A Spark is Enough in a Straw World: a Study of Websites Password Management in the Wild

Simone Raponi and Roberto Di Pietro

Hamad Bin Khalifa University,
College of Science and Engineering,
Doha, Qatar
{sraponi,rdipietro}@hbku.edu.qa

Abstract. The widespread usage of password authentication in online websites leads to an ever-increasing concern, especially when considering the possibility for an attacker to recover the user password by leveraging the loopholes in the password recovery mechanisms. Indeed, the adoption of a poor password management system by a website makes useless even the most robust password chosen by its users. In this paper, we first provide a survey of currently adopted password recovery mechanisms. Later, we model an attacker with a set of different capabilities, and we show how current password recovery mechanisms can be exploited in our attacker model. Then, we provide a thorough analysis of the password management of some of the Alexa's top 200 websites in different countries, including England, France, Germany, Spain and Italy. Of these 1,000 websites, 722 do not require authentication—and hence are excluded from our study—, while out of the remaining 278 we focused on 174—since 104 demanded information we could not produce. Of these 174, almost 25% have critical vulnerabilities, while 44% have some form of vulnerability. Finally, we point out that, by considering the entry into force of the General Data Protection Regulation (GDPR) in May, 2018, most of websites are not compliant with the legislation and may incur in heavy fines. This study, other than being important on its own since it highlights some severe current vulnerabilities and proposes corresponding remedies, has the potential to have a relevant impact on the EU industrial ecosystem.

Keywords: Authentication mechanism · Password recovery · Security.

1 Introduction

The countless attacks on websites in recent years have underlined once again that security is often considered a feature, rather than a necessity [15]. The victims of such attacks are the users that, unaware of the management of the provided confidential information, will find their identity and data compromised on the web. In fact, even if a user would adopt all known good practices to choose a really strong password to enforce access control, a single website that stores information in an insecure way would be enough to compromise (at least) confidentiality of the provided data. Examples are the data breaches suffered by both the professional networking site LinkedIn [2] and the Internet service company Yahoo [5].

LinkedIn suffered a major breach in June 2012, when 6.5 million encrypted passwords were posted on a Russian website. Things got much worse in May 2016, when 117 million LinkedIn credentials (i.e., combination of e-mails and passwords) were posted for sale on the Dark Web [1]. Although LinkedIn was a well-known and well-established website, poor cryptography techniques were implemented [23], by making easier for hackers to decrypt users' passwords. The attack targeting Yahoo was the biggest one in history, which led to the breach of 3 billion passwords. The hack began with a spear-phishing e-mail sent to a Yahoo employee, that eventually led to the acquisition of the entire users database by the attacker (containing names, phone numbers, password challenge questions and answers, password recovery e-mails and a cryptographic value unique to each account) [8]. If it were true that history taught, we should have solved (or at least mitigated) problems of this kind, but reality is different. In fact, in this work we point out that almost 44% of the top Alexa's websites we considered (out of the top 200 of respectively England, France, Germany, Italy, and Spain, shown in Table 3) do store users password in a form that can be easily exploited. While the impact of these findings are relevant on their own, their consequences for companies are magnified when taking into account the GDPR—the poor security measures adopted by the web sites are a clear violation of.

The General Data Protection Regulation (GDPR) [6] leads to an important change in the vision of both data privacy and data security. Born as an evolution of the previous Data Protection Directive, adopted in 1995, it became enforceable since May 25th, 2018. With its entry into force, all the websites, agencies, enterprises, organizations, that make use of the personal data of the users have to guarantee their protection both by design and default in any operation. In case of non-compliance, these entities are subject to very heavy penalties ranging from 10-20 million euros to 2-4% of the annual worldwide turnover of the previous financial year, unsustainable by most organizations.

Contributions. In this paper we first provide a survey of both users authentication mechanisms implemented by websites and related password recovery mechanisms. Then we model a realistic attacker with different capabilities, respectively Mail Service Provider attacker, Web Server Intruder attacker, Client Intruder attacker, and Sniffing attacker. Later we provide a thorough analysis for users password management Alexa's top 200 websites of the five aforementioned European countries. Then, we study in detail which information could be obtained by our modeled attacker and we show how she can break the access control mechanisms.

Results are striking; of the 174 analyzed websites (see Table 3) almost 25% of the websites do have from poor to very poor password management, and an overall of 43.68% are vulnerable to at least one of the presented attacks—note that all the attacks happen because of the non-compliance of the websites to the GDPR prescriptions, hence having the corresponding organizations being subject to the cited fines.

