Measuring the Prevalence of Password Manager Issues Using In-Situ Experiments

Adryana Hutchinson*
The George Washington University
adryana.hutchinson@gwu.edu

Jinwei Tang
Clark University
JeTang@clarku.edu

Adam J. Aviv
The George Washington University
aaviv@gwu.edu

Peter Story†
Clark University
PeStory@clarku.edu

***Abstract*—To protect their security, users are instructed to use unique passwords for all their accounts. Password managers make this possible, as they can generate, store, and autofill passwords within a user's browser. Unfortunately, prior work has identified usability issues which may deter users from using password managers. In this paper, we measure the prevalence of usability issues affecting four popular password managers (Chrome, Safari, Bitwarden, and Keeper). We tested these password managers with their out-of-the-box settings on 60 randomly sampled websites. We show that users are likely to encounter issues using password managers during account registration and authentication. We found that usability issues were widespread, but varied by password manager. Common issues included password managers not prompting the user to generate passwords, autofilling web forms incorrectly or not at all, and generating passwords that were incompatible with websites' password policies. We found that Chrome and Safari had fewer interaction issues than the other password managers we tested. We conclude by suggesting ways that websites and password managers can improve their compatibility with each other. For example, we recommend that password managers tailor their passwords to websites' requirements (like Chrome and Safari), or adopt alphanumeric-only password generation by default (like Bitwarden).**

I. Introduction

Password reuse is a major security issue: if reused credentials are stolen in a data breach, these credentials can be used to log into other accounts (i.e., a credential stuffing attack) [47], [27], [26], [50]. The common security advice to "avoid reusing passwords" seems reasonable, but is often ignored because of the unrealistic demands it places on users' time [31], [17], [10], [48], [38]. Password managers offer a potential solution, since they can automate the challenging tasks of creating and recalling unique passwords. When used properly, password managers can mitigate the risks of credential stuffing attacks. Unfortunately, there are known usability issues that impede password manager effectiveness [32], [33], [19], [25]. For example, some websites impose password composition policies that reject passwords suggested by password managers, discouraging use of these unique passwords. Prior work has described various usability issues that affect password managers, but has not measured the prevalence of these issues from a user's perspective. Are usability issues rare, or is a typical user likely to encounter them? If usability issues are common, real-world improvements to security may depend on fixing the most significant issues.

Password managers and websites interact with each other in complex ways, so capturing the full range of potential usability issues requires manually testing password managers on actual websites. To measure the prevalence of usability issues, we tested a diverse set of password managers (Chrome, Safari, Bitwarden, and Keeper) on a random sample of websites. Prior work suggests that users rarely change password managers' default settings, so we tested all password managers in their default configurations [29]. During testing, we recorded any behavior that interfered with account registration and authentication. We especially focused on issues related to websites' password composition policies, since prior work identified password policies as a source of usability issues [32], [12], [6], [36], [18].

We answer the following research questions:

RQ1 During registration and authentication, how prevalent are usability issues when using password managers out-of-the-box?

RQ2 Which password generation approaches can password managers adopt to maximize their compatibility with websites?

RQ3 Which password policies can websites adopt to maximize their compatibility with password managers?

Of the 100 websites we tested, we successfully created our own credentials on 60 websites, and one website provided us with default credentials. Our testing shows that usability issues are common: for the four password managers we tested, we encountered some kind of usability issue on more than a quarter of websites (§ IV). However, different password managers were affected by different usability issues. For example, sometimes password managers did not prompt users to generate credentials during account registration (§ IV-A). This issue occurred on 12 websites when using Chrome, but on only 3 websites when using Safari. Also, of the 60 websites on which we generated credentials using password managers, 19 websites (32%) rejected passwords from at least one of the four password managers we tested. Passwords generated by Keeper were rejected most often (on 13 websites), followed by Bitwarden (on eight websites), Safari (on seven websites), and

*Adryana Hutchinson was formerly an undergraduate at Clark University.
†Corresponding author: Dr. Peter Story of Clark University.

Symposium on Usable Security and Privacy (USEC) 2024
26 February 2024, San Diego, CA, USA
ISBN 979-8-9894372-5-2
https://dx.doi.org/10.14722/usec.2024.23094
www.ndss-symposium.org, https://www.usablesecurity.net/USEC/

finally Chrome, which was only rejected on a single website. Our results suggest that usability issues are widespread, and play a major part in a typical user's experience.

In our discussion, we offer recommendations for the developers of websites and password managers to improve their compatibility with each other (§ VI). In particular, we recommend that websites adopt more flexible password composition policies, in accordance with NIST's latest digital identity guidelines [16]. Also, we recommend that password managers adopt the password generation approach used by Chrome, which uses crowdsourcing to adapt to websites' password composition policies. Other approaches to learning password policies may also be effective [2]. Alternatively, password managers can adopt Bitwarden's simpler but less effective approach of omitting punctuation from password suggestions by default, since this causes fewer issues than including punctuation by default. Finally, based on the variability of password managers' usability, we suggest that security experts steer users towards the most secure and usable password managers.

II. Related Work

A. Password Manager Usability

Password managers have the potential to decrease password reuse and password guessability, thereby protecting peoples' accounts from unauthorized access. However, for a variety of reasons, password managers do not always realize this potential. Pearman et al. conducted interviews to understand how people manage their passwords [32]. Participants described password managers saving usernames and passwords incorrectly and suggesting passwords that didn't adhere to websites' password policies. Usability issues like these have the potential to undermine password managers' security benefits: participants reported reusing weak passwords when password managers did not work as expected. Lyastani et al. studied factors associated with secure credential management practices [24]. They found that people who used password generation tools tended to use stronger passwords than people who didn't use tools to create their passwords. They also found that passwords entered by the LastPass password manager were more likely to be unique than passwords entered manually or using Chrome's built-in autofill. Note that at the time of Lyastani et al.'s study, Chrome did not offer to generate random passwords by default, but Chrome does now. A subsequent study conducted by Zibaei et al. contrasted the password generation functionality of the password managers integrated into the Chrome, Safari, and Firefox web browsers [52]. By the time of their study, all three browsers featured nudges encouraging the use of randomly generated passwords when users sign up for accounts. The authors found that Safari's nudge was accepted more often than Chrome or Firefox's nudges. These findings suggest the importance of password managers encouraging the use of random, unique passwords – simply increasing convenience by remembering weak or reused passwords is unlikely to improve security. Furthermore, the details of exactly how users are nudged can have a significant impact on whether users will follow the browser's recommendations.

Other work confirms that password managers are affected by various usability issues [19], [25], [29], [33]. However, prior work does not address the probability of users encountering these issues on actual websites. Are usability issues relatively rare, or do they play a major part in a typical user's experience of using a password manager? To answer this question, we tested password managers directly on a representative sample of websites by creating and logging in to accounts on those websites. Prior work suggests that users rarely change password managers' default settings, so we tested all password managers in their out-of-the-box configurations [29]. We found that usability issues are common across password managers and websites, suggesting that usability issues do play a major part in a typical user's experience.

B. Password Composition Policies

Websites often impose password composition requirements on their users' passwords. Ideally, these requirements should protect users' accounts from compromise, without negatively impacting usability to an unacceptable degree. Research has explored the relationship between password composition policies, password strength, and usability [35], [34], [45], [40]. For example, Tan et al. studied how participants created passwords under different password composition policies [40]. To achieve high usability (e.g., memorability) and security (i.e., resistance to guessing), the authors recommend requiring that passwords contain at least 12 characters and be resistant to an estimated 10^{10} guesses. Furthermore, they recommend *against* character class requirements (e.g., uppercase, lowercase, digits, and punctuation), as character class requirements decreased usability and in some cases decreased security as well. NIST's Digital Identity Guidelines serve as an authority for password best-practices [16]. Similar to Tan et al. [40], NIST's latest recommendations warn against "requiring mixtures of different character types" [16]. This expert consensus against character class requirements stands in contrast to NIST's earlier recommendations [5] and the current state of practice online [23]. For instance, Lee et al. found that 45% of popular websites have character class requirements [23]. The authors also found that most websites allow using leaked and easily guessed passwords, which also conflicts with NIST's guidelines. Alroomi et al. developed an automated method to experimentally determine the password composition policies of more than 20K websites, and also found widespread deviation from NIST's guidelines [2].

Different from prior work, our research explores how password composition policies impact the usability of password managers. Character class requirements pose a particular challenge to password managers, since some websites require using punctuation that other websites prohibit. Of course, this complicates the task of randomly generating passwords. Researchers and industry have proposed languages that encode password policies in a machine-readable format, to help password managers suggest passwords that meet websites' requirements [3], [12], [6], [36], [18]. By quantifying the scope of password policy-related usability issues, our work may encourage websites to adopt these languages.

