Noroff University College

Bachelor in
Cyber Security

PassProof - a user friendly password checker

JEBRIL M. ALI

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF BACHELOR IN **CYBER SECURITY**

SUPERVISOR
Livinus Obiora Nweke

Noroff University College
Norway
May, 2025

Abstract

The reliance on passwords in the digital space has been the number one security protection, and as such, it becomes a paradox in cybersecurity. Despite their good security and effectiveness, they are undermined by user behaviors, and systems lack user guidance. Users regularly select weak and easily guessable credentials, while password meters and other policies fail to show the accuracy of the real-world risks of choosing weak passwords. This research tackles the critical vulnerability in a real-world breach called the AuthInfo dataset. it will examine, design, and develop PassProof, a tool for assessing the vulnerabilities of user input.

Furthermore, it involves analysis of the dataset. Utilizing Python Programming Language to develop the tool using different libraries. However, findings show that the majority of users do not use a good length; they use simplistic patterns and use a very common password, which had over 1 million users using the same password. The research gives the users a multi-component scoring algorithm to help them make stronger passwords. The calculation of the score is based on metrics like length and Shannon entropy; it will impose penalties based on character types to ensure a safer environment. Lastly, based on the findings recommendation is given to end users and organizations to avoid future breaches.

Keywords: *Password Security, User Behaviour Analysis, Data Breach Analysis, Password Strength Checker Tool, Password Strength Meter*

Acknowledgements

I want to acknowledge my supervisor, Livinus Obiora Nweke for assisting and being with me throughout the whole project. His presence was noticeable and helped shape my project; I would not be able to do this without him. I also want to thank Noroff for making it possible for me to choose this amazing topic about password security. They have taught me numerous lessons that can be used in daily life and in future settings; my achievement would not be possible without their support.

Mandatory Declaration

1.	I hereby declare that the submission answer is my own work, and that I have not used other sources other than as is referenced and cited correctly, or received help other than what is specifically acknowledged.	Yes
2.	**I further declare that this submission:** • Has not been used for another exam in another course at Noroff University College, at another department/university/college at home or abroad. • Does not refer to or make use of the work of others without acknowledgement. • Does not refer to my own previous work unless stated. • Has all the references given in the bibliography. • Is not a copy, duplicate or copy of someone else's work or answer. • Is not generated using AI generation tools.	Yes
3.	I am aware that a breach of any of the above is to be regarded as cheating and may result in cancellation of the exam and exclusion from universities and university colleges in Norway, cf. University and University College Act § 12-4 and Noroff University College Regulation § 4-5.	Yes
4.	I am aware that all components of this assignments may be checked for plagiarism and other forms of academic misconduct.	Yes
5.	I hereby acknowledge that I have been taught the appropriate ways to use the work of other researchers. I undertake to paraphrase, cite, and reference according to the acceptable academic practices, in accordance with the rules and guidelines, as taught.	Yes
6.	I am aware that Noroff University College will process all cases where cheating is suspected in accordance with the college's guidelines.	Yes

Publication Agreement

Authorisation for electronic publication of the thesis: Through submission you are accepting that Noroff University College has a perpetual, and royalty free right to retain a copy of work for its own internal use, and has the right to make work publicly available - considering any restrictions to publication.

Name: Jebril M. Ali

Place: Oslo

Date: May 1, 2025

202425 Version: Last updated 2024/10/16

Contents

List of Figures

1

Introduction

1.1 Introduction

Security has emerged as a critical consideration for both individuals and organizations. The use of strong passwords is one of the primary preventions against unauthorized access. Nevertheless, users are often clueless of the repercussions of cyber-attacks, attacks like brute force, dictionary attack, and fishing attack are on the rise. However, why does this matter if the only thing a user needs to do is fulfill the necessary requirements for a password? It is the responsibility of the organization to establish a good policy. Having a good policy is essential because users still tend to use passwords that are short, easily guessable, and weak (Chanda, 2016), (Pagar & Pise, 2017). Despite the function of passwords, their ubiquity and critical function lack security within the cybersecurity ecosystem. Individuals are under the wrong impression that having one security control guarantees their protection against attacks, which is far from the truth. Their effectiveness is often compromised by predictable user behaviors and inherent weaknesses in how password policy and management are implemented and enforced on users. This is one of the topics that are covered in the thesis, as it represents one of the risks and vulnerabilities that exist in password policies (Taneski et al., 2019).

The primary issue is a lack of awareness; certain users are unaware of the criteria for a secure password. Although there is a wealth of information on the internet on password management, a significant number of users continue to use weak and predictable passwords. Hackers have become increasingly sophisticated in their methods of hacking passwords as time has passed. Consequently, users who use insecure passwords are at increased risk. System-side defenses, such as password composition policies and strength meters, have been inconsistent, poorly designed from a usability standpoint, or they just fail to implement real-world scenarios to battle attacks from the outside. This is the reason many users are frustrated because of the inadequate protection they get (Shay et al., 2010). The result is a loop cycle of account compromises, identity theft, and data breaches in the digital services.

One fundamental problem with this situation in hand is the gap between what the best practices are for users and the actual behavior that users have. While there is a lot of information regarding secure password creation, many users still tend to use vulnerable strategies, often due to cognitive limitations, underestimation of security risk, and usability obstacles. Furthermore, there are many options designed to guide users, such as password strength meters, which may not take advantage of the wealth of empirical data available from real-world data breaches, causing misleading and incomplete assessments. Attack methodologies keep evolving; defenses must keep up with analyzing real-life breaches to understand root causes and come up with effective solutions.

This thesis aims to address this gap by directly linking the empirical analysis of real-world password data to the development of enhanced user guidance tools. The main focus will be to investigate and analyze the Authentication (AuthInfo) dataset (Güven et al., 2022), which is a repository of credentials exposed in a data breach. By performing a systematic analysis, the hope is to understand common vulnerabilities, structural patterns, and choices that users make when creating a password. This research seeks to obtain actionable insights into users downfall. The insights gotten from the research will be used to form a well-rounded tool called "PassProof", a novel password strength checking tool. The objective is to make a checker that takes what was learned from the research, analyze it, and give feedback to ensure users are well informed. The offer is to provide relevant, effective, and educational guidance to users of all levels in the critical phase of creating a password. The next section 1.2 will provide further background on password security challenges and related research that will give you a better context for the research.

1.2 Background and Related Work

A well-established practice emphasizes several key recommendations for users. Users should meet a minimum requirement of eight characters to increase resistance against brute-force and other attacks. Furthermore, users should avoid using words that are found in a dictionary, as they are the primary targets when attackers deploy a dictionary attack. Enhancing complexity involves having both uppercase and lowercase letters (Garfinkel & Spafford, 2003), as they increase the effort attackers need to put in. Firstly, it's important to understand why users need a strong password. The underlying assumption is that strong passwords are widely preferred and that people who use weak passwords do so because they are unaware that their passwords are weak. Following this premise, it is often believed that if users are alerted to the vulnerabilities of their password by a feedback mechanism such as a password meter to motivate them to enchance Security by choosing a stronger alternative. This perspective implies that consumers will adopt more secure password habits simply because they are aware of the risks (Egelman et al., 2013). These types of feedback meters can be dangerous, potentially instilling a false sense of security regarding password strength. The suggestion to use a strong password has always been present, but the meaning of strong passwords is still difficult to understand; you may go to Google and get a different measurement than Yahoo. This Challenge is often compounded by the increasing number of password-protected accounts the average user manages, estimated to be 25 accounts (Florencio & Herley, n.d.). This proliferation often leads to reuse of passwords across many accounts. Consequently, if one account is compromised, the other accounts will also be compromised simultaneously. There are many schemes for password cracking. Among the different types of password cracking, there are methods detailed by (Summers & Bosworth, 2004b):

1. **Dictionary Attack** – uses a file that contains most of the words found in a dictionary
2. **Brute Force Attack** – tries every possible combination of letters, numbers and special characters
3. **Hybrid Attack** - concatenates extra characters to dictionary words and tries different combinations.

A fundamental consideration when creating a password is achieving a balance; make the password difficult to guess without making it impossible to remember (Armstrong, 2003). This trade-off becomes particularly challenging in the context of managing multiple accounts, each requiring a complex, unique, and easy-to-remember password. The complexity of such can overwhelm an average human, risking forgetting the password unless they store it somewhere. This difficulty is fundamentally linked to the cognitive limitations of users, which involve managing multiple strong passwords at the same time (Gehringer, 2002).

Insecure password storage presents a significant vulnerability; If stored passwords get compromised, the strength of your passwords becomes irrelevant. That is why, instead of storing passwords, users tend to base their passwords on memorable words or personal names (Imperva, 2010), significant dates(Veras et al., 2012), song lyrics (Kuo et al., 2006), or personally meaningful objects (Ur, Noma, et al., 2015). Most of the time these tendencies comes from user attitude, identified by (Tam et al., 2010), where individuals acknowledge the general danger of having a weak password but are under a false sense of personal invulnerability ("it won't happen to me"). This false impression is not only limited to uneducated users; it also includes IT departments. The false impression that since someone works in IT-related fields, it makes them think they are protected against attacks. This raises concerns about where the responsibility for guidance lies, with some suggesting that IT departments should play a more direct role in security practices (Evers, 2006), recommending what kind of passwords to use and how you should treat them. Awareness is really important, which is something that gets mentioned a lot, but there is a reason for it. In recent years, there has been an increase in credential stuffing attacks associated with password reuse (Uddin et al., 2021). Credential stuffing attacks are an attack in which credentials are taken from a breach and used to log into another unrelated

service. This is alarming, as many individuals may find themselves in a similar situation, reusing the same password across many platforms.

1.3 Problem Statement

The primary objective of this research is to perform a quantitative analysis of the AuthInfo breached password data set to identify weaknesses and design, implement, and perform validation on a password strength checker called PassProof. The assessment logic behind PassProof is a direct result of the empirical findings.

In order to achieve this primary objective, sub-objectives were developed that can be followed step by step so that the end goal can be achieved:

1. Conduct a review of existing literature on password security principles, common user password creation behaviors and vulnerabilities, prevalent password cracking techniques, design and effectiveness of password strength meters, and finally how to analyze a password data set.
2. Define and document a clear methodology for preprocessing of the selected AuthInfo-Dataset to ensure that data quality remains for further analysis, including steps for structuring the data.
3. Use what was learned about different analysis techniques to perform a quantitative analysis of the prepared AuthInfo data set using the appropriate computational techniques. Get results such as average password length, character set usage, and lastly the most used passwords, and categorize them into top 500 most, with each entry having a volume of users that used that same password, pattern, or length.
4. Design and implement a scoring algorithm for the PassProof tool, ensuring that each rule and penalty is a counter to the vulnerabilities that were found in the previous objectives. Establishment of a security principle, such as Shannon entropy and length requirements.
5. Implement the PassProof tool as functional software using Python and libraries to assist in its design. Furthermore, have a good graphical user interface (GUI) capable of providing numerical feedback, as well as text feedback, with a visual strength indicator.
6. Conduct a final test and preliminary validation of the tool that was implemented; Try different test cases, such as known weak passwords, known patterns, and generally weak passwords, to see feedback and score. This is to verify the scoring logic and feedback are correct.

1.4 Research Methodology

This study has a mixed approach centered on quantitative data analysis, followed by the development of software artifacts and the preliminary evaluation. The design of the methodology is focused on empirically understanding the problem by performing a data analysis and then creating a practical tool based on the findings. The first phase was focused on the quantitative analysis of the pre-existing AuthInfo-Dataset. a technique was created using Python and leveraging libraries such as Pandas for data manipulation. To move quickly while keeping quality up, preprocessing was conducted on the data set, and then key characteristics such as the average password length, character set usage frequencies, the most common passwords, and most importantly, the most used pattern were identified by using masks. In addition, password complexity will be assessed by using metrics called Shannon entropy, which are used to calculate predictability. This specific phase addresses the objectives related to understanding the vulnerabilities in the dataset (Objective 2 & 3).

