Password guessing: learn the nature of passwords by studying the human behavior

Ca' Foscari University of Venice
Department of Environmental Sciences, Informatics and Statistics

Computer Science Master's Thesis
Academic Year 2020-2021

Graduand Alessia Michela Di Campi 861844
Supervisor Prof. Luccio Flaminia
Supervisor Prof. Focardi Riccardo

Abstract

Passwords are essential in our life. They are the access key for all systems that require a security mechanism to avoid the disclosure of sensitive data. However, a very common mistake is to create weak passwords based on frequently used patterns. This attitude is helpful to the hackers. Through an in-depth analysis and with the help of pre-existing data leaks, the study conducted investigates the relationship between human beings and the choice of passwords: how the external environment, grammar rules and linguistic habits are deleterious for users. Unlike other studies, which prefer to concentrate their work on only password analysis and their distribution, in this study the protagonist is the human, particular importance has been given to the psychological nuance which allows us to understand how human habits and simplicity are a weak point. Most common patterns, the most frequent replacements are analyzed, the relationship between passwords and some dictionaries of words such as: the most known words, names, colors and many others. Also, other pre-existing scientific and psychological studies that show the relationship between password lengths and other factors such as the choice of numbers and symbols were investigated. An in-depth analysis of the different types of cyberattacks which are not only directed to technology also but at humans and at the manipulation of their way of thinking and consequently acting has also been studied. Finally, through a socio-psychological study that involved 217 people, previous discoveries were confirmed and enriched through profiling. Nowadays, knowledge of the psychological relationship between humans and how the technology can be used to protect a system is fundamental as the psychological / cognitive element becomes the decisive factor for the effectiveness of new attacks.

Contents

Introduction

Passwords are very important to us since we use them every day to access all the computer systems at our disposal. In order to protect them, it is necessary to analyze and understand how we can make them as less guessable as possible but also easy to remember. We often hear about data leaks. Data leaks are IT incidents in which sensitive information is mistakenly exposed due to vulnerabilities, human error or incorrect business processes regarding data protection and retention. Some data leaks also include passwords which, if too weak, become vulnerable to attack.

There have been large data leaks in the past that have made millions of passwords available. So we ask ourselves: What are the reasons why users, create apparently strong but extremely weak passwords?

When we have to register to a website we are asked to enter a password. Specifically, the password entered must follow some imposed rules called policies. The policies are useful to ensure that, if malicious users are able to access the database where the passwords are saved, they will not be able to trace the password *in clear* that is the version that the user entered at the 'signing up. Once registered, the password with the data encrypted and is saved in a database. Encryption protects passwords from brute force attacks, i.e attacks that consist of trying all possible combinations of passwords until at least one of them for is discovered.

We have very specific policies that help make passwords strong against brute force attacks. It is not uncommon that every time we enter a password, if it is deemed "too weak", the system asks us to enter another one with the characteristics indicated. In most cases, systems expect a password with at least one uppercase character, at least one number, at least one symbol, and at least 8 characters long. This request is made because there is a mathematical reason behind it. In fact,

to measure how strong a password is, there is a quantity: entropy. The higher the entropy, the stronger the password. What strengthens entropy is the use of the 4 character sets we have available. The sets consist of upper and lower case letters, symbols and numbers. Having a high entropy does not preclude the possibility of being subjected to other types of attacks such as dictionary attacks, mask attacks and rule based attacks that subject the password datasets to comparisons with words belonging to well-known dictionaries and pattern rules. In fact, another problem that has to do with passwords is the use of known words included in frequently used word dictionaries. What leads to using words belonging to dictionaries is the fear of not remembering the chosen password anymore. Another consequence of the fear of forgetting is the reuse of the same password for multiple accounts. So the main goal is of this thesis is. to study how to build systems that allow you to keep passwords as safe as possible without making them difficult to remember.

Problem statement In this thesis we study problems related the password creation. (1) The first one is to create passwords that are strong enough not to be easily guessed. (2) The second one is to create easy-to-remember ones for all users.
The general reason is to ensure that the passwords are strong and therefore not vulnerable to attacks and to make this happen, rules have been imposed. However, in front of imposed rules, a defense mechanism is activated by the user by the fear of forgetting and further factors come into play and activate a process, called cognitive dissonance, in which notions and opinions expressed at the same time by the user are in contrast to each other [77]. In this case, the discomfort caused by the conflict of one's actions with the rules, leads to the subject's use of so-called neutralization techniques, expedients of various kinds, tending to exclude or weaken individual moral responsibility. By doing this, the user creates passwords with specific patterns.

Contributions This thesis analyzes pre-existing password data leaks and through different statistics made on them, it extracts feautures. In particular it analyses now. Whether patterns change depending on the website / IT system or whether

it is a direct consequence of the policy adopted by the system. Finally it extracts the cognitive aspects that arise from some choices and it proposes techniques to prevent common mistakes.
In the literature, several studies have been conducted on the phenomenon of data leaks but most of them have concentrated on the analysis of the password as a single word or on the group of available passwords. In fact, many studies use machine learning techniques by taking a very large data leak as a training set and using other data leaks as test sets to understand how much their system learned from the training set to guess the passwords of the test sets. But in doing it is not possible to understand what the machine learning algorithm has learned since it is, by definition, a black box.
In this thesis it was analyzed in-depth the presence of some categories (names, dates, slang, objects, etc.). The first k-letters were compared with dictionaries. Then, it was investigated the relationship between pattern, length, and words category that make up the chosen passwords, to see if different patterns have specific characteristics. The phenomenon of substitutions of letters with numbers and symbols has been studied and substitutions have been extended with respect to what is found in literature. Then the law of the first digit was analyzed [7]. Finally, the relationship between human culture and the choices of password was analyzed. All of this, it has been done using preexisting data leaks. It was also conducted an interview by a questionnaire which has involved 217 people. In the questionnaire, questions of a general nature were asked, on the relationship that the participants have with passwords and choice strategies. In addition, some external stimuli have been posed by applying some studies in the literature to understand how the contour affects us. The goal of the questionnaire was to study the relationship between human and chosen password. Specifically if the level of education, age and work are factors characterizing the chosen passwords.

Thesis outline The thesis is divided into five chapters. The first chapter classifies cyberattacks by analyzing cognitive hacking in detail. It discusses hacker profiling that deals with the analysis and creation of personal, socio-demographic profiles, character, and psychological characteristics of the organizers of a cyber attack.

The second chapter deals with the relationship between humans and information systems, the impact that humans have on security and user profiling.
The third chapter, discusses password creations. It analyzes how user authentication works, the longevity of a password, and how the phenomenon of reusing the same is decisive, or not, of attacks. It analyzes password attacks and shows how to calculate the strength of a password. Finally, it discusses the psychological aspect behind choosing a password and how certain cognitive processes can harm security.
Following the analysis in the first chapters, the thesis experiments with real data leaks. Calculates several meausures, applied some laws, and analyzed frequencies and patterns. As a result, it discoveres frequent and incorrect behaviors in creating passwords.
After the analysis of the data leaks, chapter five, presents create a questionnaire thar analyses the relationship between participants in the questionnaire have with their passwords, their typical attitudes, and their chosen strategies. Through the questionnaire, our goal was to get more information about users, such as the level of education, age, etc., to understand what were, and if there are, aspects of the person that characterize their passwords. In addition to this, the extracted data gave us the opportunity to analyze the patterns and characteristics of the passwords entered by the participants and then to compare them with the discoveries made in the previous chapter.
We finally conclude proposing some future works.

Chapter 1

Cyber attacks

This chapter is an introduction to cyber attacks: Section 2.1 illustrates a standard a classification of cyber-attacks while, Section 2.2 shows a less-standard classification well-known attacks under the definition of *cognitive hacking*. Finally, the last section presents hacking profiling by considering the goals of attackers.

1.1 Classification of cyber attacks

Most of the current work classifies cyber attacks by attack types and on the type of damages caused independently of the act. These classifications are interesting for this discussion as they permit us to spotlight how they are strongly characterized by the absence of human work.
Before defining the attacks we need to recall to main security goals: confidentiality, integrity and availability.

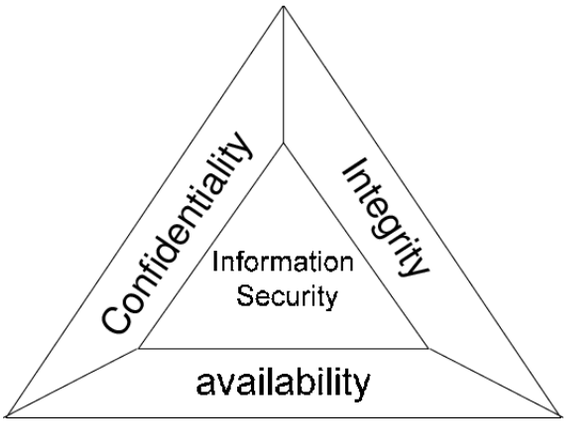

Figure 1.1: CIA triad [88]

Definition 1. *[9, 119] Confidentiality measures the protection of an information from unauthorized access and misuse.*

Confidential information often has value and systems are therefore under frequent attack as criminals hunt for vulnerabilities to exploit. Most information systems house information that has some degree of sensitivity. It might be proprietary business information that competitors could use to their advantage, or personal information regarding an organization's employees, customers or clients.

Definition 2. *[9, 119] Integrity measures the level of protection of an information from unauthorized alteration.*

This measure provides the accuracy and completeness of data over its entire lifecycle. The need to protect information includes both data that is stored on systems, and data that is transmitted between systems such as email. In maintaining integrity, it is not only necessary to control access at the system level, but also to further ensure that system users are the only one able to alter information that they are legitimately authorized to alter.

Definition 3. *[9, 119] Availability measure protects timely and uninterrupted access to the system.*

Some of the most fundamental threats to availability are non-malicious in nature and include hardware failures, unscheduled software downtime and network bandwidth issues. Malicious attacks include various forms of sabotage intended to cause harm to an organization by denying users access to the information system.
Keeping in mind these important security properties we can now define the differences between two types of attacks: Passive Attacks and Active Attacks.

Definition 4. ***Passive Attacks*** *are a danger to Confidentiality and they are the type of attacks in which, the attacker observes the content of messages or copies the content of messages.*

This type of attack, does not harm to system, it is a passive attack, the victim does not get informed about the attack.

Definition 5. ***Active attacks*** *are a danger to Integrity and availability and attacks, the attacker tries to modify the content of the message.*

Thanks to active attack, the system is usually damaged and system resources are often changed. Moreover, in active attack, the victim gets informed about the attacks.
The Computer Emergency Response Team (CERT), which is an expert group that handles computer security incidents, classifies incidents (cyberattacks) into macro-categories, such as probe and scan, packet sniffer, account and root compromise, DoS, exploitation attacks, malicious code, and Internet attacks on infrastructures. We will now discuss them in detail.

Definition 6. *[61]* ***Packet sniffing*** *is the practice of gathering, collecting, and logging some or all packets that pass through a computer network, regardless of how the packet is addressed.*

These types of attacks are performed using automated tools (software) and are defined *scan* (see Figure 1.2). These type of attacks can be motivated by simple curiosity and are often the result of a wrong configuration or of an error. Although they do not change the status of a victim in the strict sense, they are often a prelude to much more serious attacks and their purpose is often to gather information on

the vulnerabilities of the victim. Unlike following attacks, these do not require internal access to the information but they analyse the information in transit in the network.

Not all data and information monitoring attacks are passive. Some attacks that fall into this type can modify the data in transit to produce other types of attacks. In many computer systems no special privileges are required for this type of access and this makes them particularly insidious. An attack of this kind is represented by the so-called *Man-in-the-middle* (MIMT) attack. Unlike packet sniffers, in this case the monitoring of data in transit is not performed by a program but by a user who places himself between the victim and the other nodes of the network. Man-in-the-middle attacks can too be preliminary actions to subsequent attacks of another type, such as attacks on exploitation of trust. MITM attack, may be a cyberattack where the attacker secretly relays and possibly alters the communications between two parties who believe that they are directly communicating with one another. One example of a Man-in-the-middle attack is active eavesdropping, during which the attacker makes independent connections with the victims and relays messages between them making them believe they are talking on to one another over a personal connection when actually the whole conversation is controlled by the attacker. The attacker must be ready to intercept all relevant messages passing between the two victims and inject new ones. This is often straightforward in many circumstances; for instance, an attacker within the reception range of an unencrypted Wi-Fi access point could insert himself in the middle and intercept messages.

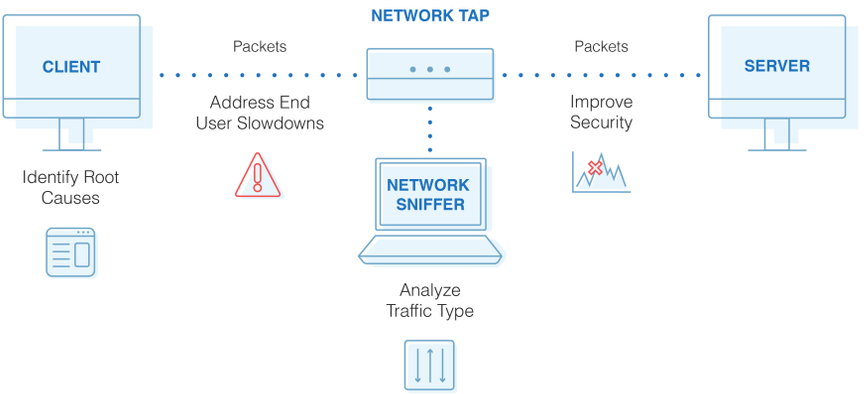

Figure 1.2: Packet sniffering [34]

Definition 7. *Compromise of an account, root, and exploitation of trust is an attack where an individual takes advantage of a trust relationship within a network.*

Attacks to accounts and root users, and trust exploitation are active attacks and their primary objective is to violate not only the confidentiality of a system but also the integrity. The compromise of an account defines all those unauthorized uses made by a person other than the owner without any type of specific privilege to access the system. The root compromise is a similar incident to the previous one but with privileges. Root-related attacks are attacks that aim to capture a code or key necessary to interpret or decrypt secure information, called compromised-key attacks.

To be performed within a network, various services or processes require identification and, authorization to perform the service. Trust exploitation attacks, recently also referred to, as identity spoofing identifies that series of attacks that aim to disguise the request for access to a process or service, in such a way as to make those who request it enabled to do so even if they are not.

Through active attacks, such as the exploitation of trust, or simple attacks passive heels, such as packet sniffers, a series of password-based attacks can also be conducted, attacks can herald various types of attacks by reusing the captured

passwords, as in the case of attacks service compromise or Internet attacks on infrastructures.

Definition 8. *Attacks based on* ***malicious code****,* ***DoS****, as well as* ***Internet attacks*** *on infrastructures, are active attacks, involve a violation of the confidentiality and integrity of availability of data, processes, and services of an IT system.*

A large set of security attacks are *malicious code* or *malware*, terms that generally refer to a series of software codes or programs which, if performed, can cause undesirable effects. Examples of malicious code are viruses, worms, Trojans, as well as Logic Bombs. Users generally are not aware of these attacks but only of the final damage. Viruses and Trojans are often hidden within harmless programs or files, altered in such a way that they do more than what one would expect from their execution. Unlike viruses, which require some action on the part of the user to propagate, worms are self-contained programs replicants that, once executed, propagate without the intervention of those who created them. It should be emphasized that the intervention of the user involved in these attacks is such as not to require any awareness: the individual triggers the attack without knowing it.

Unlike previous attacks, *Denial of service* (DoS) are explicitly and openly aimed at changing the status of a victim (see Figure 1.3). DoS do not necessarily seek specific access to a victim or its information, but aim to make the service inaccessible to legitimate users to blocking the service or attacking a computer system being attacked. Generally this happens by pouring on the victim a disproportionate amount of data that cannot be managed by the victim itself. Other DoS forms may include even the breaking of physical components or the manipulation of data in transit.

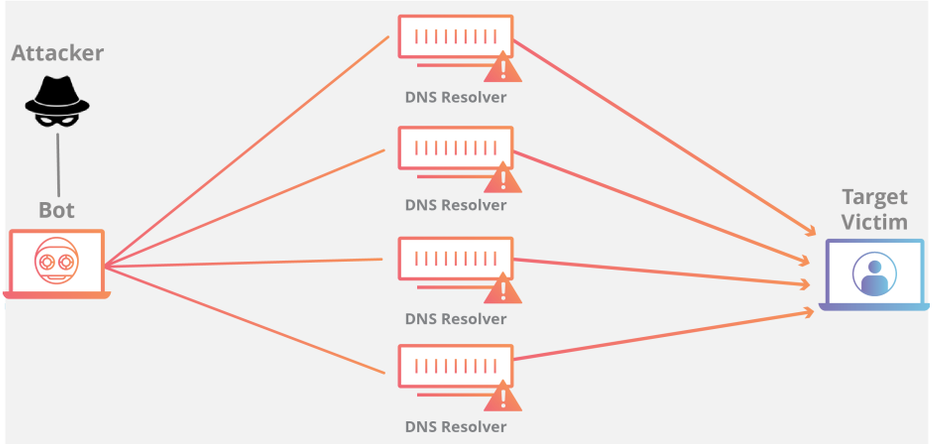

Figure 1.3: Denial of service [16]

The latest cyber attacks considered are *Internet attacks on infrastructures*. This type of attacks brings together a wide range of security incidents involving key components of the Internet: from infrastructures to specific Internet systems, such as servers, access networks to providers or large archives on which services and users depend. Large-scale incidents of this type can also affect key infrastructures and seriously compromise a large portion of the Internet, blocking all kinds of operations on the affected websites. Similar attacks are the *Application-Layer attacks*, which aim to create a flaw in the servers used to access or control a system, so as to be able to bypass the control and take possession of the information or identities of the nodes controlled by it [110].

From a psychological point of view, what remains clearly out of the classification described up to now is evidently human. Most of the deliberate causes of inweakening of IT security described almost exclusively involves the technological environment, as well as the attacks examined are always directed to this.
This classification, although it faithfully reflects the attacks aimed at the information environment, but considering as cyber attacks only those aimed at technology, focussing the protection of cyber security risks focusing only on technological security and forgetting that attackers and victims are human.

1.2 Cognitive hacking

Despite the vastness of the classifications relating to technological cyberattacks, few works have classified the attacks considering the specific human intervention within an attack or, more generally, the vulnerabilities of human-technology interaction.
The Center for Internet Security (CIS) literature, is a recent no profit organization and its mission is to "identify, develop, validate, promote, and sustain best practice solutions for cyber defense and build and lead communities to enable an environment of trust in cyberspace" [10], has highlighted several deliberate causes of cybersecurity weakening involving humans. These works have underlined how the classifications of cyberattacks tend to greatly underestimate non-technological and non-syntactic attacks. Libicki in [92] was one of the first authors to bring out the relevance of human intermediation in the deliberate causes of CIS weakening, introducing the distinction between physical cyber attacks, the aforementioned syntactic attacks, and semantic attacks.

Definition 9. *[72, 115]* ***Physical attacks*** *are set of attacks directed at the physical and electronic components of the technological environment.*

Therefore the deliberate equivalent of the natural or human causes of an accidental nature previously described.
The redundancy of the physical components of a network or a computer system, the constant reduction of the physical size of their components, as well as the ease of their replacement, have made physical attacks less dangerous and it is now easier to remedy the consequences created by these attacks. In addition to the syntactic attacks already described, those defined as semantic attacks, considered today among the most insidious causes of CIS weakening, take on particular relevance from a psychological point of view.

Definition 10. ***Semantic attacks*** *include all those attacks that exploit the way humans attribute meaning and interpret the information they handle.*

According to [115], who takes up the [92] distinction, these attacks can generally be described as characterized by the use of false information. Unlike syntactic

attacks, the purpose of the latter is almost always deceptive and therefore requires an interpretation by the user to whom they are directed. A broader distinction was adopted by [60] who distinguishes general hacking, attacks based on psychology, attacks based on the psychological manipulation of operators and users of a computer system. More generally, we can distinguish from cyber attacks related to general technological hacking the set of cyberattacks that cause damage or affect the security of a computer system and require the interpretation and direct or indirect involvement of man and his behavior. Using a distinction we can define this type of attacks with the term *cognitive attacks* (cognitive hacking) [72]. Any cyber attack that, in order to be successful, requires a change in the behavior of an individual is part of cognitive hacking.
Five forms of cyber attacks of a semantic nature attributable to cognitive hacking will be described below, in particular pretexting, phishing, online deception, misinformation attacks, and social engineering techniques.

Definition 11. *[60]* ***Pretexting*** *is an attack in which the attacker creates a scenario to try and convince the victim to give up valuable information, such as a password.*

Pretexting and Phishing distinguishes two other large areas of cognitive cyberattacks on the basis of two types of possible victims: pretexting (pretext: pretense, falsehood), generally used against the operators of a computer system or service, and phishing, commonly used against the end-users of a service.
Pretexting is a social engineering strategy where a vicious attacker retrieves sensitive information from unsuspecting users. This social engineering attack's main characteristic is that the scammer crafts a somewhat believable story (or pretext) to manipulate the victim.
Within this scope, there also are various sorts of 'personification' or the set of cognitive attacks during which the attacker uses a false identity for deceptive purposes. Deceptive scenarios and sorts of personification are often used both in face-to-face interaction and in an interaction mediated by ICT technologies.

Definition 12. *[60]* ***Phishing*** *is a type of scam carried out on the Internet through which an attacker tries to deceive the victim by convincing them to provide informations.*

Phishing is a type of social engineering attack often used to steal user data, including login credentials and credit card numbers. It occurs when an attacker, masquerading as a trusted entity, dupes a victim into opening an email, instant message, or text message. The recipient is then tricked into clicking a malicious link, which can lead to the installation of malware, the freezing of the system as part of a ransomware attack, or the revealing of sensitive information.

Definition 13. ***Online deception*** *and* ***page-jacking*** *(also defined with the term page hijacking) are a deceptive practice that uses a false reproduction of a website by copying its name and appearance in order to steal sensitive information, hijack financial transactions or simply discredit the original site.*

An attack of a cognitive nature in clear expansion is that of online deceptions, in particular through the use of page-jacking. These attacks are configured as a form of deception that exploits new technologies, and in particular the Internet.

Definition 14. *[30]* ***Misinformation*** *is false, inaccurate, or misleading information that is communicated regardless of an intention to deceive.*

Attacks that exploit false information, called misinformation attacks, are attributable to cognitive hacking. [72] define false information as the set of misleading, false, or biased information that is deliberately disseminated in order to change the beliefs and behaviors of persons who become aware of it. These attacks can use different vehicles for their dissemination, such as web pages, emails, blogs, and so on, and they can be directed both to users of a service and to security operators - in the event that the false information relates to IT security products. The attackers who exploited false information may also fall into the category of opportunists described above: in this case, the false information may come from the inside of an organization and these attackers can therefore exploit specific privileges given by their role.

Definition 15. *[50]* ***Social engineering*** *is a manipulation technique that exploits human error to gain private information, access, or valuables.*

Within cognitive hacking, it is also possible to include all those forms of non-technological hacking defined as social engineering [85]. This area describes a

series of techniques based on cognitive processes of influence, deception, and psychological manipulation aimed at obtaining confidential information or sensitive data [82]. From a psychological point of view, the main criticality of this area is based on two types of problems: on the one hand, these techniques are extremely heterogeneous, on the other, they include very different attacks within the same category.

The most common psychological techniques used in social engineering involve the exploitation of elements such as authority, guilt, panic, ignorance (which is important from the point of the password choices since many people rely on routine and make typical mistakes. See next chapter), desire, greed, and compassion. All these, if recognized, can, in turn, help the victim to avoid being attacked, and it is therefore important in the world of cybersecurity to take them into consideration to increase the awareness of users of the web.

From a psychological point of view, misinformation attacks, pretexting and phishing, as well as online deception and social engineering techniques, are all attacks that use semantic manipulations and therefore attack attributable to cognitive hacking. Cognitive-based attacks create or somehow exploit deceptive scenarios from false information that victims believe to be true or that is communicated as such. Analyzing these scenarios, what emerges is the constant presence of deception: from the point of view of the psychology of communication, deception presupposes nor the ability to mentally picture the behavior of another individual based on a representation of his or her mental states, such as intentions, desires, and beliefs. A deceptive act is any act or functional trait of an organism that has the purpose of not giving another organism true knowledge that is relevant for that organism - and which does not reveal this purpose. The psychological manipulation of deception, therefore, takes place at the level of mental states that are given as shared in that specific context. In this regard, [105] defines deception as an actor's attempt to manipulate the mental states of his partner.

The actor aims to induce false beliefs within the partner about the external environment by pushing him to require actions favorable to his purposes. Consistent with Cognitive Pragmatics [113], a discipline that studies the mental states of in-

dividuals engaged during a communicative activity, deception may be a conscious violation of a shared behavioral game. *Behavioral games* represent the knowledge structure through which interpersonal actions are coordinated (in our case the activities and transactions typical of online environments, from the sale of products to the traditional exchange of emails): people use this data structure to pick the particular meaning of a sentence during a given context. For 2 actors to cooperate at a behavioral level they must operate the idea of a partially shared action plan. Cognitive-based attacks have a standard matrix which will be described in terms of exploitation of behavioral games given for sharing. These attacks violate this game by manipulating the mental states of the victim into believing false knowledge or by silencing true knowledge relevant to the victim. Such manipulations induce false beliefs about the external environment to vary the victim's behavior by pushing him to require actions favorable to the attacker's purposes.
As previously mentioned, in addition to beliefs, cognitive hacking can also act on perceptual aspects. Some attacks of this type act through the visual manipulation of a site, a logo, or a series of images, inducing the user to accept or select some target stimuli, such as links or disguised images, even if these do not have the same functions. In this regard, think of *page-jacking*, but also to web pages or emails that contain images or symbols (such as the X that refers to the closing of a window or a screen) that turn out to be access links to unwanted web pages.

Definition 16. *[35]* ***Pagejacking*** *is the process of illegally copying legitimate website content (usually, in the form of source code) to another website designed to replicate the original website.*

To accomplish pagejacking, a fraudulent pagejacker copies a favorite Web page from a reputable site, including its actual HTML code.

1.3 Hacker profiling

A recently expanding field of investigation, linked to both cybercrime and the psychological dimension of security, is represented by hacker profiling. *Hacker profiling* deals with the analysis and creation of personal, socio-demographic profiles, character, and psychological characteristics of the organizers of a cyber attack.

Some recent works have highlighted how a more accurate understanding of the underlying reasons for a cyber attack (cognitive or technological) and the use of psychological profiles of the songs allow to respond more effectively to intrusions and allow a certain degree of predictability about future attacks [64], [90]). The use and application of psychological knowledge in this area will be central to the development of increasingly targeted and effective responses to cyber attacks: in particular, the study of the link between psychological motivations and attacks can help to understand how different attackers choose.

Among the various works that have recently dealt with elaborating schemes and classifications relating to the different types of known attackers, the ones that are more properly of interest here are the researches that have highlighted the psychological value of some of the variables involved in this sector, among these variables, in particular, the motivation that drives the behavior of the attackers.

In psychology, reference is generally made to the term motivation (human motivation) to indicate the set of factors having a given origin that induce an individual to behave towards a given goal ([109, 123]). In humans, motivations perform two fundamental functions: they activate and guide specific perceptions, cognitions, emotions, and behaviors. On the one hand, motivations activate specific behaviors by providing the energy necessary to trigger and maintain a specific behavior (Drive theories). On the other hand, the motivations can also represent the directional components of the orientation of behavior towards a specific goal (Incentive theories). In the latter case, the reasons may represent the parameters responsible for the change in human receptivity to specific environmental stimuli. According to [109], understanding motivation can restore both the meaning of behavior and reveal the values of the person who carried it out. These considerations highlight how the motivational dimension can be particularly useful if we want to analyze attack behaviors. Marcus Rogers was among the first authors to apply psychological analysis to Digital Forensics focusing in particular on the motivational aspect [111]. The author examined that different motivations that can move of a hacker to build predictive taxonomic categories applicable to the investigation [111] identifies four motivational macro-categories - such as curiosity, notoriety, revenge, and financial motivation.

The reasons that lead a hacker to attack a specific target can be used to discrim-

inate against different categories of hackers. [90] and the SANS Institute define the psychological dimension and the study of behavior human key areas in the categorization of hackers, underlining how these areas are rarely used in IT security training. A psychological dimension is used by [90]) to distinguish attackers into five categories based on their motivations, such as squatters, insiders, random attackers, organized crime, political hackers.

Definition 17. *Attackers who fall into the defined group of* ***squatters*** *are characterized by the impersonality of their attacks. Their goals are often independent of the intended- or the identity of the owner of the attacked system.*

Squatter attacks are rarely communicative. Often the intent is to gain access to large databases containing confidential information, passwords, or even video, music and images. The reasons are sometimes playful, with purposes often of a private nature and not necessarily criminal; their conduct can therefore be moved by intentions similar to those of random attackers, i.e. driven by curiosity or the need for recognition. The mere interest in researching security systems vulnerabilities can in itself justify their attack. In this sense, their attacks are in fact often aimed at security operators and generally fall within attacks that are not strictly criminal. Malicious code programmers (such as worms) belong to this type and their goal is to spread their code as much as possible in an impersonal way: this is often done to conquer individual computers - making them passive to legitimate users and active against attackers - in order to use them to their advantage for DoS attacks against third parties.
There are several techniques of squatting e.g typosquatting, combosquatting, homographsquatting, bitsquatting etc.

- *Typosquatting* [125], also called "URL hijacking", may be a sort of cybersquatting supported typing errors made by someone who types an internet address. It consists in purchasing domain names like the foremost common typing errors.
  By typing incorrectly, the user can reach the typosquatter site.
  The goal of typosquatters is:

  - intercept part of the traffic directed to the official website;
  - intercept as many e-mails as possible sent to the addresses of the targeted person or company.

- *Combosquatting* [86], also known as *cousin domains* [93], refers to the combination of a recognizable brand name with other keywords (e.g., paypal-members[.]com and facebookfriends[.]com) .

- *Homographsquatting* [23] consists in attackers that register a website that is visually similar or just like a registered target domain through the International name protocol, which allows for the display of Chinese, Arabic, Korean, Amaric, etc. characters in domain names. Some characters, just like the Russian "," appear just like certain English letters, meaning "apple.com" (English "a") and "apple.com" (Russian "a") can resolve to thoroughly different servers, with end users none the wiser

- *Bitsquatting* [124] relies on bit-flip errors that occur during the method of creating a DNS request. These bit-flips may occur thanks to factors like faulty hardware or cosmic rays. When such a mistake occurs, the user requesting the domain could also be directed to an internet site registered under a website name almost like a legitimate domain, except with one bit flipped in their respective binary representations.

Another particularly interesting type of attackers is that of **insiders** and **intruders**. Attacks attributable to this type they can be carried out both from within, or by internal operators or users to an organization or a computer system (insiders), both from the outside, or from external attackers such as spies or competitors who illegally enter an organization (intruders). Among the internal operators of an organization, distinguish security operators - with specific access privileges to the IT security system - and simple internal users or employees of an organization, staff with privileged access to the IT system, but not to the security systems. The tendency to structure cybersecurity exclusively on the basis of external attacks has led to widely underestimating attacks originating from within.
From a point of view psychological, of crucial importance among the skills required

to operate within an organization is the ability to build a relationship of trust between the organization and the people who work there and, above all, to follow and monitor the progress of this relationship.

Definition 18. ***Random Attackers*** *are attackers which seek access to or disruption of any target that appears vulnerable.*

Random Attackers are frequently motivated by curiosity, and organization of an attack gives them an emotional rather than an intellectual motivation. They are often gratified, for example, by the simple possibility of using others' subscriptions on paid sites. Random attackers represent the largest group of hackers and seem to distinguish themselves as not particularly experienced, using attack tools or methods developed by more experienced attackers. For this reason, their work can be even more risky as it is clumsy, poorly managed, the result of trial and error and often acted on an emotional basis. These attackers are not always aware of the damage caused and often leave many traces of their operations (even in situations where their attack aims to co-operate). Another important motivation of random attackers, in addition to curiosity, is being accepted by other groups or communities of more experienced hackers, probably in an attempt to acquire a stronger and more specific sense of belonging and identity. Especially in defined open attacks, such as defacing or DoS, the motivation to be recognized as attackers defines the choice of the type of attack as a sort of direct communication, rather than to users or security operators, to others attackers or their communities. In this sense, the attacks chosen often fall within the category of attacks that are not strictly criminal.

Definition 19. ***Organized crime*** *is made up of attackers who are generally professional and therefore very experienced in this sector. The reasons are essentially of an economic nature e the attacks chosen have as their ultimate goal the profit.*

Attacks and targets are carefully designed and chosen and hardly leave traces of their work. The chosen attacks are almost never opened attacks and can cover a wide spectrum of types such as access compromises, trust exploitation attacks and malicious code. Among the techniques adopted, those related to the initial phases of monitoring and information search are of fundamental importance, such

as probes / scans and packet sniffers. Attacks of this type are almost always criminal attacks properly understood.

Definition 20. ***Political attackers*** *present themselves as militant hackers for a cause. Their attacks, as well as their knowledge and experience, are almost always the result of adhering to an ideal. This category is therefore also guided by a rational as well as an emotional dimension.*

The form of attack chosen by political attackers and the type of damage caused are always consistent with the married cause, and their work is often configured as a 'communication' aimed at making their ideal public. For these reasons, political attackers are often referred to as cyber-activists) and their activity can sometimes be very dangerous, especially if it comes back in religious terrorism, as in the case of cyber-terrorists (terrorist attackers). The motivations that guide the work of these hackers can therefore be very similar to the reasons actions underlying a non-cyber terrorist attack. The objectives chosen of their attacks, such as large institutions, multinationals or even the already mentioned born critical infrastructures, they are frequently identified as opposing or, in some way, enemies of their ideals. Goals can also be simple network or single computers but almost always as part of a much larger whole.

Conclusions In this Chapter we have introduced the concept of classification of Cyber Attacks by describing their peculiarities. We then focused on cognitive hacking and then analyzed the attackers' profiling. The importance of knowing these aspects lies in better understanding who is behind a Cyber Attack.

Chapter 2

Psychology of Information Security

In this chapter, we put the other side of the coin in the spotlight: the victim of the cyber-attacks which have been described in the previous chapter. In Section 3.1 we depict how humans and technology influence each other, and we introduce the concept of "extended mind". In the next section, we show the impact that humans have in the field of cybersecurity and, proceeding to Section 3.3, how typical behaviors in everyday life allow us to profile the victims. Finally, in Section 3.4, we introduce the main topic of this thesis which is the password, especially the good rules to make a strong password in relation to the psychology view of a human.