Road-map: In Section 2, we report on the related work in the literature. In Section 3, we provide a technical background of both user authentication on websites and the password recovery mechanisms. In Section 4, we define our attacker model.

In Section 5, we describe the methodology we adopted and we present the results of the analysis, while in section 6 we report some concluding remarks.

2 Related Work

LinkedIn data breaches in June 2012 [2] and Yahoo data breaches in August 2013 [5], made respectively 6.5 million and 3 billion users accounts compromised, but these compromises are only the tip of the iceberg. However, these attacks have not affected the popularity of passwords. In fact, passwords remain the most widespread authentication mechanism on the web. The use of a password introduces a secret that is shared by only the authenticator (the website), and the user wishing to be authenticated.
From the moment they were adopted, a number of scientific articles were published with the aim of highlighting their weaknesses and vulnerabilities. In [18] the authors analyzed half a million Windows Live users' passwords, pointing out that a user has 6.5 passwords shared across 3.9 different sites on average. Furthermore, each user is the owner of about 25 accounts and types an average of 8 passwords per day. Most of the picked passwords are extremely weak, in fact, if not forced, users choose passwords composed by solely lowercase letters. Dell'Amico et al., in [17] focused on the empirical study of real-world passwords. They implemented and used several state-of-the-art techniques for password guessing to analyze the password strength of Internet application. They found that users put relatively little effort in choosing their password when compared to the choice of their usernames. The human component plays a fundamental role in both the security and the robustness of the authentication mechanisms. Indeed, even the most advanced system would be compromised if users pay little attention to their password choice.
Passwords authentication mechanisms will still be used for years, as "something you know" mechanisms are extremely less expensive (but also less secure) than both "something you have" and "something you are"—these latter ones being prone also to false positive and false negative. By considering this, it is of fundamental importance to guarantee both secure access and secure storage, as well as secure mechanism to retrieve the password in case of forgetfulness or theft. In [19], the authors presented an assessment of password practices on 10 popular websites, including Facebook, Amazon, Yahoo, Google, and YouTube. They examined password selection, the restrictions enforced on password choice, and the recovery/reset of the password if forgotten. They pointed out that no website provides adequate coverage of all the criteria taken into account. The result is worrying, as the websites analyzed are among the most visited on the web and therefore (should be) the safest.
When we lose or forget our passwords we generally use our e-mail address to retrieve them, assuming that we are the only ones who have the access. S.L. Garfinkel, in [21], wondered if an e-mail-based Identification and Authentication (EBIA) method could substitute the Public Key Infrastructure (PKI). The EBIA technique considers an e-mail address as a universal identifier and the ability to receive an e-mail at that address as a kind of authenticator. Among the many limitations, the author pointed out two many vulnerabilities: EBIA security strongly

depends on the security of e-mail servers and password; e-mail content is accessible to server operators (without encryption, system managers can intercept, read, and make copies of e-mail messages intended for end users). Although the paper is dated back 2003, EBIA is the main used secondary (also known as emergency) authentication method nowadays, with the same vulnerabilities (and a few additional ones) left unresolved. One possible countermeasure would be to use secure e-mail services, such as ProtonMail [10]. ProtonMail is an open-source end-to-end encrypted e-mail service founded in 2014 at the CERN. This service allows to protect the user's privacy and anonymity by not logging IP addresses which can be linked to the account, furthermore the end-to-end encryption makes unreadable the e-mail content even to the mail provider.
In recent years, various alternatives were proposed to manage the secondary authentication mechanism in a safer way. After "something you have", "something you are", and "something you know", in [16] the authors explored a fourth-factor authentication: "somebody you know". They focused in a process called vouching, a peer-level authentication in which a user (called helper), leverages her primary authenticator to assist a second user (called asker) to perform secondary authentication. They designed a prototype vouching system for SecurID, a hardware authentication token, to allow an helper to grant temporary access privileges to an asker who has lost the ability to use her own. Although interesting, this method requires a primary authentication mechanism based on token, hence "something you have", which is not implemented by most web sites. In [27] Schechter et al. exploited the social-authentication to let users who have forgotten their passwords to regain access to their account. The proposed system employs trustees, users previously appointed by account holders to verify their identity. To get the access into her account, the account holder contacts their trustees in person or by phone, so that their trustees may recognize her by either her appearance or her voice. Once the recognition has occurred, the trustees provide the account holder with an account recovery code, that will be necessary to authenticate her in the system. This mechanism is safer with respect to the ones currently adopted but may have usability problems. Trustees may not be available at the time of request, making it impossible to the requesting user to recover her account.
The secondary authentication mechanisms are discussed and analyzed in [26]. In this work the authors considered four key criteria (i.e., reliability, security, authentication, and setup efficiency) to evaluate several secondary authentication mechanisms, such as security questions, printed shared secrets, previously used passwords, e-mail-based verification, phones or other services, trustees, and in-person proofing. Although they provided a thorough analysis of these mechanisms, no attackers were modeled and no experimental website security analysis was performed.