After the initial data analysis was performed, the next transition was artifact development, which was strictly focused on creating the PassProof password strength checking tool. It was important that the scoring algorithm was data-driven, meaning that the feedback rules are directly derived from the patterns and weaknesses identified during the analysis phase (Objective 4). The implementation will use the Python programming language and the CustomTkinter library to create a good-looking graphical user interface (GUI) fullfilling Objective 5.

Finally, an evaluation phase where testing was developed for the PassProof tool (Objective 6). This test focused on testing the scoring system to make sure that points are not reduced or given in the wrong places. Feedback needs to be accurate by giving the feedback based on what the user is missing. Throughout this phase, ethical consideration regarding sensitive data was maintained. A more comprehensive and detailed description of these methodology phases can be found in Chapter 3.

1.5 Scope and Limits

The scope of the research was to analyze the preexisting AuthInfo dataset (Güven et al., 2022). No further data collection was performed. The project also included the design, implementation, and creation of the "PassProof" password strength checking tool as a functional software artifact. The development included implementation using Python and the CustomTkinter library (Schimansky, 2025), this was integrated to have a multicomponent scoring system where the logic is a direct mirror of the results in the AuthInfo dataset. To make it better, a good-looking graphical user interface (GUI) was developed.

Acknowledging limitations is important in this research on the tool that was developed. One of the limitations was within the dataset. The password breach comes from specific sites that have their own password policies. The lack of a good policy would affect the weakness of passwords. Findings may not perfectly show us general findings for different policies. There could be policies that would also ban those passwords. Although the AuthInfo data set is large, it may not reflect the human habits of all users. The origin of where this data set is captured is unknown, which may not impact some places. The age representation is also unknown; it may be older people or young people.

We categorized the passwords into the 500 most common passwords and patterns, and this in itself is a limitation. There could be other patterns and guessable passwords hiding more in-depth that are not that popular but are as bad, and PassProof does not directly check for those. Masking is also an issue, since PassProof only captures character types and not character repetition and sequences, which in some cases is relevant to password strength. The scoring system gives specific points depending on the input (+6, +30, -50, etc.). These thresholds and having an entropy cap at 80 are based on analysis, interpretation, and security principles; they are not derived, and having a different scoring system can give different results. PassProof does not check for passwords related to usernames, service names, more dictionary words beyond the custom word list that was made, or keyboard patterns. All these limitations are something that can be worked on and fixed in future settings.

1.6 Document Structure

The structure of this thesis is as follows as a guide before jumping in:

- **Chapter 2 Literature Review**: This chapter presents a comprehensive review of existing academic literature relevant to password security. It covers topics that are relevant to the objectives, such as common user behaviors and vulnerabilities, cracking techniques, password meters, recommendations, and ethical considerations in data handling.
- **Chapter 3 Methodology:** This chapter covers specific research methodology in this study. It outlines the research design, how the Auth-Info dataset is used, pre-processing, and quantitative analytical techniques such as masks, frequency analysis, and entropy. It discusses the decision behind the scoring system selection and the software implementation process. Finally, and perhaps most significantly, ethical issues and dependability tests were also covered.
- **Chapter 4 Results/Analysis:** This chapter presents the core findings that were found in the quantitative analysis of the dataset, including common patterns and common passwords. It compares the findings with existing recommendations and findings in the literature review and proposes criteria for creating passwords that avoid common vulnerabilities.
- **Chapter 5 Conclusion:** This is the last chapter, which concludes the thesis by summarizing the entire research project, revisiting what was discussed, and also what can be done in the future with the project based on the findings and limitations.

2

Literature Review

Passwords are one of the fundamental aspects of authentication. Their effectiveness is often undermined by vulnerabilities that come from user behavior and systemic flaws. Human behavior, such as password reuse (Taneski et al., 2019) or having weak, easily guessable, and unreliable passwords, is one of the problems users are facing. All these issues come from having a poor password policy and depending too much on password meters. Resolving these issues is essential for improving cybersecurity in general.

2.1 Enhancing Password Security: A Comprehensive Study

Many solutions have been put together over the years; tools such as password strength meters and text-based requirement feedback are in place to resolve the underlying problem but are often badly designed, making it unnecessarily difficult for users, and most of the time they are flawed. Furthermore, there have been significant data breaches, including those at LinkedIn and Yahoo (Juels & Rivest, 2013), which show that even major companies do not follow secure password practices. The purpose of this study is to resolve these gaps between user behavior and organizational security problems and to make an effective password management tool. Understanding habits will aid in creating a more complex and simpler framework for creating passwords.

However, the limitation that was found early was the inability to monitor what others were doing while making passwords. The recommendation that was found was to use multiple passwords on various platforms to increase security. Human factors have been one of the greatest cybersecurity risks; users often choose simple passwords that are easy to remember, such as their birthdate or the name of a loved one (Pagar & Pise, 2017). You may see yourself using these types of passwords, but the dangerous part is that an attacker could use open-source information to uncover these sensitive data. Users could have their pet's name on Facebook, and if the password is easy, it will become a much easier target. So, it is important to avoid easily guessable passwords, such as your cat, dog, or personal information such as favorite food or where you grew up, as these are publicly accessible.

Many large companies are targeted, and some may believe that the bigger the company, the more secure it is. This is false, since companies like Yahoo should be bulletproof. All data is valuable in all companies, whether small or big. A recent data breach suffered by Yahoo!, Dropbox, and LinkedIn exposed 732 million user credentials (Experian Data Breach Resolution, 2017), this is concerning, as these companies are quite large and well-known, suggesting the need for security awareness.

Password composition is a technique that has been identified to prevent people from using easily guessable passwords. This technique requires the password to contain a symbol or digit to make it more difficult to guess. (Komanduri et al., 2011) examined this technique to determine how well it performs compared to a password system, such as a password strength checker. Their results showed that it did, in fact, strengthen the password without burdening the user. As an illustration, the password would become ILovemydog.30416 instead of ILovemydog.

However, attacks have gotten increasingly complex in recent years, so even if a password contains symbols or numbers, it could still be broken. If these organizations paid more attention to password checker security, many of the attacks that have been discussed could be prevented. For instance, in figure 2.1, Yahoo's password meter permits a password like abc.1234 to be medium strength even though it is theoretically weak. It becomes vulnerable because a brute-force attack is used as a method of password leaks. The brute-force attacker is aware that there are 10 numerals, 8 symbols, 26 lowercase letters, and 26 uppercase letters, and that the sum of them would be 26+26+10+8=70 (Singh, 2020), and each character has a probability of 70, given that the standard minimum password length is 8. Considering the computation, it will take 1.85 hundred centuries to solve. Regretfully, brute force is one of the most popular methods against authentication, despite it seeming inefficient. The fact that users create simple and predictable passwords is one of the reasons brute-force attacks are effective. Instead of guessing every letter, brute-force attacks employ a dictionary of the most popular passwords as their attack surface. These days, the time it takes to crack a password can be as short as a few minutes, depending on its complexity. The biggest fear is that once passwords are cracked, they can be used to obtain access, and compromise accounts on several sites. Its one trend that has been emerging and growing in popularity is password reuse (Gaw & Felten, 2006).

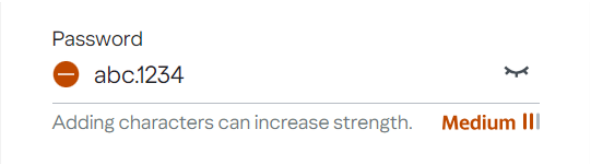

Figure 2.1: Yahoo weak password meter

2.2 Password Meters

Overall, having a password policy has been the best way to combat weak passwords. How they are developed really depends on the organization, but the simplest approach has always been to establish a minimum set of rules for the number of characters that can be used, followed by requirements for the types of character that can be used, such as symbols or numbers. Even exclude certain patterns, such as user-related information, something that involves your log-in, and keyboard shortcuts (Klein, 1990). A proactive password checker is a way to assess the strength of passwords. It operates by providing users with feedback on their passwords prior to account creation. These assessors, known as proactive password checkers, have existed for many years (Carnavalet & Mannan, 2015). Typically, a password checker is a bar with different colors for strong and weak passwords. When you select a strong password, the bar will turn green; when you select a weak password, it will turn red, and when you select a medium-strong password, it will turn yellow. Strengthening your password is a great way to improve authentication security.

In a recent assessment of users conducted by (Ur et al., 2012), they studied how meters can influence a user when they are constructing a password. They first looked at 14 distinct visual designs of meters on well-known and established websites. Initially, he examined password meters from the top 100 Alexa-visited websites. After that, he conducted an experiment with 2,931 Amazon Mechanical Turk users.

Two days later, participants were contacted again and requested to use the new password they had set in order to access the email service. This made it possible for the researchers to assess how strong the generated password was. They employed three different attack types, including Weir's cracking algorithm, to assess the strength of the password (Weir et al., 2009). The algorithm makes use of a wordlist including 14 million strings. They came to the conclusion that password meters assist users in creating more secure passwords.

This is relevant since the goal is to have an application that can provide feedback on the password that users are about to

set. Since the establishment of guidelines for what constitutes a strong password comes from this research, the focus will be to examine the dataset of compromised passwords in order to develop a comprehensive password system that will help users avoid password leaks in the future. Given the research aim to make the application useful and not a burden of any kind, it's crucial that the criteria and feedback be as simple and minimal as possible. In general, consumers were irritated by meters with overly stringent restrictions and became less concerned about meeting the meters' requirements (Carnavalet & Mannan, 2015).

2.3 Password Policy

Password policies are critical for enhancing security and preventing unauthorized access to sensitive information. Given that most people are aware of how crucial security is, why are these behaviors occurring? The study's aim is to make it safer so that passwords become stronger, making it more difficult to break. This is one of the reasons when building a password policy, it's vital to have some tight rules in order to have a constant and consistent policy while having adequate security.

However, people prefer simple tasks over complex ones; thus, some users find it difficult to adhere to the new password restrictions (Shay et al., 2010), For example, when organizations frequently change their policies, users might find it hard to adapt. Which makes it more difficult to create strong passwords. According to the study, 19% of users had already forgotten their new password after creating one for the first time under the new policy. This highlights that designing policies that are balanced between security and usability is important. A consistent, strict policy from the start may help users adjust and adopt strong practices without changing it all the time. Taking all of this into account, if the majority of websites are concerned about security, there is a reason why security breaches occur, and there is a reason why users are permitted to use simple passwords. Is it really that expensive to have a strong password system? Or is the lack of knowledge the reason why attacks occur? As demonstrated in Figure 1, Yahoo! permitted a straightforward password such as "abc.1234"; shouldn't most password regulations prohibit this kind of password? A decent password should be both "easy to remember" and "hard to guess," but it can be challenging to strike a balance between the two because an easy-to-remember password may be susceptible to dictionary attacks.

A good example of a **Strong vs Weak Policies** is implementing a policy that mandates the use of passwords that are simple to remember but difficult to figure out. This strikes a decent balance, but it's difficult because predictable passwords are easy to remember. For instance, 123456 is both easy to remember and highly predictable. Alternatively, a simple sentence can be made more complex by adding numbers. For example, **IL0veMyCat!!2025**, this is harder to predict yet easy to remember.