III. Method

To estimate the prevalence of password manager-related usability issues, we tested a diverse set of password managers on a representative sample of websites. We started by identifying 57 different password managers (§ III-A). Next, we determined

the default password generation behavior of the most popular of these password managers (§ III-B), and used this data to select four different password managers for in-depth testing (§ III-C). Then, we used data on real users' authentication behavior to select a representative sample of websites on which to test password managers (§ III-D). Finally, we evaluated the usability of four password managers by attempting to use them during registration and authentication on 100 different websites (§ III-E).

A. Identifying Popular Password Managers

Password managers exist both as separately installed apps, and as built-in features of modern web browsers. Google Chrome, Safari, Firefox, and Microsoft Edge are the most popular browsers [37], [49], and all include integrated password managers. To identify popular standalone password managers, we searched for the phrase "password manager" using Google search, the Google Play Store, the iOS App Store, and the macOS App Store. We also searched for "open source password manager" using Google search. We recorded the password managers referenced in the first ten natural search results, which gave us the 57 password managers shown in Table IV in Appendix B. Wherever possible, we retrieved data about the number of ratings and installations on the Google Play Store, the iOS App Store, and the macOS App Store, which we used to estimate each password manager's popularity. Based on their popularity, we selected ten standalone password managers for initial analysis: Microsoft Authenticator, LastPass, Keeper, Dashlane, Norton Password Manager, KeePassDroid, 1Password, Bitwarden, mSecure, and RoboForm.

Note that some password managers are not cross-platform, but offer related apps on different platforms. For example, Microsoft Authenticator is not available on Windows or macOS, but equivalent password generation functionality is available on those platforms using the Microsoft Autofill browser plugin. For completeness, we tested using Microsoft Autofill and three additional KeePass clients.

B. Exercising Default Password Generation

Next, we collected examples of passwords generated by the four popular web browsers and the ten popular standalone password managers. Note that some password managers offer the option of customizing how passwords are randomly generated (e.g., enabling or disabling use of punctuation). In all cases, we left password managers' password generation settings in their default out-of-the-box configurations, since this is consistent with how most users use password managers [29]. We gathered password data by repeatedly using each password manager to "create an account" on the simple web application shown in Figure 1. The web application included a web form consisting of a username field and two password fields. After submitting the form, the web application echoed the entered password. We collected ten examples of passwords generated by each password manager, on each of the major platforms the password managers were available (e.g., Windows, macOS, Android, and iOS). In total, we collected 520 examples of passwords generated by these password managers. Table I summarizes password managers' default password generation behavior.

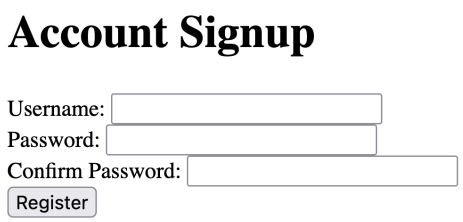

Fig. 1. To measure password managers' default password generation behavior, we repeatedly "created an account" using a simple web application.

Note that this data shows the *default* password generation behavior of password managers, but some password managers automatically tailor password generation to websites' password composition policies (e.g., Chrome, Safari, and 1Password) [4], [13], [3]. Thus, it was important to also test password managers on real-world websites to develop a more complete picture of how they generate passwords. Our data are consistent with the default password generation method described in Chrome's source code, which Chrome uses when crowdsourced password policies aren't available [44], [41].

C. Selecting Diverse Password Managers

As shown in Table I, password managers randomly generate passwords in different ways. For example, some password managers generate passwords which include punctuation, while others do not include punctuation. It would have been impractical to test many different password managers on many different websites, so we selected a small but diverse set of popular password managers for in-depth testing.

First, we selected the password managers integrated into Google Chrome and Apple's Safari, since these are the most used web browsers by a wide margin [37], [49]. By default, Chrome suggests 15 character passwords composed of only ASCII uppercase, lowercase, and digit characters. By default, Safari suggests 20 character passwords composed of only ASCII uppercase, lowercase, and digit characters, separated by ASCII hyphens at fixed positions. Also, both Chrome and Safari tailor password generation behavior on certain websites [13], [3]. We thought it was important to test these browsers' password generation functionality because they are so widely deployed, their password generation approaches are different, and they are maintained by companies with significant resources.

Next, we selected two separately installed password managers. For interpretability of our findings, we selected from among the password managers which generate passwords in a consistent format across platforms. First, we choose Keeper, the most popular standalone password manager that met this criteria. On all platforms, Keeper suggests 20 character passwords composed of ASCII uppercase, lowercase, digit, and diverse ASCII punctuation characters. Next, we selected Bitwarden, which suggests 14 character passwords that omit punctuation characters. We anticipated that some types of punctuation used by Keeper might conflict with websites' passwords policies, so Bitwarden served as a point of comparison. In addition, neither Keeper nor Bitwarden automatically tailor password generation to websites' requirements. Since Bitwarden and Chrome's default password generation approaches

Password Manager	Platform	Length	Character Classes	Example
1Password	Android	20	Upper, Lower, Digits	LW3V622zZuGiAaLpyxNW
	Windows, macOS	19	Upper, Lower, Digits, Punct.	ynf!nry8tyj6uba7BEP
	iOS	24	Upper, Lower, Digits, Punct.	yTK*pQhX63NU4yN.YjXDWH48
Bitwarden	Android, Windows, iOS, macOS	14	Upper, Lower, Digits	4dvZxgJrxXb6JL
Chrome	Android, Windows, iOS, macOS	15	Upper, Lower, Digits	8idVVRv7SSk72K3
Dashlane	Android, Windows, iOS	16	Upper, Lower, Digits, Punct.	@1Myl50dk2DLgsB$
	macOS	16	Upper, Lower, Digits	LcSbpNcXnfqstf5Q
Firefox	Windows, macOS	15	Upper, Lower, Digits	hRGuQgguLNz7H3W
KeePass	Android (KeePassDroid)	8	Upper, Lower, Digits	QWhvXIzE
	Windows	20	Upper, Lower, Digits	LlmTcJfORysygwfiSxRj
	iOS (Strongbox)	16	Upper, Lower, Digits, Punct.	+GK7Ck7Gyy*zab^u
	macOS (KeePassXC)	20	Upper, Lower, Digits, Punct.	RT?-ZYWAhe>M"pUI_|~{
Keeper	Android, Windows, iOS, macOS	20	Upper, Lower, Digits, Punct.	3.3kqvco)c?T+b3^UXqc
LastPass	Android, iOS	16	Upper, Lower, Digits, Punct.	6l#hNoa7@rc1gPAr
	Windows, macOS	12	Upper, Lower, Digits	6Va95W2QQovc
Microsoft Authenticator	Android, iOS	15	Upper, Lower, Digits, Punct.	wDb4wPCZXIgBR!8
Microsoft Autofill	Windows, macOS	15	Upper, Lower, Digits, Punct.	!Y3rEEPiyP3UAs7
Microsoft Edge	Android, Windows, iOS, macOS	15	Upper, Lower, Digits, Punct.	cRYimz:R9GLzE7z
Norton	Android, Windows, iOS, macOS	20	Upper, Lower, Digits, Punct.	lm-eWg0yj0fvtFer8viv
RoboForm	Android, Windows, iOS, macOS	16	Upper, Lower, Digits, Punct.	LzEMS%gRYXq4h#9P
Safari	iOS, macOS	20	Upper, Lower, Digits, Punct.	bokxaq-tifnEm-jygre1
mSecure	Android, Windows, iOS, macOS	18	Upper, Lower, Digits, Punct.	mE$sYA7w}j7XU2<p1#

TABLE I. WE DETERMINED THE DEFAULT PASSWORD GENERATION BEHAVIOR OF POPULAR PASSWORD MANAGERS BY REPEATEDLY "CREATING AN ACCOUNT" USING THE SIMPLE WEB APPLICATION SHOWN IN FIGURE 1. WE SELECTED FOUR POPULAR WEB BROWSERS AND TEN POPULAR STANDALONE PASSWORD MANAGERS FOR THESE TESTS, AND FOR COMPLETENESS WE ALSO TESTED RELATED PASSWORD MANAGERS ON DIFFERENT PLATFORMS. WE FOUND THAT PASSWORD LENGTH AND CHARACTER CLASSES VARIED BY PASSWORD MANAGER.

are similar, this allowed us to measure the effectiveness of Chrome's attempts to automatically tailor password generation to particular websites.

Ideally, we would have conducted in-depth testing using more than four password managers, but this wasn't feasible due to the time-intensive nature of our testing protocol. We address this limitation more fully in Limitations (§ V).

D. Selecting Representative Websites

To maximize security, users should use a unique password when they first create an account on a website. Ideally, password managers should assist by suggesting and storing randomly generated passwords at the time of account creation. However, if a randomly generated password is not compatible with a website's password policy, users may resort to the insecure practice of reusing a password they have memorized [24].