3 Technical Background

In this section, a technical background of both user authentication mechanisms on websites and password recovery mechanisms adopted by websites are provided.

3.1 User authentication on websites

With user authentication on websites we refer to the process by which the credentials provided by the user are compared to those stored either in websites database or in a cloud server. If the credentials match, the authentication process is completed and the requesting user is granted authorization to access. According to the website policy, a few user authentication methods can be implemented [20]. A combination of these methods leads to a more accurate identification of the user:

– **1FA**: the 1-Factor-Authentication (1FA) method requires only one factor to authenticate a user. Usually it takes into account "something the user knows" (e.g., a password or a PIN code). 1FA is the authentication mechanism most commonly adopted by the current websites;
– **2FA**: the 2-Factor-Authentication (2FA) method requires two factors to authenticate a user. It takes into account both "something the user knows" and either "something the user has" (e.g., physical token or a smart card) or "something the user is" (e.g., fingerprints, retinal or iris scans, voice recognition, hand geometry). This mechanism is adopted in most of the sensitive websites (e.g., bank websites) and is optional in others (e.g., Gmail [13], ProtonMail [11]);
– **3FA**: the 3-Factor-Authentication (3FA) method requires a user being authenticated with "something she knows" as well as with both "something she has" and "something she is". This mechanism has not been adopted by the websites, probably due to the infrastructure costs that would derive from it.

3.2 Password recovery

Password recovery is a mechanism implemented by websites that allows to recover the user's secret password in case the password is lost or forgotten. During the years, numerous mechanisms have been proposed to recover the password, the most frequently adopted are reported below.

Security Questions. When the account holder loses or forgets her password, the password recovery mechanism starts up and provides the user with security questions. This mechanism is based on the assumption that only the account holder is able to answer correctly. Some websites make use of a set of pre-packaged standard questions, while others allow the users to choose their own. In each case, several work demonstrated both limits and vulnerabilities of this mechanism. Using a set of pre-packaged password is quite insecure in the era of the information; answering questions like "what is the name of your primary school?" or "what is your favorite movie?" becomes trivial by having access to all the personal information the user shares on social networks. On the contrary, users could choose their own customized security questions. Even in this case, several work [22,26] demonstrated how weak the mechanism is. As they mention, users should select questions that are memorable, not researchable on-line, reasonably unpopular with other users, and unknown by any untrusted acquaintances.

Previously used passwords. Websites, as a password recovery mechanism, could ask the user to enter one (or a set of) previously used password(s). However, users tend to use a limited number of passwords for the web services they access, this phenomenon leading to the unavoidable use of the same password for one or more services. Given that, it should be easy for users to remember one of the password previously used—with high probability it will be a password they are currently using for another web service. Although with both some limitations (e.g., the user may not remember any of the previous passwords) and many security problems (e.g., the attacker may know some of the previous passwords of the victim) the adoption of this method would be less dangerous than the ones currently implemented. It is worth noting that this method cannot be applied if the password has been lost or stolen for the first time.

E-mail-based authentication. The most common password recovery mechanism is the e-mail-based authentication, which relies on the assumption that only the account holder will be able to access a secret sent to her e-mail account [21, 26]. As we will discuss in Section 5, websites can provide the requesting user with her credential information in several ways:

– sending the old password by e-mail;
– sending a new password by e-mail (temporary or not);
– sending an HTTP link by e-mail (to choose a new password); or
– sending an HTTPS link by e-mail (to choose a new password).

The vulnerabilities of this mechanism are detailed in Section 5.

Other password recovery mechanisms. Many alternatives have been proposed to manage the password recovery mechanisms in a safer way. Mobiles can be used for user authentication: the website can send users an SMS message or use an automated voice system to call them and provide an account recovery code. As for e-mail-based authentication, this alternative relies on the fact that only the device holder will be able to access a secret that has been sent to the device [26]. Brainard et al., in [16], rely on *trustees*, users previously appointed by the account holder, necessary to verify her identity. These people will be contacted in case of emergency (the password is either lost or compromised) and they will be asked to perform the recognition of the requesting user. If the recognition phase is successful they will provide the requesting user with a recovery code, useful to authenticate her on the website.