Despite having good policy, security breaches still happen. One reason is the concept of usability and security. Strong password systems often want good designs and resources, which may be expensive for some organizations, especially small companies. For example, many platforms allow users to reuse passwords across accounts; they know the risk of it. Reusing passwords enables attackers to compromise more than one account. A case involving LinkedIn where the platform was breached, when the attackers got access to users, they used the same passwords on different platforms.

To improve these challenges, the main idea involves a policy which has: **Minimum Requirements of characters and symbols**, **Ban common passwords**, and **Provide feedback.**

2.4 Blocklist

However, what are the best practices for passwords? One easy way for websites or policies in general is to use a blocklist. This works by keeping a list of the most popular, insecure passwords and preventing users from using them. For instance, the password in Figure 1, where abc.1234 was accepted, would be denied by the blocklist. Previous studies have demonstrated the effectiveness of this technique, finding that users who utilize blocklists typically produce more difficult-to-guess passwords (Ur et al., 2012), another great practice is to set minimum strength requirements. One goal one should all strive for is to create a password that is strong enough to be used on websites. Systems have changed throughout time; in 2010, 61% of websites had a minimum character length of 6, which was the most popular at the time (Bonneau & Preibusch, 2010). Ten years later, the criteria changed from six to eight characters. This could be because NIST instructed in 2019 that passwords should have a minimum length of eight characters.

2.5 A Password Generator Tools

In cybersecurity, password security is a major problem since users generate weak or frequently used passwords on a daily basis. In order to identify the habits that lead to vulnerabilities, a number of research studies have been undertaken in recent years on datasets of compromised passwords, making them susceptible to attacks. These databases provide us with information on what goes on behind the scenes. When assessing password weakness, it's critical to consider common patterns people utilize. This part of the thesis examines previous research that is relevant to the problem regarding password security, which is the analysis of authentication datasets with an emphasis on password strengthening and the discovery of common patterns.

A tool created by (Tsokkis & Stavrou, 2018) focused on personal information, the tool itself will profile the user by asking about topics; the user will enter their interests. It's a really good and user-friendly tool since it allows the user to add or remove fields. What a password generator does is it will potentially take the children names + any numbers, symbols, letters, dates, or key combinations. To test the tool that they created, they took 30 participants ranging from the age of 23 to 55; initially, they were asked to give a password they would create. What they found out was that most of them had personal information within the password. only 10% had random, 40% had date, and 23% had name within their password. Once they used the tool and were able to put in their information, they were asked to choose their strategy for creating a password, the tool implements 318 password strategies. lastly they were given their generated password. The program made it easier for researchers to compre-

Figure 2.2: Password generator tool by Tsokkis and Stavrou

hend the fundamentals of developing a tool that generates secure passwords. To make the experience more pleasurable, it is crucial that information from the user is present. This makes it more versatile by allowing the user to select what they want to include and allowing them to choose the type of strategy they want to employ.

Another tool that concentrates on creating passwords is (A.P.U. & V.S.S., 2023). The technique they employed entails creating passwords according to particular user-defined parameters using contemporary technologies like ChatGPT. Addressing the numerous difficulties of improving password security while preserving user ease was the primary goal of the study. This is significant because raising awareness shows the care for users, which is why this research exists in the first place. The technique ensures that the generated password is both strong and memorable by putting Human-Computer Interaction principles into practice. The entire idea behind A.P.U.'s password generation process is predicated on guidelines, including minimum length, character diversity, or even structure. This is a very clever method of dealing with users' reluctance to follow complicated password regulations. According to the study, people who comprehended the rationale behind the policy were more willing to adjust. This tool is excellent because it may be used for a variety of purposes, from personal accounts to accounts that aren't important.

Finally, the number of people who genuinely reuse passwords is one of the most important conclusions drawn from password breach research. Password reuse is a prevalent issue, as previously covered. In a thorough investigation, (Dhamija et al., 2021) examined more than 460 million compromised credentials. They discovered something concerning: a lot of people were using the same passwords on different websites. A breach in one service affects other platforms as a result of this behavior. (Bonneau et al., 2015) has investigated how weak password policies result from the repetition of passwords and slight modifications. People might believe it is unique and helpful to avoid using them again; however, modern password cracking tools can

readily exploit the base password by simply altering a few numbers.
With dictionary-based or brute-force attacks, increasingly complex password-cracking programs may swiftly break weak passwords. Numerous studies have shown that basic passwords that are used on multiple platforms can be cracked. Dhamija et al. (2021) demonstrated how these cracking programs can use common patterns, such as predictable word choices and numerical sequences, to find passwords from compromised datasets. According to their study's findings, the reason breaches occur is because easy passwords and password reuse are so risky that hackers may quickly and easily break into a huge number of accounts using basic cracking algorithms. This demonstrates the increasing sophistication of cracking tools and the increasing vulnerability of people. In order to stay current with cracking tools, maintaining the same level as cracking tools requires staying up to date.

According to the study's findings, consumers should begin utilizing password managers, which will assist in creating and saving passwords for them. These passwords will be distinct for every site, resolving the issue of password reuse and eliminating easy passwords. (Bonneau et al., 2015) advised to use multi-factor authentication (MFA) as well, since it will make it more difficult for hackers to access your account even if they have your password. MFA has enhanced overall security by demonstrating its ability to lessen the effect of hacked passwords.

2.6 NIST On Passwords

NIST has really helped the industry to strengthen security. NIST has recommended recently that companies should use a password manager to increase the strength of passwords; this could help employees to generate strong passwords and lock an account after multiple attempts. they don't even suggest the changing of passwords regularly because users who are asked to change their passwords regularly are likely to change to an easy password (Grassi et al., 2017). For instance, users may create strong and secure passwords the first three times, but eventually they will start to use less imaginative ones, and even if they do create strong passwords each time, they still have a tendency to forget them. At the end of the year, they will have created 12 passwords that they must remember if they generate a new one every month. Organizations shouldn't compel employees to update their passwords more than once a year, according to NIST's 2020 password guidelines. Additionally, according to a study conducted on participants, 82% of them felt that changing their password decreased the likelihood that someone else would access their account without authorization (Habib et al., 2018). Yet it was believed that setting a complex password, sharing it, or even using it again were more crucial than changing it too frequently. All of this is consistent with the recommendations made by NIST over the years. This suggests that users might be more willing to implement the advice if a security expert communicates more effectively. If implemented properly, security awareness programs can be beneficial.

2.7 Human Aspect of Password

Password security is greatly influenced by human factors, even beyond policies. Having a security policy won't necessarily encourage users to choose stronger passwords, according to numerous research. According to a study by (Tari et al., 2006), there is a general guideline on what makes a good password. they concluded that the guidelines by the Department of Defense for passwords made it a challenge for users' cognitive abilities and reduced productivity. This only proves that knowledge does not always give good password knowledge; users can know a lot but still be limited by human abilities. Some people may be really good, and others may struggle a bit. But one thing that has been consistent is that users and even IT professionals don't practice good password management. For example, in a survey done on IT professionals, 40% said that they write down their important business or passwords (Tam et al., 2010), Still, if the wrong person manages to obtain it, it could result in inadequate security or possibly a breach. Storing written passwords in a safe location is an easy solution that is often forgotten.

Although many people have minimal knowledge of security, it is theoretically assumed that IT specialists know what is best. Everyone, whether they are frequent users or professionals, should be aware of the security. Many consumers do not worry about security since they think a breach will not have any detrimental effects on them (Schneier, 2007), for example, users of web-based email systems don't want to remember a difficult password because the web host will be the victim of hacker attacks and not them. Why do users act in this way? It has frequently been stated that users' adoption of strong passwords is limited by their cognitive abilities, such as recall. A user may say, "If I have to, I can remember my password even if it is complex, but I would rather not put the mental effort into it." (Tam et al., 2010) This could be interpreted as being simply lazy. Users don't care about security concerns until they believe that they would be impacted if the account is misused, according to the survey. This conduct demonstrates that users' survival instincts do not activate prior to a breach. It is possible that all users share the same thought: why would somebody hack me out of all people? And after they make those weak passwords, they

get compromised, which is when the problem arises. Similarly to the email-based system, people did not want to remember complex passwords because they were not subject to the consequences.

Attacks known as “credential stuffing” occur when credentials stolen from a breach are used to access a different, unrelated service. This is concerning since many people can find themselves in a similar circumstance using the same password on numerous platforms. What causes this to occur? This occurs as a result of the widespread practice of password reuse, which has the following effects. Another method of password cracking is (Summers & Bosworth, 2004a): Dictionary Attack: employs a file that has the majority of dictionary terms in it. The Hybrid Attack concatenates extra characters into dictionary terms and attempts various combinations, whereas the Brute Force Attack tries every possible combination of letters, numerals, and special characters.

When a password expires, the recommendation is to change it to something stronger or comparable; however, that is not the case. Users usually simply change their existing password to deal with password changes; even when an employer implemented a password update, employees used different passwords. This was discovered and looked into by (Habib et al., 2018), They conducted a study examining how people viewed various aspects related to passwords. Password complexity, password storage, developing unique passwords that you don’t already use elsewhere, and changing passwords on a regular basis were the four guiding concepts. They discovered that people choose overly predictable tactics, and Zhang et al.’s work on password cracking supported the idea of reusing passwords.

In addition, they found that changing a password often does not make it stronger. Every person has personal information that they value; thus, it is crucial to consider what the users are attempting to safeguard. A corporate account or one that contains personal information is more valuable than a newspaper account. In the absence of a policy, people would select a weak password when creating one for a non-essential account. Although the account is irrelevant, research revealed that the presence of a password meter helped create better passwords (Egelman et al., 2013). Because every organization should have a password manager generator, this research is crucial because it will influence the password security policy to create more robust and comprehensive passwords. It is critical that a company have a strong expiration policy or one that advises users to change their passwords to something different from their old one if they are concerned that their account has been compromised. Keeping these in mind, Figure 2.3 shows the different aspects that impact how users decide on passwords.

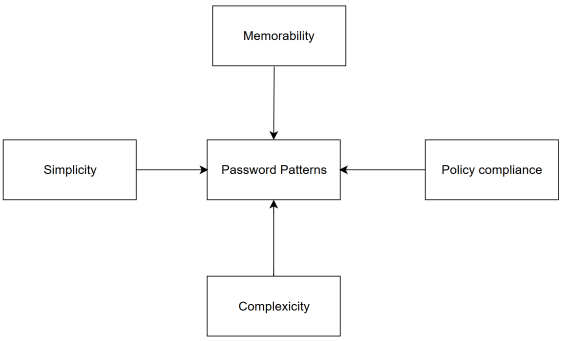

Figure 2.3: An example showing how different aspects impact password decisions.

One of the primary authentication methods that users will continue to utilize to access systems is the password. Weak password usage is causing more harm to the community and is the main cause of security breaches. Having a generator will help the community become more stable and prevent more breaches, and a robust password policy is crucial for the future. Users will be better able to understand the reasons for everything if they have awareness and evidence before receiving a tool. Although certain password choices may appear complicated at first, end users must be aware of the password construction techniques that could result in weak passwords (Tsokkis & Stavrou, 2018). However, that is not true. It can assist users in identifying and locating problems by providing them with real-world examples. The solution is useful because the goal is to have consumers use complicated systems in the near future while still using minimal energy. Since prior research has shown that password managers have experienced poor adoption rates, the wish is to increase password management (Smith, 2017). According to the study, none of the participants used password managers that were installed separately, but many used those that were integrated into their web browsers. The goal is to develop a strong password tool that can alter these facts. In order to prevent them by implementing a new policy, the study will analyze the data set from a breach to determine what trends exist in the password creation process. Hopefully, this study will have a solid foundation on what makes passwords weak and how

to improve them to prevent users from using weak passwords.

