

USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

An Analysis of the Security, Usability, and Automation Capabilities of Password Update Processes on Top-Ranked Websites

Alexander Krause, Jacques Suray, Lea Schmüser, Marten Oltrogge, Oliver Wiese, Maximillian Golla, and Sascha Fahl, *CISPA Helmholtz Center for Information Security*

https://www.usenix.org/conference/soups2026/presentation/krause

This paper is included in the Proceedings of the Twenty-Second Symposium on Usable Privacy and Security.

August 24–26, 2026 • Hannover, Germany

ISBN 978-1-939133-59-5

Open access to the Proceedings of the Twenty-Second Symposium on Usable Privacy and Security is sponsored by USENIX.

An Analysis of the Security, Usability, and Automation Capabilities of Password Update Processes on Top-Ranked Websites

Alexander Krause, Jacques Suray, Lea Schmüser,
Marten Oltrogge, Oliver Wiese, Maximillian Golla, and Sascha Fahl
CISPA Helmholtz Center for Information Security

Abstract

Password updates are a critical part of the password lifecycle and are recommended following exposure of reused passwords or suspected compromise. However, password update processes are often cumbersome, require manual password creation, and involve inconsistent website workflows that hinder reliable automation by password managers.

In this work, we conduct the first in-depth, systematic analysis of 111 password update processes deployed on top-ranked websites. We provide novel insights into their overall security, usability, and automation capabilities, and contribute to authentication security research by improving the understanding of password update processes. Websites often deploy highly diverse, complex, and confusing password update processes that are not supported by password managers. Processes are often challenging to use, and end-users struggle to transfer experience and knowledge across websites. Notably, security measures designed to enhance security often hinder password manager automation. We conclude our work by discussing our findings and giving recommendations for web developers, the web standardization community, and security researchers.

1 Introduction

User authentication security and usability have been an enormous challenge for the computer security community for decades. Most users have to manage tens or even hundreds of online accounts. While authentication technologies are diverse and research and industry have suggested alternatives to password-based authentication, such as passkeys [50, 56], passwords remain a critical authentication mechanism: Passwords are more inclusive and fail-safe; they can easily be shared with others, and end-users do not need additional devices, apps, or tokens to authenticate.

However, previous research [14, 25, 99] and a plethora of password breaches [15, 27, 39, 98] have illustrated that password-based authentication is full of challenges for end-users, service operators, and software developers. Having to manage unique and hard-to-guess passwords for all of their accounts is unrealistic for end-users [26], which is why they have developed various coping strategies like using easy-to-memorize (and guess) passwords [82, 103] or reusing similar passwords across accounts [31, 73, 107]. Although password managers can help end-users, previous research illustrates challenges concerning adoption [58, 83, 87], usability [43, 78, 111], and security [5, 55, 77]. Developers struggle to store passwords securely [64] and have trouble implementing password manager-friendly authentication flows [36, 77]. Service providers deploy password policies [2, 53] that can negatively impact overall account security [34, 42].

While password creation and management have been extensively studied [47, 53, 82, 83, 92, 102], little attention has been paid to the password update processes implemented by websites. End-users might need to update their passwords for multiple reasons: (1) they may have forgotten the password and need to use a website's reset feature [13]; (2) they might need to go through an account remediation process [66, 106], for example, in response to a breach [96]; (3) they might need to unshare the password [3, 75, 91], for example, after a breakup or in response to employee offboarding; (4) they update it as a precaution, in response to an email notification about the login reporting suspicious account activity [57]. While the above list of reasons to update a password is not exhaustive, it highlights the need for secure and user-friendly password update processes to enhance account security.

Web password updates remain non-automated and inconsistent, forcing users to locate forms and navigate multi-step procedures that vary across sites. While password managers excel at generating and storing credentials [77, 78], automated pass-

Copyright is held by the author/owner. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee.
USENIX Symposium on Usable Privacy and Security (SOUPS) 2026.
August 23–26, 2026, Hannover, Germany.

word updates remain limited. Dashlane's attempt at one-click updates illustrates this challenge: introduced (2014) [67], revamped (2021) [85], then discontinued (2022) [20] due to maintenance complexity [93]. The W3C's password update standardization effort, accessible via a well-known URL [63], could streamline access to update forms for users and tools. However, our findings reveal a limited adoption of this proposed standard and persistently complex update processes.

To date, the literature lacks a systematic analysis of password update processes on the web. We aim to contribute to a deeper understanding of the challenges posed by password update processes and give recommendations to enhance the security, usability, and automation capabilities. In this work, we answer and discuss the following research questions:

RQ1 [State of the Art] *How do websites implement password update processes?* We aim to gain in-depth insights into password update processes and investigate their impact on end-users and password managers.

RQ2 [Challenges and Obstacles] *What are common security, usability, and automation challenges and obstacles end-users and password managers face when interacting with password update processes?* We aim to identify and understand the challenges to contribute towards more secure and usable processes in the future.

RQ3 [Recommendations] *How can websites be better supported to deploy more secure, easy-to-use, and more easily automated password update processes?* We give recommendations for service providers, password managers, and the research community.

Contributions and Key Findings. We analyze password update processes on 111 top-ranked Tranco websites [52], focusing on security, usability, and automation capabilities. We found that update forms are often buried deep in the account settings, requiring up to six clicks, and use inconsistent designs. Some forms omit password confirmation fields; many lack visible password requirements, and some enforce conflicting policies during recovery versus regular password changes. Most do not provide confirmations, and many fail to send notifications, enabling stealthy, unauthorized changes. Automation faces barriers: Most forms lack "autocomplete" attributes [97], few support the W3C draft [63], and some deploy a CAPTCHA or RBA [110], blocking automated tools. Overall, many update processes on top-ranked websites are complex, confusing, and inconsistent. They might negatively impact end-user authentication usability, hinder password manager support, and decrease authentication security.

2 Reasons to Update Passwords

We distinguish password changes, where users still know their current password, from password recovery, where they have lost access to it.

Password Change (PWC). Regular password changes were once a common piece of advice and part of many password policies in the past. However, prior work has illustrated that making password changes mandatory is frustrating for users and results in weaker passwords [17, 34, 65]. Yet there are many good reasons to change a password: For example, a password change is recommended after a breach [27] and often part of the account remediation process [66, 106]. Other reasons for changing a password include unsharing a password [3, 75, 91], improving account security hygiene [86], and responding to suspicious login notifications [57].

Password Reset (PWR). In case users forget their password or lose access otherwise, most websites deploy password reset processes to allow users to set a new one. In contrast to a password change, a password reset involves different means of (fallback) authentication to confirm the identity of the user [51, 88], for example, by clicking a link in an email, entering a code received via SMS, or answering personal knowledge questions [13, 29].

3 Related Work

Prior work on passwords is extensive; we focus on two areas: breach alerting and account remediation [38, 57, 66, 100], e.g., by notifying users about their passwords' appearance in a password leak and how to respond and secure their accounts. Our work builds upon this by investigating the steps commonly required to change or reset a password. The second major area focuses on password managers and automatic password changes [67, 77], such as W3C's attempt [63] to standardize an API to facilitate automatic changes. Our work builds on this by documenting obstacles and discussing solutions that could help automate the process.

Breach Alerting & Account Remediation. All major browsers [23, 69, 100], and a variety of password managers [24] notify users if their passwords appear in a data breach. A popular free service is "Have I Been Pwned?" by Tony Hunt [39], where users can learn about their appearance in a data breach by entering their email address. Academia has proposed protocols for compromised credential checking services [54, 80, 81] and tried to improve security notifications [38, 113]. For example, Mayer et al. [59] studied individuals' awareness and perception of data breaches. They found that 74% of the reported breaches were new to the participants. Golla et al. [31] studied password reuse notifications and noted the importance of ecosystem-level strategies, including password managers and convenient password update mechanisms. Zou et al. [112] analyzed 161 breach notifications. They found that notifications are often difficult to read, sometimes downplay the consequences, and frequently provide insufficient information on the urgency and prioritization of actions. Regarding

account remediation, Neil et al. [66] analyzed advice from 57 websites. They found that highly ranked websites and those that had already experienced a data breach provided more comprehensive advice for their users. Similarly, Walsh et al. [106] studied advice given by top websites from seven different countries. They found that the provided instructions often miss essential steps and that users are forced to rely on third-party sources to resolve their problems. Oh et al. [79] evaluated factors influencing the ease of updating passwords, e.g., after a breach. Beyond remediation advice, Daffalla et al. [18] examined account security interfaces (ASIs), finding them inconsistent, insecure, and ineffective. Bhattacharya et al. [11] further showed that ASI logs and notifications remain difficult to navigate, even for expert users.

Password Managers & Automatic Changes. Prior work has studied password manager limitations: Lyastani et al. [55] quantified security, Huaman et al. [36] documented HTML compatibility, and Oesch and Ruoti [77] revealed autocomplete inconsistencies and non-standard field handling.

There have been attempts to automate or increase the usability of changing a password online. In 2014, Dashlane offered a one-click solution to automatically change passwords on popular websites [67]. In 2021, they revised the tool to run on the client side [85] to accommodate two-factor authentication (2FA) and RBA prompts caused by their outdated server-side solution. Interestingly, in 2022 Dashlane decided to discontinue the changer [20], because "[. . .] *it was an extremely complex feature that was prone to breaking as sites changed their change password forms & processes*" [93].

Well-known Uniform Resource Identifiers (URIs) are defined by the IETF in RFC 8615 [74]. It addresses the need for the consistent availability of services or information at specific URLs across servers. In 2018, an engineer working for Apple proposed the concept of "a well-known URL for changing passwords" [76], i.e., "example.com/.well-known/change-password." The goal is to make the password updating process easier to use by automatically directing users of tools, such as password managers, to the password update form, thereby easing the hassle of navigating to it manually. Shortly after, Apple Safari, Google Chrome, Microsoft Edge, and certain third-party password managers implemented the proposal. In 2021, the W3C officially published a working draft [63] standardizing the concept. Dashlane extended this concept with their own proposal, focusing on password managers [19]. They propose a similar, well-known URL, but with an extension that supports a lightweight automatic password change protocol. This protocol should enable password managers to automatically change a password by providing the username, current password, and new password. They also introduce a minimal approach to handle two-factor authentication and CAPTCHAs. Closely related to password changing APIs are password policy description languages [28, 35] and password manager quirks [8], which we do not consider here.

4 Method

Next, we describe our methodology for analyzing password change and reset implementations using new accounts. Finally, we discuss study limitations and ethics. Our dataset and protocol are available in our replication package [49].

4.1 Data Collection

We sampled websites from the Tranco list [52] and analyzed a subset of the top-ranked websites.[1] We began by analyzing websites that allowed us to create an account and did not meet our exclusion criteria (outlined below). In total, we created accounts on 111 websites between September 2024 and February 2025. A complete list of the websites can be found in Table 3 in Appendix A. For websites that required SMS-based verification in addition to an email address, we used a US-based or German phone number to create the account when necessary. To avoid any IP-based geo-blocking, we utilized a commercial VPN to spoof our location.

Account Creation Process. Before we began creating accounts, we prepared all the necessary information, including first and last name, birthdate, address, phone number, email address, and generic answers to security questions. If the website had a password policy, we generated a random password that complied with it. If the website had no policy, we generated a 20-character random password. During account creation, we did not enroll in non-mandatory 2FA.

