful tries are accepted. The downside to the latter is a vulnerability against DoS attack [117].

One case where an inherent flaw design reduced significantly the amount of possible guesses is that of the Wi-Fi Alliance in 2007. In this case, if the Wi-Fi Protected Setup (WPS) authentication failed, the access point would send back a message that allowed the attacker to pinpoint which part of the PIN was not entered correctly [118]. This design flaw allowed a significant reduction in the amount of guesses needed to recover the PIN with a brute force attack, thus making the success of the attack much more likely.

Brute force attacks for PIN-based devices can also be used in conjunction with other techniques. For example, in a case of sound delays in the acoustic feedback when pressing buttons in Automated Teller Machine (ATM) PIN keypads, a Markov Model was first employed to reduce the search space which made the subsequent brute force attack more achievable [119]

3.2.2 Distributed Approaches

A brute force attack presents as an ideal method for distributed approach to password cracking. In fact, it was shown that the benefit of concurrently computing hashes in a distributed system using Message Passing Interface (MPI) took the time it would take to find a 5 character password from 83 seconds with the pass-

word cracking tool Cain & Abel to just 8 seconds for the distributed approach [120].

The above result highlights even further the need for enforcing strict password policies and secure encryption on the side of the server. In fact, this is a serious threat for Secure Shell (SSH) servers on the internet, where attacks using a number of systems attempt multiple combinations of usernames and passwords in an attempt to find one where a weak password has been selected [121]. The problem of detecting these types of distributed attacks becomes even more difficult when attackers try them at a low rate, it becomes difficult to distinguish them from real users failing to input their password correctly [121].

The use of massive networks of bots complicates things even further, as it makes it even harder to detect an attack when hundreds of different bots query the server, as it becomes harder to distinguish between legitimate users and bots and block their IPs [122]. An example of a botnet attack is illustrated in Figure 3.1. Of course, this type of massive operation comes with a significant cost to the attacker too, and it also presents the problem of communication and synchronization between all these systems.

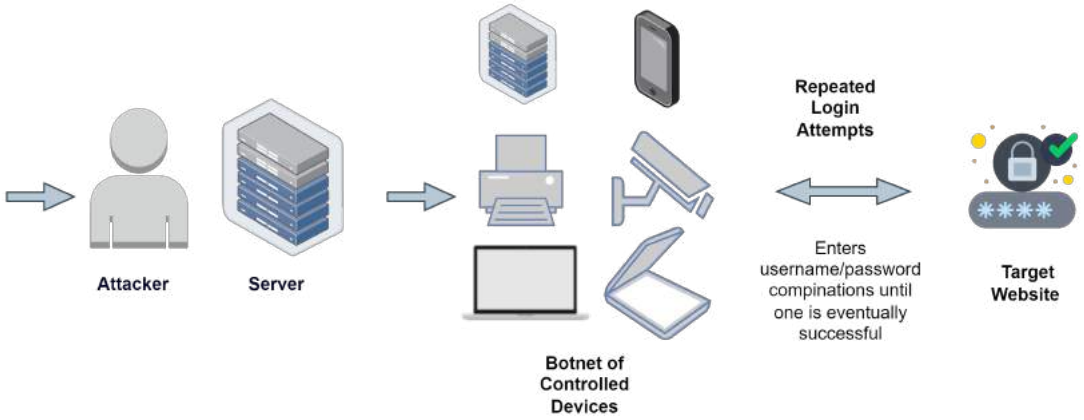


Figure 3.1: How a distributed botnet brute force attack works

3.2.3 Benefits and Limitations of the Brute Force Attack

The largest drawback of this resource-intensive approach is that it quickly reaches its limits when the password sought is long and/or using a rich alphabet, i.e., alphanumerical with special symbols [115]. If that is the case, the number of combinations that need to be tried skyrockets, and it quickly becomes impossible to do so in an acceptable time frame. This is also one of the best strategies for mitigation against brute force attacks, the selection of passwords that are long and complex enough to render this type of attack useless.

Other ways to mitigate against brute force attacks include the use of Two-Factor Authentication (2FA) – where, even if the brute force attack has been successful, a second means of authentication is required to gain access to an account. Furthermore, for online services, limiting the number of attempts one can make to enter the password after they have been unsuccessful a few times or increasing the time between subsequent attempts can also render a brute force attack inconvenient and unsuccessful.

3.3 Rainbow Tables

Hellman tables [123], a time-memory trade-off allowing to retrieve efficiently the input of a one-way function, can be used to retrieve the password in a very efficient manner. Many improvements to Hellman tables have been proposed since then [124, 125, 16, 126], in terms of shortening the time span, increasing searching efficiency, success rate, space utilisation, etc.

A Time-Memory Trade-Off (TMTO) approach relying on the principle of the Hellman table is Rainbow Tables [127] which was put forth in 2003. This approach is focused on mitigating the time required to explore a given space. Rainbow tables are based on the idea of TMTO, which focuses on pre-computing an almost exhaus-

tive 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.

Therefore, for the price of some storage capacities and the pre-computation of the space to be explored, a password belonging to this given predefined space can be retrieved in a negligible time compared to the pre-computation step. This is highly valuable if one knows that several passwords are meant to be encountered in practice.

Many projects have worked collaboratively to generate such rainbow tables, e.g., the rainbowcrack project [128], for the functions MD5, NTLM and SHA1. While efforts are still made to improve the performance for generating such rainbow tables, it is rendered almost useless in the field of password cracking due to the popular usage of a *salt* in the storage of passwords.

A salt is a random string concatenated to the password before using it as the input to the hashing function. Theoretically, rainbow tables could still be built for salted passwords, but the defined space to explore must incorporate this salt. In practice, the pre-computed table cannot be adapted to such value except by integrating it during its generation, making such a task impossible due to the number of potential salts. There is no fixed length for a salt, but it is generally long enough, e.g., 32 bits or more, to render rainbow tables no longer usable in practice.

Using salted passwords has the additional benefit that two identical passwords should have two different salts (as they are randomly generated) and will therefore have two different hashes in the database.

Rainbow tables are no longer widely used for password cracking, except in specific scenarios, not only because of the use of salts, but because it is nowadays

computationally feasible to perform millions of guesses in a short amount of time, depending of course on the hash function [129]. 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 pre-computation using cryptanalytic TMTOs [130]. An improved rainbow table attack that uses a dictionary generator for more complex and longer passwords showed a success rate of 83% in a case where the composition of the passwords was known and matched the ruleset applied to the dictionary [131].

One of the biggest advancements in password cracking methods lies in the acceleration that can be achieved as hardware evolves and moves away from using CPUs for computation. It was shown, that in the case of Rainbow Tables, the use of FPGAs (Field Programmable Gate Array) provided a 1000x speed-up to the corresponding software approach without compromising the probability of success [132]. Another similar experiment using FPGAs for creating Rainbow Tables for the A5/3 (Katsumi) block cipher showed that a 9x speed-up could be achieved for single engines and a 550x for a parallelised architecture with 64 engines [133].

3.4 Dictionary Attacks

A dictionary or wordlist attack makes use of predetermined lists of words as possible password candidates for password cracking. They are given their name by the fact that these attacks used to find words from the dictionary to use as possible password candidates - lately these dictionary lists are more refined and tend to include passwords found in previous data leaks.

Compared to a brute force attack, a dictionary attack has a much more limited search space and aims to leverage the fact that users prefer dictionary words to meaningless strings of characters, as they are much easier to recall. This means

that it's easier and faster to execute, and it is one of the most popular methods of password cracking used nowadays.

Password cracking attacks, like dictionary and brute-force attacks, are considered as the most commonly used online and offline, while attacks carried out to leaked password databases are offline attacks [134, 135]. When considering offline attacks, dictionary attacks have the advantage of time over brute-force attacks, as the number of guesses is always smaller than the latter, even though it can be fully customised with the use of mangling rules.

A comparison of a brute force attack to a dictionary attack showed that while the brute force attack was more effective for shortest passwords (6-7) characters, the dictionary attack found more passwords above 8 characters [122]. A combination approach in the same study presented more balanced results. In another password cracking study with brute force, dictionary and hybrid attacks on real-world passwords provided by users in a Slovenian University, it was again shown that the majority of passwords were cracked. The comparison between the cracked and uncracked passwords showed length playing a contributing factor, with the authors stipulating that the security of textual passwords would be further compromised with the increase of computing power [136].

Dictionary attacks can be used in a variety of attack scenarios. A simulated dictionary attack on WordPress on a fictitious person by [137] shows the requirements and tools needed to execute such an attack, as well as countermeasures and procedures for the forensic investigation of a dictionary attack. In a scenario of two-factor authentication using smart cards, it was shown by Wang and Wang [138] that once the smart card security parameters are compromised, the corresponding password factor can be guessed via an offline dictionary attack. Work has focused on the discovery and detection of online dictionary attacks, with [139] simulating SSH-based break-in attempts in a university network.

When considering a dictionary in a broader sense, dictionary attacks have been successful in a variety of biometric security factors. One such example is a dictionary of fingerprints (real and synthetic) which has successfully been used to match a large number of fingerprints, thus bypassing fingerprint security [140, 141]. Dictionary attacks have also been successfully used for speaker verification. It was shown that synthetic voices could match 20% of female and 10% of male voices for the most secure configurations, with these figures rising as much as 80% and 65% of female and male voices respectively for the least secure configurations [142].

3.4.1 Password Mangling

A common approach is the dictionary attack in combination with “mangling rules”, which are grammar substitutions and modifications that aim to imitate human tendencies during password selection. This can be one of the most important elements of a successful dictionary attack. For example, using the number *3* instead of the letter *e*, adding a *!* at the end of the password, capitalising the first letter, etc. Those rules can be manually designed or automatically learnt from previously cracked passwords [143].

The inclusion of mangling rules can produce password candidates that are likely to have been selected by users when they are trying to comply with the password policies that are enforced in the applications and services they use. For example, if the password “dragon” is not accepted because it only contains lowercase letters, and it is not of sufficient length, a password like “Dr@gon12” might be used instead, where the first letter is capitalised and numbers and special symbols are also included in the password. If the attacker has included a good ruleset, these added security steps on the part of the user, which he might believe make their password more secure, do not really pose a hindrance to the attacker.

It is important to note that mangling rules widen the set of guesses significantly,

Table 3.1: Example rulesets for mangling [11]

Name	Function	Description	Example Rule	Input Word	Output Word
Nothing	:	Do nothing (passthrough)	:	p@ssW0rd	p@ssW0rd
Lowercase	l	Lowercase all letters	l	p@ssW0rd	p@ssw0rd
Uppercase	u	Uppercase all letters	u	p@ssW0rd	P@SSW0RD
Capitalize	c	Capitalize the first letter and lower the rest	c	p@ssW0rd	P@ssw0rd
Toggle Case	t	Toggle the case of all characters in word	t	p@ssW0rd	P@SSw0RD
Reverse	r	Reverse the entire word	r	p@ssW0rd	dr0Wss@p
Duplicate	d	Duplicate entire word	d	p@ssW0rd	p@ssW0rdp@ssW0rd
Duplicate N	pN	Append duplicated word N times	p2	p@ssW0rd	p@ssW0rdp@ssW0rdp@ssW0rd
Rotate Left	{	Rotate the word left	{	p@ssW0rd	@ssW0rdp
Rotate Right	}	Rotate the word right	}	p@ssW0rd	dp@ssW0r

because each new alteration has to be checked with all the password candidates contained in the dictionary. This is why a balance has to be achieved between the number of mangling rules to be tested and the time that can be afforded for the recovery process. There are common mangling rules that are used by the community, such as the Hashcat [144] Best64 rules and other more extensive rulesets.

For example, the Best64 rules include appending popular single and special numbers at the end of the password, appending or overwriting high frequency characters, “leetifying” which is the process of replacing letters by similarly looking numbers or special characters and rotating the password. Some of these substitutions can be seen in Table 3.1 These mangling rules can also be generated automatically from data breaches, e.g., using the PACK suite of tools [145]. The input for these tools can be a list of passwords obtained from one or several data breaches or humanly designed on purpose.

3.4.2 Password Cracking Dictionaries

Dictionary attacks remain to this day one of the most popular password cracking attacks, especially for offline attacks that allow unlimited attempts. Nowadays, they are commonly executed with the use of dictionary lists that stem from leaked lists of passwords from data breaches. The advantage of using real-world leaked pass-

words in a dictionary attack is that they already contain all the important information about the choices and habits of real users when it comes to password creation.

Of course, dictionary lists can also be created by attackers, in many cases by combining a number of different leaked password lists to customise to a specific target. For example, if the attacker is limited in their number of attempts, they might prefer a smaller dictionary list that contains the most popular passwords, relying on the knowledge that users often choice convenience and memorability over security.

Leaked dictionary lists can be used as they are for a variety of attacks, not just dictionary attacks. For example, they can be used to derive mangling rules as described in Section 3.4.1, thus creating a set of mangling rules that closely imitates user choices. They can also be used as training sets for a variety of Machine Learning (ML) password cracking techniques, as those that will be described in the upcoming Section 3.5. In many cases in literature, dictionary lists are also referred to as wordlists, and these terms can be used interchangeably.

3.4.3 Real-World Password Cracking Dictionaries/Leaks

As mentioned in Section 2.5, many significant data breaches have happened in the last 20 or so years that have exposed valuable information of users, usernames, passwords, biometric data, social security information and so on. In many cases, the passwords that have been exposed in these data breaches have been plaintext, therefore creating ideal dictionary lists for password attacks. This is because a hashed password list from a data breach, especially if it has been hashed with a slow hash function, cannot always be recovered in its entirety. In that case, it can be assumed that some of the hashes that have not been cracked could constitute some of the stronger, more secure passwords, therefore not accurately representing the entire spectrum of users' password choices.

Some of the most popular data breaches of the 21st century include social media

accounts such as the LinkedIn data breaches of 2012 and 2021, the Yahoo data breaches of 2013 and 2014 and the Myspace leak of 2013. These and more data breaches have been studied extensively for research purposes, with Layton and Watters looking at the tangible cost of some data breaches for the company and Confente et al. looking at the effect on reputation for the company. Studies have been conducted on large leaked lists of passwords from data breaches, such as RockYou dataset containing 32 million accounts [12] and the Yahoo dataset [109] containing 70 million accounts, for the purpose of extracting guessing statistics for these leaks.

3.5 Artificial Intelligence and Machine Learning Attacks

Similarly to such automated generation of rules, modern approaches to password guessing rely on a ML approach exploiting the enormous quantity of real human-chosen passwords from a leaked database.

3.5.1 Markov Models and other Statistical Models

Probabilistic Context-Free Grammar (PCFG) is one example of such a modern approach, initially released in 2009 [134] and recently updated to make it one of the most successful techniques. This approach is based on dictionary attack principles [148], and focuses on the calculation of the probability of each grammar [149]. They are based on Markov chains, and many password guessing tools are making use of them. PCFGs models are variants of context-free grammars, extending them similarly to how hidden Markov models extend regular grammars [150].

In one experimental scenario, Houshmand et al. [151] tried to use context-free grammars to recover passwords from popular data leaks stored on a hard disk. The

researchers implemented filtering techniques to reduce the number of tokens that could be potential passwords, and then calculated the probabilities of the remaining tokens using context-free grammars. Subsequently, three different algorithms for ranking the tokens were employed and the results obtained. Their results demonstrated that a robust filtering of the tokens, so that only strings that have well-known password characteristics will be checked, and the 1-by-1 ranking algorithm correctly identified the majority of passwords in the disk and even those strings that were mistaken for passwords were indeed very possible password candidates.

Ordered Markov Enumerator (OMEN) [152], is a Markov model-based password cracker that outputs password candidates in decreasing probability, thus speeding up the password guessing process. PRobability INfinite Chained Elements (PRINCE) [153] makes use of one input wordlist by creating "chains of combined words". PRINCE [153] creates intelligent chains to all combinations of words from the input wordlist.

These techniques have a good success rate when they are used to recover passwords from average users, as they are designed or trained to reproduce the average human behaviour. When considering a single targeted user, additional information might or should be considered to increase the success ratio. A simple example is that the chances of a dictionary attack relying on an English wordlist may be low if the target is not an English speaker.

3.5.2 Neural Networks and Generative Adversarial Networks

In the last few years, methods based on ML and neural networks have arisen, which in many circumstances have shown to outperform traditional methods. One such method is the neural network-based solution from Melicher et al. [154], which leverages a neural network to model human chosen passwords and assess their strength and guessability. This method outperformed state-of-the-art methods when

the guess number was high and the password policy not traditional. According to the authors, a big advantage of neural network-based methods over probabilistic context-free grammars is that they can be compressed without endangering the success of the outcome.

PassGan is another very well known approach which uses a GAN to learn password rules directly from leaked password lists and thus replaces human-generated password rules [113]. The GAN generated passwords were created using a subset of RockYou as the training set and managed to crack 43.6% of unique passwords from the rest of the RockYou dataset, and 24.2% unique passwords from the LinkedIn dataset. According to the authors, an advantage of this approach is that PassGAN can produce an unlimited number of guesses, unlike rule-based approaches which are limited by the number of rules and the size of the input wordlist. For PassGAN, this effectively increases the number of passwords found as the number of guesses increases. This becomes even more evident when the authors consider a hybrid approach of combining PassGAN to Hashcat.

Another development in the field considers the combination of PCFG with a GAN. This approach is called GENPass and consists of a PCFG + Long Short-Term Memory (LSTM) password generator, where LSTM is a kind of Recurrent Neural Network [155]. In their experiments, the authors looked especially into cross-site tests with their model considering multiple datasets for training with a GAN, therefore ensuring that the output dataset is general to all. This provided a higher number of passwords found compared to just using a few different input datasets.

Looking to improve the guessability of existing methods, Yang et al. proposed VAEPass, a lightweight password guessing model that uses a Variational Auto-Encoder (VAE), where its encoder and decoder are Gated Convolutional Neural Networks (GCNNs). The authors demonstrated that this method outperformed PassGAN on both one-site and cross-site tests, while also managing to do so in 11% of

the training time [156].

3.6 Password Cracking Tools and Algorithms

There are many different password cracking tools, some of which have been around for years, like John the Ripper [157] and Hashcat [144]. Many commercial, free and open-source passwords guessing tools are currently available, e.g., Passware [158], Elcomsoft [159]. Those tools simultaneously leverage both the CPU and GPU to increase performance. There are also FPGA approaches, such as SciEngines dedicated hardware [160]. However, FPGAs are typically a more suitable choice to evaluate specific functions, especially when power consumption is an issue [161].

Password cracking contests are also often organized, helping to better grasp the capacity of experts in retrieving passwords; the most famous of which being the *Crack Me If You Can Contest* [162] from KoreLogic held during DefCon.

Password cracking tools are mainly used for criminal purposes, although they can be used legally from law enforcement officers or administrators. These tools have evolved over the years in order to keep up with the ever-changing password landscape and nowadays, there are applications that work on various platforms and OS supporting heterogeneous protocols and attacking multiple targets concurrently. A non-exhaustive list of some of the most popular tools used in password cracking is included below.

Password Cracker works on Windows and is a great tool for recovering lost passwords, as it allows access to most passwords stored in Windows applications [163].

Brutus Password Cracker also works on Windows, but it aims at retrieving passwords and usernames from websites, applications and OS. It uses dictionary attacks for password cracking, but while it works for a variety of online applications,

it cannot be used for email accounts or social media [164].

Cain and Abel is also a Windows-based password cracking tool. It uses a variety of techniques to crack encrypted passwords, and also offers extra features such as sniffing network traffic and recording Voice over IP (VoIP) conversations. Thus, it is widely used by a variety of specialists [165].

OphCrack makes use of rainbow tables to crack Windows passwords. It's a free tool with a simple interface [166].

John the Ripper is one of the most popular password cracking tools available, and it is utilised throughout this thesis. It was developed in 1996 and can be used with Unix or macOS. It has various attack modes single crack mode, wordlist mode and incremental mode'[157].

THC Hydra is an open-source online password cracking tool that can be used for different protocols running under various OSs [167].

Medusa is a command line password cracking tool with a modular design that uses thread-based parallel testing to crack passwords of remote applications [168].

CrackStation is a free online service for password cracking. It is based on a Dictionary Attack, combining words and passwords leaks using precomputed lookup tables. The lookup tables were created by extracting every word from Wikipedia, as well as adding already leaked passwords from data breaches. Word mangling was also applied. The one drawback of these lookup tables is that they do not work with salted passwords [169].

Aircrack-ng is a tool for cracking Wi-fi passwords and network traffic monitoring. It can work on various OSs and supports cracking for Wi-Fi Protected Access (WPA) and Wired Equivalent Privacy (WEP) passwords [170].

L0phtCrack is an open source application for testing password strength and for password recovery in Windows. It employs a variety of attacks such as dictionary, brute-force and hybrid attacks and more recently rainbow tables [171].

RainbowCrack is a desktop tool for cracking password hashes in Windows and Linux platforms, using memory trade-off precomputed lookup tables [172].

Hashcat is another open source tool that is used in this thesis. It works on multiple OSs (Linux, Windows, macOS) and multiple platforms (CPU, GPU, etc...) and offers multiple attack modes for password cracking [144].

3.7 Future Impact of Quantum Computing

According to Grover's Algorithm, a quantum computer can offer a quadratic speed-up for the search of an unsorted database compared to deterministic and probabilistic methods that need O(n) steps [173]. As a result of this significant speed-up, the wider use of quantum computers in the future can result in a quadratic speed-up in password cracking techniques, and more specifically the computation of hashes.

It was shown by Dürmuth et al. that in large-scale attacks, a quantum computer can take advantage of the bias of human chosen passwords while still gaining the quadratic speed-up. In two experiments with real-world data, they showed that on a fixed user the number of hash evaluations falls to 6400 while in a scenario with the weakest 10% of passwords less than 200 hash evaluations per password are needed.

As a means to counter against this, quantum computer speed-up suggestions include making users choose longer, more secure passwords. This is a highly impractical solution, considering the difficulty for users to memorise passwords along with their tendency to reuse them, or not store them safely. Another suggestion put forth by Wang et al. is the use of quantum copy-protection of point functions for password verification. In this case, the authors avail of the property of quantum information to not be copied and use point functions that have the ability to map n-bit strings to a 1-bit string. Point functions are special functions that map n-bit strings

to a 1-bit string, in which case the password can be thought of as the point function and only the correct password can provide the correct output. Another work looking into how to leverage Quantum information for One Time Password (OTP) puts forth a Quantum OTP based on users' biometrics [176].

3.8 Users' Habits in Password Creation

A password is a sequence of alphanumerical and/or special characters used to validate that a user has the right to access a computer system, an application, or an online service. The average number of passwords users needs to remember is in constant evolution and diverge a lot, from 27 in one online survey [177], to 191 in another'[178]. Unfortunately, users find it difficult to recall and manage their passwords for all the accounts they maintain, and this results in inherent security issues [109, 179, 180].

3.8.1 Password Reuse

A typical consequence of this increasing number of passwords to memorise is that user either select easy-to-remember but weak passwords [181] or reuse their potentially complex password [182, 183], sometimes applying small modifications or simply following a predefined construction process [184]. These types of modifications ranged from simple substitutions of alpha characters by similar looking numbers, such as the English letter "e "replaced by number "3 "or number "1 "replaced by the special symbol "! ". These are common substitutions users resort to, when the password policy dictates that the password must comply with certain rules.

A study showed that 80% of users kept their current passwords when it was possible, while 16% changed the current password to one of the passwords they were using on another site and only 4% changes it to something completely new [185].

One of the biggest security problems arising from password-reuse occurs when considering data breaches. Following the European Union's General Data Protection Regulation (GDPR)'[186], users are notified when a service they are using is compromised, and they are strongly encouraged to update their credentials.

However, even when the user does so, the other accounts of the user that are protected by the same passwords are still at risk. According to a security report by the popular Virtual Private Network (VPN) service Surfshark, the number of data breaches that occurred in the third quarter of 2022 was increased by 70% compared to the second quarter, with more than 100 million records being leaked [187]. The increase in data breaches has been exponential in the past decade, with many well known companies falling victims in 2022, such as Twitter, Microsoft and Uber. These data breaches can expose passwords (plaintext or hashed) and sometimes accompanying information, such as emails, names, addresses, etc with potentially opening doors to many other services, some of them being critical for the user or the society.

For an attacker wanting to access a specific account belonging to an individual, the password cracking attack could be reduced to a simple lookup for a match in leaked lists. In fact, studies of password habits of users have shown that users tend to reuse passwords that they need to enter frequently [188] and they tend to underestimate the consequences of doing so.

Furthermore, even when passwords are not reused explicitly, there are password ties between older and newer passwords of the same user [189]. But, even in the good case where that particular user does not reuse the same password across different services, knowing their previous passwords or other information about them, can give great insight to the cracking process [5]. To this end, TarGuess, a framework that makes use of Personally Identifiable Information (PII) and cross-site information, has been proposed to make targeted guesses of users' passwords, and

it was shown to outperform current models for both the cases of non-savvy and security-savvy internet users [190].

3.8.2 Users' Preconceptions Regarding Password Security

But even if previous passwords of a user are not known, a case can be made that knowledge of passwords of other users can speed up the process [4]. In fact, studies have shown that there are common misconceptions people fall prey to when creating their password, such as thinking that by adding a symbol at the end of the password they make it more secure [184].

As revealed in an American survey with users from different background and ages [191], users have generally a biased understanding of password security. As highlighted in this study, participants have overestimated the security increase obtained by adding digits in the password, and underestimated the predictability of using keyboard patterns and common phrases. In a survey by Ur et al. [192], participants not only overestimated the added security of appending passwords with symbols or digits at the end, but also chose to reuse passwords or elements of passwords frequently. Another common phenomenon is the integration of personal information in the password chosen by users.

3.8.3 Role of Age, Ethnicity and Profession in Password Selection

In a study by Liu et al. [193], where more than 20 million pieces of data from Chinese users were analysed, and it was found that professionals used passwords with an average length of from 8 to 11 digits, while students tended to use shorter passwords. Concerning the complexity of passwords, they concluded that more than 50% of the users have passwords with only digits and less than 30% have a combination with special characters. The analysis also revealed that more than 12% of

the professional users include birthday and mobile phone numbers in the password and moreover an 11.5% used their username and e-mail to create the password.

In another study of Chinese passwords [194], the use of Pinyin in a pure form or in combination with dates and numbers accounted for 26% of the total, which seems to suggest that the use of English characters is widespread. It was also pointed out, that in the case of pure Pinyin passwords they were constructed with only 2-4 Chinese characters. In a similar vein, Bonneau [195] analysed 70 million passwords from the Yahoo data breach. Specifically, they explored whether dictionary lists optimised for the user-defined language preference would impact the success of password cracking. It was found that for all user-defined languages, the dictionary optimised for that language performed better than the global one, but some language crossover was found. For example, the dictionary optimised for Chinese users performed better for Greek users than the global dictionary (9.3% and 8.6% respectively).

In a case study of passwords in North Macedonia, where a dataset of passwords from recent high school graduates was analysed, it was found that the passwords contained therein were found to be weaker than the baseline, already weak datasets they were compared against [196]. The authors stipulate that this is a result of a direct link between password security habits and general literacy.

3.8.4 Password Tendencies of Users

Usually, users create passwords that contain familiar models, including the expression of feelings, names, dates, and places. This was demonstrated by Veras et al. [197] where their semantic approach significantly improved the number of recovered passwords compared to state-of-the-art approaches. Veras et al. [198] focused on the semantic meaning of numbers and especially dates in passwords, finding that 4.5% of all passwords in the RockYou dataset were dates. In 2006, Kuo et al. [24]

Table 3.2: Top 10 digits found in RockYou [12]

Digit	Percentage	Digit	Percentage
1	10.98%	123456	1.74%
2	2.79%	12	1.49%
123	2.29%	7	1.20%
4	2.1%	13	1.07%
3	2.02%	5	1.04%

Table 3.3: Top 10 single special characters found in RockYou [12]. To compute the percentages on this table, only the passwords containing at least one special character were considered

Special Character	Percentage	Special Character	Percentage
.	17.81%	@	7.19%
_	14.72%	*	6.54%
!	11.34%	#	3.92%
-	19.25%	/	3.01%
<space>	8.72%	&	1.84%

created a survey and asked users to input either regular passwords or mnemonic passwords that were constructed by phrases and sentences. The authors found that the majority of the mnemonic passwords contained external information, while only 13% of the participants in the control group did the same.

Weir et al. [12] analysed the passwords of the RockYou dataset. Some interesting statistics include the Top 10 numbers found in the dataset contain either single digits or number sequences, as can be seen in Table 3.2. What is interesting about number characters in the RockYou dataset is that the number was found after the letter fragment in 64.28% of the cases and before in only 5.25%. This suggests that users prefer to add numbers at the end of their password, perhaps to comply with the password policy after choosing a solely alphabetical password. The Top 10

Table 3.4: Character sets in RockYou passwords according to password length [12]

Character Set	7+ Chars	8+ Chars	9+ Chars	10+ Chars
Contains Digits	57.5%	59.5%	60.2%	60.0%
Contains Special Characters	4.4%	5.1%	6.6%	8.0%
Contains Uppercase	6.5%	6.7%	6.9%	7.1%
Contains Only Lowercase and Digits	89.2%	88.4%	86.7%	85.1%

single special characters found in RockYou passwords that contained at least one special character can be seen in Table 3.3 and finally a distribution of how character sets appear in passwords of different lengths can be seen in Table 3.4.

In the case of Chinese users, and as far as contextual information is considered, Zeng et al. [199] performed a sentiment analysis on three different datasets and found that sentiments (and in their majority, positive ones) were chosen more often than other contextual information such as places and names.

Passwords based on meaningful common words, personal information, and patterns are considered as more memorable [200]. Also, culture and country of origin seems to play an important role in password selection [201].

3.8.5 Purpose of the Password

It seems that users are willing to accept more difficult authentication methods in the case of financial and e-mail accounts, but not for infrequently used web accounts [202]. They are also more likely to accept more strict password policies on a PC, than a smartphone or tablet and choose safer passwords [203]. It was also shown that in many cases, users include domain information in their password, i.e., the name of the domain the password is for [204]. Finally, a study that compared a Dynamic Personalised Password Policy (DPPP) that takes into account a user's personality traits when prompting a user to form a secure password, with commonly

used password policies, showed that the first resulted in passwords that are more resistant to guessing attacks [205].

3.8.6 Password Managers

In a study conducted with students and personnel from George Washington University it was found that while password reuse was a serious issue, with 77% of participants admitting to it, the number of participants that used an external password manager (instead of a browser built-in one) were found to reuse passwords less that their counterparts [206]. It was also shown that people with external password managers were more likely to use the random passwords created by the password manager, instead of creating their own, as was the case with those that used built-in password managers. In the same study, it was found that the most chosen reason for someone to use a password manager was not the perceived added security it would provide, but the ease-of-use.

In terms of academic research, when these datasets were stored in plaintext, 100% of the credentials in them were stolen by the attackers, which can provide useful insights into password choices of users, but it's extremely dangerous otherwise.

3.9 Measuring Password Strength

Password strength meter is another field of study that is continuously evolving following the sophistication of password cracking attacks. The most basic strength meter is a simple 0/1 metric where basic rules must be respected by the password to accept it, and reject it otherwise, like the LUDS-8 from the NIST proposal back in 2004 [207].

3.9.1 Password Guidelines/Policies

The password policy in place plays a role at the strength and guessability of a password. It has been shown that putting a password policy in place forces users to have more secure passwords, whereas users left to their own devices will generally choose weaker passwords [193]. But the stronger passwords required by password policies may lead users to either having trouble remembering and ending up writing down their passwords [24] or to use common techniques for bypassing the requirements without building a strong password [208]. For example, `Password1!` fulfils the requirement set by one of the most well known password policies, LUDS-8, for uppercase, lowercase, symbols and numbers and is above eight characters, but would not be considered a strong password.

3.9.2 Users' Attitude About Password Strength Meters and Policies

Stringent meters force users to spend longer time creating and changing their password until they satisfy the requirements, but they also found the password meter annoying and in some cases did not pay attention to satisfying the meter [209]. On top of this, this procedure causes great difficulties for users in creating and remembering their passwords [24]. Weak passwords can be remembered, but strong passwords are more likely to be written down [210, 211].

There is therefore an inherent weakness in knowledge-based authentication methods. In a study by Brown et al. [212], 15% of all passwords for email access were assigned to the users, and they had not generated them themselves. Finally, Komanduri et al. [213] concluded that increases in entropy of passwords often correlate with decreases in usability, suggesting a trade-off between these two aspects.

3.9.3 Commercial Strength Meters

Enforcing the selection of strong passwords can help to protect digital systems from password cracking attacks. Password strength meters fulfil 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 [98]. 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.

IT designers have created many password meters [102] and many of them can be found on the Internet as free tools to check a given password's strength, such as Passwordmeter [214], My1login [215] and LastPass [216], with Kaspersky [217] pointing out to never enter your real password.

Concerning the password strength meters which are included in certain web pages, they are unable to assess precisely the number of guesses one needs to retrieve a password [218], as this would demand a lot of resources and time. Yang et al. [219], pointed out that commercial meters need to be improved due to the inconsistent and inaccurate feedback they provide compared to other meters. Entropy, which is traditionally used for measuring the strength of a password, is proving inadequate when intelligence-based attacks are concerned [220]. In the case of graphical passwords, Heidt and Aviv [221] pointed out that most strength meters incorrectly assume a linear relationship between pattern features and puts forth a new meter that takes into account the guessability of the pattern.