Important conclusions on password security, especially the use of weak passwords, are highlighted in the studied literature. It gives an exceptional opportunity to examine and expand previous research on passwords and user behavior, find patterns, and understand why some passwords are ineffective in protecting users due to breached datasets. The need for a good password-generating tool for teaching is important, as all of these studies show a continuous gap in user awareness and proper password management.

The use of blocklists, examining the types of human behavior that exist, and comprehending the policy by examining the types of passwords that passed through the website's password policy are some of the important conclusions drawn from the literature. Even while compromised datasets are frequently used maliciously, The study leverages them to develop a program that proactively stops the use of weak passwords for the greater good. This task is crucial since consumers won't receive real-time feedback on the strength of their passwords. All of these holes will be filled by the planned study, which will analyze a dataset of compromised passwords to find the underlying trends and patterns in the creation of weak passwords.
In conclusion, this chapter has examined the problems that arise from using weak passwords, examined the limitations of the technologies that are now available, and provided us with information on the motivations behind compromised datasets and human minds. The dataset analysis methods will be covered in detail in the upcoming chapter.

3

Methodology

The purpose of this study is to investigate a dataset of leaked authentication passwords using the AuthInfo-Dataset (Güven et al., 2022). This section outlines the research design, data collection and preprocessing procedures, analytical techniques, as well as reliability checks and ethical considerations. The dataset is a compromised password database, and the goal of this research is to find patterns and weaknesses in the passwords that users create. Since these are real passwords, the data are authentic and a good opportunity to evaluate and discover how humans think while creating passwords, specifically the focus on common pitfalls like reliance on sequential characters, keyboard patterns, personal information and transformations (for example how simple the substitutions or capitalization is) The ultimate goal is to leverage the data to develop a solution that promotes better password practices and increases security.
This approach section incorporates some of the principles and techniques utilized in previous works, including (Bonneau, 2012) and (Ur, Kelley, Komanduri, et al., 2015). Both of these studies have provided valuable insight into the importance of password analysis in identifying vulnerabilities and raising awareness of what lies behind those analyses. This chapter will provide a full explanation of the research technique employed to accomplish this goal. First, it includes a description of what was accomplished with the dataset and the preparation procedure and then a discussion of the analysis techniques that were utilized to examine the data. Since it's a breach, cleaning of the data was handled. Ethical considerations, such as privacy protection and compliance with data protection regulations, are addressed to ensure that this study does it in the right way and meets the responsibilities. Finally, returning back to key objectives and findings at the end.

3.1 Research Design

This research used a quantitative approach to analyze measurable characteristics of user-created passwords. The main objective was to evaluate the strength of passwords using statistical and computational methods, identifying the underlying patterns that could help us in the development of a better password policy and security practices. The research followed a descriptive design, where a summary of key features of the dataset was observed and focused on structural patterns of passwords, strength analysis based on entropy calculations, and insights into behavior on how users created passwords. This quantitative and descriptive approach was the most suitable given the large scale of AuthInfo-Dataset and the objective of identifying widespread trends and statistical properties rather than exploring other aspects like user motivations through qualitative means. On the other hand, the descriptive approach allows us to paint a clear illustration of the state of passwords within this breached dataset.

Before the research, a plan was developed on how to research. Creating a research design is crucial to conducting research

in an organized and efficient way. To achieve the goals that were defined in 1, methodology was refined to have different stages. Therefore, the methodology was structured into three main phases. The first phase was about data collection and preprocessing; this is really important when the dataset is large and unstructured. Having a clean and structured dataset is important for further analysis. The second phase consisted of detailed data analysis; in the section 3.3, it covers in more detail how the analysis was done. This is where various techniques like computational and statistical techniques were used to get more detailed information to identify password patterns and vulnerabilities. The final phase focused on addressing the ethical considerations and reliability checks to validate findings and ensure compliance with data protection and standards. It's important to stay within scope and follow strict laws and ethics. The figure 3.1 demonstrates the process that was gone through with the dataset. Mainly, phases 3 and 4 was in the cycle to find patterns in weak passwords.

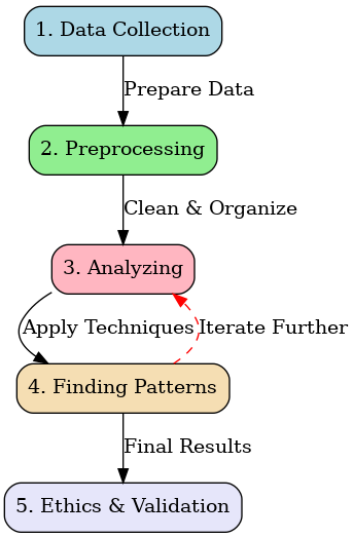

Figure 3.1: Illustration of the stages of research design, outlining the goals and methods of each step of the study.

3.2 Data Collections and Preprocessing

The AuthInfo-Dataset is the key asset and serves as the basis for this research. It stores passwords and their related information, providing insight into user activity. This study takes a quantitative approach, examining measurable aspects of passwords such as length, character types, structural masks, and counts of lowercase, uppercase, numeric, and special characters. It was created with 12 columns, each containing password information, and the user's username is salted to ensure anonymity. The structure includes plaintext passwords, password lengths, character types, structural masks, and other details such as the number of lowercase, uppercase, numeric, and special characters. The dataset was obtained in a CSV format and initially loaded into memory using Python's Pandas library, a database management system, for inspection and manipulation of data

Previous research has emphasized the need to study password datasets to identify patterns and improve security (Bonneau, 2012). These datasets provided us with valuable information about the types of passwords that humans create; this put us in the mindset of an attacker, examining how weak those passwords were. Before analyzing the dataset, a preparation procedure guaranteed that the data was good and accurate. Firstly, by deleting repeated passwords by grouping them into one column and counting how many of them there were. Specifically, near-identical strings were identified, counted, and then represented as a single entry with an associated frequency count. This helped us to reduce redundancy while preserving information about password popularity. Also, removing incorrect rows, such as those with null values or even zero-length made a significant boost to the workload. Beyond zero-length and null entries, potential encoding errors or entries containing non-standard characters that might go against the analysis were looked for, although such were rare. This simplified and improved the accuracy of the subsequent analysis; more about it will be covered in the coming chapters.

Temporary files were created during preprocessing to make sure the original data had its integrity. This practice made sure non-destructive workflow was achieved, and it allowed easy reversion back to the original data, thereby strengthening the reproducibility of the preprocessing pipeline. In the new temporary files, new columns were added for analysis purposes, such as a case-sensitive mask column to represent password structures. For example, "lBa123" was represented was "LULDDD".

These masks were used to represent different character classes. 'L' for lowercase letters (a-z), 'U' for uppercase letters (A-Z), 'D' for digits (0-9), and 'S' for special characters (for example, !"). This structured representation facilitated the classification and analysis of passwords in terms of patterns. The passwords was categorized based on length and mask, which enabled us to target analysis of weak and predictable password types.

3.2.1 Data Collection

The AuthInfo-Dataset is a repository of leaked passwords, which is the foundation of the data source. Given that this is a real-world password, it provided us with unique yet insightful information of user behavior. This phase was set in stone from the start, so there was no active collection of dataset. The foundation of the data is compiled from publicly known password breaches. This dataset's strength lies in its authenticity, it contains passwords used by real individuals, offering a direct view into user security practices. The act was solely on this existing repository. One limitation is the analyzed data was only a snapshot in time and may contain biases since it's a specific source. Having such biases could include demographic skew of the users on the breached platforms or what the specific password policies were enforced by those platforms in general at the time of the breach, which might influence the observed patterns. All these was acknowledged when looking for patterns and behaviors.

3.2.2 Preprocessing

Preprocessing involved critical steps and was one of the important steps in the methodology. The goal was to have a cleaner and reduce the size of the dataset. How would one achieve this? The task was simple: delete unnecessary rows, which are not relevant to us. This step was crucial because in this step saving time was the objective by cleaning up the data from the start. The result was a simple Python script, utilizing libraries such as Pandas for efficient data handling, like deleting null values and counting the length of each password. This simple script reduced the time to analyze the dataset and significantly reduced the dataset's size. The original Auth-Info Dataset was 1GB in size, and after preprocessing, it went down to 700MB. The removal of null values was paramount. Rows and columns that lack information gave no usable information when analyzing password characteristics, composition, and user tendencies. Including them would ruin statistics and potentially cause errors in the analysis, resulting in inaccurate results. Therefore, their removal was the most direct way to enhance data integrity for this specific research problem. This cleaned dataset, which is now devoid of clearly erroneous entries and added with password lengths, formed the reliable basis for the subsequent analytical techniques talked about in section 3.3

3.2.3 Analyzing and Finding Patterns

This phase was the one visited the most in all parts of the project. In the analysis, it was repeated just like what was done in Preprocessing by utilizing a straightforward Python script that is useful. A workaround was found to get the result in one file. It was important to have a single file exported out that could be used in PassProof. The best solution was to have a json file as a result of the analysis script. JSON (JavaScript Object Notation) was chosen for its human-readable format; its suitability fits us for representing structured data like lists of common passwords and pattern frequencies. The analysis script utilized the precomputed masks for the passwords in Auth-Dataset. This mask was a plan for us to identify what kind of pattern the user chooses for each character. By having each character classified, the understanding of when the user used lowercase, uppercase, and symbols was increased. By running the script and taking note of what pattern was prompted helped increase accuracy. So the process was going back and forth between analyzing and finding patterns. This was the approach to finding human behavior and essentially, banning it in the final product. Additionally, the most optimal way was to find the most common password; it was classified into the top 500 most used passwords alongside the most common pattern.

3.3 Analytical Techniques

After cleaning the dataset, normalization was used to make analysis easier. Normalization means in this case cleaning up the data. The quantitative methodology used in this study is based on measurable patterns and traits like entropy and character composition. Unlike an approach employed by (Ur, Kelley, Komanduri, et al., 2015), which reduced all characters to lowercase to simplify the analysis, this study retains the uppercase letters to allow for more accurate analysis. For example, a password like *mynameisoscar* may be weak, but having myNameISOscar can indicate an attempt to follow password policies (Renaud & Zimmermann, 2014). The segmentation process made grouping straightforward based on masks, length, and character variation. This segmentation was implemented programmatically, for example, by filtering the dataset based on the generated mask patterns (selecting all passwords that fit "DDDDDD" for 8-digit passwords) or isolating passwords shorter than 8 characters. For example, passwords with simple characters were separated from passwords with both characters and numerals, which were

then separated from passwords with all characters, digits, and symbols. One of the most important factors is password length, since research conducted by Bonneau has shown that longer passwords are more resistant to brute force attacks because they increase the number of possibilities they must crack. This enabled us to target specific password types, allowing us to locate only those that are weak or predictable, which was the primary goal to begin with.

To determine password strength, entropy calculations were employed. Entropy evaluates a password's randomness and unpredictability, with a high value indicating that the password is strong. Entropy is a reliable indicator for determining how durable your password is to attacks. Specifically, calculating Shannon entropy, a standard measure in information theory that is used to quantify the uncertainty of information content. The calculation is typically using the length of the password and the size of the character set potential (26 for lowercase and uppercase, 10 for digits, and 32 for special symbols), which is perfect because this mathematical algorithm helped the research find the predictability of passwords. Using such approaches and the features in the dataset with length and character diversity, parameters were identified that influenced password strength. For example, a password containing a special character mixed in with case letters will have a higher entropy value than a number password with no characters or digits.