2.1 Human and technological environment

"Where does the mind stop and the rest of the world begin?" The article Extended Mind begins with this question [73].
The Extended Mind Theory is constituted as a refusal of identification ontological and epistemological of mental and cerebral, which is equivalent to supporting the idea that the mind is not (only) the brain, and therefore cannot be explained exclusively by describing the brain mechanisms responsible for cognition.
Cognitive Sciences have interpreted technology as the product of a modification

intentional and conscious of man towards his environment. The relationship between human and technological environment is then described in terms of support and expansion of the human cognition by technology.
Research in this area has long since highlighted how human cognition, defined as a set of cognitive abilities that allow humans to process the stimuli that the interaction with their environment offers, can be considered as a system with limited resources. Several authors have tried to quantify the limits of human cognitive abilities in terms of memory, planning, reasoning, attention showing how such skills have availability limited and limited resources.
Awareness of this limit has led human to modify the surrounding environment by creating artificial devices, or artifacts, capable of extending his cognitive abilities. Cognitive artifacts could also be defined as "those artificial devices that maintain, display, or operate upon information so as to serve a representational function which affect human cognitive performance" [100]. Cognitive artifacts are in other words human-made things that appear to assist or enhance our cognitive abilities, and a few examples are calendars, to-do lists, computers, or just tying a string around your finger as a reminder. Representing information using artificial supports means, on the one hand, detaching it from an exclusively cognitive representation, and on the other, externally allocating part of the processing and thus reducing, to a certain extent, the load on the cognitive resources available.
In his conception of cognitive artifacts, Norman dissociates himself from the notion that artifacts "amplify" a person's cognitive capabilities. Instead, he argues, they modify the character of the task performed by the person. He suggests that the notion of cognitive amplification has arisen because artifacts appear to play different roles depending upon the purpose from which they're viewed. He recapitulates two views: The system view and therefore the personal view. Examples of artificial artifacts with an external representative function of information can range from taking notes on a sheet of paper etc. They are artificial artifacts as they would not exist in nature if they had not been conceived and made by man. According to Donald Norman, man uses a wide range of artifacts that extend his cognition both of a mental nature, such as reading, arithmetic, logic, language, and of a physical nature, such as paper, pencils, calculators, computers. More generally, any technological entity invented by man in order to enhance his own

thinking and actions can be defined as a cognitive artifact and be considered part of an extended cognition.

Cognitive Ergonomics, among the disciplines that have dealt with investigating the relationship between man and the environment technological, certainly stood out for greater attention given to human cognitive processes and to technology as extended cognition.

According to the definition provided by the International Ergonomics Association, ergonomics is the scientific discipline that deals with the interaction between the individual and the other elements of a system and the ergonomist is the professional who applies theories, principles, data and methods of design in order to optimize the well-being of the individual and the performance of the entire system. The expression "cognitive ergonomics" emphasizes the cognitive aspects of this interaction, that is, the way in which the user of a technology perceives, pays attention, decides and plans his actions in order to achieve a goal. Of course, the design of technological devices cannot ignore the analysis of these processes. The knowledge developed in the field of cognitive ergonomics allows, in fact, to develop usable systems, capable of reducing the workload imposed on the user and the probability of making mistakes [99]. This approach has tried to shift attention from the technological changes that man has imprinted on the external environment to the interaction between man and the environment he himself modified, on the one hand, and to how the technological environment has influenced and modified it.

The attention paid to these influences allows us to consider any weakening or fracture of this interaction as attributable not only to human limits or technological limits, but above all as determined by conditions of incompatibility between man and technology. From this level of analysis the interaction between man and technology should be treated as a somewhat cooperative process, a process in which misunderstandings given by the lack of knowledge of the tool, the difficulty in using it, human laziness and many other factors can arise from both sides.

Technology generates a potential paradox: it makes life easier and more pleasant, offering numerous advantages, and at the same time it evolves in complexity, gradually becoming less user-friendly. Research in this field has highlighted how the development of technological artifacts tends to follow a *U-shaped* path in terms

of complexity: new technology tends to follow this curve of complexity: starting high; dropping to a low, comfortable-level; then climbing again. New devices are complex and need much effort to learn their usage, as the maturity level of industry and users increases these devices becomes much more simpler, reliable and easy. But when industry reaches a stable point of the maturity, new users figure out how to add increased power and capability, and this always happens at the cost of added complexity and sometimes decreased in reliability.
The problem arises when the production objectives push towards an increase in the operating and calculation capabilities of the device for the legitimate purpose of maximizing its potential. As the technological capabilities and the number of options and functions available increase, the number and complexity of the commands also increase in the devices, and with them the interaction difficulties and incompatibilities between the technological device and the end user also increase in cascade.

Among the devices that make up the current technological environment, the one that has always affected cognitive ergonomics has certainly been the computer. The computer represents the most relevant technological artifact of the current information revolution, and the study of *human-computer interaction* (HCI) it has involved a large part of the applied developments of psychology in this field.
Human-computer interaction is a multidisciplinary field of study that specialize in the planning of technology and, especially, the interaction between users and computers. While initially concerned with computers, HCI has since expanded to hide most sorts of information technology design.
HCI is dealing not only with the influence of humans on the IT environment, but also the IT environment on humans, analyzing the design and the degree of usability of information technology and evaluating the influence of the IT environment on the organization of human behavior and its experience.[71]. According to the HCI much of the organization of this interaction is in fact dictated by technology information technology, and the resulting IT environment often brings with it little attention to human-artifact interaction and a broader set of weaknesses in the design phase. Research in the HCI field has highlighted this technological imbalance since its origins, complaining about the tendency of engineering disciplines to

adopt methodologies and objectives typical of the natural sciences - thus neglecting the human and social sciences to the detriment of the design and usability of technological and industrial environments. This imbalance has had and has wide repercussions in the area of security.

Information technologies have made the human environment a complex and articulated technological environment and the resulting interaction is far from error-free and false limiting. If in the psychological field the study of the relationship between man and technology has a relatively long history of research, the issue of the security of this relationship has only recently interested the scientific community outside this field ([80, 63]).

There may be many dimensions to be secured and safety takes on different meanings depending on the areas to be used. From technological safety generally intended to the safety of information environments from the security of services and procedures to the security of data and information [120]. From a psychological point of view, what certainly cannot be excluded on the subject of safety is the analysis of the interdependence of these environments on humans.

Two aspects make this interdependence relevant. First, the dissemination of information and communication technologies (Information and Communication Technology, ICT), or all those technologies that organize the daily exchange of information between individuals and make possible the different types of online communication line and off-line on which we rely every day (telephony, e-mailing, instant messaging, chat, social networks, websites and so on). Secondly, the computerization and digitization of information and the development of electronic systems for digital information management. The latter aspect, in particular, has asked for greater attention to the security of the information itself (Information Security, IS), no longer traceable and placeable in a physical space. What characterizes both the exchange and the electronic storage of information today is the always greater dependence on technologies and IT environments and, consequently, on their securing.

Definitions of cybersecurity should therefore consider the interdependence between the technology and those who use it, and the dependence created between some transactions and the technologies through which they are carried out, are of extreme importance. A cybersecurity approach has several layers of protection spread

across computers, networks, programs, or data that you intend to keep safe. In a business, people, processes, and technology must complement each other to create an effective defense against cyberattacks.

2.2 Impact of the human factor on IT security

Thanks to the development of mobile telephony and the Internet, transactions global economic, as well as simple interactions between private citizens, have become heavily dependent on the development of ICT. The development of technology and the importance in everyday life has followed distinct phases: from the electronic data processing phase of the 1960s to the automation of services in the 1970s, from the integration and diversification of the 1980s to a phase, the current one, of real information contagion large-scale electronics. The security of the relationship between humans, his technological environment, and the information conveyed therefore currently has profound implications.
So, where does technological security end and where does individual security begin? One of the most major criticism currently addressed to the protection of the CIA triad (see section 2.1) is the classic mind characterized by a substantially technological. From this point of view, IT security is comparable to technological security and being safe means having safe technology. What emerges is a general attitude of this area that tends to overestimate technological vulnerabilities at the expense of a more lucid consideration of other vulnerability factors. Among these forms of vulnerability, one of the first factors to be put in light it was certainly what is called the *human factor*. By human factor we mean everything that depends on human work regardless of the reference technology. Interest in the human factor in the IT sector made it possible to shift attention in favor of the safety of the interaction between man and technological artifact. Dealing with cybersecurity in this perspective has meant, for example, taking into consideration vulnerability factors not strictly dependent on technology, such as the vast domain of human error. In this sense, the vulnerabilities linked to the human factor can be broken down and analyzed at various levels: think of design errors, errors in the use of technology, organizational and managerial errors, as well as errors caused by cognitive limitations.

Technology can be seen as composed of a series of cognitive artifacts of an artificial nature, the result of modifications made actively by man on his environment and expression of an enlarged cognition. From this perspective, securing the technological environment therefore means securing a broader knowledge. Raising awareness on these issues is prompted by three aspects: the organizational dimension of security, the type of security attacks and the paradoxical improvement in technological security. In current security systems, management depends not only on the behavior of individuals, and therefore on the individual dimension, but also on the organizational dimension. In addition to the organizational sphere, several works underline how attention to the psychological dimension of IT security is a determining factor not only for security in itself, but also for the design of response strategies to external attacks or internal errors within an organization.
Finally, the analysis of the psychological dimension allows us to offer a possible key of reading to the paradoxical discrepancy that is observed between the decrease in technological environments on the one hand, and an increase in the sophistication of technologies on the other [80].

At this point another question spontaneously arises: where does human error end and where does behavior begin? Among the human factors involved, the main aspect examined in terms of information security was certainly human error. The psychological and human dimension is in fact cited as a significant cause of security incidents in different domains, and not only in the IT sector: the human factor in terms of error is implicated in the medical field, in air accidents or in banking transactions ([107]), as it has been known for some time that the human factor is involved in 80-90% of accidents in the organizational sphere [108].
Human error is identified as the most frequent cause of data and information security problems affecting organizations, causing more than half of cybersecurity incidents in this area [108]. From the point of view of an organization, the repercussions due to human errors are manifold and can be expressed in terms of production inefficiencies, loss of money and customers, the origin of vulnerability to external attacks, lawsuits, and last but not least from an economic point of view, public embarrassment [62].
The authors of [112] conducted several researches using questionnaires. The result

was that human mistakes or errors represent an important cause of vulnerability in the IT environment. For example, a research conducted using questionnaires administered via the web on behavior related to password management [112] has shown that the human beings can heavily influence the protection of an organization's IT security. In particular, the authors highlight a significant relationship between the incorrect use of passwords by users - mainly due to problems in storing and registering them - and the general decrease in the organization's IT security.

At this point we ask ourselves if human error is the only psychological variable that explains the vulnerabilities of IT environments and how humans are involved in the protection of cyber security.
First of all, let's start with two distinctions of the causes of computer security weakening: deliberate causes and accidental causes (National Research Council, 2002).

Definition 21. ***Deliberate causes*** *are the result of a conscious and intentional human choice, the typical example is cyber attacks.*

Definition 22. ***Accidental causes*** *include both natural causes, such as a lightning strike that produces a power failure of a network, and unintended human causes, ranging from programming errors to unintentional cutting of a power cable.*

While cyber attacks are considered as deliberate violations of cyber security - the result of a conscious human choice and aimed at a specific damage - human errors are considered as accidental, not deliberate actions, not specifically aimed at a breach of security but still causes its weakening (National Research Council, 2002).
These dimensions are obviously correlated in that the deliberate damage to a computer system can exploit, openly or not, an error accidentally inserted into the system. This distinction has proved useful in various areas [89] and has allowed the identification of two large dimensions of risk based on the possible distinct reasons underlying the weakening of security.
In addition to this distinction, a further discriminating factor is the role played by man in relation to these causes. [69] distinguish four levels of human involvement

in the field of information security: security operators, attackers, users and double agents.

Definition 23. ***Security operators*** *are represented by all the people who deal with protection of a computer system, and include figures such as system administrators, technical personnel, anti-virus vendors, as well as support companies the security.*

Definition 24. ***Attackers*** *are those who illegally use computer systems or deliberately cause them to be destroyed, damaged or blocked, such as cybercriminals and hackers.*

Definition 25. ***Users*** *generally includes all people who legally use computer systems, from simple users of a service to internal staff to a computer system that operates with particular usage privileges.*

It is not unusual that these latter figures can also assume the role of attackers, especially if they operate within a computer system with specific privileges; in this case they are defined as intruders. Ultimately, opportunists can take on a double role: on the one hand they can work on IT security products, such as security hardware or software or products such as anti-viruses and firewalls, on the other they can offer or sell related information bugs, problems or illegal accesses (or accesses not explicitly provided for) to the computer systems that use those same products. Depending on the opportunity, the latter can therefore operate for or against IT security.

Definition 26. ***Double agents*** *can take on a double role: on the one hand, work on safety products information technology, such as security hardware or software, or products such as anti-viruses and firewalls, on the other hand, can offer or sell information relating to bugs, problems or illegal accesses (or accesses not explicitly provided for) to the computer systems that they use those same products.*

Depending on the opportunity, the latter can therefore operate for or against IT security.
At this point we can distinguish three categories relating to computer security strictly related to human work: a first category concerns the deliberate causes

of weakening of computer security in which man is the object or the medium of a cyber attack, or the attacked or the user of a service uncle (*cognitive hacking*, section 2.2); the second concerns the deliberate causes of weakening of security in which the man is the active subject of an attack, or rather the attacker (*hacker profiling*, section 2.3); the third dimension concerns the non-deliberate causes of security weakening related to the work of man as a user or security operator.

2.3 Victims profiling

At this stage we can move to the central argument of this paper which is the study of passwords. We will focus on user profiling with regards to choosing their passwords. In fact, by studying the users it is possible to cluster them on the basis of some particularities. Such as age, job, level of education etc. These characteristics, if known and analyzed in depth, allow the attacker to narrow the search field. Most of the password studies have focused on the computer side, therefore paying attention to how strong passwords are by calculating, for example, entropy trying to find the best methods to improve passwords guessing to find a way to make passwords unguessable. Therefore studying the characteristics of passwords as a textual element without considering who creates them (the users). (see, e.g. [83, 102, 104])
On the other hand, some studies have also focused on the human aspect, such as those conducted by Dr. Helen Petrie, a professor of human/computer interaction at City University in London. She says that computer passwords are "a 21st century Rorschach inkblot test[1]." She analyzed the responses of 1200 volunteers who participated in a survey funded by CentralNic, an Internet domain-name company [95]. The survey's result divides the participants in four genres:

- **Family-oriented** respondents numbered nearly half of those surveyed. These people select their own name or nickname, the name of a child, partner or pet, or birth date. They tend to be occasional computer users and have

[1]The Rorschach test is a psychological test in which subjects' perceptions of inkblots are recorded and then analyzed using psychological interpretation, complex algorithms, or both. Some psychologists use this test to examine a person's personality characteristics and emotional functioning. [84]

strong family ties. They choose passwords that symbolize people or events with emotional value.

- **Fans** which were one of third of all participants, using the names of athletes, singers, movie stars, fictional characters, or sports teams. Those people are, generally, young and want to ally themselves with the lifestyle represented by a celebrity. Two of the most popular names were *Madonna* and *Homer Simpson*.

- 11% of responses were from **fantasists**. Those people are particularly interested in erotism and this is evident in their passwords which are, for example, "sexy," "stud" and "goddess." The researcher discovered that traditionally, these individuals are male, but 37 percent of fantasists identified themselves as female.

- The final 10% of participants are **cryptics**, because they pick unintelligible passwords or a random string of letters, numerals and symbols, such as Jxa+157. Petrie says cryptics are the most security-conscious group. They tend to make the safest but least interesting choices.

So, passwords are inadvertently revealing for two reasons. First, they are generated on the spot. Since users are focused on getting into the system, they are likely to put down something that comes readily to mind. In this sense, passwords tap into things that are just below the surface of consciousness, much the way Rorschach and word-association tests do. Also, to remember the password users pick something that will stick in their mind. They may unconsciously choose something of particular emotional significance [42, 56].
Another study has been conducted from Ian Urbina (an investigative reporter), in an article published in the New York Times, in which he shared insights from investigative journalism into the secret lives of passwords and the psychology behind choices for these strings of letters and numbers. The trend is clear: despite consistent education on the weakness of our favorite passwords, users still clinging on [49]. Every time a user type passwords he is sentimentally involved in. But this sentimentality is putting users at risk. The sense of privacy from these intimate details appears to be a more powerful force than a logical understanding of

security. As a result, passwords often better serve emotional needs than security. About that, researchers Bonneau and Preibusch [67] claim that ineffective passwords, encouraged by sites with poor security standards, are in reality more of a psychological placebo for security than reliable protection for our data.
Password ignorance is supported by another unhelpful psychological force. The attachment to "keepsake" passwords is matched by an equally human inability to evaluate risk. Despite years of hearing the message for better security and repeated exposure to threats, hacking (and the ways this can impact human lives) is not a risk we feel as strongly as some others, at least until we are personally impacted by it. Passwords are not the only risk our brains fail to respond to rationally. Jeunese Payne, a Research Associate at the Cambridge University Computer Lab, draws the comparison with our fear of flying compared to car travel, or our inability to perceive the risk of smoking [103]. To say that, sometimes knowledge can be completely ineffective at changing behavior.
So, while users seem more inclined to choose bad passwords, attackers have the tools to take advantage of that human fallibility. And those tools are not just technical. Since attackers do a lot of social engineering and the combination of their knowledge with the use of tools produces a half-man and half-machine he can codes his way into, for example, the victim's bank account. In the end, we can say that those personal passwords that are private, unique, and special are in fact typical, predictable, and not at all special since they leave many people open to attack (see Chapter 2 for attacks and Chapter 4 for an in-depth study on passwords).

2.4 Good rules to be adopted

At this point, after introducing the analysis on the victims, we describe the so-known good rules to be adopted to make strong passwords and, in the next section, we will analyze how much these "good rules" are actually helpful for users.

To increase the strength of user-chosen passwords, users are typically required to

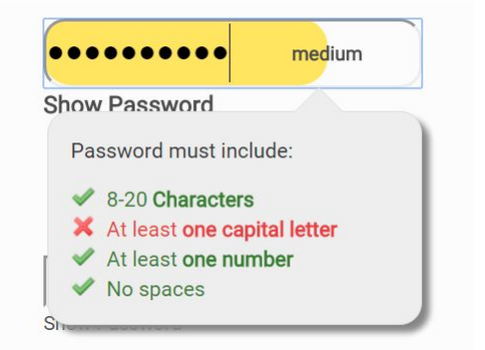

Figure 2.1: Password strength measure on input field [39]

follow a set of rules known as password guidelines when creating passwords (see Fig. 2.1). Users compose their passwords following the specific requirements given in the guidelines.
For example, a common set of rules is [2]:

- Use at least 8 characters
- Use combination of different characters
- Use at least one uppercase
- Never use common information in your password
- Never use the same password twice

But it is not so rare to find additional guidelines. As listed in [53] users should create unique passwords every time, change their passwords for all the accounts once every 6 months, never write down passwords, do not share with anyone, never keep the same password for two different sites, etc.
It's clear that adhering to all these rules is difficult for users since on average a person uses 25 online password-required accounts and uses eight passwords per day [78]. However, nowadays, users may even have far more than 25 passwords. As users are expected to use different passwords for every account to avoid security failures it is difficult for the brain to recollect many discrete sets of illogical and random bits of data then associate each set with which account. The user's response to the present situation is usually adopting strategies like choosing weak

passwords or writing them down, which ultimately undermine the security of the systems they use [87]. Some methods are wont to replace this subversive behaviour with appropriately suitable behaviour for authentication [126]. These methods aim to direct user behaviour by implementing strict password creation guidelines [128], proactive password checkers [127] or password expiry [129], to make sure a high security level.

As showed in [128] users rarely change their password unless forced to do so. In fact only the five per cent of people that have participated on their study, change their password at least one time per year. Then generating new passwords which must conform to a strict security policy is a non-trivial interruption to users' activities. Password policies are highly restrictive - but users are unclear about what the rules are. Next, passwords that are used very frequently are remembered easily; 59 unique passwords were said to be remembered "automatically" in this way. In the end, forgetting a password is always an interruption; but, in some cases, "remembering by a reset" might be a reasonable strategy in situations such as returning from vacation or for infrequently used passwords. In fact, users try a series of passwords that they use frequently if no one of them works they reset the password. When there is a password reset the effort and, more importantly, the time delay involved in resetting passwords raises a genuine fear of forgetting; considering the disruptions it causes to users' tasks and productivity. Some participants of the study reported that they are too lazy or too busy to open the email, think about another password, wait that the password is saved, and then restart to do their work.

The generation stage of the user password management lifecycle is arguably the most important yet perilous step. Fulfilling minimum length and character type requirements while attempting to create something memorable can become an arduous task, leaving the users frustrated and confused. Common user behaviors when choosing passwords turn out to be the most common mistakes to avoid.

Every time we make a decision mental process of decision-making starts. Rational thinking and decision-making does not leave much room for emotions. In fact, emotions are often considered irrational occurrences that may distort reasoning. However, there are some theories and research for both rational decision-making and emotional decision-making focusing on the important role of emotions

in decision-making and the mental process and logic on the important role in rational decision-making.
Since choosing a password is also a choice, it involves the triggering of certain emotions. Indeed from a study conducted by Google [54] the 75% of Americans are frustrated with passwords. Therefore, once again, the psychological dimension behind any type of choice is clear, especially the choice of passwords. We will explore the password theme in the next chapter by analyzing its importance and characteristics and describing how the choice process takes place.

Conclusions In this chapter we have focused on the users, on those who are attacked. We have analyzed the relationship between human and information technologies, finding some vulnerabilities. We have introduced some good rules to adopt when creating a password and how human psychology is involved in making choices.

Chapter 3

Passwords

In this chapter we will discuss how user authentication works and the importance of the use of passwords.Then, we will describe the vulnerabilities of passwords introducing, also, the concept of congnitive dissonance and neutralization.

3.1 User authentication

In most computer security contexts, user authentication is the fundamental building block and the primary line of defense. User authentication is the basis for most types of access control and for user accountability.

An authentication process which is the process of verifying an identity claimed by or for a system entity consists of two steps (see Figure 3.1):

- **Identification step:** the ability to identify uniquely a user of a system or an application that is running in the system.

- **Verification step:** the ability to prove that a user or application is genuinely who that person or what that application claims to be.

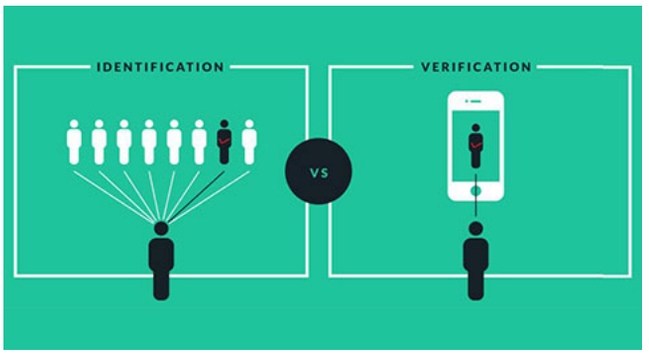

Figure 3.1: Identification and Verification step [26]

The initial requirement for performing user authentication is that the user must be registered in the system. An applicant applies to a Registration Authority (RA), which is a trusted entity that establishes and vouches for the identity of an applicant to a subscriber of a credential service provider (CSP). The CSP then engages in an exchange with the subscriber. Depending on the details of the overall authentication system, the CSP issues some sort of electronic credential to the subscriber. The credential is a data structure that authoritatively binds an identity and additional attributes to a token possessed by a subscriber, and can be verified when presented to the *verifier* in an authentication transaction. The token could be an encryption key or an encrypted password that identifies the subscriber. Once a user is registered as a subscriber, the actual authentication process can take place between the subscriber and one or more systems that perform authentication and, subsequently, authorization. The party to be authenticated is called a claimant and the party verifying that identity is called a *verifier*. When a *claimant* successfully demonstrates possession and control of a token to a *verifier* through an authentication protocol, the *verifier* can verify that the claimant is the subscriber named in the corresponding credential. The *verifier* passes on an assertion about the identity of the subscriber to the Relying Party (RP). That assertion includes identity information about a subscriber, such as the subscriber name, an identifier assigned at registration, or other subscriber attributes that were verified in the registration process. The *RP* can use the authenticated information provided by the *verifier* to make access control or authorization decisions.

There are four general means of authenticating a user's identity, which can be used alone or in combination:

- **Something the individual does** (e.g recognition by voice pattern, handwriting characteristics, and typing rhythm)
- **Something the individual is** (e.g recognition by fingerprint, retina, and face)
- **Something the individual knows** (e.g password, a personal identification number (PIN), or answers to a prearranged set of questions)
- **Something the individual possesses** (e.g electronic keycards, smart cards, and physical keys. This type of authenticator is referred to as a token)

We will now concentrate on something the individual knows which are: *passwords*.

3.1.1 Passwords

A password is a string of characters used to verify the identity of a user during the authentication process. Passwords are typically used in conjuncture with a username; they are designed to be known only to the user and allow that user to gain access to a device, application or website. Passwords can vary in length and can contain letters, numbers and special characters. Other terms that can be used interchangeably are a passphrase when the password uses more than one word, and a passcode and a passkey when the password uses only numbers instead of a mix of characters, such as a personal identification number.
As nouns the difference between passcode and passkey is that passcode is a string of characters used for authentication on a digital device while passkey is a key, especially in a hotel, that allows someone in authority to open any door [37].
A widely used line of defense against intruders is the password system. Virtually all multiuser systems, network-based servers, Web-based e-commerce sites, and other similar services require that a user provide not only a name, email, or identifier (ID) but also a password. When a user logs into a system the password is compared to a (decripted) previously stored password for that user ID, maintained

in a system password file or in a database.
There are many authentication options available today so that users do not have to rely on passwords that can be easily cracked or compromised. There are numerous predictions that passwords would soon be a thing of the past. The recurring idea in computer security is that "The password is dead" [20, 47] . The reasons often include reference to the usability as well as security problems of passwords. However, in spite of these predictions and efforts to replace them passwords are still the dominant form of authentication on the web. The authors of [66] examine why passwords have been proved to be so hard to be supplanted; in examining thirty representative proposed replacements with respect to security, usability and deployability they conclude "none even retains the full set of benefits that legacy passwords already provide." In "The Persistence of Passwords," the authors suggest that every effort should be made to end the "spectacularly incorrect assumption" that passwords are dead. They argue that "no other single technology matches their combination of cost, immediacy and convenience" and that "passwords are themselves the best fit for many of the scenarios in which they are currently used" [81].

3.1.1.1 Hashed password

For security reasons, a widely used password security technique is the use of hashed passwords and a salt value. The password and salt serve as inputs to a hashing algorithm to produce a fixed-length hash code. The hashed password is then stored, together with a plaintext copy of the salt. The hashed password method has been shown to be secure against a variety of cryptanalytic attacks [22].
The salt prevents duplicate passwords from being visible in the password file, increases the difficulty of offline dictionary attacks and becomes nearly impossible to find out whether a person with passwords on two or more systems has used the same password on all of them.

3.1.2 Other authentication options

Since passwords are easy to forget, difficult to manage across a variety of systems and easily susceptible to major hacks some alternatives are available. For example:

- **One-time password:** string of characters or numbers that authenticates a user for a single login attempt or transaction. An algorithm generates a unique value for each one-time password by factoring in contextual information, like time-based data or previous login events. The benefits on using it are: resistance to replay attacks, difficult to guess, reduced risk when passwords are compromised, easy adoption. There are also two types of OTPs: hard tokens (physical devices that transmit OTPs) and soft tokens (push notifications or SMS messages) [55].

- **Security Token:** A security token is a portable device that authenticates a person's identity electronically by storing some sort of personal information. The owner plugs the security token into a system to grant access to a network service. Security Token Services (STS) issue security tokens that authenticate the person's identity.
  There are four different ways in which this information can be used are static password token, synchronous dynamic password token, challenge response token and asynchronous password token. Unlike a password, a security token is a physical object. This object may be in the form of a smart card or may be embedded in a commonly used object such as a key fob which is practical and easy to carry, and thus, easy for the user to protect. Even if the key fob falls into the wrong hands, however, it can't be used to gain access because the PIN (which only the rightful user knows) is also needed.

- **Biometric:** Biometrics is the measurement and statistical analysis of people's unique physical and behavioral characteristics. The technology is mainly used for identification and access control or for identifying individuals who are under surveillance. The basic premise of biometric authentication is that every person can be accurately identified by their intrinsic physical or behavioral traits. The term biometrics is derived from the Greek words bio, meaning life, and metric, meaning to measure. Examples are: facial recognition, fingerprints, iris recognition, vein recognition, signature, voice [12]. The Figure 3.2 shows the cost in relation to acuracy. Iris analysis is the most accurate but also the most expensive technique while voice analysis is the least accurate but also the least expensive. In the middle we find the analysis

of the retina and the finger. Depending on application, user authentication on a biometric system involves either verification (analogous to a user logging on to a system by using a memory card or smart card coupled with a password or PIN) or identification (the individual uses the biometric sensor but presents no additional information).

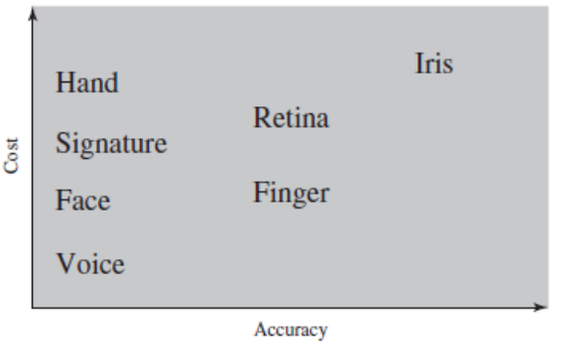

Figure 3.2: Cost versus Accuracy (Biometric user authentication) [26]

- **Single sign-on:** is claimed to eliminate the need for having multiple passwords. It facilitates the use of network resources as it allows a user to access multiple applications and corporate domains using a single set of username and password credentials. Through Single Sign-On, the user logs in once and gains access to different applications without the need to re-enter the login credentials in each application.

- **Cognitive passwords:** Cognitive passwords are based on personal facts, interests, and opinions that are likely to be easily recalled by a user. Cognitive passwords involve a dialogue between a user and a system, where a user answers a rotating set of questions about highly personal facts and opinions. A set of such brief responses replace a single password. [131] The core of a cognitive password system lies the cues. These can be photos of faces, newspapers, images, or other graphical or textual cues. One early method of assisting recall recommended the now later security questions. These questions

were designed to be more memorable than the standard username/password authentication method. As such, a measure of the strength of a cognitive password is the memorability/guessability ratio.

3.2 Longevity and reuse

One of the most common threats, and serious vulnerabilities, is not related at all to software or applications as explained in the previous sections, but rather human beings and our habits: password reuse. Password reuse is a problem in which people try to remember multiple passwords for everything they interact with regularly using the same password on multiple systems, tiers of applications, or even social sites. A recent study of LastPass shows us that the average LastPass [52] personal user has approximately 38 online accounts so the vulnerability is in a person's inability to remember a lot of passwords and exploitable using the same one on every account. Password's administrator to avoid this type of problem introduces the *password aging* in their system. They force users to change passwords frequently. Such policies usually provoke user protest and foot-dragging at best and hostility at worst. There is often an increase in the number of people who note down the password and leave it where it can easily be found, as well as help desk calls to reset a forgotten password. Users may use simpler passwords or develop variation patterns on a consistent theme to keep their passwords memorable. The risk is that once one account is compromised, all of the accounts that share that password become compromised.

A study conducted by [41] found that more than 99% of enterprise users reuse passwords, either across work accounts, or between work and personal accounts. Password reuse is widely prevalent due to the desire for convenience and speed when navigating various accounts. The report also discovered that on average, every single user password is shared across 2.7 accounts.

The more a password is reused, the more opportunities there are for that password to be compromised or stolen. If a website is compromised, hackers will use the passwords and login information on other websites in attempt to gain access to other accounts such as financial websites or email websites. Thus, instead of simply losing access to that one compromised account, people may find themselves dealing

with a cascade of issues, with devastating results for your privacy and online security. This type of habit has given rise to a new type of attack: Password Spraying.

Password spraying is a "type of brute-force attack. This attack attempts to access a large number of accounts (usernames) with a few commonly used passwords. This technique, which is leveraged in 40% of Microsoft account compromises, allows the actor to remain undetected by avoiding rapid or frequent account lockouts" [41].

Password spray campaigns typically target Single Sign-On (*SSO*) and cloud-based applications utilizing federated authentication protocols. Targeting federated authentication can help mask malicious traffic. Additionally, targeting *SSO* applications helps maximize access to intellectual property if the attack succeeds

When targeting end user devices and accounts, such as software as a service (*SaaS*) and corporate intranet logins, adversaries rely on spraying perennial password favorites, very few of which have changed over time. In 2019, the top 10 most commonly used passwords leaked in data breaches were: *123456, password, 123456789, 111111, 12345678, qwerty, 12345, Iloveyou* and *1234567*. In the case of system accounts and infrastructure devices over administrative protocols such as SSH and Telnet attackers shift to the most common passwords which are, for example: *admin, password, 12345. default.*

Another attack against password reuse is *credential stuffing*. It involves taking credentials compromised in one breach and replaying those same credentials on other sites and applications. Unlike credential cracking, credential stuffing attacks do not attempt to use brute force or guess any passwords – the attacker simply automates the logins for a large number (thousands to millions) of previously discovered credential pairs using standard web automation tools such as Selenium, cURL, PhantomJS. An example of this attack happened in 2016, attackers gained access to a private GitHub repository used by Uber (Uber BV and Uber UK) developers, using employees' usernames and passwords that had been compromised in previous breaches. The hackers claimed to have hijacked 12 employees' user accounts using the credential-stuffing method, as email addresses and passwords had been reused on other platforms [8].

To avoid these attacks there are many solutions such as Multifactor authentication

(MFA), cybersecurity education, password manager programs, proper configuration and enhanced password screening tools to help mitigate human bad habits. In February 2018 was created a communication protocol (using *k-anonymity* and *cryptographic hashing*), implemented as a public API and is now consumed by multiple websites and services, including password managers and browser extensions, to anonymously verify whether a password was leaked without fully disclosing the searched password [18].

3.3 Password policy

A password policy is a set of rules designed to enhance computer security by encouraging users to employ strong passwords and use them properly. A password policy is often part of an organization's official regulations and may be taught as part of security awareness training. Either the password policy is merely advisory, or the computer systems force users to comply with it. Policies suggest or impose requirements on what type of password a user can choose, such as the minimum and maximum length, character restrictions, frequency of password reuse, minimum password age. National Institute of Standards and Technology (NIST), a United States government agency that deals with technology management, in 2021 lists some best practices and recommendation [33].

Recommendation

- Remove periodic password change requirements
- Require length but remove password complexity
- Implement screening of new passwords

NIST 2021 Best Practices

- Minimum password length
- Password policies and password policy management

- Use a password manager which is a software application designed to store and manage online credentials. Usually, these passwords are stored in an encrypted database and locked behind a master password. Although it is recommended to use a password manager to choose passwords, this practice is not very common among users. A recent study [76] noticed that who uses password managers noted convenience and usefulness as the main reasons behind using the tool, rather than security gains, underscoring the fact that even a large portion of users of the tool are not considering security as the primary benefit while making the decision. On the other hand, who does not use password managers noted security concerns as the main reason for not using a password manager, highlighting the prevalence of suspicion arising from lack of understanding of the technology itself. Finally, analysis of the differences in emotions between "users" and "non-users" reveals that who never use a password manager are more likely to feel suspicious compared to "users," which could be due to misunderstandings about the tool.

CIS also contributed by describing a guide on password policies. Regarding the creation of passwords [11]

Password Creation

- Use "passphrases" instead of passwords
- Don't use words related to your personal information
- Limit using dictionary words

System Recommendations

- Use Multi-Factor Authentication (MFA)
- Offer Password Managers
- Use more sophisticated access lockout techniques

3.4 Password cracking and attacks

Password cracking is the process of attempting to gain unauthorized access to restricted systems using common passwords or algorithms that guess passwords. In other words, it's an art of obtaining the correct password that gives access to a system protected by an authentication method.
Password cracking employs a number of techniques to achieve its goals. The cracking process can involve either comparing stored passwords against word list or use algorithms to generate passwords that match.
We can identify the following attack strategies and countermeasures:

- **Brute-Force Attacks:** is the cyberattack equivalent of trying every key on your key ring, and eventually finding the right one. This attack involves setting up an automated script to literally attempt all possible combinations of characters for that password. An example of this might be to start with "a", then "b", then "c" and continue until "z", at which point the program would try "aa", then "ab" and so on.