4 Adversary Model

Given the websites ecosystem, we consider several categories of adversaries. An adversary may be *undetectable* or *detectable*. An undetectable adversary is able to impersonate the user on websites without the victim being aware of it. Conversely, a detectable adversary impersonating the user has a chance of getting the user being aware, or a least suspicious, of the fact that an impersonation happened (or could have happened). An adversary may also be *active* or *passive*. An active attacker interacts with websites in order to get information about the victim (e.g., she can start the recovery procedure on behalf of the victim), while a passive attacker aims to obtain the target user's sensitive information without any interaction with the website. Eventually, both active and passive attackers can use the obtained information to access websites by pretending to be the victim. Remembering these distinctions (summarized in Table 1), we introduce four different possible attacks against a single target user, taking into account a single target website:

- Mail service provider-level attack;
- Web server intruder attack;
- Client intruder attack;
- Sniffing attack.

Table 1. Attackers Types

Type	**Detectable**	**Undetectable**
Active	Can interact with the website on behalf of the victim by revealing her existence	Can interact with the website on behalf of the victim by remaining transparent
Passive	Cannot interact with the website, but her actions will eventually make the victim aware of her existence	Cannot interact with the website and will remain transparent

Table 2. Attackers Capabilities

Attackers/Access	**User e-mails**	**Website password DB**	**Website password recovery method**
Mail service provider attacker	✓	✗	✓
Web server intruder attacker	✗	✓	✓
Client intruder attacker	✓	✗	✓
Sniffing attacker	✗	✗	✓

A *mail service provider-level attack* can be undertaken by both a malicious service provider, or a malicious user that has compromised the mail service provider. This is the most dangerous attack, as the attacker has access to all the user's e-mails and can obtain a lot of sensitive information. In the *web server intruder attack*, we suppose the attacker was able to violate a website where the user has registered an account. So, the adversary can interact with the database that stores all the passwords of the users. A *client intruder attack* can be undertaken by an adversary that either has violated a user's device by obtaining a remote access, or has stolen it. In the *sniffing attack* we suppose that the attacker has no knowledge about the user, she has neither access to the website's password database nor to the user's devices. The only information she has, is about the password recovery methods of the website (to obtain this information she could register an account and start the recovery procedure). The sniffing adversary can sniff the packets transmitted during the communication between the client and the website. We assume without loss of generality that the sniffing attacker is not able to read the content of the exchanged e-mails (for instance, because the HTTPS protocol is used). The capabilities of attackers are summarized in Table 2.

5 Methodology and Results

In this section we provide the methodology employed to analyze the websites' password management security. The methodology involves two independent choices:

1. which website to take into account for the analysis; and,
2. how to analyze the websites.

As for the first choice, we decided to make use of the Amazon Alexa Top Sites web service [3]. Amazon Alexa Top Sites is a web service that provides a list of websites, ordered by Alexa Traffic Rank. This ranking is determined by a combined measure of both websites' unique visitors and websites' page views, and is updated daily [7]. So, we selected a subset of countries, respectively England, France, Germany, Italy, and Spain according to their subjectiveness to the GDPR regulation—till Brexit happens, UK is subject to GDPR as well. For each of these countries, we considered the first top 200 websites according to Alexa's ranking. The choice of this parameter is due to the fact that as the ranking goes down, the websites become less and less used–hence, with a reduced impact. We have therefore created an account and started obtaining the websites' URLs we needed. Amazon Alexa provides several ways for obtaining websites information; in particular, the top websites can be selected according to three major divisions: global; by country; and by category. The global division, as the name suggests, shows the most visited websites globally. The division by country, instead, allows to view the most visited websites from specific countries, not necessarily with the domain registered in the country taken into account (e.g., the website *youtube.com* is in the top five of all the countries considered in this analysis, but its domain is registered in the US). The third division is the more generic and allows to select the top 200 websites according to different sub-categories (e.g., adult, arts, computers, recreation, regional, sports, and so on). By selecting the regional category, we were able to choose the continent first and then the specific nations we would like to consider. In this case, Alexa shows the top 200 websites for the countries selected, that represents the top 200 websites with the domains registered in those countries. Once obtained the websites URLs, our goal was to analyze them in order to get information about the password storage management. In the second phase we started by selecting, among the top 200 websites for each nation, the ones that required a user registration.