Exclusion Criteria. We excluded websites for which we could not create a free account, such as those that require a subscription. Furthermore, we had to exclude websites where our account was banned, for example, because it was not a unique or real person. In addition, we excluded all websites that were not available in English or that we could not translate to English using Google Chrome's internal website translation feature [32]. In line with previous work [4], we also excluded websites that would have required us to sign a contract or upload a passport. In total, we had to exclude 291 websites (between the highest Tranco rank 1 and the lowest rank of 402 that we decided to consider in our analysis).

Protocol. We developed our initial analysis protocol based on related work [4, 16, 36, 110] and discussions among the team while jointly analyzing PWC and PWR processes. Our protocol focuses on attributes that help us evaluate the user experience, automation capabilities, and security implications. After evaluating the first 10 websites, we refined and expanded the protocol to include 32 attributes. To minimize data-collection errors and facilitate reproducibility, we used a custom-built form. We recorded our screens while analyzing the PWC and PWR processes on the websites.

[1] List: https://tranco-list.eu/list/X57KN, as of March 1, 2023.

Table 1: Attributes and their descriptions in the **analysis protocol** for the password change and reset processes. Findings for the update processes, navigation path lengths, and the "password change" W3C draft adoption are presented in Section 5.

ID	Attribute	Description	PWC	PWR
		Initialization		
1	RBA Email	An additional risk-based email authentication was performed.	●	○
2	RBA CAPTCHA	An additional risk-based CAPTCHA authentication was performed.	●	○
3	RBA SMS	An additional risk-based SMS authentication was performed.	●	○
4	Autocomplete Login	Is True if the attribute is set to 'email' or 'username' for the user field and 'current-password' for the password field.	●	○
		Password Update		
5	Opposite Process	The PWC process references the PWR process and vice versa.	●	●
6	Pre-Verification	To enter the password update form, the user must verify their identity again.	●	○
7	Current PW Field	The current password is required to update the password.	●	○
8	Confirm PW Field	To update the password, the new password must be confirmed once again.	●	●
9	Autocomplete PWC	Is True if the attribute is set to 'current-password' for the current-password field and to 'new-password' for the 'new' and 'confirm' password field.	●	○
10	PW Policy	Website provides information about its own password composition policy.	●	●
11	PW Visibility Toggle	Users have the option to toggle the visibility of the passwords they enter.	●	●
12	PW Generator	The website has a built-in function to generate a random password.	●	●
13	PW Strength Meter	Website provides feedback about the password strength.	●	●
14	PW Breach Check	Website checks whether user data appears in known data breaches.	●	●
15	Secure PW Tips	Website provides advice for strong passwords.	●	●
16	CAPTCHA	At least one CAPTCHA had to be solved before or after the password update.	●	●
17	PW Reuse Prevention	Updating the password to a previously used password is actively prevented.	●	●
18	Logout All Devices	Website communicates that all sessions expire when updating the password.	●	●
19	Logout Partially	Website communicates that some sessions expire, excluding the user's session.	●	●
20	Expire Current Session	Without any prior information, the user is logged out after the update.	●	○
21	Automatic Login	The user is automatically logged in after updating their password.	○	●
		Feedback		
22	Visual Feedback	The website visually communicates a successful password update.	●	●
23	Pop-up	The feedback is displayed in a temporary or closable pop-up.	●	●
24	Short-Display-Time	The feedback is only visible for a very short amount of time.	●	●
25	Embedded in Page	The feedback is persistent and embedded in the webpage.	●	●
26	Hard to Detect	The feedback is hard to notice due to the size, arrangement on the page, or color.	●	●
27	Redirect	There is an automatic redirect after updating the password.	●	●
28	Redirect Main Page	The redirect leads to the main page.	●	●
29	Redirect Same Page	The redirect directs to the same page without changing the URL.	●	●
30	Redirect Other	The redirect leads neither to the main page nor to the same page.	●	●
31	Sign-in Notification	A sign-in notification is sent to the user's email address.	●	●
32	PW Update Notification	A password update notification is sent to the user's email address.	●	●

● Attribute analyzed. ○ Attribute not applicable and therefore not part of the analysis.

Our implementation of the analysis protocol is available in the replication package. The final protocol checks multiple attributes depending on whether it is a PWC or PWR process (see Table 1). We started with the login form, as a login is often required to change a password. The form also typically allows the start of a PWR process. We analyzed automation capabilities, e.g., various HTML attributes and security measures deployed by the service, e.g., RBA, or the presence of a password strength meter. We also evaluated website feedback and notifications for successful password updates, such as those sent by email or SMS. In addition, we analyzed whether and how websites implement the well-known URL for changing passwords W3C draft by appending `/.well-known/change-password` to the domains.

Password Update Processes. Two researchers independently applied the protocol to each website using fresh Google Chrome profiles. We first analyzed the PWC process, following these steps: (1) logging in; (2) searching for the account settings to change the password; and (3) changing the pass-

word following the service's instructions. We then looked at the PWR process, following these steps: (1) searching the website for the PWR entry point; (2) following the reset instructions; and (3) setting a new password. We also tested whether the website accepted the previous password or required the creation of a new one. Screen recordings were collected during the analysis to support potential conflict resolution. In the event of a conflict, the researchers jointly reviewed the recording and reached an agreement. In total, we resolved 48 conflicts (1.6% of all analyses) for the password change process and 197 conflicts (6.4% of all analyses) for the password reset process. We followed an iterative coding approach until no new password update process emerged in the final 10 websites [12, 104]. As we resolved conflicts immediately when they emerged, we decided not to report an inter-coder agreement [33, 37, 48, 60, 108, 109]. Multiple coders independently evaluated the workflows, achieving high initial agreement: PWC: 98.2%, PWR: 94.7%, combined: 96.3%. We treated disagreements as conflicts and resolved them by reviewing the captured data. All disagreements resulted from human error rather than interpretive differences.

4.2 Data Analysis

Multiple researchers jointly applied affinity mapping [10, 84], a method for clustering information, to identify the steps needed to perform the PWC and PWR, and to organize the observations into potential automation, usability, and security issues. Usability-related systematic analysis findings are grounded in established knowledge and practices [62, 70–72].

Quantitative vs. Qualitative. We report the quantitative findings of our systematic analysis in Table 2. In addition, we use quantifiers, illustrated in Figure 5 in Appendix B, to determine which qualitative findings are most relevant throughout the results (see Section 5) and discussion (see Section 6) sections.

Limitations. Several limitations affect our work. First, our sample does not cover all websites with user accounts; lower-ranked and less popular websites are underrepresented. We may therefore have missed insights outside our study scope, but followed established web and security measurement practices [4, 21, 45]. Thus, we are confident that our dataset provides valuable insights to the community. Second, if possible, we created accounts on the US versions of the websites (enforced by our browser language and IP address) and used US- and German-based phone numbers to receive SMS. The behavior of large sites is known to change depending on the location, e.g., due to the GDPR. However, we believe the effect on account management is marginal, particularly in relation to the properties relevant to our study. Our sample is also biased by the Tranco list (Eurocentrism) and may miss popular websites from other regions around the world, such as Asia, South America, or Africa. Finally, websites are not static; developers regularly refine and improve their usability and security. Our results are a snapshot of their current state. Similarly, RBA and geo-restriction limit the reproducibility. Although we recorded our sessions, reproducing the study may yield different findings for specific websites. However, our general observations are reproducible.

4.3 Ethics Statement

Our institution's ethical review board approved this work. In our evaluation, we manually examined non-personal data. We took care not to cause unusual resource burdens on the websites and do not report any brands or website names with problematic processes in this work. No help desk staff were involved in our experiments. The involved researchers did not use their private accounts. At no point was the researchers' personally identifiable information, particularly of the involved students, part of this research. Reporting potential security vulnerabilities to the affected stakeholders is considered good scientific practice. We examined our raw coding data for vulnerabilities.

5 Results

We now present the main findings of our systematic analysis. This section is structured along (**RQ1**: State of the Art and **RQ2**: Obstacles). We provide details for the PWC and PWR processes with a particular focus on user experience, security, and the challenges faced by end-users and password managers. To increase the readability of our findings reporting, we use quantifiers when referencing a number of websites illustrated in Figure 5 in Appendix B. We also use the IDs from Table 2 throughout the sections to relate to the analyzed and reported attributes. Throughout, we report counts as n (p%); unless noted otherwise, percentages are relative to the 96 websites offering a PWC and the 109 offering a PWR.

5.1 Password Change and Reset Processes

General Process. To sign in, all services required an identifier, such as an email address, username, phone number, or account ID, along with a password and, in some cases, a second factor. Some websites deployed CAPTCHAs or RBA for additional security. Overall, websites employed various login processes.

The PWC forms on 80 of 96 websites (ID7) required us to enter the current password before we could set a new one or confirm our new password. After updating the password, most websites provided visual confirmation. The majority also sent notification emails or SMS; about half of the websites in the PWC process and the majority in the PWR processes sent HTTP redirects (PWC: 44, 46%; PWR: 77, 71%; ID27). Some enforced immediate logouts by expiring the session during the PWC process (PWC: 21, 22%; ID20). Some services did not send any notifications after the password was

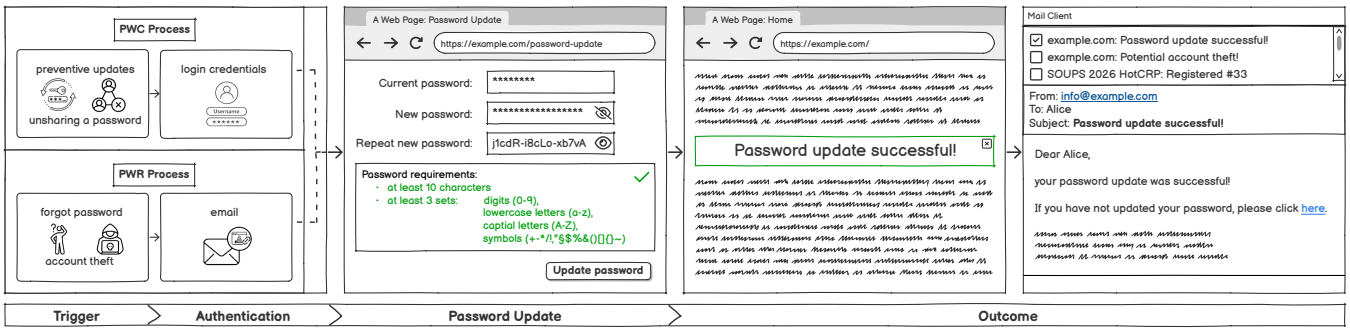


Figure 1: Illustration of a well-implemented password update (1) using PWC and PWR process **triggers**, (2) trigger-specific **authentication**, (3) **password update** forms including security features, and (4) the **outcome** communicated to the user.

changed. The PWC process was often similar to the PWR process, and some websites even forwarded us to the PWR process (PWC: 15, 14%; ID5). Instead of providing a password update form, the website sent an email with instructions for resetting the password. In contrast, only a few websites (PWR: 2, 2%; ID5) forwarded us from the PWR to the PWC process. Three websites did not offer a PWC process. On four websites, we could not find a PWR process. Instead, they sent a link to a valid session in an email and redirected us to the PWC form. To start the PWR process, websites often requested our email address. Some requested our phone number, username, or account ID. However, we also observed complex processes that require additional information, such as multiple identifiers (e.g., email and phone number). While most PWR processes were straightforward, we also observed implementations that required us to solve CAPTCHAs and answer questions. The email confirmation for PWR varied a lot. We sometimes received a link that created a session and redirected us to the PWC form without being required to log in first. Other services asked us to copy or type a code or use a one-time password. After we reached the PWR form, the processes were often comparable to the PWC processes in terms of security measures, visual feedback, and notifications. Figure 1 illustrates a generic well-implemented process.