To better understand user behavior, character composition analysis was used. It was done by counting the number of lowercase, uppercase, numeric, and special characters in the dataset. These counts were found for each password by going through its characters and incrementing counters for each class (lowercase, uppercase, digit, and symbols), allowing for more accurate statistics across the dataset. Additionally, to identify the weakest passwords, a calculation was conducted to find the frequency of each unique password string in the preprocessed dataset, and the top 500 most frequent entries were extracted. This list provided us with more concrete examples of common user choices.

3.4 Ethical Considerations and Reliability Checks

Preprocessing was carried out with a careful mentality to have an accurate and clean segmentation in order to guarantee the dependability of the results discovered. To ensure a consistent analysis throughout the study, an iteration process was conducted using the same entropy calculations and mask grouping structure in several analyses. This involved measurements such as rerunning analysis scripts to check for deterministic outputs and other information that could potentially ruin the logic for calculating key metrics like entropy and mask frequencies. To validate the technique and increase confidence, all of the data that were obtained were compared to earlier studies like (Bonneau, 2012) and (Ur, Kelley, Komanduri, et al., 2015). For example, the findings on the distribution of password lengths and the frequency of purely numeric passwords against the published results of these studies. There were similarities that were observed to increase confidence in the soundness of data handling and analysis methods.

Having all this sensitive data, ethical consideration is really important in all parts of the study. All usernames were already salted, making users anonymous, which prevents identification. Crucially, there were no attempts to reverse the salting process back to its original form. The security of personal information was the number one priority, which is why this research is done in the first place. The focus remained strictly on identifying patterns and human errors. Returning to the objective goals, comprehending patterns, and providing criteria to enhance the existing policies today. The study follows data protection regulations, such as GDPR to ensure lawful processing of the dataset. That's why ethical consideration is an important section to cover and maintain. Furthermore, temporary files were used in all parts of the analysis to maintain the original dataset. This allowed us to operate in a fast-paced environment while still maintaining a high standard. Additionally, transparency was preserved by having well-structured goals, methods, and conclusions. The research plan, the dataset, and ethical considerations were reviewed and approved by professors before the research was even started.

In conclusion, the AuthInfo dataset was studied, preprocessed, and analytical procedures were applied. The dataset was cleaned up by removing duplicates and inaccurate rows, case-sensitive masks were generated to represent password structures, and segments were created based on length, mask, and diversity. After that, Entropy was calculated to determine the strength of the password, and the technique ensured that ethical aspects were addressed. The most important thing was to follow the research instructions so that the analysis gets a lot easier; the findings are discussed in chapter 4.

4

Results/Analysis

4.1 Analysis

This section of the project focuses on the details of the process of the data processing and lastly the analysis part. PassProof has multiple factors when assessing passwords. factors like length, character diversity, entropy and pattern matching. All these are then taken into account when giving a user-friendly numerical score and feedback, giving some hints throughout the way to ensure that you can get to 100% at the end of the day. The rational reason for adopting a numerical scoring system (0-100) was because of its clear, quantifiable, and easily comparable password strength. Even so, in the background factors like entropy, length, and character diversity give specific insights; a single score offers users an immediate, high-level assessment without overwhelming them. It simplifies the complex aspects these factors have into an understandable value, allowing users to feel the severity of some password attempts. Furthermore, having a score gives the user a target to reach, which could motivate them to improve their password creation habits, which aligns with the research goal of promoting better security practices. When the user gives his input, the system will break the password, check if the password involves common passwords from the breached dataset, check if it has the same pattern, and reveal the strength or weakness in real time, meaning the strength is assessed immediately after the user clicks the *'Check Password'* button, rather than dynamically as they type. The discussion below will focus on the analytical logic and resulting outputs, highlighting the steps required to get to the end goal of a strong password policy.

The PassProof scoring system, which is implemented in Python has a complex yet easy-to-understand system. It is based on key security aspects and best practices identified in the analysis. Rather than relying on a single metric, it has a multicomponent approach. having categorized it into four core principles: *establishing base requirements*, *rewarding positive strength characteristics*, *penalizing known weaknesses*, and *integrating these factors into a final normalized score.*
Some requirements are non-negotiable minimums for fundamental security. The idea is to reward users for good behavior and penalize them if they use bad behavior. PassProof enforces this primarily through:

(a) Minimum Length Threshold. Any password less than 8 is rejected and is automatically assigned a score of 0. This is to combat short passwords against modern brute-force attacks, irrespective of other characteristics. In PassProof there is a function that checks if the input is lower than 0; it will give out feedback telling the user to enter a password with a length over 8 characters.

(b) Rewarding Positive Strength Characteristics. Following best practices is actively rewarded. Points are awarded based

on password length meaning PassProof recognizes the length and gives more reward the more length. Meeting the minimum (8-11 characters) will grant a base of 5 points. However, if complexity is sufficient (entropy score higher than 40), the points will increase by 8 points instead of 5. Having 12-15 characters increases the reward even more to 20 points, and after you exceed 15 characters, you will get the maximum points of 25.

(c) Character Diversity. Using a mix of character types will increase the reward you get, and this is essential to get full points. PassProof rewards this by granting 6 points for each of the character types (lowercase, uppercase, digits, and symbols), encouraging users to use all of them. A bonus of 4 points is given if the user decides to use a mixed-case alphabetic password even without numbers/symbols, and 8 points is given if you have the minimum length and you include both digits and special characters. The huge point poll is 30 points which is given if the user has both good length (12 or above) and has digits and special characters. Randomness with entropy is one aspect that gives a bonus. The calculated Shannon entropy provides a mathematical measure of complexity. This is added directly to the final score; this helps us to check the randomness of input and influence the score.

(d) Penalizing Known Weaknesses. While rewarding good behavior, bad behavior is also penalized. Characteristics associated with vulnerability attract penalties. Passwords characterized solely by one character type are easy to guess and brute force. So, a 30-point reduction is applied if the password contains only digits or letters. This is the result of analyzing the dataset; the number one pattern is the use of digits only. This battles pin code-like or simple word-based passwords. A more severe penalty of -50 is applied if the password matches the exact password in the top 500 common passwords or the most common pattern used in the dataset. This large penalty reflects the risk of having these passwords, since they have been used in real-world scenarios and have been breached.

(e) integrating these factors into a final normalized score. Finally, these positive and negative factors are combined, and you are left with a single score. The individual scores are summed to produce a raw score. The password is capped at 80 if you fail to include at least one character from each character type. This enforces the best practices, which require having all the types for maximum strength. The final score is classified from 0 to 100, ensuring an understandable output for end users. This breakdown shows how the scoring logic is built in PassProof with different factors like establishing minimums and rewarding positive strength characteristics, character diversity, penalizing known weaknesses, and integrating these factors into a final normalized score.

4.1.1 Gathering And Reading Data From The Script

Doing analysis manually is a big task, considering that the dataset has 1 GB worth of credentials. What the plan was from the very start and what was done in the methodology is the deletion of many unnecessary columns and values. This removed all columns that had null values, and it helped us reduce the file size. After executing a script, the new dataset's size has decreased by 300MB, to 700MB. Then a Python script was created called *analyze_data* which is a straightforward script that isolates the important column for the analysis. As researchers, the focus was to find the most used password, the most used pattern, and how many characters are used on average. All this will aid in making a policy to improve password security.

Two functions was created in the script, one called msk_passwords that focuses on finding masks in the dataset; the masks is grouped as discussed earlier. The letters are grouped after lower, upper, digits and symbol letters. This helped to identify the most used pattern so that it can be blacklisted in PassProof. The second function is called analyze_passwords. The config wordlist is the main source that was used to identify weak passwords; this wordlist has over 7000 entries of weak passwords. Analyze_passwords compared to the intial code about config worldlist, operates by analyzing the preprocessed AuthInfo dataset directly. It calculates the frequence of each of the passwords found in the datset, and classifies it into top 500 most used.

The function acts like a bridge to compare user input to the wordlist so that the blacklisting can be active, making the user lose points. Next a function called analyze_passwords that does all the analysis, like finding the average used characters, the most used password and the most used pattern. All this has generated a new json file when run. JSON (JavaScript Object Notation) was chosen beacuse of its output format, making it human-readable, its lightweight, and its ease of working with Python by using the json library. the json file has the structure of:

- **Avg_length** shows a single value of what the average count of characters used in the password is.
- **common_passwords:** the attributes contains frequently used real world password from the breached dataset; its extremely vulnerable, and its sectioned into top 500

- **common_patterns:** A set of masks to determine the structure the users used in the password; for example, ddddd means that all characters were digits. This is sectioned into top 500

This json file powers PassProof's ability to penalize inputs that are similar to those known risky passwords or passwords that follow the same pattern. From an analytical standpoint, reference to real-world evidence of user behavior is important. Having this in the empirical data is a key aspect of PassProof's design, aiming to provide valuable assessment that is similar to user behavior and how attackers' targets are closer than what a theoretical metric might imply. It makes the scoring system more trustworthy and not based on something purely theoretical; instead, it will reflect the patterns and credentials that attackers realistically will try during their brute-force attack. After cleaning, a mask was generated for each password to identify the most common pattern.

4.1.2 Characterizing the Password: Entropy

One of the primary functions in the PassProof tool is calculate_entropy, this function counts how many types of characters the passwords contain, whether they have lowercase letters, uppercase letters, digits, and special symbols. After calculating the efficiency of the character set (N), it will then multiply the password length (L) by the base-2 logarithm of N.

$$entropy = L \times \log_2(N)$$

From the perspective of analysis, what entropy does in this instance is merge two core elements of password strength.

- **Length:** A longer password makes brute-forcing it more difficult.
- **Character Diversity:** Combining letters, numbers, and special characters strengthens the password by expanding the number of possible combinations.

Where L is the password length and N is the effective size of the character set (26 for letters, 10 for digits, and 32 for symbols etc). A password will return 0 if it contains none of those recognized characters. Rather, if you significantly combine several categories, N will increase, providing a higher entropy.
But why does entropy matter? Throughout the research and design, combining these and giving a score is what the tool does best. PassProof can compare even short but highly varied passwords with long but uniform ones. By combining length and character diversity into one single mathemathical measure, entropy how resistant a password is to brute force, or guessing the password. For example, having a password that is **password123**, does not have uppercase letters or symbols. So the entropy might be lower than **P@ssword.123**. To understand how impactful this is, **password123** will give you 40 in score, while **P@assword.123** will give you 100 points. In another instance, if you have a purely numeric password like 123456, it will have a smaller N because only one type of character is used, which is digits. The code will not only score to shape an overall score, but it will also decide if an 8-11 character password gives 8 or just 5 points for the length score. This will help you get a decent score even though you have a medium-length password.

4.1.3 Key findings from Data Analysis

After finishing data processing, a more in-depth analysis was conducted to extract more information. The final result included the average length of all the passwords, the most used passwords, and lastly the most common password structure (pattern). All this was filtered into the top 500 to not overheat the computer. A mask was created for each of the passwords classified into lowercase (l), uppercase (U), digits (d), and special symbols (s). For example, if you have a password like Jb341!, it will be presented as Ulddds. The analysis showed that the average password length is approximately 8.25 characters in the data set. This aligns with what was talked about in the existing literature; specifically (Bonneau, 2012), after that recommendation, users still tend to use short passwords. That's why the system will reward users the more characters they use. With the analysis, what was found is that the most common password pattern was:

- Only Numeric Patterns **(dddddd), (ddddddd), (ddddddd)**,
- Only Alphabetic Pattern **(lllllll), (lllllllll), (llllll)**
- Alphabetic Numeric Pattern **(llddddddd), (lllddddddd), (lllllldddd)**

These kinds of passwords were the conclusion of the research, and they lead to password vulnerabilities. This is particularly terrible given that the attacker's target frequently used predictable password formats, (Taneski et al., 2019) supports that, talking about user tendencies when it comes to predictability, and using brute force to successfully attack passwords. For example the most used passwords included simple passwords that take seconds to brute force. The most common passwords were *123456*, *111111*, *123123* and alphabetic combinations like *a123456*, *123456a*, *1qaz2wsx* frequently used throughout the dataset. These passwords show how vulnerable password creation systems are. It shows how severe it is to reuse passwords, which was highlighted by (Gaw & Felten, 2006). This is an indication that people come to new webpages and uses the same password. And these simple ones are one of, if not the worst, passwords you can think of. Figure 4.1 shows 10 credentials that were concluded in the analysis with 490 other passwords.