3.9.4 Advances in Measuring Strength in Passwords

The community in this field remains active and new password strength meters have been recently designed, each of them following a different approach. Galbally et al. [218] used a very large publicly available dataset of passwords to propose a flexible probabilistic framework, that can be adjusted to different environments or password policies and able to objectively measure the strength of a given password. A multi-modal strength metric was proposed by Galbally et al. [222] based on the implementation of two new probabilistic Markov chain methods merged with an attack-based module and a heuristic-based module.

Guo and Zhang [223], proposed a Lightweight Password-Strength Estimation Method (LPSE), which performed better than other existing LPSEs, in terms of response and storage space, providing at the same time an excellent identification of the strength of the password.

The complexity of the subject led Kelley et al. [224] to propose their technique for evaluating password strength against a variety of password-guessing algorithms. Their algorithm can be trained to increase awareness of password strength.

One of the most widely accepted password meters is `zxcvbn`, which is used by Dropbox and probably in many others, as it is an open-source solution [225]. The meter ranks the password between five classes, from 0 to 4, taking into consideration many criteria, with class 0 containing the passwords that are easier to crack and class 4 those containing the strongest, harder to crack passwords.

Some meters rely on cracking techniques to assess the probability the password would be produced by such techniques, such as the OMEN-based solution [226] or PCFG-based ones [227, 228]. Based on the latter, the fuzzyPSM, Dong et al. proposed the RLS-PSM, where RLS stands for reuse, leet and separation, taking these common altering techniques of users into account and producing, according to the authors, better results. Some meters use ML techniques such as the Neu-

ral Network-based meter of Melicher et al. [154] extended by Ur et al. [230], the Natural Language Processing model by Guo and Zhang [231] and Deep Learning model by Pasquini et al. [232]. In a comparative analysis by Golla and Dürmuth [233], it was shown that when it comes to password strength metrics proposed by academics, fuzzyPSM [228], RNN Target [154] and Markov (Multi) [234] produced the best results.

While increasing the security of digital services by enforcing users to select strong and safe passwords, those meters also play a role in the analysis of password datasets to better understand human tendencies but also classifying passwords in classes of strength. The use of password strength meters can have the desired effect of users choosing more difficult passwords to fulfil the meter's requirements, but subsequently they might need to resort to writing the password down because they cannot remember them [209]. Furthermore, it has also been shown, that the results found by the various password strength meters when evaluating the same passwords have been widely inconsistent [235].

3.9.5 Password Strengthening Techniques

Various techniques regarding the creation of passwords with strengthening in mind have been proposed. The simplest ways usually proposed by IT administrators are inelastic rules for the length of the password and the type of the characters to be used, as well as a specific tolerance to the number of times credentials can be inputted incorrectly before the system locks the user out.

More sophisticated methods, such as the creation of mnemonic phrase-based passwords is another proposed way, where users usually take the first letter of each word of a favourable and memorable phrase and create a new password. It was found that the majority of users based these mnemonic passwords on phrases that can be found on the Internet, which could create problems concerning the strength

of the produced password and especially if such a mnemonic dictionary is included in password cracking tools [24].

Another alternative possibility is the use of graphical passwords [236, 237]. It is easier for users to remember pictures than complex text passwords. Graphical passwords can be utilised as a second step of verification, after the text password, in order to strengthen the verification process. It was found that users are more likely to remember graphical passwords and for longer [238].

Similar is the use of a token, but it is considered inconvenient and costly [239]. The combination with biometrics is another aspect. It is more suited for getting access to local machines and requires a high cost to implement in other activities. Furthermore, it should be noted that the use of password as a back-up or recovery option will not easily be diminished [240]. Finally, it was found that password security training can bridge the gap between the Information Technology (IT) administrators and the end users [241].

3.10 Discussion of Related Work

As the related work on the field shows, passwords are a topic of research that has occupied scientists for decades and will continue to do so, as it looks to remain the prevalent method of authentication. Different types of passwords have been introduced, like graphical passwords and other authentication keys like tokens and biometrics. Even so, the textual passwords remain the key method of authentication, due to its ease of setup and use by the majority of users.

There are various types of attacks to gain access to systems that are password protected or encrypted devices, but in all those cases, the setup of the system directs the type of attack that could be successful. For example, in an online system with a limited number of attempts where there is no inherent system vulnerability to

be exploited, a brute force attack will most likely come up short.

Dictionary attacks, in many cases, are the best of both worlds. Computationally, they are easier to carry through compared to a brute force attack, as the number of candidate passwords will be smaller. But their true advantage is that they take into account the tendency of people to resort to the familiar and memorable, i.e., short passwords that are meaningful to them or easy to remember.

For the dictionary attack, which started quite literally as an attack with English language dictionaries, nowadays, the dictionary consists of password lists from previous data breaches, aiming to exploit password reuse and the principles behind users' password selection. One gap in the literature, is in the creation of dictionary lists that take into account not only the contextual information of previous passwords of unrelated data leaks, but also the context for the which the current password is used for.

This means the creation of dictionary lists that are tailored to the purpose of the password they are trying to guess or the system to which they are trying to gain access. Considering the context of the destination can impact and increase the likelihood of success for a password cracking operation. A detailed look at the methodology for such an operation is described in Chapter 4.

Chapter 4

Methodology

4.1 Introduction

Access to the content of the encrypted devices of a suspect can be crucial for the outcome of an investigation [15]. The timeliness of accessing potentially case-progressing information can be paramount in certain scenarios, e.g., kidnapping cases or an imminent terrorist attack investigation.

There are various password cracking algorithms and tools available to investigators with different approaches to password cracking. In real-world scenarios, there are situations where the approach that has the highest chance of success might not be viable due to time constraints. It is up to the investigator to decide the balance between likelihood of success and time elapsed. For this reason, 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.

As the related work presented in Section 3.8 shows, users are creatures of habit, and they tend to navigate towards what is familiar - and therefore memorable - when it comes to selecting their passwords. This can be proven to be an advantage to an investigator who requires access to a password protected device or system. Three

scenarios where the targeted approach that is described in this work will be presented in the following section, that will be referenced throughout this thesis.

The rest of the chapter is organised as follows: Section 4.3 describes the various datasets that have been used throughout this thesis, containing datasets for evaluation as well as datasets that serve as baselines to compare the produced results against. Section 4.4 describes the analysis of a large corpus of real world passwords in order to identify the role of context in password creation, by breaking down the passwords into their constituent fragments and classifying them according to their semantic meaning. Section 4.6 provides the methodology for examining the hypothesis that password selection is to some extent connected to the website/service the password is aimed for, while Section 4.7 looks at the different measures to be taken into account when deciding the quality of a wordlist for password cracking. Section 4.8 provides the framework for evaluating generated wordlists, and Section 4.9 includes the methodology for creating custom dictionary lists for any topic. Finally, Section 4.10 looks at techniques for optimising the generated dictionary lists and ranking password candidates according to their contextual proximity to the theme of the dictionary list or target.

4.2 Three Scenarios for Contextual Password Cracking

In their majority, administrators nowadays take great steps and employ sophisticated measures to ensure the safe transmission and storage of confidential information such as login credentials. This is even more apparent with services in the financial sector such as e-banking and online trading websites, where strict password policies are enforced, and sensitive data is stored in hashed form.

When it comes to security of online systems, typically, the password remains

the weakest link to gain entry [242]. As has already been discussed in Chapter 3, there are many password creation techniques that many people employ when they create their passwords that have been known (and in cases abused) by adversaries. Many of these techniques rely on personal information being used in the password, something that can be leveraged against the owner of the password.

In this manner, the approach that is examined in this thesis is the creation of password candidates lists (dictionaries) bespoke to each individual or a community of individuals. Often, the information needed for something like this 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, interrogations, 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 significant potential benefit from this in expediting cases. During an investigation, it can be of paramount importance to gain access to encrypted devices, an often insurmountable task given limited resources [35].

The reminder of this section presents three scenarios for which the bespoke dictionary approach would be suitable.

4.2.1 Online Community Scenario

The weakness of the password, when it comes to the security of online services, is further 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 a system, rendering the effort and precautions taken by security concerned system administrators void. In these cases, attackers focus on generic approaches – effectively modelling the popular habits and trends of real-world users' password choices [243]. These attacks customarily use large

dictionaries of human-created passwords available online from previous data leaks/breaches. These attacks have evolved to become more refined and sophisticated to compensate the increase in computational cost of the underlying algorithms and the strengthening of password policies [13].

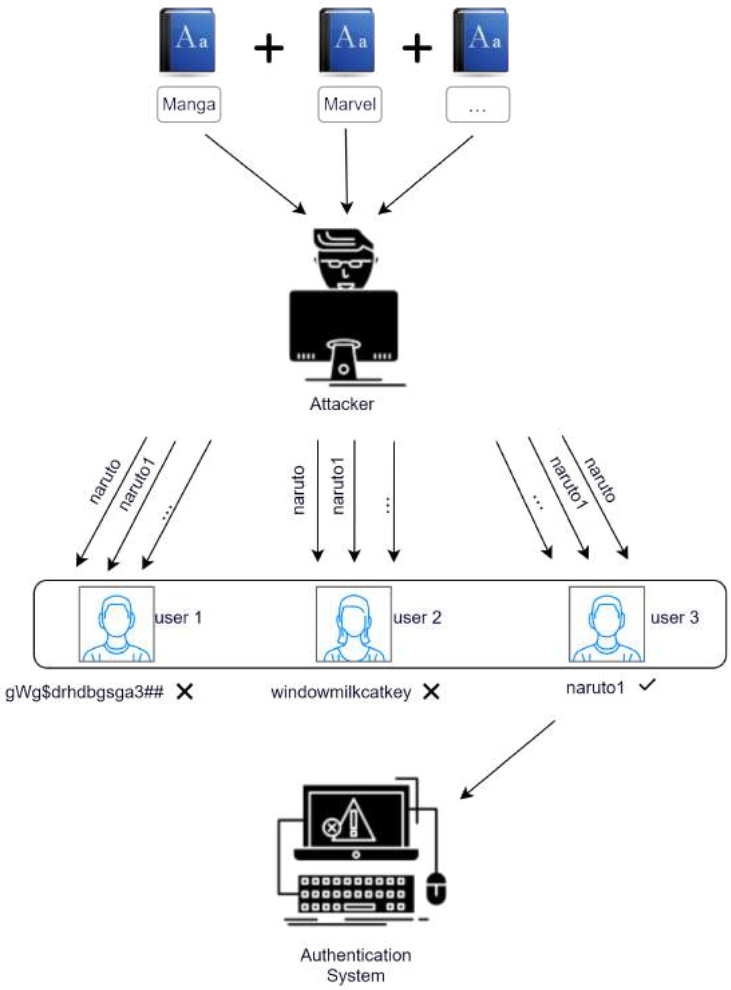


Figure 4.1: Online community scenario

As part of this thesis, the role of context in an attack like this was evaluated. In the online community scenario, the goal is to gain access into a password protected community of users that is centred around one topic. One such case would be

if LEA have knowledge that in a closed community about manga, there is covert illegal activity, and they want to gain access to that community. In such scenario, the successful guess of any user's password is a success. This is obviously simpler than having only one target and limited attempts. When many targets are involved, there is a higher chance that at least one of them has a less secure password.

Context could still be considered in such scenario for an online attack, with a limited number of tries across all accounts. If this is an online community or forum about manga, a logical assumption - and one that will be put to the test in this thesis - is that there is a higher chance of encountering passwords that are thematically closer to manga in this community than in a community that is not related to this topic. The online community scenario can be found in Figure 4.1. As can be seen there, contextual dictionaries that are thematically close to the topic of the community can be employed in order to test contextually relevant passwords against all users of the forum.

4.2.2 Offline Dictionary Attack

Of course, there are also targeted attacks that focus on one specific user. These can be both online attacks, with a limited number of guesses, or offline attacks with an unlimited number of guesses. This is for example the case when law enforcers are attempting to retrieve evidence from a suspect's online/offline account or whenever encrypted devices/container are encountered during digital forensic examination [15]. Generic approaches can be attempted there, as they rely on mimicking user tendencies, or they leverage passwords originating from actual 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 often follow certain habits when creating their passwords. Their use of numbers and symbols is often meaningful, and the placement of capital letters and non-alphabetical

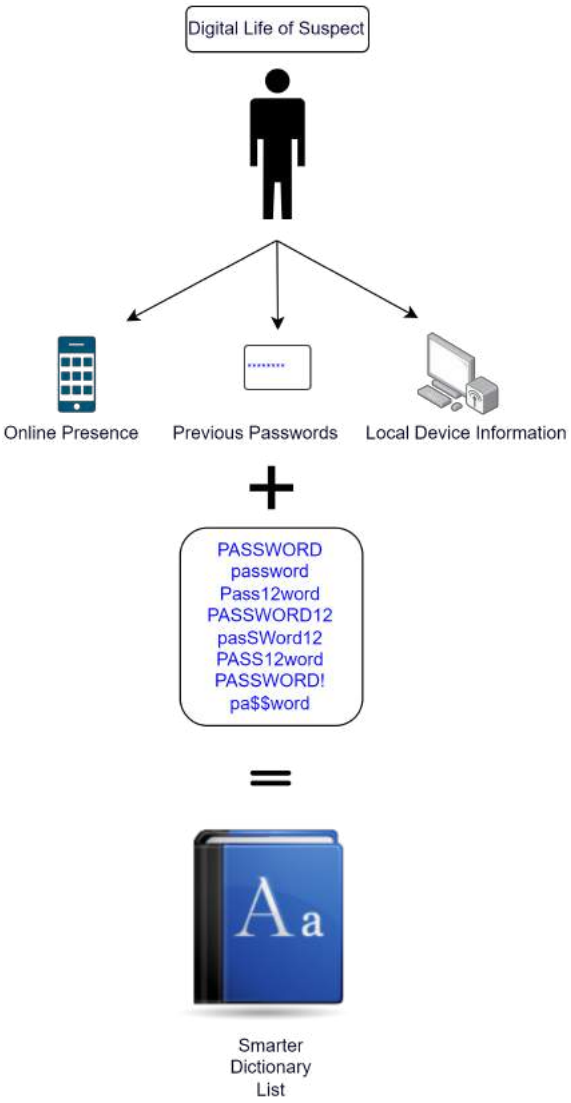


Figure 4.2: Online individual scenario

characters is often predictable, something that will be demonstrated in great detail in the following sections.

Users choose passwords that are memorable or meaningful to them. This is due to the fact that a typical user maintains tens of different passwords for different systems and devices. Since these password habits exist, the knowledge of personal

information about a specific user can lead to more educated guesses of their passwords. This 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 personal information could turn out to be their password, or part thereof.

Figure 4.2 portrays an example of this scenario. If LEA are encountered with an encrypted computer of a suspect that they need to gain access to, to further an investigation, personal information about said suspect can be gathered in a number of ways.

One such way is by looking at their online presence, their social media accounts, what they post about, who they follow and interact with, the type of content they consume. This will help identify areas of interest of theirs, as well as insight into their personal circle of friends and acquaintances. Furthermore, previous passwords of theirs can offer great insight into their thought process behind password selection. Do they tend to capitalise the first word of the password? Do they add numbers at the end? Do they reuse passwords regularly? And finally, what other type of information can be gathered from their other devices or from objects in their room or house.

All this information can be used, along with mangling rules, to create a smarter dictionary that is tailored to that specific suspect. This step is crucial since this dictionary will be generated, so mangling rules can help simulate human behaviour when it comes to password cracking.

4.2.3 Combination Approach

In any digital investigation, this bespoke dictionary generation step could be one of the first after collecting evidence on the individual related 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 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 using the approach outlined in this work, 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 carried out, evidence and information that can help launch the bespoke context-based dictionary attack can be collected.

This begs the question of where exactly in the password cracking pipeline the proposed approach might fit. The answer is that there is no one-size-fits-all solution to this question. If time is of the essence, and it is known that the suspect is someone technologically and security savvy, then a reasonable assumption can be made that an exhaustive search of up to 8 characters is not likely to produce results; therefore, this choice may be skipped or postponed. If this is the case, but the process of collecting evidence to launch the targeted dictionary attack is still ongoing, another dictionary attack might take precedent. This case-by-case scenario is illustrated in Figure 4.3.

Ideally, when talking about context-based decryption in a digital forensic setting, experimentation would be conducted on real cases by a digital investigator focusing on one specific target. But as this constitutes privileged/sensitive information, it is not possible to do this in a research context. Therefore, the focus is shifted in this work to “the community approach”, which can be evaluated using leaked lists of real-world, human-chosen passwords stemming from data breaches.

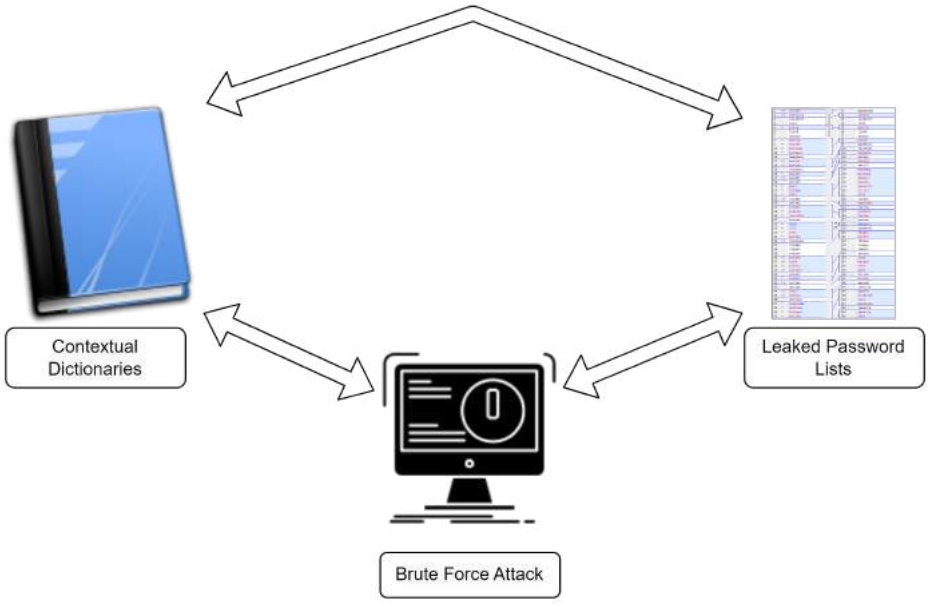


Figure 4.3: Combination approach scenario

4.3 Dataset Sources

In this section, the datasets that have been used throughout this thesis, either for evaluation purposes or to act as baselines for comparison of the newly created contextual dictionaries, are presented. More details about their use, the clean-up process for them, if any, or any modifications to them will be discussed in the appropriate sections.

4.3.1 Have I Been Pwned

A very well known assembled list of passwords found in data breaches is Have I Been Pwned (HIBP) [244]. The original website for *Have I Been Pwned* (HIBP), was created by Troy Hunt, a web security expert, in 2013, to help users detect if their email address(es) appear in data breaches. Its main purpose is to help victims being aware that their accounts have been compromised, but it also serves as a

blacklist for passwords and to highlight the seriousness of data breaches. In 2017, Troy Hunt launched an API to check whether a given password appeared in a previously leaked database. In 2021, after 500 million of phone numbers were leaked in the Facebook data breach, phone numbers are also searchable'[245]. The current version is version 8, and it contains about 850 unique passwords (corresponding to around 11 billion accounts, on the accounts that duplicate passwords have been removed). Figure 4.4 shows the amount of new leaked passwords from data breaches that have been added over the years, and the increase in the last five years is staggering.

The objective behind this tool is to reduce the password reuse phenomenon and prevent credential stuffing attacks [246] by implementing a searchable password blacklist, as strongly encouraged by the latest NIST directive [101]. It also serves to allow individuals to check if their passwords or other credentials (phone number, username) have been compromised, or*pwned*. All the passwords from various breaches have been concatenated in a single dataset and made publicly available for companies, governmental services and institutions to implement their own black listing of passwords independently. In addition, it has been utilised for academic purposes, e.g., validating passwords created from song lyrics through the haveibeenpwned Application Programming Interface (API) [247] and measuring the frequency of compromised passwords in an Asian Pacific college [248].

4.3.2 Hashes.org

As mentioned above, HIBP is a dataset that consists of many different data breaches concatenated together. For the scenarios outlined at the beginning of this chapter, an evaluation against datasets stemming from single sources is needed. For this reason, the datasets from hashes.org have been utilised. Hashes.org was a website running from 2011 to 2020, with the goal of being a single point of reference for

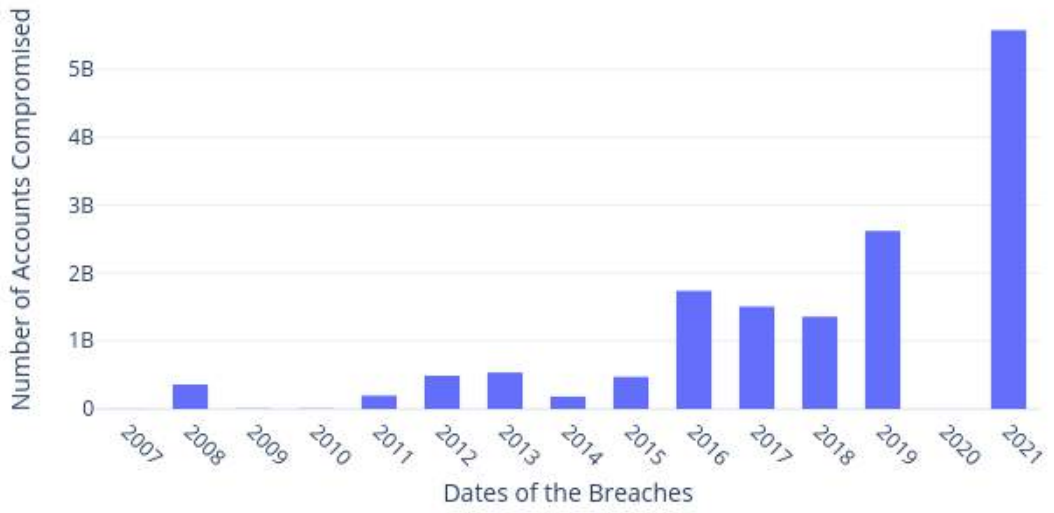


Figure 4.4: Number of breached accounts listed in Have I Been Pwned

hashes and their solutions, where multiple users could contribute towards cracking the hashes from various data leaks. The database contained a variety of hashed password lists, from various communities and forums, as well as the cracked plaintext passwords and metadata about the origin of the dataset and the percentage of cracked passwords. The website was taken down in 2020, but the database was obtained before that.

4.3.3 RockYou

Most dictionary attacks use wordlists that originate from one or more different data breaches. The passwords in these data breaches are sometimes leaked in their plaintext form, but more often they are hashed (and salted). This means that in order to make use of these lists in dictionary and/or other password cracking attacks, the passwords need to be cracked first. That is not always possible for 100% of the

leaked data – meaning that it can be argued that some of these cracked lists do not contain the “harder to crack” passwords.

One list that does not fall in this category is called “RockYou” and contains 32 million passwords that were leaked in 2009. Because RockYou was leaked in plain text, it is a very popular wordlist that has been used by many researchers as a way to extract insights on users’ password habits [249, 250] or as a baseline to compare other attacks against.

4.3.4 Ignis

While the plain text passwords offer a significant advantage compared to an incomplete list from hashed leaks, one drawback of RockYou is that it was leaked in 2009. Since then, password policies around the globe have changed and stricter measures have been adopted – with passwords often needing to be longer and containing more than one of uppercase, lowercase, number, and symbol characters. Therefore, in terms of adopting a baseline to compare the contextual dictionary approach against another dictionary that has been used at the later stages of this thesis is Ignis-10M [251].

Ignis contains 10 million passwords from a variety of data leaks and was assembled in 2020. A statistical analysis comparing its makeup against RockYou can be found on the project’s GitHub. Besides Ignis-10M, smaller versions of the Ignis wordlists are also included in the GitHub page, but the larger list always achieves a higher success rate in all experiments and was therefore the one used.

4.4 Analysis of Real World Passwords

As a first step to tackling the issue of context in passwords, whether it be for the community or individual approach, it is paramount to gather every bit of informa-

tion regarding password selection from users. This will provide insight into their choices when it comes to passwords, and some very useful statistics and patterns will emerge. As discussed in Chapter 3, there is work that has been done in this area, albeit these studies focused on smaller individual datasets with specific characteristics. Some notable examples that have been already mentioned in Chapter 3 are the work of Mazurek et al., measuring password guessability for an entire university and the work of Wang et al., which highlighted specific habits of Chinese password users. In 2010, the largest real-world password component and pattern analysis performed was on 32 million accounts from the RockYou data breach [12]. This analysis number increased to 70 million from a Yahoo data breach in 2012 [195]. No more comprehensive password pattern and component analysis has been performed since, nor has any such analysis been performed on a very large corpus of passwords stemming from several contributing data breaches. This was the motivation behind the work in this section.

4.4.1 Have I Been Pwnd Dataset

The source dataset used for this analysis is the Have I Been Pwned version 5 (HIBP_v5) password dataset. At the time this research had been conducted, five incremental versions of this list had been released since 2017, with each newer version containing more passwords, updated counts of each password's occurrence and the removal of "garbage" passwords, i.e., badly encoded, duplicates, etc. Version 5 which was released in July 2019 is the one that was used for this research. Since then three more versions of the dataset have been released, version 6 in June 2020, upping the number of unique passwords to 573 million, version 7 in November 2020 with a further 40 million unique passwords and finally version 8 in December 2021 with a further 38% new unique passwords, bringing the total to 847 million.

The dataset does not provide any additional information about each password

such as the breach it came from nor the date discovered. However, it can be assumed that the entries of the dataset come from the data leaks listed on the HIBP website. The date spread of the total number of accounts compromised by those data breaches is displayed in Figure 4.4 of Chapter 3.

Focusing on version 5, the total number of accounts compromised in these breaches is over 9.4 billion. However, HIBP_v5 does not contain this number of passwords. This can be attributed to several explanations. It is known that there was no password associated with over 2.8 billion of the breached accounts. Furthermore, as declared by Troy Hunt, the list is composed only of passwords that were initially gathered in plaintext whereas the website can still list the username as breached when the password is not stored in clear. That means that not all passwords for which usernames/emails are listed are in the dataset.

This composition is not without consequence for the results of the analysis presented in this thesis, for the two following reasons. Firstly, the strongest passwords could be missing from the list obtained by Troy Hunt if the original source was not in clear text, as only the passwords that have been previously found are included. This could skew the results of this analysis and wrongly underestimate the number of users that choose secure passwords.

On the other hand, if the passwords were stored in plaintext, then the strongest passwords are contained in the related leak. However, the corresponding service/website was not following basic security recommendations for safely storing passwords and sensitive information, which can easily lead one to believe that little attention has also been given to ensuring that users choose passwords that adhere to password policies and security recommendations. Even so, thanks to the large size of the list, this analysis is relevant for an overwhelming proportion of accounts. It is furthermore particularly challenging to obtain a dataset that contains the whole spectrum of passwords, including strong passwords, to complement the analysis

and bridge this bias.

4.5 Pattern Analysis

The first step of analysing real-world passwords is focusing on the types of patterns that can be discerned from them. Extracting patterns that repeat on many different passwords can provide useful insights on the tendencies of users to prefer one or another way of building a password. The analysis of the patterns found in the passwords can also be the stepping stone to identifying and extracting distinct parts of the password for further analysis.

There are typically four classes of characters considered in the password analysis community; lowercase, uppercase, numbers, and special characters. Strict password policies nowadays recommend (and in many cases require) all four types to be included in a password, however that has not always been the case, with many websites, especially in the past, enforcing no rules regarding the password makeup or length.

In the architecture of a password, not only the combination of these four classes of characters is important, but also the sequence. One of the goals of this password analysis is to see which types of architectures are prevalent among users, if any. Other work on the field, as shown in Chapter 3 showed prevalence of some architectures already, like the addition of numbers at the end of the password [220], but on datasets of smaller size.

4.5.1 Masks

In password cracking, these architectures mentioned above are called masks. Masks can be distinguished into two different categories, those that take into account the order of the different character sets in the password and those that do

not. For example, when the sequence is not taken into account, the passwords `1password` and `password1` would fall under the same mask, one that includes lowercase characters and numbers.

Masks can be even more specific, with specifying exactly the number of characters, numbers and/or symbols. For example, the mask `?l?l?l?l?l?l?l?l` corresponds to all lowercase combinations of length 8. Masks are also used for password cracking, as they allow for creating passwords of only specific architectures, which is useful if there exists some prior knowledge of the encrypted password or a reduction of the search space is important. This is their main advantage over the brute force attack, as the creation of masks that mimic user choices can be tested first.

4.5.2 Base Words

As the successful use of masks in password cracking and the various password policies dictates, passwords are in many cases a combination of different character types, alphabet letters, numbers and special symbols. Being able to extract these popular combinations from leaked lists of passwords plays an important role in creating better masks and rules for password cracking. But when it comes to passwords that are made up from dictionary words, it is equally important to also look at which dictionary words are most popular. For example, `password1` which is one of the most popular passwords on the NordPass list of the most popular passwords of 2022 that is mentioned in Chapter 2 would be classified as a `stringdigit` mask and while it is important to know that it belongs in the category, it is equally important to extract what is called the base word, which in this case is `password`. Extracting the base word of each password (by essentially removing the non-alpha characters at the beginning and end of the password), allows for a compilation of the most popular dictionary words that are used in passwords.

4.5.3 Fragmentation

As defined in the previous section, a base word removes any non-alpha character at the beginning and end of the password. Frequently, this is not enough for more complicated passwords. For example, if a password is made up of more than one alpha fragments, it is important to consider the type of fragmentation that will yield the most interesting results when analysing real-world passwords. As was stated at the Introduction of this Thesis, one of the aims is to identify if and how meaningful contextual information is in a password and whether it can help optimise password cracking with dictionary lists.

For example, a password like `alexander1998` indicates that a male first name is chosen along with a year, which could be a birth year. If this type of configuration is common, it could indicate that people often choose first names and years (maybe even their own name and date of birth or that of a relative). In the scenario of the individual suspect, the name and date of birth of the suspect or their family members could be some of the first passwords an investigator should check to bypass the authorisation system. On the community scenario, this information is also useful; common first names and birth years can be concatenated and checked in the hope that one of the users of the community has this configuration in their password.

Classifying the fragments according to their semantic context is useful to see not only which topics are more commonly chosen in passwords, but also which of them commonly go together and generate password candidates according to that knowledge.

4.5.4 Strength Analysis

One of the most useful characteristics about passwords is their strength. Users are probably not always concerned in having strong and safe passwords, or simply not

aware of the consequences of having a weak password. This hypothesis is supported by the massive use, and re-use, of weak passwords. However, the strength of the password becomes crucial when it is about protecting critical service, e.g., bank accounts or the security of a large infrastructure. To this matter, password metrics are often put in place to ensure a minimum strength of the password. The most spread one is the one proposed in 2012, and updated in 2017 [101], by NIST recommending a minimum of 8 characters including lowercase, uppercase, special and digit.

However, this approach has shown its limits with time and attackers have adapted their attacks to mimic the typical patterns followed by humans in general. A plethora of other metrics have emerged, each of them being based on different heuristics and methods to assess the strength of passwords. Galbally et al. [253] and Golla and Dürmuth [254] proposed a comparison of those metrics. While the method proposed by Galbally et al. [253] is interesting because it provides different evaluation criteria for each password and therefore better understanding of why a password is strong or weak, the proposed implementation is not fast enough to analyse more than 500 million passwords in a timely manner. The best password metric according to Golla and Dürmuth [254] is based on the HIBP API, and therefore it does not seem at all suitable to us to assess a dataset using an approach based exactly on such dataset.

The common point in these two articles is that the `zxcvbn` password strength metric, originally deployed in the Dropbox service, provide good results. This is the metric that will be used on for this analysis, and further details about how it works are presented in Section 5.4.3 of Chapter 5.

4.5.5 Hardware Consideration

It is essential to include an evaluation on the hardware needed in digital forensic laboratories to make password cracking viable. As previously mentioned, passwords are predominantly stored in a hashed/salted hash format. The hash function employed is therefore a security parameter in case of a data breach. Indeed, if the hash function is quick to evaluate, an attacker will have the capacity to evaluate more candidates than if the function is slow. The MD5 hash function has been widely used to store passwords and even though it is deprecated, it is still commonly used in online services. A single gaming graphics card, an Nvidia 2080 Ti, is able to evaluate 50×10^9 password candidates per second. In order to better visualize these figures, a single 2080 Ti can fully evaluate all possible MD5 passwords up to length 8 considering an alphabet of 95 characters (26 lowercase, 26 uppercase, 10 digits, and 33 special characters) in less than 2 days. Considering the BCRYPT hash function, specifically designed to be slow on graphic cards, only up to five characters can be brute forced in practical time, as the card can evaluate approximately 25,000 passwords per second. Each increment of the length of the targeted password multiply the time of the attack by a factor of approximately 100.

4.5.6 Steps of the Analysis of the HIBP Dataset

Two cases were possible for the analysis of the HIBP dataset: either analysing the unique passwords, or analysing the passwords considering the number of occurrences in the dataset. The latter option better maps the human behaviour and therefore the result of the analysis that are presented in Chapter 6 rely on the 3.9 billion non-unique passwords of HIBP_v5.

The steps of the analysis can be summed up as follows:

- The extraction of statistics about the HIBP dataset

- A pattern and Mask Analysis of the HIBP passwords
- A look at the constituent fragments of HIBP passwords
- The classification of fragments and passwords according to context
- An analysis of the guessability of passwords in HIBP
- A brief contextual analysis of a single dataset (from a single community)

4.6 Contextual Information in Leaked Password Lists

Looking back at RQ1, `What impact does a context-based password cracking approach have on the likelihood of success during a digital investigation?`, it is also important to extract contextual information on a bigger level than single passwords in a diverse dataset such as HIBP. HIBP is composed of various data leaks from widely different sources and as research has shown, the way users approach password creation differs according to the purpose of the service. While it is useful to see that, for example, football team names make popular passwords, it does not mean that they would make popular passwords in a forum about cars.