5.2 User Experience and Interfaces

Below, we illustrate the user experiences of logging in, navigating to the password update form, and interacting with the form. Those three aspects substantially impact the likelihood of successfully updating a password.

Login & Initialization (ID1–3,5). We could sign in on many websites by entering our email address and password. No further interaction was required. All but two of the websites that offered a PWR process directly provided a "Forgot Password" link on their login form. On some websites, we triggered RBA [110] (email: 19, 20%, ID1; CAPTCHAs: 7, 7%, ID2; SMS: 2, 2%, ID3) or other means of credential stuffing or bot protection [40], which required additional user interactions, e.g., entering a code or solving a CAPTCHA. As is inherent with RBA, we encountered it in a non-deterministic way. In such cases, we often had to provide a verification code received via email, SMS, or both. Most websites allowed us to copy and paste the code. However, we also had to manually type the code on some websites. This was necessary because the website would sometimes drop parts of the code or put the complete code into a single-digit text field, which impacted the user experience. Several websites implemented CAPTCHAs, including Google's reCAPTCHA v3, which required users to complete tasks like image selection and text transcription. Audio-based variants challenged users to identify clips containing birdsong from multiple samples. A notable implementation involved dice-based challenges in which users matched target numbers to images of five dice. The correct selection required calculating the sum of visible dice faces, with multiple iterations per session. These complex implementations risked user frustration due to their time-consuming nature. While CAPTCHAs occasionally appeared during password changes or resets (PWC: 5, 5%; PWR: 16, 15%; ID16), security questions remained uncommon.

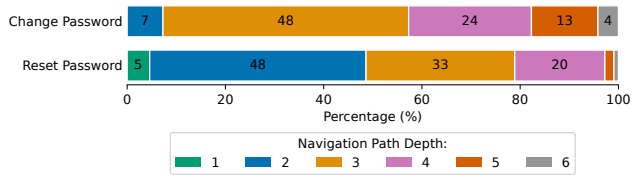
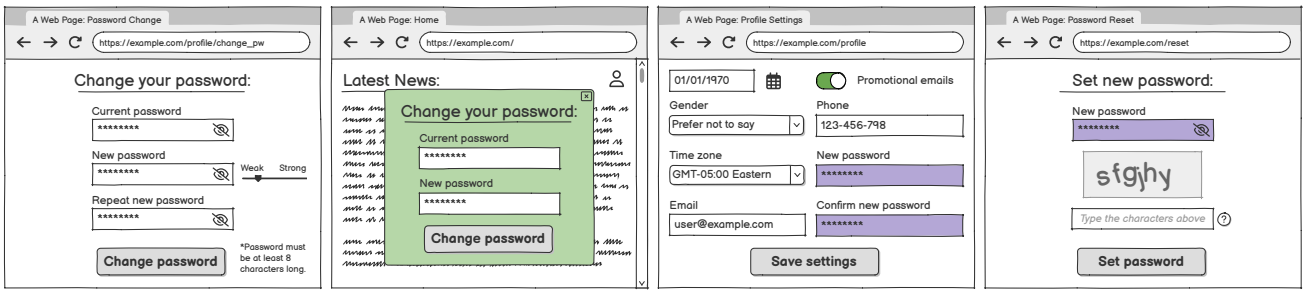


Figure 2: Distribution of the **navigation path depth** of PWC and PWR processes (higher depth often requires more effort). Example: *Profile* → *Advanced* → *Security* → *Password* (4).

Navigate to the Form. After logging in, we had to locate the PWC form, which was generally challenging since websites often implemented confusing site paths. Figure 2 visualizes and compares the path depth for both processes. In the PWC process, the navigation path consisted of two to six steps, with three steps being the most frequent. Despite similar



(a) Dedicated Page with URL (b) Modal without URL (c) Profile Settings (d) Reset with CAPTCHA

Figure 3: Common **password forms** we observed in PWC and PWR processes. (a) illustrates a password change page accessible via a **dedicated URL**, (b) shows a **modal popup** of a web application that does not offer a URL, (c) displays **profile settings** that include some password fields, (d) shows a reset form with a **CAPTCHA** but no password confirmation field.

labels, the name for the password update process differed from site to site. For example, we frequently encountered: "Authentication," "Account Security," or "Edit Profile." While three websites offered multiple paths to reach their password update forms, we had to use a search engine to find the starting point for at least one of them. To trigger the PWR process, we had to perform between one and six navigation steps, but mostly only two. Typically, we had to access the login form and could trigger the PWR by clicking on "Forgot Password?" or a similar label. This disparity uniquely affects password updates, as resets are almost always initiated by a direct link provided in breach or recovery emails. Conversely, PWC workflows often lack standardization and consistent terminology, as evidenced by labels such as 'Account Security' or 'Change Password,' which can lead to usability issues.

Interaction on the Form (ID7–8). We found different implementations of update forms in both the PWC and PWR processes. A few are illustrated in Figure 3. Some were on a dedicated page, reachable via a dedicated URL. Others were implemented as popups, either as a modal dialog within the page or as a separate window; i.e., the form cannot be reached directly via a URL. Most password change forms require users to enter their current password before setting a new one. For both password change and password reset, most websites also require users to confirm the new password to avoid typos (PWC: 75, 78%; PWR: 75, 69%; ID8). Still, confirmation was not required in all cases; some forms accepted the new password without asking users to retype it. One particular website asked us to enter the current password in a second step after confirming the password change, which caused some frustration. Since we used a password manager for our analysis, we had already updated the password manager entry for this website, leading to difficulties accessing the old entry in the vault, and thus, we could not complete the PWC in the end. Consequently, we had to reset the password using the PWR process to regain access.

5.3 Support Measures for Improved Security

Table 2 shows the security features, which we detail below.

Session and Password Expiration (ID18–21). Only a few websites automatically logged us out of all devices after completing the PWC process, leaving attackers with persistent session tokens (PWC: 8, 8%; ID18). By comparison, session termination was equally rare but less critical during PWR, where the reset invalidates previously used credentials (PWR: 7, 6%; ID18). We could choose to terminate all active sessions, but more often we were informed of an automatic session expiration, or learned about it by navigating to a different page on the website after the PWC, only to find ourselves logged out. In one case, we had to confirm that all active sessions would become invalid because of the PWC. We also observed that on about half of the websites, the user was automatically logged in during the PWR process (PWR: 51, 47%; ID21). Surprisingly, some websites implemented (the outdated and no longer recommended) automatic password expiration. On some sites, we could manually opt in to password expiration (see Figure 7 in Appendix C). On a few, we were informed, e.g., "*the password will expire in 80 days.*"

Password Policies (ID10). The majority of websites implement some password composition policy (PWC: 64, 67%; PWR: 81, 74%; ID10). Still, the implementation and policy varied across websites and, within the same website, between processes. We discovered that password policies can vary based on whether the user is in the PWC or PWR process. Specifically, a password containing a hyphen ("-") was permitted in the PWR process but was not allowed in the PWC process on the same website. We also observed websites that lacked a policy. Some displayed the policy requirements permanently, while others showed them after starting to type in the new password field. The latter implementation often had the issue that the policy was not displayed when copying and pasting or auto-filling a

Table 2: **Quantitative findings**. Because we reference the opposing processes (ID5), the total number of websites examined for each process is lower. The term *equality* refers to the ratio of attributes implemented equally for both processes on the same site. Not every attribute is always applicable (n/a).

ID	Attribute	Change (N=96)		Reset (N=109)		Equality
		True	False	True	False	%
	Initialization					
1	RBA Email	19	77	n/a	n/a	n/a
2	RBA CAPTCHA	7	89	n/a	n/a	n/a
3	RBA SMS	2	94	n/a	n/a	n/a
4	Autocomplete Login	43	68	n/a	n/a	n/a
	Password Update					
5	Opposite Process	15	96	2	109	n/a
6	Pre-Verification	11	85	n/a	n/a	n/a
7	Current PW Field	80	16	n/a	n/a	n/a
8	Confirm PW Field	75	21	75	34	100
9	Autocomplete PWC	15	96	n/a	n/a	n/a
10	PW Policy	64	32	81	28	84
11	PW Visibility Toggle	53	43	66	43	85
12	PW Generator	2	94	2	107	100
13	PW Strength Meter	30	66	27	82	89
14	PW Breach Check	1	95	1	108	100
15	Secure PW Tips	20	76	19	90	84
16	CAPTCHA	5	91	16	93	87
17	PW Reuse Prevention	48	48	47	62	76
18	Logout All Devices	8	88	7	102	91
19	Logout Partially	12	84	4	105	92
20	Expire Current Session	21	75	n/a	n/a	n/a
21	Automatic Login	n/a	n/a	51	58	n/a
	Feedback					
22	Visual Feedback	78	18	74	35	63
23	Pop-up	44	52	16	93	63
24	Short-Display-Time	18	78	8	101	76
25	Embedded in Page	35	61	59	50	58
26	Hard to Detect	9	87	1	108	91
27	Redirect	44	52	77	32	56
28	Redirect Main Page	4	92	28	81	74
29	Redirect Same Page	15	81	6	103	84
30	Redirect Other	26	70	43	66	63
31	Sign-in Notification	17	79	6	103	80
32	PW Update Notification	62	34	64	45	85

password, e.g., generated from a password manager, into the text field. Hence, websites rejected passwords that did not meet their composition requirements. We also found websites where the policy was displayed after clicking an additional help button, or when the entered password did not comply with it, i.e., we could not read the requirements before entering a password. In total, most websites displayed the password policy during the PWR process, whereas only a majority did so during the PWC process. Additionally, some policies prohibited using special characters like "%", or "#". More often, deployed password policies forced us to use a password of a certain length. Particularly, very long passwords ($\geq$ 120 characters) were often rejected. Overall, inconsistent implementations can degrade the user experience. Enforcing unnecessarily stringent requirements can even break compatibility with passwords generated by password managers [36, 77], resulting in even more user frustration. Despite the intention to ensure users are provided with strong passwords, we observed policies that mandate only a minimum of six characters, which is not in line with recommendations from academia [95] and requirements published by organizations such as NIST [96].

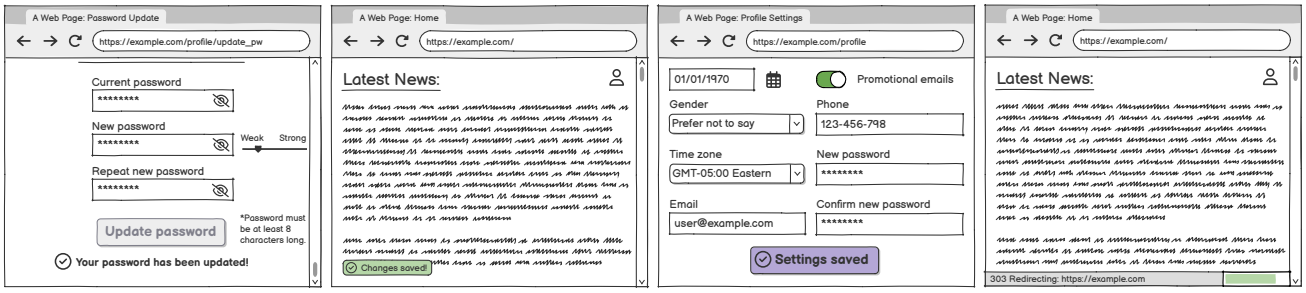
Password Visibility Toggles (ID11). A toggle for password visibility allows users to review their password. It is a crucial user interface component, as it can help to prevent typos and copy & pasting errors, especially on mobile devices [61]. The majority of websites implemented this feature for both processes (PWC: 53, 55%; PWR: 66, 61%; ID11). We also found an unfavorable combination of a new password field without a confirmation field and without a visibility toggle.