Since we are interested in studying the prevalence of problems that occur at the time of account creation, the binary question of whether a user has an account on a website is more relevant than the question of how much time a user spends on a website. Rankings of website popularity are typically based on the amount of internet traffic a domain receives [22], and the relationship between website popularity and the likelihood of having an account is unclear. Furthermore, popular websites may have more resources to test and improve compatibility with password managers, and password manager behavior may be customized to work well on popular websites [4], [13], [3]. Thus, focusing only on the most popular websites could give a misleading estimate of how likely users are to encounter issues when using a password manager during the account creation process. Instead of using website popularity rankings to select websites for our analysis, we used data on website login events collected by Carnegie Mellon University's Security Behavior Observatory (SBO) [51], [11], [31]. The SBO collected data from instrumented browsers on participants' PCs. The data was anonymized before it was shared with our research group, in accordance with the SBO's IRB at Carnegie Mellon University. Our university's IRB declared our research Not Human Subjects Research. For our study, we used data collected by the SBO from December 2016 to January 2019. Data was collected from 207 participants, and login attempts to 4,343 different domains were recorded. On average, the participants logged into 51.7 different domains.

As shown in Figure 2, SBO participants had accounts on many less popular websites. In fact, Tranco's 1000 most popular websites only accounted for 29% of participants' accounts. Since users have accounts on many less popular websites, it was essential that we tested password managers on a sample of websites representative of the websites where users have accounts. We selected websites to test password managers on by randomly sampling from the domains present in the SBO data, weighting each domain by the number of participants observed logging into that domain. Based on our time constraints, we tested on a sample of 100 different websites, which are listed in Table V in Appendix B.

E. Testing Diverse Password Managers on Representative Websites

To understand the prevalence of password manager-related usability issues, we tested using Chrome, Safari, Bitwarden, and Keeper on the 100 websites we sampled from the SBO data. We tested each combination of website and password manager by creating a new account, then logging out of and back into that account. Appendix A describes the details of our protocol, and Figure 3 summarizes the protocol.

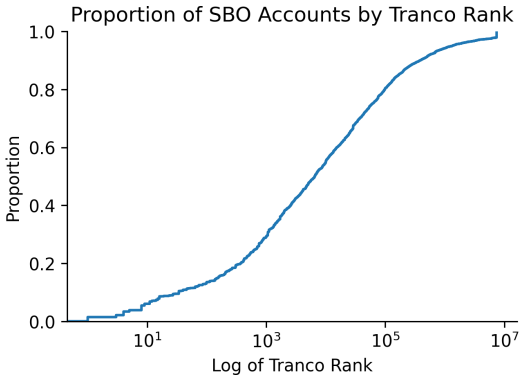

Fig. 2. This ECDF plot shows the distribution of SBO participants' accounts by Tranco popularity rank.[1]Looking at a particular x-axis value shows the proportion of observed accounts with that Tranco rank or lower (i.e., more popular). Consider statista.com, with a Tranco rank of 1000: the graph shows that 29% of participants' accounts were on websites of equal or greater popularity. We consider domains without Tranco rankings to be less popular, so we assign them the maximum Tranco rank plus one, resulting in the jump at the top of the graph.

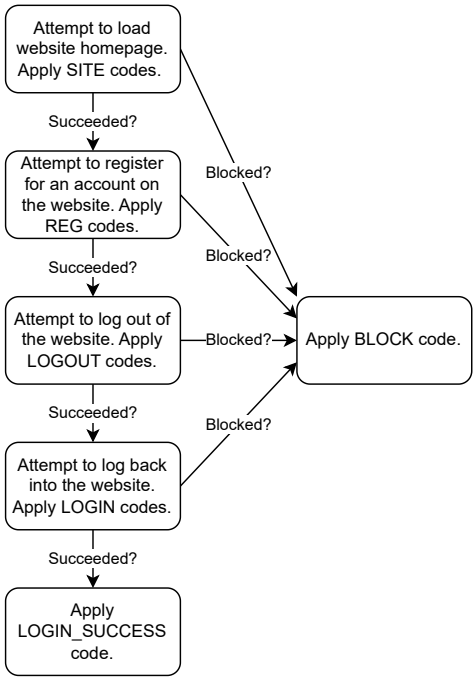

Fig. 3. In each step of testing, researchers applied codes describing the usability issues they encountered.

We systematically coded password manager related usability issues as we performed our testing. We started with a small set of codes for each step of testing, along with catch-all codes for new types of issues. Two researchers recorded themselves testing each combination of website and password manager, to identify cases when a password manager might sometimes encounter issues, and sometimes not. For example, since password managers generate passwords randomly, sometimes passwords may be compatible with a website's password composition policy, and sometimes not. After testing a batch of websites, we met to refine and expand our codebook. Also, in cases where our codes didn't match, we reviewed recordings to determine whether the codes were genuinely different (e.g., due to random behavior of the website or password manager), whether the codes were simply misapplied, or if our testing protocol wasn't followed correctly. We resampled in cases where our testing protocol wasn't followed correctly (e.g., if a prompt from the password manager was mistakenly ignored).

Table VII in Appendix B lists code descriptions and occurrences. We interpret these codes in our results section (§ IV). Of the 100 websites we tested on, we completed data collection on 61 websites (i.e., we successfully created an account, then logged back into that account). In most cases, the reason we couldn't complete data collection was because account creation required information we couldn't feasibly provide. For example, creating an account on comcast.net requires a Social Security Number (SSN) or phone number associated with an Xfinity subscriber account. Other reasons included websites not offering the option to sign up, websites not loading, and websites not being available in English. We give a complete list of reasons in Table V in Appendix B. We collected data from June to December 2022. We allowed password managers to update themselves, to avoid incompatibly between the password manager clients and their backend services. We started testing using versions 2022.6.1, 103, 16.4.4.1, 15.5 of Bitwarden, Chrome, Keeper, and Safari, respectively. We finished testing using versions 2022.10.1, 108, 16.4.8, 15.6.1 of Bitwarden, Chrome, Keeper, and Safari, respectively. We reviewed the password managers' changelogs, and found no description of changes to password generation or autofill between these versions.

IV. Results

We used the Google Chrome, Safari, Bitwarden, and Keeper password managers to test account creation and authentication on a representative sample of 100 websites. Of the 100 websites we tested, we successfully created our own credentials on 60 websites, and one website provided us with default credentials. As shown in Table II, usability issues associated with using password managers out-of-the-box to register (§ IV-A) and authenticate (§ IV-B) on websites were common (RQ1). Such usability issues are problematic because users are more likely to reuse passwords when password managers don't function properly [32]. It was especially common for websites to reject passwords suggested by password managers. Although, there is no simple way for password managers to satisfy every websites' password composition policy, some approaches work better than others (RQ2, § IV-C). Finally, we show that websites can accommodate passwords from all popular password managers by making their password policies less restrictive (RQ3, § IV-D).

A. Issues Related to Account Registration

We encountered account registration-related issues on between 21% and 46% of the websites we tested, depending on password manager (Table II). Note that each password manager assists with account registration in slightly different ways, and some types of issues only affect particular password managers. In particular, when registering for an account:

[1]Throughout the paper, we use the Tranco list which covers December 25, 2018 to January 23, 2019, to align with the end of SBO data collection. For more details about this Tranco list, visit: https://tranco-list.eu/list/25Y9/full

	Websites on which we encountered:		
Password Manager	**Issues Registering**	**Issues Authenticating**	**Either Type of Issue**
Bitwarden	21% (13)	30% (18)	39% (24)
Chrome	20% (12)	15% (9)	28% (17)
Keeper	46% (28)	38% (23)	61% (37)
Safari	21% (13)	11% (7)	30% (18)

TABLE II. THE NUMBER OF WEBSITES ON WHICH EITHER RESEARCHER ENCOUNTERED USABILITY ISSUES, OUT OF THE 61 WEBSITES ON WHICH WE CREATED ACCOUNTS. NOTE THAT SOME TYPES OF USABILITY ISSUES ONLY AFFECTED PARTICULAR PASSWORD MANAGERS, AND SOME ISSUES ARE MORE SEVERE THAN OTHERS. TABLE VII IN APPENDIX B GIVES THE COUNT OF EACH USABILITY ISSUE BY PASSWORD MANAGER.

- Bitwarden requires the user to take the initiative to generate a password by either clicking on the plugin's icon or otherwise activating Bitwarden. Then, Bitwarden users can create, save, and fill credentials for the website.
- On an account registration page, Keeper sometimes displays a banner with a "Create New Record" button; when this button is clicked, Keeper gives the option to create, save, then fill the credentials. If this banner isn't displayed, Keeper still offers to fill user IDs and generate passwords inline on the page as those fields are clicked.
- Similarly, Chrome and Safari offer to generate a password when the user clicks the password field.