Therefore, it is important to try to extract contextual information on a different level, where the focus is solely on one dataset from a specific data leak, as per the community scenario. This would answer the following question, does the purpose of the website/service the password is aimed for, have any impact in the password itself?

To this end, an experiment is set up where datasets stemming from data breaches of specific communities will be evaluated with datasets stemming from similar communities as well as against baseline datasets, to see if there is an increase in found passwords that are thematically close. For example, if in the commu-

Table 4.1: Contextual datasets in use from leaked password lists

	Dataset	Size
Comb4	AxeMusic	252,752
	JeepForum	239.347
	Minecraft	143,248
	MangaTraders	618,237
Evaluation	Boostbot	143,578
	MangaFox	437,531
	RockYou	14,344,391

nity scenario, access to a website about manga is required, would having a leaked list of passwords from another manga community produce password candidates that are successful when a more generic list like RockYou might not?

4.6.1 Setup and Datasets Used

The datasets that were used to conduct experiments to gauge the role of context in password selection for specific forums are highlighted in Table 4.1. All the datasets used in this experiment are from *hashes.org*. For the evaluation, two datasets from two different online communities are used, `mangafox` and `boostbot` which are from communities about manga and video games respectively. Additionally, RockYou is used as a baseline to compare the other datasets against. There are two publicly available versions of RockYou. The first consists of 32 million passwords with repeated password entries (providing insight to the most frequently used passwords). The version of RockYou that was used is the one with 14 million unique passwords, but the full version of 32 million passwords was also tested, but yielded similar results.

Comb4

The dataset named Comb4 is a combination of four different leaked datasets from four categories, music, cars, video games and manga. The aim was to create a combination of datasets from different sources to cover a wide spectrum of user interests and ascertain whether the purpose of the forum for which the password is created for, plays a role during the creation process. The assumption to be tested is whether a dataset of real-world passwords from a manga community would be able to crack more (or harder) passwords than a generic dataset as RockYou.

Information about the tools used for this experiment can be found in Chapter 5 and the results of this experiment are described in Section 6.3.

4.7 How Can Password Candidate Dictionary Quality Be Measured?

Before delving into the creation of bespoke contextual dictionaries, it is important to well define how to measure the success of these dictionaries. The definition of a metric to measure and classify the quality of a wordlist given as input to a password cracking process is a difficult one. The expected features a wordlist should have and be evaluated on, are likely to vary depending on the final cracking process and its context. The particular scenario of the attack, such as whether it is a targeted attack to a specific individual or a fishing attack that targets a group of people, plays a role in the approach taken for creating a wordlist.

Other factors, such as the language of the target(s), the type of service, etc, also have to be taken into account, since the approach will be different [255].Therefore, using a single metric for measuring the quality of dictionaries is not suggested. Instead, the focus shifts to a number of factors that can be taken into account, alone or combined, when deciding the type and makeup of a wordlist that is likely to make

it the optimal password candidate list for a specific scenario. These factors are presented below.

4.7.1 Final Percentage of Passwords Cracked

This is the most straight-forward metric in password cracking, where a wordlist is evaluated based on the amount of passwords it has cracked from a target list. This metric is typically the most important one – especially in the case where the concern is the volume of cracked passwords and the focus is not on a single target or a relatively small number of targets.

Some password cracking processes have a fixed limit of candidates they can generate based on the size of the input wordlist. For example, a straight dictionary attack will generate as many candidates as there are in the wordlist, potentially multiplied by the number of mangling rules, if they are used. Some other processes can be considered as endless, such as for example Markov-based ones if they are not limited, and will continuously produce candidates like an unbounded exhaustive search would do. As a consequence, those endless processes would theoretically always retrieve 100% of the passwords if they are given enough time, which in most cases is not practical. That is why it is necessary to set a limit to the number of candidates that a process is allowed to generate and test. Such limit can be adjusted depending on the complexity of the scenario for assessment.

4.7.2 Number of Guesses until Target

The previous metric alone is not enough to evaluate a wordlist, as other factors can be relevant in some scenarios. For example, one wordlist may recover 75% of passwords, while a second may get only 60%. But, it might be the case that the second one reached a score of 50% with fewer candidates generated than the first one. In some scenarios, the number of candidates that can be evaluated in a reason-

able time is strongly limited because of hardware constraints or high complexity of the underlying function, making the second wordlist more interesting for a particular scenario. Assessing the number of guesses needed to reach a targeted percentage of retrieved passwords can help to select a wordlist more suited to the conditions of some scenarios.

4.7.3 Progress over Time

Another metric that is strongly related to the amount of found passwords, is the pace in which the passwords are retrieved. During password cracking, an updated percentage of results at pre-established checkpoints, can give us insights into the performance of the dictionary over time. For example, at some point in the cracking progress, the amount of new passwords guessed at every checkpoint might start decreasing, which means that the new password candidates that are checked no longer recover new passwords. This is often another criterion to stop the process and also a hint, for dictionary lists that are ordered by count, that the size of the input wordlist can be decreased without a remarkable effect on performance. This criterion is the second derivative of the curve of found passwords over number of guesses and represents a process with an upper bound, compared to the metric outlined in Section 4.7.1.

4.7.4 Size of Wordlist

Closely related to the stop criterion of incremental progress over time, the size of the wordlist is another metric that can be taken into account. For example, when two wordlists, with a significant difference in size, produce similar numbers of cracked passwords, the smaller wordlist can be thought of as of better quality, as it needs less information to achieve the same results. From another point of view, when the foreseen process is ML-based, a larger wordlist could be preferred to reinforce the

training phase.

4.7.5 Better Performance with Stronger Passwords

Another metric that should be considered is the performance of a wordlist against difficult passwords. For example, if two wordlists are similar in the previous criteria, i.e., crack about the same number of passwords, do it at about the same amount of time and are of similar size, the one that cracks more difficult to recover passwords is stronger, and should be assigned a higher score. Often, in real world scenarios and if the hash function permits it, an exhaustive search is performed first for the weaker passwords. This means that a wordlist that performs well against passwords that cannot be recovered by a brute force attack, is more valuable.

4.7.6 Compound Metric

The above metrics, cannot accurately assess any individual wordlist. Focus on one, or more of the above is necessary, according to the target case. For example, when the goal is to recover as many passwords as possible, the percentage of success is what matters most. But, when the largest number of passwords in a specified amount of time is wanted, the trade-off between success and time is important. When the focus is on a single target, or a few targets, like during the course of an investigation, speed and possibly the performance against stronger passwords are important factors to consider. This criterion can be refined to look at the number of guesses needed to retrieve a given percentage of passwords of a certain strength class.

Furthermore, it has been shown that large corpora of passwords obey Zipf's Law, meaning that the frequency of each popular password, would be inversely proportional to each rank, i.e., the second most popular password would appear approximately half as many times as the first [256]. According to this analysis, the

level of fit of a particular dataset under this model, could be an indicator of strength of the dataset.

This brings forward the need for a compound metric, one that combines two or more of the above criteria, to get an evaluation tailored to a specific case.

4.8 Dictionary Evaluation Methodology

Based on the metrics analysed in Section 4.7, the evaluation of input wordlists, with the purpose of arriving at the optimal one, is a case by case scenario and is based on the individual needs, be it rate of success, time it takes to achieve a certain threshold or success at recovering one specific strong password.

The evaluation of the quality of password cracking dictionaries based on the above metrics can be done with the Password Cracking Wordlist Quality (PCWQ) Framework as shown in Figure 4.5. As the methodology flowchart shows, the performance of the different input dictionaries is evaluated and presented or fed back to pre-processing. The pre-processing step, which will be discussed in greater detail in the following sections, contains the creation of tailored input lists from existing or custom dictionaries and the tailoring of mangling rules to the specific scenario (while keeping in mind whether the end goal is success ratio, time efficiency, recovery of a targeted password, etc). The feedback from the evaluation process will re-trigger the wordlist creation process in order to modify the size of the list, the number and quality of mangling rules and the level of contextual information, with the end goal being to optimise the generation of a password candidate list.

The goal of this framework will be the evaluation of all created password candidate lists under the same scenarios and, by taking into account the metrics discussed in Section 4.7, to arrive at the optimal wordlist for each scenario.

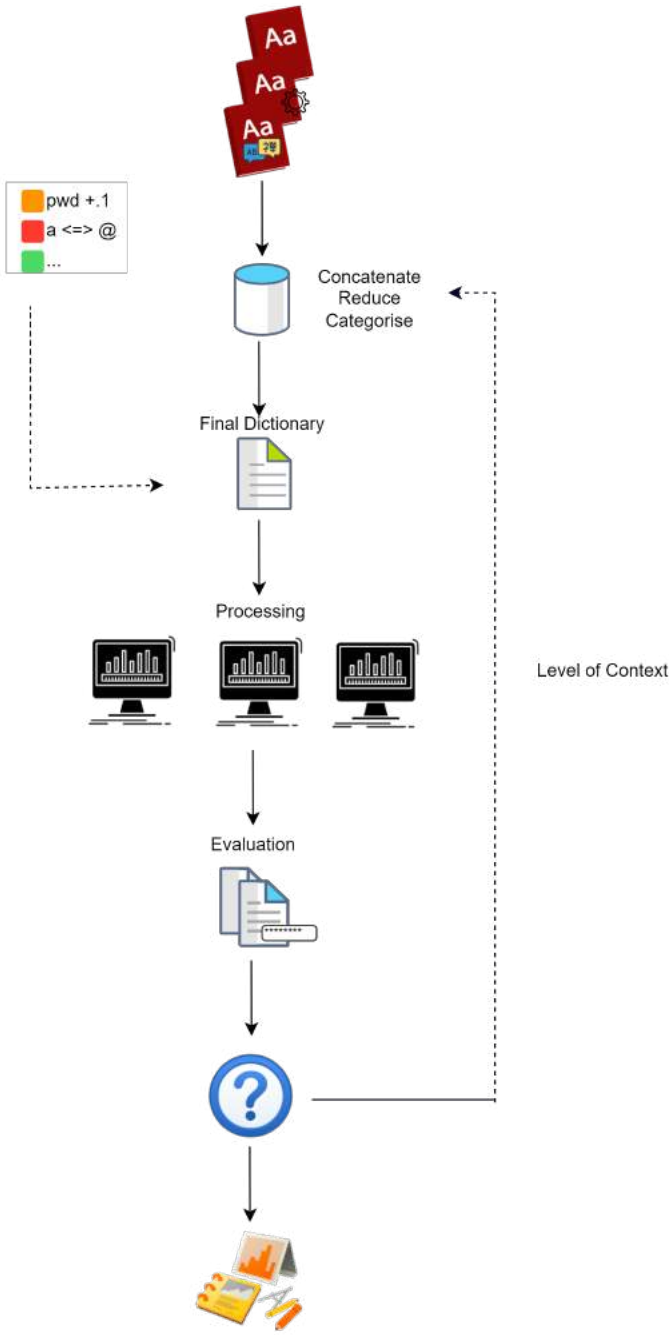


Figure 4.5: Methodology for the evaluation of bespoke, contextual dictionaries

4.9 Dictionary Creation Methodology

As mentioned throughout this thesis, dictionary attacks are an effective way to crack passwords. There are many publicly available dictionary lists that are used for the purpose of password cracking, many of which originating from leaked password lists from data breaches.

To this end, it seems logical that the best way to increase the chances of cracking a password (or cracking as many passwords as possible) from a list of hashed passwords is to create a more robust dictionary list. 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, birthdates, places, and interests, e.g., sports, cars, popular cultural references, etc.

This selection of features is based on the statistical analysis of over 3.9 billion real-world passwords as described in Section 4.4 in which case the passwords of the HIBP dataset are deconstructed into their constituent components and classified according to context. This analysis demonstrated 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 of 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 topics can be used alone or in combination according to a specific target.

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

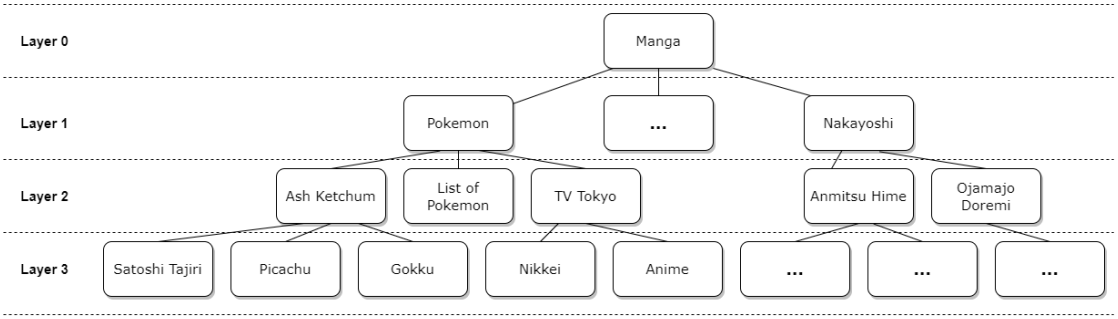


Figure 4.6: A depiction of the tree-like structure of Wikipedia

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.

The approach outlined as part of this thesis for creating dictionaries starts with Wikipedia [257]. The reasoning behind this is that each page on 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 the exploration.

An example of the Wikipedia-driven topic hierarchy is shown in Figure 4.6. Assuming that the seed topic is "Manga", each of the links referenced in Manga's Wikipedia entry leads 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, which is a database version of Wikipedia.

4.9.1 DBPedia

DBPedia [258] is a crowdsourced project aiming to offer a structured manner to access the information found on Wikipedia. The DBPedia information contains the abstract of each article found on each Wikipedia page, as well as the information contained in the article's `infobox`. The infobox contains a summary of the most relevant information related to each article. As infoboxes on 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 [259]. This provides an easy way to leverage the structure and links between Wikipedia pages, providing an interconnecting web of content that is thematically related. The extraction of the needed information from DBPedia is outlined in further detail in Chapter 5.

A comprehensive diagram of the proposed process is shown in Figure 4.7. The next sections outline each of the parameters used as part of the experimentation and describes how/why they were chosen.

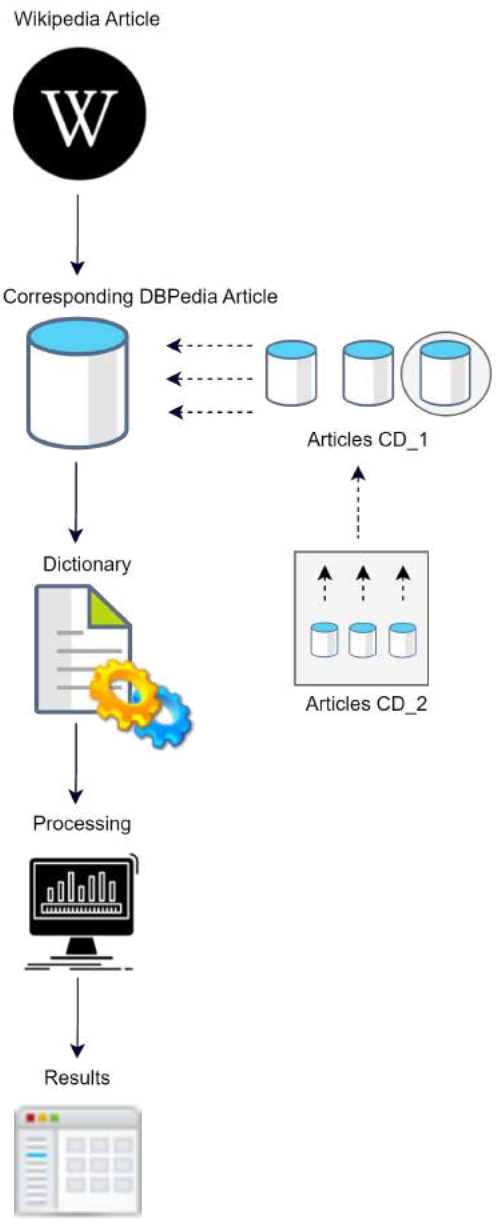


Figure 4.7: Methodology for the creation of bespoke, contextual dictionaries

4.9.2 Selection of Evaluation and Control Datasets

As a baseline to compare this approach against existing ones, the dictionary Ignis-10M (as described in Section 4.3.4) has been selected. The reason Ignis has been selected instead of RockYou 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.

4.9.3 Dataset Selection

In order to prove the importance of contextual information in password cracking using a community of users around a specific interest/topic, ten datasets were chosen from hashes.org from different online community leaks. As can be seen from Table 4.2, the datasets have been picked to represent a variable sample of topics and interests. These include data breaches from forums focused on music, cars, video games, recipes, and shopping. The datasets are also of various length; the smallest being approximately 25,000 and the largest being 23 million – to encompass as big a variance as possible.

These dictionary lists are then used as input with the password cracking tool

Table 4.2: The ten datasets involved in the experiments in password cracking

Dataset	Size
AxeMusic	252,752
JeepForum	239.347
Minecraft	143,248
MangaTraders	618,237
Wattpad	23,531,304
Battlefield	419,940
Wanelo	2,130,060
EverydayRecipes	25,271
Zynga	42,908,386
DoSportsEasy	46,113

of choice in order to crack the passwords of the data sets listed in Table 4.2. The password candidate creator tool that was chosen for this set of experiments was PRINCE, as it had the best performance in the experiments with leaked password lists that were described in the previous section.

4.9.4 Parameter Optimisation

For the purpose of this experiment, ten seed words were chosen in order to correspond thematically to each leaked dataset that is shown in Table 4.2. These seed words can be found in Table 4.3. It should be noted here that the seed word for Battlefield is Battlefield_(video_game_series) in order to represent the video game Wikipedia article, but will be referred to as simply “Battlefield” for the remainder of this work. The seeds words were chosen to be as thematically close to the topic as possible. For example, for the Zynga leak, the world “Zynga” was also chosen as the starting point for creating the dictionary. For Wanelo, a leak from a website about shopping, the word “shopping” was used. As observed in the table, half of the seed words were chosen to be the same word as the target dataset, such as “Minecraft”

Table 4.3: The ten dictionaries produced by DBPedia

Dataset	Seed Word	Size
AxeMusic	Music	1,001,173
JeepForum	Car	853,825
Minecraft	Minecraft	243,803
MangaTraders	Manga	180,641
Wattpad	Fanfiction	641,007
Battlefield	Battlefield	415,311
Wanelo	Shopping	627,487
EverydayRecipes	Cooking	524,269
Zynga	Zynga	443,443
DoSportsEasy	Sports	31,918

and “Battlefield”, while the other five were chosen to be a generic one-word description/category of the purpose of the website, such as “Cooking” for EverydayRecipes and “Sports” for DoSportsEasy. The rationale behind this was to study whether an identical seed word to the target leak over a generalised topic would be significant.

Generated Dictionary Level Depth

The seed words mentioned above were subsequently used with the methodology outlined in Figure 4.7 in order to create custom dictionary lists. One parameter that needs to be defined at this stage, is the depth of these datasets, i.e., how many layers down from the seed word should be explored during dictionary creation. For this purpose, multiple dictionary lists for each seed word were generated; ranging from one layer to three layers, and in some cases four layers. Of course, the time to generate these dictionaries depends on the number of links in each level. The latency of the Internet connection has a significant impact on the speed to gather the pages from the online version of DBPedia. As indicative durations, Layer 1 is almost instantaneous, Layer 2 took $\sim$20s, Layer 3 took $\sim$30m, and Layer 4 took

approximately $\sim$1 day.

The performance of these varying layer depths was assessed for a selection of the aforementioned datasets, and it was found that the dictionary lists produced by only 1 or 2 layers achieved lacklustre performance. For example, in the experiment using the Wattpad leak, the custom 2 layer dictionary cracked 1.6% of the total passwords, while the 3 layer dictionary managed to crack 42.1%.

A 4 layer dictionary was produced with the “Manga” seed word. This was used with the leak from Mangatraders, and it was still found that the 3 layer dictionary performed better than the 4 layer one. More specifically, the 3 layer dictionary found 57.2% of the passwords, while the 4 layer one found 34.4%. This is due to smaller dictionaries facilitating more mangling for a fixed number of guesses than a larger one. Therefore, selecting a depth of 3 layers is the optimal choice. When keeping the number of guesses constant across the experimentation, it is important for the list to be long and detailed enough, but not too long as to include words that are too thematically distant from the seed word.

Finally, as can be seen in Table 4.3, even though each of these datasets are of depth 3, their size varies according to how many links are contained in each Wikipedia/DBPedia page visited.

Password Mangling Rules

As mentioned in Section 3, password mangling rules are set during password cracking processes in order to imitate real users’ password habits. For example, adding numbers or symbols at the end of a chosen password when the corresponding password policy requires them. These are generally useful and should be tailored according to the target. For the experiments outlined as part of this work, the default mangling rules of John the Ripper were used on both the contextual dictionaries and the baseline dictionary.

Number of Guessing Attempts

When it comes to password cracking, the time taken to explore the password search space defined is directly related to the number of attempts permitted during the cracking phase's execution. Despite the brute-force cracking mantra of every password being crackable given enough time, this is realistically impractical in real-world scenarios. With a reduced search space and using a non-brute-force technique, more attempts will crack more passwords and/or have a higher likelihood of cracking a specific password – but at the expense of time and resources. As a result, password cracking typically requires a reasonable limit for the number of attempts to be decided upon.

In order to decide on the number of attempts to limit each experiment presented as part of this work, a number of options were evaluated. To overcome the difference in dictionary sizes generated for a specific topic and/or generated dictionary level, a fixed size of guessing attempts was selected after experimentation and this was 10 billion. A lower number of guessing attempts produced worse results for both the baseline dictionary and the contextual dictionaries. On the other hand, more guessing attempts did result in more cracked passwords, but the trade-off between the additionally found passwords and the running time of the cracking process was deemed inefficient for the purposes of this work.

4.10 Methodology for Ranking and Optimising Contextual Dictionaries

For taking dictionary generation to the next level with the aim of getting an even better success ratio, a next step can be the ranking and optimisation of bespoke dictionaries for specific topics.

For this purpose, four data leaks out of the ten of the previous experiment, have

Table 4.4: Size of selected datasets from four communities

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

been selected from four communities, about music, cars, fanfiction and manga. These datasets and their sizes can be found in Table 4.4 and as previously, only 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.

4.10.1 Size of the Created Dictionary

It has been observed that the size of the wordlist in a dictionary attack plays an important role in the percentage of found passwords [136]. 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 the previous experiment with 10 datasets, all 10 were chosen to be Layer 3 and as the results in Chapter 6 showed, when there was a large discrepancy in size, the much smaller dictionaries underperformed. For this reason, in this new experiment, 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 ĩ80,000 candidates and considerably under-performed compared to the equivalent of Layer 4. The sizes of the produced dictionaries can be found in Table 4.5.

Table 4.5: 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

4.10.2 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 [260]. More information about this model is given in Section 5.8.1 of Chapter 5.

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, e.g., words below a certain threshold could be considered as irrelevant to the seed word and therefore disregarded.

The specifics of the calculation of the proximity score as well as the implementation of the ranking are discussed in Section 5.8 of Chapter 5 and the results of the experiments can be found in 6.6 of Chapter 6.

4.11 Design Benefits, Limitations and Trade-offs

As can be seen in Figure 4.7, 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. As mentioned at the beginning of this chapter, where the possible scenarios in which a contextual approach in password cracking might make sense, every case is different, therefore the sequence of steps cannot be predetermined.

As mentioned in Chapter 3, dictionary attacks are one of the most popular types of password cracking techniques used. A dictionary attack with a list stemming from a password leak is a good bet and in many cases the go-to approach, because real world passwords are being evaluated (with mangling rules) and it is still relatively fast (of course that depends on the size of the list and the number of mangling rules).

It can be argued either way whether a regular dictionary attack could take precedent over a context-based dictionary attack, depending on the specific case and the number of passwords to be retrieved. A good approach, for the offline scenario, would be to target easy-to-guess passwords first with a regular dictionary approach and then follow with a more intelligent attack for more difficult passwords later. For the online scenario, or that of a targeted individual, if the investigator is in possession of previous passwords, variations thereof should be tested first. These can also offer insights into the suspect user's personal mangling rule selection. In any case, the specific parameters of the case will dictate the choice.

Another advantage of this approach is that these contextual 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.

A significant consideration when choosing the proposed approach is the length

of the generated dictionary. A smaller dictionary will allow for a larger number of combinations of mangling rules to be attempted over a fixed time period (or fixed number of guesses). Smaller dictionaries will result in more mangled attempts being made based on more relevant password candidates, e.g., passwords in Layer 1 (which are direct links to the seed word) will be contextually closer to the seed word.

Of course, with ranking based on contextual proximity and setting thresholds for thematic distance, a very well tailored and targeted dictionary list can be created. As a result, given a fixed time (or fixed number of attempts), there is a trade-off to consider between checking more, i.e., more distant, password candidates and checking fewer, i.e., more related, candidates with more mangling rules. This is an especially important choice since with more layers added, the length of the dictionary list increases correspondingly.

Furthermore, the time it takes to add one more layer to the dictionary list, the distance from the seed word, increases exponentially. Unless there is a bank of common pre-computed seed words to be availed of, a smaller dictionary list might make more sense in some cases.

The last consideration for the proposed approach is the information that is included in it. As the traversal from the seed word to subsequent layers is taking place, the decision was made to only include links found in each DBPedia article. The reason for this is once again based on the trade-off.

In the initial design of this approach, adding the sanitized text of the abstract and/or article was considered. The approach consisted of an extraction of keywords from this text and the incorporation of them into the list along with the links. Ultimately, the inclusion of words from the abstract/article itself was decided against, as this did not offer any significant increase in value. It is also reasonable to assume that the links contained in each Wikipedia article are also the most important related topics to the original seed word. While, there is the possibility that some good pass-

word candidates are missed as a result of this decision, this trade-off is deemed acceptable to result in more relevant password candidates.

Finally, a key difference of the contextual, generated dictionaries to the traditional leaked lists of passwords from data breaches is that they are not human generated. A leaked list will have an advantage over any generated dictionary in so far that it includes inherently the human factor in the passwords. The techniques humans resort to when they create a password can be found and leveraged with a use of a leaked list, especially because the number of leaked password lists from data breaches is big enough to offer valuable insights into the most popular techniques chosen, as the results of the analysis of the 3.9 billion passwords will show. This is something that a generated list of passwords, contextual or not, cannot compete against. To some extent, mangling rules can rectify this, but the information contained in real-world password lists will still be one of the most valuable assets in every password cracking attempt.

Chapter 5

Implementation

5.1 Introduction

In password cracking, whether done for in the context of a lawful investigation or by an adversary looking to illegally gain access into a system, there are some tools and methods that are universally acknowledged and used across the board. They offer capabilities to parallelise password cracking, the ability to introduce custom mangling rules, and also to tailor the attack's parameters to one's needs. This chapter discusses the implementation of the setup of the experiments that are described in Chapter 4 and the use of the tools that have been used throughout this thesis for analysis and evaluation. Section 5.2 describes the process for retrieving and cleaning the HIBP dataset, while Section 5.3 presents the tools that were used for the analysis of said dataset, some of which like the zxcvbn strength meter are used for ensuing experiments as well. Section 5.5 introduces the Password Cracking Wordlist Quality (PCWQ) framework and discusses its procedural flow and tools that were used in it. Section 5.7 looks at the building blocks of contextual dictionaries and finally Section 5.8 looks at the setup used for the ranking and optimisation experiments of the contextual dictionaries.

5.2 Preparing the HIBP Dataset for Analysis

As mentioned in Chapter 4, the Have I Been Pwned dataset is analysed for extracting information about the makeup of users' passwords. This section specifies the steps that were taken to retrieve the plaintext and clean the dataset. In a nutshell, the plaintext passwords were gathered from `Hashes.org` and the `CynosurePrime` team. An identification and removal of non-human chosen passwords took place. A more detailed explanation follows.

5.2.1 Retrieving the Plaintext

In order to conduct a statistical analysis of the passwords from the HIBP_v5 dataset, the passwords are first required to be in clear text form. The Hashes.org website [261] contains lists of clear text values for many password datasets – including the five versions of the HIBP dataset. The recovery ratio from the HIBP_v5 hash list is above 99.2%. In 2017, the CynosurePrime team, a password research collective, managed to recover almost all passwords from the first version of the HIBP list [262], claiming a final recovery ratio of 99.9999%. One of the purposes of their work being research, their list of recovered clear text passwords was shared to the researchers in this work. CynosurePrime initially focused on HIBP_v1, and therefore their list contains passwords from this list removed from later versions. Those passwords were removed essentially because they were somehow corrupted, e.g., badly encoded, duplicates, or not generated by humans. For this analysis, the CynosurePrime list was merged with the one collected from hashes.org to enrich the dataset with passwords from the later versions of HIBP. While this list contains more than 99% of the passwords of HIBP_v5, it should be mentioned that the small percentage of passwords that has not been included has not been recovered by either the CynosurePrime team, or the Hashes.org team. These passwords can be

assumed to be some of the strongest in HIBP, which is something to be taken into account.

5.2.2 Cleaning the Dataset

One of the first steps in the clean-up process of the dataset was to remove all the passwords encoded in hexadecimal format, corresponding to approximately 35 million passwords. While being valid passwords, the tool used for the basic analysis would not handle them properly. Further, a majority of these hex encoded passwords consisted of inputs which were wrongly encoded or handled on the HIBP dataset creation.

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

The clear text list used for the remainder of this work is therefore composed of 515,680,539 unique passwords. Considering the count value from the HIBP_v5 for each password's occurrence in data breaches, this dataset represents a total of 3,951,907,330 passwords.

5.3 Tools Used for the Statistical Analysis of HIBP

In order to execute the analysis of the passwords of HIBP, several tools have been employed. An overview of their capabilities and use throughout this work is presented below.

5.3.1 Password Analysis and Cracking Kit

The objective of this analysis is to present global characteristics about the passwords, including the type of alphabet used and the most frequent patterns. The PACK [145] was used to analyse the HIBP_v5 dataset. PACK provides several analysis tools, but the included `statsgen` script provides the functionality needed to perform this analysis. More specifically, it includes information on password length, character sets used and simple and advanced masks. The `rulegen` script could have provided interesting results as well, but it unfortunately also requires an extensive use of memory and was not able to process the dataset. Another script of general interest (which does not apply for this analysis) is `PolicyGen`, which takes into account the password policy in place and produces rules and masks that adhere to this policy [263]. This script would be useful to reduce the search space for password cracking.

PACK analyses the composition of passwords and classifies them according to the type of character set used. For example, a password is associated with the category *loweralphaspecialnum* when it contains lowercase, special characters and numbers, e.g., `pa$$w0rd`, no matter what the order or frequency of appearance of the component characters are. A description of each of the categories used by PACK is shown in Table 5.1. Using this classification, PACK outputs the count of passwords in each category.

The analysis can be further refined as it focuses on character sets without con-

Table 5.1: Types of patterns used by PACK

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

sidering the internal password structure. For example, the category *loweralphanum* contains passwords like *12password*, *password12*, and *pass12word*. A more refined classification, where the internal order is considered, would separate these into three different categories. This further classification is important because the approach to guess these passwords will be different. Following the vocabulary used in password guessing techniques, these internal password structures are called "masks". Therefore, for the aforementioned examples, the corresponding masks would be *digitstring*, *stringdigit* and *stringdigitstring*, respectively.

5.3.2 pipal

As part of the analysis, another password analysis tool called `pipal`[264] was used. Pipal offers many functionalities, some of which exist in PACK as well, and these include information about character sets used and the internal architecture of the password. Pipal also offers the functionality of searching how many times a specific word is found in a password such as looking up the frequency of specific months

or years in passwords, allowing for the extraction of very specific contextual details from leaked lists of real-world passwords. For the purpose of this research, pipal was used to extract the top 100 passwords, as well as the top 100 base words. As stated in Chapter 4, a base word is defined as a password where non-alpha characters from the beginning and end have been removed.

5.4 Fragmentation

PACK can provide a basic overview of the dataset's composition. For a more in depth analysis, the Óðinn Framework [265] is used, which has been adapted and enriched particularly for this advanced analysis.

5.4.1 Fragmentation with the Óðinn Framework

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

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

appropriate to do so. This second step is performed using `SymSpellPy` [266], a Python implementation of `SymSpell` [267], which is one of the most efficient spelling correction algorithms [268].

As a ground truth for splitting text into single words, SymSpell needs a dataset of words with their corresponding frequency counts. This dataset has to be seen as a vocabulary list and not as a set of password candidates. The `SymSpellPy` library comes with a small English dictionary with counts as default. Such approach is very efficient for tasks such as autocorrection modules or other natural language processing tasks. However, passwords are likely to contain foreign expressions, purposely mistyped words, popular culture references/characters, celebrities, or slang words which are missing from standard language datasets and therefore using classical dictionaries would fail to properly fragment passwords.

An ideal solution relies on the existence of a dataset composed of fragments properly extracted from real passwords, which, from reference, does not exist. A new dataset is therefore produced by extracting words from 3,937,684,877 Reddit comments [269]. This source was chosen for two reasons: 1) the comments contain slang words and common expressions used on the internet, and 2) these comments are written in several languages, resulting in a multilingual dictionary.

5.4.2 Fragment Classification

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

- WordNet – To classify normal English words, WordNet [270] provides a *synset*, i.e., a set of synonyms relating to a single given word. As WordNet is built hierarchically, the tree can be climbed to get synsets with a broader meaning for the classified word.