- **Dictionary Attacks:** is a form of brute force attack technique for defeating a cipher or authentication mechanism by trying to determine its decryption key or passphrase by trying thousands or millions of likely possibilities, such as words in a dictionary or previously used passwords, often from lists obtained from past security breaches.

- **Combined Dictionary Attacks:** This type of attack on difficult and compound passwords is very similar to the simple dictionary attack, except that instead of using a single word for password verification here we use a combination of words or a phrase created by combining words from specified dictionaries.

- **Hybrid Dictionary and Rule-Based Dictionary Attacks:** is the method of taking the words listed in a dictionary and combining them with a brute-force attack. Rule based attack is used when attacker gets some information about the password. This technique involves use of brute force, dictionary and syllable attacks.

- **Rainbow Table Attacks:** A rainbow table is a pre-compiled table used for recovering hashes. Each rainbow table is for a specific length of password containing a well-defined set of characters. In this approach the attacker generates a large dictionary of possible passwords. For each password, the attacker generates the hash values associated with each possible salt value. The result is the rainbow table.

- **Markov Chains Attacks:** A Markov Model is a sequence of events for which the probability is dependent only on the event immediately preceding it. To use the Markov Chains technique attackers need to assemble a certain password database, split each password into n-grams (sequences of n number of elements, which may consist of characters or words), develop a new alphabet where these different elements act as letters and then match it with the existing password database.

3.5 Password strength

Password Strength, also known as Password Entropy, is the measure of password strength or how strong the given password is. Password entropy is based on the character set used (which is expansible by using lowercase, uppercase, numbers as well as symbols) as well as password length. It predicts how difficult a given password would be to crack through guessing, brute force cracking, dictionary attacks or other common methods (explained in section 3.4).
Password entropy is usually expressed in terms of **bits**: a password that is already known has zero bits of entropy; one that would be guessed on the first attempt half the time would have 1 bit of entropy. A password's entropy can be calculated by finding the entropy per character, which is a log base 2 of the number of characters in the character set used, multiplied by the number of characters in the password itself. As explained in section 3.3 NIST provides some guidelines to make strong passwords. These guidelines are for user-selected passwords with 30 bits of entropy:

- Use a minimum of 8 characters selected from a 94-character set.

- Include at least one upper case letter, one lower case letter, one number and one special character.

- Use a dictionary of common words that users should avoid, like a *password blacklist* which is a list of words disallowed as user passwords due to their commonplace use. Blacklists can prevent the use of a string of characters that might pass password entropy checks. For example, *P4ssWord3* has a good entropy and is rated as an acceptably strong password in many password strength meters because it employs several password hardening measures, but it is just the word *password* (one of the most common passwords) with typical modifications.

- Do not use any permutations of your username as your password so it is weak.

The password length parameter is a basic parameter for the password guessability. The value of which affects password strength against brute force attack. The following formula shows the P probability that a password can be guessed in its L maximum lifetime where R is the number of guesses per unit of time, and S is the number of algorithm-generated passwords:

$$P = L * \frac{R}{S}$$

The following table shows the time required for the Brute Force attack based on the length of the password, the character set used and the use of a single computer with the speed of 500,000 keys per second.

Password length	Uppercase	Lower case and digits	Upper and lower case	ASCII
≤ 4	immediate	immediate	immediate	2 minutes
5	immediate	2 minutes	12 minutes	4 hours
6	10 minutes	72 minutes	10 hours	18 days
7	4 hours	43 hours	23 days	4 years
8	4 days	65 days	3 years	463 years
9	4 months	6 years	178 years	444530 years

Table 3.1: Time required for the Brute Force attack based on the length of the password

The formula for **entropy** is [94]:

$$E = log_2(R^L)$$

Where E is the password entropy, R the pool of distinct characters, L the number of character of the password. So R^L corresponds to the number of possible password and $log_2(R^L)$ to the number of bits of entropy.

Symbol set	**Symbol count N**
Arabic numerals	10
Lowercase	26
Alphanumeric	36
Lower Upper Case	52
Alphanumeric Upper Case	62
Common ASCII Characters	30
Diceware Words List	7776
English Dictionary Words	171000

Table 3.2: Entropy per symbols.

The **power of a brute force attack** can be quantified through a formula that calculates the number of all possible combinations before finding the correct key [21]:

$$NT = L^m + L^{m+1} + ... + L^M$$

Where NT is the numeric total of attempts, L is the length of the character set, m is the minimum key length, M is the maximum key length.

Example 3.5.1. If an information system requires to create password with at least 5 characters and at most 7 characters from the character set with lowercase letters a-z, uppercase letters A-Z and digits 0-9. The total number of possible passwords which can be created from it is:

$$NT = \sum_{k=5}^{7} (26 + 26 + 10)^k = 3.5 * 10^{12}$$

Then, knowing that the time required to an attacker to crack a password is given by [38]:

$$T = NT * rate * accuracy$$

where rate indicates the amount of time taken by the information system to guess a password and accuracy indicates the information system's ability to guess pass-

words correctly.

Assuming that attacker uses a machine with a test capacity of cracking 2.5 million passwords/second and on an average success is achieved if it can test 75% of the overall number of the password. The time required by the attacker to crack the password of the previous example is:

$$T = 3.5 * 10^{12} * \frac{1}{2.5 * 10^6} * \frac{75}{100} = 1050000s$$

Which corresponds to 12 days to crack the password using brute force attack.

Finally, knowing that the password entropy formula is: $E = log_2(R^L)$ and applying it to the example the result is:

$$E = log_2(62^{10}) = 59.54bits$$

That are bits of entropy per character which correspond to password strength.

Required bits of entropy of a password A password that is already known has zero bits of entropy. A password that requires at most 2 guesses to find has 1 bit of entropy. A password with n bits of entropy would require 2^n guesses to guarantee that password will be found.

"Randomness Requirements for Security" [75], presents some example threat models and how to calculate the entropy desired for each one. The minimum number of bits of entropy needed for a password depends on the threat model for the given application. Passwords with more entropy are needed if key stretching is not used.

Storing passwords in plain text is really insecure. In fact if this list is leaked, someone knows all the passwords with no effort.

Now if the user has a naive password you could not crack the password by doing a simple search. The attacker can find the hash value of the password by searching, but not the hash value of qwerty + random salt, because although the former is common the latter is probably unique. The attacker could still crack the password if the hash is insecure, but it would take a little effort.

If an attacker has a list of salt values and corresponding hash values for salt +

password. Is possible to guess passwords, hashing each with a salt value, to see if any hash values match. Key stretching is a way to make this brute force search more time consuming by requiring repeated hashing. In the following stretching algorithm, p is the password, s is the salt, h is the hash function, and $||$ means string concatenation. [48]

$$x_0 = \phi$$

$$x_i = h(x_{i-1}||p||s) \forall i = 1, .., r$$

$$K = x_r$$

Now the time required to test each password has been multiplied by r. The idea is to pick a value of r that is affordable for legitimate use but expensive for attacks. Key stretching leaves an attacker with two options. The first one is to attempt possible combinations of the enhanced key, the second one is attempt possible combinations of the weaker initial key, potentially commencing with a dictionary attack if the initial key is a password or passphrase. Attackers could guess passwords,also, starting, for example, with the most common passwords (see 3.5.1) and get some matches.

Password strength is determined with this chart [40]:

Bits	**Strength**
$<$ 28 bits	Very Weak
28 - 35 bits	Weak
36 - 59 bits	Reasonable
60 - 127 bits	Strong
128+ bits	Very Strong

Table 3.3: Password strength per bits.

3.5.1 Power laws in Passwords

In the literature, it has been shown that password databases follow some power laws. These include Zipf law, Pareto rule and Brevity law.

3.5.1.1 Zipf's law

The result of the studies of the authors of the paper [74] led to the fact that the empirical distribution of real passwords follows a power law. In particular they investigated whether *Zipf's law* also exists in passwords.

Definition 27. ***Zipf's law*** *[59] is an empirical law that describes the frequency of an event P_i which is part of a set, as a function of the position i (called rank) in the ordering decreasing with respect to the frequency of this event.*

Zipf's law was originally formulated in terms of quantitative linguistics, stating that given some corpus of natural language utterances, the frequency of any word is inversely proportional to its rank in the frequency table.
The frequency of the rth most common password (f_r) is proportional to $\frac{1}{r}$. More precisely we have: $f_r = Cr^{-s}$ where s is on the order of 1 and C is a constant depending on the particular corpus. This means that the most frequent word will occur about two times as often as the second most frequent word, three times as often as the third most frequent word, and so on. Meaning that the most common passwords are very common so easy to guess (we applied this study to the RockYou data leak 4.3).

Under the Zipf model, the number of times we would expect to see the most common password is NC where N is the size of the data set and C is the constant what it has to be for the frequencies to sum to 1. C depends on the number of data points N and the exponent s and is given by

$$C_{N,s} = \frac{1}{\sum_{r=1}^{N} r^{-s}}$$

Example 3.5.2. Knowing that the range of s values found by the authors of the paper [74] varied from roughly 0.5 to 0.9.
If we have N = 1,000,000 passwords. Let's first set s = 0.5. Then C is roughly 0.0005.
This mean the most common password appears about 500 times.

3.5.1.2 Pareto's rule

If passwords come from an alphabet of size A and have length n, then there are An possibilities. For example, if a password has length 10 and consists of uppercase and lowercase English letters and digits, there are $62^{10} = 839,299,365,868,340,224$ possible such passwords. If users chose passwords randomly from this set, brute force password attacks would be impractical. But, since passwords are not chosen uniformly from this large space of possibilities, the attack becomes practical. Attackers do not randomly try passwords, they try with the most common passwords and work their way down the list applying the *Pareto's rule* [98].
The Pareto principle is a statistical-empirical result that is found in many complex systems endowed with a cause-effect structure.

Definition 28. *[36] The Pareto principle states that about 20% of the causes cause 80% of the effects.*

These values are to be understood as qualitative and approximate.

3.5.1.3 Brevity law

It's a statistical regularity that can be found in natural languages and other natural systems and that claims to be a general rule. We introduced it following the statistics made in Chapter 4 after noting that the more the password length increased, the lower the frequency of passwords of that length.

Definition 29. *[130]* ***Brevity law****, (called Zipf's law of abbreviation, too) is a linguistic law that qualitatively states that the more frequently a word is used, the shorter that word tends to be, and vice versa.*

3.5.2 Random passwords

Random passwords consist of a string of symbols of specified length taken from some set of symbols using a random selection process in which each symbol is equally likely to be selected. The symbols can be individual characters from a character set, syllables designed to form pronounceable passwords, or even words from a word list (thus forming a *passphrase*).

A *passphrase* is a sentence-like string of words used for authentication that is longer than a traditional password, easy to remember and difficult to crack [106]. Typical passwords range from 8-16 characters on average while passphrases can reach up to 100 characters in length. Passphrases differ from passwords. A password is usually short six to ten characters. Using a long passphrase instead of a short password to create a digital signature is one of many ways that users can strengthen the security of their data, devices and accounts. A passphrase can also contain symbols, and does not have to be a proper sentence or grammatically correct. The main difference of the two is that passwords do not have spaces while passphrases have spaces and are longer than any random string of letters. Passphrases are better than passwords because are easier to remember, satisfy complex rules easily, are next to impossible to crack because most of the highly-efficient password cracking tools breaks down at around 10 characters. Hence, even the most advanced cracking tool won't be able to guess, brute-force or pre-compute these passphrases. An example of passphrase compared to a password could be the one listed in the table 3.4.

	Difficulty to remember	**Difficulty to hack**
P4$$word!	Hard	Easy
Guessing my horse home	Easy	Hard

Table 3.4: Example of passphrase compared to a password

The strength of random passwords depends on the actual entropy of the underlying number generator; however, these are often not truly random, but *pseudorandom*. A random password generator is software program or hardware device that takes input from a random or pseudo-random number generator (PRNG) and automatically generates a password. In Figure 3.3 is shown the flow of a random password generator and after a pseudo-code is provided. Many publicly available password generators use random number generators found in programming libraries that offer limited entropy. However most modern operating systems offer cryptographically strong random number generators that are suitable for password generation. It is also possible to use ordinary dice to generate random passwords. Random password programs often have the ability to ensure that the resulting password complies with a local password policy; for instance, by always producing a mix of letters, numbers and special characters.

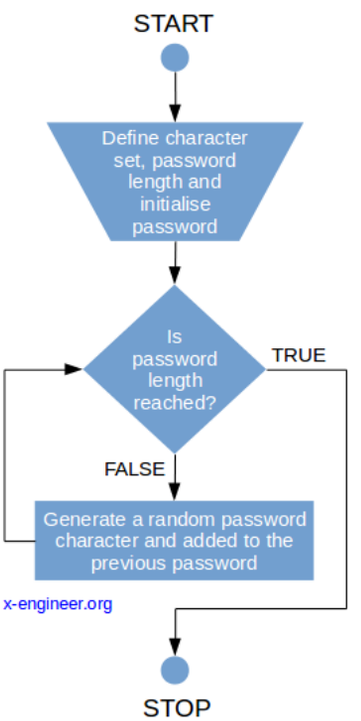

Figure 3.3: Logical diagram of random password generator [45]

Algorithm 1 Random password generation

Data: character_sets, passLength
Result: password
initialization
for $index = 1, 2, \ldots, passLength$ **do**
    character = Math.floor(Math.random() * character_sets.length)
    password = password + character_sets.charAt(character)
end

The Algorithm 1 shows how a password can be generated randomly. Once the character set and the length of the password have been chosen, one character for each for loop is randomly generated and added to the variable that will return the generated password.

For passwords generated by a process that randomly selects a string of symbols of length, L, from a set of N possible symbols, the number of possible passwords can be found by raising the number of symbols to the power L. Increasing either L or N will strengthen the generated password. The strength of a random password as measured by the information entropy is just the base-2 logarithm or log_2 of the number of possible passwords, assuming each symbol in the password is produced independently. Thus a *random password's information entropy, H,* is given by the formula [114]:

$$H = log_2 N^L = L * log_2 N = L\frac{logN}{log2}$$

where N is the number of possible symbols and L is the number of symbols in the password.

Example 3.5.3. If we had a very weak password of 4 characters consisting of letters of the same case:
Length: 4
Possible Symbols: 26
Possible combinations: $26^4 = 456,976$
Bits of Entropy : $log_2(26^4) = 18.80$
Expected Number of guesses $= 2^{Entropy-1} = 2^{18.80-1}$

Example 3.5.4. If we had 5 random words taken from the Diceware wordlist:
Length: 5
Possible Symbols: 7776
Possible combinations: $7776^5 = 2.8430288e + 19$
Bits of Entropy : $log_2(7776^5) = 64.62$**Expected Number of guesses** $= 2^{Entropy-1} = 2^{64.62-1}$

3.5.3 Diceware

Diceware is a method for creating passphrases, passwords, and other cryptographic variables using ordinary dice as a hardware random number generator to select words at random from a special list called the Diceware Word List. Each word in the list is preceded by a five digit number. All the digits are between one and six. A Diceware word list is any list of 6^5=7,776 unique short words, abbreviations and easy-to-remember character strings. The average length of each word is about 4.2 characters. The biggest words are six characters long.
The creator suggests to download the complete Diceware list [13], then decide how many words you want in your passphrase. A five word passphrase provides a level of security much higher than the simple passwords most people use, roll the dice and write down the result, in the end, look up each five digit number in the Diceware list and find the word next to it. For example, 11326 means your next passphrase word would be "adonis". An example of Diceware computation is:
supposing we want a six word passphrase. We will need 6 times 5 or 30 dice rolls. Let's say they come out as: 1, 2, 6, 5, 5, 2, 5, 6, 5, 5, 3, 5, 2, 3, 5, 2, 2, 4, 5, 5, 6, 2, 6, 6, 5, 1, 2, 3, 5 and 5.
Writing down the results in groups of five rolls a table is created:

1	2	6	5	5
2	5	6	5	5
3	5	2	3	5
2	2	4	5	5
6	2	6	6	5
1	2	3	5	5

Then, looking up each group of five rolls in the Diceware word list [13] by finding

the number in the list the resulting passphrase is: *ave floppy keel verse append*
According to the author, for extra security is possible to add another word, inserting one special character or digit chosen at random into the passphrase. It could be done securely in this way: roll one die to choose a word for the passphrase, roll again to choose a letter in that word. Roll a third and fourth time to pick the added character from the following table:

0	1	2	3	4	5	6
1	∼	!	#	$	%	∧
2	&	*	(	)	-	=
3	+	[	]	\	{	}
4	:	;	”	’	<	>
5	?	/	0	1	2	3
6	4	5	6	7	8	9

The main reasons why the Diceware technique is recommended is that the generated passphrases are easy to remember (the key to a memorable passphrase is imagery. The idea is to take 4 or 5 random words and use those words to create an image in your head. The more ridiculous the image, the easier it will be to remember) and the technique is simple to use. It is secure having a high level of entropy and is completely transparent since the user uses physical (or digital) dice, not relying on websites that create passwords.
The author of [70] discusses some problems on some Diceware. The first drawback of the diceware method with variable-length dictionaries: they have less entropy than it may appear. In the specific case of the 5-word passphrase with the 7,776-word Beale wordlist, it is at a minimum 22.68 6.41 times weaker than it could be if its wordlist was made of fixed-length words.
The second one is that many words in the dictionary are not widely-known words in English, but numbers, symbols, and abbreviations like “25%”, “3000”, “2nd”, “5/8”, “9:30”, etc.
The authors [70] have proposed a modified version of Diceware composed by:

- A smaller dictionary with 6^4 =1,296 word, yielding log_2 64 10.34 entropy bits per word making it much more difficult to design more common and familiar words.

- Make all words have only four characters. This makes the full passphrases always 24 characters long, for a fixed entropy of 62.04 bits which are slightly more than the best entropy of 61.94 from the classic method. The disadvantage are that passphrases are now two characters longer than the average classic ones.

A number of other groups are developing English word lists for generating passphrases. For example: the Electronic Frontier Foundation (EFF) [14] and the Natural Language Passwords (NLP) [15].

This type of password generation differs from those made by humans for the reasons listed in the subsection 2.4 in which are provided some examples.

3.6 Choice of passwords, cognitive dissonance and neutralization

Every day we are forced to choose something. From the most important to the least important. Even when we have to subscribe to a new site or portal we have to choose a password. As mentioned in Section 3.4 several problems arise when choosing a password. The lack of common standards for passwords makes it difficult for a user to remember which password is used for which system. Some systems constrain users to have a certain minimum length, or to require that the password contains a combination of letters and numbers, imposes maximum lengths, and some systems prohibit special characters. So from the human point of view various psychological factors come into play that lead him to make bad choices. In fact, as with all choices, a theory also comes into play when it comes to passwords, the theory of cognitive dissonance. Figure 3.4 shows the schema of the cognitive dissonance theory.

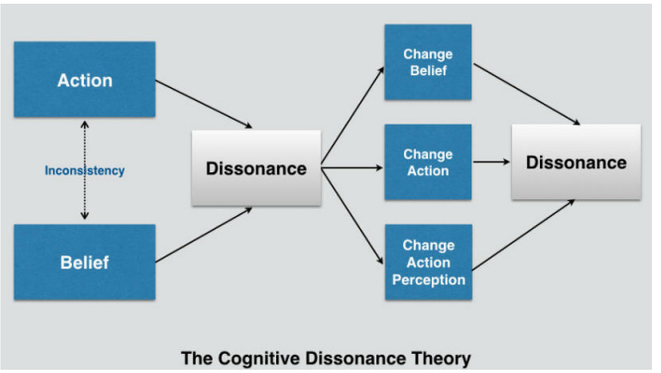

Figure 3.4: Cognitive dissonance schema [1]

Festinger's cognitive dissonance theory suggests that we have an inner drive to hold all our attitudes and behavior in harmony and avoid disharmony (or dissonance). This is known as the principle of cognitive consistency [68]. The term *cognitive dissonance* is used to describe the mental discomfort that results from holding two conflicting beliefs, values, or attitudes. People tend to seek consistency in their attitudes and perceptions, so this conflict causes feelings of unease or discomfort. This inconsistency between what people believe and how they behave motivates people to engage in actions that will help minimize feelings of discomfort. People attempt to relieve this tension in different ways, such as by rejecting, explaining away, or avoiding new information.
Factors that affect the degree of cognitive dissonance that a person experiences include:

- **Forced compliance behavior:** when someone is forced to do (publicly) something they (privately) really don't want to do, dissonance is created between their cognition (I didn't want to do this) and their behavior (I did it).

- **Decision making:** life is filled with decisions, and decisions (as a general rule) arouse dissonance. Password behavior shows a tendency toward cognitive dissonance, an imbalance between knowledge and action. Internet users know how dangerous it is to use the same password with different applications, but they do it anyway. They neutralize the negative feelings by

deliberately suppressing information. For example, they believe they are invulnerable to cyberattacks. This contrasts sharply with the fact that 40% of Germans that have already been the victim of a cyberattack. In the corporate context, an even much higher figure is startling: 96% of all German companies have already suffered a business-damaging cyberattack. In a recent LastPass survey [52] 91% of users say they know using the same or a variation of the same password is a risk. however, when creating passwords, 66% of respondents always or mostly use the same password or a variation – this is up 8%. Then 80% agree that having their passwords compromised is something they're concerned about and yet 48% said if it's not required, they never change their password - which is up from 40% in 2018 and in the end 77% say they are informed of password protection best practices however 54% keep track of passwords by memorizing them.

- **Effort:** it also seems to be the case that we value most highly those goals or items which have required considerable effort to achieve.

In order to reduce this dissonance, individuals are self-motivated either to change their behaviours or beliefs, or to rationalize their behaviour. Neutralization is a technique used by criminals to rationalize maleficence. In terms of the insider threat, it has been proposed that if the justifications for committing an offence are eliminated, then the insider is less likely to commit the offence. This process is known as **neutralization** mitigation. Techniques of neutralization are a theoretical series of methods by which those who commit illegitimate acts temporarily neutralize certain values within themselves which would normally prohibit them from carrying out such acts, such as morality, obligation to abide by the law, and so on. In simpler terms, it is a psychological method for people to turn off "inner protests" when they do, or are about to do something they themselves perceive as wrong.
Matza and Sykes [117] created some methods by which, they believed, "delinquents" (users for us) justified their illegitimate actions and [116] provided some examples for each maethod. We will show two of them, the most suitable in our thesis:

- **Denial of responsibility** in which the offender will propose that they

were victims of circumstance or were forced into situations beyond their control. For example, a denial-of-responsibility argument in the context of information security could be that individuals justify their non-compliance by claiming that they are not well-versed on the company's password guidelines

- **Denial of injury** in which the offender insists that their actions did not cause any harm or damage. In an information security context, individuals can, for example, justify their behavior by claiming that it is acceptable to use simple password at work if no one gets hurt.

- **Condemnation of the condemners** in which the offenders maintain that those who condemn their offense are doing so purely out of spite, or are shifting the blame off of themselves unfairly. In the case of in- formation security, a condemnation-of-the-condemners neutralization is to claim that information security policies are unreasonable.

- **Appeal to higher loyalties** in which the offender suggests that his or her offense was for the greater good, with long term consequences that would justify their actions, such as protection of a friend. In the context of compliance with information security procedures, individuals could utilize the argument of an appeal to higher loyalties by arguing that he or she must violate corporate security procedures to get his or her work done.

- **Entitlement** suggests that people have a right to engage in certain behaviors. In the context of information security behavior, individuals could justify their behavior by saying that they should be free to choose any password they want.

- **Relative acceptability** is used to remove blame for one's actions by pointing out that others are even "worse than me." In the context of the present study, individuals could justify their use of weak passwords by alleging that other individuals' passwords are much weaker than theirs.

- **Defense by comparison**: the person is excusing his or her actions by suggesting that while the action might not be good, the person could have acted even worse (but did not). In the context of our study, a person using

this argument might say that the use of weak passwords at work is not a big deal compared with other issues, such as being lazy on the job.

So [116] ask if individuals' neutralization techniques can be overcome. Doing a training treatment to overcome the neutralization technique in the contest of passwords its result shows that who received the training exhibited substantially less intent to use neutralization techniques and were significantly more likely to use secure passwords. Additionally, a follow-up measurement three weeks after the training session showed that the experimental treatment retained its effectiveness, i.e., the experimental group exhibited substantially less intent to use neutralization techniques and a greater likelihood of using strong passwords in the future. So with the right training it is possible to teach people to fight against an involuntary action committed by their brain.

Conclusions Users are authenticated through passwords that must be properly guarded. But there is not only one type of authentication, in fact, in the chapter, we have also described other methods. We discussed how long a password should be and discussed the problem of always reusing the same password for different systems. We have discussed in depth the policies to be adopted which have been recommended by NIST. We talked about how to crack the strength of a password in terms of cracking times and finally illustrated some power laws to apply to password datasets that will be useful for guessing. Finally, how human psychology is the author of the most common mistakes in creating a password.

Chapter 4

Data leak analysis

In this chapter we will show our contribution and original results on password choices. We have analyzed several data leaks to produce some statistics and to analyze in depth some common patterns. Section 5.1 describe, the datasets of passwords which we have used to conduct our research. Section 5.2 will show how we have applied the Levenshtein distance on passwords to find the frequent replacements. Section 5.3 will explain another way to calculate the password entropy using edit distance. The rest of the sections focus on analyzing distributions, frequencies and categorization of words used as passwords.

4.1 Used datasets

The datasets which we have used to conduct our researches are: **RockYou**, **Hotmail**, **Phpbb** and **Ashley Madison** [46, 25, 44, 5]. We chose these datasets on the basis of the amount of passwords they contain (other datasets were too small to be an acceptable sample) and for the variety between them because belonging to different categories of users we expected a diversification at the string level. All datasets are publicly accessible.
We have chosen to analyze these four data leaks to have the most heterogeneous passwords possible. In the literature we have often found in-depth analyzes on RockYou. But analyzing a single dataset, no matter how large it is, specializes the results only on it. What we want to achieve is to have results applicable to any

given leak as much as possible.
In our studies we have tried to understand the relationship between the most used words in the vocabulary, names, dates, etc, the way passwords are created, the most common patterns, the most common attitudes and the passwords of the various data leaks. To do this we had a dataset of English words that we will call *"most_common_words"*, a dataset of *proper name of person* and some datasets of the most used words also in other languages. Since the most widely spoken languages in America (from which the dataset sites come from) are English, Spanish, Chinese, French and German we decided to use the English dictionary first, then Spanish, French and finally German. As for Chinese, it was difficult to find a good dictionary so it was left out. Figure 4.1 shows the frequency of the word length contained within the *most_common_words* dataset.

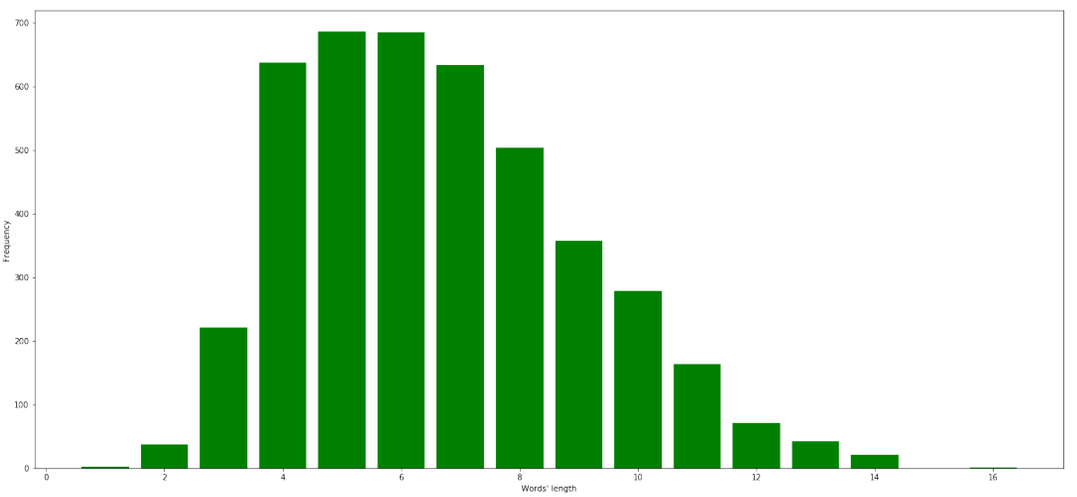

Figure 4.1: Words length frequency in "most_common_words" dictionary

4.1.1 RockYou

The first data leak we are going to analyze is RockYou. RockYou was a company that developed widgets for MySpace and implemented applications for various social networks and Facebook. Since 2014, it has engaged primarily in the purchases

of rights to classic video games; it incorporates in-game ads and re-distributes the games. In 2009, the company suffered a data breach resulting in the exposure of over 32 million user accounts. One of the most problems is that the company used an unencrypted database to store user account data, including plaintext passwords, as well as passwords to connected accounts at partner sites (including Facebook, Myspace, and webmail services). The second problem is that RockYou would also e-mail the password unencrypted to the user during account recovery and account creation only enforced password of a minimal length of 5 characters, there was no requirement for mixed-case, numbers or punctuation. Figure 4.2 shows the mean of the password length of Rockyou.

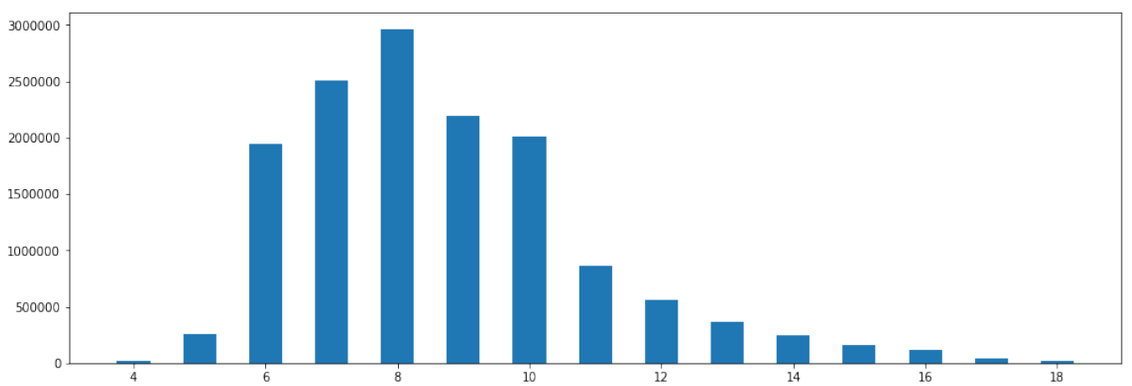

Figure 4.2: Mean password length Rockyou

The platform actually encouraged simple passwords by not allowing any punctuation at all. So the attackers using only a 10-year-old SQL exploit these vulnerabilities to gain access to the database.
Rockyou database is publicly available [46] and it is composed by 14,341,564 unique passwords, used in 32,603,388 accounts.

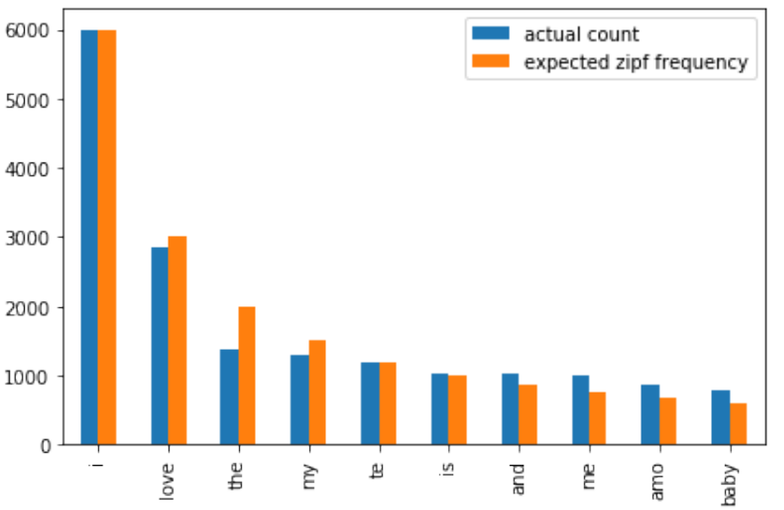

Figure 4.3: Zipf's law - Rockyou

We also want to understand how terms are distributed across the dataset. A commonly used model of the distribution of terms in a collection is Zipf's law (as discussed in Section 3.5.1). It states that, if t_1 is the most common term in the collection, t_2 is the next most common, and so on, then the collection frequency cf_i of the ith most common term is proportional to $1/i$: $cf_i \propto \frac{1}{i}$.. So if the most frequent term occurs 1 times, then the second most frequent term has half as many occurrences, the third most frequent term a third as many occurrences, and so on. The intuition is that frequency decreases very rapidly with rank (see Section 3.5.1 for the definition). In Figure 4.3 is represented the distribution of first ten words in passwords, according to the law. Each bar represents a word. It can be seen that the values obtained are very close to the values expected by the law as show in [74].

Example 4.1.1. In the case of Figure 4.3 the most common term t_1 is "I", the second, t_2, is "love", the third, t_3, is "the" and so on. According to the definition of Zipf's law t_1 occurs $\frac{1}{1}$ times, t_2 occurs $\frac{1}{2}$ times and t_3 $\frac{1}{3}$ times. As you can see in the figure, the first term occurs 6000 times, so the expected frequency of the second one is $\frac{1}{2}$*6000 which is 3000. In fact the second term occurs 3000 times.

The expected frequency of the third one is $\frac{1}{3}$*6000 which is 2000. In this case the third term appears about 1800 which is very close to the expected one. The other terms follow the same formula.

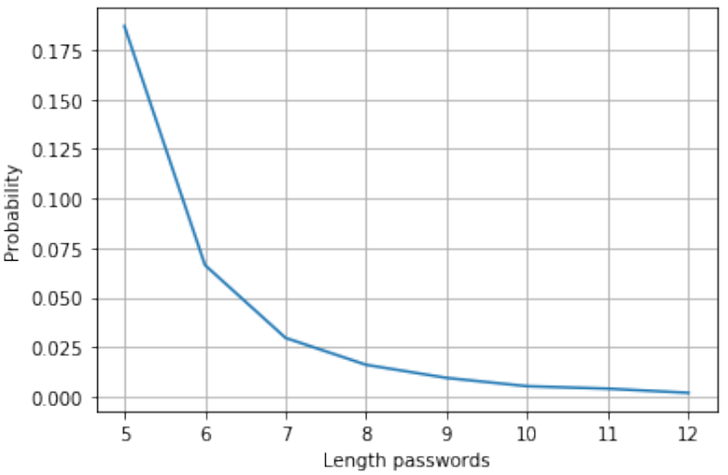

Figure 4.4: Most common words as passwords - Rockyou

The next analysis concerns the probability of having passwords that correspond to the *most_common_words*. The highest probability is that for shorter passwords going down for longer and longer passwords.
We subjected each group of passwords to a comparison with the dictionary of common words and calculated the similarity through a simple algorithm that scrolls the text file containing the passwords and the text file containing the *most_common_words*. According to the **brevity law** (see Section 3.5.1 for the definition) the percentage of passwords that are equal to the most used words in the English dictionary is greater for passwords of length 5 going to scale (see Figure 4.4). Since the dictionary of the most used English words has on average words 6 characters long and the length that has greater frequency is 5 it demonstrates the great influence and use of these most common words (see Figure 4.1).
What we wanted to investigate, therefore, is the use of numbers in various passwords. Given the large number of passwords of length 5, we wondered how the

longer ones are composed. According to the previous point we discovered, in fact, that as the password length increases (see Figure 4.5), the need to use numbers inside it also increases. The reason why this need arises is given by the combination of the use, with greater probability, of short words and the need not to enter a password that is too short. This involves entering numbers, especially at the end, to lengthen the password as discovered in further analyzes described in the next sections.