Next, we discuss three of the major account registration-related usability issues we observed: password managers not prompting users to generate credentials, password managers not filling registration forms appropriately, and websites rejecting the passwords suggested by password managers.

Password Managers May Not Prompt Users To Generate Credentials: Our testing showed that password managers sometimes don't show their usual prompts to encourage users to generate secure credentials.

Sometimes, the prompt is not displayed unless the user performs an additional action, such as right-clicking the password field, activating the password manager using its icon, clicking a password field that is already focused, or clicking away from and returning to the webpage. We saw this on 12, 7, and 3 websites with Chrome, Keeper, and Safari, respectively.

We observed that Keeper's user ID and password prompts sometimes appeared, then disappeared before they could be used. In these cases, using Keeper required either clicking in the fields repeatedly to reactivate the prompts, or accessing Keeper's user interface in another way. We saw this issue on nine different websites.

In some cases, Safari's password generation prompt is displayed in a more subtle fashion than usual. Instead of displaying a default nudge (Figure 4), Safari sometimes displays a smaller drop-down menu with the option to suggest a password (Figure 5). We observed this issue on five different websites. Although the difference between these prompts may seem small, prior work suggests that the style of the prompt can have a significant impact on users' behavior. Zibaei et al. compared

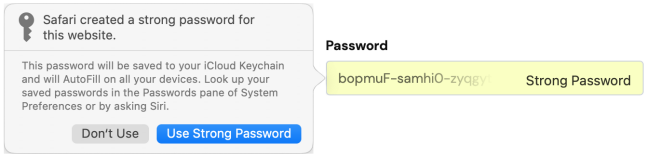

Fig. 4. Usually, Safari displays a default nudge to encourage users to adopt randomly generated passwords.

Fig. 5. In some cases, Safari displays a subtle drop-down menu with the option to suggest a password.

Safari's default nudge to Chrome and Firefox's more subtle prompts [52]. They found that Safari was more effective at nudging users to use its password generation features than were Chrome and Firefox. A follow-up study found that Safari's prompt is so effective because it fills the password field with a random password by default [53]. However, since Safari sometimes doesn't do this, our research suggests that Safari may be less effective at nudging users in practice.

Finally, Safari sometimes wouldn't suggest passwords on any websites until the browser was restarted. We couldn't identify the cause of this issue, but we did observe it on multiple devices. To avoid conflating this behavior with website-specific issues, at the start of each data collection session we confirmed that Safari could suggest passwords on our simple web application before proceeding. Since this issue was not website-specific, we did not include it in our codes. Nevertheless, this bug could deter users from using Safari's password manager, which is a serious concern.

Password Managers May Not Fill Registration Forms Appropriately: We found that Bitwarden and Keeper sometimes had difficulty filling user IDs and passwords during the registration process.

On three websites, Keeper couldn't fill the password field, even when the "Fill" button was clicked. Instead, we had to manually copy-paste the password from the Keeper plugin.

Account registration forms often include more than just user ID and password fields. For example, registration forms can include fields for the user's name, birthday, or address. When Bitwarden and Keeper offer to fill the credentials they have stored, they sometimes fill information into the wrong fields. We encountered this issue on six websites when using Bitwarden and Keeper. Usually, the password managers filled the user ID into an inappropriate text field, such as when Keeper filled the email address into the first name field on yahoo.com. However, the issue can also affect password data: on helpowl.com, Bitwarden filled the password into the security question answer field. This situation is more concerning, because if the user submitted the form, it could result in the password being stored in plaintext in the website's database. Huaman et al. also found that websites containing more input fields than necessary cause password managers to autofill

	Websites on which passwords were:		
Password Manager	**Accepted Consistently**	**Rejected Once**	**Rejected Consistently**
Bitwarden	87% (52)	0% (0)	13% (8)
Chrome	98% (58)	0% (0)	2% (1)
Keeper	78% (47)	5% (3)	17% (10)
Safari	88% (53)	2% (1)	10% (6)

TABLE III. TWO RESEARCHERS TESTED USING EACH PASSWORD MANAGER TO GENERATE CREDENTIALS ON EACH WEBSITE WHERE THIS WAS POSSIBLE. SOME WEBSITES ACCEPTED BOTH PASSWORDS, SOME WEBSITES REJECTED JUST ONE PASSWORD, AND SOME WEBSITES REJECTED BOTH PASSWORDS. SINCE GOOGLE CHROME DOESN'T SUPPORT CREATING CREDENTIALS ON GOOGLE.COM, CHROME GENERATED PASSWORDS ON 59 DIFFERENT WEBSITES, WHEREAS THE OTHER PASSWORD MANAGERS GENERATED PASSWORDS ON 60 WEBSITES.

inappropriate fields [19]; our testing supports their findings and demonstrates the security risks of complex registration forms.

Many Websites Reject Password Managers' Suggested Passwords: Of the 60 websites on which we generated credentials using password managers, 19 websites (32%) rejected passwords from at least one password manager. Table VI in Appendix B lists these websites, the password managers which generated incompatible passwords, and the password policies shown by these websites. Table III summarizes the number of websites which rejected passwords from each password manager. Passwords generated by Keeper were rejected most often (on 13 websites), followed by Bitwarden (on 8 websites), Safari (on 7 websites), and finally Chrome, which was only rejected on a single website. Overall, the password managers which tailored password generation to websites' composition requirements (Chrome and Safari) performed better than the password managers which used a static approach to password generation (Bitwarden and Keeper). We interpret these findings more fully in § IV-C.

B. Issues Related to Authentication

After creating an account on a website, users are either automatically logged into their new account, or they must authenticate to log in. Regardless, users are likely to authenticate at some point, so we tested how well password managers support the authentication process. When logging in, password managers should offer to fill the password field and the user ID field (e.g., the username or email address), but our testing shows that password managers can't always do this reliably; we encountered account authentication-related usability issues on between 11% and 38% of the websites we tested, depending on password manager (Table II). If users cannot rely on password managers to assist them with logging in, they may be discouraged from using password managers in the future.

Password Managers May Not Fill User IDs Correctly: During the account creation process, Chrome and Safari try to automatically record the user ID. In some cases, these browsers record wrong or inconsistent information as the user ID, or don't record the user ID at all. Bitwarden and Keeper use a different approach, giving the user the opportunity to define the user ID when registering credentials, and sometimes offering to store or update the user ID automatically. When testing Bitwarden and Keeper, we always tried to enter the user ID correctly, but we still encountered issues in certain situations. When logging in, we saw password managers suggest incorrect user IDs (on 8, 5, 13, and 4 websites with Bitwarden, Chrome, Keeper, and Safari, respectively), suggest both correct and incorrect user IDs (on 8, 1, and 2 websites with Bitwarden, Chrome, and Keeper, respectively), and not suggest a user ID at all (on 2 and 3 websites with Chrome and Safari, respectively). The reasons for these issues weren't always clear, but we noticed some patterns.

Some websites supply the user ID for accounts, and this makes it difficult for password managers to associate the user ID with credentials. For example, hilton.com uses the Hilton Honors number as the user ID when logging in. However, this number isn't displayed until after the account is created. As a result, Chrome and Safari incorrectly capture the email address as the user ID. The same issue applies to Bitwarden and Keeper: since the Hilton Honors number is not available when registering credentials in these password managers, anything the user enters as their user ID will be incorrect. All password managers had similar issues on aa.com, which requires using an AAdvantage number to log in. aa.com's login form also includes a last name field in addition to the AAdvantage and password fields. After filling the AAdvantage and password fields, Keeper automatically submitted the form before the researcher could enter their last name. Since aa.com only allows "6 tries before your account is locked," automatically submitting incorrect credentials is especially problematic.

Several websites had composition requirements for user IDs, which sometimes caused problems. For example, yahoo.com rejected a username with our usual format (e.g., "peter.story.bitwarden@yahoo.com"), explaining that "You can't have more than one '.' in your username." Since we had already stored the user ID with multiple periods in Bitwarden, and Bitwarden didn't automatically detect the edits we made directly on the website, Bitwarden later suggested the incorrect user ID when it was time to log in. The same issue applied to Keeper, for the same reasons. Issues related to user ID composition requirements manifested as either the wrong user ID being suggested when trying to log in, or multiple user IDs being offered when trying to log in (i.e., in cases where the password manager captured the revised user ID, but also kept the original user ID). We also saw these kinds of issues on steampowered.com, delta.com, irctc.co.in, neopets.com, playerauctions.com, helpowl.com, and engageme.tv.