- Functions – Functions check if a given input matches the patterns defined within them, e.g., years or dates.
- Dictionaries – Óðinn contains a collection of dictionaries, each of them listing words of a specific class, e.g., cities. These lists are mostly hand-crafted and refined.

Tests with Óðinn have shown that in most cases, WordNet is classifying words correctly. However, it quickly reaches its limit. For example, simple typos or slang words are not correctly classified by WordNet, which is only looking for exact matches. This is an issue with passwords, as it is common to use slang words and phrases, e.g., *iluvmymom*.

To compensate for this insufficient classification, enriching the dataset of words used with the non-classified fragments was a focus of this work. GloVe [271] was used to automate this process with its *Common Crawl 42B 300d*, a pre-trained model in English for GloVe [272]. The process used can be summarized as follows. A proximity score between each non-classified fragment and the previously defined categories is computed. This proximity score is the Euclidean distance between the embeddings of those words in GloVe. The fragment is then added to the categories for which the distance is smaller than a given threshold. This process was repeated as some fragments could remain unclassified after one pass but be classified in the second pass thanks to the previous extension of the dataset of words. Many fragments were still not classified using this process; mainly random strings, typos and slang. This is because they do not have a representation in the *Common Crawl* and therefore cannot be compared to the categories.

Once the classification is achieved, Óðinn produces the frequency counts for all the observed combination of classes, e.g., the number of times passwords are composed of a name followed by a year. As one of the motivations of this work was to analyse in more detail those classes and their combinations, Óðinn was

configured to save the classification of each password in addition to the aggregated data. Letter fragments that were classified by Óðinn as single and double letters without meaning are removed, e.g., “xf” is removed but “it” remains).

5.4.3 `zxcvbn` - Password Strength Meter

Classifying the passwords according to how easily they are cracked is one of the most important metrics when it comes to providing insight into password selection by humans and inform password policies and individuals on the best practices when it comes to password creation. There are many strength meters in use, in fact most services implement their own strength meters, and as was seen in Figure 2.3 of Section 2.4.1 these metrics can give widely different results. The metric zxcvbn [225] which is open source and also used by Dropbox as their password strength meter is the one used to assess the strength of the 500 million unique passwords in HIBP as well as in all experiments involving the measure of strength of found passwords throughout this thesis. This metric attributes an integer score between 0 and 4 to each password according to strength, with passwords in class 0 being the weakest and those in class 4 the strongest.

5.5 Password Cracking Wordlist Quality Framework

In order to assess dictionary quality, a methodology must be developed, that takes into account the metrics of quality that were discussed in Section 4.7.

This methodology starts with one or more input dictionaries that could stem from leaked lists of passwords or be generated, like the contextual dictionaries that have already been discussed in Chapter 4. These can be combined, categorised and their size can be reduced based on whether they have been ranked for popularity (in the case of leaked password lists, reducing the size would mean eliminating

passwords that don't feature repeatedly). At this stage, and especially for generated dictionaries that do not stem from leaked lists that contain human's behaviour inherently in them, mangling rules will be employed to create variations of the password candidates to imitate human behaviour.

The final dictionary would be fed into one or more password cracking tools, and the parameters of the experiment would be defined at this stage. E.g. if time is the metric of importance, the password cracking attacks would have a predefined time limit or number of attempted guesses. The results of this process will be evaluated based on the metrics presented in Section 4.7 and depending on the results the process could be restarted from the second step with tweaked parameters, e.g. a smaller or larger dictionary, more, less or different mangling rules, etc.

5.6 Password Cracking Tools

The main password cracking tools and algorithms that have been used throughout this work are presented in this paragraph. More information on them can be found in Section 3, but they are also briefly presented in the following section. The reason that more than one password cracking tools have been selected for the evaluation stage of PCWQ is that the differences in the way they work, could affect the results greatly, therefore having a more well-rounded view is helpful. More than that, some password tools need the information that is found within leaked passwords to create password candidates, like tools based on neural networks. Dictionaries that are generated from English words that do not contain this information would have an inherent disadvantage with these tools.

The success of the input wordlist is not only based on the above factors, but also on the tools used to do the password cracking. For example, PCFG works better and estimates more accurate probabilities when the input dictionary does not contain

only unique entries, but repeated ones. However, such datasets with repetitions are rarely available to the research community. Therefore, in order to test wordlists, it is essential to have a few different tools to evaluate them with.

For this purpose, the four tools mentioned in the list below were chosen.

- **John the Ripper** (JtR) is one of the most well known open source password cracking tools. It supports various OS and can crack many different types of hashes. It supports various attack modes, including brute force and dictionary attacks.

- **Ordered Markov Enumerator** (OMEN) [152] is a password cracking tool using a Markov model and produces password candidates in order of decreasing probability.

- **PRobability INfinite Chained Elements** (PRINCE) [153] is a password candidate generator that 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 to create new password candidates.

- **Probabilistic Context-Free Grammar** (PCFG) [273] utilises Machine Learning to train on leaked password lists and generate models that mimic password creation habits of users. PCFG is one of the state-of-the-art password cracking algorithms, but one of the key factors to its use that is of importance in the approach of this thesis is that it offers its best results when it is trained on leaked lists of passwords rather than generated ones, so it can leverage the information of password tendencies inside the real-world leaked passwords.

The aim of using these four tools is not to compare them and find which is the better one, rather to make sure the input dictionaries are compared as thoroughly

as possible. In order to perform this part of the process, the Password Guessing Framework (PGF)[274] is used. This tool is an open source tool to automate the process of comparing different password guessers.

The reason for using PGF is to avail of its ability to automatise the setting up of the cracking process. Indeed, PGF allows the setting up of 'jobs' which will be processed sequentially, where the parameters of the guessing tool can be defined, such as the input dictionary, target list (hashed or plain), the maximum number of guesses, etc. The results come in the form of `*.txt` and `*.csv` files containing an analysis of the number of found passwords, a list of those as well as data on cracking performance over time. All this information is then used for the creation of graphs and the evaluation of the input dictionaries.

5.7 Building Blocks of Contextual Dictionaries

The PCWQ framework can evaluate existing dictionaries lists that stem from leaked passwords from data breaches, as well as generated dictionary lists. In the quest to look at the role of context in password selection, leaked lists of dictionaries were evaluated against datasets of the same semantic topic, i.e., a dictionary list stemming from a website about Manga was used to try to crack passwords from another dataset about Manga, the results of which are reported in Section 6.3.

But the main focus of this thesis is the generation and evaluation of contextual dictionaries. For the generation of them, the method that has been selected is illustrated in Figure 4.7 of the previous chapter. As can be seen there, the starting point is a Wikipedia article that covers the topic which the generated contextual dictionary should be about. From there, the equivalent DBPedia article is used to gather the information needed to create the dictionary. The DBPedia version of Wikipedia uses mappings to assign an ontology type for each piece of information found in

the Wikipedia article, something that allows to harvest the links found within the Wikipedia article. This process is described in more detail in the following sections.

In order to extract information from DBPedia, the Python library `rdflib` [275] is used, which is a library for the Resource Description Framework (RDF) [276]. RDF is a data model that is used to merge graph data when the underlying schemas differ.

5.7.1 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, 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 into 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 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 added significantly increases the complexity. 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 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 [277].

The length and scope of the dictionary list can be configured at the moment of generation. It can be limited to one layer, two layers, three layers (CD_3), etc. With each new layer added, the quantity of data increases exponentially. Therefore, the trade-off between speed, dictionary length, and ultimate success rate is a consideration.

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

5.7.2 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 *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". Although the contents of these are thematically relevant and useful, these entries themselves do not offer added value and are therefore excluded from the 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 the process, and 2) as the size of the dictionary length decreases, a corresponding decrease in

processing time follows [278]. As an example, if the entry *The Girl From Ipanema* is found, these three entries are added to the list: *TheGirlFromIpanema*, *Girl*, *Ipanema*.

5.8 Dictionary Optimisation

For the purpose of this work, 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 time of 15 minutes 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 minute 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 targeted with an attack leveraging the latest Nvidia RTX 4090 GPU [279] running at 6.6 kH/s, evaluating the same number of candidates with a single GPU would take approximately 20 days.

5.8.1 Wikipedia2Vec

`Wikipedia2Vec` is a Natural Language Processing (NLP) model based on `Word2Vec` [280]. 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 context tend to have similar meanings.

The word embeddings can subsequently be used to estimate the similarity of the context in which they have appeared 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 [281]. The use of Wikipedia2Vec returns a similarity score between the seed word and each entry in the contextual dictionary, allowing for identification of contextual relevant entries and the ranking of them higher in the dictionary list.

5.8.2 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).

The 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 4.5. Before ranking, both versions of the unranked dictionary with either whole entities or split up to individual words, as described in Section 5.8.2, 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 [282]. For this experiment, a larger ruleset was chosen, `OneRuleToRuleThemAll` [283]. 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` [144], which is an open-source password cracking tool instead of the PGF that was used in previous experiments and the reason for this was speed of results. Hashcat was compared against the PGF framework and not only did it offer better results, but the decrease in running time was significant.

Chapter 6

Results

6.1 Introduction

This chapter presents the results of the experiments outlined in the previous two chapters of this thesis. More specifically, Section 6.2 discusses the results of the analysis of the 3.9 billion real-world passwords found in HIBP, including a look at the statistical makeup of the passwords, a strength classification and a split into their constituent fragments for further analysis. Section 6.3 contains the results of the evaluation of real-world passwords with leaked lists stemming from similar communities, and the first chance in this thesis to put the theory about the impact of context in passwords to the test. Section 6.4 presents a preliminary evaluation to showcase the methodology for creating contextual password lists, and Section 6.5 presents a much larger scale of this evaluation with 10 different datasets from various online communities. Finally, ranked and optimised contextual password lists are tested against some of the datasets of the previous experiment in Section 6.6 to showcase the improvement these techniques present compared to the unranked dictionaries.

6.2 Results of Statistical Analysis of HIBP

Table 6.1 shows the 20 most popular passwords found in the HIBP dataset, along with the percentage of the total accounts associated with each password. Many of these passwords feature heavily on the most common or worst password lists every year. As can be seen, sequences of numbers and keyboard walks are the most popular choices found in the Top 20 passwords, as well as some simple English words.

Table 6.1: Top 20 passwords in HIBP_v5

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

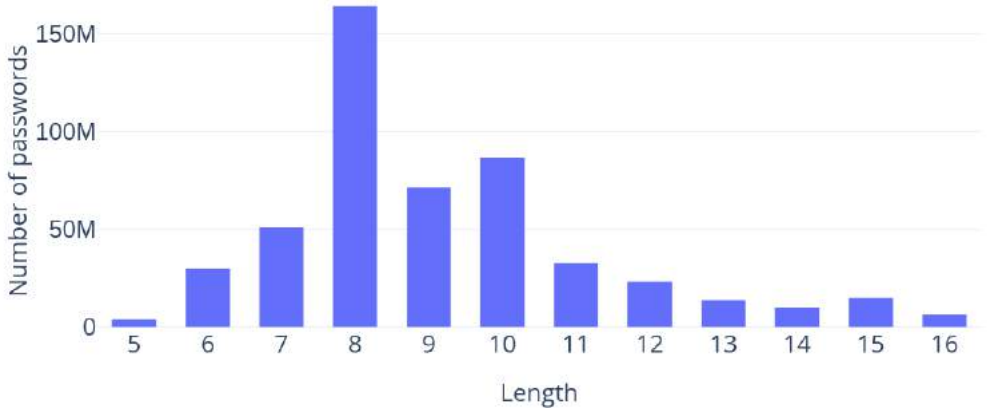


Figure 6.1: Most common password lengths in HIBP

6.2.1 Length Distribution

Figure 6.1 provides an overview of the most common lengths of unique passwords in the dataset, i.e., the aforementioned 515,680,539 passwords. One statistic that immediately stands out is that more than 30% of the unique passwords from HIBP_v5 are eight characters long. A highly probable explanation for this is that most password guidelines and policies specify minimum length requirements, such as the 8 characters minimum in the NIST recommendation [101]. The second most frequent length is 10 – corresponding to 17% of the passwords. The overall password length ranges from 1 to 449 characters, yet 84% of the passwords have a length that falls into the 6-12 character range.

6.2.2 Character Sets Usage

An analysis of the character type composition of the unique HIBP_v5 passwords can be seen in Figure 6.2.

Figure 6.2 shows the distribution of these categories, where in *other* the lowest represented categories are combined. As can be seen in the figure, 46% of the passwords are composed of a mix of lowercase characters and numbers, which

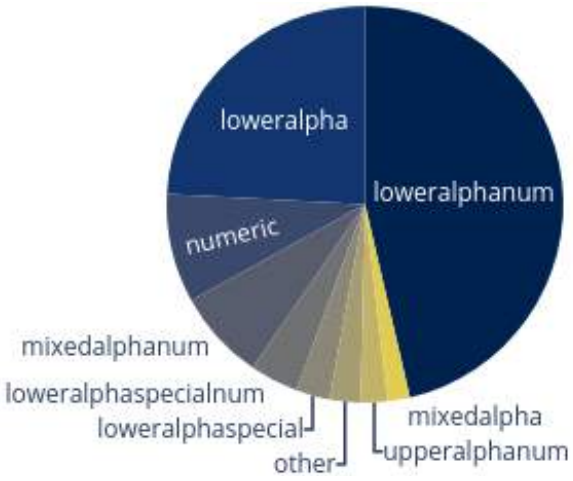


Figure 6.2: Occurrence of combinations of character categories in HIBP

is consistent with Weir et al. [12] and their findings from analysis of the RockYou dataset. The second and third-largest classes correspond to passwords composed of only lowercase (24%) or only numbers (8%) respectively. One notable observation from this analysis, is that over 75% of passwords from the dataset contain neither special nor uppercase characters. This is not such an unexpected outcome, as most password policies require at least 2 different character sets to be present in a password [284].

6.2.3 Pattern Analysis

The 15 most common masks from the HIBP_v5 dataset are shown in Figure 6.3. The most common mask is *stringdigit*, meaning that the passwords of this category are composed of a string (lowercase and/or uppercase) immediately followed by one or more numbers, e.g., `paSSword123`). As determined by Tatlı [285], users typically pick an alphanumerical string, commonly a word or a name, and add numbers at the end to fulfil the length and character set requirement of the enforced password policy. The next most common masks are *string*, *digit* and *digitstring*. These four masks combined represent over 75% of the passwords.

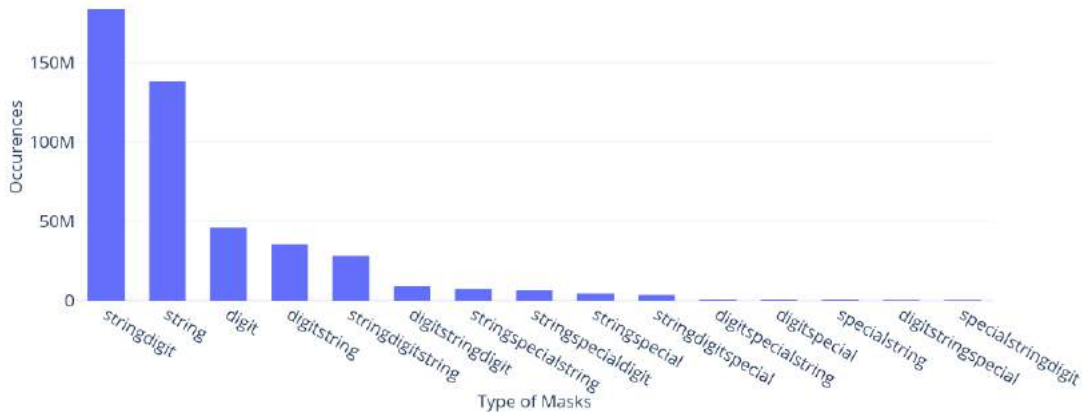


Figure 6.3: Most common simple masks in HIBP

Table 6.2: Breakdown of password fragments per category

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

6.2.4 Analysis on Password Fragments

At this point, two cases were possible for the advanced analysis: either analysing the unique passwords, or analysing the passwords considering the number of occurrences in the dataset. The latter option better maps the human behaviour, and therefore the below analysis relies on the 3.9 billion non-unique passwords of HIBP_v5.

Table 6.3: Top 50 letter, number and special character fragments

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

Óðinn produced 1,575,290,376 fragments out of the unique passwords in HIBP_v5, the breakdown of which can be seen in Table 6.2. The three lists, namely letters, numbers and special characters, were further processed in order to see the most common fragments of each category. A full table of the Top 50 most frequent fragments in all three categories can be found in Table 6.3. Interestingly, 9 out of the top 10 number fragments found across the HIBP dataset are the same as those found in RockYou in an earlier study [12]. The only entry not to feature in the top 10 of RockYou is 123456789 which is replaced 13.

a and i, which were respectively classified as an article and a pronoun, hold the top spots. As they are frequently encountered in the "Top Worst Passwords" lists verbatim or as parts of a password, qwerty, password and love are expectedly rounding out the top 5 [286]. In the top 50 there are some fragments that consist of phrases, such as iloveyou. The reason this is not broken down further is that, as mentioned in Section 5.4, the training of Óðinn was done with Reddit comments and this phrase appeared verbatim there and is therefore considered a single word.

Furthermore, keyboard walks such as qwerty, qwe, qaz are featuring prominently in the top 50 for both word and number fragments. The same holds true for the top 50 number fragments, where 3 out of the top 5 most frequent fragments are sequences of numbers. Furthermore, single digits, 1, 2, 3, double digits 12, 11, 13, and number repetitions 111111, 000000, are encountered in the top 50 number fragments. When it comes to special characters, the top 15 most encountered special characters are single, followed mostly by patterns of repetition. It is worthy to mention that the order of magnitude for the top 50 special characters is one order smaller than the top 50 letters and numbers. This corroborates the suggestion that users prefer alphanumeric characters and tend to avoid those that require multiple keys to type, as is often the case with special characters [195].

Looking further down at the number-based fragments, some noteworthy frag-

ments are found in the top 500. When it comes to numbers, many four-digit numbers were found in the top 500 number fragments falling within the 1900 to 2020 range, i.e., common years. The first appearance of a four-digit number that is presumably a year is `2010` at no. 56 and subsequently an overall of 37 four-digit numbers between 1970 and 2010 appear in the top 200 alone. This leads us to believe that users often choose memorable patterns even for the number portion of their passwords, e.g., year of birth or other important dates. In what concerns special-based fragments, most of them are repetitions of the same character, e.g., "!!" at rank 16. Some meaningful structure was still present in the top 500 in the form of emojis, such as ":)" at rank 65 or "^_^" at rank 198.

6.2.5 Analysis on Classified Fragments and Passwords

Table 6.4 Lists the most frequent classes of fragments occurring in the HIBP passwords. The fragments that were not classified at all or those not semantically meaningful, i.e., char/twochar/threechar, were filtered from this list. The three first classes are related to numbers, either generic ones like single digits, common ones, e.g., `123456` or `1111`, etc., or years. On one hand, this can be explained by the fact that many password policies require passwords to contain more than just letters. On the other hand, numbers are also very popular in Asian countries, most probably due to the fact that they can be digitally entered more easily than ideograms, especially on mobile devices [252]. The top 25 classes contains semantically-rich categories such as cities, animals, food and sports, reinforcing the idea that the surrounding context of a person might influence the choice of the password. However, it is not possible to affirm with conviction that this is the case, e.g., the name of a city can be unrelated to the person who chose it.

Identifying the most common combinations of component passwords classes enables the analysis of the unique classes. The results are displayed in Table 6.5.

Table 6.4: Most frequent classes of component password fragments. The count represents how many passwords in which this class occurred at least once.

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

Similar to the most frequent fragments, numbers and names are commonly used in combination with other classes. The number-based passwords are followed by various combinations of female and male names in combination with appended single digits or larger numbers. When password policies require more than one type of character, users might consider “padding” their passwords with special symbols and/or numbers, like years, at the end in order to fulfil the length requirement. Furthermore, keyboard walks and cities are also popular choices.

The division of passwords among those classes is displayed in Table 6.6.

Table 6.6: Percentage of unique passwords per `zxcvbn` class

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

6.2.6 Password Guessability

The analysis of the guessability of passwords is outlined below for two scenarios, namely a fast and a slow hash function. For this purpose, the length of passwords in each of those classes has been measured. Figure 6.4 shows the proportion of passwords of a given length for each of the classes produced by zxcvbn. In the case of a fast hash function, passwords belonging to class 2 and below can be recovered by an exhaustive search and should therefore be considered as really weak.

zxcvbn provides, together with the score, an approximation of the number of guesses an adversary would need to guess a password. Based on this figure, a

Table 6.5: Most frequent password fragment combinations *x* represents fragments that were not classified.

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

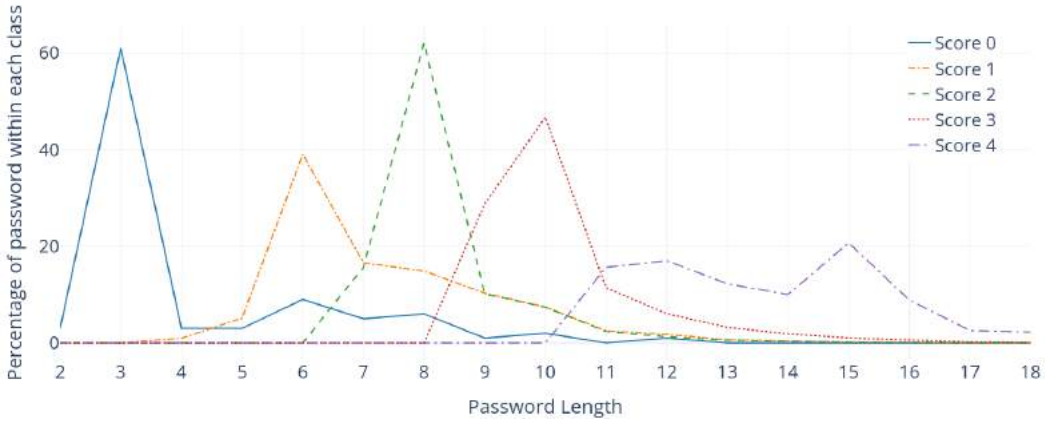


Figure 6.4: Password length distribution within `zxcvbn` score classes for HIBP passwords

password belonging to class 3 could be recovered using a single 2080 Ti graphics card in a time frame of approximately 5 days in the case of a slow hash function. Therefore, a digital investigator targeting a single password will manage to retrieve it. While this figure is indicative, it reveals that passwords in class 3 and below should be considered weak, especially as this time frame is only considering the use of a single graphics card. Adding another graphics card will effectually reduce the time linearly.

Class 4 passwords, at a first glance, are more secure. The minimum length of these passwords is 11 and 75% of those passwords have a length between 11 and 15. Based on the results from Óðinn, those passwords are composed of more fragments than on average, with 4.4 fragments for class 4 passwords versus 2.1 fragments for all passwords in HIBP_v5. According to the number of guesses required, which has an average of 5.8×10^{24}, passwords in this class are more resistant to classical attacks – even considering a fast hash function. However, 42% of these passwords are solely composed of lowercase characters and numbers. If prior knowledge about a given password is known, such as frequent used pattern(s) derived from other passwords of the same user, specific targeted attacks become

possible. The time required to fully explore the most common patterns of the password from class 4 considering a fast hash function is highlighted below:

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

Table 6.7: Comparison of the most frequent classes of password fragments between all the passwords and those from Class 4 in HIBP

Class	All Passwords	Class 4 Passwords
number	30.97%	**49.95**%
common-number	17.07%	5.03%
year	8.57%	14.8%
masculine_name	7.53%	8.34%
feminine_name	6.76%	7.41%
name	4.53%	8.75%
article	2.78%	7.05%
pronouns	2.59%	6.14%
city	2.47%	2.24%
special	2.33%	**12.73**%

Exhaustive search is nevertheless not the recommended approach to recover strong passwords. These figures serve to illustrate that even passwords considered as secure can be recovered when prior knowledge is available. To reinforce this idea, the advanced analysis results for the passwords of this specific class is presented. Table 6.7 highlights, for the 10 most used types of fragment, how often they appear in all passwords compared to class 4. As rightly recommended

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

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

6.2.7 A Brief Contextual Analysis of MangaTraders

As part of this analysis, and in order to investigate the hypothesis that there is a link between the thematic content of a website and the password chosen, it was decided to look at one specific leak from hashes.org. The leak that was chosen came from the website *MangaTraders.com*. The leak contains 881,468 entries (with 618,237 unique passwords). The pipal tool was used to extract the top 100 passwords, as well as the top 100 base words. A base word is defined as a password where non-alpha characters from the beginning and end have been removed. Table 6.8 shows that the top 100 passwords represent 4.76% of the total number of accounts. From these 41,821 passwords, 15,758 (or 37.6%) are manga related (representing 1.79% of the total number of accounts). Interestingly, looking at unique passwords only (and not counting the number of occurrences, 51 out of the top 100 passwords were related to manga. When it comes to base words, the percentage of manga

related base words is even higher (3.29% of the total and 63.8% of the top 100 base words).

Table 6.8: Manga related passwords in MangaTraders.com

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

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

6.3 Results of Dictionary Quality Assessment

In this section, an experimental analysis of how the dictionary evaluation process framework works is presented. In this experiment, the main focus is assessing password candidates that stem from leaked databases, to see whether a wordlist that is thematically similar to the list of passwords to be cracked can yield better results than a generic wordlist.

In this example use case, the evaluation datasets, BoostBot and MangaFox, as well as RockYou are used without modification with all four password crackers and 10 billion candidates were generated and evaluated for each process. The reason that 10 billion candidates were chosen is analysed in Section 4.9.4. The results of the cracking progress over time for RockYou, MangaFox and BoostBot can be found in Figure 6.5. As can be seen in all three figures, PCFG performs better for all three datasets and, especially in the case of RockYou, the result is much more distinguished. Comb4 contains `1,253,531` passwords, of which `1,096,481` are unique. Of these, RockYou PCFG managed to crack more than 60% (`768,341`)

or in case of unique passwords, 617,016. This result is significantly more than the other three guessers, but also significantly more than MangaFox and BoostBot. In fact, RockYou performed better than both of those datasets with all four guessers.

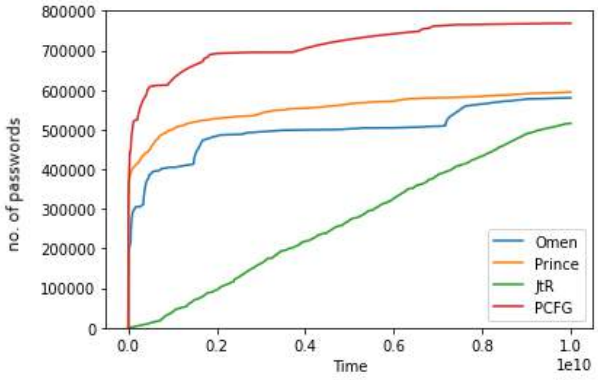
This result is not a surprise because the first key difference between RockYou and MangaFox and BoostBot is their size, as seen in Table 4.1. RockYou is about 32 times larger than MangaFox and 100 times larger than BoostBot. This means that there is certainly more diversity in the password candidates generated with RockYou. If then the focus shifts on only MangaFox and BoostBot, MangaFox performed slightly better, which can be on account of its larger size but also on the fact that the largest dataset in Comb4 is Manga traders, which is also another manga related leak.

Furthermore, it can be seen that PRINCE under performed with the smaller datasets, while it had the second-best performance with RockYou. This is due to the principle of PRINCE combining entries of the input dictionary to create new candidates. The input in the two smaller datasets are more sophisticated than those in RockYou. There, their concatenation leads to very complex candidates with a low probability of being in the targeted list. A pre-processing could be applied in PRINCE to better integrate such type of input wordlist.

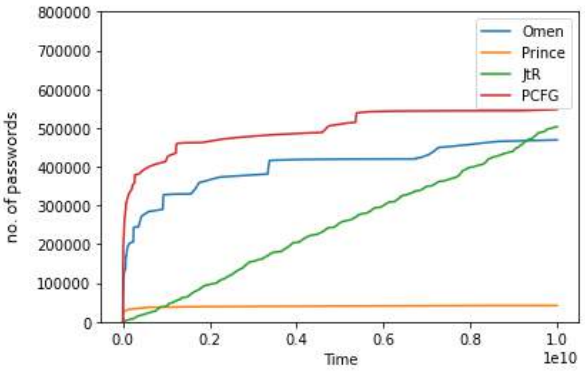
JtR on the other hand, steadily improved throughout the cracking process, almost reaching PCFG towards the end for both MangaFox and BoostBot.

Because the amount of cracked passwords, as mentioned in Section 4.7 cannot be the only metric to take into account - otherwise RockYou would have been the clear winner - the strength classes of the cracked passwords by each dataset were also examined.

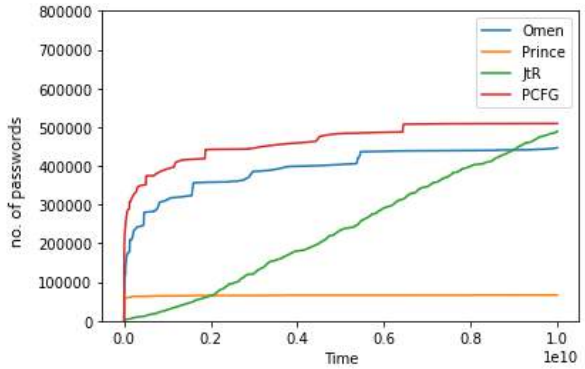
In order to evaluate that, zxcvbn, as referenced in Section 3.9.4, was used. With zxcvbn passwords are divided into 5 classes, according to their strength, i.e., how well they would withstand a cracking attack, with class 0 being the least secure and



(a) RockYou



(b) MangaFox



(c) BoostBot

Figure 6.5: Cracking Comb4: Progress over time for each dictionary

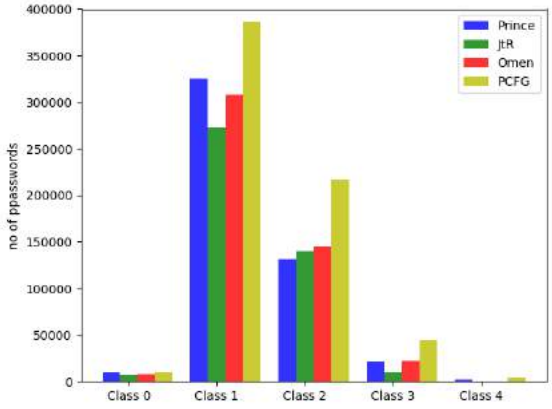
Table 6.9: Strength distribution of Comb4

	Comb 4	AxeMusic	JeepForum	MangaTraders	Minecraft
Class 0	46,645	4,143	34,832	6,471	1,199
Class 1	503,809	93,100	128,279	241,745	40,685
Class 2	395,202	87,158	58,189	205,218	44,637
Class 3	226,243	55,395	16,657	118,840	35,351
Class 4	81,624	12,898	1,388	45,962	21,376

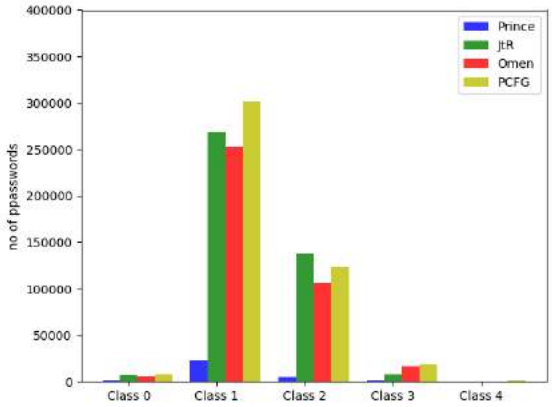
class 4 being the most secure. This classification takes into account rules set by common password policies but also l33t speak, common passwords and patterns to make a determination. Table 6.9 shows the classification of Comb4 in these classes and Figure 6.6 shows the cracked passwords per class for RockYou, MangaFox and BoostBot respectively, with all four password guessers.

As can be seen in the figures for all three datasets, the distribution of found passwords follows the distribution amongst classes of the Comb4 dataset, which can be seen in Table 6.9. For RockYou, it can be seen that PCFG as expected, performed better in all classes (except class 0, where all four are on par) with an especially big difference for class 3 and 4 compared to the other guessers. When it comes to MangaFox, other than PRINCE, the performance was similar for the three other guessers. Interestingly, MangaFox and BoostBot were able to find about one third as many passwords in class 4 as RockYou with PCFG, especially considering the big difference in size. Even more remarkably, MangaFox and BoostBot outperformed RockYou in the case of Class 4 with both JtR and OMEN.

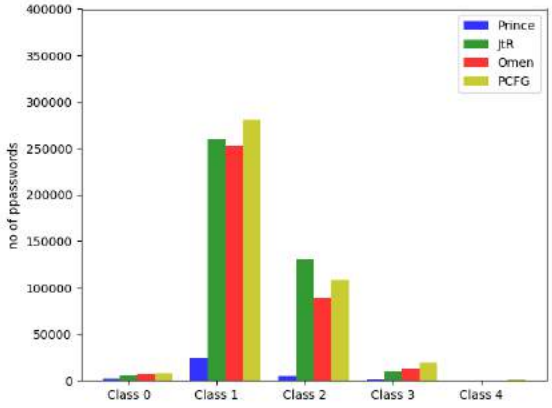
Frequently, in real world scenarios, the way to go would not be to choose one password cracking guesser or input wordlist over the other, but stack them. For this reason, the next step was to see how using a big input dataset like RockYou could be complemented, rather than beat. Table 6.10 shows the number of unique



(a) RockYou



(b) MangaFox



(c) BoostBot

Figure 6.6: Strength of cracked passwords for each dictionary

Table 6.10: Passwords found by BoostBot and MangaFox but not by RockYou

		JtR	OMEN	PRINCE	PCFG
All	BoostBot	26,109	32,911	5,788	17,811
	MangaFox	26,694	37,977	3,608	22,121
Class 0	BoostBot	182	265	22	73
	MangaFox	210	227	35	109
Class 1	BoostBot	11,960	16,698	3,095	3,659
	MangaFox	12,005	14,603	1,664	5,393
Class 2	BoostBot	10,439	11,628	1,730	8,303
	MangaFox	12,192	16,702	1,476	11,303
Class 3	BoostBot	3,512	4,171	796	5,266
	MangaFox	2,285	6,325	373	4,868
Class 4	BoostBot	16	149	145	510
	MangaFox	2	120	60	448

passwords that were found only by MangaFox and BoostBot and not by RockYou, in total, and also their distribution amongst the 5 classes of `zxcvbn`.