Password Leak & Breach Checks (ID14). Only one website (PWC: 1, 1%; PWR: 1, 1%; ID14) effectively implemented a password leak check, i.e., a check for whether the user-generated password is part of a list of known breached credentials. Two websites rejected passwords that were part of a known leak. One website disabled the password update button, while another required users to reset their passwords. In contrast, another website routinely processed the update, resulting in a redirect to the main page, though with a warning that the password was not set.

Password Generators (ID12). A few websites (PWC: 2, 2%; PWR: 2, 2%; ID12) offered us to generate a secure and policy-compliant password, e.g., "wMwY$fnSNHnB WtIq#wYiHmuA" or "6chUmazy-9krovnoe-lyuminal8". We observed differing views on secure passwords across websites, particularly regarding length and character set. However, we did not observe any particular recommendation optimized for memorability or ease of typing and use.

Password Strength Meters (ID13). While we found some websites (PWC: 30, 31%; PWR: 27, 25%; ID13) using a password strength meter to rate the guessability of a password during the PWC and PWR process, the used implementations differed in terms of rating and feedback. We found websites using different visual representations, e.g., a single-colored three-step bar or a multi-colored five-step bar. In addition, the ranking was performed by calculating some "entropy" and was often based on character classes and length (LUDS) [30].

Password Reuse Prevention (ID17). About half of the websites (PWC: 48, 50%; PWR: 47, 43%; ID17) tried to mitigate the risk caused by reusing an old password. For this, most websites use the "current-password" field in the PWC process to compare the current and the new password. We also found websites that restricted reuse of the current password, for example, by requiring users to change more



(a) Explicit (ID25) (b) Tiny and Disappearing (ID23) (c) Button (ID25) (d) Redirect (ID28)

Figure 4: Common visual **feedback** we observed in PWC and PWR processes. (a) Illustrates an **explicit** password change **message**, (b) shows a tiny, quickly **disappearing cue** about "Changes" that have been saved, (c) **abuses a button** to inform the user about saved "Settings," (d) does not display any text and instead **redirects** the user to the start page.

than five characters. Performing such similarity checks on cryptographically hashed passwords is challenging to implement and raises concerns about their secure storage. Such checks can also impact user experience by interfering with users' password choices, potentially causing additional unnecessary mental load and memorability issues.

Additional Security Advice (ID15). Some websites (PWC: 20, 21%; PWR: 19, 17%; ID15) provide advice while updating. This includes general security advice to protect user accounts, e.g., not to share passwords with others or how to create secure passwords, such as specific password lengths or using special characters (see Figure 6 in Appendix C).

5.4 Feedback After Setting a New Password

We observed that websites implemented different types of feedback. Below, we describe our findings on visual feedback, browser redirects, and notifications sent via email or SMS. Some websites combined methods. We visualize some representative examples in Figure 4.

Visual Feedback and Redirects (ID22–30). Browser feedback included messages confirming a successful update, such as flash messages or redirects to a new page. Most sites displayed at least some kind of visual feedback after an update (PWC: 78, 81%; PWR: 74, 68%; ID22). Many websites displayed feedback and then redirected to another page, while others did the opposite. A few (PWC: 15, 16%; ID29) redirected to the same page without displaying visual feedback; others did not provide any. Many times, we filled out the form and confirmed the update by clicking the *submit* button without receiving any feedback. Hence, it was unclear whether the update was successful. The place and duration of flash messages varied widely. In the PWC process, many websites, and in the PWR process, about half of the websites implemented messages as a persistent part of the website, directly above or below the password update form (PWC: 35, 36%; PWR: 59, 54%; ID25). Other messages were hard to catch, as they quickly faded. Most websites showed temporary messages for around one to two seconds (PWC: 18, 19%; PWR: 8, 7%; ID24). Some required users to actively acknowledge and close them.

Notifications via Email and SMS (ID31–32). During the update process, websites either sent an email or SMS notification or did not notify us at all. Some notified us of a login (PWC: 17, 18%; PWR: 6, 6%; ID31), while the majority notified us of a successful password update (PWC: 62, 65%; PWR: 64, 59%; ID32). Sporadically, we received notifications about a login from a new device or location.

5.5 Automated Password Update Challenges

We now report findings that affect password managers' ability to automate password updates. We found that many websites do not adhere to modern web standards and best practices for their password update forms [36], making automated password updates unnecessarily challenging.

Adoption of the "Password-Change" W3C Draft. While we analyzed all websites in our sample regarding their use of the W3C draft for changing passwords [63] with the ".well-known" standard [74], we found that only 6 of 111 (5%) implemented it. 4 websites redirected to the PWC process. Another 2 websites redirected to the PWR process. From those that redirected us to the PWC process, only one website redirected us to the form where we could change the password. All the other websites redirected us to an account security page, where we had to navigate further to the actual password update form.

HTML Attributes (ID4,9). Upon further investigation, we discovered that websites lacked support for autocomplete HTML attributes. Only 9 of 111 (8%) implemented the autocomplete attribute correctly in all login and password update

form fields. The benefit of automation on these websites is huge since password managers can automatically fill every necessary form field during the PWC process. We also found that some websites supported only specific processes, such as the login process (43, 39%; ID4), or only the update form during the PWC process (PWC: 15; ID9).

Furthermore, we observed that the autocomplete flow was incomplete on almost all websites (102 of 111, 92%), e.g., the autocomplete attribute was only available on the username or was missing in one of the fields in the change form. Moreover, some websites used non-standardized website-specific names or identifiers for relevant forms, making it difficult to automatically distinguish between the current and new password fields, e.g., "ooui-php-1" for a new password field or "from-textbox 16899" for the current password. Additionally, we noticed that some websites had misleading HTML names, such as "text" for the username. Some websites posed challenges due to missing field identifiers (names, IDs, autocomplete attributes). We even observed dynamically changing identifiers between page loads, likely from A/B testing, which occasionally caused autocomplete attribute conflicts, such as the unintended mapping of current password fields to new password inputs. A few websites implemented the HTML "passwordrules" attribute [8]. This attribute can support password managers in generating policy-compliant random passwords.

6 Discussion

Next, we discuss our findings and their implications for password update automation, end-user usability, developers, and web standardization, and contextualize them with prior work. Our study descriptively measures deployed update workflows; without a user study, usability statements should be read as hypotheses grounded in established principles and prior work. Where possible, we link observed design patterns to potential consequences, such as update errors, reduced password manager effectiveness, and weaker account security.

6.1 Heterogeneity and Security are Obstacles

Fully automating password updates could substantially streamline the process. Our work illustrates many challenges for automating updates: Most sites did not implement the "change-password" W3C draft (6 of 111, 5%) and made limited use of machine-readable interface practices (ID4, ID9). Both are optional proposals (a working draft and vendor conventions), so we report adoption descriptively: non-adoption is not a deficiency per se, but limits automation. Furthermore, no website offered an API to update passwords.

Interfaces Target Humans. Our analysis showed that current processes target humans. They largely did not implement common practices for machine-readable interfaces [90, 97], e.g., HTML attributes. Only 9 websites implemented attributes to support password managers in the login and password update forms. A plausible cause is that machine-readability is not a design goal: flows are built and tested for humans, and tool support yields little direct benefit. Dashlane's discontinued password changer [20, 85, 93] illustrates that automation requiring constant maintenance of site-specific heuristics is not sustainable at scale. Our findings show that without adoption of optional aids such as the "change password" W3C draft, even automatically navigating to the update form is challenging [79] (see Section 5.5). Recent industry efforts to automate credential updates [105] indicate that updating a compromised password remains a relevant target for automation. Our results suggest that broader deployment of such functionality would benefit from machine-readable account-management interfaces rather than continued reliance on reverse-engineering human-facing workflows. Currently, the names of links and paths to the update form vary significantly, making automated navigation difficult. Forms often do not include HTML attributes such as "autocomplete" or "passwordrules." Many sites also do not assign stable, descriptive IDs and names. The user-facing consequence: when a password manager cannot fill or generate the new password, users may fall back to manually created passwords, which can be weaker or reused, or abandon the update [36, 77]. Websites also confuse password managers by failing to communicate policies without prior (failed) user interaction (see Section 5.3). Thus, we suggest that developers carefully test their forms with password managers to verify their implementation [36].

Gatekeepers Block Automation. Many of the tested websites deployed RBA, 2FA, or CAPTCHAs (see Section 5.3). While these measures aim to enhance security, they require user interaction and hinder automated processing. This restricts password managers, but it reflects a genuine trade-off rather than a flaw: the same mechanisms block credential stuffing. Any future interface (see Section 6.3) must preserve these protections; likewise, automation-friendly design must not come at the expense of users who create passwords manually.

6.2 Inconsistent Patterns Overwhelm Users

Although all analyzed update processes targeted end-users, we identified multiple patterns that, following established usability heuristics, may impair usability and security. Our dataset contained no two sites with the same update pattern; instead, sites differed in navigation paths, update forms, and security measures. The lack of common patterns across sites limits end-users' ability to transfer their experience and knowledge from one site to another. Our findings suggest that each update process is a new quest, potentially involving complex challenges [11]. Although end-users might be able to develop heuristics such as looking for terms like "account," or "security settings," the inconsistencies in the deployed

processes cause unnecessary difficulties for end-users. Concretely, missing confirmation fields or visibility toggles (ID8, ID11) can lead to unintended new passwords and lost account access; sessions on other devices usually remain valid after an update (terminated on only 8 of 96 PWC and 7 of 109 PWR websites; ID18), so a new password does not necessarily end unauthorized access; and missing feedback or notifications (ID22 to ID32) leave users unsure whether the update succeeded and let unauthorized changes go unnoticed. (See the consequences outlined above and Section 5.4.)

Visibility Toggles. To protect against shoulder surfing attacks [9], most websites hide the entered password. However, setting a new password can be error-prone, especially when using a software keyboard on a mobile device. While our evaluation was limited to desktop browsers and mobile interfaces may differ, we found that some websites provided neither password confirmation fields nor visibility toggles. Previous work suggests that offering a password visibility toggle encourages users to choose stronger passwords on mobile devices [61]. To reduce end-user frustration, we recommend implementing a visibility toggle to improve the password update process.

Strength Meters. Password strength meters that promote weak passwords might lead to a user's false sense of security [30]. We noticed that certain strength meters did not behave correctly when dealing with copying and pasting or auto-filled passwords generated by a password manager, e.g., generated passwords may not be considered secure by a website's self-implemented password strength meter.