Registering on some websites requires supplying both a username and email address, and this seemed to result in several usability issues. For example, when registering for neopets.com, the registration form accepts both a username and email address, but only the username can be used when logging in. Chrome stored the email address as the user ID, which wasn't accepted when logging in. Safari also had issues on neopets.com, because it didn't store a user ID at all. Similarly, pittplusme.org collects both username and email address, but Bitwarden automatically associated the password with the email address, which could not be used to log in.

Password Managers May Not Fill Passwords Correctly: As part of our testing, we always tried to use password managers to generate passwords. Nevertheless, password managers sometimes filled incorrect passwords when logging in (on 5, 1, 14, and 1 websites with Bitwarden, Chrome, Keeper,

and Safari, respectively). Most often, this happened when a website's password composition policy rejected the automatically generated password, but the password manager did not automatically store edits made to the password. For example, samsung.com rejected Bitwarden's password because it did not include a symbol, so we appended a trailing exclamation mark to meet the requirements. Bitwarden did not capture this edit, so it filled the outdated password when we tried to log in. Although some of these issues could be avoided by editing the password within the password manager, we think our method of revising the password directly on the website will be representative of many users' behavior, since it requires fewer steps.

Some of these login issues were also caused by a security vulnerability in Keeper. We observed that when attempting to log in, Keeper sometimes supplied the credentials of a completely unrelated website. We discovered this issue by examining cases where logins initially failed when filling credentials using Keeper's autofill banner, but succeeded after either refreshing the webpage or filling the credentials using Keeper's menu. On four websites, it was directly visible that the incorrect user ID had been filled. For example, on costco.com the username "peter.story.keeper" was filled instead of the email address "peter.story.keeper2@gmail.com". Inspecting Keeper's credential vault showed that the user ID for costco.com's credentials was recorded as an email address, but the username that was filled was from a different website's credentials. Furthermore, Keeper had also filled the password from the other website. We confirmed that Keeper erroneously disclosed passwords on at least two other websites. In total, we saw evidence of this bug on seven websites. There seemed to be an element of randomness, and most examples of the bug were captured by only one researcher. Although we couldn't reliably reproduce the bug, we encountered it on four different days. After completing data collection, we reported this security vulnerability to Keeper Security, Inc. Keeper's security team could not replicate the issue using the latest version of Keeper.

We also observed two instances where password managers didn't record passwords. On google.com, Chrome neither suggests nor automatically records credentials, which may be an intentional design choice. On socialmedialink.com, Safari suggested a password, but didn't record the user ID or password. Only one researcher observed this issue, so it seems to occur randomly.

Keeper Requires Payment After Its Trial Ends: Although Keeper offers a 30-day trial, after the trial expired we were unable to access our credentials from the plugin without paying. Opening the plugin after the trial expired gave the message: "Your free trial has expired. Please purchase a subscription." It is possible to access one's credentials by logging in to Keeper's mobile app. However, users of the browser plugin who don't use the mobile app would likely be unaware of this option. This means that users of the Keeper browser plugin who are unable or unwilling to pay could be locked out of their accounts if they only stored their credentials in Keeper.

C. Static Approaches To Generating Passwords Work Poorly

Bitwarden and Keeper use their default password generation approach on every website, whereas Chrome and Safari adapt their password generation approach depending on the website. As shown in Table III, Chrome's passwords were accepted far more often than were passwords from the other password managers. Keeper's passwords were rejected the most often. Chrome's passwords followed its default pattern only 22% of the time. When Chrome's passwords didn't follow the default pattern, they always included some kind of punctuation, and they occasionally had 14 instead of 15 characters. Chrome's passwords were only rejected on pittplusme.org, and these passwords did follow Chrome's default password generation behavior. pittplusme.org has a relatively low Tranco rank (Table V in Appendix B), so perhaps Google had insufficient data to tailor password generation on this website. Safari's passwords generated on real world websites followed Safari's default pattern 83% of the time.[2] Taking a closer look at these passwords, we see that Safari can generate passwords with fewer characters, with varying types of punctuation, and without any punctuation at all. Nevertheless, in three cases Safari's adaptive passwords were still rejected. Two such passwords were rejected by gofobo.com and one password was rejected from irctc.co.in, due to these websites' constraints on punctuation. Notably, irctc.co.in appears in Apple's list of hardcoded password generation rules, but the rule incorrectly includes punctuation that the website disallows [3].

Which password generation approaches can password managers adopt to maximize their compatibility with websites? (RQ2): Our findings show that there is not a simple approach to generating passwords that will satisfy all websites' password composition policies. All the passwords we generated contained uppercase, lowercase, and digits; password managers' behavior only differed in terms of length and use of punctuation. A static approach to generating passwords cannot satisfy all websites' password composition policies, since some websites require using punctuation that other websites prohibit. For example, gofobo.com requires using special characters that match.com disallows (Table VI in Appendix B). Keeper's approach of always including punctuation seemed to cause the most issues (rejections on 22% of websites). Bitwarden's approach of always omitting punctuation worked almost as well as Safari's dynamic approach to generating passwords (rejections on 13% and 12% of websites, respectively). However, Chrome's dynamic approach worked better than Safari's, as Chrome's passwords were only rejected on a single website (less than 2% of websites). Our results suggest that password managers can significantly improve the compatibility of their suggested passwords if they can determine websites' password composition policies. At the very least, password managers should not include diverse punctuation characters by default.

D. Overly Prescriptive Password Policies Will Reject Randomly Generated Passwords

Which password policies can websites adopt to maximize their compatibility with password managers? (RQ3): To understand how different password managers generate

[2]We consider the pattern we observed when testing with our simple web application to be the "default" pattern. Table I shows the default password generation behavior of each password manager.

passwords, we used four popular web browsers and ten popular standalone password managers to repeatedly "create an account" on a simple web application (Figure 1). Table I summarizes our findings, and includes examples of passwords generated by each password manager.

We found that password length and character class defaults differed between password managers. In some cases, a password manager's defaults also varied by platform. For example, passwords generated by 1Password differed in length depending on whether Android, Windows, iOS, or macOS was used. Also, 1Password's passwords included punctuation by default on Windows, iOS, and macOS, but not on Android. By default, the shortest passwords were generated by KeePassDroid on Android (eight characters) and the longest passwords were generated by 1Password on iOS (24 characters). All password managers generated passwords composed of uppercase, lowercase, and digits, and some password managers also included punctuation. However, in some cases password managers generated passwords that omitted a particular character class at random (i.e., if a password manager generates passwords composed of uppercase, lowercase, and digits, that doesn't necessarily imply that *every* password it generates will include digits). Considering all the password managers we tested, we generated passwords containing every non-whitespace printable ASCII character.

Our data show that websites with password composition policies which enforce length and character class requirements are likely to disallow passwords generated by popular password managers. For example, requiring punctuation, limiting length below 24 characters, or prohibiting certain characters would disallow different passwords in our dataset. Nevertheless, it should be technically feasible for websites to store all the passwords we generated, since all of the passwords consisted of only non-whitespace printable ASCII characters.

To emphasize the security value of accepting passwords generated by password managers, we attempted to "crack" all the passwords we generated using Carnegie Mellon University's publicly available Password Guessability Service (PGS) [46], [7]. Using the service's default settings, none of the passwords were guessed by Hashcat, John the Ripper, the markov model, or the probabilistic context-free grammar. PGS's neural network estimated that even the weakest password we generated, an eight-character password suggested by KeePassDroid, would resist more than 68 trillion guesses. Thus, password managers help users avoid password reuse and create passwords that are resistant to guessing. These findings support NIST's claim that "password managers . . . in many cases increase the likelihood that users will choose stronger memorized secrets" [16].

V. Limitations

Several limitations should be considered when evaluating our findings.

First, we only tested four password managers on real-world websites (i.e., Bitwarden, Chrome, Keeper, and Safari), and we only considered 100 real-world websites. Ideally, we would have tested more password managers on more websites, but this was not feasible due to the time-consuming nature of our data collection process. Testing a website involved locating the account registration page, filling multiple fields with information, possibly waiting for a registration confirmation email, then logging back into the website. Furthermore, two researchers needed to complete this process for each combination of password manager and website, to account for randomness (e.g., in the passwords suggested by password managers) and as a quality check to ensure we followed our protocol and applied codes consistently. Using crowdworkers was not an option because our researchers needed training to understand the data collection process, and it took a significant amount of time to configure our testing environments. For example, we needed to install each password manager in an isolated environment, to configure software for screen capture, and to set up multiple email addresses and VoIP phone numbers. Due to the limited amount of data we collected, we don't make claims about the statistical significance of our findings. However, we selected representative password managers and websites (§ III-A, III-C, and III-D), so our findings are still meaningful. Huaman et al.'s lab tests showed that all of the 15 password managers they tested were affected by usability issues [19]. Together with our work showing the ubiquity of issues on real-world websites, this suggests that usability issues will be a major part of a typical user's experience, regardless of password manager; of course, manual testing of additional password managers would be necessary to confirm this.