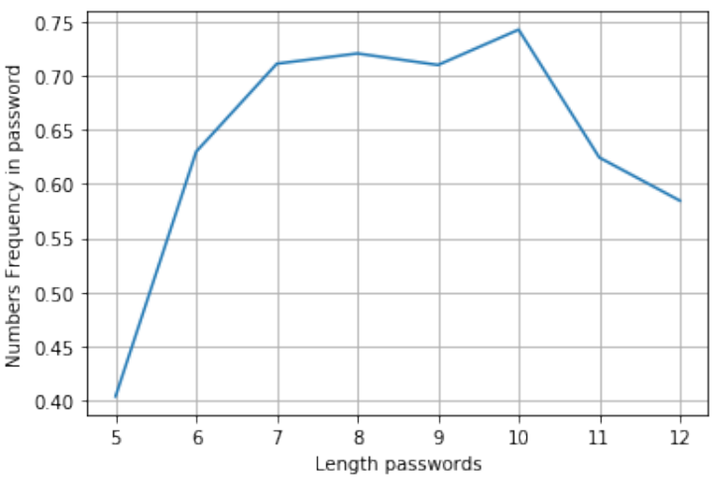

Figure 4.5: Frequency of numbers inside the passwords depending on password length

We made a further analysis. We calculated the frequency of the first character in all passwords in Rockyou and in the *most_common_words* dataset (see Fig. 4.6). It is possible to see from the figure how the two lines follow the trend almost faithfully.

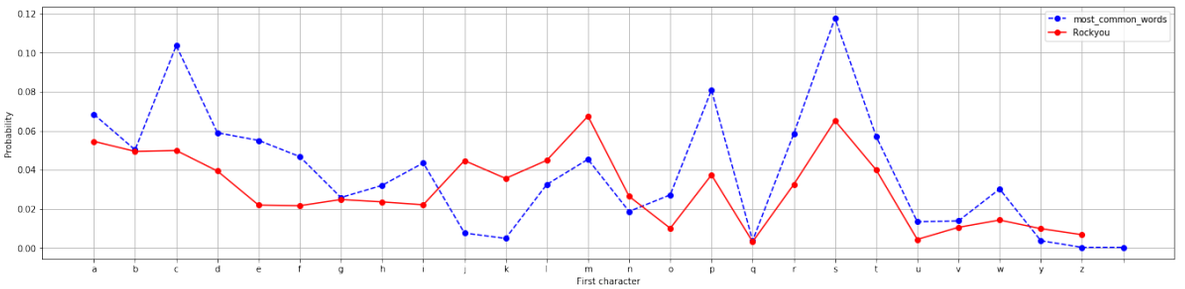

Figure 4.6: First Character in Rockyou and *most_common_words*

As with the *most_common_words*, we have compared the first character of the passwords in Rockyou and the first character of the dataset of personal names. Also in this case the trend of the two lines is almost faithful (see Fig. 4.7).

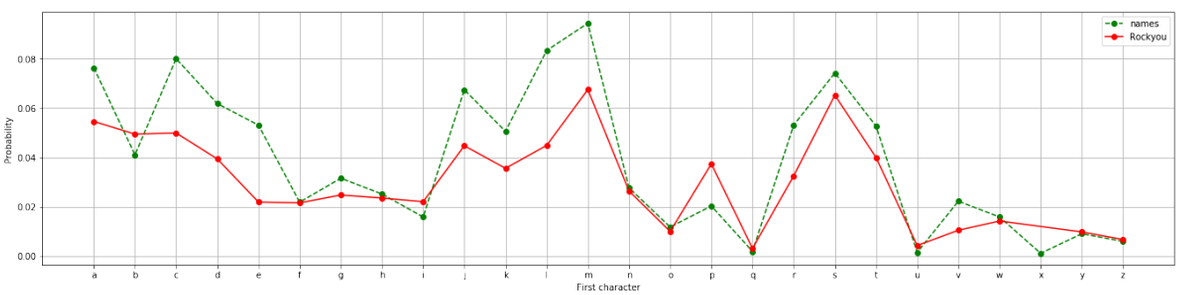

Figure 4.7: Names in Rockyou and most_common_words

Further analyzes were carried out on the dataset. In the next sections they will be described.

4.1.2 Hotmail

The second data leak analyzed was that of Hotmail [24]. Windows Live Hotmail (formerly known as MSN Hotmail) was a web-based email service developed by

Microsoft. Now its place has been taken by Outlook.com, even if it is still possible to access the mailboxes with the Hotmail domain.
In 2019 it suffered a data breach which has involed 773 million emails, and tens of millions of passwords, from a variety of domains. The dataset [25] contains 8930 of the stolen passwords. In average each password's length is 8 but the there is a lot of passwords whose length is 6 (in Figure 4.8 we report our analysis).

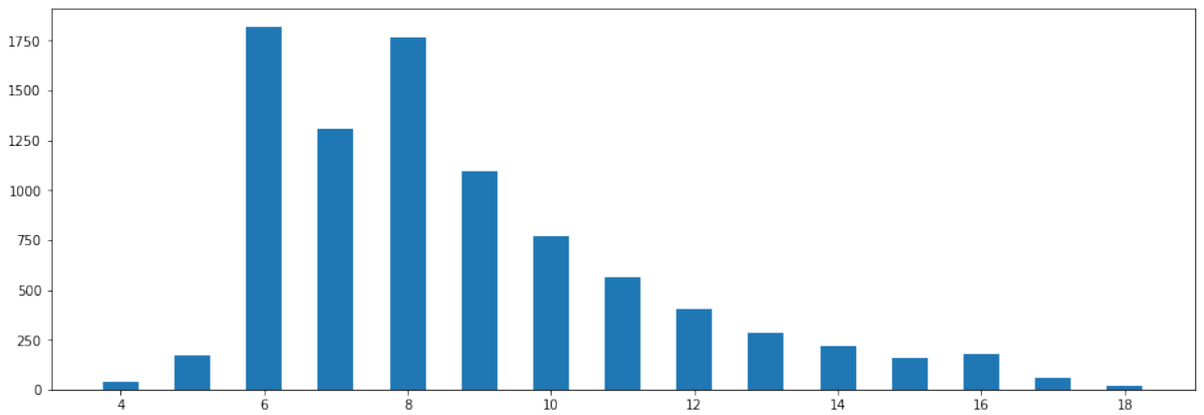

Figure 4.8: Mean password length Hotmail

As with RockYou we wanted to investigate the similarity between the first character of the *most_common_words* dataset and the Hotmail dataset (see Figure 4.9 and 4.10).

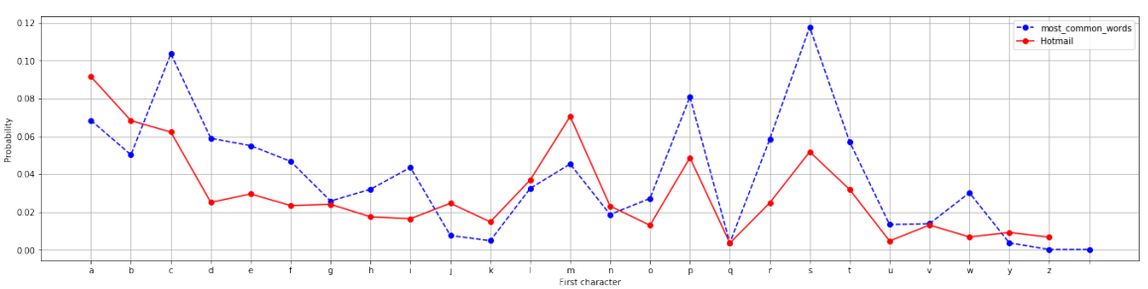

Figure 4.9: First Character in Hotmail and most_common_words

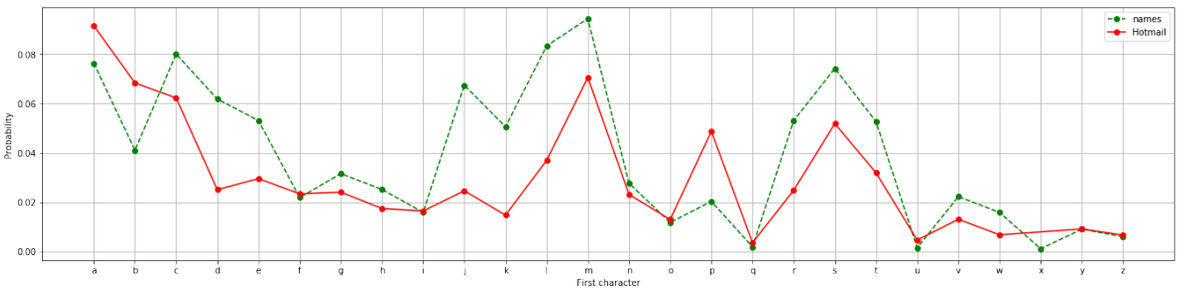

Figure 4.10: Names in Hotmail and most_common_words

We then subjected the data leak to the same algorithm that calculates the similarity of the first letter between it and the *most_common_words* dictionary. Once again the trends of the two lines are very similar. In particular, the letters s, p, m and a are those with the highest degree of similarity. So we can deduce that there is a strong similarity between the two dictionaries (both for the most common words and for the names).

4.1.3 Phpbb

The third data leak is PhpBB [43]. PhpBB is one of the biggest popular free forum management systems written using the PHP programming language: the name is an abbreviation of PHP Bulletin Board.
In 2009 it suffered a data breach which has involed 400,000+ accounts. The dataset [44] contains 184388 of the stolen passwords. In average each password's length is 8 (see the Figure we created by the analysis 4.15).

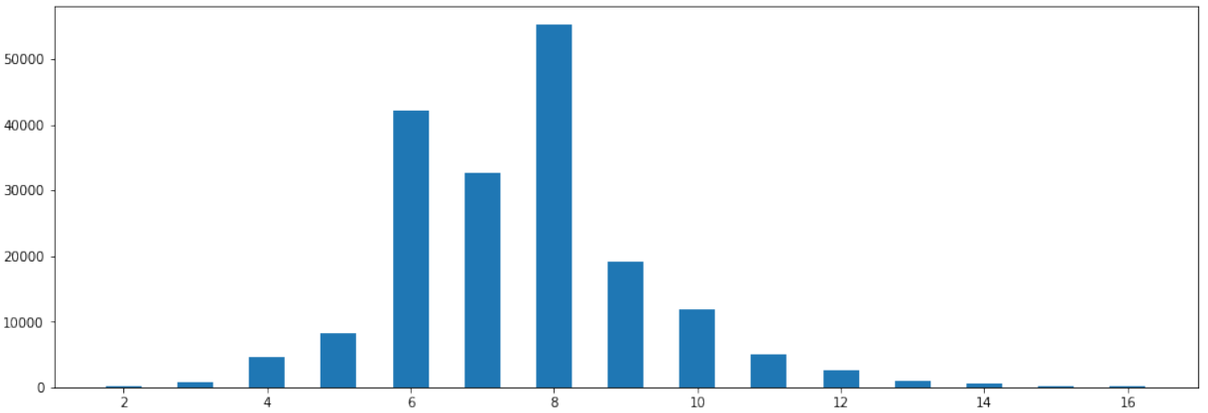

Figure 4.11: Mean password length PhpBB

As with RockYou and Hotmail we wanted to investigate the similarity between the first character of the *most_common_words* dataset and the PhpBB dataset (see the Figures that we created 4.12 and 4.13).

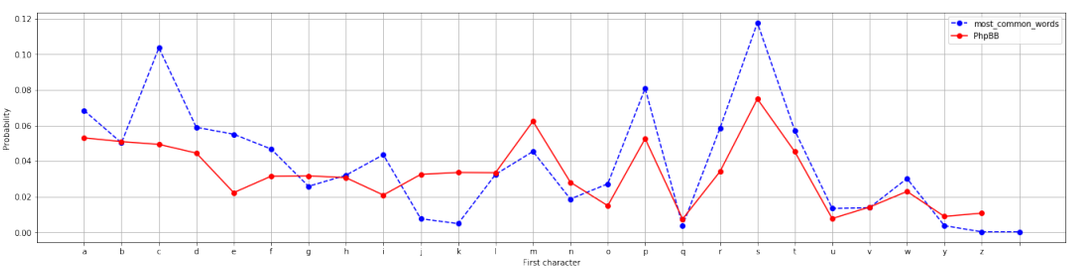

Figure 4.12: First Character in PhpBB and most_common_words

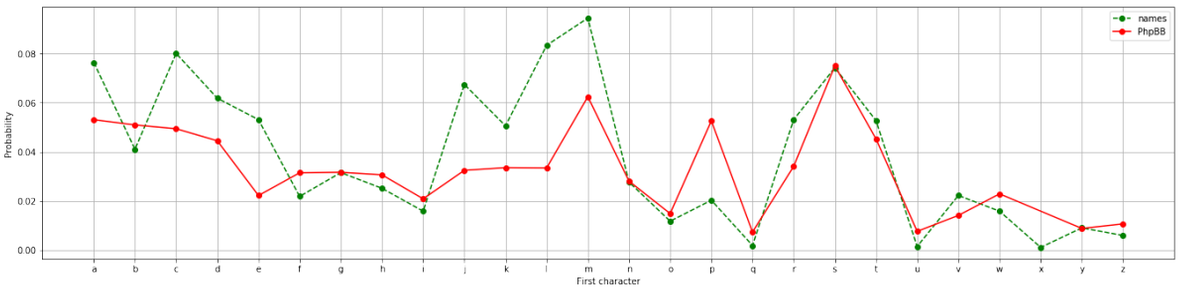

Figure 4.13: Names in PhpBB and most_common_words

From the previous figures it is possible to see how the trend of the lines between the first character in the PhpBB password dataset almost faithfully follows the trend of the first letter in the two dictionaries. As with previous data leaks, PhpBB also has strong similarities with the two dictionaries.

4.1.4 Ashley Madison

The fourth and last is Ashley Madison [4]. Ashley Madison is a Canadian online dating service and social networking service marketed to people who are married or in relationships.
In July 2015, a group calling itself *"The Impact Team"* stole the user data of Ashley Madison. The group copied personal information about the site's user base and threatened to release users' names and personally identifying information if Ashley Madison would not immediately shut down. The dataset [5] contains 375831 of the stolen passwords. In average each password's length is 8 (see Figure we made 4.14).

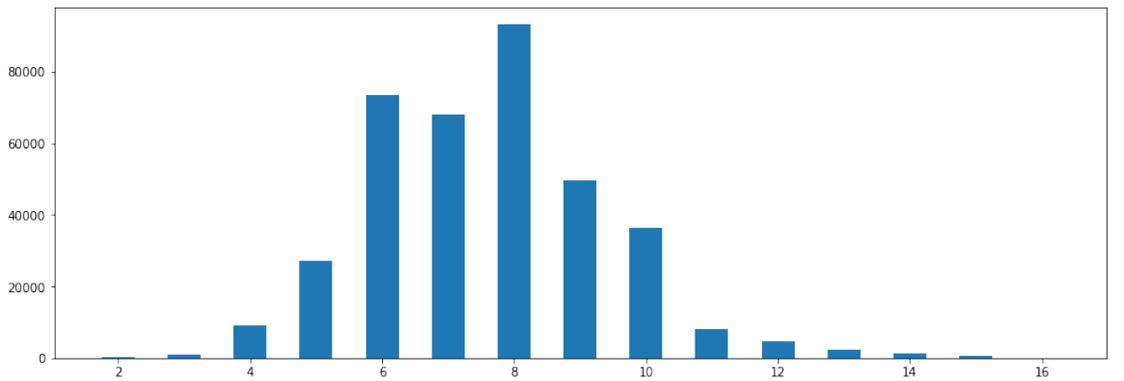

Figure 4.14: Mean password length Ashley Madison

Also for this dataset we wanted to investigate the similarity between the first character of the *most_common_words* dataset and the Ashley Madison dataset (see Fig: 4.12 and 4.13).

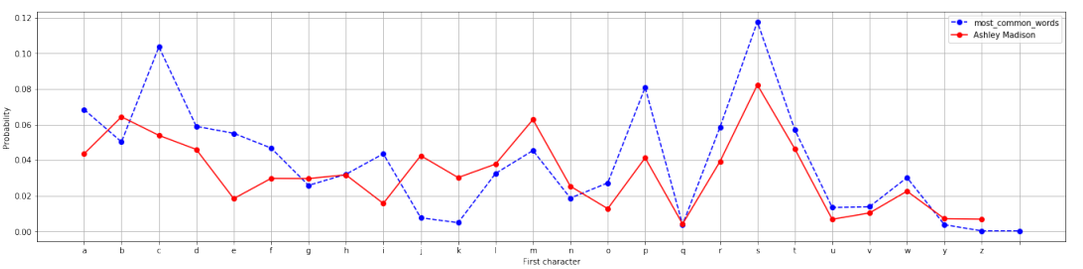

Figure 4.15: First Character in Ashley Madison and most_common_words

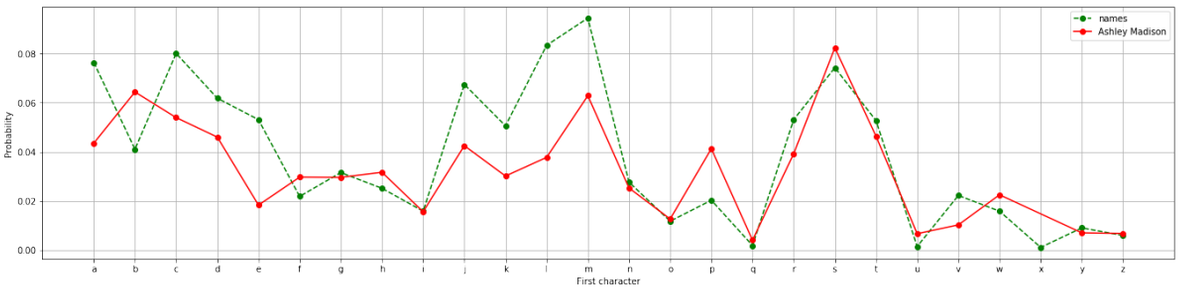

Figure 4.16: Names in Ashley Madison and most_common_words

Finally we analyzed Ashley Madison who, as for the three previous data leaks, almost faithfully follows the trend of the two dictionaries. In fact in the previous figures it is possible to see how the curves fit. As for PhpBB and Rockyou, in the case of the comparison between passwords and *most_common_words*, the letters j and k are more present in the data leak than in the dictionary.

4.2 Levenshtein distance for frequent replacements

The research also focused on finding typical password substitutions using an edit distance algorithm for strings. First we define the edit distance algorithm used.

Levenshtein distance is a measure of the similarity between two strings. The distance is the number of deletions, insertions, or substitutions required to transform the first string into the second one.

Mathematically, given two strings of length N and M, $D(N,M)$ is the distance. Accounting for the weights, edit distance can be computed this way:

$$lev(a,b) = \begin{cases} |a|, & \text{if } |b| = 0 \\ |b|, & \text{if} |a| = 0 \\ lev(tail(a), tail(b)), & \text{if} a[0] = b[0] \\ 1 + min \begin{cases} lev(tail(a), b) \\ lev(a, tail(b)) \\ lev(tail(a), tail(b)) \end{cases} & \text{otherwise} \end{cases} \tag{4.1}$$

Where $|a|$ and $|b|$ is the length of the string a and b respectively and the tail of some string x is a string of all but the first character of x, and $x[n]$ is the nth character of the string x, starting with character 0.
Below is an example for each case.

Example 4.2.1. lev("hello","")
Since $|b| = 0$ because the second string is empty the distance between the two strings is the length of the first one, so 5.

Example 4.2.2. lev("","hello")
Since $|a| = 0$ because the first string is empty the distance between the two strings is the length of the second one, so 5.

Example 4.2.3. lev("hello","hat")
Since *if a[0] = b[0]* becasue the two strings start with the same character we can apply the lev to the rest of the strings, which are the tails: lev("ello", "at"). With these two strings we enter the fourth case that we explain in the following example.

Example 4.2.4. lev("ello","at")
Since the two strings are not empty and the first character of the two is different, we apply the fourth case. We therefore look for the minimum distance between the two strings.

$$lev(\text{"}ello\text{"},\text{"}at\text{"}) = \begin{cases} 1 + min \begin{cases} lev(tail(\text{"}ello\text{"}),\text{"}at\text{"}) \\ lev(\text{"}ello\text{"},tail(\text{"}at\text{"})) \\ lev(tail(\text{"}ello\text{"}),tail(\text{"}at\text{"})) \end{cases} & \text{otherwise} \end{cases} \quad (4.2)$$

The first equation will return the distance between "llo" and "at" which is 3.
The second equation will return the distance between "ello" and "t" which is 4.
The third equation will return the distance between "llo" and "t" which is 3.
The final result is 1 + min (3,4,3) = 4

There are many algorithms to compute the edit distance. Many of them are classified as dynamic programming algorithms. One of them is the *Wagner-Fischer* [97]. Computing the Levenshtein distance is based on the observation that if we reserve a matrix to hold the Levenshtein distances between all prefixes of the first string and all prefixes of the second, then we can compute the values in the matrix in a dynamic programming fashion, and thus find the distance between the two full strings as the last value computed.
The straightforward pseudocode implementation for the distance is the following one:

Algorithm 2 LevenshteinDistance [91]

```
Data: s, t, n, m
Result: d[m, n]
initialization
declare int d[0..m, 0..n]
set each element in d to zero
for i = 1, 2, ..., m do
    d[i, 0] := i
end
for j = 1, 2, ..., n do
    d[0, j] := j
end
for j = 1, 2, ..., n do
    for i = 1, 2, ..., m do
        if s[i] = t[j] then
            substitutionCost := 0
        end
        else
            substitutionCost := 1
        end
        d[i, j] := minimum(d[i-1, j] + 1, d[i, j-1] + 1, d[i-1, j-1] + substitutionCost)
    end
end
return d[m, n]
```

Algorithm 2 takes as input two strings, s of length m, and t of length n, and returns the Levenshtein distance between them.

The algorithm works as follows: first a matrix d of size m*n is declared. Once the first column is fixed, in the first cycle all the rows are scrolled and the values from 1 to m respectively entered. Once the first row is fixed, in the second cycle all the columns are scrolled and the values from 1 to n respectively entered. The third and fourth cycles apply what is described in the equation 4.1.

Consider the two words "rain" and "shine" (see Figure 4.17). To transform "rain" into "shine", we can replace 'r' with 's', replace 'a' with 'h' and insert 'e'.

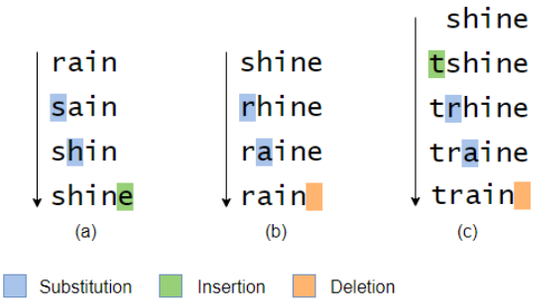

Figure 4.17: Levenshtein Distance example [29]

The resultant matrix is the shown in the Table 4.1.

		r	a	i	n
		1	2	3	4
s	1	1	2	3	4
h	2	2	2	3	4
i	3	3	3	2	3
n	4	4	4	3	2
e	5	5	5	4	3

Table 4.1: Matrix exit distance

Thus, the edit distance between these two words is 3. This is assuming that all operations have the same cost of 1. If we assign a higher cost to substitutions, for example 2, then the edit distance becomes 2*2 + 1 = 5. To transform "shine" to "rain", the operations are reversed (insertions become deletions) but the edit distance is the same when costs are symmetric.

4.2.1 Frequent substitutions analysis

As we already know a strong password requires characters from different character sets. In addition, the password length is also a metric used to determine its strength. Adding a number and/or special character to a password might thwart some simple dictionary attacks. For example, the password "kitten" could be

munged in the following ways: k1tten, k1tt3n, k!tten, k!tt3n. This type of substitution is called **Password Munging**.

Definition 30. *[32]* ***Password munging*** *is the art of changing a word that is easy to remember until it becomes a strong password.*

This is how most people make up passwords to attempt to create a strong, secure password through character substitution. "*Munge*" is sometimes backronymmed as *Modify Until Not Guessed Easily*. The usage differs significantly from Mung, because munging implies destruction of data, while mungeing implies creation of strong protection for data.

The substitutions can help users to remember better their passwords, and may increase an attacker's difficulties. Common substitution are listed in Table 4.2.

Character	**Symbol**
a	@
b	8
c	(
d	6
e	3
f	#
g	9
h	#
i	1
i	!
k	¡
l	1
l	i
l	;
o	0
q	9
s	5
s	$
t	+
v	>
v	<
x	%
y	?
w	uu
w	2u

Table 4.2: Common substitution

Other ways to make substitutions have also developed such as *Faux Cyrillic* (which are also used to create fake URLs) [17] and *Leets* [28].
Initially using these techniques was efficient to mitigate dictionary attacks but, for now, this is the wrong way to do it since even the attackers are aware of these simple changes.
Using the *LevenshteinDistance* we would like to find these types of common substitutions to check how they are used by users. Given a dataset of English common words, *LevenshteinDistance* was applied on each plaintext of each dataset and the strings of the common word dataset to check the modification of the plain-

texts. The naive implementation of the algorithm which we have developed is the following one:

Algorithm 3 FindCommonSubstitutionNaive

Data: Plain, DictCommonWords
Result: D_{res}
initialization
declare $D_{res}[0..len(Plain)]$
for $j = 1, 2, \ldots, len(Plain)$ **do**
  for $i = 1, 2, \ldots, len(DictCommonWords)$ **do**
    if *LevenshteinDistance(Plain[i], DictCommonWords[j])* ≤ 2 **then**
      $D_{res} := Plain[i]$
      end
    end
  end
  return D_{res}

Since iterating through each plaintext and each string carries a cost of $\mathcal{O}(NM)$ where M and N are the lengths of the two strings and the cost of the Levenshtein distance is $\mathcal{O}(NM)$ the algorithm implemented in a naive way has the following cost: $\mathcal{O}(NM\mathcal{O}(NM))$ where N is the length of the first string and M is the length of the second one. Therefore this implementation is too expensive to be able to apply to even larger datasets than those we have analyzed so it was necessary to make improvements. The second version of our algorithm that we have developed is shown below.

Algorithm 4 FindCommonSubstitutionImproved

Data: Plain, DictCommonWords, treshold
Result: D_{res}
initialization
declare $D_{res}[0..len(Plain)]$
for $j = 1, 2, \ldots, len(Plain)$ **do**
  for $i = 1, 2, \ldots, len(DictCommonWords)$ **do**
    if *(thereIsSymbolInside(Plain) and len(Plain[i] = len(DictCommonWords[j]))* **then**
      if *LevenshteinDistance(Plain[i], DictCommonWords[j]) = treshold* **then**
        $D_{res} := Plain[i]$
        end
      end
    end
  end
  return D_{res}

The new algorithm has some filtering. First of all, we check that the plaintext has symbols inside and that it has the same length as the dictionary word visited. If so then plaintext is added to the resulting word list only if the Levenshtein Distance is greater than or equal to a threshold passed in input. This threshold corresponds to the level of similarity that we expect between the two strings. In fact, if the threshold were set to 1 it would mean that we take into consideration words that have at most one different character from each other.
The function *thereIsSymbolInside* takes a plaintext as input and returns TRUE if there is a symbol inside it.
The second version, although better than the first, still processed too much data since filtering took place inside the for loops. The final version we have adopted in our studies is the third (which is shown below). Since we decided to analyze passwords in the range of 5 to 12 characters, we created seven different files containing passwords of each length. We did the same for the dictionary of common words. In this way the processed data has been drastically reduced. So as input the algorithm takes PlainFiltered, the file containing passwords of arbitrary length between 5 and 12, DictCommonWordsFiltered the file containing the most common words of arbitrary length between 5 and 12, and the minimum threshold of

difference that strings can have between their.

Algorithm 5 FilterFileByLength

```
Data: Plain, length
Result: PlainFiltered
initialization
declare PlainFiltered
for j = 1, 2, ..., len(Plain) do
    if len(Plain[j]) = length and hasSymbols(Plain[j]) then
        PlainFiltered := Plain[j]
    end
end
return PlainFiltered
```

Algorithm 5, that we developed to filter, takes as input a file and the desired length and returns a new file with the same strings contained in the input file but with a length equal to the one passed in input. Furthermore, we only insert words that have symbols (*hasSymbols(string)*) in them so as not to make the Levenshtein algorithm process useless data.

Algorithm 6 FindCommonSubstitutionImprovedV2

```
Data: PlainFiltered, DictCommonWordsFiltered, treshold
Result: D_res
initialization
declare D_res[0..len(PlainFiltered)]
for j = 1, 2, ..., len(PlainFiltered) do
    for i = 1, 2, ..., len(DictCommonWordsFiltered) do
        if LevenshteinDistance(PlainFiltered[i], DictCommonWordsFiltered[j]) =
        treshold then
            D_res := PlainFiltered[i]
        end
    end
    return D_res
```

In terms of complexity, this algorithm does not differ from the first but radically changes the weight of the two viato files that are initially massively filtered. Sup-

pose you have 14 million passwords and a dictionary of common words of 100,000 words. Filtering strings 12 characters long, since they are the least present, from 14 million we could reach 1 million and from 100 thousand words to 2 thousand words. Therefore the comparison would no longer take place between datasets 14 million * 100 thousand but between 1 million * 2 thousand, a net reduction of data to be calculated.
We have subjected all four datasets to this algorithm and we found more than 600 different types of replacements. In Figure 4.18 some of them have been inserted.

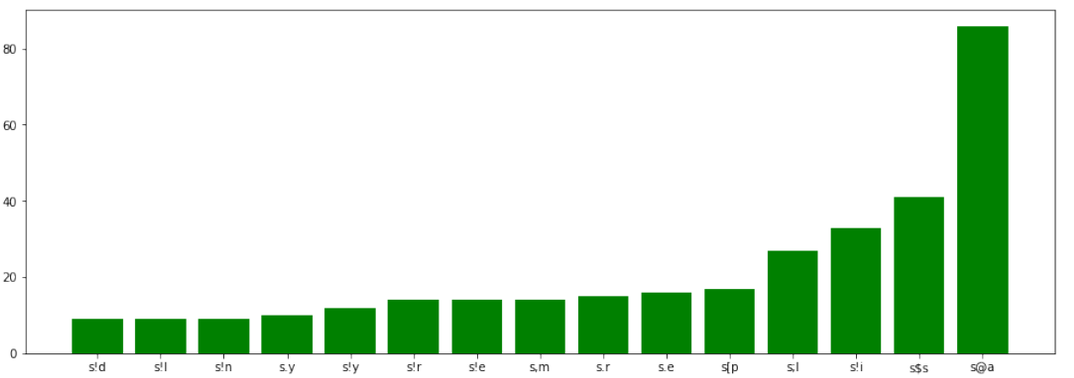

Figure 4.18: Most common replacements

The figure shows the main replecements we found in our analysis. On the far right we find "s @ a" which stands for "@" replaced with "a". In the x axis we have the substitutions, in the y axis the percentage of presence. On the right side of the graph you can see which are the most used replacements. While in the leftmost part there are substitutions also not listed in Table 4.2. This result was surprising and we wanted to do an even more in-depth analysis by checking what were the plaintext that generated these replacements.
By filtering the plaintext that had such substitutions we noticed that most of the ones on the left of the graph are substitutions made at the end of words. Users, therefore, make a sort of *stemmatization* by truncating the last letter and replacing it with specific symbols. [51] ***Stemming*** is a method of normalization of words in Natural Language Processing. In stemming a set of words in a sentence are

converted into a sequence to shorten its lookup. In this method, the words having the same meaning but have some variations according to the context or sentence are normalized. For example love{. → r} (the word *lover* becomes *love.*), blood{+ → y}, welcom{. → e} and so on.
All of this confirms the use of the most common password substitutions Knowing what the typical replacements are is especially useful for when we want to compare the passwords of the data leaks with the words contained in the captions for greater accuracy since a password like "P@55w0rd" and a "password" string differ a lot and risk seem two completely different words while it would be enough to apply some rules that replace the 5 with s, the 0 with o, the @ with a and so on to discover that the two strings are exactly the same thing. This is what attackers do, so replacing letters with numbers or symbols is not as safe as some users believe.

4.3 Passwords Entropy and Password Quality

At this point, as we already seen, the strength of password authentication relies on the strength of the passwords. Password strength is measured by calculating the entropy. Since password entropy is mentioned as a quality indicator for passwords in many occasions. Measuring the quality of password becomes an interesting topic. In the literature it is often mentioned that whoever creates a password must create it with a high entropy. But the concept of entropy applied to a password loses its meaning. The concept of information entropy that we discussed in detail in Chapter 3 has been most widely used in several technological areas. The fundamental reason why entropy cannot be used as a quality indicator for passwords is that the calculation of the entropy is based on a statistic distribution model of a language and is conducted on a model of the n-order Markov process. The main problems are that password guessing is not a Markov process since guessing a password is an all-or- nothing game.
During our analysis, we wondered how strong were the passwords that were inside the various data leaks. We started analyzing them and found that many of the passwords containing dictionary words were considered strong by the tools that calculate password entropy. At which we wondered if it was actually the right

method or if something was missing from the basic calculation.
We thought it would be useful, therefore, to filter passwords that were too similar or equal to words already present in dictionaries. So, first of all, we filtered all passwords that had a Levenshtein's distance less than three. Before comparing the passwords, we eliminated numbers and symbols. The reason we chose 3 as the minimum distance is the average word length. Since words are 6 characters long on average, having a 3+ character difference between passwords and common words means that, at most, only slightly more than half of the password would be equal to a common word. This means that having the password "Love123" the numbers are eliminated becoming "Love", subjected to the Levenshtein's distance with the words of the dictionary of common words. Since "Love" is a very common word, the result would have been *0* therefore the password considered to be of very low quality and therefore easily guessable. If the password was considered to be of good quality then the classic password entropy calculation would have occurred, if the password has a high entropy then it is good otherwise to be changed. A pseudocode of our naive password classifier is as follows:

Algorithm 7 passwordFilter

Data: Password, DictCommonWords
Result: goodPassword
initialization
goodPassword = False
for $i = 1, 2, \ldots, len(DictCommonWords)$ **do**
    if *(LevenshteinDistance(removeNumbersAndSymbols(Password),DictCommonWords[i])* $\geq$ *3* **then**
        if *(entropy(Password))* **then**
            goodPassword = True
        end
    end
end
return goodPassword

The function *removeNumbersAndSymbols*, on the basis of the analysis of the most common replaces did before, replaces and eliminates numbers and symbols in order to make the password as similar as possible to a "clean" string, therefore composed

only of characters. This function checks if the number or symbol are inside the password, replaces them with the corresponding letter, if it exists otherwise it deletes them, while if they are at the end or at the beginning it deletes them. The algorithm is a naive version that will be improved in the future by making replacements more sophisticated. Function *entropy* returns True if the entropy of the password is high enough according to the canons described in chapter 4, otherwise returns False. To test our algorithm we took a subset of RockYou. We subjected the subset to the entropy check alone and 8 out of 10 passwords were considered to have very high entropy while our algorithm 1 in 10 (an example is shown in Figures 4.19 and 4.20).