Policies. Most websites deploy password policies to avoid weak passwords. Adhering to password policies can be challenging for both end-users and password managers. In particular, the wide variety of deployed password policies poses a significant challenge. For password managers, generating a secure password according to the password policy of websites is not always possible, mostly due to certain length requirements or special character requirements [28]. Moreover, these policies are almost never machine-readable due to the absence of the "passwordrules" HTML attribute [7]. Transparent, easy-to-understand, and early communication of the policy can reduce obstacles during updates, i.e., by ensuring the user sees and understands the policy when visiting the password update form without additional interactions [95].

Generators. Some websites have embedded password generators and suggest them to users during the password update. While embedding generators into password update forms might help users choose stronger passwords or adhere to password policies, they often come with a usability drawback: Generated passwords are often difficult to memorize and manage [55], which makes good password manager support even more critical.

Compromised Credential Checking (C3). We observed that few websites used password leak checks [41]. A password leak check can be a helpful feature to increase account security and protect users against using breached or compromised credentials, and is recommended for implementation by NIST [96]. One limitation of the checks that prior work has observed is that they are usually strictly limited to the time when the password is updated [73].

Session Expiration. After a PWC or PWR, some websites require or enforce terminating all active sessions on all devices and browsers. This feature is important, as it can drastically increase account security and is very welcome for ending online account sharing [75]. Without it, an attacker might reuse an old active session after the account owner changes their password. User preferences may differ by update reason (e.g., routine change vs. unsharing), suggesting a user-controlled option (see R1). Moreover, the efficacy of session termination as a security measure against attackers depends on the interplay with notification and confirmation practices for password and email address updates.

Supporting Users. Websites utilize various tools to enhance account security, with many supporting updates through confirmation fields, visibility toggles, feedback, and post-change notifications. However, critical gaps persist in PWC and PWR workflows: while most sites communicate a policy (ID10), strength meters appear on about a third (ID13), and generators (ID12) and leak checks (ID14) are almost absent (2 and 1 of 111). While simplification of password updates is evident, robust assistance for creating secure credentials remains inconsistently implemented, highlighting a disconnect between usability efforts and proactive security reinforcement.

6.3 Recommendations

We organize our recommendations by priority, ground each in Table 2, and then discuss the actors best positioned to act.

High Priority: Security-critical.

- **R1: Treat Updates as Revocation, Not Replacement.** Only 8/96 (PWC) and 7/109 (PWR) sites terminated all sessions (ID18). Lingering sessions let attackers persist despite a new password. List and offer to expire other sessions, tokens, and cookies in every update flow.
- **R2: Always Notify the User Out of Band.** 34/96 (PWC) and 45/109 (PWR) sites do not notify users about the update (ID32), enabling stealthy changes. Send helpful notifications with a remediation option.
- **R3: Provide Guided Account Remediation.** Guide users through related account-security steps, including active sessions, recovery, and 2FA, rather than treating the password update as a complete action.

Medium Priority: Prevents errors and failed updates.

- **R4: Make Password Updates Unambiguous.** Provide a confirmation field or visibility toggle, missing on 21 and 43 of 96 PWC forms, respectively (ID8,11), visual feedback (ID24–26), and display the policy (ID10).
- **R5: Support and Test Password Manager Autofill.** Only 9/111 sites used autocomplete correctly (ID4,9).

Lower Priority: Optional ecosystem adoption.

- **R6: Implement the Change-Password URL.** Only 6/111 sites adopted this proposal. This redirect can remove the navigation hurdle of two to six steps.
- **R7: Evolve Standards Toward a Guarded API.** An API must preserve 2FA, RBA, and anti-abuse controls.

Developers, Frameworks, Operators, and Vendors. Our results suggest that design choices in websites and frameworks significantly contribute to end-user usability and automation capabilities, and that both can be substantially improved. Since developers struggle with usable security features and advice [1, 33, 46, 68], frameworks must ship secure and usable defaults. Hence, we recommend that web framework developers better support usability and the interaction between password managers and password change and reset processes (R4, R5). In particular, they should improve HTML attributes such as "name", "id", "autocomplete", "pattern", and even "passwordrules" to make them the new default, i.e., standardized form templates that follow best practices [7, 22, 97]. Frameworks should implement the "change-password" W3C draft (R6) [63]. In addition, server operators and web servers [6, 94] themselves could further lower the burden via dedicated directives, as improved tooling is known to increase adoption [44, 101]. Major browsers like Google Chrome could drive automated password reset adoption, mirroring their successful HTTPS standardization efforts that labeled HTTP as "not secure" [89].

An Extended Standard for Password Managers. While the "change-password" W3C draft aims to support websites, end-users, and password managers in deploying more usable password update processes, the standard has limitations and lacks widespread adoption. The standard is limited to redirecting end-users to the website's actual password change URL. Even where deployed, managers must still parse update forms that are often difficult to process automatically. Overall, our findings suggest that requiring password managers to interact with password change and reset processes primarily designed for end-user interactions has limited chances of success. Instead, we recommend extending the "change-password" W3C draft to a more powerful API that allows browsers and password managers to automatically discover a website's password change URL and complete the process (R7). An effective API must balance password policy compliance [19], server-side security controls, 2FA, and anti-abuse protections—a complex challenge requiring cryptographic, security, and HCI expertise. In particular, it must not weaken the gatekeepers discussed in Section 6.1 (user-initiated, rate-limited). More work is needed to better support developers in implementing password manager-friendly processes.

Recommendations for Future Research. Our findings revealed multiple patterns for PWC and PWR processes. While the considerable heterogeneity of patterns negatively impacts users' abilities to transfer experiences and knowledge between websites, it is also unclear which PWC and PWR processes provide end-users the best usability and security. As our study is descriptive, a user study should test the usability hypotheses derived from our measurements (e.g., whether deeper paths, missing confirmation fields, or ephemeral feedback increase errors and abandonment) and identify the most promising process designs. Previous work [47] has shown that end-users distrust password managers. Future research could focus on understanding and improving the acceptance of automated password changes, particularly in the context of browsers and password managers. Future work could also examine the root causes of the limited adoption of such standards and how future standards can be improved to increase adoption.

7 Conclusion

"*So what makes changing a password hard?*" Our analysis of 111 websites shows that the difficulty lies not only in the password itself, but in fragmented account-management processes. In security-relevant situations, such as suspected compromise, users face hidden entry points, inconsistent labels, unclear policies, and ambiguous outcomes. As a result, they may avoid password updates altogether. Our findings suggest that password updates should be treated as security-critical workflows. Websites should provide direct update paths, communicate requirements before users fail, clearly confirm successful changes, notify users out of band, and support expiring old sessions. They should also expose predictable, machine-readable interfaces for password managers. Yet, only a few websites implemented the well-known change-password URL or correctly supported password autofill across login and update forms. Secure password remediation, therefore, requires ecosystem-level support from developers, web frameworks, and standards bodies, so that password updates become reliable and usable when users need them most.

Acknowledgments

We remember our co-author Marten Oltrogge with deep gratitude. Marten made important contributions to the ideas, analysis, and writing of this paper. Sadly, he passed away before the paper was submitted and finalized. We dedicate this work to his memory; any remaining errors are our own.

This research was partly funded by VolkswagenStiftung Niedersächsisches Vorab (ZN3695) and the European Union as part of the program Digital Europe. Neither the European Union nor the other granting authorities can be held responsible. Any findings and opinions expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] Yasemin Acar, Michael Backes, Sascha Fahl, Simson Garfinkel, Doowon Kim, Michelle L. Mazurek, and Christian Stransky. Comparing the Usability of Cryptographic APIs. In *IEEE Symposium on Security and Privacy*, SP '17, pages 154–171, San Jose, California, USA, May 2017. IEEE.

[2] Suood Abdulaziz Al-Roomi and Frank Li. A Large-Scale Measurement of Website Login Policies. In *USENIX Security Symposium*, SSYM '23, pages 2061–2078, Anaheim, California, USA, August 2023. USENIX.

[3] Aniqa Alam, Elizabeth Stobert, and Robert Biddle. "This Is Different from the Western World": Understanding Password Sharing Among Young Bangladeshis. In *Workshop on Usable Security and Privacy*, USEC '23, San Diego, California, USA, February 2023. ISOC.

[4] Sabrina Amft, Sandra Höltervennhoff, Nicolas Huaman, Alexander Krause, Lucy Simko, Yasemin Acar, and Sascha Fahl. "We've Disabled MFA for You": An Evaluation of the Security and Usability of Multi-Factor Authentication Recovery Deployments. In *ACM Conference on Computer and Communications Security*, CCS '23, pages 3138–3152, Copenhagen, Denmark, November 2023. ACM.

[5] Simone Aonzo, Alessio Merlo, Giulio Tavella, and Yanick Fratantonio. Phishing Attacks on Modern Android. In *ACM Conference on Computer and Communications Security*, CCS '18, pages 1788–1801, Toronto, Ontario, Canada, October 2018. ACM.

[6] Apache Software Foundation, and Community. The Apache HTTP Server Project, April 1995. https://httpd.apache.org, as of June 17, 2026.

[7] Apple, Inc. Apple Developers: Password Rules, June 2018. https://developer.apple.com/password-rules/, as of June 17, 2026.

[8] Apple, Inc. Password Manager Resources, May 2020. https://opensource.apple.com/projects/password-manager-resources/, as of June 17, 2026.

[9] Adam J. Aviv, John T. Davin, Flynn Wolf, and Ravi Kuber. Towards Baselines for Shoulder Surfing on Mobile Authentication. In *Annual Conference on Computer Security Applications*, ACSAC '17, pages 486–498, Orlando, Florida, USA, December 2017. ACM.

[10] Hugh Beyer and Karen Holtzblatt. *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann Publishers, San Francisco, California, USA, 1 edition, 1997.

[11] Arkaprabha Bhattacharya, Alaa Daffalla, Kevin Lee, Rosanna Bellini, Nicola Dell, and Thomas Ristenpart. Inconsistent, Incomplete, and Insecure: A Survey of Account Security Interfaces. In *USENIX Security Symposium*, SSYM '26, Baltimore, Maryland, USA, August 2026. USENIX.

[12] Melanie Birks and Jane Mills. *Grounded Theory: A Practical Guide*. SAGE Publications, Thousand Oaks, California, USA, 3 edition, 2022.

[13] Joseph Bonneau, Elie Bursztein, Ilan Caron, Rob Jackson, and Mike Williamson. Secrets, Lies, and Account Recovery: Lessons from the Use of Personal Knowledge Questions at Google. In *The World Wide Web Conference*, WWW '15, pages 141–150, Florence, Italy, May 2015. ACM.

[14] Joseph Bonneau, Cormac Herley, Paul C. van Oorschot, and Frank Stajano. The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. In *IEEE Symposium on Security and Privacy*, SP '12, pages 553–567, San Francisco, California, USA, May 2012. IEEE.

[15] Thomas Brewster. Why You Shouldn't Panic About Dropbox Leaking 68 Million Passwords, August 2016. https://www.forbes.com/sites/thomasbrewster/2016/08/31/dropbox-hacked-but-its-not-that-bad/, as of June 17, 2026.

[16] Andre Büttner, Andreas Thue Pedersen, Stephan Wiefling, Nils Gruschka, and Luigi Lo Iacono. Is it Really You Who Forgot the Password? When Account Recovery Meets Risk-Based Authentication. In *Conference on Ubiquitous Security*, UbiSec '23, Exeter, United Kingdom, November 2023. Springer.