Second, the protocol we used when testing password managers on real-world websites captures a limited picture of what actual users might experience. For example, we did not modify the default settings of the password managers we tested, which seems to be consistent with typical user behavior [29]. This was done to understand the behavior of freshly-installed password managers (RQ1). A user who changed the default behavior might encounter more or fewer issues. We designed our protocol to approximate a best-effort attempt to use each password manager when registering for websites, with the goal of capturing a "typical" user's experience.

Third, we limited our research focus to what we could observe while using password managers during account creation and authentication. Consequently, we choose not to address some potentially related research questions. For example, although we recorded websites' stated password policies (Table VI in Appendix B), we did not attempt to validate the precise details of each website's password policy. Websites' stated password policies can be vague and may not be enforced as described, so prior work used extensive experimentation to determine the details of websites' password policies (e.g., exactly which characters are required or prohibited) [23], [2]. Our focus was on the prevalence of usability issues, so it was sufficient to observe whether passwords were accepted or not. Thus, we decided against trying to determine the precise details of websites' password policies.

Finally, our study is only a snapshot of the current state of password manager and website compatibility. Password managers and websites will change over time, hopefully in the direction of improved usability. For example, we tested using Safari 15 (§ III-E), but Safari 16 introduced a new user interface for suggesting passwords that makes it easier to customize password suggestions (Figure 7 in Appendix B). Furthermore, the tailored password generation rules used by

Safari and Chrome are (presumably) always being updated. Replicating our study after some time has passed can show whether usability has improved.

VI. Discussion and Future Work

We tested four popular password managers on a representative sample of websites, measuring the prevalence of issues associated with account registration and authentication. Using each password manager in its default configuration, we encountered some kind of usability issue on more than a quarter of websites (Table II). This suggests that users are very likely to encounter various usability issues when they use password managers in the real-world. Thus, password managers and websites should improve their compatibility with each other to improve the real-world usability of password managers.

RQ1: We found widespread usability issues during account registration (§ IV-A) and authentication (§ IV-B). These issues interfered with using a password manager to create secure credentials, and interfered with logging back into accounts. Multiple issues were related to password managers not identifying the correct input field as the user ID or as the new password field, which prior work suggests can be caused by non-standard form implementations [19]. Other issues were downstream effects of password and user ID composition requirements, which resulted in password managers storing incorrect credentials. And finally, some issues were due to bugs in the password managers themselves, such as the credential disclosure vulnerability we discovered in Keeper, and the bug which caused Safari to suggest but not record a password.

To mitigate these usability issues, we have several recommendations for website developers. **First, we recommend that websites revise their password policies to adhere to NIST's latest digital identity guidelines,** as we explain in more detail when we address RQ3. **Second, we recommend that websites allow users to log in using their email address instead of a custom user ID.** Custom user IDs can be forgotten by users and stored incorrectly by password managers. **Third, we suggest that website developers test password managers on their websites to evaluate the impact of their design decisions.** Specifically, website developers should use password managers during account creation and registration, and should test that password managers can suggest and fill secure passwords, as well as store and fill user IDs correctly. For example, this kind of usability testing could detect if the design of the website's account registration form makes it difficult for password managers to identify the user ID or password fields. For smaller organizations, it may be sufficient to test using the password managers integrated into the most popular web browsers (i.e., Chrome and Safari) and using a free standalone password manager (e.g., Bitwarden). Organizations with more resources should consider testing using additional popular password managers (§ III-A). Chrome's documentation offers guidelines for web developers to improve compatibility with password managers, such as grouping related fields in a single form element, and annotating fields using "autocomplete" attributes [8], [42].

We recommend that developers of password managers perform the same kind of usability testing on a representative sample of websites. Improvements to password managers' ability to identify the user ID and password fields may mitigate some of the issues we documented. Chrome uses a combination of heuristics and crowdsourcing to recognize form fields [43], yet we still encountered various usability issues when using Chrome, so improvements may be possible. For now, developers can test using the same set of websites we used (Table V in Appendix B), but in the long-term this list will become outdated. Future work could involve crowdsourcing data on the websites people log in to, in a privacy-preserving fashion. This data would be an asset for developers of password managers, and could be used to rerun this study in a few years.

Finally, developers of password managers must hold themselves to high standards of security and ethics. Passwords are incredibly sensitive information, and users won't adopt password managers if they don't trust them [1], [20]. We found it unacceptable that Keeper prevents users from accessing their credentials after its free trial ends. Keeper's security flaw that shared credentials with the wrong websites was also highly concerning (§ IV-B). The recent hack of the LastPass password manager, and subsequent criticism of LastPass's security practices, raises concern about the security of password manager offerings across the industry [9], [21]. We identified more than 50 password managers (Table IV in Appendix B), and given the security weaknesses of even the most popular [9], [28], it seems implausible that all these password managers are protecting users' credentials adequately. Ultimately, some form of regulation may be necessary to assure users of password managers' security. Until then, **security experts should avoid giving the general recommendation to simply "start using a password manager," and should instead recommend specific password managers. Password manager recommendations should be informed by academic research, the news, and experts' own experiences.** We encountered fewer usability issues when testing Chrome, Safari, and Bitwarden than when testing Keeper (Table II). Keeper was also the only password manager we tested that required a subscription. Thus, we recommend Chrome, Safari, or Bitwarden over Keeper. Furthermore, promoters of password managers should inform people of the common issues they might encounter, and help people form coping strategies to overcome those challenges [39]. For example, users should know that if Chrome doesn't initially suggest a password, they can simply right-click to kickstart the process.

RQ3: We found that conflicts between websites' password composition policies and the randomly generated default passwords suggested by password managers were especially widespread: 32% of the websites on which we generated credentials rejected a password from at least one password manager (§ IV-A). **We recommend that websites revise their password policies to adhere to NIST's latest digital identity guidelines [16].** Following NIST's guidelines will ensure compatibility with passwords suggested by Bitwarden, Chrome, Keeper, and Safari, as well as the ten other popular password managers included in our preliminary testing (§ IV-D). **In particular, websites should:**

1) Allow all non-whitespace printable ASCII characters. NIST recommends also allowing the space character.
2) Allow passwords of at least 24 characters. NIST recommends supporting at least 64 characters.

3) **Not** impose other composition rules, such as character class requirements. Research suggests that such requirements do not improve security [40].

Industry standards should be updated to match NIST's latest guidelines, and to explicitly promote the adoption of password managers [30]. By improving support for password managers, websites can encourage the use of unique credentials, and thereby protect themselves from credential stuffing attacks.

RQ2: Until websites update their password composition policies, password managers must do their best to adhere to websites' diverse requirements. We found that passwords generated by Keeper, Bitwarden, and Safari were rejected on 22%, 13%, and 12% of websites, respectively, whereas Chrome's passwords were only rejected on a single website. Chrome automatically adapts to many websites' password composition policies, which seems to explain why Chrome's passwords are usually accepted. Chrome's source code describes using crowdsourced data to determine websites' password policies [41]. Safari's manually curated list of password policies did not work as well as Chrome's approach [3]. **Other password managers should consider using crowdsourcing to adapt to websites' password policies.** However, smaller password managers may struggle to collect data about the long tail of website logins. Another approach is to use automated experimentation to determine password policies [2]. **Alternatively, Bitwarden's inclusion of just digits, uppercase letters, and lowercase letters was a simple approach that worked fairly well.** We would not recommend that password managers include randomly chosen punctuation by default, since Keeper used this approach and had the most issues. Our recommendation is consistent with Alroomi et al.'s finding that websites are more likely to disallow punctuation than they are to require punctuation [2].

Furthermore, giving the user control over a password manager's approach to generating passwords (e.g., a setting to enable or disable use of punctuation) would not solve the problem of passwords initially being rejected, since most of the websites which rejected our passwords did not display their password policies until after a password had been generated (Table VI in Appendix B). Thus, including an easy way to edit a generated password after it is rejected may help users choose unique passwords. For example, Safari 16 recently introduced a new user interface for suggesting passwords that makes it easier to customize password suggestions (Figure 7 in Appendix B).

Comparison With Prior Work: It is challenging to directly map some of the usability issues we encountered to prior work [19], [25], [29], [32], [33]. First, prior work used different methodologies (e.g., testing on smartphones and performing password changes). Second, prior work sometimes described usability issues in broad terms (e.g., Oesch described "inconsistent autofill and autosave functionality" [29]) whereas we documented issues at a fine-grained level. Nevertheless, most of the usability issues we documented are at least alluded to in prior work; we extend prior work by showing the high prevalence of usability issues across the web. Also, to the best of our knowledge we are the first to document several issues with Safari and Keeper: Safari sometimes displayed a subtle password prompt, Safari sometimes didn't suggest passwords until it was restarted, Keeper's password prompts sometimes disappeared, and Keeper sometimes filled credentials onto completely unrelated websites.