As can be seen in Table 6.10, this is a substantial addition of found passwords. In fact, with PCFG, the addition of either the passwords recovered by MangaFox or BoostBot, brings a 14% increase to the total, which is an important addition, being that this is the class of passwords that is the least easy to recover. Even in the case of PRINCE that generally underperformed, 73% for MangaFox and 63% for BoostBot, of the passwords that were found with these two datasets, were not recovered by RockYou. The recovery of class 4 passwords with OMEN was even more impressive because about twice as many passwords were recovered with MangaFox or BoostBot compared to RockYou. Finally, even in the case of JtR, with a meagre 2 passwords recovered from MangaFox and 16 from BoostBot, RockYou did not manage to find any of class 4.

Table 6.11: Passwords found of each dataset of Comb4, by each input wordlist for all four password guessers

		JtR	OMEN	PRINCE	PCFG
AxeMusic	RockYou	60,583	86,417	88,776	131,485
	MangaFox	57,923	67,934	6,456	93,843
	BoostBot	57,120	63,969	8,027	86,090
JeepForum	RockYou	96,894	105,232	109,847	133,665
	MangaFox	93,250	74,535	7,461	92,265
	BoostBot	89,084	72,753	6,966	83,477
MangaTraders	RockYou	267,553	289,903	299,890	373,483
	MangaFox	260,126	234,834	18,067	255,964
	BoostBot	252,338	221,328	22,121	241,774
Minecraft	RockYou	39,050	42,630	36,335	55,226
	MangaFox	41,417	43,171	3,561	50,221
	BoostBot	40,624	39,345	5,741	43,953

6.3.1 Breakdown of Comb4

In order to assess the quality of the input wordlists even further, Comb4 was broken down into the four individual datasets it was generated from, AxeMusic, JeepForum, MangaTraders and Minecraft. The breakdown of these datasets to zxcvbn classes is shown in Table 6.9. Additionally, Table 6.11 shows the amount of passwords found of each dataset of Comb4, by each input wordlist for all four password guessers. The result that pops up is that in the case of Minecraft, and excluding the underperforming PRINCE, the amount of passwords found by RockYou, MangaFox and BoostBot are very similar. A possible explanation of these results is that the thematic proximity compensates for the difference in size. In fact, BoostBot is the smallest dataset (about 100 smaller than RockYou) but thematically is the one closest to Minecraft. And MangaTraders is still a lot more relevant to Minecraft than, for example, JeepForum.

Table 6.12: Breakdown by dataset for PCFG, Class 3 and Class 4

	MangaFox		BoostBot	
PCFG	Class 3	Class 4	Class 3	Class 4
AxeMusic	1.3%	0.4%	1.6%	0.5%
JeepForum	0.9%	0.6%	0.9%	0.4%
MangaTraders	2.2%	0.7%	2.7%	0.9%
Minecraft	4.1%	0.3%	3.4%	0.2%

Still, it can be seen that for the other three datasets, RockYou performs a lot better than MangaFox and BoostBot. Even in the case of MangaTraders, while the results for JtR and OMEN are close, RockYou's performance for PCFG is significantly better, since for PCFG the full RockYou list of 32 millions was used, so that PCFG can take advantage of repetitions of passwords to form better probabilities.

In this case, the size of the input wordlist makes the difference, and this along with the percentage of success is the one to watch. Still, as mentioned above, in real cases the goal is not to choose one wordlist over the other but to complement it. For this reason another metric is considered again, the performance for stronger passwords. In Table 6.12 PCFG is the focus, the Class 3 and Class 4 passwords that were recovered by only MangaFox and BoostBot and not by RockYou. As can be seen, the percentage of passwords found by these two datasets and not RockYou was significantly higher for minecraft and MangaTraders, the two datasets that were contextually closer to MangaFox and BoostBot.

6.4 Experiments with Contextual Dictionaries

To measure the impact of contextual dictionaries, a number of password cracking experiments were conducted to compare the results of a contextual dictionary against a commonly-used baseline dictionary.

6.4.1 A Preliminary Experiment

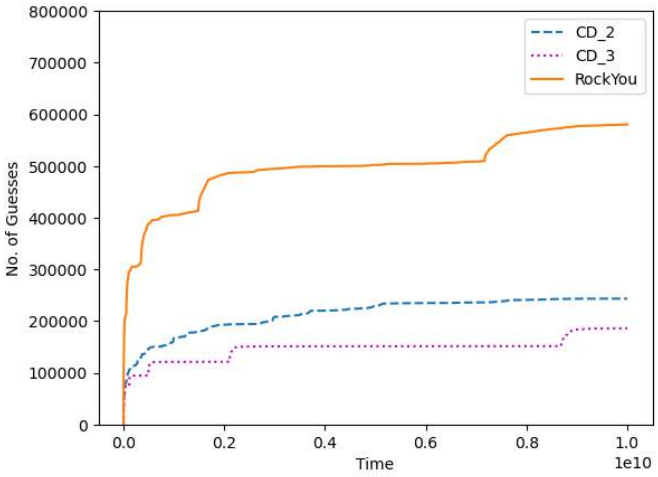
As with the previous experiments, where already existing dictionary lists were evaluated, in this case a newly created dictionary list will be evaluated with the framework proposed in Section 4. The seed word “Manga” was used for generating dictionaries of two and three layers, called CD_2 and CD_3 respectively. The lengths of CD_2 and CD_3 are 40,489 and 724,060 respectively, while the lengths of the Comb4 (and its constituent dictionaries as well as RockYou can be found in Table 4.1 as they are the same as the previous experiment. For the evaluation of the results, OMEN and PRINCE were used.

To conduct this preliminary experiment, University College Dublin’s Sonic High-Performance Computing Cluster was used. This cluster consists of 43 nodes with memory sizes ranging from 128Gb to 1.5Tb [287]. While time is dependent on the resources available for password cracking, as a reference, using the High Performance Computing (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 performed and an analysis of the results.

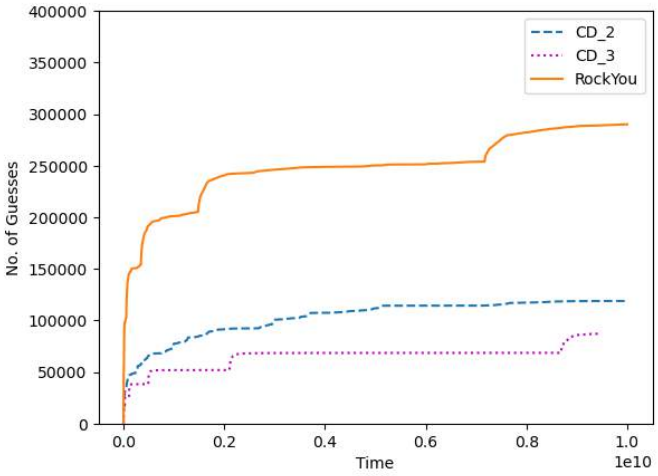
6.4.2 Results of the Preliminary Analysis

Both Comb4 and MangaTraders were evaluated using CD_2, CD_3, and RockYou as input dictionaries. 10 billion password candidates were generated again 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 6.7. 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 6.8 (a) and Figure 6.8 (b) respectively.



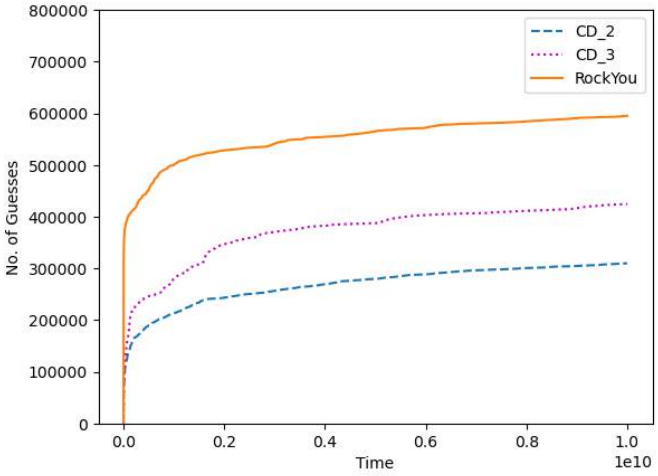
(a) Comb4



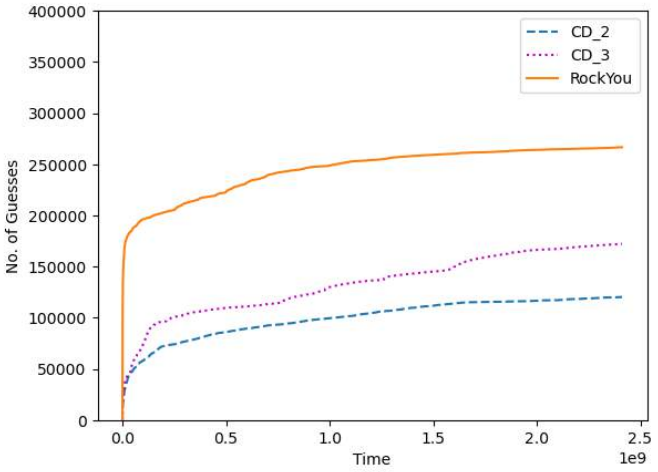
(b) MangaTraders

Figure 6.7: Dictionary evaluation for CD_2, CD_3 and RockYou using OMEN

A key difference between Figures 6.7 (a) and 6.7 (b) (which represents OMEN) and Figures 6.8 (a) and 6.8 (b) (which represents PRINCE), is that CD_2 is more performant compared to CD_3 using OMEN and CD_3 is better with PRINCE. The explanation for this resides in the inner configurations of each of these tools.



(a) Comb4



(b) MangaTraders

Figure 6.8: Dictionary evaluation for CD_2, CD_3 and RockYou using PRINCE

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, i.e., 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 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 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 RockYou is also a diverse dictionary that represents to a very large extent how people create their real-world passwords. RockYou is indicative of the password culture across society, which is why it is one of the most popular dictionaries for password cracking attacks.

When comparing Figure 6.7 (a) to Figure 6.7 (b) and comparing Figure 6.8 (a) to Figure 6.8 (b), 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.

Strength Analysis

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.

This is why it is important to also examine other metrics. For example, how strong are the passwords being cracked? For this, the password strength meter zxcvbn, which is the Dropbox-developed strength meter, has been used. According

to this meter, passwords are classified into five different classes based on 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.

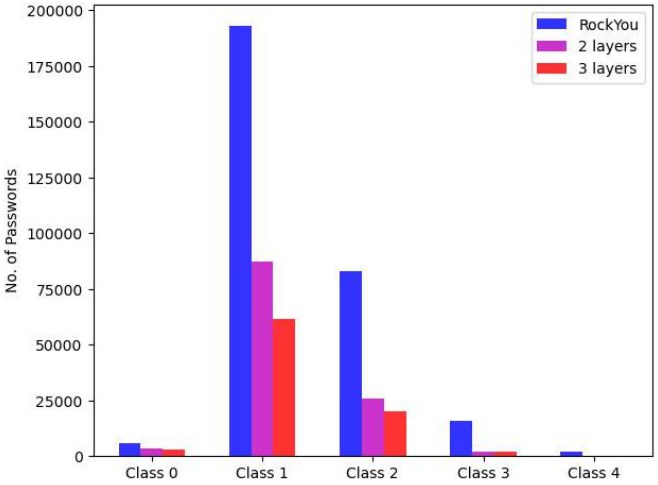


Figure 6.9: Passwords cracked by OMEN with CD_2, CD_3 and RockYou, classified by `zxcvbn`

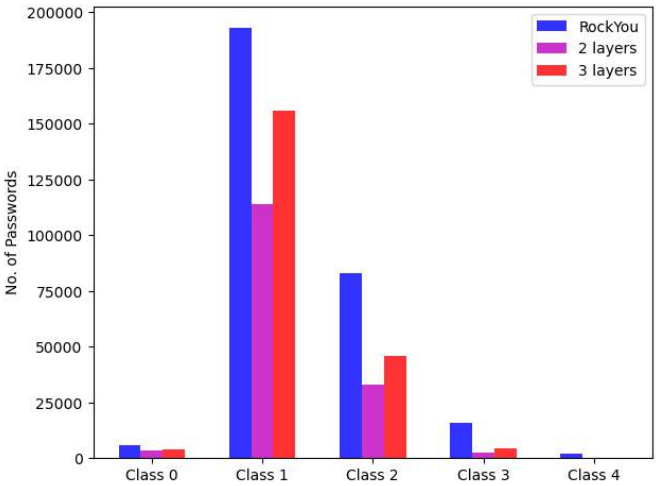


Figure 6.10: Passwords cracked by PRINCE with CD_2, CD_3 and RockYou, classified by `zxcvbn`

Figure 6.9 shows how many passwords have been cracked per zxcvbn Class for CD_2, CD_3 and RockYou using OMEN and Figure 6.10 shows the correspond-

Table 6.13: Strength distribution using `zxcvbn` for CD_2, CD_3 and RockYou, using OMEN

	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

Table 6.14: Strength distribution using `zxcvbn` for CD_2, CD_3 and RockYou using PRINCE

	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

ing results from using PRINCE. It can be seen that, for both OMEN and PRINCE, the number of Class 1 passwords that have been cracked with RockYou is very large. The reason for this is that RockYou is a generic dictionary list of popular passwords. It is reasonable that RockYou would perform well for passwords that are easy to crack. With zxcvbn, passwords from Classes 0 to 2 belong to this “easy” category [4].

Tables 6.13 and 6.14 offer a breakdown of how many passwords were cracked by each dictionary per class and per cracking tool. As can be seen in Table 6.13, 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 rest of the classes, the results

Table 6.15: Passwords only found using the contextual-based approach of Manga 2 layers or 3 layers (OMEN)

	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

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 6.14. 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 the results achieved using CD_2 and CD_3 versus RockYou is not what demonstrates the true value of the proposed approach.

6.4.3 Considerations of a Real-World Application/Unique Passwords

If a real-world law enforcement password cracking scenario is considered, RockYou (or similar) can be used to crack passwords while simultaneously using the approach proposed as part of this research. 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 6.15 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 6.16 shows the same for PRINCE. From these two

Table 6.16: Passwords only found using the contextual-based approach of Manga 2 layers or 3 layers (PRINCE).

	CD_2 Unique	CD_3 Unique
Class 0	12	12
Class 1	3,265	14,545
Class 2	4,092	12,927
Class 3	1283	2,619
Class 4	46	179

tables, it can be observed that, in fact, there is 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 simply running RockYou. A similar increase can be observed in the case of CD_3 in the recovery of unique passwords for CD_3 versus RockYou. Therefore, 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 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 a digital investigation with a tech-savvy suspect is considered, where - if their password is vulnerable

to dictionary attacks - it's still more likely to be Class 3 and above.

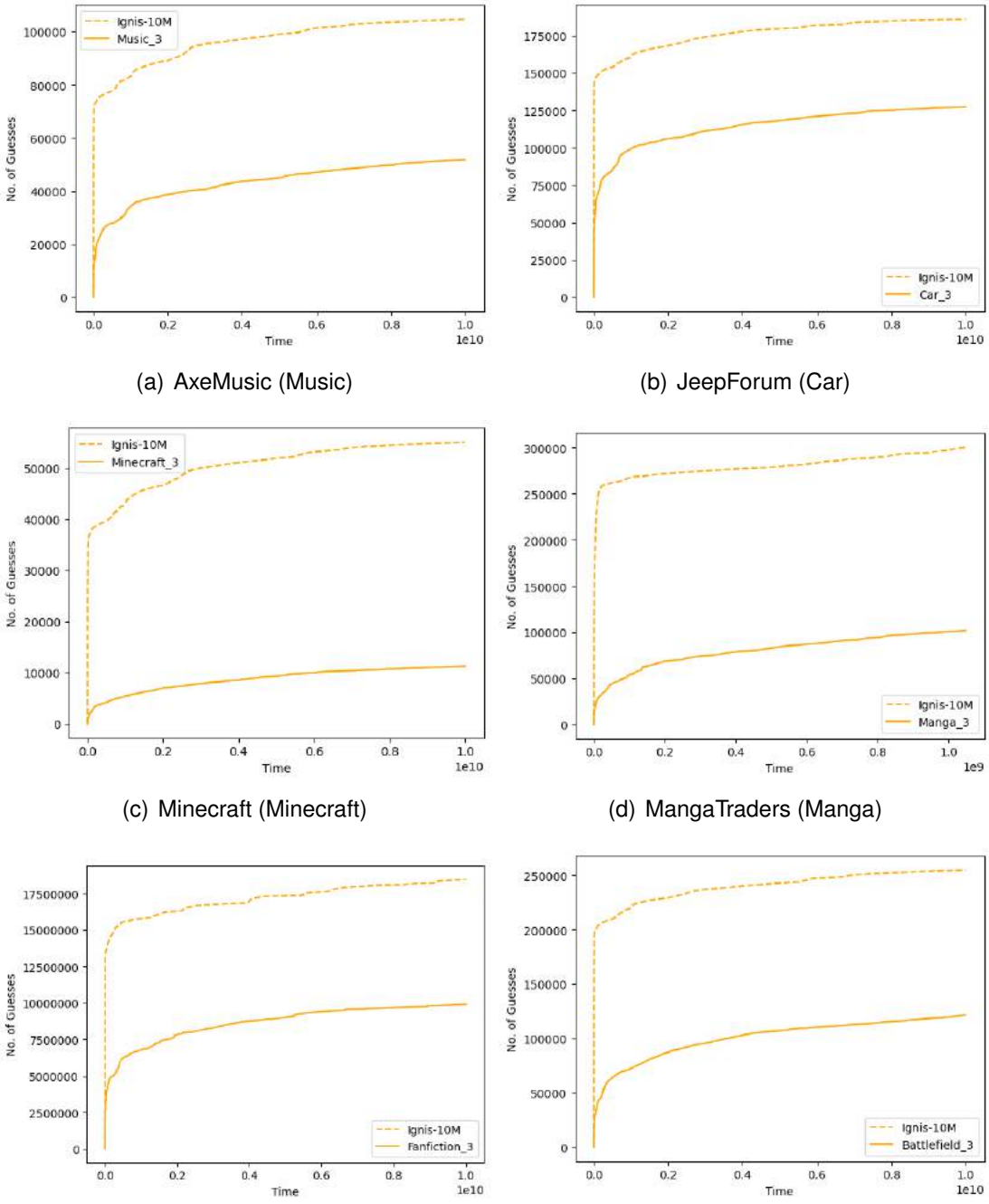
In general, the highest increase in found passwords was achieved with CD_3 and PRINCE. CD_2 achieved finding 10.1% more passwords that were not already recovered by Rock You. For Class 1 passwords, this increases to 15.5%. This is a very significant percentage, especially considering that - as mentioned 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.

6.5 Evaluating Dictionary Generation

As mentioned in Chapter 4, ten datasets across various topics were chosen. Related contextual dictionaries with the help of Wikipedia/DBPedia were created. The depth of the dictionaries was selected as 3 and the number of guesses as 10 billion, as defined in the previous section. The cracking process over time for these ten datasets, with both the baseline dictionary (Ignis-10M) and the contextual dictionary, can be seen in Figure 6.11.

From Figure 6.11, it can be observed that for all ten datasets, Ignis-10M is the best performing dictionary. This is to be expected, as Ignis-10M is a compilation of different data leaks and contains some of the most popular passwords used by real-world users. Ignis-10M is also a 10 million entry dictionary, while the contextual dictionaries, as seen in Table 4.5 range from 1 million to 30 thousand candidates. It is therefore expected that Ignis-10M will perform better in comparison, and it will crack the most passwords across all different datasets as it is the most varied dictionary.

Focusing a little more into the varying results of the ten different contextual dic-



(a) AxeMusic (Music)

(b) JeepForum (Car)

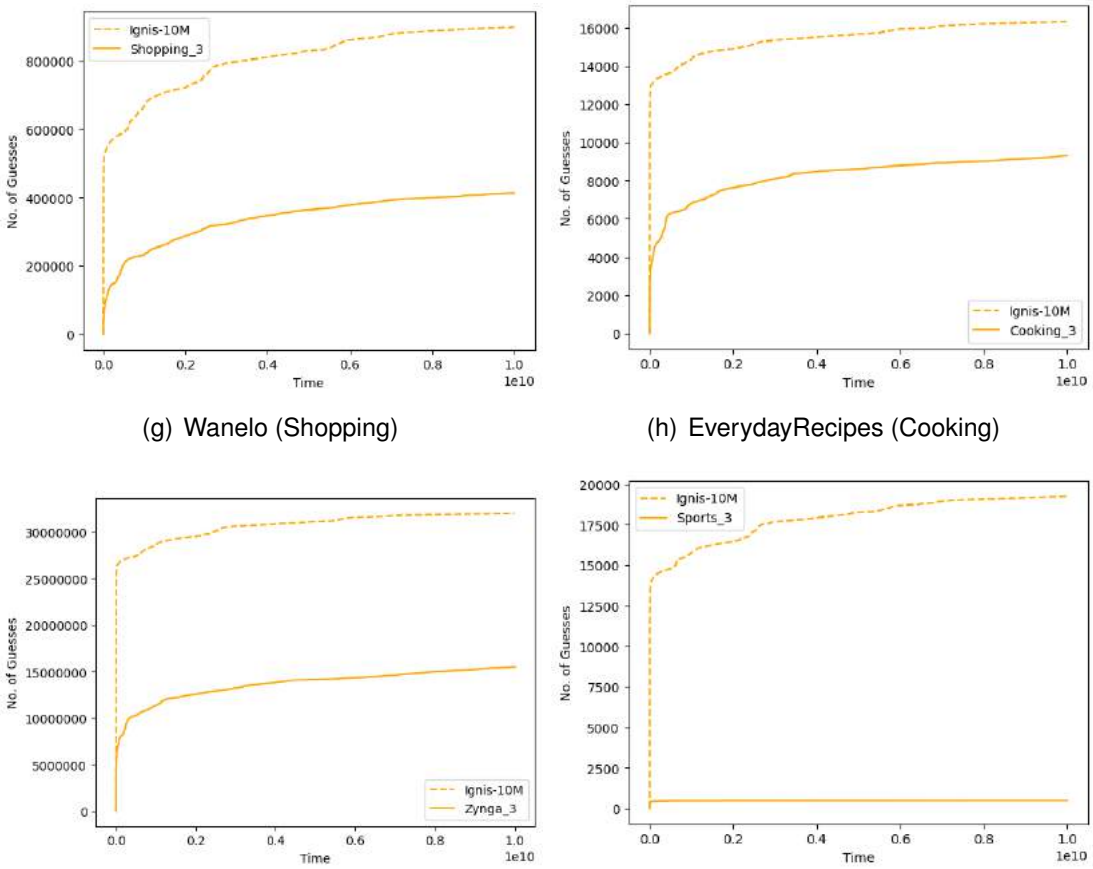
(c) Minecraft (Minecraft)

(d) MangaTraders (Manga)

(e) Wattpad (Fanfiction)

(f) Battlefield (Battlefield)

Figure 6.11: Passwords cracked by Ignis-10M and bespoke layer 3 dictionaries (seed word for bespoke dictionary in parentheses



(g) Wanelo (Shopping)

(h) EverydayRecipes (Cooking)

(i) Zynga (Zynga)

(j) DoSportsEasy (Sports)

Figure 6.11: (cntd.) Passwords cracked by Ignis-10M and bespoke layer 3 dictionaries (seed Word for bespoke dictionary in parentheses)

tionaries (denoted as L_3s), it can be seen in Figure 6.11 that Music_3 and Car_3 had some of the best performances, while Sports_3 had the worst. This can be explained by the size of these dictionaries, with Music_3 being 1 million while Sports_3 is only 30 thousand. A dictionary of 30k candidates, even with the permutations allowed by 10 million guesses, cannot produce enough variance. This serves to highlight the importance of picking the correct seed word for generating a dictionary. If instead of “Sports”, “Sport” was chosen as the seed word, the layer 3 dictionary Sport_3 would contain 1,068,758 candidates, which is a very significant increase

Table 6.17: Total passwords cracked and improvement of the combination approach. The L_3 Excl. column contains passwords found only by L_3 dictionaries, while the L_3 Imp. column contains the improvement over Ignis-10M provided by the L_3 dictionaries.

Dataset	Ignis-10M	L_3	L_3 Excl.	L_3 Imp.
AxeMusic	41.3%	20.5%	2.47%	5.97%
JeepForum	68%	39.2%	2.32%	5.19%
Minecraft	38.4%	11.2%	0.76%	3.88%
MangaTraders	57.2%	28.2%	2.61%	4.56%
Wattpad	39.7%	15.2%	0.69%	17.86%
Battlefield	60.6%	29%	2.21%	3.64%
Wanelo	42.1%	19.3%	2.38%	5.64%
EverydayRecipes	64.4%	36.7%	2.24%	3.47%
Zynga	37.9%	15.7%	1.22%	10.61%
DoSportsEasy	41.7%	1%	0.06%	0.15%

over Sports_3. If “Sport” was used as a seed word instead, then better results would be achieved when cracking DoSportsEasy, but the decision was made to use “Sports” to demonstrate the pitfalls of picking a bad seed word.

One interesting metric when it comes to the performance of these contextual dictionaries is how well they would do “stacked”, i.e., in a combination attack. To this end, Table 6.17 shows the percentage of unique passwords cracked by Ignis-10M and the L_3 contextual dictionaries. Column 3 of the table also presents the percentage of passwords that were only cracked with the L_3 dictionaries for each of the ten cases, i.e., the exclusively cracked passwords. Finally, Column 4 presents the improvement over Ignis-10M if it is combined with the contextual approach.

As can be seen in Table 6.17, in most cases the contextual dictionary has found approximately half the passwords found by Ignis-10M. Although in some cases, like JeepForum and EverydayRecipes, this number is even higher. Considering that Ignis-10M is compiled by a number of different data leaks and therefore contains actual used passwords across a range of services, the results of the L_3 dictionar-

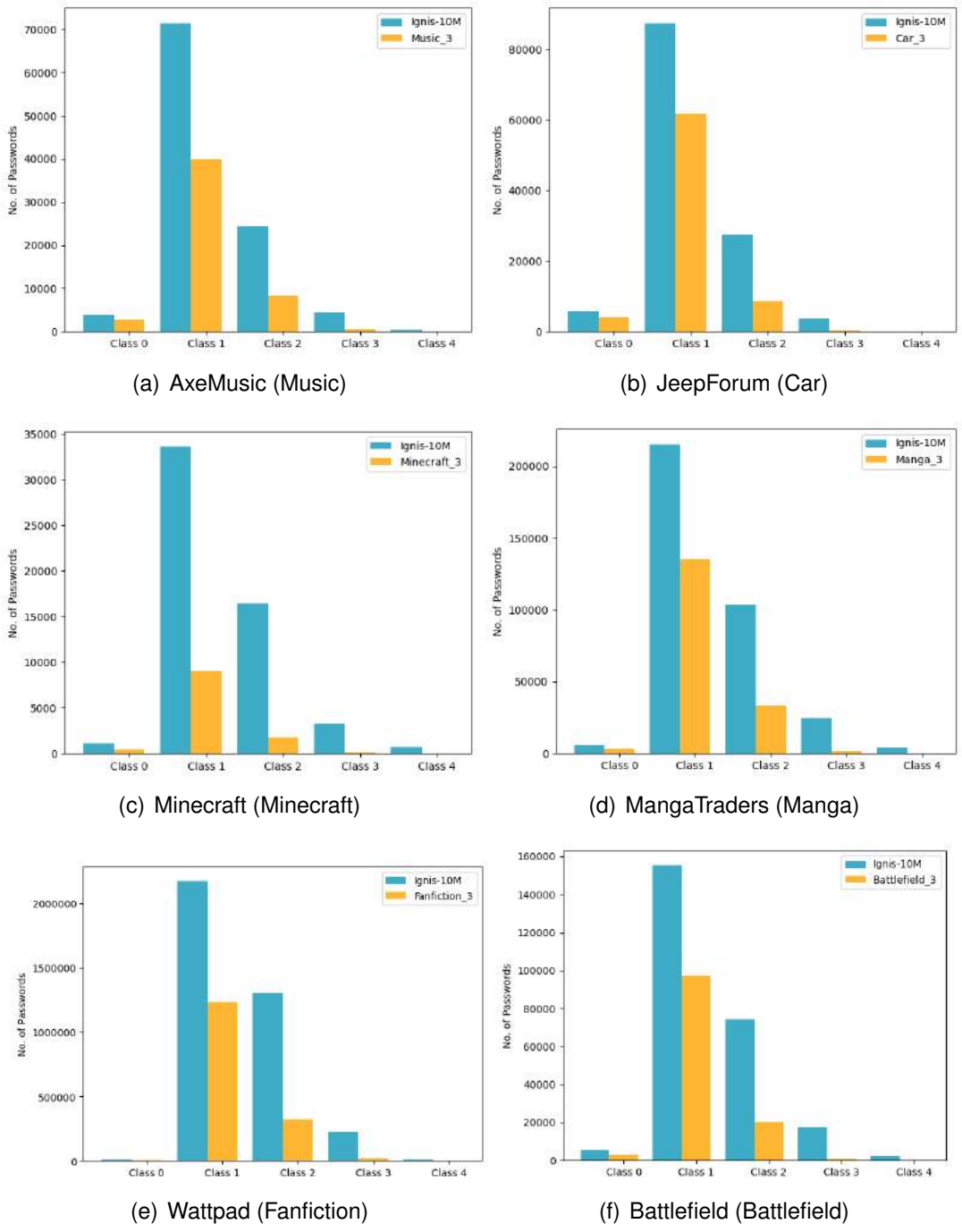
ies that are only dictionary words without any extra modification, is quite impressive. Once again, the only outlier is Sports_3, but this is somewhat expected since the input dictionary that was created from DBPedia contained only 30 thousand candidates.The passwords found exclusively by the contextual dictionaries offer on average an additional 2% of passwords, which in some cases represents a significant improvement over what was found by Ignis-10M alone.

For example, with the Wattpad leak, while the passwords found exclusively by Fanfiction_3, represent a 0.69% increase, this translates to a 17.86% improvement over Ignis-10M. The reason for this is that while Ignis-10M finds more passwords, these are passwords that are repeated many times in the leak, while the passwords found by Fanfiction_3 do not have as many repetitions. This could indicate that the passwords found by Fanfiction_3 are less frequently chosen by users and therefore less encountered.

It can also be observed that the choice of either generalising the seed word or keeping it the same as the target dataset did not influence the results in any significant manner. Nonetheless, from the five best performing dictionaries, four were from the “generalised seed word” category.

Looking at the case of a single law enforcement officer wanting to gain access to an encrypted device, the number of popular passwords cracked from one data leak is not the optimal way to judge the effectiveness of a dictionary. In fact, if a suspect is hiding behind an encrypted device, it is reasonable that they are more tech-savvy, and it follows that there is a good chance their password would be stronger than those found on the most popular password lists.

It is therefore important to also look at the quality of passwords cracked by the baseline dictionary and the contextual approach, i.e., the strength of these cracked passwords. To this end, Figure 6.12 shows the breakdown of the found passwords by both approaches, classified into five classes of strength. The password strength



(a) AxeMusic (Music)

(b) JeepForum (Car)

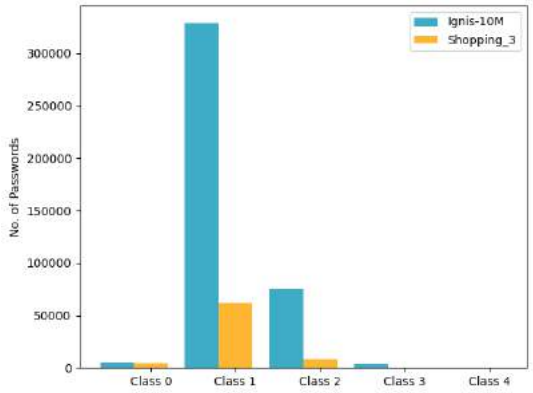
(c) Minecraft (Minecraft)

(d) MangaTraders (Manga)

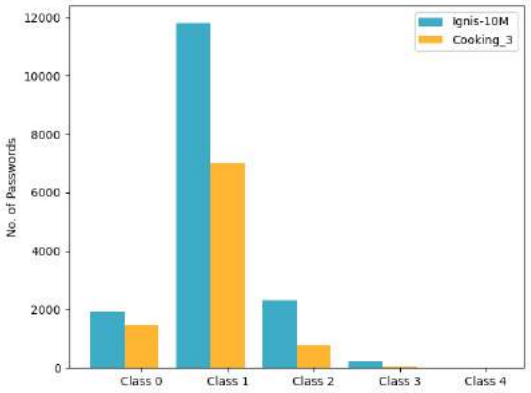
(e) Wattpad (Fanfiction)

(f) Battlefield (Battlefield)

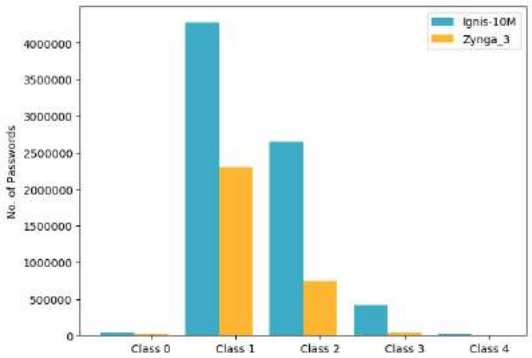
Figure 6.12: `zxcvbn` classification of passwords cracked by Ignis-10M and bespoke layer 3 dictionaries (seed word for bespoke dictionary in parentheses)



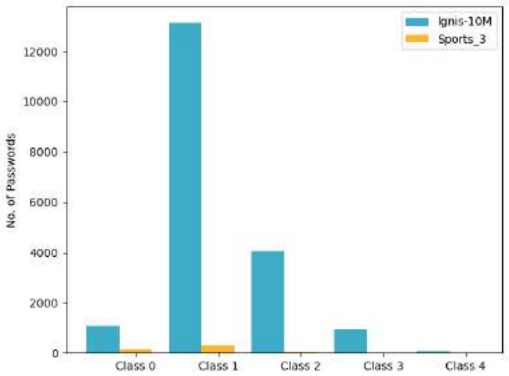
(g) Wanelo (Shopping)



(h) EverydayRecipes (Cooking)



(i) Zynga (Zynga)



(j) DoSportsEasy (Sports)

Figure 6.12: (cntd.) `zxcvbn` classification of passwords cracked by Ignis-10M and bespoke layer 3 dictionaries (seed word for bespoke dictionary in parentheses)

meter that was used for this classification is zxcvbn and the five classes range from 0 to 4, with Class 0 being the weakest passwords and Class 4 containing the strongest.