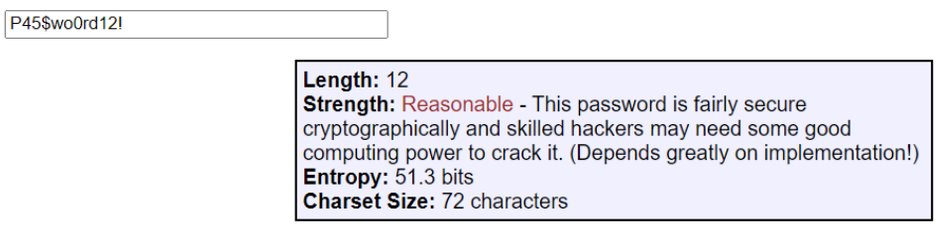

Figure 4.19: Original password subjected to the algorithm that calculates entropy.

We submitted the password in the figure to our filter and the result was the following:

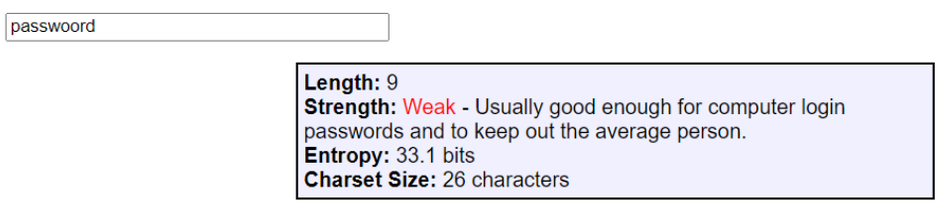

Figure 4.20: Password filtered subjected to our algorithm

From the previous figures, we have replaced the 4 with the letter a, the 5 and the dollar with the s, the 0 with the o and eliminated the two numbers and the symbol at the end. We also made the initial p lowercase. It is possible to see how a reasonably strong password has become weak after applying our filtering algorithm. So it is clear that most common replaces are a weak attempt to make a weak starting password stronger.
A similar method was adopted by the authors of the paper [122] which we will describe below.

To the authors opinion it is meaningless to calculate password entropy based on the composing characters, there is no statistic distribution for passwords and it only establishes the low boundary for how many guesses needed to crack passwords but not the quality of the password. The authors of the paper advocate the different means of measuring password quality (PQI).

Definition 31. *PQI: The PQI of a password is a pair λ = (D, L) , where D is the Levenshtein's editing distance of the password to the base dictionary words, and L is the effective password length. The effective password length is the equivalent length of the password in the standard password format, which consists of only the 10 digit characters (0-9).*

There are many different types of password attacks. In general, a likely path to crack a password is, in the order of trying dictionary words, trying 1 (and perhaps 2) character variations to the dictionary words, trying to enumerate all possible spellings of a smaller character set, trying to enumerate all possible spellings of a smaller character set.
The quality of a password depends on how long it takes to find out the right match. The longer it takes, the better the quality is. Thus, a good way to measure the quality of a password is to calculate *how different it is from the dictionary words*, *how long it is*, and *how big the password character set is*. In order to measure the distance between a string-password and the words contained in a dictionary of words it is useful to use Levenshtein's editing distance which can accurately measure how different two strings are (see the Section 5.2 for more details).
The authors of the paper develop a concise rule for choosing a good password. *It*

should be at least 8 characters long, with at least 3 special characters plus other alphanumeric characters.
The *password quality indicator*, when D≥3 and L≥14, indicates that we have a good password. Having D≥3 means that the password is at least 3 characters different from the base dictionary words, and L≥14 means that there are at least 10^{14} possible password candidates to be tried to crack the password.
So we can say that only the calculation of the entropy of a password is not an excellent measurement metric, it is necessary to resort to additional methods.

Compared to the authors of the paper we have added the filtering of the most common replaces. In fact, they first calculate the Levenshtein distance of the password with a dictionary of words and then apply the entropy formula. The problem arises with passwords like the one shown in Figure 4.19. Levenshtein distance between "Pa45$o0rd123!" (we added two characters to get to a length of 14 but that do not significantly change the computation of algorithms) and "password" is 9. Having a length of 14 characters and a Levenshtein distance of 9 characters with the word "password" belonging to the dictionary of words, according to the PQI of the authors of the paper the password "Pa45$ o0rd123!" it is of good quality. While we have previously demonstrated the fact that it is weak therefore of not good quality.

4.4 Pattern frequency analysis

To access the systems, at least one password that follows certain patterns is required. Often such patterns involve entering passwords that are difficult to remember. A password fulfilling these complexity requirements would provide high entropy and therefore should be more resistant against password guessing attacks. On the other hand, it is questionable if a password fulfilling the complexity rules including minimum length can be considered as a strong password as discussed in Section 4.3. For example taking this password "P45sw0rd1." into consideration. It has the length of ten characters and contains five lowercase letters, one uppercase letter, four digits and one special symbol so it is considered secure and accepted in general as a strong password according to many password policies of

enterprise companies and organizations. But by using a pattern-based attack this is an insecure password and can be easily guessed. The password from the example contains three different common patterns. The first one is capitalization of the first letter. The second one is replacing certain letters with numbers (a $\rightarrow$ 4, o$\rightarrow$0, s$\rightarrow$5) and the third pattern is appending "1." to the password. The problem is that if many passwords share the same patterns, they can be identified and then misused to guess passwords successfully with the help of automated tools.

We introduce basic elements and types of users' passwords in Table 4.3.

Type	#	**Basic Elements**
Numeric (N)	10	0123456789
Lowercase (L)	26	abcdefghijklmnopqrstuvwsyz
Uppercase (U)	26	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Other (O)	32	'!@#$% &*() +=[]\{}|;':,./<>?

Table 4.3: Types of users' passwords

Since users create their passwords as patterns based on a combination among the strings of character types, we define the password patterns and the pattern class as: The pattern of a password is the combination of character strings of the type N, L, U and O. It is therefore represented as a combination of strings of the type N_n, L_n, O_n, U_n where the subscript corresponds to the length of the type to which it corresponds. The pattern class is represented as the password patterns and p a combination of strings made up of N^+, L^+ and so on.

Considering, therefore, this definition we can say that, for example, the password *Pa5sw0rd1* corresponds to the pattern U_1, L_1, N_1, L_2, N_1, L_2, N_1 since it contains an uppercase letter, a lowercase letter, a number, two lowercase letters, a number, two lowercase letters and finally a number.
In Table 4.4 there is an example of the 10 most used passwords in the world [31].

Password Example	**Pattern Class**	**Password Pattern**
123123	N^+	N_6
abc123	L^+N^+	L_3N_3
password1	L^+N^+	L_8N_1
iloveyou	L^+	L_8
letmein	L^+	L_7
27653	N^+	N_5
qwerty123	L^+N^+	L_6N_3
1qaz2wsx	$N^+L^+N^+L^+$	$N_1L_3N_1L_2$
sunshine	L^+	L_8
1q2w3e4r	$N^+L^+N^+L^+N^+L^+N^+L^+$	$N_1L_1N_1L_1N_1L_1N_1L_1$

Table 4.4: Password class and password patterns on top 10 most famous passwords

We have analyzed each of the four datasets that we use as a reference to find the most used patterns. In Table 4.5 we report the top ten most frequent patterns we found in RockYou.

Pattern Class	#	%
L^+N^+	4720184	32.91%
L^+	3726129	25.98%
N^+	2346744	16.36%
N^+L^+	499167	3.48%
$L^+N^+L^+$	388157	2.71%
U^+N^+	325941	2.27%
$U^+L^+N^+$	236331	1.65%
U^+	229875	1.6%
$L^+O^+L^+$	172279	1.2%
$L^+O^+N^+$	144129	1.0%

Table 4.5: Top ten pattern classes from RockYou

The first three patterns that we have reported in the table we find them for 75.25 % of the passwords contained in RockYou, this indicates that most users have, for the most part, entered passwords consisting of only numbers, only lowercase characters or a mix of both. In fact, as initially described, RockYou did not require a particular pattern so there were no restrictions and controls. Since the most popular pattern class of all is L^+N^+ we have decided to analyze in detail the corresponding password patterns.

Pattern Class	#	%
L^+N^+	140016	37.25%
L^+	124530	33.13%
N^+	46298	12.31%
N^+L^++	16199	4.31%
U^+	12872	3.43%
$L^+N^+L^+$	7001	1.86%
U^+N^+	4336	1.15%
$L^+O^+N^+$	3682	0.97%
$U^+L^+N^+$	3426	0.92%
$N^+L^+N^+$	2942	0.78%

Table 4.7: Top ten pattern classes from Ashley Madison

But first of all we report the pattern classes of the other three datasets to analyze the four datasets together and see if the common patterns are specific to RockYou or if there are confirmations from the other datasets as well.

Pattern Class	#	%
L^+	3716	41.61%
L^+N^+	1730	19.37%
N^+	1654	18.52%
N^+L^+	279	3.12%
U^+	197	2.20%
$L^+N^+L^+$	127	1.42%
U^+N^+	112	1.25%
$L^+O^+N^+$	89	0.99%
$U^+L^+N^+$	72	0.80%
$N^+L^+N^+$	64	0.72%

Table 4.6: Top ten pattern classes from Hotmail

From the Tables 4.6, 4.7, 4.8 it is clear that the most common patterns correspond between the various datasets. This means that although we had several data leaks available, various users on average used the same patterns.

Pattern Class	#	%
L^+	76069	41.25%
L^+N^+	43425	23.55%
N^+	20730	11.24%
N^+L^+	8513	4.61%
U^+	4704	2.55%
$L^+N^+L^+$	2697	1.46%
U^+N^+	2533	1.37%
$L^+O^+N^+$	2455	1.33%
$U^+L^+N^+$	1883	1.02%
$N^+L^+N^+$	1752	0.95%

Table 4.8: Top ten pattern classes from PhpBB

Pattern Class	#	%
L_6N_2	420,318	8.91%
L_5N_2	292,306	6.19%
L_7N_2	273,624	5.80%
L_4N_4	235,360	4.99%
L_4N_2	215,074	4.56%
L_8N_2	213,109	4.51%
L_6N_1	193,097	4.10%
L_7N_1	189,847	4.02%
L_5N_4	173,559	3.68%
L_6N_4	160,592	3.40%

Table 4.9: Top ten password patterns from password class L^+N^+ RcokYou

Proceeding with the analyzes, we will focus on RockYou analytics since the others follow the same patterns as well and because it is the largest dataset respect to the others. In Table 4.9 we have listed the top ten password patterns of the most common class. The first pattern is L_6N_2 so for most of the passwords we find strings long six lowercase characters and at the end two characters numbers. Furthermore, by adding the length of each most used pattern, we can deduce the average of the passwords which is 8 confirming what has been analyzed in the appropriate section of RockYou where, after an analysis on the average length of the strings contained, it is derived that on average it has password length 8.
A further analysis was to calculate the occurrences and percentages of strings with

N^+ pattern. What emerged was that N_2 is the most common (24.38%). Next we find N_4 and N_1. Discovered this we went even deeper looking for what were the two most common numbers that make up N_2. In table 4.10 we have reported the top five of the pairs that form N_2.

N_2 **elements**	#	%
12	102,590	4.81%
13	76,775	3.60%
11	65,201	3.06%
22	58,058	2.72%
23	57,825	2.71%

Table 4.10: Top 5 of pattern N_2 RockYou

The most present patterns see the number one, two and three as protagonists. As we will describe in section it is not uncommon for the most present numbers to be the first three.

The last analysis concerns the "Others" that is the symbols, which we will resume in the Section . We calculated the occurrences from one symbol up to five. In the table 4.11 we show the top five.

Symbols	#	%
O_1	238,652	79.26%
O_2	38,780	12.88%
O_3	16,184	5.37%
O_4	3,544	1.18%
O_5	1,260	0.42%

Table 4.11: Top 5 of pattern O^+ RockYou

In Figure 4.27 we report the most frequent symbols for the reference datasets.

4.4.1 Summary

We can conclude this section with a summary of the most common patterns we have found from our statistics. First of all, we identified several patterns which

belong mainly to ten categories (according to the paper [118]): Prefixing, Appending, Inserting, Repeating, Sequencing, Replacing, Capitalizing, Reversing, Special-format and Mixed Patterns.
Many passwords are composed of the pattern class L^+N^+, which means that most of the passwords contained in the dataset are composed of a string of lowercase characters of arbitrary length and subsequently a string of arbitrary length composed of numbers. After that, numbers and symbols are mostly inserted at the end or at the beginning (*appending*, *prefixing pattern*) of a string of characters, for example in the case of L_4N_2 we have $L_4 \rightarrow love$ and $N_2 \rightarrow 12$ which together make up "love12". In addition to appending and prefixing patterns, we identified many password examples of *inserting pattern* by which a certain digit and/or punctuation character (or digit/character groups) is inserted into a dictionary word. For example: *abc123def*, *my3love*, *love4ever*.
Then, there are many passwords that have a common pattern repeated multiple times (see Table 4.5). These form the repeating patterns, for example: *kisskiss*, *121212*, *11111*. The *replacing pattern* which consist in replacing certain letters with a number or symbol was already analyzed in section .
Regarding the reversing pattern, we looked for how many words in the dataset were in a reverse order. As an example, the word "password" is converted into "drowssap", "file" is converted with "elif" and so on. We have found that more than 1% of the passwords contained in RockYou follow this pattern. Tables 4.12, 4.13, 4.14, 4.15, 4.16 show some examples for five of the ten pattern categories which we deduced from the four data leaks.

Repeating Pattern Example	**Password Example**
Repeating number groups	123123, 11111, 333333, 22222, 121212
Repeating words	lovelovelove, byebye, catcat, kisskisskiss
Repeating birth years	19871987, 19891989, 19931993

Table 4.12: Repeating pattern examples

Sequencing Pattern Example	Password Example
Digit Sequences	12345, 1234, 123, 3456
Keyboard Sequencesmixed with DigitSequences	qwer, asdf1234, 1q2w3e4r
Alphabet LetterSequences	abcd, cdef, abcdefg

Table 4.13: Sequencing pattern examples

Appending Pattern Example	Pattern Example
Appending numbers [0-9]	password1, princess1, angel1
Appending !	iloveyou!, password!, rockyou!
Appending 123	test123, red123, qwe123

Table 4.14: Appending pattern examples

Prefixing Pattern Example	Pattern Example
Prefixing numbers [0-9]	1password, 1lover, 1love
Prefixing !	!password, !iloveyou, !red
Prefixing 123	123abc, 123asd, 123fgh

Table 4.15: Prefixing pattern examples

Replacing Pattern Example	Pattern Example
a replaced with 4	b4sketball, p4assword, dr4gon
a replaced with @	p@ssqord, t@ylor, di@mond
b replaced with 6	septem6er, remem6er, sponge6ob

Table 4.16: Replacing pattern examples

Having these statistics, it is possible to generate efficient passwords for reducing the number of guesses and increasing the hit rate. Instead of trying to create rules that mimic common password patterns, we can assign probabilities to these patterns, and then use those probabilities directly to generate fine-grained rules, and generate passwords for cracking. Further analyzes were conducted in the following sections.

4.5 Benford's law for number distribution

Benford's law, also known as the Law of First Digits or the Phenomenon of Significant Digits, is an observation about the frequency distribution of leading digits in many real-life sets of numerical data [7].

The discovery on the subject dates from 1881, in the work of an American-Canadian astronomer and mathematician. Simon Newcomb, while flipping through pages of a book of logarithmic tables, noticed that in logarithm tables the earlier pages (that started with 1) were much more worn than the other pages than the pages at the end. This meant that his colleagues, who shared the library, preferred quantities beginning with the number one in their various disciplines.
In 1938, the American physicist Frank Benford revisited the phenomenon, which he called the *"Law of Anomalous Numbers"* in a survey with more than 20,000 observations of empirical data compiled from various sources, ranging from areas of rivers to molecular weights of chemical compounds, cost data, address numbers, population sizes, and physical constants. All of them, to a greater or lesser extent, followed such an exponentially diminishing distribution.
In the end, Ted Hill, in 1995, proved the result about mixed distributions. His proof was based on the fact that numbers in data series following Benford's Law are, in effect, "second generation" distributions, i.e. combinations of other distributions.
The law states that in many naturally occurring collections of numbers, the leading digit is likely to be small. In sets that obey the law, the number 1 appears as the leading significant digit about 30% of the time, while 9 appears as the leading significant digit less than 5% of the time (see Figure 4.21). If the digits were distributed uniformly, they would each occur about 11.1% of the time. Benford's law also makes predictions about the distribution of second digits, third digits, digit combinations, and so on.

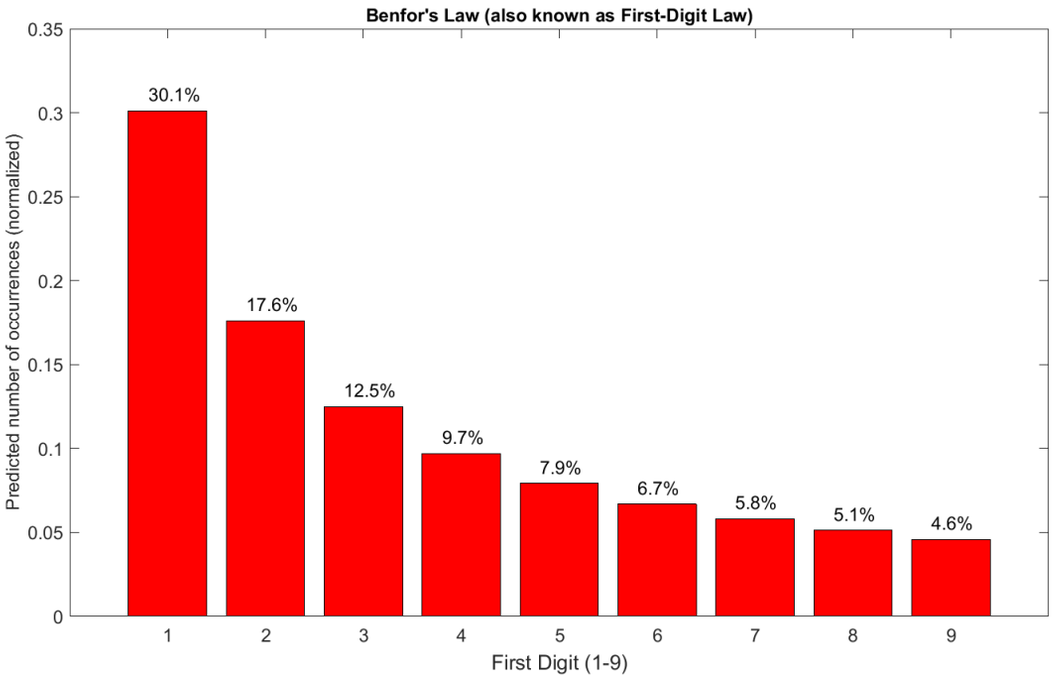

Figure 4.21: Benford's law plot [7]

Benford's law applies to data that are not dimensionless, so the numerical values of the data depend on the units. If there exists a universal probability distribution P(x) over such numbers, where x is the number and k an arbitrary number, then it must be invariant under a change of scale, so

$$P(kx) = f(k)P(x)$$

If $P(x)dx = 1$ then $P(kx)dx = 1/k$, normalization implies $f(k) = 1/k$. Settings $k = 1$ gives $xP' = -P(x)$ having solution $P(x) = 1/x$. So the probability of a digit d is:

$$P(d) = log_{10}(d+1) - log_{10}(d) = log_{10}(\frac{d+1}{d}) = log_{10}(1 + \frac{1}{d}))$$

Table 4.17 shows the distribution of the first 9 digits according to the Benford's

Law.

d	P(d)
1	0.301
2	0.176
3	0.125
4	0.097
5	0.079
6	0.067
7	0.058
8	0.051
9	0.046

Table 4.17: Benford's Law probabilities

However, Benford's law applies not only to scale-invariant data, but also to numbers chosen from a variety of different sources. Explaining this fact requires a more rigorous investigation of central limit-like theorems for the mantissas of random variables under multiplication. As the number of variables increases, the density function approaches that of the above logarithmic distribution. Hill rigorously demonstrated that the "distribution of distributions" given by random samples taken from a variety of different distributions is, in fact, Benford's law. In [6] are listed some examples which demostrate the power of this law.

Benford's law is widely used to **account fraud detection**. In fact Hal Varian suggested that the law could be used to detect possible fraud in lists of socio-economic data submitted in support of public planning decisions. Based on the plausible assumption that people who fabricate figures tend to distribute their digits fairly uniformly, a simple comparison of first-digit frequency distribution from the data with the expected distribution according to Benford's law ought to show up any anomalous results [57]. Accountancy data generally follows the four assumptions required for a valid conclusion on a Benford curve: general ledgers, income statements, and inventory listings can all be compared to the curve to determine genuineness.

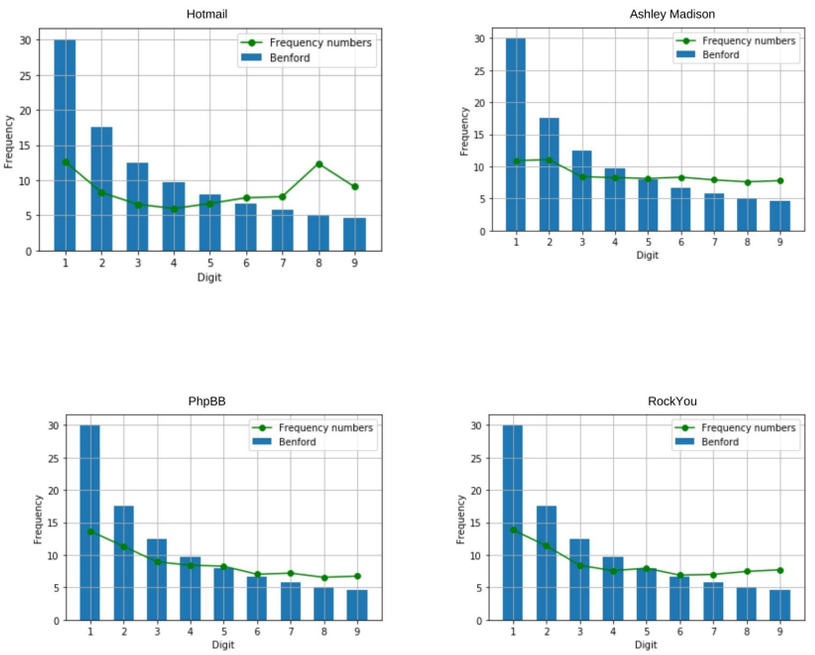

Figure 4.22: Benford's law plot on Hotmail, Ashley Madison, PhpBB and RockYou

So, we decided to analyze the password datasets we had available to see if even in this case Benford's law could be applied. What emerged was that the law actually applies to the datasets used. In Figure 4.22 it is possible to see the plot of the frequency of the digits in the passwords with respect to Benford's law. There is no actual fitting but the fact that Benford does not mention the number 0 should also be taken into consideration, a figure which is instead quite used mainly for substitution with the "o" character. An interesting aspect that can be seen from the figures is the peak that occurs in numbers 8 and 9. Knowing the years to which the data leaks belong, we immediately thought that the numbers in question belonged to dates since the most used categories in passwords are dates and ages. We will deepen this aspect in the next sections.

4.6 Characters, symbols, numbers frequency analysis

For each dataset we have carried out checks on the distribution of characters in the passwords.
First we have divided the dataset into passwords from 5 to 12 characters in length. We chose this range because under 5 characters and above 12 the passwords were too few to be a good model of study. We report in Figures 4.23 and 4.24 two of the four datasets we have analyzed. We decided to analyze these two types of data leaks because RockYou didn't have a specific policy, people could enter any type of password, while Ashley Madison did.
What we studied in particular was the presence of the four categories: capital letter, lowercase letter, symbol, number within the passwords.
The green bar, in the graphs, corresponds to uppercase letters, the blue bar to numbers, the orange bar to lowercase letters and finally the red bar to symbols. On the x axis we have entered the positions of each character and on the y axis the frequency of each category in each position.
The first three figures show the 5, 8 and 12 character long passwords of the RockYou data leak. As the password length increases, it is more and more possible to notice how as the length increases, the lower cases increase going down towards the end of the password. The opposite happens for numbers. As for the upper cases, they follow a descending line from the first character to the end of the password.

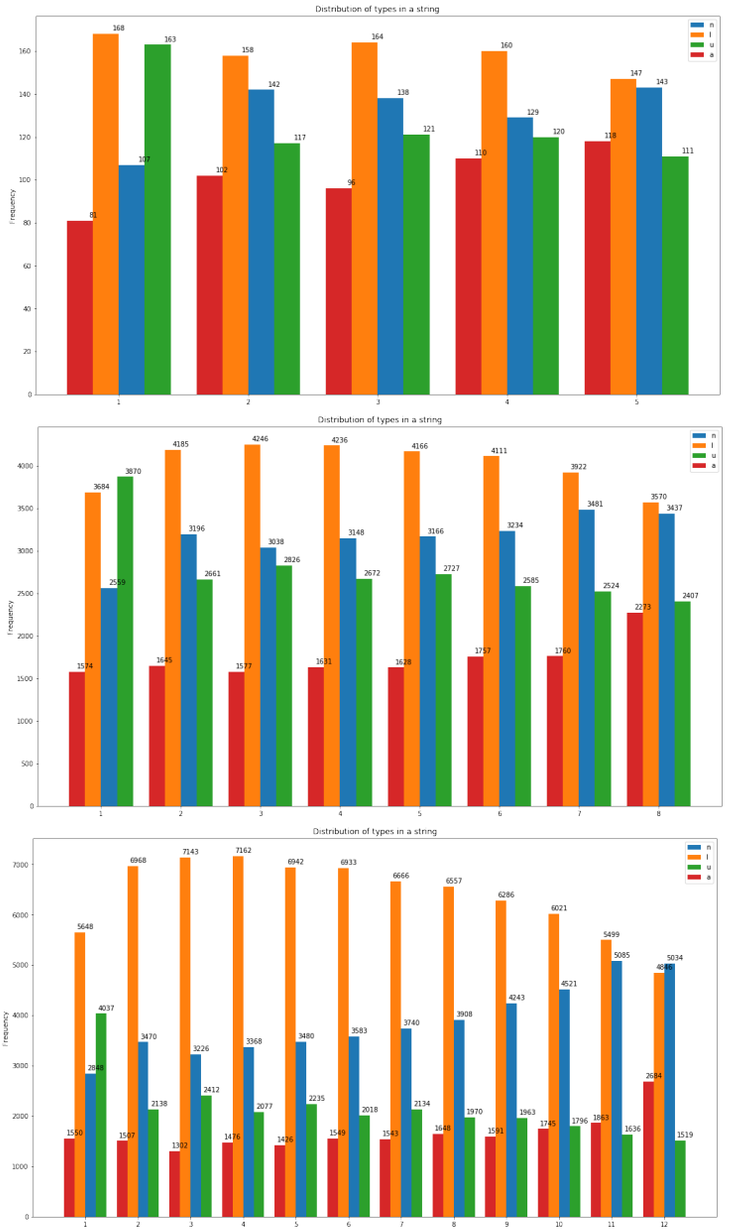

Figure 4.23: Trend lower case, upper case, numbers and symbols in passwords which length is 5, 8 and 12 - RockYou

The last three figures show Ashley-Madison's passwords 6, 8 and 11 characters long. We have chosen other cardinalities because they are more significant for

this data leak. In Ashley Madison's data leak, as in Rockyou, numbers and lower case are inversely proportional. Unlike Rockyou, however, uppercases are much more present at the beginning of passwords. In particular 8 long passwords it is possible to notice that the first letter is an upper case, then we have a series of lower case letters and the last two characters are numbers. This follows the typical patterns analyzed so far. In longer passwords the first character is mostly a lowercase character but with little difference we find many upper cases.

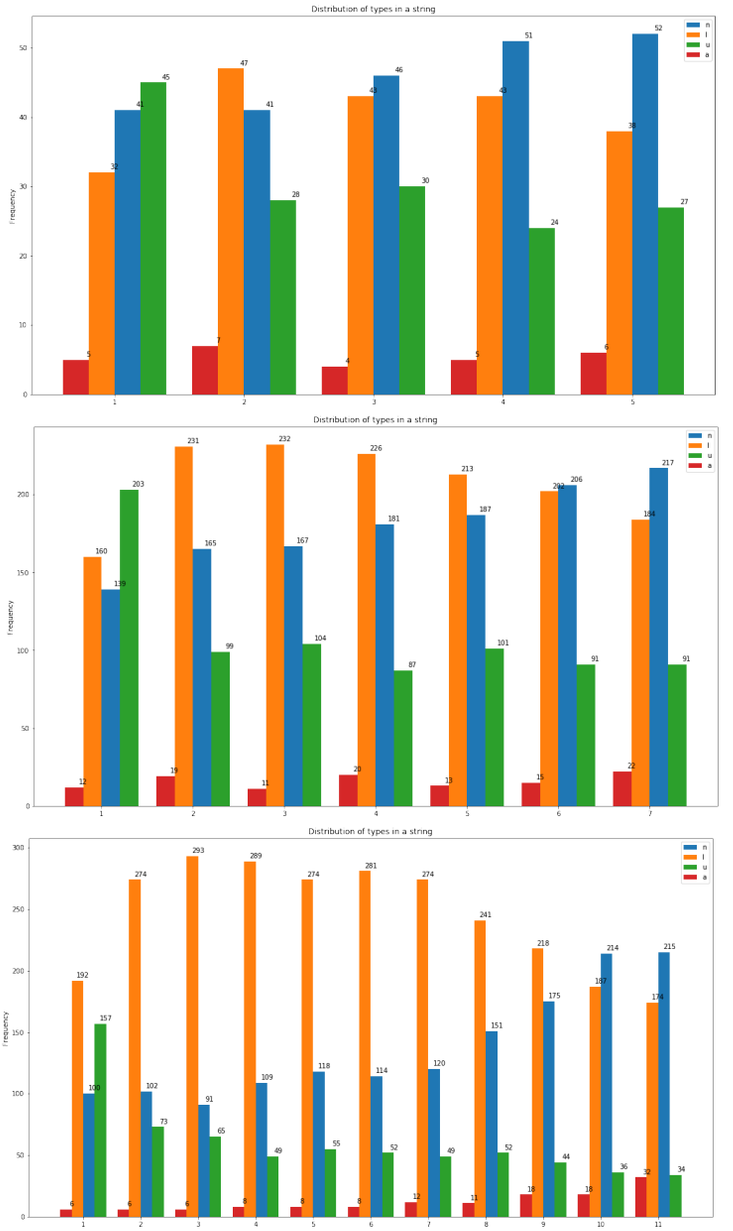

Figure 4.24: Trend lower case, upper case, numbers and symbols in passwords which length is 6, 8 and 11 - Ashley Madison

We have also analyzed the numbers most present within the various passwords according to the position and length of the passwords themselves. We eliminated

the first four numbers (0, 1, 2, 3) from the chart since they were the most present in all positions. We have chosen two representative datasets and reported the analysis only for passwords of length 5 because on average the other lengths follow the same trend.

First of all, from Figures 4.25 and 4.26 it is clear how the presence of the numbers increases as we approach the end of the password. There is a peak of the numbers 8 and 9 in the second-last place. While in the last place, the most present numbers are 4, 5 and 7.

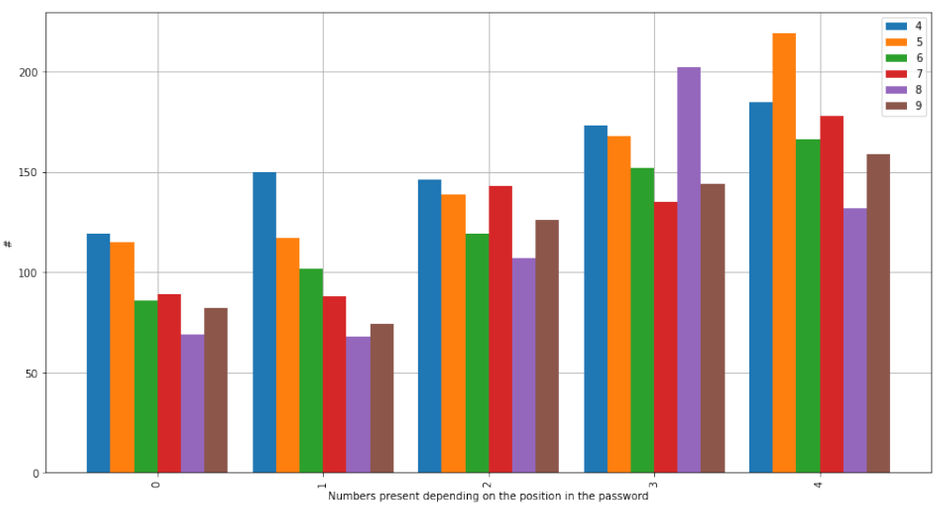

Figure 4.25: Numbers from 4 to 9 in passwords depending on location - PhpBB

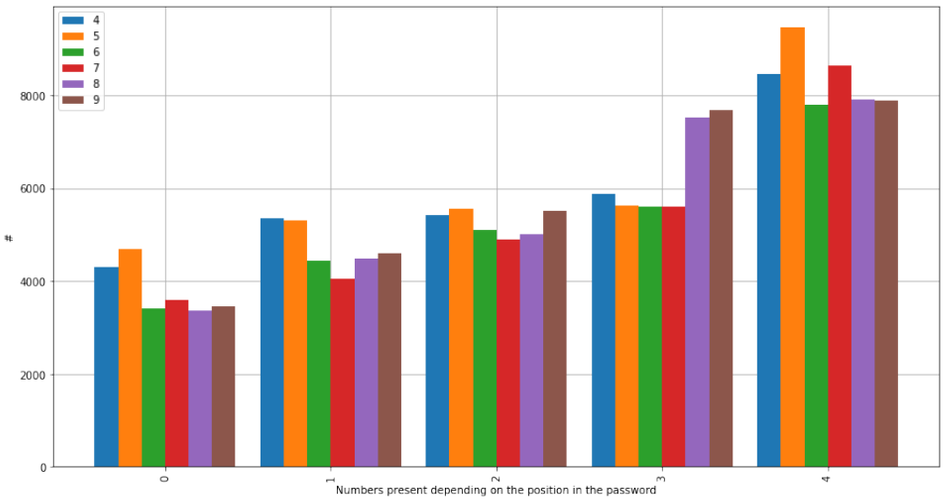

Figure 4.26: Numbers from 4 to 9 in passwords depending on location - Rockyou

The main reason why in the second-last position we find more 8 and 9 is because there is a high presence of dates within the passwords in the format YYYY but also YY (therefore with only the last two digits). Counting the period of data leaks and the fact that dates of birth are the most present category in passwords, it is not uncommon to find years in the 80-90 range. In PhpBB, in the second-last position 8 is more present than 9 while in RockYou it dominates 9. We subjected both datasets to an analysis on the presence of years in the YYYY format and divided the results by decades. What emerged is visible in the tables 4.18 and 4.19. It is possible to notice that in PhpBB there is a greater concentration of the 80s while in RockYou of the 90s as aspected.

Decade	%
50-59	3.91%
60-69	7.47%
70-79	14.33%
80-89	26.28%
90-99	11.1%
2000+	30.87%

Table 4.18: Percentage years present in passwords divided by PhpBB decades

Since, as we have seen in other sections, birth years are often entered accompanied by an arbitrary string, we investigated the average age of the population that used the internet in the years in which the data leaks occurred. As the article [3] shows, the 18-29 and 30-49 year-olds use the internet more than others. Counting the data leaks occurred in 2009, those born in 1980 were 29 years old at the time so it seems reasonable that that decade is the most present.