Figure 4.1: Top 10 Pattern and Passwords in the dataset

The PassProof checker was implemented to address the vulnerability found in the dataset. Making up the system involved multiple stages, aligning closely with the objective and what was learned from the literature review and data analysis. One of the decisions that was evaluated was about the interface, which will be discussed in section 4.2.2. The discovery was that many users rely heavily on numerical characters, with many utilizing solely numerical passwords. The use of special characters and uppercase letters was limited, reducing the complexity and strength of many passwords observed. This lack of diversity is often not penalized by most password policies, explaining to us why some weak passwords might have been accepted by the original system where they were used and breached.

4.2 Implementation

This chapter will talk about the implementation of the PassProof password-checking application. It will explain the reasoning behind the design choices, some of the technical approaches done while in the development process, and the key trade-offs that were encountered along the way.
PassProof has a more modern design yet a robust password strength checker capable of giving education to users when creating stronger passwords. From the very start the fundamental objective was clear:

- Provide a good, and actionable feedback
- Use a real life data to reflect what is missing in the password
- Give a user-friendly and visually appealing interface

The final goal was not just to give out numeric scores but also to guide the user towards best practices in passwords. That's why even though you get 100/100 in score, the feedback will still tell you to increase the character length for even more security.

4.2.1 Design Choice

The purpose of the interface was to be clean and straightforward. To achieve this objective, libraries like CustomTkinter was used (Schimansky, 2025). This library was selected because of its modern widget styling and built-in dark mode support—a

popular feature among developers today because it reduces eye strain, and it's easy to set up and work with. CustomerTKinter provides a more polished and professional look than standard Tkinter, enhancing the user experience. The way it's built made it easy for us to make a prototype early and develop the desired layout quickly, focusing more on the core logic of the code rather than the UI.

A fancy real-time character counter was implemented that shows how many characters the user typed. This counter is dynamically updated on every key release within the password entry field, giving the user the length without requiring the user to wait or click on check password to get an update. This small usability feature saves time and subtly reinforces the importance of length in passwords. The interface also includes a strength indicator for feedback; this is what was talked about in chapter 2. A feedback bar, which goes up and down depending on the score you achieve. Finally, you will receive a pinpoint of the weakness in your password, including whether it lacks uppercase or special characters.
This was the plan from the very start and were inspired by (Ur et al., 2012). which demonstrated that real-time visual feedback enhances the ability to create stronger passwords by making the abstract concept of 'strength' more tangible to the conclusion that since users are actively participating, real-time visual feedback enhances password strength. *Specifically, the implementation of a numerical scoring system both as text and visually* because it provides feedback directly linked to the user's input upon clicking the 'Check Password' button. This choice is direct inspiration from usability principles and research (Ur et al., 2012).

The 0-100 scale is a good system in general since it's familiar to many users, and with that it includes a progress bar visually present, offering a 'gamified' game where it tries to encourage users to experiment more with different types of characters to 'fill the bar' and achieve higher security. This mapping of password characteristics makes the process of creating a strong password less complex and more engaging.

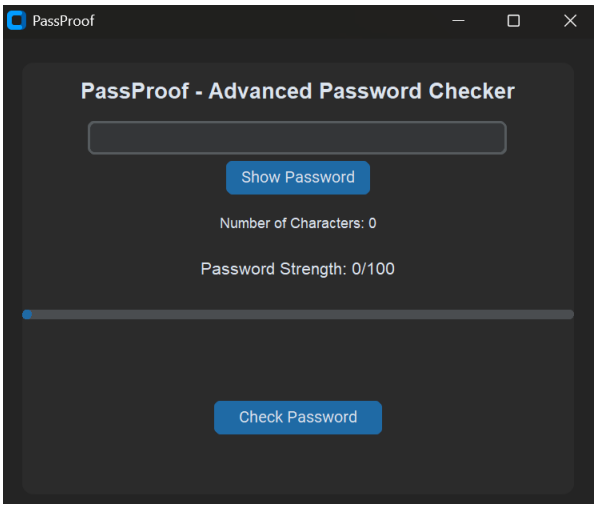

Figure 4.2: PassProof GUI Interface

Furthermore, when you enter a password into PassProof, users receives immediate feedback, showing the user the difference between a weak and a strong password. Passwords like abcd123 will score very low, with feedback telling the user what to do next. However, a more complex password like *Su3erC9mpl1xas!* will provide the highest possible score and motivate the user to go one step further and create a 16-character password.

4.2.2 Implementation Key Trade-Offs

The key trade-offs during development primarily involved balancing security analysis depth with complexity, usability, and performance. *Complexity and Usability* is one of the aspects that had to be focused on. Aiming at usability was more optimal since complexity will only drive users away rather than help users. For example, while implementing extensive dictionary checks against massive wordlists or sophisticated linguistic analysis could enhances security, they could have their limits, like significantly slowing down analysis and overly reducing creativity. Instead of telling the users that every password they type is in a breach, it will give simple feedback to change it more. Simple yet sophisticated passwords are preferred by certain users, the final product prioritizes insightful feedback rather than providing no feedback, which forces the user to assume what the password issue is. The tool has great performance since it's packed with real-time feedback gotten from the loaded dataset. The JSON data storage gave us a way to look up information and have a minor loading time before you get your

score; this is good for user experience. *Ethical Data Handling* was considered because these data have sensitive data and are a real-life breach, which raises privacy concerns. To comply with GDPR regulation, the dataset was already anonymized without usernames and generalized pattern information that only provided us with upper, lower, and symbol results. These trade-offs were thoroughly examined in order to have a solid objective and match the pre-existing literature by (Shay et al., 2010), prioritizing both user experience and robust security at once.

4.2.3 Testing PassProof

One of the most used password is *wmsxie123* how will PassProof react to this password? In Figure 4.3 PassProof will give a score of 0 and give the user feedback telling them that the password is commonly used and that they should use another password or change it more. This is very insightful and easy to understand. The program will compare each password the user enters with a list that was already created during the study. If the pattern is the same or very similar, it will provide feedback, and the user will receive a score of zero. This shows that the tool is user-friendly, and all that is needed is for the user to run the program and start creating passwords. Some programs require an instruction file so that the user knows what to do and how it operates.

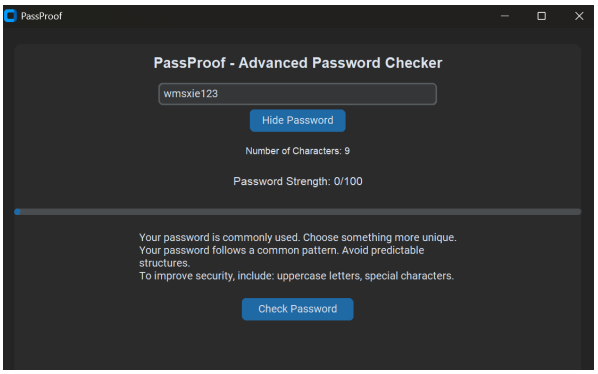

Figure 4.3: Example of common password and common pattern

The score of 0 is due to the large -50 points reduction for pattern penalty being applied because the password *wmsxie123* was identified inside the top 500 most used common passwords. Furthermore, its structure being *lllllddd* matches one of the top 500 most used patterns giving it the penalty as well. The feedback shown in the figure being ”Your password is commonly used“ and ”Your password follows a common pattern“ is accurate, and below it you have tips to improve it by telling the user to include uppercase letters and special characters.

This chapter gave an insight into key findings from the analysis of the AuthInfo dataset; it established the landscape of common password weaknesses. It then had a section talking about the analytical engine of the PassProof tool, explaining how empirical data was integrated and how the multi-component scoring system evaluates passwords by using different aspects like entropy, etc. Finally, the discussion of the practical implementation of UI design choices, technical trade-offs and more. This structure shows that a data-driven problem got the result of a great, evaluated solution to the problem. Coming sections will offer a broader discussion of these results, with future works and more.

4.3 Discussion

This chapter will focus on discussing what is presented in the research; reviewing all the phases, meaning the analysis, development, and implementation of the PassProof password checker. This chapter aims to interpret the results with broader content and link it to Chapter 2. It will critically evaluate user habits and compare them with the pre-existing literature. Furthermore, recommendations will be covered derived from the research and about the limitations the tools might take up. Finally, this discussion will be linked back to chapter 1.1, and an estimation will be provided of the extent to which it has been achieved.

4.3.1 Interpreting of key findings

While the analysis is one part of the research the understanding of the work is important. Interpreting these findings in comparison of the existing literature provides a valuable context and highlights challenges that were faced in password security. It will

show how some of the findings in the literature review are still relevant some years later in 2025. Some of the literature review is papers done several years before but still relevant because of the lack of the ability to improve and fix mistakes.

Average Password Length

The finding that the average password length was just above eight characters heavily supports early studies like (Bonneau & Preibusch, 2010). It was highlighted that in 2010, 61% of websites had a minimum requirement of 6 characters, but after ten years, it had changed from six to eight characters, and it might be because of what NIST instructed in 2019, that the recommended mimumum character should be eight. What (Bonneau & Preibusch, 2010) found that many users aim to only meet the minimum requirement imposed upon them. This is likely because of what was discussed in Chapter 2, including that cognitive abilities can be a factor meaning users prefer shorter, easier-to-remember passwords. While 8 characters meets NIST minimum requirements (Grassi et al., 2017), it offers significantly less resistance to modern cracking techniques compared to password lengths that are over 12 characters. The analysis does complement what Bonneau et al said, and it aligns with NIST recommendations. But even though it does aliign with them the security was still breached. In figure 4.4 shows the most used

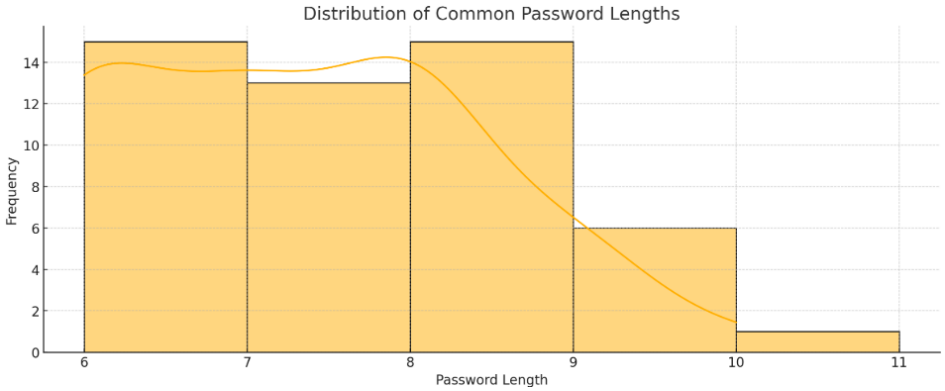

Figure 4.4: Most used length

length.
What is concerning is that there are almost the same number of people using 6 characters as 8 characters. Which shows that some users still live in the 2010 era. The graph and analysis show the top 100 most used, and of them, 14% of users use more than 8 characters, and none use 12 or longer. The recommendation is to have a minimum of 12 characters, but PassProof will still let you have 8 characters and give you a bad score if it's weak. throughout the process of creating the password, it will always encourage the user to use 12 characters to achieve 100%, and take it even further to encourage them to try 16 characters. 8 characters will never give you a full score; the objective is to help prevent what happened to those users, and part of that is to even upgrade the length.