[17] Sonia Chiasson and Paul C. van Oorschot. Quantifying the Security Advantage of Password Expiration Policies. *Designs, Codes and Cryptography*, 77(2–3):401–408, December 2015.

[18] Alaa Daffalla, Marina Bohuk, Nicola Dell, Rosanna Bellini, and Thomas Ristenpart. Account Security Interfaces: Important, Unintuitive, and Untrustworthy. In *USENIX Security Symposium*, SSYM '23, pages 3601–3618, Anaheim, California, USA, August 2023. USENIX.

[19] Dashlane, Inc. Password Changer .well-known [DRAFT], April 2021. https://github.com/Dashlane/password-changer-well-known, as of June 17, 2026.

[20] Dashlane, Inc. Dashlane Web Extension Patch Notes: September 15, 2022 – 6.2232, September 2022. https://support.dashlane.com/hc/en-us/articles/360014075119-Dashlane-release-notes, as of June 17, 2026.

[21] Nurullah Demir, Matteo Große-Kampmann, Tobias Urban, Christian Wressnegger, Thorsten Holz, and Norbert Pohlmann. Reproducibility and Replicability of Web Measurement Studies. In *The World Wide Web Conference*, WWW '22, pages 533–544, Lyon, France, April 2022. ACM.

[22] Django Software Foundation. Django Python Web Framework: Authentication Form, February 2026. https://github.com/django/django/blob/main/django/contrib/auth/forms.py, as of June 17, 2026.

[23] Filipe Espósito. iCloud Keychain Now Alerts Users about Leaked Passwords, July 2020. https://9to5mac.com/2020/07/04/ios-14-icloud-keychain-now-alerts-users-about-leaked-passwords-more/, as of June 17, 2026.

[24] Michael Fey. Watchtower Notifications: Timely Security Alerts for the Websites You Use, July 2020. https://blog.1password.com/announcing-watchtower-notifications/, as of June 17, 2026.

[25] Dinei Florêncio, Cormac Herley, and Paul C. van Oorschot. An Administrator's Guide to Internet Password Research. In *Large Installation System Administration Conference*, LISA '14, pages 44–61, Seattle, Washington, USA, November 2014. USENIX.

[26] Dinei Florêncio, Cormac Herley, and Paul C. van Oorschot. Password Portfolios and the Finite-Effort User: Sustainably Managing Large Numbers of Accounts. In *USENIX Security Symposium*, SSYM '14, pages 575–590, San Diego, California, USA, August 2014. USENIX.

[27] Lorenzo Franceschi-Bicchierai. Another Day, Another Hack: 117 Million LinkedIn Emails and Passwords, May 2016. https://www.vice.com/en/article/78kk4z/another-day-another-hack-117-million-linkedin-emails-and-password, as of June 17, 2026.

[28] Anuj Gautam, Shan Lalani, and Scott Ruoti. Improving Password Generation Through the Design of a Password Composition Policy Description Language. In *Symposium on Usable Privacy and Security*, SOUPS '22, pages 541–560, Boston, Massachusetts, USA, August 2022. USENIX.

[29] Maximilian Golla and Markus Dürmuth. Analyzing 4 Million Real-World Personal Knowledge Questions (Short Paper). In *International Conference on Passwords*, PASSWORDS '15, pages 39–44, Cambridge, United Kingdom, December 2015. Springer.

[30] Maximilian Golla and Markus Dürmuth. On the Accuracy of Password Strength Meters. In *ACM Conference on Computer and Communications Security*, CCS '18, pages 1567–1582, Toronto, Ontario, Canada, October 2018. ACM.

[31] Maximilian Golla, Miranda Wei, Juliette Hainline, Lydia Filipe, Markus Dürmuth, Elissa Redmiles, and Blase Ur. "What was that site doing with my Facebook password?" Designing Password-Reuse Notifications. In *ACM Conference on Computer and Communications Security*, CCS '18, pages 1549–1566, Toronto, Ontario, Canada, October 2018. ACM.

[32] Google, Inc. Google Translate Documents & Websites, June 2026. https://support.google.com/translate/answer/2534559, as of June 17, 2026.

[33] Marco Gutfleisch, Jan H. Klemmer, Niklas Busch, Yasemin Acar, M. Angela Sasse, and Sascha Fahl. How Does Usable Security (Not) End Up in Software Products? Results From a Qualitative Interview Study. In *IEEE Symposium on Security and Privacy*, SP '22, pages 893–910, San Francisco, California, USA, May 2022. IEEE.

[34] Hana Habib, Pardis Emami Naeini, Summer Devlin, Maggie Oates, Chelse Swoopes, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. User Behaviors and Attitudes Under Password Expiration Policies. In *Symposium on Usable Privacy and Security*, SOUPS '18, pages 13–30, Baltimore, Maryland, USA, August 2018. USENIX.

[35] Moritz Horsch, Mario Schlipf, Johannes Braun, and Johannes Buchmann. Password Requirements Markup Language. In *Australasian Conference on Information Security and Privacy*, ACISP '16, pages 426–439, Melbourne, Victoria, Australia, July 2016. Springer.

[36] Nicolas Huaman, Sabrina Amft, Marten Oltrogge, Yasemin Acar, and Sascha Fahl. They Would Do Better If They Worked Together: The Case of Interaction Problems between Password Managers and Websites. In *IEEE Symposium on Security and Privacy*, SP '21, pages 1367–1381, Virtual Conference, May 2021. IEEE.

[37] Nicolas Huaman, Alexander Krause, Dominik Wermke, Jan H. Klemmer, Christian Stransky, Yasemin Acar, and Sascha Fahl. If You Can't Get Them to the Lab: Evaluating a Virtual Study Environment with Security Information Workers. In *Symposium on Usable Privacy and Security*, SOUPS '22, pages 313–330, Boston, Massachusetts, USA, August 2022. USENIX.

[38] Yue Huang, Borke Obada-Obieh, and Konstantin Beznosov. Users' Perceptions of Chrome Compromised Credential Notification. In *Symposium on Usable Privacy and Security*, SOUPS '22, pages 155–174, Boston, Massachusetts, USA, August 2022. USENIX.

[39] Troy Hunt. *Have I Been Pwned?* – Check If Your Email Has Been Compromised in a Data Breach, December 2013. https://haveibeenpwned.com, as of June 17, 2026.

[40] Troy Hunt. Password Reuse, Credential Stuffing and Another Billion Records in *Have I Been Pwned?*, May 2017. https://www.troyhunt.com/password-reuse-credential-stuffing-and-another-1-billion-records-in-have-i-been-pwned/, as of June 17, 2026.

[41] Troy Hunt. *Have I Been Pwned?* – Pwned Passwords v3 is Now Live!, July 2018. https://www.troyhunt.com/pwned-passwords-v3-is-now-live/, as of June 17, 2026.

[42] Philip G. Inglesant and M. Angela Sasse. The True Cost of Unusable Password Policies: Password Use in the Wild. In *ACM Conference on Human Factors in Computing Systems*, CHI '10, pages 383–392, Atlanta, Georgia, USA, April 2010. ACM.

[43] Emiram Kablo, Katharina Kader, and Patricia Arias-Cabarcos. "I'm actually going to go and change these passwords": Analyzing the Usability of Credential Audit Interfaces in Password Managers. In *ACM Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '24, pages 2:1–2:13, Maui, Hawaii, USA, May 2024. ACM.

[44] Eiji Kitamura. Help Users Change Passwords Easily by Adding a Well-Known URL for Changing Passwords, September 2020. https://web.dev/articles/change-password-url, as of June 17, 2026.

[45] David Klein, Marius Musch, Thomas Barber, Moritz Kopmann, and Martin Johns. Accept All Exploits: Exploring the Security Impact of Cookie Banners. In *Annual Conference on Computer Security Applications*, ACSAC '22, pages 911–922, Austin, Texas, USA, December 2022. ACM.

[46] Jan H. Klemmer, Marco Gutfleisch, Christian Stransky, Yasemin Acar, M. Angela Sasse, and Sascha Fahl. "Make Them Change it Every Week!": A Qualitative Exploration of Online Developer Advice on Usable and Secure Authentication. In *ACM Conference on Computer and Communications Security*, CCS '23, pages 2740–2754, Copenhagen, Denmark, November 2023. ACM.

[47] Sabrina Klivan, Sandra Höltervennhoff, Nicolas Huaman, Yasemin Acar, and Sascha Fahl. "Would You Give the Same Priority to the Bank and a Game? I Do Not!" Exploring Credential Management Strategies and Obstacles during Password Manager Setup. In *Symposium on Usable Privacy and Security*, SOUPS '23, pages 171–190, Anaheim, California, USA, August 2023. USENIX.

[48] Alexander Krause, Jan H. Klemmer, Nicolas Huaman, Dominik Wermke, Yasemin Acar, and Sascha Fahl. Pushed by Accident: A Mixed-Methods Study on Strategies of Handling Secret Information in Source Code Repositories. In *USENIX Security Symposium*, SSYM '23, pages 2527–2544, Anaheim, California, USA, August 2023. USENIX.

[49] Alexander Krause, Jacques Suray, Lea Schmüser, Marten Oltrogge, Oliver Wiese, Maximilian Golla, and Sascha Fahl. Replication Package: An Analysis of the Security, Usability, and Automation Capabilities of Password Update Processes on Top-Ranked Websites, June 2026. https://doi.org/10.6084/m9.figshare.28741874, as of June 17, 2026.

[50] Leona Lassak, Annika Hildebrandt, Maximilian Golla, and Blase Ur. "It's Stored, Hopefully, on an Encrypted Server": Mitigating Users' Misconceptions About FIDO2 Biometric WebAuthn. In *USENIX Security Symposium*, SSYM '21, pages 91–108, Virtual Conference, August 2021. USENIX.

[51] Leona Lassak, Philipp Markert, Maximilian Golla, Elizabeth Stobert, and Markus Dürmuth. A Comparative Long-Term Study of Fallback Authentication Schemes. In *ACM Conference on Human Factors in Computing Systems*, CHI '24, pages 970:1–970:19, Maui, Hawaii, USA, May 2024. ACM.

[52] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *Symposium on Network and Distributed System Security*, NDSS '19, San Diego, California, USA, February 2019. ISOC.

[53] Kevin Lee, Sten Sjöberg, and Arvind Narayanan. Password Policies of Most Top Websites Fail to Follow Best Practices. In *Symposium on Usable Privacy and Security*, SOUPS '22, pages 561–580, Boston, Massachusetts, USA, August 2022. USENIX.

[54] Lucy Li, Bijeeta Pal, Junade Ali, Nick Sullivan, Rahul Chatterjee, and Thomas Ristenpart. Protocols for Checking Compromised Credentials. In *ACM Conference on Computer and Communications Security*, CCS '19, pages 1387–1403, London, United Kingdom, November 2019. ACM.

[55] Sanam Ghorbani Lyastani, Michael Schilling, Sascha Fahl, Michael Backes, and Sven Bugiel. "Better managed than memorized?" Studying the Impact of Managers on Password Strength and Reuse. In *USENIX Security Symposium*, SSYM '18, pages 203–220, Baltimore, Maryland, USA, August 2018. USENIX.

[56] Sanam Ghorbani Lyastani, Michael Schilling, Michaela Neumayr, Michael Backes, and Sven Bugiel. Is FIDO2 the Kingslayer of User Authentication? A Comparative Usability Study of FIDO2 Passwordless Authentication. In *IEEE Symposium on Security and Privacy*, SP '20, pages 268–285, Virtual Conference, May 2020. IEEE.