As can be seen in Figure 6.12, Class 1 and Class 2 passwords are those most commonly found, mostly from both Ignis-10M and the contextual dictionaries. This is because the passwords in these categories are easier to crack and would most likely be found by various approaches, as confirmed by [4]. It is therefore the Class 3 and Class 4 passwords that are the most interesting.

Tables 6.18 and 6.19 outline the passwords found by Ignis-10M, L_3, the ex-

Table 6.18: Class 3 passwords. The L_3 Excl. column contains passwords found only by L_3 dictionaries, while the L_3 Imp. column contains the improvement over Ignis-10M provided by the L_3 dictionaries.

Dataset	Ignis-10M	L_3	L_3 Excl.	L_3 Imp.
AxeMusic	4,504	581	286	6.3%
JeepForum	3,770	276	118	3.1%
Minecraft	3,247	80	46	1.5%
MangaTraders	24,524	1,906	942	3.8%
Wattpad	223,567	16,854	7,758	3.5%
Battlefield	17,330	755	281	1.6%
Wanelo	47,604	3,855	1709	3.6%
EverydayRecipes	254	41	24	9.4%
Zynga	417,404	33,752	15,735	3.8%
DoSportsEasy	934	6	1	0.1%

clusively retrieved by L_3, and the improvement percentage for Class 3 and Class 4 passwords cracked. By examining these two tables, it is notable that on average, approximately half the passwords found by L_3, are not found by Ignis-10M, cementing the importance of the proposed contextual dictionaries further. Furthermore, it can be observed that although the numbers are smaller compared to Class 3, Class 4 contains the strongest passwords and the percentage improvement of using L_3 on top of Ignis-10M is higher for Class 4. Notable examples are Fanfiction and Music achieving a 4.9% and 7.7% improvement respectively, In addition, for EverydayRecipes the percentage improvement is 42.8%, while acknowledging that the absolute numbers of recovered passwords are quite low for both.

6.6 Evaluation of the Ranked and Optimised Generated Dictionaries

As mentioned in Section 4.10, 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

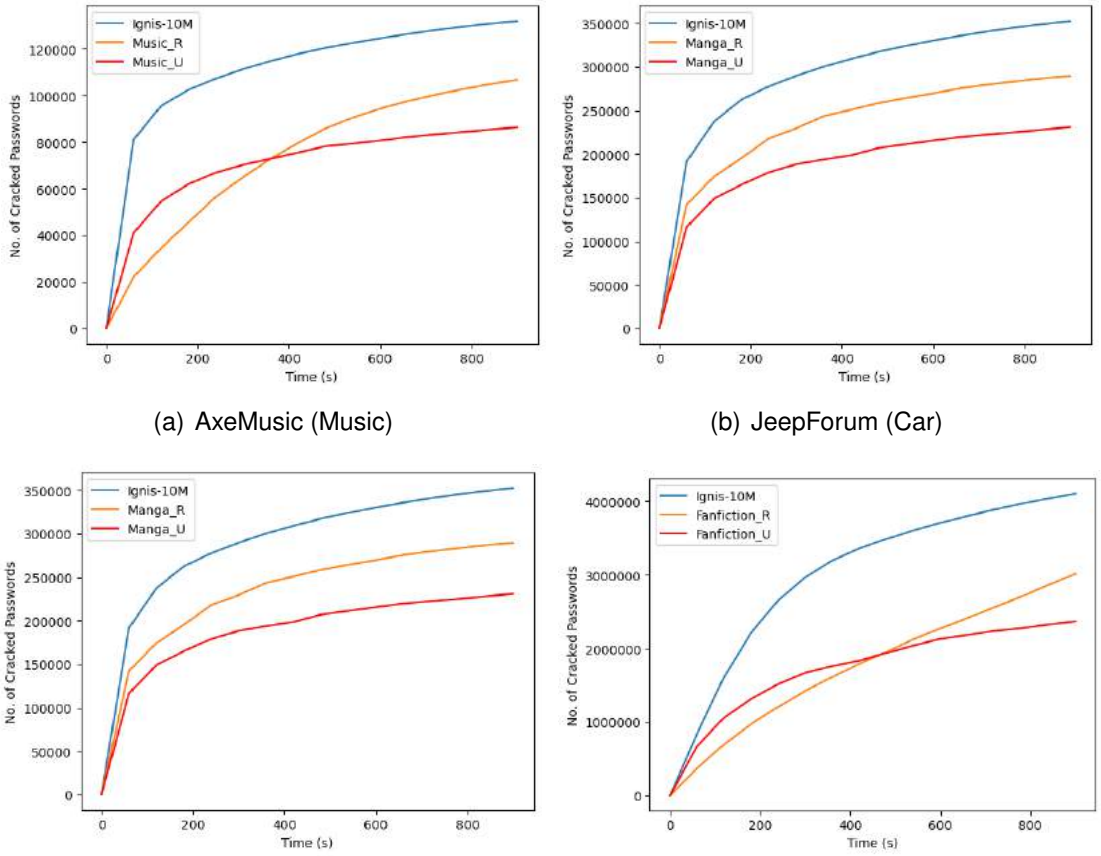
Table 6.19: Class 4 passwords. The L_3 Excl. column contains passwords found only by L_3 dictionaries, while the L_3 Imp. column contains the improvement over Ignis-10M provided by the L_3 dictionaries.

Dataset	Ignis-10M	L_3	L_3 Excl.	L_3 Imp.
AxeMusic	351	42	27	7.7%
JeepForum	96	9	5	5.2%
Minecraft	667	5	3	0.4%
MangaTraders	4,554	152	90	1.9%
Wattpad	15,022	1,095	673	4.9%
Battlefield	2,487	51	25	1.0%
Wanelo	2,953	199	100	3.4%
EverydayRecipes	7	3	3	42.8%
Zynga	28,211	1,403	849	3.0%
DoSportsEasy	60	0	0	0%

version. The ranked version was then produced with the methodology described in the same 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).

6.6.1 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 Figures 6.13(a) to 6.13(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.



(a) AxeMusic (Music)

(b) JeepForum (Car)

(c) MangaTraders (Manga)

(d) Wattpad (Fanfiction)

Figure 6.13: Number of passwords cracked over time by Ignis-10M and the rank and unranked versions of the contextual dictionaries

An interesting fact that can be observed in Figures 6.13(a) to 6.13(d) is that for AxeMusic and Wattpad, the unranked versions are performing better at the beginning until they are overtaken by the ranked versions. For JeepForum and MangaTraders, the ranked and unranked dictionaries have almost identical performance for the first couple of minutes of the experiment. The reason that the unranked dictionaries are either having a similar performance or outperforming the unranked dictionaries at the beginning of the experiment could be that even without ranking, the first entries in the unranked dictionaries are those directly linked from the seed

Table 6.20: Total number of passwords found. The R Excl. column contains passwords found only by ranked dictionaries, The U Excl. column contains passwords found only in 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

page, therefore are highly relevant. Soon after, the ranked versions of the dictionaries outperform the unranked for all four categories because the ranking helps promote forward the candidates with closer semantic proximity to the seed word.

Table 6.20 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 4.1. 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 6.20 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 sig-

nificant 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%.

Table 6.21: Top 20 password candidates that found the most passwords for each of 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

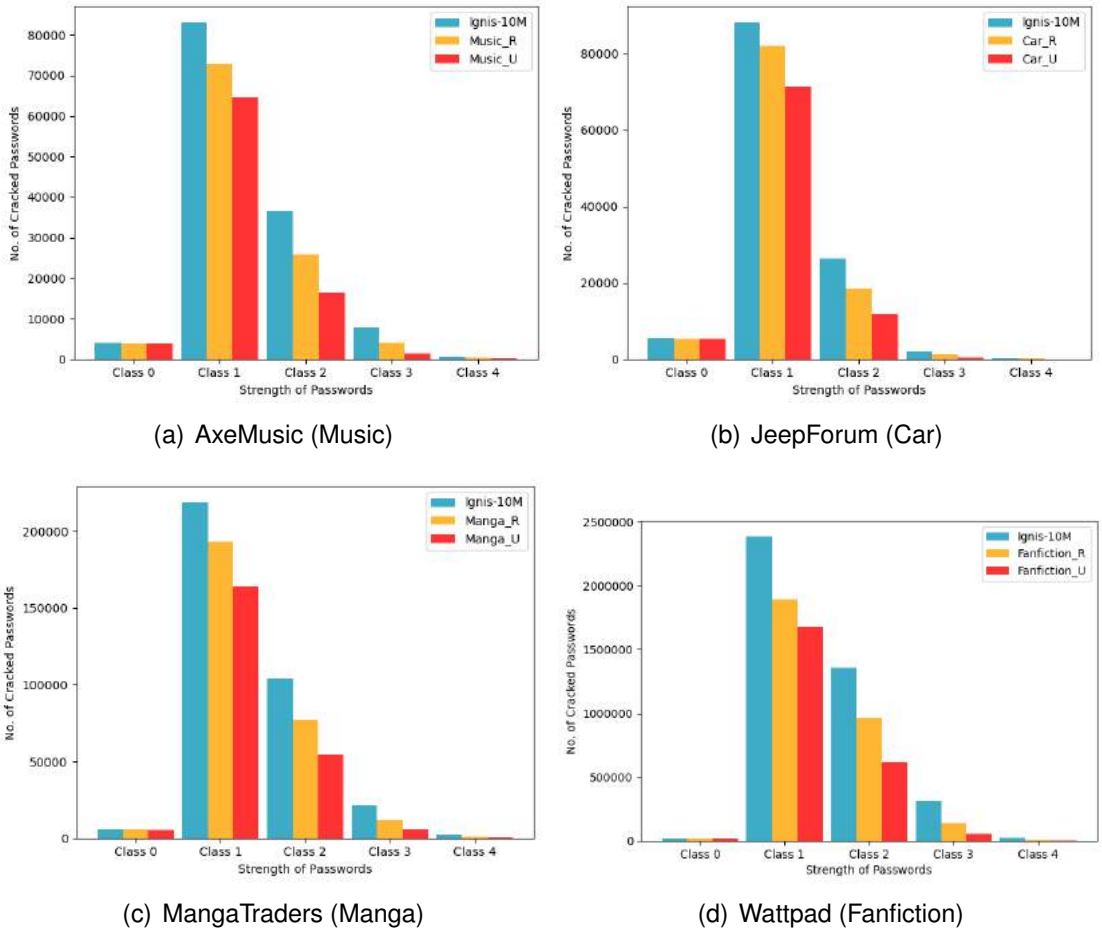
The number of passwords found exclusively by the contextual dictionaries leads to a new and interesting question. Which password candidates in the dictionary list performed better, i.e., which found the most passwords in their respective data leaks? Table 6.21 shows the top 20 password candidates that 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 [288].

6.6.2 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 determine this, the password strength meter zxcvbn was employed again. As mentioned previously, 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). This classification for each of the four data leaks with Ignis-10M and the ranked and unranked dictionaries are shown in Figure 6.14.



(a) AxeMusic (Music)

(b) JeepForum (Car)

(c) MangaTraders (Manga)

(d) Wattpad (Fanfiction)

Figure 6.14: Strength of passwords cracked by Ignis-10M and the rank and unranked versions of the contextual dictionaries

As can be seen in Figures 6.14(a) to 6.14(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 [4]. Therefore, the focus is mostly on the passwords belonging to Class 3 and Class 4.

Tables 6.22 and 6.23 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

Table 6.22: 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 6.23: 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

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 AxeMusic, JeepForum, Wattpad, and Manga Traders respectively.

6.7 Summary of Results

The experiments in this chapter aim to showcase the role of contextual information in password cracking and the merit of using the contextual approach for this purpose. The results of Section 6.2 show that there is context inherently in passwords. Even in a big and broad dataset as HIBP, there are trends that users follow when they choose their passwords that suggest the role of context and the importance of leveraging it for more targeted attacks.The results of Section 6.3 show a preliminary experiment where leaked lists of passwords from specific communities were used as dictionaries to crack passwords of other thematically close communities. These are preliminary results that show the improvement of using wordlists that are targeted towards the passwords(s) that are to be cracked, and encourage the idea of creating bespoke dictionary lists for this purpose. Sections 6.4 and 6.5 show the results of these bespoke dictionary lists, first with a smaller experiment and then with the one of a larger scale with ten different datasets from different communities used for evaluation. These results show the significant increase in cracked passwords with the contextual approach and especially the improvement over Class 3 and Class 4 passwords which are the hardest to crack. Finally, Section 6.6 shows the improvement of using NLP techniques to rank the bespoke dictionary lists by how close they are thematically to the password(s) to be cracked. Indeed, in this section some of the best numbers of cracked passwords in Class 3 and Class 4 were achieved. The number of passwords found exclusively by the ranked approach and not by Ignis constitute

an improvement of even 50% in some cases. The results of Table 6.21 solidify even more why this approach works - the password candidates that performed best were in majority directly related to the seed word.

Chapter 7

Discussion and Analysis

This chapter focused on presenting the results of the evaluation of the methodology outlined in Chapter 4, with the aim of answering the research questions posed at the start of this thesis about the role of context in password selection and the ways it can be leveraged on behalf of the password cracker. Section 7.3 looks back at the alternative approaches in password cracking dictionary generation and assesses where the contextual approach fits. The rest of this chapter explores these research questions sequentially and discusses how the results presented address each research question.

7.1 Reviewing the Research Questions

7.1.1 Research Question 1

RQ1: What impact does a context-based password cracking approach have on the likelihood of success during a digital investigation?

The analysis of the HIBP dataset decisively shows that clear trends of contextualisation can be found in passwords. As this analysis shows, users use passwords they easily remember, something that makes them weak and easier to guess. The

515 million reversed-engineered passwords from the HIBP dataset produced 3 times as many password fragments, which shows that there is merit in this approach and, in fact, a deeper analysis of the fragments is warranted. The new insights provided by password fragments can help inform not only password cracking but also on the other side of the equation, password policy creation.

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

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

Looking at the contextual information that can be found through the classification of the fragments, attention should be paid to how it can be translated to viable password candidates. Such information is often available through classical means of investigation in the case of law enforcement, and could tilt the balance in their favour. In the case of an attacker targeting an individual, this type of information may be found by unlawful means or in some cases by what the victims themselves have shared online. This is why it is especially prudent to be mindful of an attacker's targeted approach.

Taking the hunt to find contextual information in passwords a step further, the focus shifted to datasets stemming from specific communities such as Manga and video games instead of a bigger, more generic dataset like HIBP. Throughout this thesis, the community scenario, as described in Section 4.2.1, is used for the eval-

uation, along with RockYou (a generic dictionary from data breaches). Two other dictionaries, focusing on Manga and video games, have also been used during evaluation.

The question that arises from the preliminary results is, what do these two datasets have that RockYou does not? Both MangaFox and BoostBot stem from online leaks and there is no processing, augmentation or other customisation done to them. Furthermore, their size is small compared to the 14 million of RockYou passwords (32 million considering repetitions). The one advantage these datasets have, is that they are thematically closer to the target datasets.

Overall, the performances between the three wordlists are comparable when considering the JtR approach and close when considering OMEN, both of which are Markov-based models. The results are poor when it comes to PRINCE (for MangaFox and BoostBot) but a pre-processing of the wordlist to make a better usage of it could modify those results. PCFG works better than the other processes, but with a clear advantage for RockYou. This is probably thanks to the difference of size, giving more chances for PCFG to infer and reuse the grammar.

Both MangaFox and BoostBot have a better ratio of passwords found in Class 3 for Minecraft, and in a less impressive manner for Class 4 for MangaTraders. The found passwords are significantly fewer for Axemusic and JeepForum, probably due to a lesser proximity of the communities. Surprisingly, MangaFox performs better than BoostBot on Minecraft and BoostBot better on MangaTraders than MangaFox, while the other way around would have been expected. Still, the communities of Manga and video games are more closely associated with each other than Music and Cars, so this close proximity might be the explanation. Finally, while MangaFox performs poorly on Class 3 of JeepForum, it performs relatively well, even if the numbers are small, on Class 4.

These results showcase that even against a generic, much larger dataset like

RockYou, using a smaller dictionary list that is thematically closer to the topic of the community can yield comparable results and offers the first proof that context can indeed impact password cracking, and it can be leveraged for this reason.

7.1.2 Research Question 2

RQ2: How can a context-based password cracking dictionary be generated, bespoke to the interests of an individual suspect or a group of suspects?

Humans are creatures of habit. When choosing passwords, they tend to repeat words and patterns and select words that are familiar and meaningful to them. Their passwords naturally tend to make sense for them so that they can remember them more easily. Even in the case of users choosing random words, e.g., 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, therefore secure passwords, on their behalf. Therefore, neither typical dictionary attacks nor context-based approaches would prove effective against them.

Nevertheless, there is merit to the proposed targeted dictionary approach and the method that was adopted to create these dictionaries showed very positive results. The experiments with the generated contextual dictionaries demonstrate that, conclusively, 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 a need to act fast, it may prove more useful to use the proposed targeted approach first. The metrics that were introduced in 4.7 can be used alone or combined on a case-by-case scenario, in order to create the appropriate contextual dictionary list.

The insights provided by the experiments already point to the most successful

techniques that can be adopted, but it of course depends on the specifics of each case. For example, the poor performance of Sports_3 compared to the other nine L_3 contextual dictionaries suggests that more layers or a different seed word (one with a more detailed Wikipedia page, hence more links) should be considered to reach a sufficient starting size for the contextual dictionary. Another thing to be taken into consideration is the password cracking tool that is selected. For example, with tools like PCFG where the dataset is used for training to create the grammar, a generated dataset will have a disadvantage. One way to possibly improve the results could be to reuse the grammar trained from RockYou or Ignis-10M, but then feed that special list with the content of the generated dataset.

As mentioned, RockYou and Ignis-10M are significantly larger than any generated dictionary can be - unless the depth is chosen to be more than 5 or 6 layers, which is something that would defy the purpose of context as the dictionary would be too broad, and it would be too time-consuming to generate it. Therefore, the difference in size between RockYou and Ignis-10M and the generated dictionaries is that the first two will take longer to execute. A smaller, more focused, bespoke dictionary, which prioritizes the most likely password candidates first, might be the best option to choose in the first instance. The advantage of it, is that it will take less time to run it, if time is the metric of interest, or more permutations of the password candidates will be tested. The performance of the contextual generated dictionaries already has shown a clear benefit, in finding a significant number of strength 4 passwords that were not found by the generic approaches. If a strong password is the target, this approach should be considered.

Of course, if the aim is to crack more than one password, other factors need to be considered too, including how customisable the list should be. Is it better to start with one or more seed words? Is the number of passwords cracked enough to determine success, or is there a need for other, more sophisticated metrics, i.e.,

the number 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, all of which can be established using PCWQ. An advantage of this process is that the customisation is easily done, i.e., the addition of more seed words, the re-definition of size or attack duration, etc.

In a community-based approach, a bespoke, targeted dictionary can provide a significant increase in the number of found passwords and can be adopted ahead or alongside the baseline in the password cracking pipeline. Of course, one scenario that could not be tested as part of this thesis is the one of the single suspect and their seized encrypted device(s). In such a case, a bespoke dictionary whose parameters can be tweaked and tailored to the suspect can be created with ease using the proposed methodology and procedure as described in this thesis. This would result in the investigator easily having the means to produce a custom dictionary, or dictionaries, for a specific case. When racing against the clock or when an encrypted device presents the largest roadblock in an ongoing case, contextual dictionaries tailored to the suspect at hand could prove invaluable to progressing an investigation.

While instinctively the password cracking community felt that context matters in password cracking, this is the first time that this suspicion has been exploited to create bespoke, context-based dictionaries. This work is the first in literature that categorically proves that context matters in the password cracking process and opens up a whole host of further avenues for exploration, as discussed later in future work.

7.1.3 Research Question 3

RQ3: How can password candidates be contextually prioritised in a dictionary, and what impact does this prioritisation have?

The results of Section 6.6 show the added value of considering context in password cracking. The number of passwords found 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 any large community, a generic dictionary will always be at an advantage. This is due to a significant proportion of users choosing passwords that are not thematically close to the content of the website the password is for, and many will 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 6.21, 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.

7.2 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). But manually creating customised dictionaries for each case would be a very time-consuming process. To overcome this, having some established lists of commonly encountered topics and interests could result in an optimised start to the password guessing process. 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.

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 easy-to-guess passwords. This is where the contextual approach could make a difference, since the passwords that it cracked - that the generic approaches did not in the experiments - were majorly related to the seed word of the dictionary and focus of the community. Further than that, the performance of the contextual dictionaries with strength 4 passwords was impressive, considering that on average 50% of the strength 4 passwords found by the ranked dictionaries are not found using Ignis-10M. These two elements could be the decisive factor in the outcome of a case against a tech-savvy suspect.

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 pa-

rameter 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 combination approach with the technique described as part of this work will likely improve the chances of overall success further.

Furthermore, as already mentioned in Chapter 4, contextual dictionaries, unlike common password lists such as Ignis-10M and RockYou, are lists of words not lists of passwords. Therefore, the performance of the contextual dictionary cannot be judged against the generic approaches, as they do not function in the same way and their end goal is different.

The use of mangling rules can help remedy to an extent the fact that the contextual dictionaries are not human generated and therefore do not contain this important information. Still, 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 6.21.

Finally, this approach, as all dictionary attacks, would not work on suspects that use password managers or randomise their passwords in a way that does not include dictionary words. These types of passwords would not easily succumb to any of the password cracking techniques that are available at this time.

7.3 Comparison with Alternative Approaches

When discussing the results of this work, the position where this contextual approach fits in the grand scheme of password cracking methods and tools must be considered. Unlike the artificial intelligence approaches that require large lists of real-world passwords from data breaches, the proposed contextual approach does not require any data from breaches. As the contextual dictionaries are generated based on the seed word(s), as long as there is a corresponding Wikipedia entry, this practically means a dictionary list can be generated for any topic.

This is something novel and something that is not achievable by any other method currently available. The current state-of-the-art methods, be it rule-based, artificial intelligence-based, or those based on Markov/statistical models focus on what a typical password looks like. These methods use real human generated passwords as training data for models to create new password candidates that *look* like them, i.e., that contain similar components and patterns as real passwords. Of course, this is a valid approach to generating dictionaries, but the key difference between these and the contextual-based approach is that the latter considers not what the password looks like but its semantic meaning. The context-based approach can generate dictionary lists on topics of interest to a suspect or group of suspects, or indeed on topics in the future that do not currently exist, e.g., a new TV series, new books, new movies, etc.

There have been several cases where researchers have hinted at the role of context in password selection in the past. Veras et al. [197] classified the RockYou dictionary entries with the help of Wordnet and used PCFG to create probabilities for these semantic patterns, as well as the structural ones of classic PCFG. Li et al. [204] also enriched PCFG by adding a few more categories besides letters, digits and symbols such as birthdate or email patterns. But the use of context in these cases remains focused on semantic word sequences or looking for some very com-

mon patterns, e.g., the inclusion of the email address in the password. The methodology presented as part of this thesis is the first work that explores how to crack passwords exploiting context in a targeted approach and by creating new, bespoke dictionaries for each case.

Looking at some of the results in the previous chapter, where the contextual approach is pitted against RockYou or Ignis-10M, this approach could work in combination or, in certain cases, before existing approaches – especially when the triage scenario is considered. In general, the combination approach proves greater than the sum of its parts, mainly because of the success of the contextual dictionary method in generating passwords that the other approaches cannot.

Dictionary lists have been the most important tool for a password cracker, who then, according to the problem at hand, utilise them with different password cracking tools and methods. Up to today, for the conventional (but also best performing) approaches, a dictionary list is a prerequisite, but these approaches focus on what to do with a dictionary list after they obtain it, and data leak password lists are commonly used. In many cases, the dictionary lists in use are a combination from different data leaks, sorted by order of popularity, an example of which is Ignis. These lists are then fed to the tool of choice, be it a PCFG that uses the dictionary list to calculate probabilities or a GAN that uses it to create new password candidates. The clear improvement of the contextual approach is that it allows for bespoke dictionary lists that can take the place of (or even be combined with) existing lists from data leaks, allowing for complete control and targeted password cracking that can be tailored to each specific case.

Machine learning approaches and GANs aim to use existing data leaks as input to generate new similar password candidate strings that look like that input – most likely motivated by trying to reduce the amount of password candidate mangling needed. One drawback of these methods is that, in general, they require a greater

number of guesses to match the results of rule generated approaches. For example, in order to achieve a similar number of cracked passwords as the other tools that were tested, PassGAN needed significantly more password guesses, and in some cases this was an order of magnitude more [289]. This would not be an issue in an offline attack where time is not of the essence, but it could be prohibitive in a triage scenario. Furthermore, artificial intelligence-based approaches require an input dictionary (which in the majority of cases to date, RockYou from 2009 is still used) so they could still benefit from contextual dictionaries as input – something that will be further expanded upon in the future work outlined in Section 8.2.

The performance of the proposed methodology should also be considered. Machine learning-based approaches are mostly candidate generation tools that must be piped to a password cracker software, such as Hashcat or JtR, to conduct a password search. The pace at which the passwords must to be generated should be high enough to feed the cracker – otherwise the generation becomes the bottleneck of the password cracking process. This is directly linked to the targeted hash function and how fast the cracker can evaluate passwords for this function. As the crackers nowadays are well optimised and the available computing resources much larger, these ML-based generation tools often have trouble reaching a sufficiently fast pace. The generation could be sustained by the use of mangling rules on the cracking side, but it is somehow going against the idea of ML methods, of generating *better* candidates in the first place without requiring any additional step. The methodology described in this manuscript produces a dictionary that is meant to be used directly as input of the cracker, together with mangling rules. This attack, together with exhaustive search, is the fastest combination for maximising the usage of the available hardware. While it would not make a difference for slow hash functions, e.g., Veracrypt or scrypt, it can be a game changer for both medium-speed, e.g, SHA-256 or RIPEMD-160, or fast hash functions, e.g., MD5 or NTLM.

Chapter 8

Conclusion & Future Work

8.1 Conclusion

Despite known security concerns, password-based authentication remains the most widely used method of authentication. A 2021 study showed that the online identity of almost one in three Americans was stolen in the last year alone, and another 13% were uncertain whether their credentials were part of a data breach [290]. In a spirit of strengthening security, password policies are nowadays more restrictive and require users to select stronger passwords. Additionally, salting the passwords increases the complexity of the password cracking process, as each salt must be considered sequentially. Salting renders the commonly used rainbow table-based password cracking approaches obsolete.

This thesis firstly looks at the role of context in existing real-world, human-generated passwords, stemming from data breaches. The insights gained from studying lists like HIBP, which consists of various data breaches of the last few years, are important from both an offensive and a defensive perspective. From an offensive perspective because they can inform password cracking attacks and give LEA the necessary knowledge to create tailored attacks, something that could be the deciding factor in an investigation. From a defensive perspective, this knowledge

that is extracted from the statistical analysis and the fragment analysis of the constituent pieces of the passwords can benefit password policies in order to ensure the safety of the passwords that users choose.

The PCWQ framework that is developed as part of this thesis provides a new methodology to assess and compare wordlists. It highlights that wordlists behave differently depending on the context of the target dataset, and it can therefore be used to develop and assess wordlist generation processes in several scenarios. Focusing on the different classes of strength is also useful to evaluate the quality of wordlists to retrieve stronger passwords.

Evaluating a dictionary list is a complex topic and there are many parameters to take into consideration. 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.

The evaluation of wordlists with the framework highlighted that the size and the composition of the wordlists have a strong impact on some processes, e.g., PRINCE and PCFG, while it is less visible for some other processes. Therefore, when a generated wordlist is considered, dedicated pre-processing is needed to better prepare the wordlists according to which password cracking tool has been selected. There-

fore, it is clear that not one metric can stand alone, evaluate a wordlist thoroughly and assign a score that can predict how well that wordlist will do against a target. A compound metric is needed for the evaluation, and even then, there should be room left for its parameterisation for each attack scenario.

The primary contribution described in this thesis is a novel framework for creating new, custom dictionary lists for any topic of interest that may be required, ones that can be useful to a digital investigator for cracking the password of an encrypted device. This methodology leverages natural language processing and the power of structured information found on Wikipedia (and DBPedia) to create bespoke dictionary lists. This can provide the blueprint for easily creating customized dictionary lists for any topic, combine them, tailor them according to how deep and comprehensive they need to be, and personalize them to the needs of each investigation.

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.

The experiments conducted in this thesis provide a definite proof of the value of considering contextual information in password cracking. Humans are creatures of habit, and that is no different in their password selection process – where they often choose familiar words that are more easily remembered. This information can be leveraged in an investigation, and the ability to exploit it could prove invaluable during an investigation.

The experiments have demonstrated that often people choose passwords related to the topic of the website/system that the password is for, or that are thematically close to that topic. Therefore, using a custom dictionary list can offer a significant advantage to the cracking process and ultimately result in higher success rates compared to using a generic dictionary alone. The conducted exper-

iments show that the use of the proposed approach, in conjunction with existing approaches, results in up to 50% additional passwords being cracked over existing approaches alone in certain circumstances when considering the final experiments with the ranked, optimised contextual dictionary lists. 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.

Of course, a contextual dictionary based around a single seed word cannot compete on equal footing with a 20 to 300 times larger and more well-rounded dictionary like Ignis-10M when the objective is to crack as many passwords as possible. This means that when no information is known about the target or the goal is to gain access to a system by cracking the password of any user and not a specific one, using a dictionary like Ignis-10M would provide a higher chance of success.

If the usage scenario surrounds a single case and/or a single password and information can be determined about its owner and their interested, then the contextual approach can be utilised. This would make even more sense, considering that in digital cases, suspects might be more likely to *try harder* to conceal their tracks and therefore would choose their password with more prudence.

The most notable improvement when it comes to the results of the contextual dictionaries is the number of extra passwords cracked with the contextual dictionaries, which offer a significant improvement when combined with the generic approach. The extra passwords that were cracked not only lend credit to a combination approach, but also showcase further that a smaller dictionary built around one seed word related to the target data leak can indeed boost the number of cracked passwords significantly.

8.1.1 Implications of This Work

The findings of this work have the following implications in the domain of password cracking and the protection of user authentication.

Role of the Contextual Dictionary Approach in the Digital Forensic Investigation Process

The contextual approach, as demonstrated by the results of the experiments in Chapter 6 can, depending on the case, stand alone, but performs best in conjunction with other more generic approaches. For the experiments conducted in this thesis, it was not possible to test the individual scenario where information about a specific target could be accumulated and used to create a bespoke dictionary, tailored to them. This is something that can easily be done by LEA and certainly one avenue that can be explored to gain insight of how well the contextual approach can perform in real digital investigation cases.

Investigators do not have to know anything about the desired topic to be able to build a custom dictionary list of the most important words about that topic. Additionally, this dictionary generation utility helps investigators keep up with current trends in password cracking and easily create new dictionary lists to accommodate them.

This approach can be further enhanced with knowledge of previous passwords that can provide insight into the ways the particular suspect picks their password as well as any other information about them that could be useful, e.g., names and dates of births of relatives and friends, information found on their social media accounts about their likes and interests and who they correspond with. These could be used to further tailor the contextual dictionary to the suspect. 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.

Finally, the contextual approach (alone or combined with other password cracking attacks) would fit in triage in a time-critical case, when faced with an encrypted device. Depending on the parameters of the case, the contextual dictionary could be run first if for example the suspect is technologically savvy or second if the investigator would like to first eliminate easy passwords with a brute force attack. In a non-triage situation, the contextual approach can be part of the investigator's arsenal if other, more conventional methods fail.

Password Cracking

As outlined in Chapter 6, the use of these contextual dictionaries can in some cases offer competing results with much larger and variant dictionary lists, such as Ignis-10M, although this is not their purpose. Contextual dictionaries can offer a significant increase in found passwords if the generic and contextual approach are combined with them. 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.