The Use of Simple Patterns

The identification of purely numeric (dddd), lowercase alphabetic (lllllll), and mixed passwords (lllldddddd), shown in Figure 4.1, strongly aligns with what was discussed by (Taneski et al., 2019) regarding user predictability. He talked about users relying on simple, easily guessable passwords. The use of digits only can indicate the use of dates, phone numbers, or simple sequences that can be easily obtained by the attacker. Most users use lowercase letters only. When segmented into the top 500 to make it smoother and easier to manage, there were few users who used uppercase letters. Most of the passwords were lowercase letters or one type. This lack of structural diversity increases predictability and reduces the search space for attackers when they brute-force passwords, as discussed in Chapter 2. This demonstrates a failure by either the user or the original site policies to encourage a stricter and more complex policy. The recommendation is to have a mixture of all the types to make it harder for attackers; the more unpredictable you are, the longer it takes for them to crack your password. PassProof will at each step recommend to you in real time what you are missing; you must have an uppercase letter, a lowercase letter, and special characters to obtain 100%. In Figure 4.5, it shows the most common patterns.
However, having a mixed pattern is the most used. The lack of upper letters ruins it, and the length is really low. Combining Figure 4.4, and Figure 4.5 puts it into perspective that there is a lot that is missing that can be fixed. There are more than five patterns that are just digits and letters only. Keep in mind that there are not only five people that are using this; this is the top 500 of the most used patterns, meaning first place has 1,418,286 people using the pattern (dddddd).

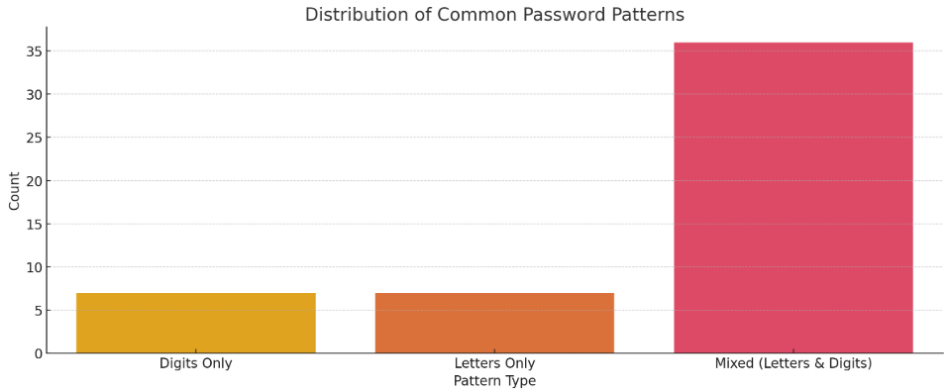

Figure 4.5: Most used patterns

Prevalence of Common Passwords

After finding the weakest passwords, like *123456*, and *111111*, it was beyond devastating. This finding is unfortunately very consistent across many password breach analyses. The extreme use of 123456 mirrors its ranking in the global list, showing its universal poor practices. Other variants like *wmsxie123*, *a123456* which are slightly more complex but still very common passwords. This reinforces the idea that many users create passwords based on simple rules or simple sequences they can remember. These types of passwords are blocklisted in PassProof, which is something that was recommended by NIST in chapter 2. Crucially, its high use of these passwords can be an indication of shared passwords, which was the concern raised by many researchers that we already talked about such as (Gaw & Felten, 2006), and (Dhamija et al., 2021) regarding password reuse. Each entry in 123456 shows a potential access point for credential-stuffing attacks on multiple platforms, showing a risk created by individual choices. These types of passwords are usually the first passwords attackers use since they are widely reused. The recommendation is to always think twice when creating a password and have a second variation ready that is complex compared to the first one you came up with. This makes the password unpredictable and not used by many. Avoid using personal information that can be guessed, and dont use the same password on all platforms, which increases the risk of getting breached in multiple places at the same time.

4.3.2 Recommendations

Based on the research and findings of the AuthInfo analysis, the following recommendations are proposed to both end users and developers and organizations:
End users:

(a) Prioritize Length and Complexity. Aim to have more than 8 characters. By using PassProof, you will come up with a strong password that is at least 12 characters. Make sure to use a mix of all four character types (lowercase, uppercase, digits, and symbols) to ensure you are well protected against brute-force and other attacks. To get a full score on the PassProof tool, this is one of the requirements.

(b) Enable Multi-Factor Authentication. Just in case you are hacked, it is always better to have another layer of security. Multifactor authentication is a second layer of protection in case your password is breached. It can be in the form of a text message password or a third-party application confirmation that it is you and not an attacker. For the most part, MFA is really hard to bypass.

(c) Utilize feedback. When using PassProof or similar ones, make sure to always pay attention to feedback like password meters and text feedback that tell you what you are missing. This is a way to show the application that you want better security, and in return the application rewards you by protecting your password from future breaches. In PassProof, feedback is crucial since if you do not follow it, you will not receive a full score.

(d) Unique Password. Try to create a unique password on important online accounts; you could make an advanced password with PassProof and have a password manager to store all passwords. One of the issues with weak passwords is that the

passwords that users make are easy to remember, which makes them weak. This practice allows them to reuse the same password across various platforms, leading to the misconception that having a single password for all platforms is optimal, which is misleading.

(e) Avoid Predictability. Don't use personal information that can be directly sourced back to you. Don't use easily guessable passwords with common patterns; also, stay away from common keyboard sequences.

For developers and organizations implementing a new password policy system:

(a) Longer Minimum. Instead of having 8 as the minimum number of characters, take it up to 10 or 12. Having more means a stronger password. Brute forcing will take longer since there are more characters that need to be cracked. The priority is to mitigate potential breaches; it is advisable to enforce a minimum requirement of at least 10 characters, thereby strengthening the system's security.

(b) Create a custom block list. Have your own block that involves known passwords; create your own database that updates on breaches and block all passwords that come up in that breach. Exactly like what PassProof does. This ensures that users know what passwords to not use; having feedback informing users that their password has been compromised in a data breach is quite helpful.

(c) More narrow feedback. Instead of giving broader feedback, give more specific feedback and guidance rather than vague strengths and simple rejection, leaving the user to wonder what the issue is. Instead of saying 'this password is weak', it is more efficient to specify the issue; 'include upper characters' is a better way to give constructive feedback.

(d) Ban certain patterns. Ban passwords consisting of only one character type. As discussed, having one type of character is a weakness, and one should use all four character types to increase the complexity. It is not difficult to include all character types, so minimal effort to get something better in return.

This chapter provided an in-depth discussion on the key findings of the AuthInfo-data set analysis, linking them to the existing literature review. Comparison of vulnerabilities observed by other researchers with this research on user behavior, cracking techniques, and policy effectiveness. All of this played an enormous role in the PassProof design choices, particularly its scoring system and data-driven block list, and these were evaluated based on best practices and the literature on password security. Having a third view led us to give recommendations to users and developers to improve password security hygiene and system design. Finally, there was a discussion of the limitations of the dataset, analysis methods, scoring system, and the tool itself to show that limitations are an important aspect to be acknowledged. This discussion merges two aspects of the presentation of results and their challenges when it comes to security.

5

Conclusion

5.1 Introduction

This last chapter draws together the preceding chapters presented in this thesis. In chapter 1, the discussion involved the pertinent topic of the utilization of weak password usage and highlighted limitations in existing recommendations. Chapter 2 offered a review of the relevant literature, and Chapter 3 described a comprehensive methodology used to analyze the data set. Chapter 4 then presented the results obtained by implementing the PassProof tool. Following this, there was a detailed discussion and interpretation of the findings in relation to existing knowledge, which is crucial because it led us to make recommendations and criteria for a good password policy. This chapter will focus on summarizing the key aspects of the research that was conducted, reviewing the research objectives to evaluate whether the goal was met, and proposing potential improvements for future research.

5.2 Summary of Research

This research had a mixture of methods to address the vulnerabilities found in password practices. Chapter 2 was the foundation for getting theoretical background by reviewing the existing literature on the fundamentals of passwords, common user behaviors that contribute to weak passwords such as password reuse or predictability, password cracking techniques, how security controls like password meters and policies have vulnerabilities, and ethical considerations when dealing with the data set. Chapter 3 contributed to outlining the research methodology. The chapter outlines and explains the methodology that was used to design and execute the quantitative analysis in the dataset. Furthermore, it highlighted the technical aspect of the analysis, which included frequency analysis, pattern masking, and entropy calculation, and finally how the planning of the PassProof tool was when developing the tool based on the findings.

Chapter 4 identified the core result and implementation in detail. The chapter is one of the most important chapters to make sense of what the research concluded. The examination of the dataset revealed core key findings, including average length, common patterns, and common passwords. The main goal of this chapter was to improve the security by implementing stricter standards to avoid common user errors. The issue at hand was already on our mind, and there was some sense of what the issues were in user behavior. In addition, having implementation of a predictability scoring system that prevents users from receiving the full score if their password is weak. The result chapter also involved the implementation of the PassProof tool using Python and libraries to aid us, such as Pandas and CustomTkinter, explaining the architecture and the logic derived from the

results. Subsequently, Chapter 4.3 delves into a comprehensive discussion, comparing all the findings with the literature review and giving a conclusion of the PassProof policy applicable to both end users and organizations. User suffering is something to avoid, so having password creation criteria will just improve us as a society.

5.3 Research Objectives

The research was guided by the objectives we covered in Chapter 1.3. These objectives were addressed as this:

1. Objective 1 was directed towards literature review and was about conducting a comprehensive review of existing literature. This objective was met early in Chapter 2. Chapter 2 provided an extensive review covering password security, user behavior, password meters, policy analysis, and breach data ethics, which assisted us in understanding the core principles and weaknesses of the research.
2. Objective 2 was to develop and document a methodology for pre-processing and the different stages that were undertaken. This objective has been met; Chapter 3 gave an outline of the steps taken while working with the data set to ensure data quality and prepare it for analysis. Figure 3.1 outlined
3. Objective 3 was directed to the execution of a quantitative analysis, extracting key metrics, and getting the top 500 of important information from the dataset. This objective was met, which was presented in Chapter 4. In this Chapter it described the results, including average length, character usage insights, and top 500 most common and most used patterns (Figure 4.1
4. Objective 4 was to design a multicomponent password scoring algorithm, ensuring that the logic is derived from the results and identifies weaknesses. Objective 4 is met. Chapter 4 describes the implementation of PassProof using Python and libraries to create the scoring algorithm. It also discusses how the scoring system works and the consequences of not following it.
5. Objective 5 was the implementation of the PassProof tool, developing a GUI, and also designing a scoring algorithm. This objective is met; Chapter 4 describes the implementation of the PassProof tool and the handling of GUI (Figure 4.2). It confirms scoring logic and the existence of functional code as an artifact.
6. Objective 6 was to conduct a functional testing, to verify the scoring logic and feedback. This objective is met. Chapter 4 has a section about testing various inputs, including a common password input, and patterns to demonstrate the tools ability to handle input.

)

5.4 Research Contribution

This research contributes to the use of password security on several levels:

Empirical Analysis. It provides an analysis of the AuthInfo-Dataset, identifying weaknesses found by us. This includes documenting common passwords, common patterns, and prevalent structures (lowercase letters, uppercase letters, digits, and symbols), and finally, average length. The findings in Chapter 4 gives us examples of trends within users, which is a direct comparison to the literature review in Chapter 2. They go hand in hand regarding common user password weaknesses and support what was concluded.