[57] Philipp Markert, Leona Lassak, Maximilian Golla, and Markus Dürmuth. Understanding Users' Interaction with Login Notifications. In *ACM Conference on Human Factors in Computing Systems*, CHI '24, pages 853:1–853:17, Maui, Hawaii, USA, May 2024. ACM.

[58] Peter Mayer, Collins W. Munyendo, Michelle L. Mazurek, and Adam J. Aviv. Why Users (Don't) Use Password Managers at a Large Educational Institution. In *USENIX Security Symposium*, SSYM '22, pages 1849–1866, Boston, Massachusetts, USA, August 2022. USENIX.

[59] Peter Mayer, Yixin Zou, Florian Schaub, and Adam J. Aviv. "Now I'm a bit angry:" Individuals' Awareness, Perception, and Responses to Data Breaches that Affected Them. In *USENIX Security Symposium*, SSYM '21, pages 393–410, Virtual Conference, August 2021. USENIX.

[60] Nora McDonald, Sarita Schoenebeck, and Andrea Forte. Reliability and Inter-Rater Reliability in Qualitative Research: Norms and Guidelines for CSCW and HCI Practice. In *ACM Conference on Computer-Supported Cooperative Work and Social Computing*, CSCW '19, pages 72:1–72:23, Austin, Texas, USA, November 2019. ACM.

[61] William Melicher, Darya Kurilova, Sean M. Segreti, Pranshu Kalvani, Richard Shay, Blase Ur, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Michelle L. Mazurek. Usability and Security of Text Passwords on Mobile Devices. In *ACM Conference on Human Factors in Computing Systems*, CHI '16, pages 527–539, San Jose, California, USA, May 2016. ACM.

[62] Rolf Molich and Jakob Nielsen. Improving a Human-Computer Dialogue. *Communications of the ACM*, 33(3):338–348, March 1990.

[63] Ricky Mondello and Theresa O'Connor. W3C Working Draft: A Well-Known URL for Changing Passwords, June 2024. https://www.w3.org/TR/change-password-url/, as of June 17, 2026.

[64] Alena Naiakshina, Anastasia Danilova, Eva Gerlitz, Emanuel von Zezschwitz, and Matthew Smith. "If You Want, I Can Store the Encrypted Password": A Password-Storage Field Study with Freelance Developers. In *ACM Conference on Human Factors in Computing Systems*, CHI '19, pages 140:1–140:12, Glasgow, Scotland, United Kingdom, May 2019. ACM.

[65] National Cyber Security Centre. The Problems with Forcing Regular Password Expiry, December 2016. https://www.ncsc.gov.uk/articles/problems-forcing-regular-password-expiry, as of June 17, 2026.

[66] Lorenzo Neil, Elijah Bouma-Sims, Evan Lafontaine, Yasemin Acar, and Bradley Reaves. Investigating Web Service Account Remediation Advice. In *Symposium on Usable Privacy and Security*, SOUPS '21, pages 359–376, Virtual Conference, August 2021. USENIX.

[67] Casey Newton. Dashlane Can Now Change All Your Passwords with a Single Click, and It's Amazing, December 2014. https://www.theverge.com/2014/12/9/7357251/, as of June 17, 2026.

[68] Duc Cuong Nguyen, Dominik Wermke, Yasemin Acar, Michael Backes, Charles Weir, and Sascha Fahl. A Stitch in Time: Supporting Android Developers in WritingSecure Code. In *ACM Conference on Computer and Communications Security*, CCS '17, pages 1065–1077, Dallas, Texas, USA, October 2017. ACM.

[69] Nick Nguyen. Introducing Firefox Monitor: Helping People Take Control After a Data Breach, September 2018. https://blog.mozilla.org/en/products/firefox/introducing-firefox-monitor-helping-people-take-control-after-a-data-breach/, as of June 17, 2026.

[70] Jakob Nielsen. Enhancing the Explanatory Power of Usability Heuristics. In *ACM Conference on Human Factors in Computing Systems*, CHI '94, pages 152–158, Boston, Massachusetts, USA, April 1994. ACM.

[71] Jakob Nielsen. Usability Inspection Methods. In *ACM Conference on Human Factors in Computing Systems*, CHI '95, pages 377–378, Denver, Colorado, USA, May 1995. ACM.

[72] Jakob Nielsen and Rolf Molich. Heuristic Evaluation of User Interfaces. In *ACM Conference on Human Factors in Computing Systems*, CHI '90, pages 249–256, Seattle, Washington, USA, March 1990. ACM.

[73] Alexandra Nisenoff, Maximilian Golla, Miranda Wei, Juliette Hainline, Hayley Szymanek, Annika Braun, Annika Hildebrandt, Blair Christensen, David Langenberg, and Blase Ur. A Two-Decade Retrospective Analysis of a University's Vulnerability to Attacks Exploiting Reused Passwords. In *USENIX Security Symposium*, SSYM '23, pages 5127–5144, Anaheim, California, USA, August 2023. USENIX.

[74] Mark Nottingham. Well-Known Uniform Resource Identifiers (URIs). RFC 8615, RFC Editor, May 2019.

[75] Borke Obada-Obieh, Yue Huang, and Konstantin Beznosov. The Burden of Ending Online Account Sharing. In *ACM Conference on Human Factors in Computing Systems*, CHI '20, pages 503:1–503:13, Honolulu, Hawaii, USA, April 2020. ACM.

[76] Theresa O'Connor. A Well-Known URL for Changing Passwords, November 2018. https://github.com/w3c/webappsec-change-password-url, as of June 17, 2026.

[77] Sean Oesch and Scott Ruoti. That Was Then, This Is Now: A Security Evaluation of Password Generation, Storage, and Autofill in Browser-Based Password Managers. In *USENIX Security Symposium*, SSYM '20, pages 2165–2182, Virtual Conference, August 2020. USENIX.

[78] Sean Oesch, Scott Ruoti, James Simmons, and Anuj Gautam. "It Basically Started Using Me:" An Observational Study of Password Manager Usage. In *ACM Conference on Human Factors in Computing Systems*, CHI '22, pages 33:1–33:23, New Orleans, Louisiana, USA, April 2022. ACM.

[79] Sanghak Oh, Heewon Baek, Jun Ho Huh, Taeyoung Kim, Woojin Jeon, Ian Oakley, and Hyoungshick Kim. Understanding and Improving User Adoption and Security Awareness in Password Checkup Services. In *ACM Conference on Human Factors in Computing Systems*, CHI '25, pages 386:1–386:32, Yokohama, Japan, April 2025. ACM.

[80] Bijeeta Pal, Mazharul Islam, Marina Sanusi, Nick Sullivan, Luke Valenta, Tara Whalen, Christopher A. Wood, Thomas Ristenpart, and Rahul Chatterjee. Might I Get Pwned: A Second Generation Compromised Credential Checking Service. In *USENIX Security Symposium*, SSYM '22, pages 1831–1848, Boston, Massachusetts, USA, August 2022. USENIX.

[81] Dario Pasquini, Danilo Francati, Giuseppe Ateniese, and Evgenios M. Kornaropoulos. Breach Extraction Attacks: Exposing and Addressing the Leakage in Second Generation Compromised Credential Checking Services. In *IEEE Symposium on Security and Privacy*, SP '24, pages 1405–1423, San Francisco, California, USA, May 2024. IEEE.

[82] Sarah Pearman, Jeremy Thomas, Pardis Emami Naeini, Hana Habib, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Serge Egelman, and Alain Forget. Let's Go in for a Closer Look: Observing Passwords in Their Natural Habitat. In *ACM Conference on Computer and Communications Security*, CCS '17, pages 295–310, Dallas, Texas, USA, October 2017. ACM.

[83] Sarah Pearman, Shikun Aerin Zhang, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Why People (Don't) Use Password Managers Effectively. In *Symposium on Usable Privacy and Security*, SOUPS '19, pages 319–338, Santa Clara, California, USA, August 2019. USENIX.

[84] Kara Pernice. Affinity Diagramming for Collaboratively Sorting UX Findings and Design Ideas, February 2018. https://www.nngroup.com/articles/affinity-diagram/, as of June 17, 2026.

[85] Jay Peters. Dashlane Is Giving Its One-Click Password Changer a Big Upgrade, March 2021. https://www.theverge.com/2021/3/11/22320467/, as of June 17, 2026.

[86] Jennifer Pullman, Kurt Thomas, and Elie Bursztein. Password Checkup: Protect Your Accounts From Data Breaches With Password Checkup, February 2019. https://security.googleblog.com/2019/02/protect-your-accounts-from-data.html, as of June 17, 2026.

[87] Hirak Ray, Flynn Wolf, Ravi Kuber, and Adam J. Aviv. Why Older Adults (Don't) Use Password Managers. In *USENIX Security Symposium*, SSYM '21, pages 73–90, Virtual Conference, August 2021. USENIX.

[88] Robert W. Reeder and Stuart Schechter. When the Password Doesn't Work: Secondary Authentication for Websites. *IEEE Security & Privacy*, 9(2):43–49, March 2011.

[89] Emily Schechter. A Milestone for Chrome Security: Marking HTTP as "Not Secure", July 2018. https://blog.google/products/chrome/milestone-chrome-security-marking-http-not-secure/, as of June 17, 2026.

[90] Lisa Seeman and Michael Cooper. W3C Working Draft: Cognitive Accessibility Roadmap and Gap Analysis, December 2018. https://www.w3.org/TR/coga-gap-analysis/, as of June 17, 2026.

[91] Yunpeng Song, Cori Faklaris, Zhongmin Cai, Jason I. Hong, and Laura Dabbish. Normal and Easy: Account Sharing Practices in the Workplace. In *ACM Conference on Computer-Supported Cooperative Work and Social Computing*, CSCW '19, pages 83:1–83:25, Austin, Texas, USA, November 2019. ACM.

[92] Elizabeth Stobert and Robert Biddle. The Password Life Cycle: User Behaviour in Managing Passwords. In *Symposium on Usable Privacy and Security*, SOUPS '14, pages 243–255, Menlo Park, California, USA, July 2014. USENIX.

[93] Caden Sumner. Any Plans of Reviving Auto Change Passwords Feature?, August 2023. https://www.reddit.com/r/Dashlane/comments/160z0be/comment/jyk1nv8/, as of June 17, 2026.

[94] Igor Sysoev and Community. NGINX: Advanced Load Balancer, Web Server, & Reverse Proxy, October 2004. https://nginx.org, as of June 17, 2026.

[95] Joshua Tan, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Practical Recommendations for Stronger, More Usable Passwords Combining Minimum-Strength, Minimum-Length, and Blocklist Requirements. In *ACM Conference on Computer and Communications Security*, CCS '20, pages 1407–1426, Virtual Conference, October 2020. ACM.

[96] David Temoshok, James L. Fenton, Yee-Yin Choong, Naomi Lefkovitz, Andrew Regenscheid, Ryan Galluzzo, and Justin P. Richer. Digital Identity Guidelines – Authentication and Authenticator Management: NIST Special Publication 800-63B, Revision 4, July 2025.

[97] The Chromium Projects. Password Form Styles that Chromium Understands: Comprehensible Password Forms, June 2016. https://www.chromium.org/developers/design-documents/form-styles-that-chromium-understands/, as of June 17, 2026.