Decade	%
50-59	5.08%
60-69	8.10%
70-79	13.98%
80-89	30.65%
90-99	36.03%
2000+	6.16%

Table 4.19: Percentage years present in passwords divided by RockYou decades

We analyzed how many, and which, symbols there were inside the data leak, so which symbols were the most used in passwords. In Figure 4.27 we have analyzed the symbols present in RockYou and, for each length, their cardinality. We have several peaks. The most important are in the exclamation mark symbol, in the asterisk, in the period, in the at sign and in the underscore.

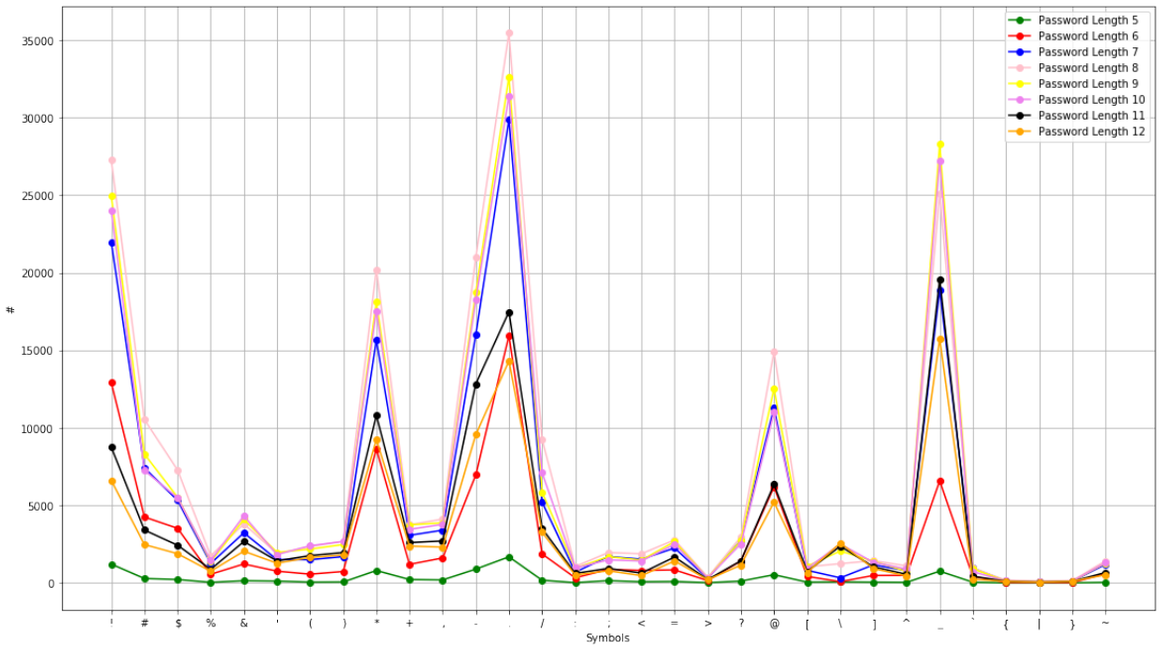

Figure 4.27: Trend Symbols in Rockyou

In section 4.7.1 we have deepened the study on dates.

We also analyzed the other three datasets to confirm what we found with RockYou. As you can see from the Figures 4.28, 4.29, 4.30 also for Ashley Madison, Hotmail and PhpBB the symbols most present are the same as RockYou so we can say that those symbols have a higher trend than the others for users who create passwords. In fact, the symbols that we find most in passwords are the most used symbols on the keyboard [58]. For example the exclamation is a punctuation mark used in some languages to denote an exclamatory statement then, the symbol is known as the number sign or the pound sign (not to be confused with the Pound symbol denoting currency) or hash in various countries and so on. Intrigued by the discovery, we also tested the passwords entered by the participants of the questionnaire described in Chapter 6. Again, what was found also occurs in the passwords entered by them (see Figure 4.31).

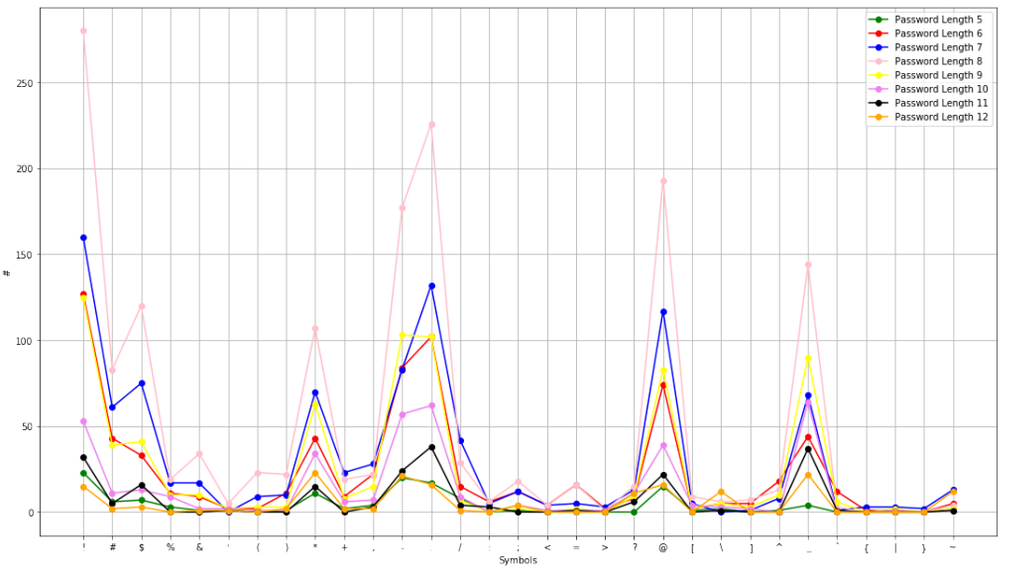

Figure 4.28: Trend Symbols in PhpBB

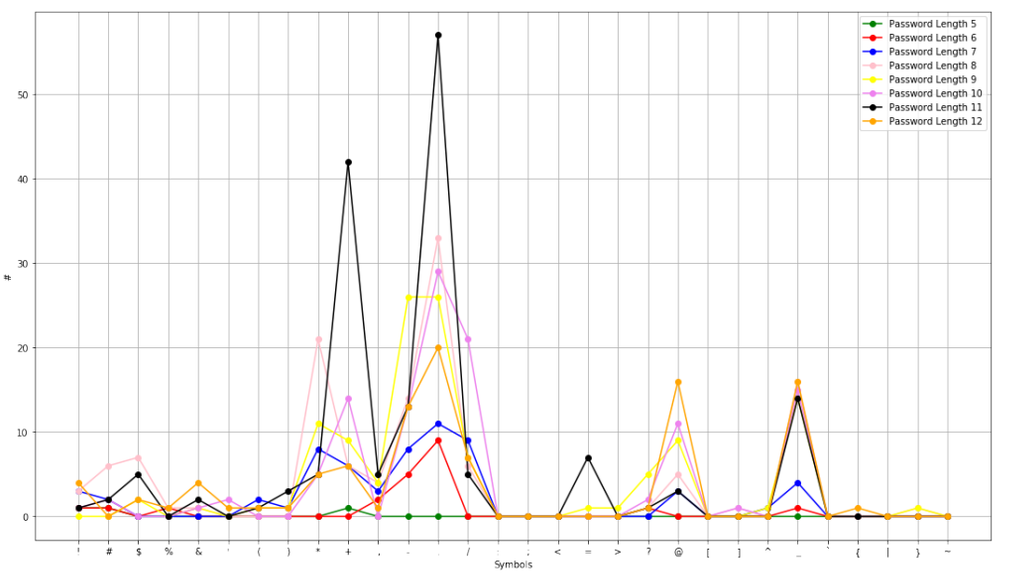

Figure 4.29: Trend Symbols in Hotmail

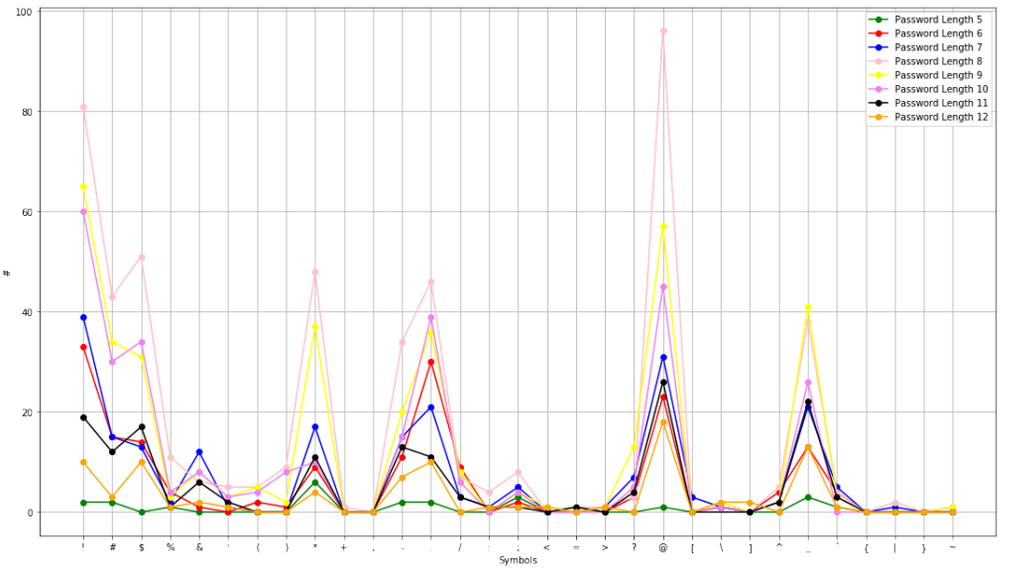

Figure 4.30: Trend Symbols in Ashley Madison

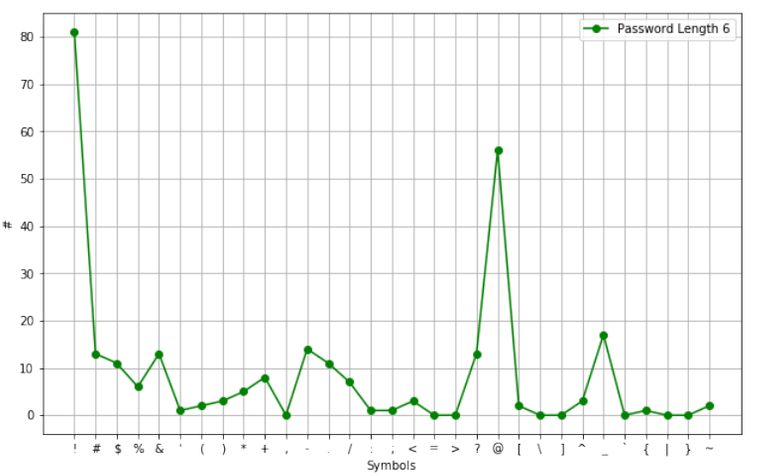

Figure 4.31: Presence of symbols in the passwords entered in the questionnaire we created. See Chapter 6

In Figure 4.32 we have created a comparison between the various datasets to better visualize that the trend of the symbols used is almost identical for all 4.

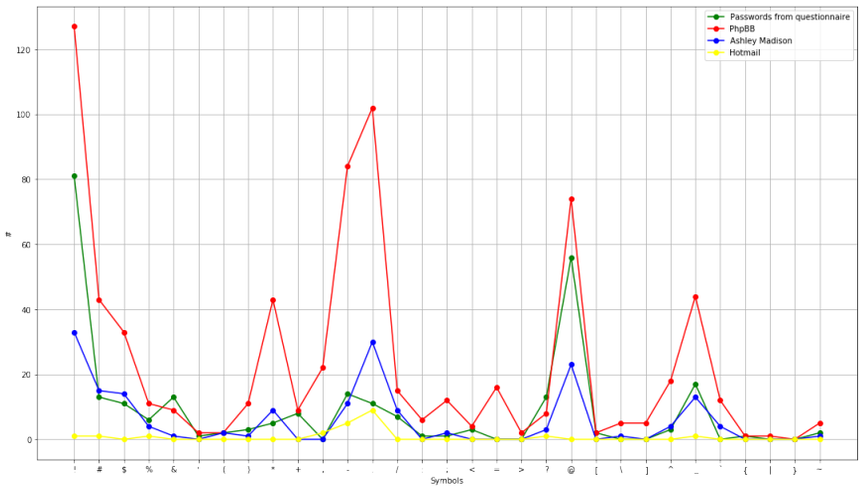

Figure 4.32: Comparison of datasets

Further analysis was to verify which was the most used symbol and if there was a direct connection with the various positions within the passwords. In figures 4.33, 4.34 we report for the RockYou and PhpBB datasets the trend of the symbols in passwords of length 5. We have chosen to show passwords 5 characters long for simplicity and because on average the other lengths follow the same trend.

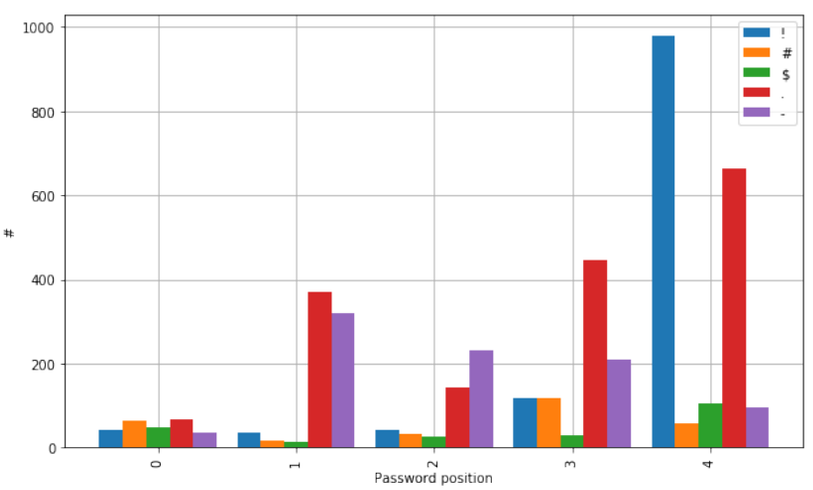

Figure 4.33: Position of the most used symbols in passwords of length 5 - RockYou

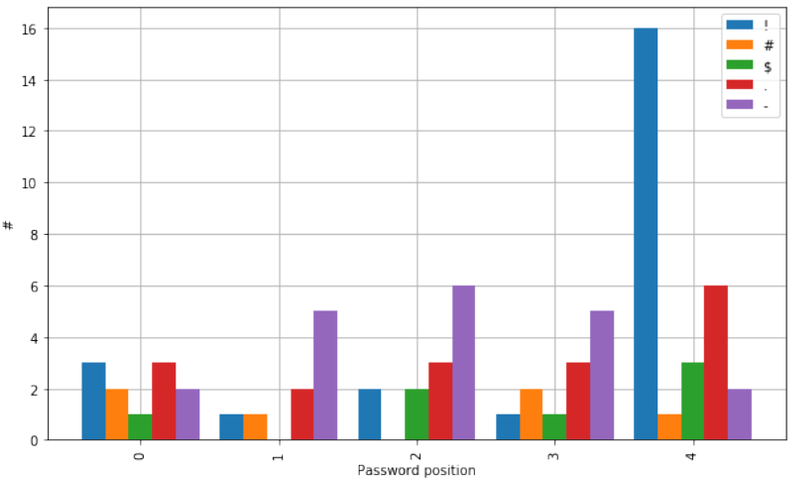

Figure 4.34: Position of the most used symbols in passwords of length 5 - PhpBB

It is possible to note, for both datasets, that the dot and the dash compete for the central places while more frequently we find, as the last character, the exclamation point.

4.7 Password categorization

In this section you describe some categories of words found in the passwords of the datasets we have analyzed. We used RockYou as a training set following a bottom-up design. So, by sampling some passwords from the dataset, we found some categories used. These include personal names, surnames, pet names, names of famous inventions / objects, dates, ages, numbers, superhero names, eroticism, songs, band, singers and colors. Having discovered this, we tested their presence in the various datasets by obtaining files of strings of the categories found and applying a Levenshtein Distance with a minimum threshold (minimum distance) variable depending on the accuracy we desired. Too small a minimum distance is too selective, too large a minimum distance also risks admitting false positives, so

the minimum difference we used is 2 or 3. For greater accuracy we will repeat the tests in the future trying to replace the symbols with the respective most common characters.

4.7.1 Dates frequency analysis and meanings

According to the paper [121] we focus, also, on dates in passwords. We have analyzed the most present dates and months. In Figure 4.35 we report a table of the paper in which some statistics are described on the numbers present in the passwords and on how many of these form passwords. In the reference paper, the analysis is done only on the RockYou dataset. Instead, we applied what we found in the other three reference datasets that we have used so far to verify that the discoveries are not specific only to a specific dataset.

Subset Description	# of Passwords	% of RY Passwords
(1) Passwords containing sequences of at least 4 digits	8,056,329	24.72%
(2) Passwords from (1) above that match a numerical pattern (see Section 3.2)	1,346,410	4.13%
(3) Passwords containing 5–8 consecutive digits	4,974,602	15.26%
(4) Passwords that are exactly 5–8 digits (all numeric digits)	3,951,852	12.13%
(5) Passwords containing 5–8 consecutive digits and match a date	1,934,821	5.93%
(6) Passwords that are exactly 5–8 digits and match a date	1,469,662	4.51%
(7) Passwords that contain a date and other text	358,562	1.10%
(8) Passwords that are exactly 5–8 digits, match a date and numerical pattern	114,724	0.35%
(9) Passwords that are exactly 5–8 digits, match a date, no numerical pattern	1,354,938	4.16%

Figure 4.35: Table of statistics of how numbers and dates appear in the RockYou [121]

We compared the 4 datasets and analyzed which month of the year was most entered in the passwords and, as can be seen from Figure 4.36, May is the one that has the most frequency.

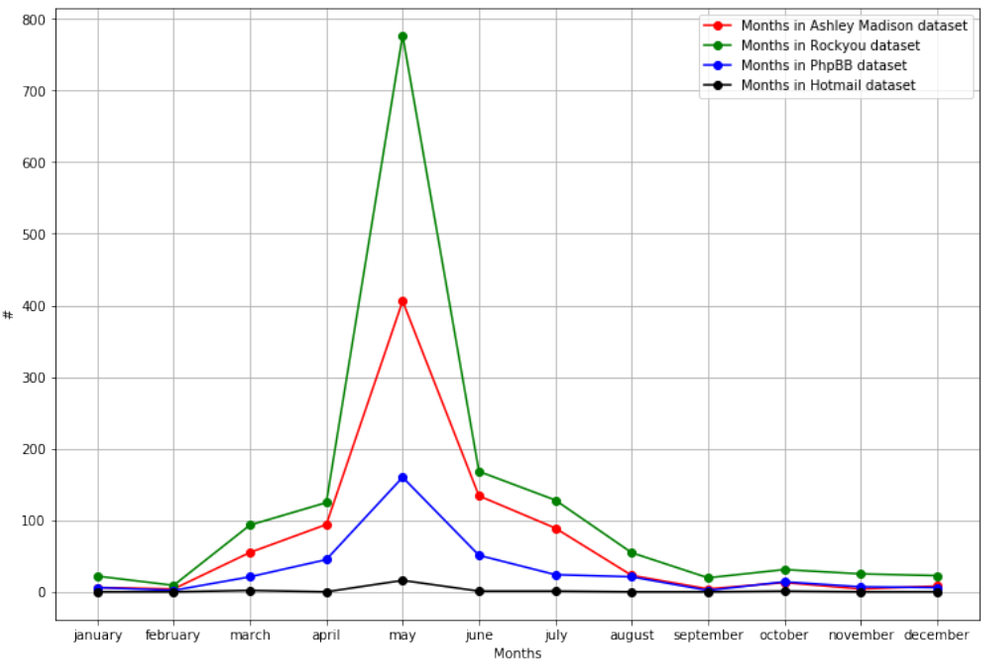

Figure 4.36: Months of the year in passwords for each dataset

Based on what was discussed in the paper [121], in the graph 4.37 we show the holidays most inserted in the passwords and noteworthy dates appearing more frequently than expected. The search was done in the dd/mm format. The dates sought were:

- March 21 (First day of spring; Persian new year)
- December 21, 2012 (date associated with the "2012 phenomenon")
- August 17, 1945 (Indonesian Independence Day)
- April 14 and 15, 1912 (Titanic sank)
- September 11 (Fall of the twin towers)
- December 25, 24 (Christmas and Christmas Eve)
- February 14 (Valentine's Day)

- December 31 (End of Year)
- January 01 (New Year's Day)

The red bar of the Figure 4.37 corresponds to RockYou, the green one to Ashley Madison, the orange one to PhpBB and the blue to Hotmail.

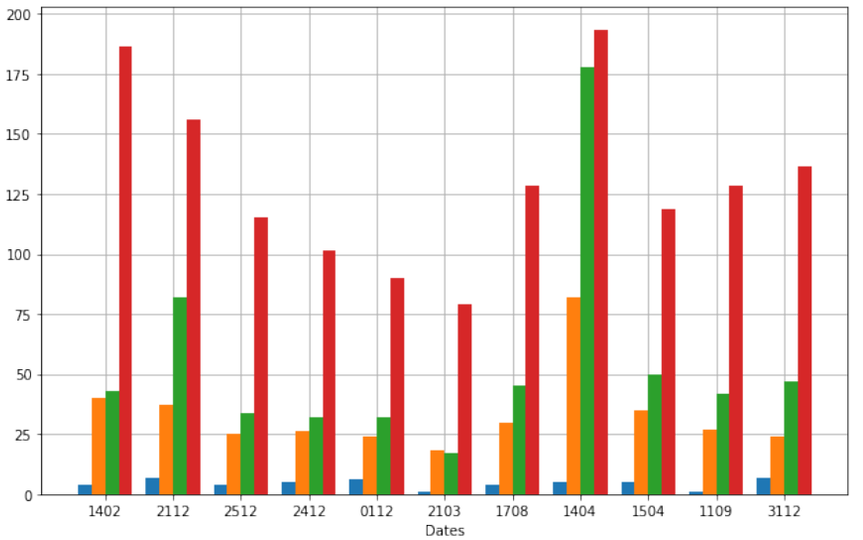

Figure 4.37: Most used dates in passwords for each dataset

Searching for dates didn't turn out to be very easy as there are multiple ways to write the same date and it could often be confused with random numbers.
The importance of knowing these patterns, as demonstrated in the paper by Veras et al. is that, knowing these patterns we can correctly capture approximately 27% of date passwords, which corresponds to approximately 1% of all RockYou passwords.

4.7.2 Other categories

We also analyzed other categories, such as personal names, superheroes, colors (see) etc. We report below the analysis carried out on personal names. We compared, for each group of passwords divided by length, the names of the superheroes that we saved in a dataset.

Personal names As far as personal names are concerned, we looked for the most used names in America, created a dataset and, for each group of passwords (5, 6..,12 characters long) we calculated the probability of finding a personal name. On average, the length of the most used names in America is 6 characters. First we checked the average presence of names for each length. What emerged is that 5-long passwords have more names than the others. In fact, as you can see in Figure 4.38, starting from the 5 characters long group up to the 12 long one, the line decreases. We noticed a similar behavior for the other three data leaks we analyzed.

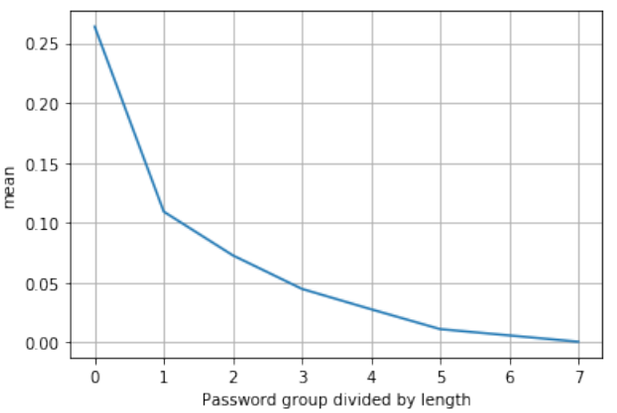

Figure 4.38: Password group divided by length. 0 corresponds to 5 long passwords, 7 to 12 long passwords - RockYou

A further check we did was to see if there was any kind of link between names, dates and numbers. What we have found is that as the password length increases, names accompanied by two numbers at the end of the password increase which can be categorized as age or as a contracted form of the year of birth. For passwords 5 characters long, in fact, the presence of numbers is minimal while for longer passwords we find a much higher frequency of numbers. Referring to paper [96], we deduced that the reason why longer passwords have more numbers than shorter ones is based on the external control of he stream of consciousness. The paper in question subjects some users to a test by offering them different images. The

images whose name is short leads them to count the letters and therefore to think of numbers. Which happens with much more effort for longer words. So we can assume the same happens in passwords. The fact of having words that are on average 6 characters long and the need to enter passwords long from 8 characters upwards immediately makes one think of adding symbols or numbers (assuming that no specific policy has been requested).

The last check we did was to generate the top ten names most present in the four data leaks. We have introduced a threshold under which the name is not taken into consideration since many of them also correspond to commonly used words and therefore risk generating false positives.
As shown in Table 4.20 many of the top ten names for each dataset also occur in another dataset (marked in italics). The most common names are Love and Mari.

PhpBB	**Hotmail**	**Ashley Madison**	**RockYou**
Love	*Love*	*Love*	*Love*
Star	Dani	*Star*	*Star*
King	Ella	*King*	*Ella*
Mari	*Mari*	*Mari*	*Mari*
John	Nita	*John*	Andy
Anna	Juan	*Anna*	*Anna*
Anne	Illa	Andy	Bell
Jack	Bert	*Jack*	Alex
Erma	Lita	Mike	Dani
Angel	*Angel*	Rick	*Angel*

Table 4.20: Ten most common names for each dataset

4.8 Different languages, same choices

In this subsection we have compared the data leaks with the most used words in the other languages mentioned at the beginning of the chapter. We wondered if users of different languages resulted in different patterns. We analyzed the length of each word within the dataset of the most used words in the different languages. While Spanish has 7 characters long in most words, German has 6 characters long. (Fig. 4.39, 4.40)

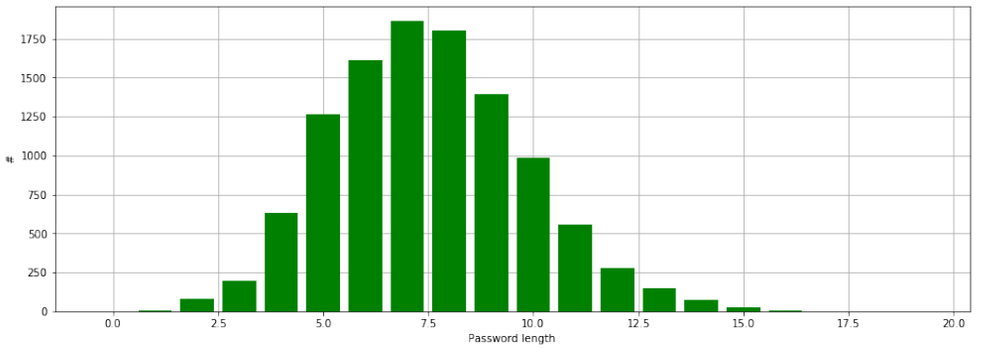

Figure 4.39: Password length most common words in Spanish

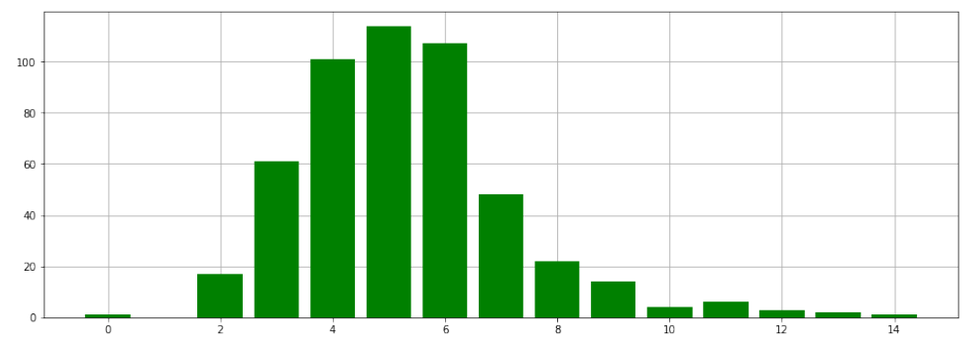

Figure 4.40: Password length most common words in German

For these other two languages we have decided to focus on the patterns used within the passwords. We then divided the data leaks into passwords that had a minimum Levenshtein distance from the words contained in the two language datasets. Thanks to this division we have subjected the files created to the analyzes made with the passwords in English that we have done previously. As there are fewer words in Spanish and German we have fewer frequencies but enough to analyze patterns. We report the long passwords 5, 8, 12 for RockYou and 6, 8 for Ashley Madison as we did in section 4.6 and 10 instead 11 because there are too many password long 11 characters in spanish.