Data-driven tool. It shows a practical methodology for taking in information from a breached password dataset and directly designing a tool to combat the weakness. There is a link to every section of the research leading back to the tool that was developed. The pattern and common password that are found have a specific penalty within the PassProof scoring logic, which makes it a data-driven approach.

PassProof Artifact. The tool itself serves as a proof of concept, demonstrating how someone can implement a great tool battling common weaknesses alongside having standard metrics behind the scenes like length, complexity, and entropy into a user-friendly checker giving actionable feedback with a score, progress bar, and advice. It offers so much with just a simple interface.

In essence, this research contributes by analyzing existing literature and doing work based on that to understand and develop a ”bridge“ between what was found in the dataset and provide users with practical, data-informed guidance while making a new password through the PassProof tool.

5.5 Future Work

Building upon the work is important since it refines the general mindset, and the result becomes even more accurate after reviewing the potential limitations in Section 1.5. There are several aspects that can be developed further. *Expanding Blocklist*, *Pattern Analysis*, *Enchaning Scoring Algorithm and Checks*, *Larger Usability testing*, and *Suggestion function* are some of the aspects that have been considered. Let's go through them quickly:

5.5.1 Expand Blocklist and Pattern Analysis

In the current tool, it's relying on a top 500 most used list, and it's a limitation. Future work could include to increase this list into a huge list to cover more breaches, since AI is on the rise. Integrating AI to find breaches and add them to a breach list is a very good idea that could be optimized and be efficient. There could be an expansion to create a more comprehensive block list for common passwords and patterns. Additionally, there could be a new technique that checks for sequences, repetitions, or keyboard patterns that are not captured in the current version. This research is something researchers are passionate about, and all these creative ideas imply the passion and how this is achievable within a short amount of time.

5.5.2 Pattern Analysis

The scoring system is a specific one that tries to mitigate the problem that was found in the result. Expansion on datasets or user studies is possible, and then check for more when looking for weaknesses. Having a system where the user can log in and provide some information about themselves would help us detect contextual information to reject personal information and detect common keyboard patterns. This can be achieved by making a database that has many entries of common personal information to inform the user not to use that. For example, having the password *Cat.3214851* is a bad choice of password; its relation to personal information is irrelevant in this case.

5.5.3 Enhancing Scoring Algorithm and Checks

There can always be improvement in the scoring algorithm; the algorithm is using many components already, but what can be done in the future is reduce the points in more entries than the top 500. Even though the top 500 covers a lot of the data set already, it can always be increased and make it more efficient at the same time. One good idea is to also make regular updates to the scoring algorithm while new trends arise in today's society. If it can be incorporated with AI, it's a big improvement; it will be 10x better and also up-to-date on today's market.

5.5.4 Larger Usability Testing

Beta is a smart concept; there could be a period of time where PassProof invites users of different ages to test it out and give us feedback. This is typically what video game developers do before the official launch to grasp what the customer thinks. This would evaluate how effectively users understand the PassProof feedback feature, whether it helps to create a stronger password, and identify issues with the interface or feedback mechanism. There could be a deployment on a website system where users can log in so that PassProof can see how the system handles a lot of traffic at the same time to optimize and be more efficient

5.5.5 Suggestion Function

Lastly, having a suggestion function is one of the most helpful functions that a user can have. It will guide a user to choose a good password based on their input. Helping users is essential; that can be achieved by analyzing the user's input and providing them a good, easy-to-remember password suggestion that they can apply right away. Having this alongside, password management can also help the user store all the suggested passwords, so if they want to review which passwords have been suggested, they can always go back and apply them again. This management function would also help users store passwords from different platforms so that they can easily manage them.

References

A.P.U. & V.S.S. (2023). Balancing password security and user convenience: Exploring the potential of prompt models for password generation. *Future Internet*, *15*(9), 327. https://www.mdpi.com/1999-5903/15/9/327

Armstrong, I. (2003). Passwords exposed: Users are the weakest link. *SC Mag*.

Bonneau, J., Herley, C., Van Oorschot, P. C., & Stajano, F. (2015). The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. *Proceedings of the 2015 IEEE Symposium on Security and Privacy*, 553–567. https://doi.org/10.1109/SP.2015.44

Bonneau, J. (2012). The science of guessing: Analyzing an anonymized corpus of 70 million passwords. *2012 IEEE Symposium on Security and Privacy*, 538–552.

Bonneau, J., & Preibusch, S. (2010). The password thicket: Technical and market failures in human authentication on the web. *WEIS*.

Carnavalet, X. D. C. D., & Mannan, M. (2015). A large-scale evaluation of high-impact password strength meters. *ACM Transactions on Information and System Security (TISSEC)*, *18*(1), 1–32.

Chanda, K. (2016). Password security: An analysis of password strengths and vulnerabilities. *International Journal of Computer Network and Information Security*, *8*(7), 23.

Dhamija, R., Jain, R., & Agrawal, A. (2021). The next domino to fall: Empirical analysis of user passwords across online services. *Proceedings of the 2021 ACM Conference on Computer and Communications Security*. https://people.cs.vt.edu/~pds2/papers/password_reuse.pdf

Egelman, S., Sotirakopoulos, A., Muslukhov, I., Beznosov, K., & Herley, C. (2013). Does my password go up to eleven? the impact of password meters on password selection. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2379–2388.

Evers, J. (2006). Security expert: User education is pointless. *CNet News*.

Experian Data Breach Resolution. (2017). DATA BREACH INDUSTRY FORECAST [Accessed: Month Day, Year].

Florencio, D., & Herley, C. (n.d.). A large-scale study of web password habits [Accessed: 2024-9-19].

Garfinkel, S., & Spafford, G. (2003, March). *Practical UNIX and internet security* (3rd ed.). O'Reilly Media.

Gaw, S., & Felten, E. W. (2006). Password management strategies for online accounts. *Proceedings of the Symposium on Usable Privacy and Security (SOUPS)*.

Gehringer, E. F. (2002). Choosing passwords: Security and human factors. *IEEE 2002 International Symposium on Technology and Society (ISTAS'02). Social Implications of Information and Communication Technology. Proceedings (Cat. No. 02CH37293)*, 369–373.

Grassi, P. A., Garcia, M. E., & Fenton, J. L. (2017, June). *Digital identity guidelines* (tech. rep. No. 800-63-3) (Includes updates as of 03-02-2020). National Institute of Standards and Technology (NIST). Applied Cybersecurity Division, Information Technology Laboratory, National Institute of Standards, Technology; Altmode Networks, Los Altos, Calif. https://doi.org/10.6028/NIST.SP.800-63-3

Güven, E. Y., Boyaci, A., & Aydin, M. A. (2022). A novel password policy focusing on altering user password selection habits: A statistical analysis on breached data. *Computers Security*, *113*, 102560. https://doi.org/https://doi.org/10.1016/j.cose.2021.102560

Habib, H., Naeini, P. E., Devlin, S., Oates, M., Swoopes, C., Bauer, L., Christin, N., & Cranor, L. F. (2018). User behaviors and attitudes under password expiration policies. *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*, 13–30.

Imperva. (2010). Consumer password worst practices [Accessed: [Insert Date You Accessed the Document]]. http://www.imperva.com/docs/WP_Consumer_Password_Worst_Practices.pdf

Juels, A., & Rivest, R. L. (2013). Honeywords: Making password-cracking detectable. *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS '13)*, 145–160. https://doi.org/10.1145/2508859.2516671

Klein, D. V. (1990). Foiling the cracker: A survey of, and improvements to, password security. *USENIX Security Workshop*.

Komanduri, S., Shay, R., Kelley, P. G., Mazurek, M. L., Bauer, L., Christin, N., Cranor, L. F., & Egelman, S. (2011). Of passwords and people: Measuring the effect of password-composition policies. *Proceedings of the sigchi conference on human factors in computing systems*, 2595–2604.

Kuo, C., Romanosky, S., & Cranor, L. F. (2006). Human selection of mnemonic phrase-based passwords. *Proceedings of the Symposium On Usable Privacy and Security (SOUPS)*.

Pagar, V. R., & Pise, R. G. (2017). Strengthening password security through honeyword and honeyencryption technique. *2017 International Conference on Trends in Electronics and Informatics (ICEI)*, 827–831.

Renaud, K., & Zimmermann, V. (2014). Ethical guidelines for password research. *arXiv preprint arXiv:1405.3831*.

Schimansky, T. (2025). Customtkinter: A modern and customizable python ui-library based on tkinter.

Schneier, B. (2007). Information security and externalities [Discusses security trade-offs and the concept of negative externalities in information security.]. *ENISA Quarterly*. https://www.schneier.com

Shay, R., Komanduri, S., Kelley, P. G., Leon, P. G., Mazurek, M. L., Bauer, L., Christin, N., & Cranor, L. F. (2010). Encountering stronger password requirements: User attitudes and behaviors. *Proceedings of the sixth symposium on usable privacy and security*, 1–20.

Singh, U. (2020). Password security: Method and techniques to avoid breaches.

Smith, A. (2017, January). Americans and cybersecurity [Pew Research Center]. http://www.pewinternet.org/2017/01/26/2-password-management-and-mobile-security

Summers, W. C., & Bosworth, E. (2004a). Password policy: The good, the bad, and the ugly. *Proceedings of the Winter International Symposium on Information and Communication Technologies (WISICT)*, 1–6.

Summers, W. C., & Bosworth, E. (2004b). Password policy: The good, the bad, and the ugly. *Proceedings of the winter international synposium on Information and communication technologies*, 1–6.

Tam, L., Glassman, M., & Vandenwauver, M. (2010). The psychology of password management: A tradeoff between security and convenience. *Behaviour & Information Technology*, *29*(3), 233–244.

Taneski, V., Heričko, M., & Brumen, B. (2019). Systematic overview of password security problems. *Acta Polytechnica Hungarica*, *16*(3), 143–165.

Tari, F., Ozok, A. A., & Holden, S. H. (2006). A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords. *Proceedings of the second symposium on Usable privacy and security*, 56–66.

Tsokkis, P., & Stavrou, E. (2018). A password generator tool to increase users' awareness on bad password construction strategies. *2018 International Symposium on Networks, Computers and Communications (ISNCC)*, 1–5.

Uddin, M. A., Strufe, T., & Rieck, K. (2021). Detecting credential stuffing attacks in web traffic. *ACM Transactions on Privacy and Security (TOPS)*, *24*(4), 21:1–21:31.

Ur, B., Kelley, P. G., Komanduri, S., Lee, J., Maass, M., Mazurek, M., Passaro, T., Shay, R., Vidas, T., Bauer, L., Christin, N., & Cranor, L. F. (2012). How does your password measure up? the effect of strength meters on password creation. *USENIX Security Symposium*.

Ur, B., Kelley, P. G., Komanduri, S., Shay, R., Leon, P. G., & Cranor, L. F. (2015). Measuring real-world accuracies and biases in modeling password guessability. *USENIX Security Symposium*, 463–481.

Ur, B., Noma, F., Bees, J., Segreti, S. M., Shay, R., Bauer, L., Christin, N., & Cranor, L. F. (2015). "■i added '!' at the end to make it secure"■: Observing password creation in the lab.veras2012visualizingweir2009password

Word count metrics

NUC Bachelor Project Word Count:
Total Sum count: 16644 Words in text: 16442 Words in headers: 139 Words outside text (captions, etc.): 62 Number of headers: 53 Number of floats/tables/figures: 9 Number of math inlines: 0 Number of math displayed: 1
NOTE: References are excluded.