[98] Sam Thielman. Yahoo Hack: 1bn Accounts Compromised by Biggest Data Breach in History, December 2016. https://www.theguardian.com/technology/2016/dec/14/yahoo-hack-security-of-one-billion-accounts-breached, as of June 17, 2026.

[99] Kurt Thomas, Frank Li, Ali Zand, Jacob Barrett, Juri Ranieri, Luca Invernizzi, Yarik Markov, Oxana Comanescu, Vijay Eranti, Angelika Moscicki, Daniel Margolis, Vern Paxson, and Elie Bursztein. Data Breaches, Phishing, or Malware? Understanding the Risks of Stolen Credentials. In *ACM Conference on Computer and Communications Security*, CCS '17, pages 1421–1434, Dallas, Texas, USA, October 2017. ACM.

[100] Kurt Thomas, Jennifer Pullman, Kevin Yeo, Ananth Raghunathan, Patrick Gage Kelley, Luca Invernizzi, Borbala Benko, Tadek Pietraszek, Sarvar Patel, Dan Boneh, and Elie Bursztein. Protecting Accounts from Credential Stuffing with Password Breach Alerting. In *USENIX Security Symposium*, SSYM '19, pages 1556–1571, Santa Clara, California, USA, August 2019. USENIX.

[101] Christian Tiefenau, Emanuel von Zezschwitz, Maximilian Häring, Katharina Krombholz, and Matthew Smith. A Usability Evaluation of Let's Encrypt and Certbot: Usable Security Done Right. In *ACM Conference on Computer and Communications Security*, CCS '19, pages 1971–1988, London, United Kingdom, November 2019. ACM.

[102] Blase Ur, Felicia Alfieri, Maung Aung, Lujo Bauer, Nicolas Christin, Jessica Colnago, Lorrie Faith Cranor, Henry Dixon, Pardis Emami Naeini, Hana Habib, Noah Johnson, and William Melicher. Design and Evaluation of a Data-Driven Password Meter. In *ACM Conference on Human Factors in Computing Systems*, CHI '17, pages 3775–3786, Denver, Colorado, USA, May 2017. ACM.

[103] Blase Ur, Fumiko Noma, Jonathan Bees, Sean M. Segreti, Richard Shay, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. "I Added '!' at the End to Make It Secure": Observing Password Creation in the Lab. In *Symposium on Usable Privacy and Security*, SOUPS '15, pages 123–140, Ottawa, Ontario, Canada, July 2015. USENIX.

[104] Cathy Urquhart. *Grounded Theory for Qualitative Research: A Practical Guide*. SAGE Publications, Thousand Oaks, California, USA, 1 edition, 2013.

[105] Arin Waichulis. Security Bite: Apple's Most Impressive Agentic AI Feature Yet Is Hiding in the Passwords App, June 2026. https://9to5mac.com/2026/06/11/security-bite-apples-most-impressive-agentic-ai-feature-yet-is-hiding-in-the-passwords-app/, as of June 17, 2026.

[106] Kathryn Walsh, Faiza Tazi, Philipp Markert, and Sanchari Das. My Account Is Compromised – What Do I Do? Towards an Intercultural Analysis of Account Remediation for Websites. In *Workshop on Inclusive Privacy and Security*, WIPS '21, Virtual Conference, August 2021. USENIX.

[107] Rick Wash, Emilee J. Rader, Ruthie Berman, and Zac Wellmer. Understanding Password Choices: How Frequently Entered Passwords are Re-used Across Websites. In *Symposium on Usable Privacy and Security*, SOUPS '16, pages 175–188, Denver, Colorado, USA, July 2016. USENIX.

[108] Dominik Wermke, Jan H. Klemmer, Noah Wöhler, Juliane Schmüser, Harshini Sri Ramulu, Yasemin Acar, and Sascha Fahl. "Always Contribute Back": A Qualitative Study on Security Challenges of the Open Source Supply Chain. In *IEEE Symposium on Security and Privacy*, SP '23, pages 1545–1560, San Francisco, California, USA, May 2023. IEEE.

[109] Dominik Wermke, Noah Wöhler, Jan H. Klemmer, Marcel Fourné, Yasemin Acar, and Sascha Fahl. Committed to Trust: A Qualitative Study on Security & Trust in Open Source Software Projects. In *IEEE Symposium on Security and Privacy*, SP '22, pages 1880–1896, San Francisco, California, USA, May 2022. IEEE.

[110] Stephan Wiefling, Luigi Lo Iacono, and Markus Dürmuth. Is This Really You? An Empirical Study on Risk-Based Authentication Applied in the Wild. In *International Conference on ICT Systems Security and Privacy Protection*, IFIP SEC '19, pages 134–148, Lisbon, Portugal, June 2019. IFIP.

[111] Samira Zibaei, Dinah Rinoa Malapaya, Benjamin Mercier, Amirali Salehi-Abari, and Julie Thorpe. Do Password Managers Nudge Secure (Random) Passwords? In *Symposium on Usable Privacy and Security*, SOUPS '22, pages 581–597, Boston, Massachusetts, USA, August 2022. USENIX.

[112] Yixin Zou, Shawn Danino, Kaiwen Sun, and Florian Schaub. You 'Might' Be Affected: An Empirical Analysis of Readability and Usability Issues in Data Breach Notifications. In *ACM Conference on Human Factors in Computing Systems*, CHI '19, pages 194:1–194:14, Glasgow, Scotland, United Kingdom, May 2019. ACM.

[113] Yixin Zou and Florian Schaub. Beyond Mandatory: Making Data Breach Notifications Useful for Consumers. *IEEE Security & Privacy*, 17(2):67–72, March 2019.

A Analyzed Websites

Table 3: Full list of analyzed websites, their Tranco rank, and website category (extracted from alphamountain.ai).

Rank	Website	Category	Rank	Website	Category	Rank	Website	Category
1	google.com	Search Engines/Portals	140	quora.com	Reference	283	walmart.com	Shopping
7	twitter.com	Social Networking	141	stackoverflow.com	Information Technology	286	cnbc.com	Finance
12	apple.com	Information Technology	145	roblox.com	Games	288	hubspot.com	Information Technology
13	cloudflare.com	Information Technology	147	theguardian.com	News	290	unsplash.com	Media Sharing
14	linkedin.com	Social Networking	150	bbc.com	News	298	unity3d.com	Information Technology
16	wikipedia.org	Reference	152	spankbang.com	Pornography	299	mediafire.com	File Sharing/Storage
18	live.com	Search Engines/Portals	159	sciencedirect.com	Education	302	espn.com	Sports
20	amazon.com	Shopping	162	etsy.com	Shopping	316	onlyfans.com	Adult/Mature
31	github.com	Information Technology	163	twitch.tv	Video/Multimedia	319	deviantart.com	Arts/Culture
32	pinterest.com	Social Networking	166	sourceforge.net	Software Downloads	327	dailymotion.com	Video/Multimedia
33	reddit.com	Forums	168	imgur.com	Media Sharing	330	grammarly.com	Productivity Applications
34	wordpress.org	Information Technology	171	shopify.com	Information Technology	332	figma.com	Information Technology
41	fastly.net	Content Servers	174	hp.com	Information Technology	333	ft.com	Finance
42	zoom.us	Virtual Meetings	181	dell.com	Information Technology	335	fiverr.com	Business/Economy
44	mail.ru	Search Engines/Portals	186	oracle.com	Information Technology	337	w3schools.com	Education
46	adobe.com	Information Technology	192	epicgames.com	Games	342	eventbrite.com	Entertainment
48	yandex.ru	Search Engines/Portals	199	issuu.com	Productivity Applications	347	snapchat.com	Social Networking
54	gandi.net	Hosting	200	tradingview.com	Brokerage/Trading	351	npr.org	News
56	intuit.com	Finance	211	deepl.com	Translation	358	time.com	News
66	tiktok.com	Social Networking	216	discord.com	Chat/IM/SMS	361	surveymonkey.com	Business/Economy
78	pornhub.com	Pornography	218	pixiv.net	Arts/Culture	364	chaturbate.com	Pornography
82	paypal.com	Finance	222	alibaba.com	Shopping	365	pixabay.com	Media Sharing
83	tumblr.com	Social Networking	223	washingtonpost.com	News	368	telegraph.co.uk	News
96	nytimes.com	News	226	reuters.com	News	371	goodreads.com	Reference
97	fandom.com	Entertainment	229	ibm.com	Information Technology	372	scribd.com	Reference
99	dropbox.com	File Sharing/Storage	231	samsung.com	Business/Economy	380	notion.so	Information Technology
105	ebay.com	Auctions/Classifieds	238	bloomberg.com	Finance	381	yelp.com	Reference
106	webex.com	Virtual Meetings	245	wix.com	Hosting	386	no-ip.com	Dynamic DNS
108	imdb.com	Entertainment	248	atlassian.com	Information Technology	387	sentry.io	Information Technology
109	flickr.com	Media Sharing	250	linktr.ee	Information Technology	389	name.com	Hosting
113	naver.com	Search Engines/Portals	252	reg.ru	Hosting	392	ted.com	Education
120	soundcloud.com	Audio	267	stripe.com	Finance	395	themeforest.net	Information Technology
121	digicert.com	Information Technology	275	nature.com	Education	396	statista.com	Reference
122	aliexpress.com	Shopping	276	ilovepdf.com	Productivity Applications	397	wired.com	News
127	xhamster.com	Pornography	278	kaspersky.com	Information Technology	400	typeform.com	Productivity Applications
129	archive.org	Reference	281	businessinsider.com	News	401	binance.com	Finance
137	meraki.com	Information Technology	282	cloudns.net	Hosting	402	rakuten.co.jp	Shopping

B Used Quantifiers

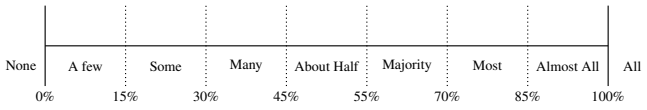


Figure 5: Quantifiers used throughout our results.

C Example Screenshots

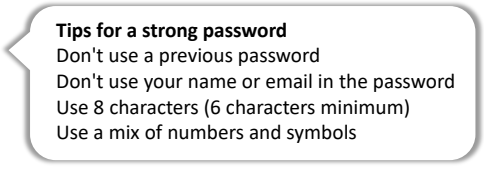


Figure 6: Advice when changing the password on etsy.com.

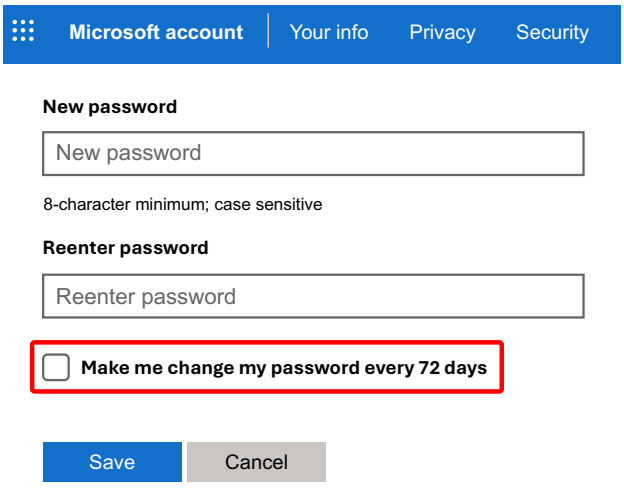


Figure 7: Optional expiration setting on microsoft.com.