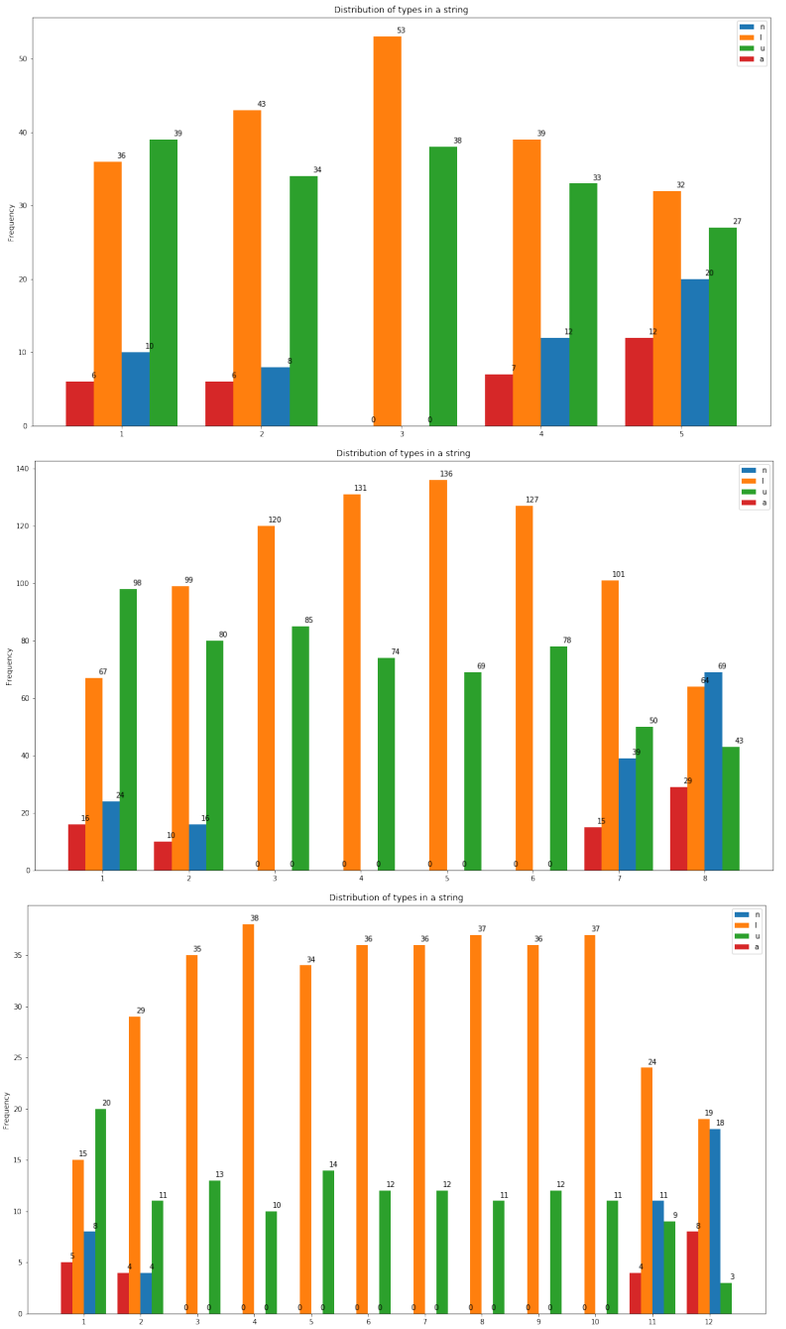

Figure 4.41: RockYou 5, 8, 12 Spanish

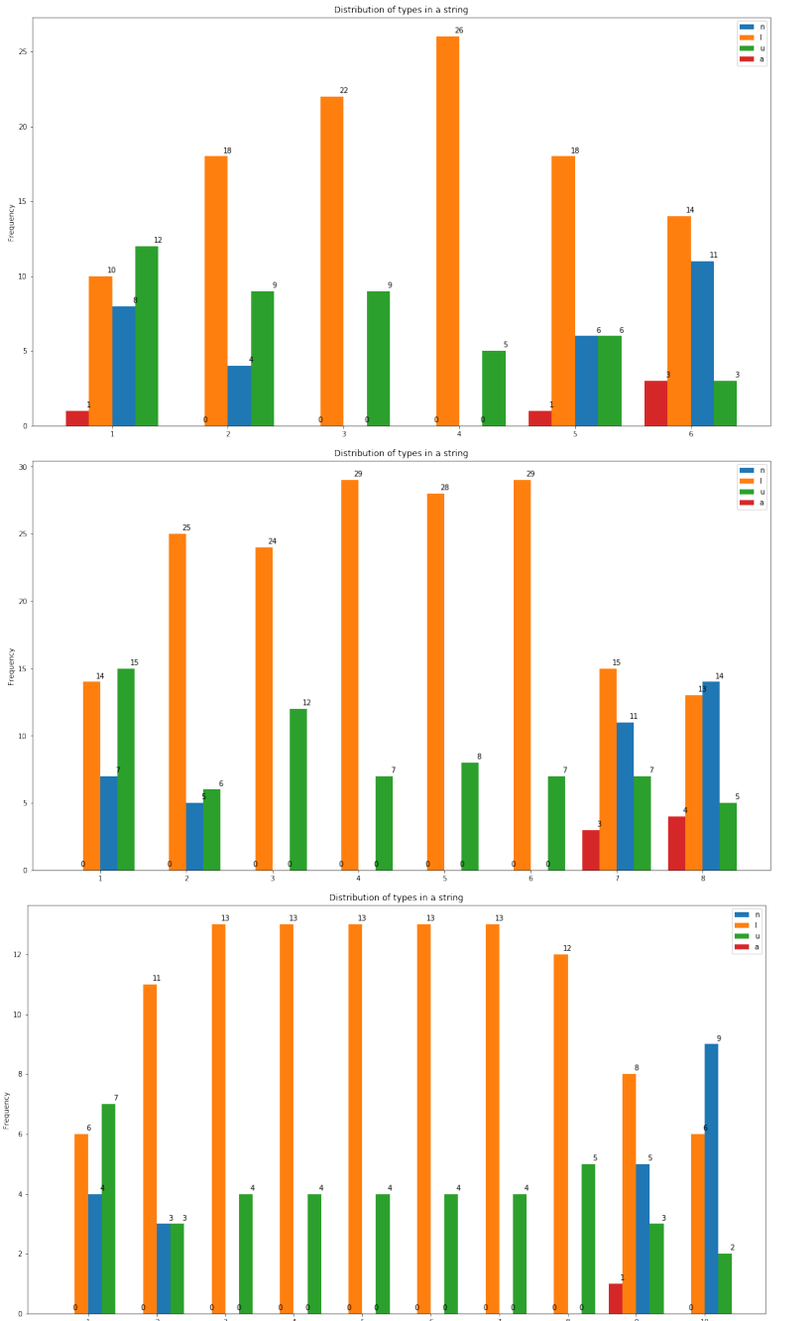

Figure 4.42: Ashley Madison 6, 8, 10 Spanish

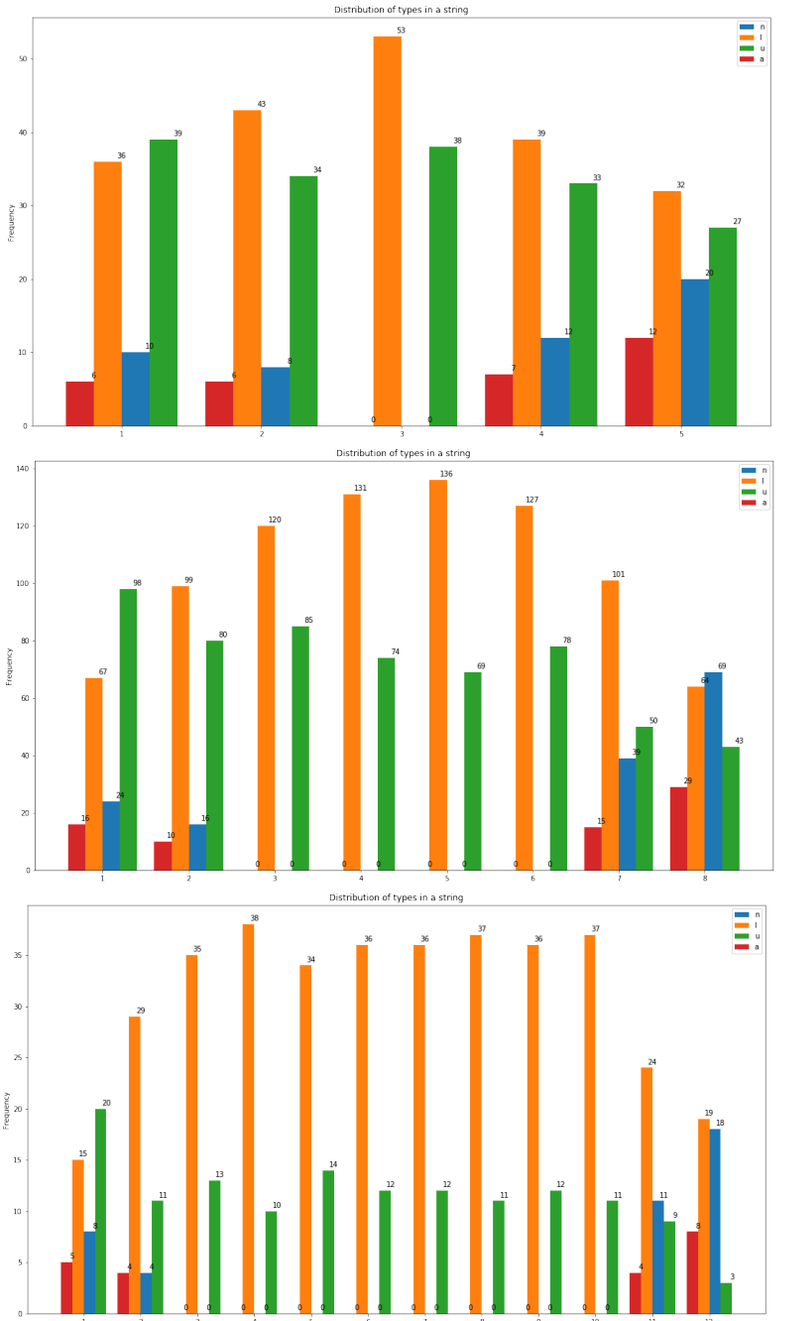

Figure 4.43: RockYou 5, 8, 12 German

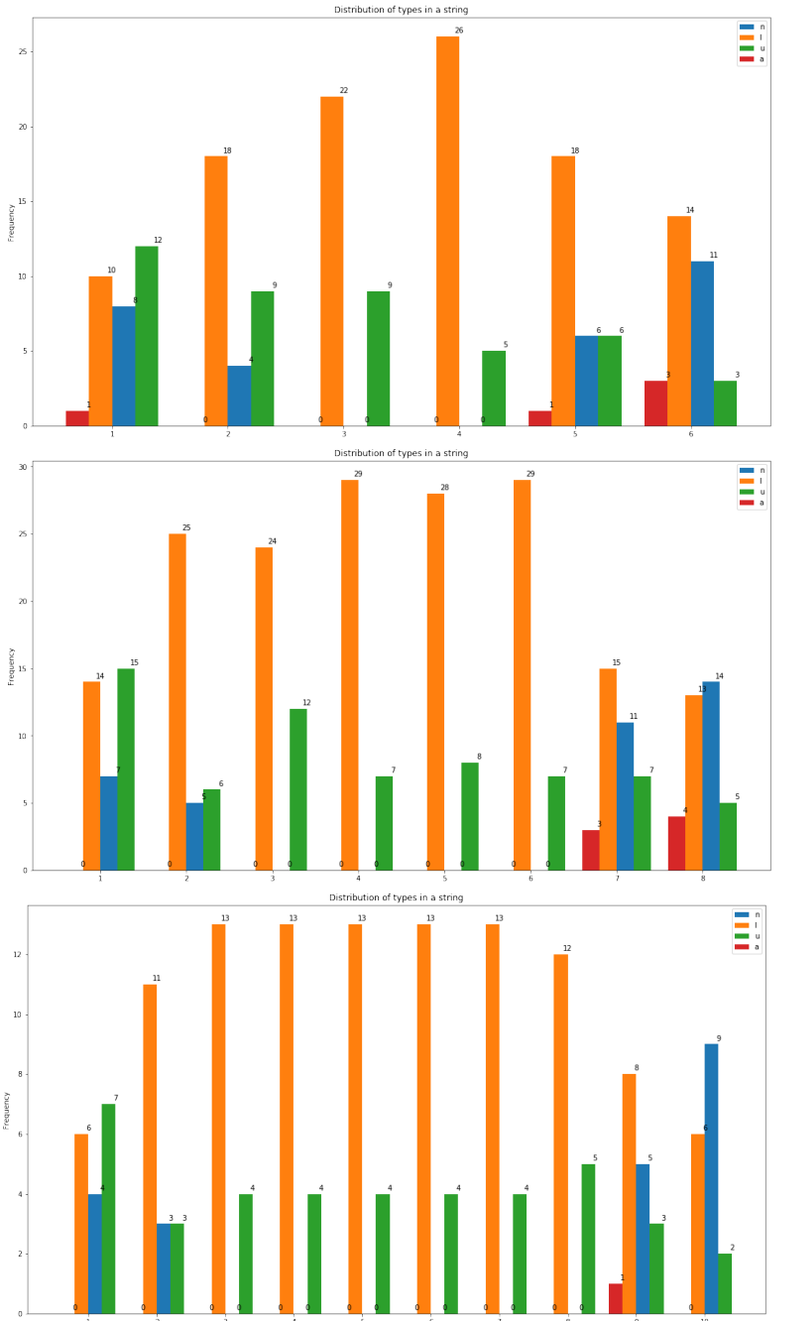

Figure 4.44: Ashley Madison 6, 8, 10 German

Having filtered the passwords for words more similar to the datasets of the most

used words in Spanish and German there are few numbers because passwords composed only of numbers were not included. Knowing this we can see from the images above Figures 4.41, 4.42, 4.43, 4.44 that even those who wrote their password in a language other than English followed the usual patterns. Therefore we can say that even when changing language the patterns do not change, so they are not directly related to the language used.

4.9 Best practices

4.9.1 Users-side

So far we have been discussing policies and passwords. The main goal is always to keep passwords safe with the (almost) certainty of not forgetting them. But what are the best methods? We propose below some best practices based on our studies. First of all it is important that the password is not too short, the longer it is the longer it will take to crack it. It is also very important that it is as varied as possible. So it contains both lowercase, uppercase, digits and special characters. Adding digits and characters does not have to lead to making the classic mistakes we have described. Avoid known patterns such as appending patterns, prepending, inserting and pattern substitution. In fact, inserting symbols or numbers instead of characters is not as safe as it is believed since this technique is known just apply an inverse "function" to go back to the original word. At this point we must be careful not to use words deriving from dictionaries of known words. Although this point is the most complicated to avoid, it is important that this is done because there are specific attacks for this type of password. Don't rely on online entropy calculator tools. Even if the password has a high entropy but follows some of the patterns mentioned it will be easy to guess the string. It is important that a password is also of good quality, so avoid the typical ways.
Table 4.21 shows a summary:

Best practices	Details
Minimum length	six-ten characters. The longer the password, the longer it will take to crack
Variety in the password	contain at least three of lowercase alpha, uppercase alpha, digit, and special character
Avoid patterns	alpha, number and special characters must be mixed up
Dictionary words	do not use common words
Avoid password entropy tools	check the quality of the password not only the entropy

Table 4.21: Summary

We suggest to create a good passwords to use the first letters of a phrase with appropriate substitutions for different letters. For example, "May the force be with you" becomes Mt4%wU where the F in force becomes 4 and the b in be becomes %. Another example might be "Houston, we have a problem." becomes Hwh4p1.
In this case there are no fixed rules so everyone can interpret a sentence as they want. It will be this personalization that will make the entered password remain etched in the mind.

4.9.2 IT administrators-side

User-side suggestions may seem standard but are effective when followed. The biggest tip goes to systems administrators. Knowing the typical errors analyzed so far, it might be useful to apply rules to the system that checks the password. We suggest the use of a system which checks that the user has followed the recommended policy, uses dynamic policies and implements blacklists and autocompletion, without forgetting the usability of the system. It could make easier the choose of the password in the most correct form without making it difficult to remember. Our future developments introduce the devepoling of a system that could take care of all this.

Conclusions In this chapter we have dealt with several issues. We have analyzed the passwords of each dataset in detail. Unlike what is reported in most of the

literature, we did not focus on a single dataset but compared four different datasets. This implies that the discoveries made are not specific to a single dataset but more than one, resulting in confirmation of what we have discovered and analyzed.

Chapter 5

Experimental password evaluation

This chapter describes our questionnaire that we submitted to a sample of 217 people via google forms. We were inspired by paper [65] by acquiring the most interesting parts. We also decided to expand it by studying whether there is a direct relationship between education, work, computer skills and the correct use of passwords. In the previous chapters we talked about the human psychological side and analyzed data leak passwords. In this chapter, we try to find the direct relationship between human and chosen passwords. We will report the questions that have returned the most interesting values with P(n) where n is a progressive number and P stays for "Problem". You will be able to see the question with its "code" and the answers available in appendix (5.1.7.1). (The questionnaire is available here: [19])

5.1 Description of the test and type of questions

In the previous section, we showed some statistics and conjectures about pre-existing data leaks. At this point we would like to know how a human's behaviour conditions the relationship with passwords. To do this, we have created this questionnaire which initially asks general questions: age, gender, most spoken language, maximum level of education and the category to which this level of education belongs. Finally we asked about the type of work and computer knowledge.

5.1.1 Participants description

The questionnaire received 217 responses. It was published on various social networks and shared among various friends and acquaintances. We tried to keep a heterogeneous sample. Of the 217 participants 120 are men and 95 women as shown in Figure 5.1. The average age is 30, the median is 25 years old and the ages predominantly are 23, 24 and 25 years old (see Figure 5.2). We asked participants to respond on impulse without ever going back to old responses. This is because, as written in the old chapters, when we create passwords we react more by impulse than by reasoning.

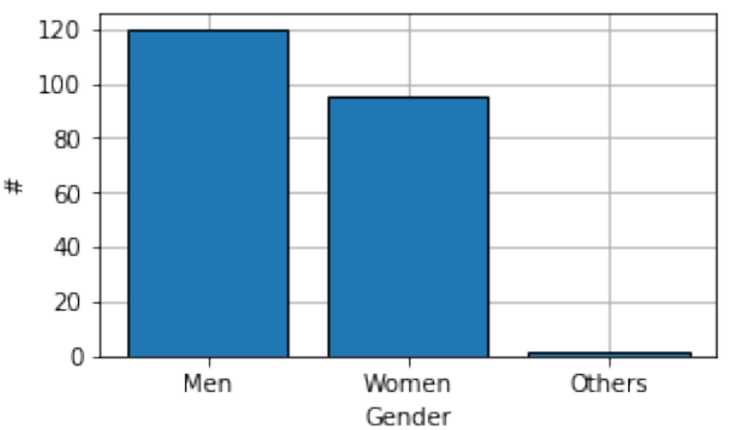

Figure 5.1: Gender distribution

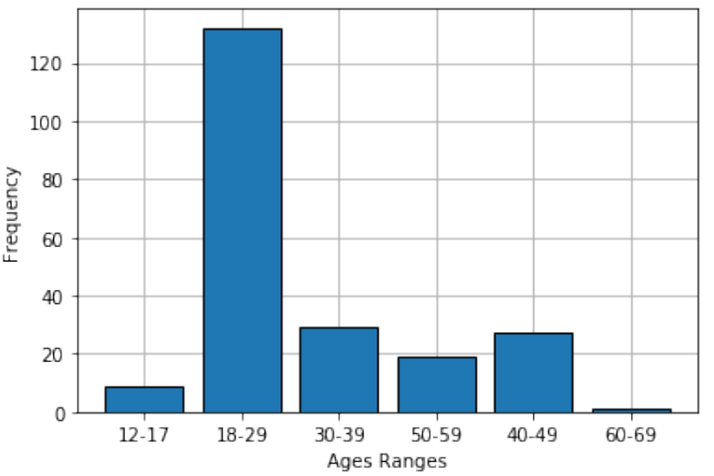

Figure 5.2: Ages distribution

The maximum education level is: 37.3 % Branchelor degree, 42.4 % high school and 12 % Master degree. The others are divided between PhD, master and primary school. 111 out of 217 participants have a university degree as a minimum level of education, increasing to higher levels.
In Figure 5.3 and Figure 5.4 are shown the most frequently choose type of universities and high school.

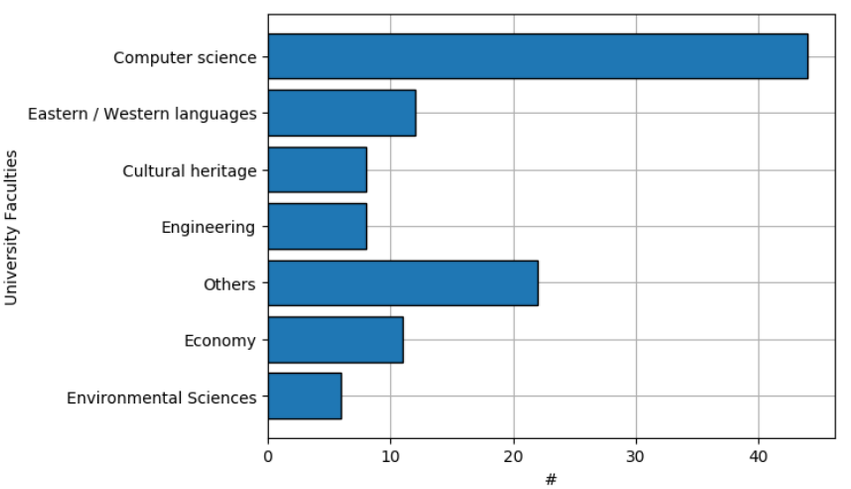

Figure 5.3: Most frequently choose bachelor/master degree

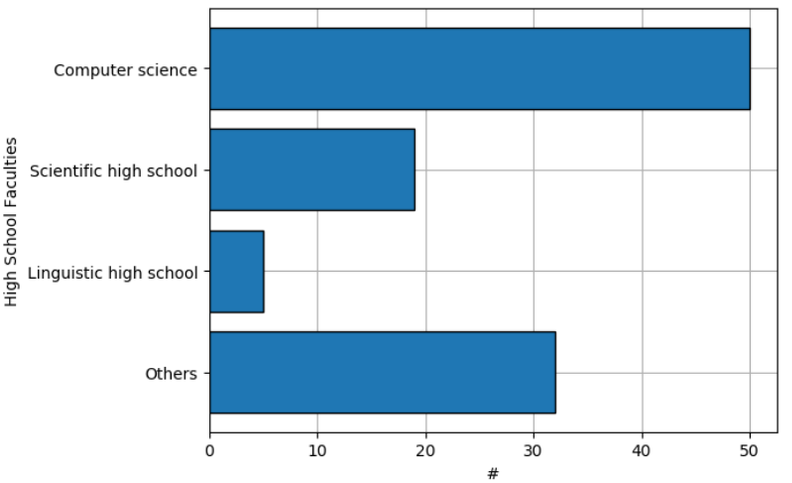

Figure 5.4: Most frequently choose High School

The questionnaire then led users to describe their relationship with passwords.

5.1.2 Relationship with passwords

The questions that have been asked are about how the user believes passwords should be: easy to remember, secure, or both. Whether it is right to reuse the same password often, the pattern most frequently used and what are the categories of names, colors, numbers etc that are most present in their passwords.
The first question in the section asks participants if they often use the same password. The answer is binary, yes or no (P(1)), 66.40% answered yes. We then asked how easy to remember and how secure a password should be. The possible answers were *many*, *enough*, *little*, and *not at all*. They could only give one answer for the two possibilities (P(2)). In the case of *"easy to remember"* the answer *"enough"* masters while in the *"safe"* answer it masters the answer *"a lot"* (see Tables 5.1.2, 5.1.2). In Figure 5.6 we have compared the answers of "*easy to remember*" with "*security*". The x-axis graph has the types of answers that apply to both questions (quite, much..). The height of each square equals how many people have chosen

security for the x-axis value (which corresponds to easy to remember). So those who have chosen that a password should be very easy to remember, most of them have chosen that it should be secure enough.
Having available a scale from 1 to 10 where 1 indicates complete unsafe or absolute difficulty in remembering and 10 the exact opposite, in Figure 5.5 we report the scale of the four values used.

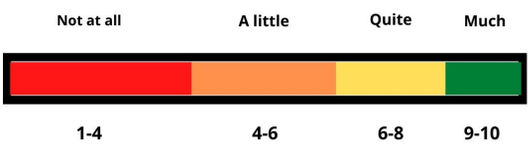

Figure 5.5: Scale Definition

Nothing too surprising, it is common for people to expect their passwords to keep them safe from attackers and malicious people but at the same time they should be easy to remember.
The next question asks you to choose between multiple possibilities. We asked why a password should be easy to remember (P(3)). Those who think that a password should be easy to remember because *they are afraid of forgetting it* (55.30 %), *all sites ask for the same pattern so they don't have the patience and time to think of a new one* (28.90 %) and *doing password recovery bothers them* (23.20 %). The rest have entered their own answers since we have left a free field.

Easy remember	**Percentage**
Much	37.79%
Quite	42.40%
A little	13.36%
Not at all	6.45%

Table 5.1: How important it is that a password is easy to remember

Level of Security	**Percentage**
Much	64.51%
Quite	31.34%
A little	3.23%
Not at all	0.92%

Table 5.2: How important it is that a password is secure

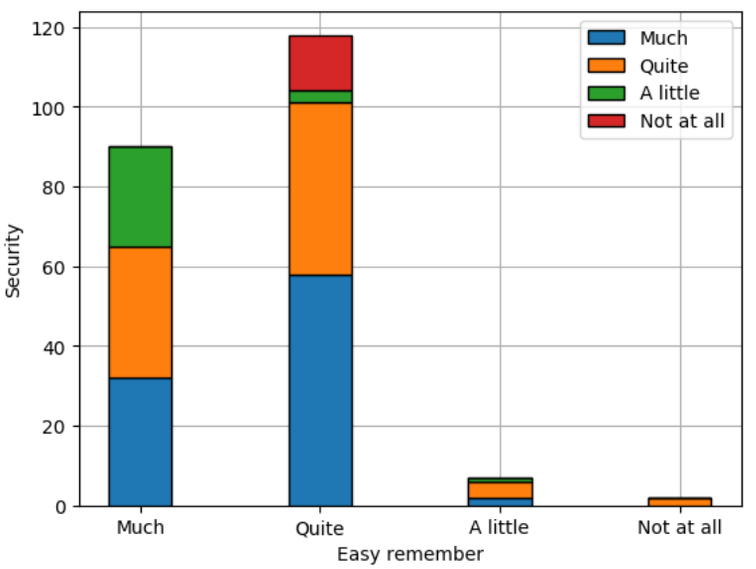

Figure 5.6: Comparision between "Easy to remember" and "Secure"

We asked the participants to think of a random password and to choose the policy they followed (P(4)). The crucial point that we have deeply analyzed in the last chapter (5) are the most frequently used pattern. Also in this case the capital letter prevails in the first position, the lowercase letter in the remaining ones, in the end number precedes symbols (see Figure 5.7).

Figure 5.7: Type of patterns choosed by position

We wanted to investigate which categories of words were most used in the passwords of the participants (P(5)). The categories that have been most successful are numbers and dates. In second place nicknames or modifications of the proper name, in third place personal names and all other categories follow (see the Table 5.3).

Categories on passwords	Yes answers
dates	51.15 %
age	6.45 %
pets names	25.80 %
band / singers names	12.90 %
erotism	5.99 %
superhero names	8.29 %
colors	15.66 %
names of son / daughter	10.13 %
names of relatives	16.12 %
proper name	32.71 %
random numbers	61.75 %
surnames	15.20 %
films	11.98 %
email	4.60 %
nicknames	35.94 %
football team / names of footballers	2.30 %
slang	17.97 %
numbers that remind you of important events	55.29 %

Table 5.3: Categories of words used in password

5.1.3 Passwords comparison

Then we asked the participants, given two related passwords, to choose which of the two was the more secure under their opinion. We have created a scale from 1 to 7. The closer the number is to one of the two passwords, the more the password in question is considered important. For example in figure 5.8 we show an example of a question. If the user chooses 1 it would mean that he considers the password on the left more secure than the one on the right, if he chooses 7 then he considers the password on the right more secure, 4 equally secure.

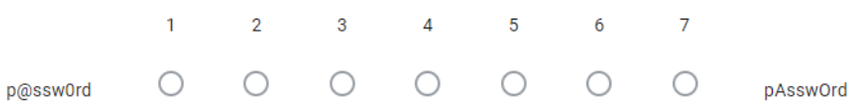

Figure 5.8: Example question comparison between two passwords

In Figure 5.9 we show some of the passwords we asked to compare. In the next table, we have entered the passwords with their relative crack times according to the *Kaspersky password checker* [27] which uses both brute-forcing techniques and comparisons with passwords present in old data leaks.

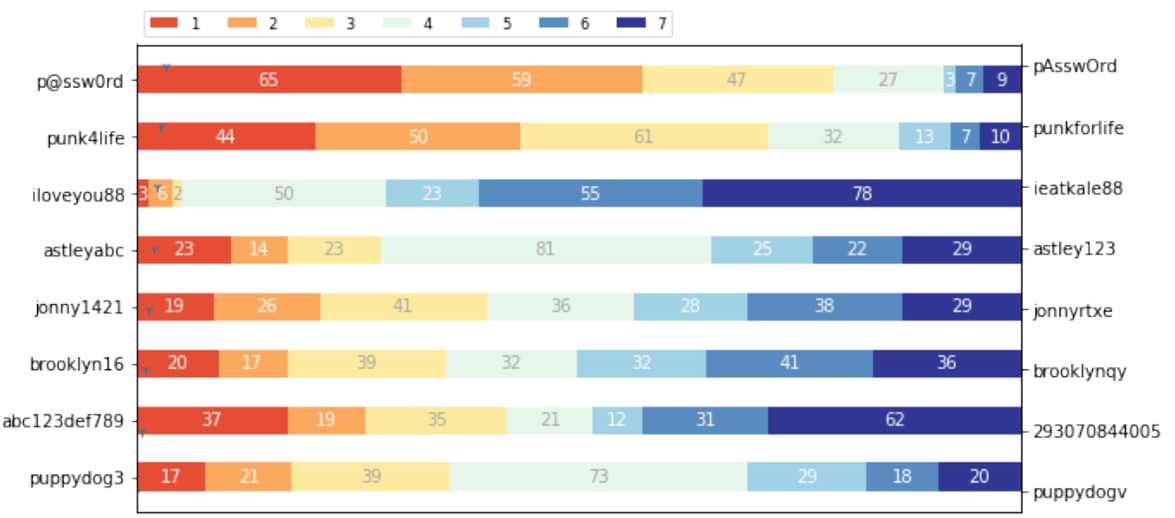

Figure 5.9: Results from question: *Choose which of the two passwords is more secure in your opinion (the closer you get to a password the more you think it is secure, if you choose 4 they are equally secure)*

First password	**Time to crack** (s)	**Second password**	**Time to crack** (s)
p@ssw0rd	1	pAsswOrd	3
punk4life	259200	punkforlife	345600
iloveyou88	180	ieatkale88	1,261e+8
astleyabc	14400	astley123	14400
jonny1421	432000	jonnyrtxe	950400
brooklyn16	780	brooklynqy	780
abc123def789	120	293070844005	1,261e+10
puppydog3	1320	puppydogv	1320

Table 5.4: Passwords subjected to a password checker (Kaspersky [27]) with the relative crack times expresses in seconds

In the table above we have entered some of the passwords present in the questionnaire with the relative cracking time expressed in seconds.
Participants responded in greater percentage that passwords *p@ssw0rd, punk4life, ieatkale88, astleyabc, jonnyrtxe, brooklynqy, 293070844005 and puppydog3* are the safest ones.

• *p@ssw0rd* was considered stronger than *pAsswOrd* because there are a symbol and a number so the character set used is larger than the second. Unfortunately, this password suffers from a strong vulnerability that we talked about in the password chapter: *password munging*. In fact, the character *a* has been replaced with an *at sign* and the *o* with a *0*. This way of behaving towards weak passwords to make them seem stronger is now a common way of doing that nullifies the attempt to make the password stronger.

• As for the first password, *punk4life* also suffers from a way of doing things that have become common. In fact, it is not strange that the 4 is replaced with the for since the sound of the two words is the same. So even in this case, although the character set is more varied, it is muffled by a common way of doing that makes it vulnerable. *punkforlife* is considered stronger because it is long even though the character set includes only lowercase letters.

• *ieatkale88* is surprisingly behind *iloveyou88* in terms of crack times. Furthermore, the participants found it, rightly, stronger. The reason is that it was confronted with a password already known as weak. In our point of view, however, *ieatkale88* could be easily cracked by using a dictionary of English words as an aid

to a dictionary attack.

• *astleyabc* and *astley123* are tied for both the checker and the participants. In fact, although the second has numbers, these are the classics used in most passwords so even if the character set is more varied than the first, the fact that the numbers are common makes heterogeneity vain. If we use a triplet of other numbers, the password is considered more secure.

• *jonnyrtxe* was rated as more secure password than *jonny1421*. Until now most of the participants have preferred passwords with numbers and symbols but this time the focus has shifted to the strangeness of the word since *rtxe* is not a known and used word. The checker has voted *jonnyrtxe* as more secure since it does not belong to any data leak but the entropy of the password is very low so an expert user could easily guess it.

• *brooklynqy* was considered stronger than *brooklyn16* as the previous password. This is because, as for jonnyrtxe, although brooklyn is a very well-known word, *qy* is a rather strange extension of it, therefore, considered stronger than two simple numbers. Unfortunately, however, *qy* is also among the common mistakes since *q*, in the keyboard, is the letter under the 1 and the *y* is the letter under the 6 and this is an already known method of changing passwords. In fact, the checker deems them safe in the same way.

• *abc123def789* is clearly a weak password as it has two common patterns put together. Instead, *293070844005* is a numeric-only password, rarely used in previous data leaks but has a vulnerability: being only numerical. In fact, its entropy is only 21 bits so it can be easily cracked with a brute force attack.

• *puppydog3* compared to *puppydogv* was considered stronger by the participants because compared to *puppydogv* it has a number and the common belief is that adding a number to a weak password makes it stronger. Unfortunately, this is not the case. Even though its entropy is higher than *puppydogv*, it still makes it vulnerable to brute force attacks along with a dictionary attack. In fact, with a dictionary attack, it is easy to find the word puppydog and brute-forcing the last character.

The percentages of correct answers are listed in the Table 5.5

Safest Passwords	**Correct answers** (%)
pAsswOrd	8.75
punkforlife	13.82
ieatkale88	71.89
astleyabc/astley123	37.33
jonnyrtxe	43.78
brooklynqy/brooklyn16	14.75
293070844005	48.39
puppydog3	33.64

Table 5.5: Correct answers

5.1.4 Attacks and malicious users

Then, we investigated the thoughts and knowledge of the world of cyberattacks focusing, of course, on password attacks by asking their opinion on the types of attacks, who would care to know their password and why they would do it.
We gave a list of possible attackers and asked participants to specify who, in their opinion, would be most likely to have their passwords. The choices were *stranger*, *family member*, *friend*, *colleague*, *other people they know.* They could choose more than one answer (P(6)) (see Table 5.6).

Then we gave different ways of attacking and asked the participants what, according to them, an attacker does to try to guess their password. The answers were multiple so they could choose more than one answer. The available answers were: *use software, brute forcing, test most used and known words and names in my language, test common passwords, test dates and numbers* (P(7)) (see Table 5.7).

Finally, we asked the participants to share their ideas on why an attacker should try to guess their password (P(8)). Most attendees think an attacker would want their password to collect personal information and identity theft as in [65]. Financial reward follows (see Table 5.8).

Who Tries to Guess Passwords	#	%
Stranger	142	65.40%
Family	52	24%
Friend	47	21.70%
Co-worker	36	16.60%
People I know (generic)	46	21.20%

Table 5.6: Answers to: *Given the following list, choose who, in your opinion, would be more inclined to steal one of your passwords*

How Tries to Guess Passwords	#	%
Use software	159	73.30%
Brute force	64	29.50%
Try popular words and names and known in my language	72	33.20%
Try common passwords	91	41.90%
Try dates and numbers	84	38.70%

Table 5.7: Answers to: *What do you think a malicious user does to try to guess your password?*

Why Tries to Guess Passwords	#	%
Financial reward	95	44%
Collects personal information	159	73.60%
Identity theft	140	64.80%
Fun / proof they can	76	35.20%
Spamming	43	19.90%
Espionage	45	20.80%

Table 5.8: Answers to: *Why would an attacker try to guess your password?*

5.1.5 External stimuli

The third to last section dealt with external stimuli. We presented three websites through 3 figures. The first was a site dealing with dogs, the second selling CDs and the third selling helicopters. We wanted to investigate how much the external stimulus (the image) would have influenced the entered passwords (P(9)). The interesting result we had was that one of the questions in the survey asked *how safe it was to choose a password based on the site* the user wanted to sign up

for. 84% of respondents said *it was unsafe*. When they arrived at the part of the external stimuli, where we asked to enter three passwords for the three different sites, 24% of them, however, entered passwords strictly related to the website we presented to them. This behavior leads back to the cognitive dissonance because, despite being aware of the fact that being in uenced by the site where they are is not a good technique of secure passwords, they still preferred to be deceived by the images in fact, 24% even knowing that what they were doing was insecure they still preferred to proceed. In addition, 6.91% entered the same passwords for all three sites, and 13% of them answered No to the question *"Do you often reuse the same password?"*. An interesting event has emerged. As many as 62.77% of the people who claimed to reuse the same password used 3 different passwords for the sites. This probably happened because they knew they shouldn't have remembered them in the future. And therefore, the realization that always using the same password is wrong came into play. So the fear of forgetting the passwords entered did not arise, giving vent to the side aware of the correct habits to adopt.

5.1.6 Choice strategies

The second-last section focuses on password selection strategies. That is, how some ways of composing a password are easier and / or safer in the eyes of users.

We subjected the participants to 6 questions in which we asked them to enter the level of security and how easy it was to remember the pattern mentioned in the question.
We report in Figure 5.10 (P(10)) and 5.11 (P(10.5)) the first and last questions respectively. Participants find a simple password juxtaposed with numbers very easy to remember. In addition to being very easy to remember, they also think it is quite safe. On the other hand, they find it very safe to write down their passwords in diaries or notes on their mobile phones as well as make passwords easy to remember. Obviously no one can compromise a piece of paper, but what if that paper is lost? What would happen if someone found someone else's unlocked cell phone?
46% of participants found it quite safe to use a phrase that they create exclusively

for the account and that has nothing to do with the account itself (ex: Amazon123 for the facebook.com site) (P(10.1)) 39% also find it quite easy to remember but a good portion, 36%, find it quite difficult to remember.
The next question asked how easy it was to remember and how safe it was to enter the name of one of their family members and their year of birth (P(10.2)). 71% find it very easy to remember but 51% don't find it safe.

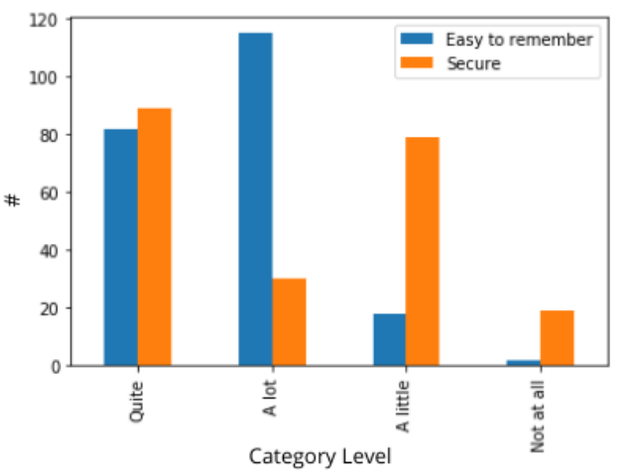

Figure 5.10: Question title: S*tart with a word that comes to mind and then add digits or symbols at the end (ex: dog12!))*.

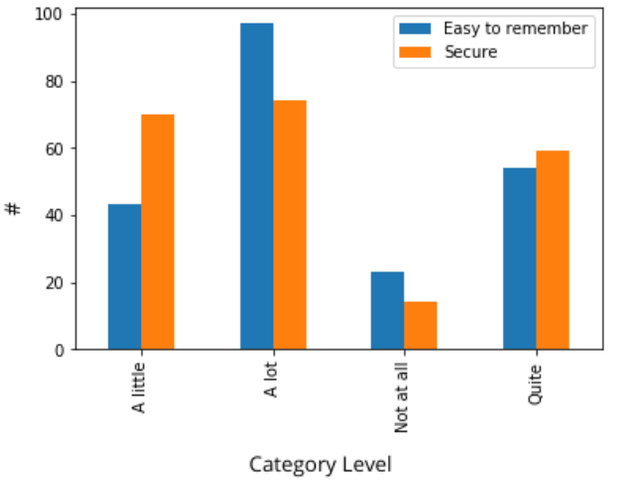

Figure 5.11: Question title: *Choose a strong password and write it down on a piece of paper or write it in your phone notes.*

5.1.7 Password Habits

The last section is about habits. We asked the participants to enter two passwords, changing the request slightly. The reason we entered these two requests is to see how much attention is paid to the request specifications at first glance, how many would have entered the same passwords and how many would have actually followed the specifications. This is to demonstrate how the habit of always behaving the same way in front of passwords (because maybe the same pattern is always requested) has started an autopilot in our minds. Since almost all websites require the same policy it is easy for users not to read what is requested. One had as a specification *"Enter a password that must be at least 6 characters long with lowercase, uppercase, symbols, and numbers"* (P(11)), the second instead *"Enter a password that must be at least 8 characters long with uppercase, lowercase, numbers, and symbols"* (P(12)). The requests are similar, in reality, two passwords are requested that are formed differently.
Regarding these two requests we decided to investigate, again, who had written passwords that respected the request and were strong.