A further filtering step was to remove those websites that, in the user registration phase, required privileged information (e.g., id number for universities, account number for banks, customer code for wholesalers). In detail, of these 1,000 websites, 722 did not require authentication—and hence were excluded by our study—, while out of the remaining 278, we focused on 174—since 104 demanded information we could not produce. The results of this filtering are shown in Table 3.

Table 3. Number of analyzed websites (among the top 200 per country)

Country	**Websites (#)**
England	71
France	31
Germany	19
Italy	36
Spain	17
Total	**174**

By considering the number of websites, we decided to manually perform the analysis of each one, in order to obtain more accurate and detailed information—while this activity took quite some amount of time (around three months), the quality of results is unpaired. In particular, for each website:

- we registered an user account;
- we pretended to have lost the password and we started the recovery procedure; and,
- we collected password recovery information.

Table 4. Websites password recovery mechanisms

Country	**Websites (#)**	**Old Pw**	**New Pw**	**Temp Pw**	**HTTP link**	**% of vulnerable websites**
England	71	1	5	2	16	33.8
France	31	0	10	0	7	54.84
Germany	19	1	1	0	5	36.84
Italy	36	4	11	1	4	55.55
Spain	17	2	5	0	1	47.06
Total	**174**	**8**	**32**	**3**	**33**	**43.68**

We registered on the websites by using a Gmail account created for the experiment, i.e., *gdpr.experiment@gmail.com*. The results of the analysis are shown in Table 4. Each column of the table represents a different method of password recovery adopted by the analyzed websites—these password recovery methods can be observed from left to right in decreasing order of vulnerability.

5.1 Password recovery methods

In the following, we describe each password recovery method adopted by the analyzed websites.

Old Password. The most vulnerable websites are the ones that use the *Old Pw* method for the password recovery phase. In detail, after having completed the "forgotten password" procedure, the website sends the original password of the user in her registration e-mail address. Hence, it appears that websites do not make use of hash functions or other mechanisms in order to avoid to store the passwords in cleartext. Considering that most of the users make use of the same password to access many different services [12], obtaining access to the database of password stored by the website (by an hacker attack or an internal website error) would seriously jeopardize both the security and the privacy of all the registered users. This method of storing password is deprecated since at least 30 years [25].

New Password. In this recovery method, after the "forgotten password" procedure, the website sends a new password to the registration e-mail address of the user, without obliging her to change the password after the first access. The security of passwords sent by using this method is summarized in Table 5. In this analysis we take a lenient stance and consider respectively: *strong* the passwords with at least 2^{70} possible combinations; *medium* the passwords with at least 2^{50} possible combinations; and *weak* all the others. We have obtained this data by requesting a new password 125 times, and by analyzing the type of provided password. For instance, *VUZUK8G3*, *484ad5b5*, *ugrxpn* are three of the passwords provided by a given website following the recovery procedure. It is possible to see that all the passwords use only uppercase or lowercase letters, sometimes numbers, never special characters, with an overall length of 8. In this case, the strength has been computed as $62^8 \approx 2^{48}$.

Table 5. Password robustness analysis

	$> 2^{10}$	$> 2^{20}$	$> 2^{30}$	$> 2^{40}$	$> 2^{50}$	$> 2^{60}$	$> 2^{70}$
New	100%	93.75%	75%	62.5%	3.125%	3.125%	3.125%
Temporary	100%	100%	100%	100%	33.33%	0	0

Note that only 3.125% of the new passwords provided to the users by the website have what could be considered a decent level of security, while more than 90% are considered weak. This password choice makes the websites vulnerable to brute-force attacks, where malicious users can try to guess the users' password after requesting the re-sending on behalf of the victims.

Temporary Password. The *Temp Pw* method consists in sending a temporary password to the requesting user, where this password must be changed on her next access. Only few websites make use of this recovery method, and those who do, they send either weak or medium security password to the requester e-mail account (see Table 5).

HTTP link. Almost 19% of the websites use the *HTTP link* as recovery method. In this case, once the user has requested for the password recovery, the web service sends an HTTP link to her registration e-mail address. By clicking on the link, the requester is redirected to the website on which she can enter a new password that will be associated with her account. We consider vulnerable the websites that make use of this recovery method. Indeed it is well-known that the HTTP protocol does not provide insurance with respect to attacks such as man in the middle [9], or even simple snooping. Considering that all communications between user browser and websites are not encrypted, a malicious user can intercept the message exchange, eavesdrop and modify the communication, compromising both its confidentiality and integrity.

HTTPS link. The last analyzed recovery method of the websites is the *HTTPS link* one, in which the link sent to the user to let her choose a new password is based on HTTPS protocol. This method is safer with respect to the others but it is subject to being exploited as well, as we will detail in the following.