Informing Password Policies

On the other side of the coin, the information extracted during the course of this thesis that pinpoint the role of context in password creation can also be used to inform password policies. Taking the fact that humans contextualise their passwords to make them more memorable should be taken into account by password policies. That's not to say password policies should prevent it, because a balance between safety and memorability must be achieved.

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

8.2 Future Work

Based on the findings of this work, several future directions can be identified. They are described in the following subsections in no particular order.

Refinement of the Contextual Dictionaries

A crucial step in the refinement of the contextual dictionaries is to fine-tune the sanitisation process for the dictionary words. For example, a link could contain more than one word. Therefore, tweaking the manner in which these are combined could result in better password candidates. In the experiments performed as part of this thesis, even though 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”, “Manof-Steel” 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.

In this vein, the trimming of some branches during the dictionary generation process can reinvigorate the progress when relevancy declines. For example, during the exploration of layer 3, some candidates might be already too thematically distant from the seed word. If these candidates can be disregarded at this stage, they could give way to an exploration of deeper relevant layers rather than shallower irrelevant entries. As a result, the end dictionary may be the same length as layer 3, but would contain more relevant entries.

Enhancement of the Contextual Dictionaries

As part of future work, more consideration will be given into enhancing the generated dictionaries. While dictionaries of 3 layers provided the best results in 9 out of 10 experiments, it was not the case for the 10th. In a scenario where the bespoke dictionary is too short, additional seed words could be provided, and the resultant dictionaries could be merged, additional layers could be used, or a combination thereof. Moreover, these dictionaries are exactly that – dictionaries, whereas Ignis-10M contains real-world passwords. It is therefore important to look into ways of transforming the dictionaries into password candidates, with the help of well refined mangling rules, that could better imitate the behaviour of users when they choose their password.

Finally, two other avenues to be explored for optimising the results of the candidate generation process are searching backwards during the link exploration process and ranking the dictionary entries. For example, in the case that the seed

word is “Manga”, backwards searching would include Layer -1, i.e., all the pages on Wikipedia that link *to* Manga. This could provide a substantial addition of very relevant password candidates. As part of future work, a system for keeping track of how many times each entry was found could help indicate how relevant it is to the seed word and be used for ranking the resultant list.

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”.

New Sources of Information

Wikipedia (and DBpedia) provide a great way to assemble a contextual dictionary as the database of entries on Wikipedia is immense and covers a vast amount of topics, so much so that it would be very difficult not to be able to find a starting article to use as a seed word for almost any topic. Furthermore, the tree like structure of Wikipedia ensures that topics and concepts that are semantically close to that starting seed will be featured in the contextual dictionary. But, not everything is included on Wikipedia, in fact there are many cases where information that might be relevant about a topic is not included there. Usually, this type of information would include current trends, idioms and pop culture.

Fortunately, there are many avenues to explore to further enriching the process of creating context-based dictionaries. Other 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 insight into the slang, colloquialisms, and common words or phrases associated with each keyword.

Adding Context to Other Dictionary Generation Approaches

The combination of current approaches with the contextual dictionary approach can be another avenue to explore in future work. Current state-of-the-art approaches to password cracking, whether they are rule-based, probabilistic or machine learning-based, they require the use of a dictionary at some point, either a leaked list they feed into a password cracking tool or as input for training to create a new dictionary list based on it. The go-to approach to this is the use of leaked dictionary lists from data breaches. In fact, RockYou remains one of the most popular lists for this purpose even today.

Combining the bespoke, contextual approach as input for training in some of the state-of-the-art AI and ML approaches could serve to create new password candidates that could not be recreated using the traditional leaked list approach.

Bibliography

[1] Aikaterini Kanta, Iwen Coisel, and Mark Scanlon. A Comprehensive Evaluation on the Benefits of Context Based Wordlists for Password Cracking. *Journal of Information Security and Applications*, 2023. ISSN 2214-2134. Under Review.

[2] Aikaterini Kanta, Iwen Coisel, and Mark Scanlon. Harder, Better, Faster, Stronger: Optimising the Performance of Context-Based Password Cracking Dictionaries. *Forensic Science International: Digital Investigation*, 2023. ISSN 2666-2817.

[3] Aikaterini Kanta, Iwen Coisel, and Mark Scanlon. A novel dictionary generation methodology for contextual-based password cracking. *IEEE Access*, 10: 59178–59188, 2022. doi: 10.1109/ACCESS.2022.3179701.

[4] Aikaterini Kanta, Sein Coray, Iwen Coisel, and Mark 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.

[5] Aikaterini Kanta, Iwen Coisel, and Mark Scanlon. A Survey Exploring Open Source Intelligence for Smarter Password Cracking. *Forensic Science International: Digital Investigation*, 35:301075, 2020. ISSN 2666-2817. doi: https://doi.org/10.1016/j.fsidi.2020.301075.

[6] Aikaterini Kanta, Iwen Coisel, and Mark Scanlon. PCWQ: A Framework for Evaluating Password Cracking Wordlist Quality. In Pavel Gladyshev, Sanjay Goel, Joshua James, George Markowsky, and Daryl Johnson, editors, *Digital Forensics and Cyber Crime*, pages 159–175, Cham, 2022. Springer International Publishing. ISBN 978-3-031-06365-7.

[7] Aikaterini Kanta, Iwen Coisel, and Mark Scanlon. Smarter Password Guessing Techniques Leveraging Contextual Information and OSINT. In *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, pages 1–2, 2020.

[8] Marcus K Rogers, James Goldman, Rick Mislan, Timothy Wedge, and Steve Debrota. Computer forensics field triage process model. *Journal of Digital Forensics, Security and Law*, 1(2):2, 2006.

[9] Analyzing Password-Strength Meters - Password Multi-Checker Tool. `https://madiba.encs.concordia.ca/software/passwordchecker/index.php`. [Online; accessed 11-March-2022].

[10] Top 200 Most Common Passwords. `https://nordpass.com/most-common-passwords-list/`, 2022. [Online; accessed 4-December-2022].

[11] Rule Based Attack. `https://hashcat.net/wiki/doku.php?id=rule_based_attack/`. [Online; accessed 22-July-2022].

[12] Matt Weir, Sudhir Aggarwal, Michael Collins, and Henry Stern. Testing Metrics for Password Creation Policies by Attacking Large Sets of Revealed Passwords. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, page 162–175, New York, NY, USA, 2010.

Association for Computing Machinery. ISBN 9781450302456. doi: 10.1145/1866307.1866327. URL https://doi.org/10.1145/1866307.1866327.

[13] Xiaoyu Du, Chris Hargreaves, John Sheppard, Felix Anda, Asanka Sayakkara, Nhien-An Le-Khac, and Mark 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. ISBN 9781450388337. doi: 10.1145/3407023.3407068.

[14] Shelia M. Kennison and D. Eric Chan-Tin. Predicting the adoption of password managers: A tale of two samples. *Technology, Mind, and Behavior*, 11 2021.

[15] Asanka Sayakkara, Nhien-An Le-Khac, and Mark Scanlon. Electromagnetic side-channel attacks: Potential for progressing hindered digital forensic analysis. In *Companion Proceedings for the ISSTA/ECOOP 2018 Workshops*, ISSTA '18, page 138–143, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450359399. doi: 10.1145/3236454.3236512. URL https://doi.org/10.1145/3236454.3236512.

[16] Vrizlynn LL Thing and Hwei-Ming Ying. A novel time-memory trade-off method for password recovery. *Digital Investigation*, 6:S114–S120, 2009.

[17] David Lillis, Brett Becker, Tadhg O'Sullivan, and Mark Scanlon. Current Challenges and Future Research Areas for Digital Forensic Investigation. In *The 11th ADFSL Conference on Digital Forensics, Security and Law (CDFSL 2016)*, pages 9–20, Daytona Beach, FL, USA, 05 2016. ADFSL.

[18] Asanka Sayakkara, Nhien-An Le-Khac, and Mark Scanlon. A Survey of Electromagnetic Side-channel Attacks and Discussion on their Case-progressing

Potential for Digital Forensics. *Digital Investigation*, 29:43 – 54, 2019. ISSN 1742-2876. doi: https://doi.org/10.1016/j.diin.2019.03.002.

[19] Europol. Internet organised crime threat assessment (iocta) 2021. Technical report, Publications Office of the European Union, Luxembourg, 2021.

[20] Steven Ryder and Nhien-An Le-Khac. 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)*, pages 904–907, 2016. doi: 10.1109/CLOUD.2016.0133.

[21] Ding Wang, Haibo Cheng, Ping Wang, Xinyi Huang, and Gaopeng Jian. Zipf's law in passwords. *IEEE Transactions on Information Forensics and Security*, 12(11):2776–2791, 2017. doi: 10.1109/TIFS.2017.2721359.

[22] Ding Wang, Zijian Zhang, Ping Wang, Jeff Yan, and Xinyi Huang. Targeted online password guessing: An underestimated threat. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 1242–1254, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341394. doi: 10.1145/2976749.2978339. URL https://doi.org/10.1145/2976749.2978339.

[23] Viktor Taneski, Marjan Heričko, and Boštjan Brumen. Systematic overview of password security problems. *Acta Polytechnica Hungarica*, 16(3), 2019.

[24] Cynthia Kuo, Sasha Romanosky, and Lorrie Faith Cranor. Human selection of mnemonic phrase-based passwords. In *Proceedings of the Second Symposium on Usable Privacy and Security*, pages 67–78, 2006.

[25] Alyssa Newcomb. FBI Most Wanted Hacker Jeremy Hammond Used His Cat's Name for Password. https://abcnews.go.

com/Technology/fbi-wanted-hacker-jeremy-hammond-cats-password/story?id=26884738, 2014. [Online; accessed 3-January-2021].

[26] Gongzhu Hu. On Password Strength: A Survey and Analysis. In *International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pages 165–186. Springer, 2017.

[27] Aaron L-F Han, Derek F Wong, and Lidia S Chao. Password cracking and countermeasures in computer security: A survey. *arXiv preprint arXiv:1411.7803*, 2014.

[28] Xiaoyu Du, Nhien-An Le-Khac, and Mark Scanlon. Evaluation of Digital Forensic Process Models with Respect to Digital Forensics as a Service. In *Proceedings of the 16th European Conference on Cyber Warfare and Security (ECCWS 2017)*, pages 573–581, Dublin, Ireland, 06 2017. ACPI.

[29] Janet Williams. *ACPO Good Practice Guide for Digital Evidence*. Association of Chief Police Officers of England, Wales & Northern Ireland, 2012.

[30] M Al Fahdi, Nathan L Clarke, and Steven M Furnell. Challenges to digital forensics: A survey of researchers & practitioners attitudes and opinions. In *Information Security for South Africa*, pages 1–8. IEEE, 2013.

[31] Vikram S Harichandran, Frank Breitinger, Ibrahim Baggili, and Andrew Marrington. A cyber forensics needs analysis survey: Revisiting the domain's needs a decade later. *Computers & Security*, 57:1–13, 2016.

[32] Nickson M Karie and Hein S Venter. Taxonomy of challenges for digital forensics. *Journal of Forensic Sciences*, 60(4):885–893, 2015.

[33] Darren Quick and Kim-Kwang Raymond Choo. Impacts of increasing volume

of digital forensic data: A survey and future research challenges. *Digital Investigation*, 11(4):273–294, 2014.

[34] Bardia Safaei, Amir Mahdi Hosseini Monazzah, Milad Barzegar Bafroei, and Alireza Ejlali. Reliability side-effects in internet of things application layer protocols. In *2nd International Conference on System Reliability and Safety (ICSRS)*, pages 207–212. IEEE, 2017.

[35] Mark Scanlon. Battling the digital forensic backlog through data deduplication. In *2016 Sixth International Conference on Innovative Computing Technology (INTECH)*, pages 10–14. IEEE, 2016.

[36] Mukrimah Nawir, Amiza Amir, Naimah Yaakob, and Ong Bi Lynn. Internet of things (iot): Taxonomy of security attacks. In *2016 3rd International Conference on Electronic Design (ICED)*, pages 321–326. IEEE, 2016.

[37] Tina Wu, Frank Breitinger, and Ibrahim Baggili. IoT Ignorance is Digital Forensics Research Bliss: A Survey to Understand IoT Forensics Definitions, Challenges and Future Research Directions. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pages 1–15, 2019.

[38] Keyun Ruan, Joe Carthy, Tahar Kechadi, and Mark Crosbie. Cloud forensics: An overview. In *proceedings of the 7th IFIP International Conference on Digital Forensics*, pages 16–25, 2011.

[39] Keyun Ruan, Joe Carthy, Tahar Kechadi, and Ibrahim Baggili. Cloud forensics definitions and critical criteria for cloud forensic capability: An overview of survey results. *Digital Investigation*, 10(1):34–43, 2013.

[40] Bharat Manral, Gaurav Somani, Kim-Kwang Raymond Choo, Mauro Conti, and Manoj Singh Gaur. A systematic survey on cloud forensics challenges,

solutions, and future directions. *ACM Computing Surveys (CSUR)*, 52(6):1–38, 2019.

[41] Vincent Liu and Francis Brown. Bleeding-edge anti-forensics. *Presentation at InfoSec World*, 2006.

[42] Owen Bowcott. Police mishandling digital evidence, forensic experts warn, 2018. URL `https://www.theguardian.com/law/2018/may/15/police-mishandling-digital-evidence-forensic-experts-warn`.

[43] House of Commons Justice Committee. Disclosure of evidence in criminal cases, 2018.

[44] James Plunkett, Nhien-An Le-Khac, and Tahar Kechadi. Digital forensic investigations in the cloud: A proposed approach for irish law enforcement. Technical report, Science Foundation Ireland, 2015.

[45] Elizabeth Dinan. Police: Defendant hindering investigation by keeping mum on cell password. `https://eu.seacoastonline.com/story/news/local/portsmouth-herald/2012/12/13/police-defendant-hindering-investigation-by/49222895007/`, 2012. [Online; accessed 4-September-2022].

[46] Jim Edwards. Man Gets Five Years In Prison For Refusing To Reveal Uncrackable Password To British Police. `https://eu.rrstar.com/story/special/2014/01/15/man-gets-five-years-in/40826769007/`, 2014. [Online; accessed 23-April-2022].

[47] Kim Zetter. Apple's FBI Battle Is Complicated. Here's What's Really Going On. `https://www.wired.com/2016/02/apples-fbi-battle-is-complicated-heres-whats-really-going-on/`, 2016. [Online; accessed 23-April-2022].

[48] Morten Bay. The ethics of unbreakable encryption: Rawlsian privacy and the san bernardino iphone. *First Monday*, 22(2), Jan. 2017. doi: 10.5210/fm.v22i2.7006. URL https://journals.uic.edu/ojs/index.php/fm/article/view/7006.

[49] J. Riley Atwood. The encryption problem: Why the courts and technology are creating a mess for law enforcement. *Saint Louis University Public Law Review*, 34(2), 2015.

[50] John Negroponte. Intelligence community directive. Technical report, Office of the Director of National Intelligence, 07 2006.

[51] Matthew Kott. British intelligence and Hitler's empire in the Soviet Union, 1941–1945. *Journal of Baltic Studies*, 49(2):268–271, 2018. doi: 10.1080/01629778.2018.1469843. URL https://doi.org/10.1080/01629778.2018.1469843.

[52] Stephen C Mercado. Fbis against the axis, 1941-1945. *Studies in Intelligence*, 11:33–43, 2001.

[53] Stephen C Mercado. Sailing the sea of osint in the information age. *Secret intelligence: A reader*, 78, 2009.

[54] Nihad A. Hassan and Rami Hijazi. *The Evolution of Open Source Intelligence*, pages 1–20. Apress, Berkeley, CA, 2018. ISBN 978-1-4842-3213-2. doi: 10.1007/978-1-4842-3213-2_1. URL https://doi.org/10.1007/978-1-4842-3213-2_1.

[55] B.G. Thompson. *Giving a Voice to Open Source Stakeholders: A Survey of State, Local, and Tribal Law Enforcement: Congressional Report*. DIANE Publishing Company, 2010. ISBN 9781437918694. URL https://books.google.it/books?id=3u3K86aXuo4C.

[56] Danny Bradbury. In plain view: open source intelligence. *Computer Fraud & Security*, 2011(4):5–9, 2011.

[57] Max Roser, Hannah Ritchie, and Esteban Ortiz-Ospina. Internet. *Our World in Data*, 2019. https://ourworldindata.org/internet.

[58] Vernon Turner, John F Gantz, David Reinsel, and Stephen Minton. The digital universe of opportunities: Rich data and the increasing value of the internet of things. *IDC Analyze the Future*, 16, 2014.

[59] Cody Burke. *Freeing knowledge, telling secrets: Open sourceintelligence and development*. Number 13 in Research paper series: Centre for East-West Cultural & Economic Studies. Bond University, 5 2007.

[60] Stevyn Gibson. Open source intelligence. *The RUSI Journal*, 149(1):16–22, 2004. doi: 10.1080/03071840408522977. URL https://doi.org/10.1080/03071840408522977.

[61] Arthur S. Hulnick. The downside of open source intelligence. *International Journal of Intelligence and Counter Intelligence*, 15(4):565–579, 2002. doi: 10.1080/08850600290101767. URL https://doi.org/10.1080/08850600290101767.

[62] Bowman H. Miller. Open Source Intelligence (OSINT): An Oxymoron? *International Journal of Intelligence and CounterIntelligence*, 31(4):702–719, 2018. doi: 10.1080/08850607.2018.1492826. URL https://doi.org/10.1080/08850607.2018.1492826.

[63] NATO. NATO glossary of terms and definitions. *NATO Standardisation Agency*, 2003.

[64] John Sano. The changing shape of HUMINT. *Intelligencer Journal*, 21(3): 77–80, 2015.

[65] Francois Mouton, Louise Leenen, Mercia M Malan, and HS Venter. Towards an ontological model defining the social engineering domain. In *IFIP International Conference on Human Choice and Computers*, pages 266–279. Springer, 2014.

[66] Joseph M Hatfield. Social engineering in cybersecurity: The evolution of a concept. *Computers & Security*, 73:102–113, 2018.

[67] Katharina Krombholz, Heidelinde Hobel, Markus Huber, and Edgar Weippl. Advanced social engineering attacks. *Journal of Information Security and applications*, 22:113–122, 2015.

[68] Laura K Donohue. The dawn of social intelligence (SOCINT). *Drake Law Review*, 63:1061, 2015.

[69] Adrian Liviu Ivan, Claudia Anamaria Iov, Raluca Codruta Lutai, and Marius Nicolae Grad. Social media intelligence: opportunities and limitations. *CES Working Papers*, 7(2A):505, 2015.

[70] Habibul Haque Khondker. Role of the new media in the arab spring. *Globalizations*, 8(5):675–679, 2011. doi: $10.1080/14747731.2011.621287$. URL `https://doi.org/10.1080/14747731.2011.621287`.

[71] Leonardo Nizzoli, Marco Avvenuti, Stefano Cresci, and Maurizio Tesconi. Extremist propaganda tweet classification with deep learning in realistic scenarios. In *Proceedings of the 10th ACM Conference on Web Science*, WebSci '19, page 203–204, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362023. doi: $10.1145/3292522.3326050$. URL `https://doi.org/10.1145/3292522.3326050`.

[72] Swati Agarwal and Ashish Sureka. Applying social media intelligence for predicting and identifying on-line radicalization and civil unrest oriented threats. *arXiv preprint arXiv:1511.06858*, 2015.

[73] Mark Daniel Jaeger and Myriam Dunn Cavelty. From madness to wisdom: intelligence and the digital crowd. *Intelligence and National Security*, 34(3): 329–343, 2019. doi: $10.1080/02684527.2019.1553375$. URL `https://doi.org/10.1080/02684527.2019.1553375`.

[74] Jeff Howe. The rise of crowdsourcing. *Wired*, 14, 01 2006.

[75] Ricardo Buettner. A systematic literature review of crowdsourcing research from a human resource management perspective. In *2015 48th Hawaii International Conference on System Sciences*, pages 4609–4618. IEEE, 2015.

[76] Zheng Xu, Yunhuai Liu, Neil Yen, Lin Mei, Xiangfeng Luo, Xiao Wei, and Chuanping Hu. Crowdsourcing based description of urban emergency events using social media big data. *IEEE Transactions on Cloud Computing*, 2016.

[77] Daniel Trottier. Crowdsourcing CCTV surveillance on the Internet. *Information, Communication & Society*, 17(5):609–626, 2014. doi: $10.1080/1369118X.2013.808359$. URL `https://doi.org/10.1080/1369118X.2013.808359`.

[78] Giulia Berlusconi, Francesco Calderoni, Nicola Parolini, Marco Verani, and Carlo Piccardi. Link prediction in criminal networks: A tool for criminal intelligence analysis. *PloS One*, 11(4):1–21, 2016.

[79] Renee C Van der Hulst. Introduction to Social Network Analysis (SNA) as an investigative tool. *Trends in Organized Crime*, 12(2):101–121, 2009.

[80] Alice E. Marwick and Danah Boyd. I tweet honestly, i tweet passionately: Twitter users, context collapse, and the imagined audience. *New Media & Society*, 13(1):114–133, 2011. doi: 10.1177/1461444810365313. URL https://doi.org/10.1177/1461444810365313.

[81] Jamie Bartlett, Carl Miller, Jeremy Crump, and Lynne Middleton. *Policing in an information age*. Demos London, 2013.

[82] Clive Norris and Gary Armstrong. Cctv and the social structuring of surveillance. *Crime Prevention Studies*, 10(1):157–178, 1999.

[83] Daniel Trottier. Open source intelligence, social media and law enforcement: Visions, constraints and critiques. *European Journal of Cultural Studies*, 18 (4-5):530–547, 2015. doi: 10.1177/1367549415577396. URL https://doi.org/10.1177/1367549415577396.

[84] Andrew Staniforth. *Police Use of Open Source Intelligence: The Longer Arm of Law*, pages 21–31. Springer International Publishing, Cham, 2016. ISBN 978-3-319-47671-1. doi: 10.1007/978-3-319-47671-1_3. URL https://doi.org/10.1007/978-3-319-47671-1_3.

[85] Kathryn C Seigfried-Spellar and Sean C Leshney. The intersection between social media, crime, and digital forensics: #WhoDunIt? In *Digital forensics*, pages 59–67. Elsevier, 2016.

[86] Johnny Nhan, Laura Huey, and Ryan Broll. Digilantism: An analysis of crowdsourcing and the boston marathon bombings. *The British journal of criminology*, 57(2):341–361, 2017.

[87] Neal Ungerleider. How Reddit Became A Hub Of The Crowdsourced Boston Marathon Bombing Investigation. https://www.fastcompany.com/3008466/

how-reddit-became-hub-crowdsourced-boston-marathon-bombing-investigation, 2017. Accessed: 2019-10-08.

[88] Mike Cunningham. Law Enforcement Social Media Investigations. https://crimecenter.com/leverage-manage-crowdsourced-leads/, 2018. Accessed: 2019-10-08.

[89] Stop Child Abuse – Trace an Object. https://www.europol.europa.eu/stopchildabuse, 2022. [Online; accessed 4-August-2022].

[90] https://traffickcam.com/, 2015. [Online; accessed 22-July-2022].

[91] EUROPOL. With your help we are 21,000 steps closer to saving a child from sexual abuse, 2018. URL https://www.europol.europa.eu/newsroom/news/your-help-we-are-21-000-steps-closer-to-saving-child-sexual-abuse.

[92] Bas Testerink, Daphne Odekerken, and Floris Bex. AI-Assisted Message Processing for the Netherlands National Police. In Luthe Karl Branting, editor, *Proceedings of the Workshop on Artificial Intelligence and the Administrative State co-located with 17th International Conference on AI and Law (ICAIL 2019), Montreal, QC, Canada, June 17, 2019*, volume 2471 of *CEUR Workshop Proceedings*, pages 10–13. CEUR-WS.org, 2019. URL http://ceur-ws.org/Vol-2471/paper2.pdf.

[93] Darren Quick and Kim-Kwang Raymond Choo. Digital forensic intelligence: Data subsets and Open Source Intelligence (DFINT+ OSINT): A timely and cohesive mix. *Future Generation Computer Systems*, 78:558–567, 2018.

[94] Edmond Major III. Awesome OSINT. https://github.com/jivoi/awesome-osint, 2022. [Online; accessed 24-October-2022].

[95] Justin Nordine. OSINT Framework. `https://osintframework.com/`. [Online; accessed 11-December-2020].

[96] Sara Morrison. The police want your phone data. here's what they can get – and what they can't., 2020. URL `https://www.vox.com/recode/2020/2/24/21133600/police-fbi-phone-password-rights`.

[97] Tony Cook, Steve Hibbitt, and Mick Hill. *Blackstone's Crime Investigator's Handbook*. Oxford University Press, 2013.

[98] Xavier De Carné De Carnavalet and Mohammad Mannan. A large-scale evaluation of high-impact password strength meters. *ACM Transactions on Information and System Security (TISSEC)*, 18(1):1–32, 2015.

[99] Cybersecurity and Infrastructure Security Agency (CISA). Security tip (st04-002): Choosing and protecting passwords, 2009. URL `https://www.us-cert.gov/ncas/tips/ST04-002`.

[100] Paul A. Grassi, James L. Fenton, Elaine M. Newton, Ray A. Perlner, Andrew R. Regenscheid, William E. Burr, Justin P. Richer, Naomi B. Lefkovitz, Jamie M. Danker, Yee-Yin Choong, Kristen K. Greene, and Mary F. Theofanos. Digital Identity Guidelines. `https://pages.nist.gov/800-63-3/sp800-63b.html`, 2017. [Online; accessed 6-January-2021].

[101] Paul A. Grassi, James L. Fenton, Elaine M. Newton, Ray A. Perlner, Andrew R. Regenscheid, William E. Burr, Justin P. Richer, Naomi B. Lefkovitz, Jamie M. Danker, Yee-Yin Choong, Kristen K. Greene, and Mary F. Theofanos. NIST Special Publication 800-63B - Digital Identity Guidelines: Authentication and Lifecycle Management. Technical report, National Institute for Standards and Technology, 2017.

[102] Richard Shay, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Alain Forget, Saranga Komanduri, Michelle L Mazurek, William Melicher, Sean M Segreti, and Blase Ur. 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*, pages 2903–2912, 2015.

[103] Gaëtan Leurent and Thomas Peyrin. From collisions to chosen-prefix collisions application to full sha-1. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 527–555, Cham, 2019. Springer International Publishing. ISBN 978-3-030-17659-4.

[104] Guide to Cryptography. `https://wiki.owasp.org/index.php/Guide_to_Cryptography#Hashes`, 2017. [Online; accessed 21-December-2020].

[105] Ronald Rivest. The MD5 Message-Digest Algorithm. Technical report, MIT Laboratory for Computer Science and RSA Data Security, Inc., 1992.

[106] Quynh Dang. Secure hash standard, 2015-08-04 2015.

[107] Paul A Grassi, Michael E Garcia, and James L Fenton. Digital identity guidelines. *NIST special publication*, 800:63–3, 2017.

[108] David McCandless and Tom Evans. World's Biggest Data Breaches and Hacks. `https://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/`, 2022. [Online; accessed 12-November-2022].

[109] Joseph Bonneau, Cormac Herley, Paul C Van Oorschot, and Frank Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *2012 IEEE Symposium on Security and Privacy*, pages 553–567. IEEE, 2012.

[110] Zahid Maqbool, Palvi Aggarwal, V. S. Chandrasekhar Pammi, and Varun Dutt. Cyber Security: Effects of Penalizing Defenders in Cyber-Security Games via Experimentation and Computational Modeling. *Frontiers in Psychology*, In press, 01 2020. doi: $10.3389/\mathrm{fpsyg}.2020.00011$.

[111] Nihad A Hassan. *Digital Forensics Basics: A Practical Guide Using Windows OS*. Apress, 2019.

[112] Martti Lehto and Pekka Neittaanmäki. *Cyber Security: Power and Technology*, volume 93. Springer, 2018.

[113] Briland Hitaj, Paolo Gasti, Giuseppe Ateniese, and Fernando Perez-Cruz. PassGAN: A Deep Learning Approach for Password Guessing. In *Applied Cryptography and Network Security*, pages 217–237. Springer, 2019.

[114] Christof Paar and Jan Pelzl. *Understanding cryptography: a textbook for students and practitioners*. Springer Science & Business Media, 2009.

[115] Mudassar Raza, Muhammad Iqbal, Muhammad Sharif, and Waqas Haider. A survey of password attacks and comparative analysis on methods for secure authentication. *World Applied Sciences Journal*, 19(4):439–444, 2012.

[116] Hashcat v6.2.6 benchmark on the Nvidia RTX 4090. `https://gist.github.com/Chick3nman/32e662a5bb63bc4f51b847bb422222fd`, 2022. [Online; accessed 26-November-2022].

[117] Sadeq mohammed, Sefer KURNAZ, and Alaa Hamid Mohammed. Secure pin authentication in java smart card using honey encryption. In *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pages 1–4, 2020. doi: $10.1109/\mathrm{HORA}49412.2020.9152936$.

[118] Stefan Viehböck. Brute forcing wi-fi protected setup. *Wi-Fi Protected Setup*, 9, 2011.

[119] Denis Foo Kune and Yongdae Kim. Timing attacks on pin input devices. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, page 678–680, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450302456. doi: $10.1145/1866307.1866395$. URL `https://doi.org/10.1145/1866307.1866395`.

[120] Emanuel Tirado, Brendan Turpin, Cody Beltz, Phillip Roshon, Rylin Judge, and Kanwal Gagneja. A new distributed brute-force password cracking technique. In Robin Doss, Selwyn Piramuthu, and Wei Zhou, editors, *Future Network Systems and Security*, pages 117–127, Cham, 2018. Springer International Publishing. ISBN 978-3-319-94421-0.

[121] Mobin Javed and Vern Paxson. Detecting stealthy, distributed ssh bruteforcing. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, page 85–96, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450324779. doi: $10.1145/2508859.2516719$. URL `https://doi.org/10.1145/2508859.2516719`.

[122] Laatansa, Ragil Saputra, and Beta Noranita. Analysis of gpgpu-based brute-force and dictionary attack on sha-1 password hash. In *2019 3rd International Conference on Informatics and Computational Sciences (ICICoS)*, pages 1–4, 2019. doi: $10.1109/ICICoS48119.2019.8982390$.

[123] Martin Hellman. A cryptanalytic time-memory trade-off. *IEEE transactions on Information Theory*, 26(4):401–406, 1980.

[124] Alex Biryukov, Sourav Mukhopadhyay, and Palash Sarkar. Improved time-

memory trade-offs with multiple data. In *International Workshop on Selected Areas in Cryptography*, pages 110–127. Springer, 2005.

[125] Nurdan Saran and Ali Doganaksoy. Choosing parameters to achieve a higher success rate for Hellman time memory trade off attack. In *2009 International Conference on Availability, Reliability and Security*, pages 504–509. IEEE, 2009.

[126] Xiao-jian Wang, Xiao-feng Liao, and Hong-yu Huang. Improvement of rainbow table technology based on number cutting of reduction function. *Computer Engineering*, 7:36, 2013.

[127] Philippe Oechslin. Making a faster cryptanalytic time-memory trade-off. In *Annual International Cryptology Conference*, pages 617–630. Springer, 2003.

[128] List of Rainbow Tables. `http://project-rainbowcrack.com/table.htm`, 2020. [Online; accessed 19-October-2022].

[129] Sudhir Aggarwal, Shiva Houshmand, and Matt Weir. *New Technologies in Password Cracking Techniques*, pages 179–198. Springer International Publishing, Cham, 2018. ISBN 978-3-319-75307-2. doi: $10.1007/978\text{-}3\text{-}319\text{-}75307\text{-}2\_11$. URL `https://doi.org/10.1007/978-3-319-75307-2_11`.

[130] Gildas Avoine, Xavier Carpent, and Diane Leblanc-Albarel. Precomputation for rainbow tables has never been so fast. In Elisa Bertino, Haya Shulman, and Michael Waidner, editors, *Computer Security – ESORICS 2021*, pages 215–234, Cham, 2021. Springer International Publishing. ISBN 978-3-030-88428-4.

[131] Lijun Zhang, Cheng Tan, and Fei Yu. An improved rainbow table attack for

long passwords. *Procedia Computer Science*, 107:47–52, 2017. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2017.03.054. URL https://www.sciencedirect.com/science/article/pii/S1877050917303290. Advances in Information and Communication Technology: Proceedings of 7th International Congress of Information and Communication Technology (ICICT2017).

[132] Kostas Theocharoulis, Ioannis Papaefstathiou, and Charalampos Manifavas. Implementing rainbow tables in high-end fpgas for super-fast password cracking. In *2010 International Conference on Field Programmable Logic and Applications*, pages 145–150, 2010. doi: 10.1109/FPL.2010.120.

[133] Panagiotis Papantonakis, Dionisios Pnevmatikatos, Ioannis Papaefstathiou, and Charalampos Manifavas. Fast, fpga-based rainbow table creation for attacking encrypted mobile communications. In *2013 23rd International Conference on Field programmable Logic and Applications*, pages 1–6, 2013. doi: 10.1109/FPL.2013.6645525.

[134] Matt Weir, Sudhir Aggarwal, Breno De Medeiros, and Bill Glodek. Password cracking using probabilistic context-free grammars. In *2009 30th IEEE Symposium on Security and Privacy*, pages 391–405. IEEE, 2009.

[135] Benny Pinkas and Tomas Sander. Securing passwords against dictionary attacks. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 161–170, 2002.

[136] L. Bošnjak, 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)*, pages 1161–1166, 2018. doi: 10.23919/MIPRO.2018.8400211.