Future Work: Looking towards the future, efforts to replace passwords are ongoing. In particular, multiple platforms' implementation of the "passkey" standard is an exciting development [14], [15]. However, usable security research like ours shows how usability issues can prevent promising technologies from being adopted. We recommend that future work consider the usability of passkey-based authentication, especially since it is a significant departure from the authentication mechanisms users are familiar with (e.g., passwords and federated login). Furthermore, it seems unlikely that passkeys will completely replace passwords, especially on older websites. **As authentication mechanisms proliferate, future work should compare the usability of different authentication mechanisms, and measure how users choose between these different options.**

VII. Conclusions

Passwords are likely to remain the dominant form of authentication for the foreseeable future, making password managers an essential tool for helping users manage their credentials. To measure the usability of freshly-installed password managers, we tested four popular password managers on a representative sample of websites. We identified issues that interfered with using password managers during account registration, such as websites rejecting the randomly generated passwords suggested by password managers. We also encountered issues when trying to authenticate, such as password managers not filling passwords correctly. Both website operators and password manager developers should take steps to improve the compatibility of their offerings. For example, website operators should revise their password policies to adhere to NIST's latest recommendations, and password manager developers should perform thorough testing on real-world websites.

Based on the prevalence of the problems we found, users are likely to encounter usability issues even when using the most popular password managers. Despite our findings, people should still use password managers: without a password manager, it simply isn't possible to use unique passwords on a large number of websites. However, experts should recommend only the most secure and usable password managers, since users may have difficulty differentiating between the overwhelming number of products marketed as password managers.

Acknowledgment

Clark University's LEEP Fellowship Award supported Adryana Hutchinson and Jinwei Tang. This material is based upon work supported by the National Science Foundation under Grant No. 1845300. Thank you to Sarah Pearman, Lorrie Faith Cranor, and Nicolas Christin at Carnegie Mellon University for coordinating the sharing of anonymized SBO data. Also, thank you to Yaxing Yao, Yuanyuan Feng, Norman Sadeh, and Florian Schaub for feedback on the study's design.

References

[1] F. Alodhyani, G. Theodorakopoulos, and P. Reinecke, "Password Managers—It's All about Trust and Transparency," *Future Internet*, vol. 12, no. 11, p. 189, Oct. 2020, https://www.mdpi.com/1999-5903/12/11/189.

[2] S. Alroomi and F. Li, "Measuring Website Password Creation Policies At Scale," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, Nov. 2023, pp. 3108–3122, http://arxiv.org/abs/2309.03384.

[3] Apple, "Password Manager Resources," Apple, Dec. 2022, https://github.com/apple/password-manager-resources.

[4] M. Barker, "A smart(er) password generator," Mar. 2021, https://blog.1password.com/a-smarter-password-generator/.

[5] W. E. Burr, D. F. Dodson, E. M. Newton, R. A. Perlner, W. T. Polk, S. Gupta, and E. A. Nabbus, "Electronic Authentication Guideline," National Institute of Standards and Technology, Tech. Rep. NIST SP 800-63-1, Dec. 2011, https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-63-1.pdf.

[6] J. Campos, "SmartPasswords: Increasing Password Managers' Usability by Generating Compliant Passwords," Ph.D. dissertation, University of Lisbon, Nov. 2021, https://passcert-project.github.io/publication/2021/joao-campos-thesis/joao-campos-thesis.pdf.

[7] Carnegie Mellon University, "Password Guessability Service," 2015, https://pgs.ece.cmu.edu/.

[8] Chrome Developers, "Password Manager Compatible," Nov. 2022, https://developer.chrome.com/password-manager-compatible/.

[9] M. Clark, "The LastPass disclosure of leaked password vaults is being torn apart by security experts," Dec. 2022, https://www.theverge.com/2022/12/28/23529547/lastpass-vault-breach-disclosure-encryption-cybersecurity-rebuttal.

[10] D. Florencio and C. Herley, "A large-scale study of web password habits," in *Proceedings of the 16th International Conference on World Wide Web - WWW '07*. Banff, Alberta, Canada: ACM Press, 2007, p. 657, http://portal.acm.org/citation.cfm?doid=1242572.1242661.

[11] A. Forget, S. Komanduri, A. Acquisti, N. Christin, L. F. Cranor, and R. Telang, "Security Behavior Observatory: Infrastructure for Long-term Monitoring of Client Machines," Carnegie Mellon University, Tech. Rep. CMU-CyLab-14-009, Jul. 2014.

[12] A. Gautam, S. Lalani, and S. Ruoti, "Improving Password Generation Through the Design of a Password Composition Policy Description Language," in *Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022)*, Aug. 2022, pp. 541–560.

[13] M. Golla, "Using Google's PasswordRequirementsSpec API," Feb. 2021, https://github.com/apple/password-manager-resources/issues/427.

[14] D. Goodin, "How Apple, Google, and Microsoft will kill passwords and phishing in one stroke," May 2022, https://arstechnica.com/information-technology/2022/05/how-apple-google-and-microsoft-will-kill-passwords-and-phishing-in-1-stroke/.

[15] ——, "Passkeys—Microsoft, Apple, and Google's password killer—are finally here," Oct. 2022, https://arstechnica.com/information-technology/2022/10/passkeys-microsoft-apple-and-googles-password-killer-are-finally-here/.

[16] P. A. Grassi, J. L. Fenton, E. M. Newton, R. A. Perlner, A. R. Regenscheid, W. E. Burr, J. P. Richer, N. B. Lefkovitz, J. M. Danker, Y.-Y. Choong, K. K. Greene, and M. F. Theofanos, "Digital identity guidelines: Authentication and lifecycle management," National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep. NIST SP 800-63b, Jun. 2017, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63b.pdf.

[17] C. Herley, "So long, and no thanks for the externalities - the rational rejection of security advice by users." *NSPW*, pp. 133–144, 2009.

[18] M. Horsch, M. Schlipf, S. Haas, J. Braun, and J. Buchmann, "Password Policy Markup Language," *Proceedings of Open Identify Summit*, pp. 135–147, 2016.

[19] N. Huaman, S. Amft, M. Oltrogge, Y. Acar, and S. Fahl, "They Would do Better if They Worked Together: The Case of Interaction Problems Between Password Managers and Websites," in *2021 IEEE Symposium on Security and Privacy (SP)*. San Francisco, CA, USA: IEEE, May 2021, pp. 1367–1381, https://ieeexplore.ieee.org/document/9519389/.

[20] A. Karole, N. Saxena, and N. Christin, "A Comparative Usability Evaluation of Traditional Password Managers," in *Information Security and Cryptology - ICISC 2010*, K.-H. Rhee and D. Nyang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, vol. 6829, pp. 233–251, http://link.springer.com/10.1007/978-3-642-24209-0_16.

[21] B. Krebs, "Experts Fear Crooks are Cracking Keys Stolen in LastPass Breach," https://krebsonsecurity.com/2023/09/experts-fear-crooks-are-cracking-keys-stolen-in-lastpass-breach/, Sep. 2023.

[22] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczynski, and W. Joosen, "Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation," in *Proceedings 2019 Network and Distributed System Security Symposium*. San Diego, CA: Internet Society, 2019, https://www.ndss-symposium.org/wp-content/uploads/2019/02/ndss2019_01B-3_LePochat_paper.pdf.

[23] K. Lee, S. Sjöberg, and A. Narayanan, "Password policies of most top websites fail to follow best practices," in *Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022)*, Aug. 2022, pp. 561–580.

[24] S. G. Lyastani, M. Schilling, S. Fahl, M. Backes, and S. Bugiel, "Better managed than memorized? Studying the Impact of Managers on Password Strength and Reuse," *27th USENIX Security Symposium*, p. 19, 2018.

[25] P. Mayer, C. W. Munyendo, A. J. Aviv, and M. L. Mazurek, "Why Users (Don't) Use Password Managers at a Large Educational Institution," *31st USENIX Security Symposium*, pp. 1849–1866, Aug. 2022.

[26] N. Mueller, "Credential stuffing," Sep. 2021, https://owasp.org/www-community/attacks/Credential_stuffing.

[27] M. H. Nguyen, J. Bennett, M. Gallagher, and S. Bhunia, "A Case Study of Credential Stuffing Attack: Canva Data Breach," *2021 International Conference on Computational Science and Computational Intelligence*, pp. 735–740, 2021.

[28] S. Oesch and S. Ruoti, "That Was Then, This Is Now: A Security Evaluation of Password Generation, Storage, and Autofill in Browser-Based Password Managers," *Proceedings of the 29th USENIX Security Symposium*, p. 19, Aug. 2020.