Enter a password which must be at least 6 characters long with lowercase, uppercase, symbols and numbers In this case, 92 % entered a password correctly, following the policy reading carefully what was requested. Of these people only 15.22 % entered strong passwords and only 6 % is made up of computer scientists.

Enter a password which must be at least 8 characters long with upper and lower case letters, numbers and symbols In this case, 90 % entered a password correctly, following the policy. So they read carefully what was requested. Of these people only 24 % entered strong passwords and only 11 % of them is made up of computer scientists.

The last question asked if the participants realized that *the two previous questions were different*. 76.50 % answered no. This is a clear sign of how strong the habit is and how the attention is selective. Since attention is a *limited resource* [101], we have to be selective about what we decide to focus on so we must also filter out an enormous number of other items. The habit of focusing only on part of the request led participants to enter, in the penultimate and the third last question, two identical passwords up to 16.10 %.

Symbols As in chapter 5, we wanted to subject the passwords entered by users to a check on the symbols most used within their passwords. As for the data leaks used in chapter 5, also in this case, the most used symbol is the exclamation point, the at sign and the hyphen. (see Fig. 5.12).

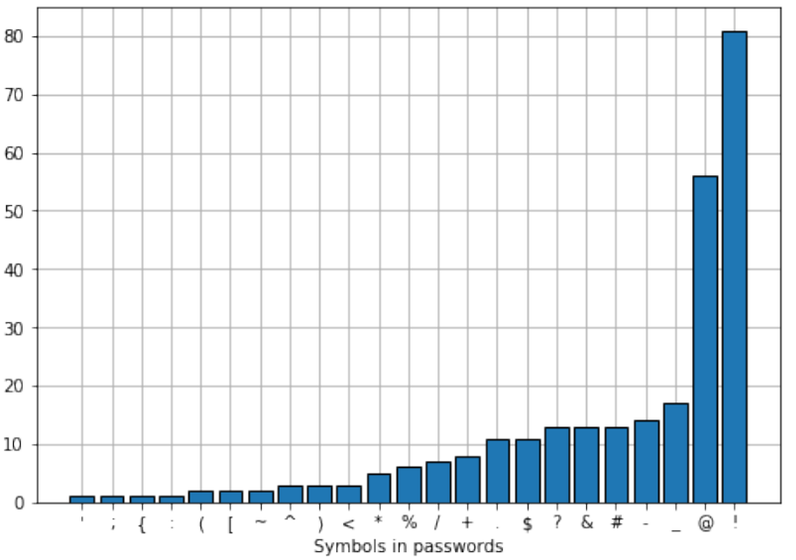

Figure 5.12: Symbols in passwords

Pattern Frequency Analysis Taking up what was analyzed in section [**?**]. We calculated the pattern frequency also for the passwords entered by the participants in the questionnaire, in the same way we calculated the data leak patterns.

Pattern Class	#	%
$U^+L^+N^+O^+$	39	18.22%
$U^+L^+O^+N^+$	11	5.14%
$L^+U^+O^+N^+$	9	4.20%
$U^+L^+O^+L^+N^+$	7	3.27%
$U^+L^+N^+$	5	2.33%

Counting the policy that had been requested, the participants in the questionnaire also followed the frequent patterns analyzed in subsection [**?**] following the Prefixing pattern, the Appending Pattern, Capitalizing pattern and so on.

5.1.7.1 Participant profiling

For each of the passwords we have decided to investigate the relationship between computer knowledge and the right view of password security.

University In this paragraph we want to study if there is a correlation between computer knowledge and the good use of passwords. We expect computer college graduates to be the most knowledgeable of how to choose good passwords.

First of all, in Figure 5.13 we show the university faculties attended by the participants.

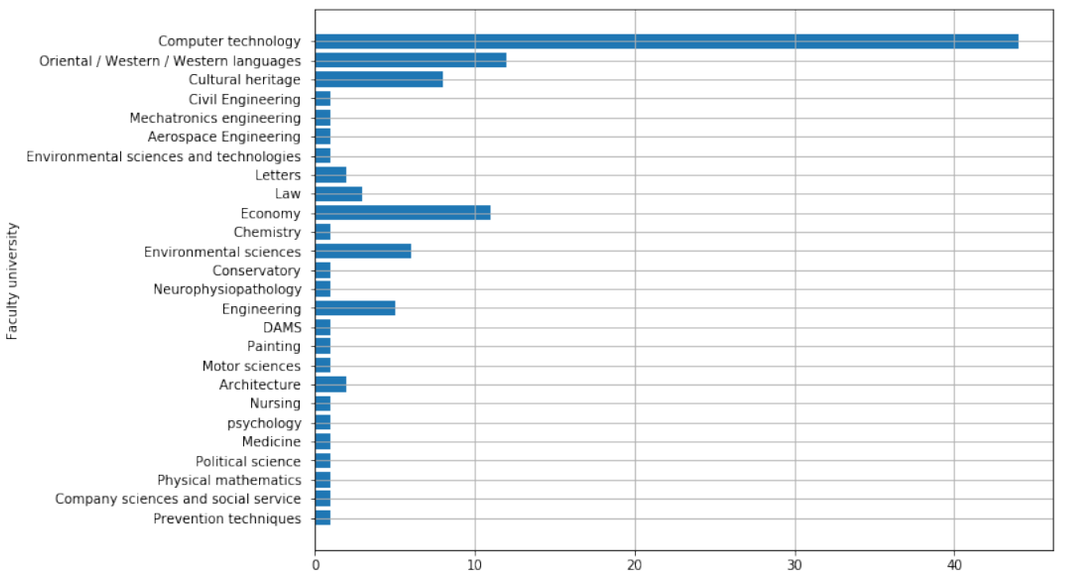

Figure 5.13: University Faculties

We have filtered, for each password, the faculties attended by those who have had a more correct perception of the strength between the two passwords submitted to them. Figure 5.14 shows the means of the correct answers of all comparisons.

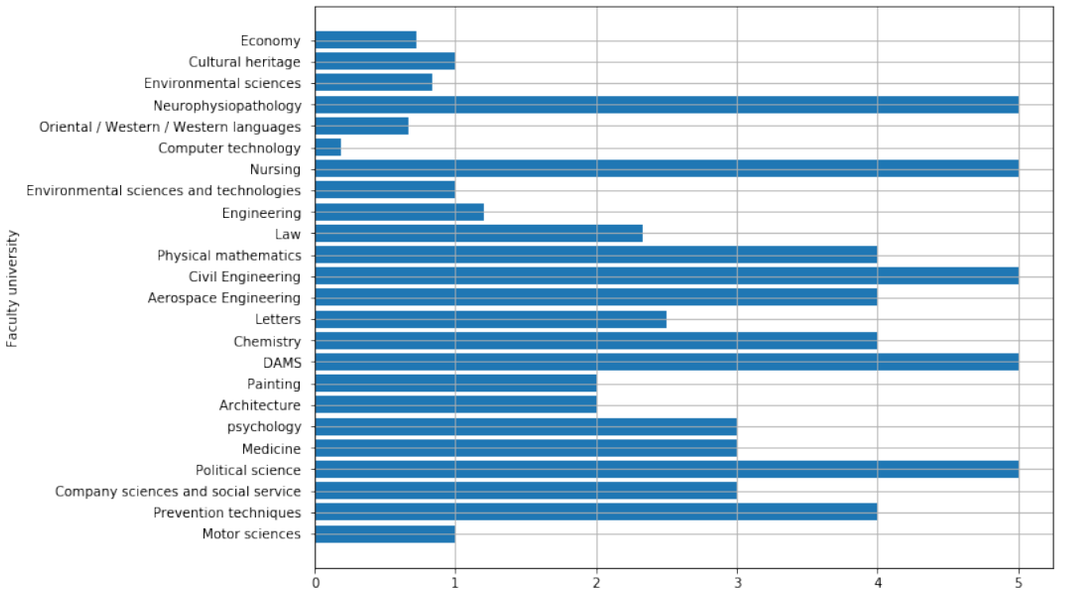

Figure 5.14: Faculties that answered in average correctly to the all the comparisons

The result is surprising since it highlights that those who study computer science have the highest percentage of wrong answers compared to all other faculties. We wondered if there was a reason why this event happened. Since the number of those studying computer science is much higher than the others this event is very curious. We tried to give a psychological response to this event and some research showed that when the participants were faced with a request for comparison, the so-called *attentional processes* were activated. There are different types of attention such as *Sustained Attention, Alternating Attention, Selective Attention, Focused Attention and Limited Attention.* What happens when a person is new to a field is that their attention is sky high and therefore pays attention to the smallest details. While, those who usually deal with issues such as passwords go into *"energy saving"* since *attentional processes* consume a lot of energy. This *"energetic saving"* leads, therefore, those who should be experts in the field to *focus* their *attention* only to the final goal: the registration to the site, forgetting the most important part which is the security.

High School

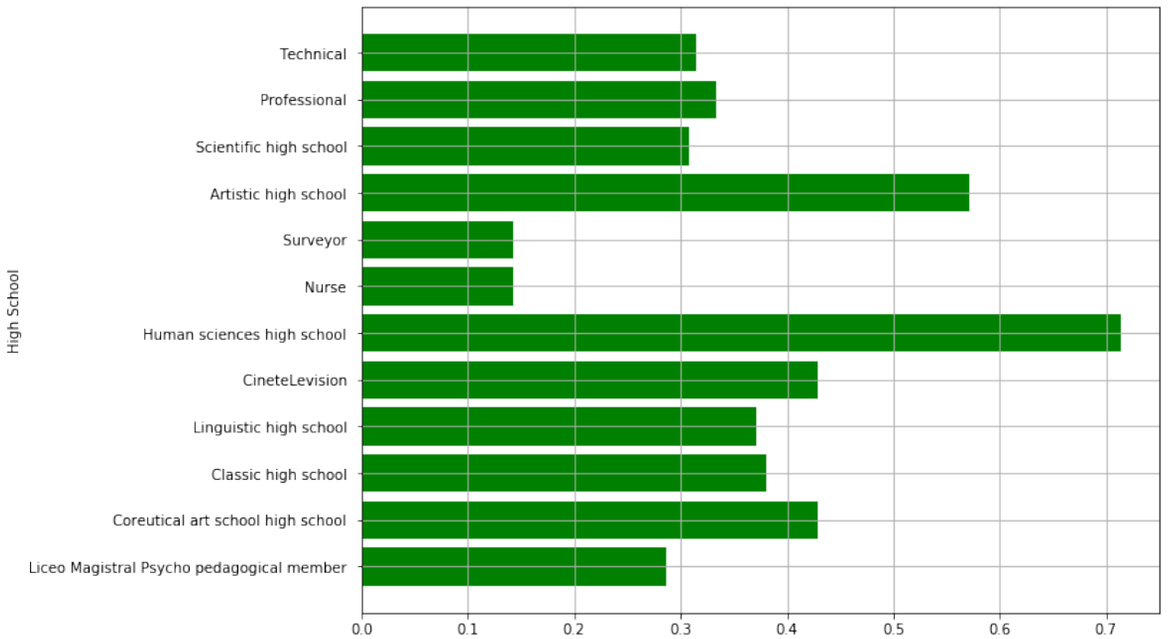

Figure 5.15: High school that answered in average correctly to the all the comparisons

We also extracted the high schools of the participants who answered the question of *maximum education level*, *high school* as show in fig. 5.15. As in the case of the university, those who attended the technical institute responded less correctly than other types of high schools. The reason is the same as mentioned in the paragraph on universities.

Conclusions In this section we have shown some user misconception and we have highlighted how IT knowledge does not directly correspond to a correct attitude towards IT security. Users understand quite a bit about the characteristics of strong and weak passwords, which should be leveraged to help users create stronger passwords.

Unluckily, users receive not too much advice on the overall password ecosystem [79]. Participants did have some critical misunderstandings. They severely overes-

timated the benefit of adding digits and symbols to passwords and underestimated the predictability of keyboard patterns and common phrases.
Our finding that participants mostly knew whether particular characteristics would make passwords easier or harder for attackers to guess may seem at odds with the widespread of poor passwords. This gap, however, may be the result of slack to help users understand the spectrum of attacks against passwords. In addition, similar findings were confirmed with respect to the analyzes conducted in Section 4, such as the most common patterns used writing down passwords.

We also had very similar results as the paper [65] from which we took inspiration for the questionnaire. This confirms what we found in Section 4 when we looked at whether changing the language and country would change the human behavior.

In the future it will be interesting to make the questionnaire continuous so as to be able to test the part of the participants' memory as well.

Conclusions

In this thesis it has been analyzed how the attitude of the human can harm the creation of good passwords. Human intended as a user but also as a developer since some security problems arise both from those who use the systems but also from those who create and maintain them. We have made an in-depth analysis of various data leaks in order to draw some common aspects which will be useful for the future developments. We have outlined the most commonly used patterns by also studying some social phenomena that have conditioned the knowledge of the correct use of passwords.

One of the problems analyzed was that of password strength. Since current password-strength meters in most cases calculate the entropy of a password and not the quality and only tell users if a password is weak or strong, not why.

Future work in this area, could be hot to auto-complete passwords and the development of better password meters based on the knowledge of the most common patterns and on the typical mistakes made by users. In the future we would like to develop a library based on the adaptive policies that pose particular attention to the usability of the system. So we would like to increase security without losing the usability for the users making the system easy to use and understandable to everyone.

The questionnaire we submitted to various users also made us understand why simple passwords are often chosen. In most cases, the fear of not remembering the password is the main reason. Since in large part, our results suggest that users are already aware of ways to make their passwords stronger, but they do not do so another future development could be to guide users, through some systems, in overcoming neutralization which it has been discussed a lot in the central part of the thesis.

A further development will be, using the information extracted in this thesis, the verify our findings using Hashcat which provides role-based attacks, and observe if it is possible to crack most of the datasets we are using, with the rules we created. Knowing these rules and their effectiveness will be a starting point for other possible developments described before.
We would like to further investigate the machine learning (ML) techniques proposed in the literature and run some data statistics using preexisting data leaks on existing passwords and all the information extracted with ML techniques to propose new machine learning approaches to improve existing ones trying to understand if further assumptions on password creation can be extrapolated. This is not a simple task since these learning techniques are black boxes so we do not have information on what they are learning from the training set.

Appendix

Table Question P(n) and Answers In this section we show for each question, mentioned during the chapter, the answers received in detail.

P(1) Do you often reuse the same password?

- Yes 66.40%;
- No 33.60%

P(2) When you create a password you think it must be

- Easy
  - Much 37.78%
  - Quite 42.39%
  - A little 13.36%
  - Not at all 6.45%
- Secure
  - Much 64.52%
  - Quite 31.33%
  - A little 3.22%
  - Not at all 0.92%

In this question we have left the possibility to enter your own answer. Answers entered autonomously by users are in italics,

P(3) If you think a password should also be easy to remember it's because..

- I'm afraid to forget it: 59.30%
- All sites ask me for the same pattern so I don't want to waste time thinking about a new password: 28.90%

- Doing "recover password" bothers me: 23.20%
- I have no imagination: 9.80%
- *Easy to remember only for passwords that are asked of me every day*: 0.50%
- When I think of a new password, one that I use often comes to mind: 20.10%
- *I don't want to use password management applications like 1Password even though I know it would be safer to use different passwords for each site.*: 0.50%
- *For convenience, because it can also be used when you are away from home and therefore unable to read it where you wrote it*: 0.50%
- *I can always write them but it's more fun to leave the same one or only in rare cases change it:* 0.50%
- *In reality I distinguish the passwords: important things all different and articulated; sites that I use very little and where I do not register the same cards:* 0.50%

P(4) What is the pattern you use the most when creating a password? Sort the positions taken by the various categories from 1 to 4. ES: "Cane12!" will answer the pattern: 1 → uppercase letter, 2 → lowercase letter, 3 → number, 4 → symbol

- Position 1:
  - Lowercase letter: 28.10%
  - Uppercase letter: 59.44%
  - Symbol: 10.13%
  - Number: 2.30%
- Position 2:
  - Lowercase letter: 54.83%
  - Uppercase letter: 27.65%

  - Symbol: 9.67%
  - Number: 7.84%

- Position 3:
  - Lowercase letter: 12.90%
  - Uppercase letter: 14.75%
  - Symbol: 29.95%
  - Number: 42.39%

- Position 4:
  - Lowercase letter: 11.04%
  - Uppercase letter: 6.45%
  - Symbol: 47.00%
  - Number: 35.48%

P(5) Do you ever enter [...] in your passwords?

- Date (day, month, year in any format): 51.12%
- Your age: 6.45%
- Pets names: 25.80%
- Names of bands / singers: 12.90%
- Eroticism (erotic words, allusion to erotic events)]: 5.99%
- Superhero names: 8.29%
- Colors: 15.66%
- Name of the son/daughter: 10.13%
- Name of relatives (grandparents, parents ...): 16.12%
- Proper name: 33%

- Random Numbers: 61.75%
- Surnames: 15.21%
- Film / TV series: 12%
- Email: 4.60%
- Nicknames: 36%
- Names of football players: 2.30%
- Slang: 18%
- Dates that remind you of events that are important to you: 55.29%

P(6) Given the following list, choose who, in your opinion, would be more inclined to steal one of your passwords

- Stranger: 65.40%
- Familiar: 24%
- Friend: 21.70%
- Colleague: 16.60%
- Other people I know: 21.20%

P(7) What do you think a malicious user does to try to guess your password?

- Use software: 73.30%
- Brute forcing : 29.50%
- Try popular words and names and known in my language: 33.20%
- Try common passwords: 41.90%
- Try dates and numbers: 38.70%

P(8) What do you think a malicious user does to try to guess your password?

- Financial reward: 44%
- Collect personal information: 73.60%
- Identity theft: 64.80%
- Fun / proof they can: 35.20%
- Spamming: 19.90%
- Espionage: 20.80%

P(10) Start with a word that comes to mind and then add digits or symbols at the end (ex: cane12!))

- Secure
  - Much 13.82%
  - Quite 41.01%
  - A little 36.40%
  - Not at all 8.76%
- Easy to remember
  - Much 52.99%
  - Quite 37.78%
  - A little 8.29%
  - Not at all 0.92%

P(10.1) Enter the name of one of your family members and their year of birth

- Secure
  - Much 2.76%

  - Quite 11.98%
  - A little 51.15%
  - Not at all 34.10%

- Easy to remember
  - Much 71.42%
  - Quite 22.58%
  - A little 4.15%
  - Not at all 1.84%

P(10.2) Use a date that is meaningful to you

- Secure
  - Much 9.22%
  - Quite 36.87%
  - A little 39.17%
  - Not at all 14.74%

- Easy to remember
  - Much 70.04%
  - Quite 26.26%
  - A little 1.84%
  - Not at all 1.84%

P(10.3) Base the password on the relationship between you and the account created (ex: lovemeetic for the meetic site))

- Secure
  - Much 3.68%

  - Quite 12.44%
  - A little 53.46%
  - Not at all 30.42%

- Easy to remember
  - Much 5.07%
  - Quite 32.26%
  - A little 12.90%
  - Not at all 3.68%

P(10.4) Use the same password you use for other accounts.

- Secure
  - Much 3.68%
  - Quite 20.27%
  - A little 54.84%
  - Not at all 21.19%

- Easy to remember
  - Much 82.94%
  - Quite 13.82%
  - A little 2.30%
  - Not at all 0.92%

P(10.5) Choose a strong password and write it down on a piece of paper or by writing it in your phone notes

- Secure
  - Much 34.10%

  - Quite 27.18%
  - A little 32.25%
  - Not at all 6.45%

- Easy to remember
  - Much 44.70%
  - Quite 24.88%
  - A little 19.81%
  - Not at all 10.59%

Acknowledgments

Thank you to my supervisors, Prof.ssa Luccio Flaminia and Prof. Focardi Riccardo, for providing guidance and feedback throughout this project.
I would like to thank those who love me, I love you too.
And my biggest thanks to my family for all the support you have shown me through my life.
In the end, thank you Dr. Marta Brocca, Psychologist, Piscotherapist and Specialist in Neuropsychological Cognitive Psychotherapy to give me valuable advice.

—"*The oldest and strongest emotion of mankind is fear.*"
— H.P. Lovecraft

Bibliography

[1] 3 tips to apply the cognitive dissonance theory in elearning. https://elearningindustry.com/apply-cognitive-dissonance-theory-elearning.

[2] 6 rules on how to create a secure password. https://www.bettertechtips.com/how-to/create-secure-password/.

[3] Americans' internet access: 2000-2015. shorturl.at/rwEQX.

[4] Ashley madison. https://www.ashleymadison.com/.

[5] Ashley madison dataset. https://github.com/danielmiessler/SecLists/blob/master/Passwords/Leaked-Databases/Ashley-Madison.txt.

[6] Benford. https://mathworld.wolfram.com/BenfordsLaw.html.

[7] Benford distribution plot. https://la.mathworks.com/matlabcentral/fileexchange/53355-benford-s-law-mini-toolbox.

[8] Brute-force password cracker. https://web.archive.org/web/20160302055156/http://www.oxid.it/ca_um/topics/brute-force_password_cracker.htm.

[9] Certmike. https://www.certmike.com.

[10] Cis. https://www.cisecurity.org.

[11] Cis password policy guide: Passphrases, monitoring, and more. https://www.cisecurity.org/.

[12] Definition of biometrics. https://searchsecurity.techtarget.com/definition/biometrics.

[13] Diceware list. DicewareList.

[14] Diceware list. https://github.com/NaturalLanguagePasswords/system.

[15] Diceware list. https://www.eff.org/files/2016/07/18/eff_large_wordlist.txt.

[16] Dos image. https://www.cloudflare.com/it-it/learning/ddos/what-is-a-ddos-attack/.

[17] Faux cyrillic. https://getcomputeractive.co.uk/protect-your-tech/fake-urls-with-cyrillic-letters.

[18] Find out if your password has been pwned—without sending it to a server. https://arstechnica.com/information-technology/.

[19] Form. https://forms.gle/US2KjY67vnyRBe8s9.

[20] Gates predicts death of the password. https://www.cnet.com/news/gates-predicts-death-of-the-password/.

[21] Github password targeted in password reuse attack. https://techcrunch.com/2016/06/16/github-accounts-targeted-in-password-reuse-attack/.

[22] Hashed passwords. https://www.dshield.org/diary/Hashing+Passwords/11110.

[23] Homographic. https://swimlane.com.

[24] Hotmail. https://www.msn.com/.

[25] Hotmail dataset. https://github.com/danielmiessler/SecLists/blob/master/Passwords/Leaked-Databases/hotmail.txt.

[26] Identification and verification image. https://www.bayometric.com/identification-verification-segmented-identification/.

[27] Kasperky. https://password.kaspersky.com/.

[28] Leet. https://www.cyberdefinitions.com/definitions/1337.html.

[29] Levenshtein distance. https://devopedia.org/levenshtein-distance.

[30] Misinformation. https://www.merriam-webster.com/dictionary/misinformation.

[31] Most used passwords. https://tinyurl.com/668y95vc.

[32] Munging passwords. https://th3s3cr3tag3nt.blogspot.com/2017/03/munging-passwords.html.

[33] Nist password guidelines 2021: Challenging traditional password management. https://pages.nist.gov/800-63-FAQ/.

[34] Packet sniffering image. https://www.dnsstuff.com/packet-sniffers.

[35] Pagejacking. https://www.techopedia.com/definition/15476/pagejacking.

[36] Pareto principle. http://www.gassner.co.il/pareto/.

[37] Passcode vs passkey. https://wikidiff.com/passcode/passkey.

[38] Password recovery methods. http://lastbit.com/password-recovery-methods.asp#Brute%20Force%20Attack.

[39] Password strength image. https://www.jqueryscript.net/.

[40] Password strength per bits. http://rumkin.com/tools/password/passchk.php.

[41] Passwords in the enterprise. https://www.balbix.com/app/uploads/State-of-Password-Use-Report.pdf.

[42] Passwords reveal your personality. https://www.psychologytoday.com/intl/articles/200201/passwords-reveal-your-personality.

[43] Phpbb. https://www.phpbb.com/.

[44] phpbb dataset. https://github.com/danielmiessler/SecLists/blob/master/Passwords/Leaked-Databases/phpbb.txt.

[45] Random password generator with algorithm. https://x-engineer.org/graduate-engineering/programming-languages/scilab/random-password-generator-with-algorithm/.

[46] Rockyou dataset. https://www.kaggle.com/wjburns/common-password-list-rockyoutxt.

[47] Russian credential theft shows why the password is dead. https://www.computerworld.com/article/2490980/russian-credential-theft-shows-why-the-password-is-dead.html.

[48] Salting and stretching a password. https://www.johndcook.com/blog/2019/01/25/salt-and-stretching/.

[49] The secret life of passwords. https://www.nytimes.com/2014/11/19/magazine/the-secret-life-of-passwords.html.

[50] Social engineering. https://usa.kaspersky.com/resource-center/definitions/what-is-social-engineering.

[51] Stemming. https://www.guru99.com/stemming-lemmatization-python-nltk.html.

[52] The online behavior that's putting you at risk. https://lp-cdn.lastpass.com/lporcamedia/document-library/lastpass/pdf/en/LastPass-B2C-Assets-Ebook.pdf.

[53] The ultimate guide for creating strong passwords. https://www.thegeekstuff.com/2008/06/the-ultimate-guide-for-creating-strong-passwords/.

[54] The united states of p@ssw0rd. https://storage.googleapis.com/gweb-uniblog-publish-prod/documents/PasswordCheckup-HarrisPoll-InfographicFINAL.pdf.

[55] What is a one-time password (otp)? https://www.okta.com/blog/2020/06/what-is-a-one-time-password-otp/#:~:text=A%20one%2Dtime%20password%20or,data%20or%20previous%20login%20events.

[56] What your passwords reveal about your personality. https://www.tesh.com/articles/what-your-passwords-reveal-about-your-personality/.

[57] Letters to the editor. *The American Statistician*, 26(3):62–65, 1972.

[58] Common keyboard symbols. *ThoughtCo*, 2021.

[59] L. Adamic and B. A. Huberman. Zipf's law and the internet. *Glottometrics*, 3(1):143–150, 2002.

[60] R. Anderson. Security engineering. a guide to building dependable distributed systems. 2008.

[61] S. Ansari, S.G. Rajeev, and H.S. Chandrashekar. Packet sniffing: a brief introduction. *IEEE Potentials*, 21(5):17–19, 2003.

[62] W. W. Banks and C. W. Charles. Human error: an overlooked but significant information security problem. *Computers Security*, 12(1):51–60, 1993.

[63] D. Besnard and D. Greathead. A cognitive approach to safe violations. *Cognition, Technology Work*, 5:272–282, 12 2003.

[64] J.V Beveren. A conceptual model of hacker development and motivation. *Journal of EBusiness*, 2000.

[65] Ur Blase, J. Bees, S. M. Segreti, L. Bauer, N. Christin, and L. F. Cranor. Do users' perceptions of password security match reality? In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 3748–3760, New York, NY, USA, 2016. Association for Computing Machinery.

[66] J. Bonneau, C. Herley, P. Oorschot, and F. Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. *IEEE Symp. on Security and Privacy*, pages 553–567, 05 2012.

[67] J. Bonneau and S. Preibusch. The password thicket: technical and market failures in human authentication on the web. 2010.

[68] J. W. Brehm and A. R. Cohen. Explorations in cognitive dissonance. *John Wiley Sons Inc*, 1962.

[69] A. Budi and B. Denis. Technical and human issues in computer-based systems security. 03 2003.

[70] M. Carnut and E. C. Hora. Improving the diceware memorable passphrase generation system. In *Proceedings of the 7th International Symposium on System and Information Security. São José dos Campos: CTA/ITA/IEC*, 2005.

[71] John M. Carroll. Human-computer interaction: Psychology as a science of design. *Annual Review of Psychology*, 48(1):61–83, 1997. PMID: 15012476.

[72] G. Cybenko, A. Giani, and P. Thompson. Cognitive hacking: A battle for the mind. computer. (35):50–56, 2002.

[73] A. C. David and J. Chalmers. The extended mind. *The Extended Mind, in Analysis*, 58(1):7–9.

[74] W. Ding, C Haibo, W. Ping, and J. Gaopeng. Zipf's law in passwords. *IEEE Transactions on Information Forensics and Security*, 12:2776–2791, 06 2017.

[75] D. Eastlake, J. Schiller, and S. Crocker. Randomness requirements for security. *RFC*, 4086:1–48, 2005.

[76] M. Fagan, Y. Albayram, and M.M.H. Khan. An investigation into users' considerations towards using password managers. *Hum. Cent. Comput. Inf. Sci*, 7:12, 2017.

[77] L. Festinger. Cognitive dissonance. *Scientific American*, pages 93–106, 1962.

[78] D. Florencio and C. Herley. A large-scale study of web password habits. page 657–666, 2007.

[79] D. Florêncio, C. Herley, and P. V. Oorschot. Password portfolios and the finite-effort user: Sustainably managing large numbers of accounts. In *USENIX Security Symposium*, 2014.

[80] J. Gonzalez, , and A. Sawicka. A framework for human factors in information security. 05 2021.

[81] C. Herley and P. Oorschot. A research agenda acknowledging the persistence of passwords. *IEEE Security Privacy*, 10:28–36, 05 2012.

[82] G. Hinson. Social engineering techniques, risks, and controls. *EDPACS*, 37(4-5):32–46, 2008.

[83] B. Hitaj, P. Gasti, G. Ateniese, and F. Perez-Cruz. Passgan: A deep learning approach for password guessing. pages 217–237, 05 2019.

[84] K. Iwasa and T. Ogawa. Psychological basis of the relationship between the rorschach texture response and adult attachment: The mediational role of the accessibility of tactile knowledge. *Journal of personality assessment*, 98:1–9, 11 2015.

[85] D. M. Kevin and L. S. William. The art of deception: Controlling the human element of security. *ISBN: 978-0-471-23712-9*, 2002.

[86] P. Kintis, N. Miramirkhani, C. Lever, Y. Chen, R. Romero-Gómez, N. Pitropakis, and M.Antonakakis N.Nikiforakis. Hiding in plain sight. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, Oct 2017.

[87] D. Klein. "foiling the cracker": Asurvey of, and improvements to, password security. 06 1992.

[88] M. Ko, K. Osei-Bryson, and A. Dorantes. Investigating the impact of publicly announced information security breaches on three performance indicators of the breached firms. *IRMJ*, 22:1–21, 04 2009.

[89] S. Kraemer and P. Carayon. Human errors and violations in computer and information security: The viewpoint of network administrators and security specialists. *Applied ergonomics*, 38:143–54, 04 2007.

[90] Y. Lafrance. Psychology: A precious security tool. 2004.

[91] V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics. Doklady*, 10:707–710, 1965.

[92] M. Libicki. The mesh and the net: Speculations on armed conflict in a time of free silicon. *National Defense University Press, Washington, D.C.*, 1994.

[93] J. Markus. The human factor in phishing. *Privacy Security of Consumer Information*, 01 2007.

[94] J.L. Massey. Guessing and entropy. In *Proceedings of 1994 IEEE International Symposium on Information Theory*, pages 204–, 1994.

[95] B. Merdenyan and P. Helen. Perceptions of the risks of password related activities. pages 1–10, 07 2017.

[96] C. Merrick, F. Melika, T. K. Jantz, A. Gazzaley, and E. Morsella. External control of the stream of consciousness: Stimulus-based effects on involuntary thought sequences. *Consciousness and Cognition*, 33:217–225, 2015.

[97] G. Navarro. A guided tour to approximate string matching. *ACM Comput. Surv.*, 33(1):31–88, March 2001.

[98] M. Newman. Power laws, pareto distributions and zipf's law. *Contemporary Physics*, 29:323–351, 2005.

[99] F. Di Nocera. Ergonomia cognitiva. *Dimensioni della psicologia.*, 2011.

[100] D. A. Norman. Design principles for cognitive artifacts. 1992.

[101] K. Oberauer. Working memory and attention - a conceptual analysis and review. 08 2019.

[102] D. Pasquini, A. Gangwal, A. Ankit, G. Ateniese, and M. Bernaschi. Improving password guessing via representation learning. 01 2021.

[103] J. Payne. Debunking graphical password myths. *9th International Conference on Passwords (Passwords15 London)*, 2010.

[104] S. Pearman, J. Thomas, N. Emami, H. Pardis, L. Bauer, N. Christin, L. Cranor, S. Egelman, and A. Forget. Let's go in for a closer look: Observing passwords in their natural habitat. pages 295–310, 10 2017.

[105] J. Perner. Learning, development, and conceptual change.understanding the representational mind. *The MIT Press.*, 1991.

[106] S. Porter. A password extension for improved human factors. *Computers Security - COMPSEC*, 1:81, 01 1981.

[107] J. Reason. *Human Error.* Cambridge University Press, 1990.

[108] J. Reason. *Managing the Risks of Organizational Accidents.* Routledge, 1997.

[109] S. Reiss. Multifaceted nature of intrinsic motivation: The theory of 16 basic desires. *Review of General Psychology*, (8):179–193.

[110] M. Needham Roger. Denial of service. page 151–153, 1993.

[111] M. K. Rogers, K. Seigfried, and K. Tidke. Self-reported computer criminal behavior: A psychological analysis. *Digital Investigation*, 3:116–120, 2006. The Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06).

[112] A. Sasse, P. Lunt, and Adams A. Making passwords secure and usable. 08 1997.

[113] H. Schmid. *Cognitive Pragmatics*. De Gruyter Mouton, 1 2012.

[114] B. Schneier. *Applied Cryptography*.

[115] B. Schneier. Semantic attacks: The third wave of network attacks. *Crypto- Gram Newsletter*, (14), 2000.

[116] M. Siponen, P. Puhakainen, and A. V. Petri. Can individuals' neutralization techniques be overcome? a field experiment on password policy. *Computers Security*, 88:101617, 09 2019.

[117] G. M. Sykes and D. Matza. Techniques of neutralization: A theory of delinquency. *American Sociological Review*, 22(6):664–670, 1957.

[118] E. Tatlı. Cracking more password hashes with patterns. *IEEE Transactions on Information Forensics and Security*, 10:1–1, 08 2015.

[119] M. Uma and P. Ganapathi. A survey on various cyber attacks and their classification. *International Journal of Network Security*, 15:390–396, 01 2013.

[120] J.R. Vacca. *Computer and Information Security Handbook*. 01 2013.

[121] R. Veras, J. Thorpe, and C. Collins. Visualizing semantics in passwords: the role of dates. In *VizSec '12*, 2012.

[122] M. Wanli, j. Campbell, D. Tran, and D. Kleeman. Password entropy and password quality. In *2010 Fourth International Conference on Network and System Security*, pages 583–587, 2010.

[123] R.W. White. Motivation reconsidered: The concept of competence. *Psychological Review*, (66):297–333, 1959.

[124] Wikipedia contributors. Bitsquatting — Wikipedia, the free encyclopedia, 2021.

[125] Wikipedia contributors. Typosquatting — Wikipedia, the free encyclopedia, 2021.

[126] D. Wood and G. Ross. The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry*, 17:89 – 100, 12 2006.

[127] J. Yan. A note on proactive password checking. *Proceedings New Security Paradigms Workshop*, 05 2003.

[128] M. Yıldırım and I. Mackie. Encouraging users to improve password security and memorability. *International Journal of Information Security*, 18, 12 2019.

[129] Y. Zhang, F. Monrose, and M. Reiter. The security of modern password expiration: An algorithmic framework and empirical analysis. *Proceedings of the ACM Conference on Computer and Communications Security*, pages 176–186, 01 2010.

[130] G. Zipf. Human behavior and the principle of least effort. 1949.

[131] M. Zviran and W. J. Haga. Cognitive passwords: The key to easy access control. *Computers Security*, 9(8):723–736, 1990.