5.2 Attackers capabilities

In this section we describe the capabilities of attackers as well as their characteristics. Results of this study are summarized in Table 6 for passive attackers, and in Table 7 for active attackers.

Mail service provider-level attacker. A *passive mail service provider-level attacker* may obtain the user's log-in information in any case, regardless of the password recovery methods adopted by the website. In fact, this kind of attacker has access to the e-mails of the victim, the emergency authentication mechanisms currently adopted by websites. In case of *Old Pw* or *New Pw* recovery methods, the attacker can even remain undetectable. In fact, by easily reading the password inside the e-mail, the attacker and the victim would share the same account unbeknownst to the latter. The attacker is forced to adopt a detectable method in the other cases (i.e., the adversary can use either the temporary password or both the HTTP and HTTPS link to log-in with the credentials of the victim, but once logged in she is forced to change the password, by no longer granting access to the victim). In this situation, the provider could obtain the credentials (by reading

either the new or the temporary password) as well as the links (either HTTP or HTTPS) and destroy the received e-mail. The user would not receive the e-mail but could attribute the fact to a website malfunction. In the meantime, the adversary may have entered, obtained the information she needed, and logged out. There is a good chance of not being suspected at all.
An *active mail service provider-level attacker* could remain undetectable if the target website stores the passwords of registered users in clear. In fact, by starting the recovery procedure, she can obtain the original password of the user by e-mail. This e-mail will be deleted from the system as soon as possible. In all other cases, the password would change and the mail service provider-level attacker will delete the compromising e-mail. The victim would have no longer access to the website but could attribute the fact to either a website malfunction or to having forgotten the password.

Web server intruder attacker. The information that a *passive web server intruder attacker* can gain is strongly dependent on the password storage management of the website she has violated. Indeed, if the passwords of the registered users are stored in clear text (i.e., without the support provided by hash functions), the attacker could transparently make use of the credentials of the victim to access the target website.
The capability does not change from the perspective of an *active web server intruder attacker*, because they would still depend to the storage methods. In addition, note that the victim would be aware of an attack as it would receive e-mails with instructions for recovering the password.

Client intruder attacker. A *passive client intruder attacker*, with remote or physical access to the device of the victim, would have most of the time exactly the same capabilities of an mail service provider-level attacker (i.e., there is often no need to insert a password to access services, most of users allow the browser to remember it [28]). Furthermore, either remote or physical access to the device would allow the attacker to find any password file written by the victim [29], so the attacker capabilities depend on the victim's behavior. If the victim does not allow the browser to remember the information about her credentials, an attacker with remote or temporarily physical access to the device could install a key-logger software [24], that would capture user input and provides the attacker the information about the keys typed (including the passwords of the websites that the victim will visit in that session).
An *active client intruder attacker* would have the same capabilities of the active mail service provider-level attacker, because of the same access to the e-mail account of the victim.

Sniffing attacker. A *passive sniffing attacker* can obtain the access information of a target user mainly in two ways: she can intercept the HTTP communications, as well as undertake a man-in-the-middle attack [9] during an HTTPS communication, or she can try to brute-force a target website until the victim's credentials are

guessed [14]. In this case, unless the website implements some type of protection (e.g., CAPTCHA, blocking of the account log-in for a few seconds after n attempts) or warning system (e.g., sending an e-mail to the account owner after a number of failed log-in attempts), the attacker would be undetectable due to her knowledge of the password of the victim.
The capabilities of an *active sniffing attacker*, instead, would be subject to various changes. If the server adopts the old Pw recovery method (e.g., it stores the registered users passwords in clear text and, when requested, it sends back the original user's password), the sniffing attacker can ask for the password recovery on behalf of the user but it would be both detectable (given that the user will receive the e-mail with the old password) and useless (she should brute-force the same password as before). In case the web server adopted either New or Temp Pw password recovery method, instead, the active sniffing attacker would have useful information about the password she has to brute-force. In fact, even if the user has a very strong password, the attacker could apply for a new (or a temp) one on her behalf. In this way, the attacker would know exactly the structure and the level of security of the new website-generated password and could brute-force it accordingly. For example, the target user has a very hard-to-guess password, and the attacker starts the password recovery methods on a target website in her behalf. The attacker, by registering a personal account on that website, knows how temp and new passwords are chosen (e.g., only numbers, both capital letter and numbers, etc.), so she would have useful information about the password structure as well as its length. In case of HTTP or HTTPS links used for password recovery, the attack will become detectable and the attacker would not gain any additional information about the password with respect to the passive attacker ones.