[137] Ar Kar Kyaw, Franco Sioquim, and Justin Joseph. Dictionary attack on wordpress: Security and forensic analysis. In *2015 Second International Conference on Information Security and Cyber Forensics (InfoSec)*, pages 158–164, 2015. doi: 10.1109/InfoSec.2015.7435522.

[138] Ding Wang and Ping Wang. Offline dictionary attack on password authentication schemes using smart cards. In Yvo Desmedt, editor, *Information Security*, pages 221–237, Cham, 2015. Springer International Publishing. ISBN 978-3-319-27659-5.

[139] Jan Vykopal, Tomas Plesnik, and Pavel Minarik. Network-based dictionary attack detection. In *2009 International Conference on Future Networks*, pages 23–27, 2009. doi: 10.1109/ICFN.2009.36.

[140] Aditi Roy, Nasir Memon, and Arun Ross. Masterprint: Exploring the vulnerability of partial fingerprint-based authentication systems. *IEEE Transactions on Information Forensics and Security*, 12(9):2013–2025, 2017. doi: 10.1109/TIFS.2017.2691658.

[141] Philip Bontrager, Aditi Roy, Julian Togelius, Nasir Memon, and Arun Ross. Deepmasterprints: Generating masterprints for dictionary attacks via latent variable evolution. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–9, 2018. doi: 10.1109/BTAS.2018.8698539.

[142] Mirko Marras, Pawel Korus, Nasir D Memon, and Gianni Fenu. Adversarial optimization for dictionary attacks on speaker verification. In *Interspeech*, pages 2913–2917, 2019.

[143] Sudhir Aggarwal, Shiva Houshmand, and Matt Weir. New technologies in

password cracking techniques. In *Cyber Security: Power and Technology*, pages 179–198. Springer, 2018.

[144] hashcat. `https://hashcat.net/hashcat/`, 2022. [Online; accessed 11-October-2022].

[145] Password Analysis and Cracking Kit. `https://github.com/iphelix/pack`, 2019. [Online; accessed 14-October-2022].

[146] Robert Layton and Paul A. Watters. A methodology for estimating the tangible cost of data breaches. *Journal of Information Security and Applications*, 19 (6):321–330, 2014. ISSN 2214-2126. doi: https://doi.org/10.1016/j.jisa.2014.10.012. URL `https://www.sciencedirect.com/science/article/pii/S2214212614001483`.

[147] Ilenia Confente, Giorgia Giusi Siciliano, Barbara Gaudenzi, and Matthias Eickhoff. Effects of data breaches from user-generated content: A corporate reputation analysis. *European Management Journal*, 37(4):492–504, 2019. ISSN 0263-2373. doi: https://doi.org/10.1016/j.emj.2019.01.007. URL `https://www.sciencedirect.com/science/article/pii/S0263237319300234`.

[148] Shiva Houshmand, Sudhir Aggarwal, and Randy Flood. Next gen PCFG password cracking. *IEEE Transactions on Information Forensics and Security*, 10 (8):1776–1791, 2015.

[149] Frederick Jelinek, John D Lafferty, and Robert L Mercer. Basic methods of probabilistic context free grammars. In *Speech Recognition and Understanding*, pages 345–360. Springer, 1992.

[150] Hong Jeong. *Architectures for Computer Vision: From Algorithm to Chip with Verilog*. John Wiley & Sons, 2014.

[151] Shiva Houshmand, Sudhir Aggarwal, and Umit Karabiyik. Identifying passwords stored on disk. In Gilbert Peterson and Sujeet Shenoi, editors, *Advances in Digital Forensics XI*, pages 195–213, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24123-4.

[152] Markus Dürmuth, Fabian Angelstorf, Claude Castelluccia, Daniele Perito, and Abdelberi Chaabane. Omen: Faster password guessing using an ordered markov enumerator. In *International Symposium on Engineering Secure Software and Systems*, pages 119–132. Springer, 2015.

[153] Prince Processor. `https://github.com/hashcat/princeprocessor`, 2021. [Online; accessed 14-October-2022].

[154] William Melicher, Blase Ur, Sean M. Segreti, Saranga Komanduri, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Fast, Lean, and Accurate: Modeling Password Guessability Using Neural Networks. In *Proceedings of the 25th USENIX Conference on Security Symposium*, SEC'16, page 175–191, USA, 2016. USENIX Association. ISBN 9781931971324.

[155] Yunyu Liu, Zhiyang Xia, Ping Yi, Yao Yao, Tiantian Xie, Wei Wang, and Ting Zhu. Genpass: A general deep learning model for password guessing with pcfg rules and adversarial generation. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6, 2018. doi: $10.1109/\mathrm{ICC}.2018.8422243$.

[156] Kunyu Yang, Xuexian Hu, Qihui Zhang, Jianghong Wei, and Wenfen Liu. Vaepass: A lightweight passwords guessing model based on variational auto-encoder. *Computers & Security*, 114:102587, 2022. ISSN 0167-4048. doi: $\mathrm{https://doi.org/10.1016/j.cose.2021.102587}$. URL `https://www.sciencedirect.com/science/article/pii/S0167404821004107`.

[157] John the Ripper Password Cracker. https://www.openwall.com/john/. [Online; accessed 11-October-2022].

[158] Passware Kit Forensic. https://www.passware.com/kit-forensic/. [Online; accessed 14-October-2022].

[159] Elcomsoft Desktop Forensic Bundle. https://www.elcomsoft.fr/edfb.html. [Online; accessed 14-October-2022].

[160] SciEngines Hardware. https://www.sciengines.com/technology-platform/sciengines-hardware/. [Online; accessed 14-October-2022].

[161] Lubos Gaspar, Iwen Coisel, and Laurent Beslay. FPGA performances in Cryptography. Performance analysis of different cryptographic algorithms implemented in an FPGA, 2014.

[162] Hashcat has won CMIYC 2019! https://contest-2019.korelogic.com/, 2019. [Online; accessed 12-October-2022].

[163] Password Cracker. https://download.cnet.com/Password-Cracker/3000-2092_4-10226556.html, 2020. [Online; accessed 11-October-2022].

[164] Brutus Password Cracker – Download brutus-aet2.zip AET2. https://www.darknet.org.uk/2006/09/brutus-password-cracker-download-brutus-aet2zip-aet2/, 2006. [Online; accessed 11-October-2022].

[165] Cain and Abel. https://web.archive.org/web/20190603235413/http://www.oxid.it/cain.html], year = 2019, note = '[Online; accessed 11-October-2022].

[166] What is ophcrack? `https://ophcrack.sourceforge.io/`, note = '[Online; accessed 11-October-2022].

[167] van Hauser. thc-hydra. `https://github.com/vanhauser-thc/thc-hydra`. [Online; accessed 12-October-2022].

[168] JoMo-Kun. Medusa Parallel Network Login Auditor. `http://foofus.net/goons/jmk/medusa/medusa.html`, 2016. [Online; accessed 12-October-2022].

[169] Free Password Hash Cracker. `https://crackstation.net/`, 2019. [Online; accessed 12-October-2022].

[170] Aircrack-ng. `https://www.aircrack-ng.org/`, 2022. [Online; accessed 12-October-2022].

[171] L0phtCrack is Now Open Source. `https://l0phtcrack.gitlab.io/`, 2021. [Online; accessed 13-October-2022].

[172] RainbowCrack. `http://project-rainbowcrack.com/`, 2020. [Online; accessed 13-October-2022].

[173] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 212–219, New York, NY, USA, 1996. Association for Computing Machinery. ISBN 0897917855. doi: $10.1145/237814.237866$. URL `https://doi.org/10.1145/237814.237866`.

[174] Markus Dürmuth, Maximilian Golla, Philipp Markert, Alexander May, and Lars Schlieper. Towards quantum large-scale password guessing on real-world distributions. In Mauro Conti, Marc Stevens, and Stephan Krenn, editors,

Cryptology and Network Security, pages 412–431, Cham, 2021. Springer International Publishing. ISBN 978-3-030-92548-2.

[175] Sherry Wang, Carlisle Adams, and Anne Broadbent. Password authentication schemes on a quantum computer. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 346–350, 2021. doi: 10.1109/QCE52317.2021.00051.

[176] Mohit Kumar Sharma and Manisha J. Nene. Quantum one time password with biometrics. In Jennifer S. Raj, Abul Bashar, and S. R. Jino Ramson, editors, *Innovative Data Communication Technologies and Application*, pages 312–318, Cham, 2020. Springer International Publishing. ISBN 978-3-030-38040-3.

[177] Joseph Bernstein. Survey Says: People Have Way Too Many Passwords To Remember. `https://www.buzzfeednews.com/article/josephbernstein/survey-says-people-have-way-too-many-passwords-to-remember`, 2016. [Online; accessed 3-January-2021].

[178] 8 truths about-passwords. `https://blog.lastpass.com/2017/11/lastpass-reveals-8-truths-about-passwords-in-the-new-password-expose.html`, 2019. [Online; accessed 25-Septembery-2019].

[179] Verena Zimmermann and Nina Gerber. The password is dead, long live the password–a laboratory study on user perceptions of authentication schemes. *International Journal of Human-Computer Studies*, 133:26–44, 2020.

[180] Elizabeth Stobert and Robert Biddle. Memory retrieval and graphical passwords. In *Proceedings of the Ninth Symposium on Usable Privacy and Security*, pages 1–14, 2013.

[181] Dinei Florencio and Cormac Herley. A large-scale study of web password habits. In *Proceedings of the 16th international conference on World Wide Web*, pages 657–666, 2007.

[182] Elizabeth Stobert and Robert Biddle. The password life cycle: User behaviour in managing passwords. In *10th Symposium On Usable Privacy and Security (SOUPS 2014)*, pages 243–255, 2014.

[183] Rick Wash, Emilee Rader, Ruthie Berman, and Zac Wellmer. Understanding password choices: How frequently entered passwords are re-used across websites. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, pages 175–188, 2016.

[184] Taiabul Haque, Matthew Wright, and Shannon Scielzo. Hierarchy of users' web passwords: Perceptions, practices and susceptibilities. *International Journal of Human-Computer Studies*, 72(12):860–874, 2014.

[185] Youngsok Bang, Dong-Joo Lee, Yoon-Soo Bae, and Jae-Hyeon Ahn. Improving information security management: An analysis of ID–password usage and a new login vulnerability measure. *International Journal of Information Management*, 32(5):409–418, 2012.

[186] intersoft consulting. General Data Protection Regulation. `https://gdpr-info.eu/`. [Online; accessed 22-November-2022].

[187] Surfshark in Cybersecurity. Data breaches rise globally in Q3 of 2022. `https://surfshark.com/blog/data-breach-statistics-2022-q3`, 2022. [Online; accessed 4-October-2022].

[188] Rick Wash, Emilee Rader, Ruthie Berman, and Zac Wellmer. Understanding Password Choices: How Frequently Entered Passwords Are Re-used

across Websites. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, pages 175–188, Denver, CO, June 2016. USENIX Association. ISBN 978-1-931971-31-7. URL https://www.usenix.org/conference/soups2016/technical-sessions/presentation/wash.

[189] Emanuel von Zezschwitz, Alexander De Luca, and Heinrich Hussmann. Survival of the Shortest: A Retrospective Analysis of Influencing Factors on Password Composition. In Paula Kotzé, Gary Marsden, Gitte Lindgaard, Janet Wesson, and Marco Winckler, editors, *Human-Computer Interaction – INTERACT 2013*, pages 460–467, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-40477-1.

[190] Ding Wang, Zijian Zhang, Ping Wang, Jeff Yan, and Xinyi Huang. Targeted online password guessing: An underestimated threat. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 1242–1254, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341394. doi: 10.1145/2976749.2978339. URL https://doi.org/10.1145/2976749.2978339.

[191] Blase Ur, Jonathan Bees, Sean M Segreti, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Do users' perceptions of password security match reality? In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 3748–3760, 2016.

[192] Blase Ur, Fumiko Noma, Jonathan Bees, Sean M Segreti, Richard Shay, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. "I Added '!'at the End to Make It Secure": Observing Password Creation in the Lab. In *Eleventh Symposium on Usable Privacy and Security (SOUPS 2015)*, pages 123–140, 2015.

[193] Zhipeng Liu, Yefan Hong, and Dechang Pi. A large-scale study of web pass-

word habits of chinese network users. *Journal of Software (JSW)*, 9(2):293–297, 2014.

[194] Gang Han, Yu Yu, Xiangxue Li, Kefei Chen, and Hui Li. Characterizing the semantics of passwords: The role of Pinyin for Chinese Netizens. *Computer Standards & Interfaces*, 54:20–28, 2017.

[195] Joseph Bonneau. The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords. In *2012 IEEE Symposium on Security and Privacy*, pages 538–552. IEEE, 2012.

[196] Andrej Cvetkovski and Flavio Esposito. The password literacy in north macedonia: A case study. In *Proceedings of the Third Central European Cybersecurity Conference*, pages 1–6, 2019.

[197] Rafael Veras, Christopher Collins, and Julie Thorpe. On Semantic Patterns of Passwords and their Security Impact. In *Network and Distributed System Security (NDSS) Symposium*, 2014.

[198] Rafael Veras, Julie Thorpe, and Christopher Collins. Visualizing semantics in passwords: The role of dates. In *Proceedings of the Ninth International Symposium on Visualization for Cyber Security*, pages 88–95, 2012.

[199] Jianping Zeng, Jiangjiao Duan, and Chengrong Wu. Empirical study on lexical sentiment in passwords from chinese websites. *Computers & Security*, 80: 200–210, 2019.

[200] Ruba Alomari, Miguel Vargas Martin, Shane MacDonald, Amit Maraj, Ramiro Liscano, and Christopher Bellman. Inside out-a study of users' perceptions of password memorability and recall. *Journal of Information Security and Applications*, 47:223–234, 2019.

[201] Mashael AlSabah, Gabriele Oligeri, and Ryan Riley. Your culture is in your password: An analysis of a demographically-diverse password dataset. *Computers & Security*, 77:427–441, 2018.

[202] Ruba Alomari and Julie Thorpe. On password behaviours and attitudes in different populations. *Journal of Information Security and Applications*, 45: 79–89, 2019.

[203] Emanuel Von Zezschwitz, Alexander De Luca, and Heinrich Hussmann. Honey, I shrunk the keys: influences of mobile devices on password composition and authentication performance. In *Proceedings of the 8th nordic conference on human-computer interaction: fun, fast, foundational*, pages 461–470, 2014.

[204] Yue Li, Haining Wang, and Kun Sun. Personal Information in Passwords and its Security Implications. *IEEE Transactions on Information Forensics and Security*, 12(10):2320–2333, 2017.

[205] Yimin Guo, Zhenfeng Zhang, Yajun Guo, and Xiaowei Guo. Nudging personalized password policies by understanding users' personality. *Computers & Security*, page 101801, 2020.

[206] Peter Mayer, Collins W. Munyendo, Michelle L. Mazurek, and Adam J. Aviv. Why users (don't) use password managers at a large educational institution. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1849–1866, Boston, MA, August 2022. USENIX Association. ISBN 978-1-939133-31-1. URL `https://www.usenix.org/conference/usenixsecurity22/presentation/mayer`.

[207] William E. Burr, Donna F. Dodson, and W. Timothy Polk. NIST Special Publi-

cation 800-63 - Electronic Authentication Guideline. Technical report, National Institute for Standards and Technology, 2004.

[208] Richard Shay, Saranga Komanduri, Adam L. Durity, Phillip (Seyoung) Huh, Michelle L. Mazurek, Sean M. Segreti, Blase Ur, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Designing Password Policies for Strength and Usability. *ACM Trans. Inf. Syst. Secur.*, 18(4), May 2016. ISSN 1094-9224. doi: $10.1145/2891411$. URL `https://doi.org/10.1145/2891411`.

[209] Blase Ur, Patrick Gage Kelley, Saranga Komanduri, Joel Lee, Michael Maass, Michelle L Mazurek, Timothy Passaro, Richard Shay, Timothy Vidas, Lujo Bauer, et al. How does your password measure up? the effect of strength meters on password creation. In *Presented as part of the 21st USENIX Security Symposium (USENIX Security 12)*, pages 65–80, 2012.

[210] Krzysztof Gołofit. Click passwords under investigation. In *European Symposium on Research in Computer Security*, pages 343–358. Springer, 2007.

[211] Karen Renaud and Antonella De Angeli. Visual passwords: Cure-all or snake-oil? *Communications of the ACM*, 52(12):135–140, 2009.

[212] Alan S Brown, Elisabeth Bracken, Sandy Zoccoli, and King Douglas. Generating and remembering passwords. *Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition*, 18(6): 641–651, 2004.

[213] S. Komanduri, R. Shay, P.G. Kelley, M.L. Mazurek, L. Bauer, N. Christin, L.F. Cranor, and S. Egelman. Of passwords and people: Measuring the effect of password-composition policies. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, pages 2595–2604, 2011.

[214] The Password Meter. `http://www.passwordmeter.com/`. [Online; accessed 22-October-2022].

[215] How Secure is Your Password? `https://www.passwordmonster.com/`. [Online; accessed 22-October-2022].

[216] How Secure is Your Password? `https://lastpass.com/howsecure.php`. [Online; accessed 22-October-2022].

[217] Check your password. `https://password.kaspersky.com/`. [Online; accessed 22-October-2022].

[218] Javier Galbally, Iwen Coisel, and Ignacio Sanchez. A probabilistic framework for improved password strength metrics. In *2014 International Carnahan Conference on Security Technology (ICCST)*, pages 1–6. IEEE, 2014.

[219] Shukun Yang, Shouling Ji, Xin Hu, and Raheem Beyah. Effectiveness and soundness of commercial password strength meters. *Network and Distributed System Security Symposium (NDSS)*, 2015.

[220] Michelle L Mazurek, Saranga Komanduri, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Patrick Gage Kelley, Richard Shay, and Blase Ur. Measuring password guessability for an entire university. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 173–186, 2013.

[221] Susanna Heidt and Adam J Aviv. Refining Graphical Password Strength Meters for Android Phones. In *Poster presented at the Twelfth Symposium on Usable Security and Privacy, SOUPS*, volume 16, 2016.

[222] Javier Galbally, Iwen Coisel, and Ignacio Sanchez. A New Multimodal Approach for Password Strength Estimation–Part II: Experimental Evaluation.

IEEE Transactions on Information Forensics and Security, 12(12):2845–2860, 2017.

[223] Yimin Guo and Zhenfeng Zhang. Lpse: lightweight password-strength estimation for password meters. *Computers & Security*, 73:507–518, 2018.

[224] Patrick Gage Kelley, Saranga Komanduri, Michelle L Mazurek, Richard Shay, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Julio Lopez. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In *2012 IEEE symposium on security and privacy*, pages 523–537. IEEE, 2012.

[225] Daniel Lowe Wheeler. zxcvbn: Low-budget password strength estimation. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 157–173, 2016.

[226] Claude Castelluccia, Markus Dürmuth, and Daniele Perito. Adaptive Password-Strength Meters from Markov Models. In *NDSS*, 2012.

[227] Shiva Houshmand and Sudhir Aggarwal. Building Better Passwords Using Probabilistic Techniques. In *Proceedings of the 28th Annual Computer Security Applications Conference*, ACSAC '12, page 109–118, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450313124. doi: $10.1145/2420950.2420966$. URL `https://doi.org/10.1145/2420950.2420966`.

[228] D. Wang, D. He, H. Cheng, and P. Wang. fuzzyPSM: A New Password Strength Meter Using Fuzzy Probabilistic Context-Free Grammars. In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 595–606, 2016. doi: $10.1109/\mathrm{DSN}.2016.60$.

[229] Qiying Dong, Chunfu Jia, Fei Duan, and Ding Wang. Rls-psm: A robust and accurate password strength meter based on reuse, leet and separation. *IEEE Transactions on Information Forensics and Security*, 16:4988–5002, 2021. doi: 10.1109/TIFS.2021.3107147.

[230] Blase Ur, Felicia Alfieri, Maung Aung, Lujo Bauer, Nicolas Christin, Jessica Colnago, Lorrie Faith Cranor, Henry Dixon, Pardis Emami Naeini, Hana Habib, Noah Johnson, and William Melicher. Design and Evaluation of a Data-Driven Password Meter. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 3775–3786, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346559. doi: 10.1145/3025453.3026050. URL https://doi.org/10.1145/3025453.3026050.

[231] Yimin Guo and Zhenfeng Zhang. Lpse: Lightweight password-strength estimation for password meters. *Computers & Security*, 73:507–518, 2018. ISSN 0167-4048. doi: https://doi.org/10.1016/j.cose.2017.07.012. URL https://www.sciencedirect.com/science/article/pii/S0167404817301530.

[232] Dario Pasquini, Giuseppe Ateniese, and Massimo Bernaschi. Interpretable probabilistic password strength meters via deep learning. In Liqun Chen, Ninghui Li, Kaitai Liang, and Steve Schneider, editors, *Computer Security – ESORICS 2020*, pages 502–522, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58951-6.

[233] Maximilian Golla and Markus Dürmuth. On the accuracy of password strength meters. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 1567–1582, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356930.

doi: 10.1145/3243734.3243769. URL https://doi.org/10.1145/3243734.3243769.

[234] Maximilian Golla, Benedict Beuscher, and Markus Dürmuth. On the security of cracking-resistant password vaults. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 1230–1241, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341394. doi: 10.1145/2976749.2978416. URL https://doi.org/10.1145/2976749.2978416.

[235] Xavier de Carné de Carnavalet and Mohammad Mannan. From very weak to very strong: Analyzing password-strength meters. In *Network and Distributed System Security (NDSS) Symposium 2014*. Internet Society, February 2014. URL https://spectrum.library.concordia.ca/id/eprint/978105/. In Press.

[236] Julie Thorpe and Paul C Van Oorschot. Towards secure design choices for implementing graphical passwords. In *20th Annual Computer Security Applications Conference*, pages 50–60. IEEE, 2004.

[237] Jean-Camille Birget, Dawei Hong, and Nasir D Memon. Robust discretization, with an application to graphical passwords. *IACR Cryptology ePrint Archive*, 2003:168, 2003.

[238] Thomas S. Tullis, Donna P. Tedesco, and Kate E. McCaffrey. Can users remember their pictorial passwords six years later. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, page 1789–1794, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450302685. doi: 10.1145/1979742.1979945. URL https://doi.org/10.1145/1979742.1979945.

[239] L. O'Gorman. Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE*, 91(12):2021–2040, 2003.

[240] Kamran Siddique, Zahid Akhtar, and Yangwoo Kim. Biometrics vs passwords: a modern version of the tortoise and the hare. *Computer Fraud & Security*, 2017(1):13–17, 2017.

[241] D. Charoen, M. Raman, and L. Olfman. Improving end user behaviour in password utilization: An action research initiative. *Systemic Practice and Action Research*, 21(1):55–72, 2008.

[242] Viktor Taneski, Marjan Heričko, and Boštjan Brumen. Password security—no change in 35 years? In *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1360–1365. IEEE, 2014.

[243] L Bošnjak, J Sreš, and Bosnjak 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)*, pages 1161–1166. IEEE, 2018.

[244] Troy Hunt. Password reuse, credential stuffing and another billion records in have i been pwned. *troyhunt. com*, 2017.

[245] Hannah Knowles. 533 million Facebook users' phone numbers, personal information exposed online, report says. `https://www.washingtonpost.com/business/2021/04/03/facebook-data-leak-insider/`, 2021. [Online; accessed 28-October-2022].

[246] Sarah Pearman, Jeremy Thomas, Pardis Emami Naeini, Hana Habib, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Serge Egelman, and Alain Forget. Let's go in for a closer look: Observing passwords in their natural

habitat. In *Proceedings of the 24th ACM SIGSAC Conference on Computer and Communications Security*. ACM, October 2017. doi: 10.1145/3133956.3133973. URL https://www.ece.cmu.edu/~lbauer/papers/2017/ccs2017-password-reuse.pdf.

[247] E. Sai Kishan, M. Hemchudaesh, B. Gowri Shankar, B. Sai Brahadeesh, and K. P. Jevitha. Password generation based on song lyrics and its management. In Mayank Singh, Vipin Tyagi, P. K. Gupta, Jan Flusser, and Tuncer Ören, editors, *Advances in Computing and Data Sciences*, pages 278–290, Cham, 2022. Springer International Publishing. ISBN 978-3-031-12641-3.

[248] JV Roig, J de la Cuesta, J Castillo, J Cabardo, E Casiño, E Salalima, and M Sanchez. Frequency of compromised passwords used by students and staff of asia pacific college: an analysis using nist sp 800-63b and pwned passwords. *IOP Conference Series: Materials Science and Engineering*, 482 (1):012035, feb 2019. doi: 10.1088/1757-899X/482/1/012035. URL https://dx.doi.org/10.1088/1757-899X/482/1/012035.

[249] Joseph Bonneau. The science of guessing: Analyzing an anonymized corpus of 70 million passwords. In *2012 IEEE Symposium on Security and Privacy*, pages 538–552, 2012. doi: 10.1109/SP.2012.49.

[250] Patrick Gage Kelley, Saranga Komanduri, Michelle L. Mazurek, Richard Shay, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Julio Lopez. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In *2012 IEEE Symposium on Security and Privacy*, pages 523–537, 2012. doi: 10.1109/SP.2012.38.

[251] Pwdb-Public. https://github.com/ignis-sec/Pwdb-Public, 2020. [Online; accessed 2-October-2022].

[252] Ding Wang, Ping Wang, Debiao He, and Yuan Tian. Birthday, Name and Bifacial-security: Understanding Passwords of Chinese Web Users. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1537–1555, 2019.

[253] J. Galbally, I. Coisel, and I. Sanchez. A New Multimodal Approach for Password Strength Estimation—Part II: Experimental Evaluation. *IEEE Transactions on Information Forensics and Security*, 12(12):2845–2860, 2017. doi: 10.1109/TIFS.2017.2730359.

[254] Maximilian Golla and Markus Dürmuth. On the Accuracy of Password Strength Meters. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, page 1567–1582. Association for Computing Machinery, 2018. doi: 10.1145/3243734.3243769.

[255] Ding Wang, Ping Wang, Debiao He, and Yuan Tian. Birthday, name and bifacial-security: Understanding passwords of chinese web users. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1537–1555, Santa Clara, CA, August 2019. USENIX Association. ISBN 978-1-939133-06-9. URL https://www.usenix.org/conference/usenixsecurity19/presentation/wang-ding.

[256] Ding Wang, Haibo Cheng, Ping Wang, Xinyi Huang, and Gaopeng Jian. Zipf's law in passwords. *IEEE Transactions on Information Forensics and Security*, 12(11):2776–2791, 2017. doi: 10.1109/TIFS.2017.2721359.

[257] Wikipedia. https://www.wikipedia.org/. [Online; accessed 14-April-2022].

[258] Global and Unified Access to Knowledge Graphs. https://www.dbpedia.org/. [Online; accessed 4-November-2022].

[259] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, and Christian Bizer. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6, 01 2014. doi: 10.3233/SW-140134.

[260] Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. 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*, pages 23–30, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.4. URL https://aclanthology.org/2020.emnlp-demos.4.

[261] Hashes.org. https://hashes.org/. [Online; accessed 21-May-2020].

[262] 320 Million Hashes Exposed. https://blog.cynosureprime.com/2017/08/320-million-hashes-exposed.html?m=1, 2017. [Online; accessed 22-August-2021].

[263] Peter Kacherginsky. Tool release: Password analysis and cracking kit, August 2013. URL https://iphelix.medium.com/tool-release-password-analysis-and-cracking-kit-31a3587f550f.

[264] Robin Wood. pipal. https://github.com/digininja/pipal, 2022. [Online; accessed 14-October-2022].

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

[266] Symspellpy. https://github.com/mammothb/symspellpy, 2022. [Online; accessed 18-October-2022].

[267] SymSpell. https://github.com/wolfgarbe/SymSpell, 2022. [Online; accessed 18-October-2022].

[268] Worth Garbe. 1000x faster Spelling Correction, May 2017. URL https://towardsdatascience.com/symspellcompound-10ec8f467c9b.

[269] Reddit Pushshift Directory Contents. https://files.pushshift.io/reddit/comments/. [Online; accessed 2-November-2022].

[270] WordNet® a lexical database for English. https://wordnet.princeton.edu/. [Online; accessed 11-February-2020].

[271] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[272] GloVe. http://nlp.stanford.edu/data/glove.42B.300d.zip. [Online; accessed 11-February-2020].

[273] Probabilistic Context Free Grammar (PCFG) Password Guess Generator. https://github.com/lakiw/pcfg_cracker, 2022. [Online; accessed 14-October-2022].

[274] Maximilian Golla. Password Guessing Framework. https://github.com/RUB-SysSec/Password-Guessing-Framework, 2015. [Online; accessed 09-August-2022].

[275] RDFLib 6.2.0 package for working with RDF. https://rdflib.readthedocs.io/en/stable/. [Online; accessed 4-November-2022].

[276] Resource Description Framework (RDF). https://www.w3.org/RDF/, 2014. [Online; accessed 6-November-2022].

[277] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[278] W. John Wilbur and Karl Sirotkin. The automatic identification of stop words. *Journal of Information Science*, 18(1):45–55, 1992.

[279] Nvidia GeForce RTX 4090. https://www.nvidia.com/en-gb/geforce/graphics-cards/40-series/rtx-4090/. [Online; accessed 22-November-2022].

[280] Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013. URL http://arxiv.org/abs/1301.3781.

[281] Pretrained Embeddings. https://wikipedia2vec.github.io/wikipedia2vec/pretrained/. [Online; accessed 21-May-2022].

[282] Best64 Mangling Rule. https://github.com/hashcat/hashcat/blob/master/rules/best64.rule, 2018. [Online; accessed 14-October-2022].

[283] One Rule to Rule Them All. https://notsosecure.com/one-rule-to-rule-them-all, 2017. [Online; accessed 14-October-2022].

[284] Dinei Florêncio and Cormac Herley. Where Do Security Policies Come From? In *Proceedings of the Sixth Symposium on Usable Privacy and Security*, pages 1–14, 2010.

[285] Emin Islam Tatlı. Cracking More Password Hashes with Patterns. *IEEE Transactions on Information Forensics and Security*, 10(8):1656–1665, 2015.

[286] Ruth Rawlings. Top 200 Worst Passwords of 2019, Dec 2019. URL https://nordpass.com/blog/top-worst-passwords-2019.

[287] Sonic HPC. https://www.ucd.ie/itservices/ourservices/researchit/researchcomputing/sonichpc/. [Online; accessed 22-July-2022].

[288] Rafael Veras, Julie Thorpe, and Christopher Collins. Visualizing semantics in passwords: The role of dates. In *Proceedings of the Ninth International Symposium on Visualization for Cyber Security*, VizSec '12, page 88–95, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450314138. doi: 10.1145/2379690.2379702. URL https://doi.org/10.1145/2379690.2379702.

[289] Briland Hitaj, Paolo Gasti, Giuseppe Ateniese, and Fernando Perez-Cruz. Passgan: A deep learning approach for password guessing. In *International Conference on Applied Cryptography and Network Security*, pages 217–237. Springer, 2019.

[290] Security.org. Password Manager and Vault 2021 Annual Report: Usage, Awareness, and Market Size, 2021. URL https://www.security.org/digital-safety/password-manager-annual-report/.

Appendix A

List of Abbreviations

The following describes the significance of various acronyms and terms used throughout this thesis.

Acronyms

2FA Two-Factor Authentication.

ACPO Association of Chief Police Officers.

AI Artificial Intelligence.

API Application Programming Interface.

ATM Automated Teller Machine.

CCTV Close-Circuit Television.

CFFTPM Computer Forensics Field Triage Process Model.

CPU Central Processing Unit.

DFINT Digital Forensic Intelligence.

DoS Denial of Service.

DPPP Dynamic Personalised Password Policy.

DRbSI Data Reduction by Selective Imaging.

FBI Federal Bureau of Investigation.

FPGA Field Programmable Gate Array.

GAN Generative Adversarial Network.

GCNN Gated Convolutional Neural Network.

GDPR General Data Protection Regulation.

GPU Graphic Processing Unit.

HIBP Have I Been Pwned.

HIBP_v5 Have I Been Pwned version 5.

HPC High Performance Computing.

HUMINT Human intelligence.

IoT Internet of Things.

IT Information Technology.

JtR John the Ripper.

LEA Law Enforcement Agency.

LPSE Lightweight Password-Strength Estimation Method.

LSTM Long Short-Term Memory.

MD5 Message-Digest Algorithm.

ML Machine Learning.

MPI Message Passing Interface.

NATO North Atlantic Treaty Organization.

NIST National Institute of Standards and Technology.

NLP Natural Language Processing.

NTLM New Technology LAN Manager.

OMEN Ordered Markov Enumerator.

OS Operating System.

OSINT Open Source Intelligence.

OTP One Time Password.

OWASP Open Web Application Security Project.

PACK Password Analysis and Cracking Kit.

PCFG Probabilistic Context-Free Grammar.

PCWQ Password Cracking Wordlist Quality.

PGF Password Guessing Framework.

PII Personally Identifiable Information.

PIN Personal Identification Number.

PRINCE PRobability INfinite Chained Elements.

RDF Resource Description Framework.

RIPEND RACE Integrity Primitives Evaluation Message Digest.

SHA Secure Hash Algorithms.

SNA Social Network Analysis.

SOCMINT Social Media Intelligence.

SSH Secure Shell.

SSO Single-Sign-On.

TMTO Time-Memory Trade-Off.

UK United Kingdom.

VAE Variational Auto-Encoder.

VoIP Voice over IP.

VPN Virtual Private Network.

WEP Wired Equivalent Privacy.

WPA Wi-Fi Protected Access.

WPS Wi-Fi Protected Setup.

WWW World Wide Web.