Table 6. Synoptic table related to passive attackers

Attacks / Recovery methods	**Old Pw**	**New Pw**	**Temp Pw**	**HTTP link**	**HTTPS link**
Mail service provider-level	*undetectable*	*undetectable*	*detectable*	*detectable*	*detectable*
Web server intruder	*undetectable*	Depends on the storage method	Depends on the storage method	Depends on the storage method	Depends on the storage method
Client intruder	*undetectable*	*undetectable*	*detectable* / Depends on the user's behavior	*detectable* / Depends on the user's behavior	*detectable* / Depends on the user's behavior
Sniffing	*undetectable*	*undetectable*	*undetectable*	*undetectable*	*undetectable*

Note that two of the main differences between passive and active attackers involve attack timing and attack extension. On the one hand, active attackers may get information about the victim's credentials at any time, while passive ones must wait for a move from the user. On the other hand, an active attacker must take immediate actions while a passive attacker can be implemented as a fragment of autonomous software that will be triggered by certain events (e.g., starting of a recovery password procedure by the user, receiving a mail with credentials/links). This means that passive attacker softwares could be easily and quickly propagated.

Table 7. Synoptic table related to active attackers

Attacks / Recovery methods	**Old Pw**	**New Pw**	**Temp Pw**	**HTTP link**	**HTTPS link**
Mail service provider-level	*undetectable*	*detectable*	*detectable*	*detectable*	*detectable*
Web server intruder	*undetectable*	*detectable* / Depends on the storage method	*detectable* / Depends on the storage method	*detectable* / Depends on the storage method	*detectable* / Depends on the storage method
Client intruder	*undetectable*	*detectable*	*detectable*	*detectable*	*detectable*
Sniffing	*detectable - useless*	*detectable - simpler*	*detectable - simpler*	*detectable*	*detectable*

Best Practices. The provided analysis has highlighted serious security deficiencies related to the storage and management of websites passwords. This paragraph contains a list of best practices to be used for a first mitigation phase.

Basic security guidelines dictate, for at least 40 years, that passwords should never be stored in clear text. For instance, passwords could be hashed by using a dash of salt, different for each one, making any salted rainbow table useless or extremely slow (because of the prohibitively size) Another solution can be the implementation of slow hashes by websites. The website would suffer negligible delay every time a user registers an account or logs in, but the password table attack by the attacker would be infeasible due to the time required. Moreover, the password length plays a fundamental role. Computers, Field-programmable gate array (FPGA), and Application-specific integrated circuit (ASIC) technologies, are faster, enabling to brute-force both non-salted and salted passwords, if not long enough. 8-character passwords are not enough robust even by using a combination of numbers, lowercase letters, uppercase letters, and special characters. In fact, even by considering a fully random password, the complexity will be equal to 95^8 ($< 2^{56}$), a negligible number if we consider that the validation of a block of BitCoin transactions requires, as of writing, 2^{72} hashes [4]. Considering these premises, websites should be able to accept only passwords with an high level of security ($> 2^{70}$).

Since currently, for most websites, the ability to access an e-mail address identifies the registered user as the owner of that address, some further good practices could be provided to users:

– do not allow the browsers/websites (especially the mail service providers' ones) to save the password. Better to waste a few seconds to rewrite it than losing control of your identity on the web;
– always log-out from websites and remove open sessions;
– avoid using the same password for more than one website, otherwise check how the passwords are stored (and password recovery are managed) from the website before entering the sensitive password;
– before inserting either passwords or other sensitive information, make sure that there are no key loggers in your system (at least the software ones, more easily detectable);

6 Conclusion

In this paper, we first provided a survey of both user authentication mechanisms implemented by websites, and password recovery mechanisms currently adopted. Then, we modeled an attacker with a set of capabilities and we showed how simply she is able to get the users' confidential information—e.g. recovering the password, or accessing e-mail. Later, we performed a detailed analysis of users password management for Alexa's top 200 websites (see Section 5) of five European countries, respectively England, France, Germany, Italy, and Spain. It is worth noting that analyzed websites are among the top ones according to the Alexa's ranking, so, they are supposed to be way too secure compared to those present in the lower ranking. Our results show that, nowadays, almost 44% of the analyzed websites are vulnerable—simply because lacking adoption of best practices, and hence subject to GDPR's fines. We are currently working on an open-source host-based software that can manage authentication and password recovery from any website in a more secure way, effectively thwarting the discussed attacks.

References

1. 2012 linkedin breach just got a lot worse: 117 million new logins for sale (Last checked June 2018), https://threatpost.com/2012-linkedin-breach-just-got-a-lot-worse-117-million-new-logins-for-sale/118173/
2. 6.5 million linkedin passwords reportedly leaked, linkedin is looking into it (Last checked June 2018), http://goo.gl/dWMvd7
3. Amazon alexa topsites (Last checked June 2018), http://money.cnn.com/2017/10/03/technology/business/yahoo-breach-3-billion-accounts/index.html
4. Blockchain last block (Last checked June 2018), https://blockchain.info/block/0000000000000000001ba3bf33b8c46856dfc05a3b3aebffa245f16269b54dc1
5. Every single yahoo account was hacked - 3 billion in all (Last checked June 2018), https://www.alexa.com/topsites
6. Gdpr portal (Last checked June 2018), https://www.eugdpr.org/

7. How are alexas traffic rankings determined? (Last checked June 2018), http://goo.gl/jMjpeS
8. Inside the russian hack of yahoo: How they did it (Last checked June 2018), https://www.csoonline.com/article/3180762/data-breach/inside-the-russian-hack-of-yahoo-how-they-did-it.html
9. Man-in-the-middle attack (Last checked June 2018), https://www.owasp.org/index.php/Man-in-the-middle_attack
10. Protonmail (Last checked June 2018), https://protonmail.com/
11. Protonmail - two factor authentication (2fa) (Last checked June 2018), https://protonmail.com/support/knowledge-base/two-factor-authentication/
12. The real life risks of re using the same passwords (Last checked June 2018), https://pixelprivacy.com/resources/reusing-passwords/
13. Stronger security for your google account (Last checked June 2018), https://www.google.com/landing/2step/
14. Using burp to brute force a login page (Last checked June 2018), https://goo.gl/jfwiCJ
15. Website security statistics report (Last checked June 2018), https://info.whitehatsec.com/rs/whitehatsecurity/images/2015-Stats-Report.pdf
16. Brainard, J., Juels, A., Rivest, R.L., Szydlo, M., Yung, M.: Fourth-factor authentication: somebody you know. In: Proceedings of the 13th ACM conference on Computer and communications security. pp. 168–178. ACM (2006)
17. Dell'Amico, M., Michiardi, P., Roudier, Y.: Password strength: An empirical analysis. In: INFOCOM, 2010 Proceedings IEEE. pp. 1–9. IEEE (2010)
18. Florencio, D., Herley, C.: A large-scale study of web password habits. In: Proceedings of the 16th international conference on World Wide Web. pp. 657–666. ACM (2007)
19. Furnell, S.: An assessment of website password practices. Computers & Security **26**(7-8), 445–451 (2007)
20. Furnell, S.: A comparison of website user authentication mechanisms. Computer Fraud & Security **2007**(9), 5–9 (2007)
21. Garfinkel, S.L.: Email-based identification and authentication: An alternative to pki? IEEE security & privacy **99**(6), 20–26 (2003)
22. Just, M., Aspinall, D.: Personal choice and challenge questions: a security and usability assessment. In: Proceedings of the 5th Symposium on Usable Privacy and Security. p. 8. ACM (2009)
23. Kamp, P.H., Godefroid, P., Levin, M., Molnar, D., McKenzie, P., Stapleton-Gray, R., Woodcock, B., Neville-Neil, G.: Linkedin password leak: Salt their hide. ACM Queue **10**(6), 20 (2012)
24. Ladakis, E., Koromilas, L., Vasiliadis, G., Polychronakis, M., Ioannidis, S.: You can type, but you cant hide: A stealthy gpu-based keylogger. In: Proceedings of the 6th European Workshop on System Security (EuroSec) (2013)
25. Parker, D.B.: Fighting computer crime. Scribner New York, NY (1983)
26. Reeder, R., Schechter, S.: When the password doesn't work: Secondary authentication for websites. IEEE Security & Privacy **9**(2), 43–49 (2011)
27. Schechter, S., Egelman, S., Reeder, R.W.: It's not what you know, but who you know: a social approach to last-resort authentication. In: Proceedings of the sigchi conference on human factors in computing systems. pp. 1983–1992. ACM (2009)
28. Stobert, E., Biddle, R.: The password life cycle: user behaviour in managing passwords. In: Proc. SOUPS (2014)
29. Yan, J., Blackwell, A., Anderson, R., Grant, A.: Password memorability and security: empirical results. IEEE Security Privacy **2**(5), 25–31 (Sept 2004). https://doi.org/10.1109/MSP.2004.81