[29] S. Oesch, S. Ruoti, J. Simmons, and A. Gautam, ""It Basically Started Using Me:" An Observational Study of Password Manager Usage," in *CHI Conference on Human Factors in Computing Systems*. New Orleans LA USA: ACM, Apr. 2022, pp. 1–23, https://dl.acm.org/doi/10.1145/3491102.3517534.

[30] PCI Security Standards Council, "Payment Card Industry Data Security Standard," Mar. 2022.

[31] S. Pearman, J. Thomas, P. E. Naeini, H. Habib, L. Bauer, N. Christin, L. F. Cranor, S. Egelman, and A. Forget, "Let's Go in for a Closer Look: Observing Passwords in Their Natural Habitat," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. Dallas Texas USA: ACM, Oct. 2017, pp. 295–310, https://dl.acm.org/doi/10.1145/3133956.3133973.

[32] S. Pearman, S. A. Zhang, L. Bauer, N. Christin, and L. F. Cranor, "Why people (don't) use password managers effectively," *USENIX Symposium on Usable Privacy and Security*, Aug. 2019.

[33] S. Seiler-Hwang, P. Arias-Cabarcos, A. Marín, F. Almenares, D. Díaz-Sánchez, and C. Becker, ""I don't see why I would ever want to use it": Analyzing the Usability of Popular Smartphone Password Managers," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. London United Kingdom: ACM, Nov. 2019, pp. 1937–1953, https://dl.acm.org/doi/10.1145/3319535.3354192.

[34] R. Shay, S. Komanduri, A. L. Durity, P. S. Huh, M. L. Mazurek, S. M. Segreti, B. Ur, L. Bauer, N. Christin, and L. F. Cranor, "Designing Password Policies for Strength and Usability," *ACM Transactions on Information and System Security*, vol. 18, no. 4, pp. 1–34, May 2016, https://dl.acm.org/doi/10.1145/2891411.

[35] R. Shay, S. Komanduri, P. G. Kelley, P. G. Leon, M. L. Mazurek, L. Bauer, N. Christin, and L. F. Cranor, "Encountering stronger password requirements: User attitudes and behaviors," *Proceedings of the sixth symposium on usable privacy and security*, p. 20, Jul. 2010.

[36] F. Stajano, M. Spencer, G. Jenkinson, and Q. Stafford-Fraser, "Password-Manager Friendly (PMF): Semantic Annotations to Improve the Effectiveness of Password Managers," in *Technology and Practice of Passwords*, S. F. Mjølsnes, Ed. Cham: Springer International Publishing, 2015, vol. 9393, pp. 61–73, http://link.springer.com/10.1007/978-3-319-24192-0_4.

[37] StatCounter Global Stats, "Browser Market Share Worldwide," May 2022, https://gs.statcounter.com/browser-market-share#monthly-202205-202205-bar.

[38] E. Stobert and R. Biddle, "The Password Life Cycle," *ACM Transac-*

tions on Privacy and Security, vol. 21, no. 3, pp. 1–32, Jun. 2018, https://dl.acm.org/doi/10.1145/3183341.

[39] P. Story, D. Smullen, R. Chen, Y. Yao, A. Acquisti, L. F. Cranor, N. Sadeh, and F. Schaub, "Increasing Adoption of Tor Browser Using Informational and Planning Nudges," *Proceedings on Privacy Enhancing Technologies*, vol. 2022, no. 2, pp. 152–183, Apr. 2022, https://petsymposium.org/popets/2022/popets-2022-0040.php.

[40] J. Tan, L. Bauer, N. Christin, and L. F. Cranor, "Practical Recommendations for Stronger, More Usable Passwords Combining Minimum-strength, Minimum-length, and Blocklist Requirements," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. Virtual Event USA: ACM, Oct. 2020, pp. 1407–1426, https://dl.acm.org/doi/10.1145/3372297.3417882.

[41] The Chromium Authors, "Password_requirements.proto," May 2018, https://chromium.googlesource.com/chromium/src/+/f5071d7e1124a71380c2740defa0a578f27b805d/components/autofill/core/browser/proto/password_requirements.proto#42.

[42] ——, "Create Amazing Password Forms," Apr. 2022, https://www.chromium.org/developers/design-documents/create-amazing-password-forms/.

[43] ——, "Server.proto," 2022, https://chromium.googlesource.com/chromium/src/+/refs/heads/main/components/autofill/core/browser/proto/server.proto.

[44] ——, "Password_generator.cc," Nov. 2023, https://chromium.googlesource.com/chromium/src/+/HEAD/components/password_manager/core/browser/generation/password_generator.cc#48.

[45] B. Ur, F. Noma, J. Bees, S. M. Segreti, R. Shay, L. Bauer, N. Christin, and L. F. Cranor, ""I Added '!' at the End to Make It Secure": Observing Password Creation in the Lab," *Eleventh Symposium On Usable Privacy and Security*, p. 18, 2015.

[46] B. Ur, S. M. Segreti, L. Bauer, N. Christin, L. F. Cranor, S. Komanduri, D. Kurilova, M. L. Mazurek, W. Melicher, and R. Shay, "Measuring Real-World Accuracies and Biases in Modeling Password Guessability," *24th USENIX Security Symposium (USENIX Security 15)*, Aug. 2015.

[47] S. Vinberg and J. Overson, "2021 Credential Stuffing Report," Feb. 2021, https://www.f5.com/labs/articles/threat-intelligence/2021-credential-stuffing-report.

[48] R. Wash, E. Rader, and R. Berman, "Understanding Password Choices: How Frequently Entered Passwords are Re-used Across Websites," *Twelfth Symposium on Usable Privacy and Security*, pp. 175–188, 2016.

[49] Wikimedia Foundation, "Dashiki: Simple Request Breakdowns," Jun. 2022, https://analytics.wikimedia.org/dashboards/browsers/#all-sites-by-browser.

[50] D. Winder, "Thousands Of PayPal Accounts Breached—Is Yours One Of Them?" Jan. 2023, https://www.forbes.com/sites/daveywinder/2023/01/19/thousands-of-paypal-accounts-hacked-is-yours-one-of-them/.

[51] A. Yamada, K. Crichton, Y. Sawaya, J.-D. Dong, S. Pearman, A. Kubota, and N. Christin, "On recruiting and retaining users for security-sensitive longitudinal measurement panels," in *Proceedings of the Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022)*, Aug. 2022.

[52] S. Zibaei, D. R. Malapaya, B. Mercier, A. Salehi-Abari, and J. Thorpe, "Do Password Managers Nudge Secure (Random) Passwords?" in *Proceedings of the Eighteenth Symposium on Usable Privacy and Security (SOUPS 2022)*, Aug. 2022, pp. 581–597.

[53] S. Zibaei, A. Salehi-Abari, and J. Thorpe, "Dissecting Nudges in Password Managers: Simple Defaults are Powerful," in *Proceedings of the Nineteenth Symposium on Usable Privacy and Security (SOUPS 2023)*, Aug. 2023, pp. 211–225.

APPENDIX A
NEW SAMPLE CHECKLIST

We followed this protocol when testing password managers on real-world websites. As shown in Figure 6, the protocol was displayed alongside the web form we used to collect data.

- Use this checklist to ensure you follow each step in the data collection process. Note that the checklist itself isn't saved to the database.
- Codes are organized based on where in the sampling workflow they occur.
- All samples should contain either a _BLOCK_ code, or the LOGIN_SUCCESS code.
- If there isn't a code to describe something that you think should be coded, use an OTHER code; we will revisit OTHER codes, and potentially recode them with new codes.
- Bold indicates extra steps you should follow when you apply a code.
- Any deviations from the protocol or checklist should be described in the Notes field.

1) Start streaming **and** locally recording your screen
2) Select the password manager you are using
3) Load the website's homepage
4) Register for an account on the website
   - Proceed through the registration process linearly (e.g., supply data from the top to the bottom of the page).
   - If a password manager prompts you to remember a password, update a password, create credentials, etc., accept the choice it offers.
   - If the password manager displays a form, correct any defaults which a typical user would know are incorrect. If the website has you create a username, store the username as the login ID – otherwise, use the email address as the login ID.
   - If a website rejects a password manager's password, edit the password directly on the website (e.g., appending an exclamation mark), or in a text editor (i.e., the user creates their own password).

   a) Use the password manager to suggest a password. **For now, don't customize or edit the password in any way.**
      - Note that Bitwarden never prompts for credential creation. Instead, you must manually add credentials to Bitwarden. **As soon as you reach the page with the password field,** scroll down so the password field is visible. Next, generate credentials and autofill the password field(s) by clicking on the item in Bitwarden.

   b) **Immediately** after the password manager has suggested a password, **collect** the sample using the bookmarklet, and **paste** it into the web application using the "Paste Sample Data" button.
      - Usually, the Password, URL, DOM, DOCTYPE, and User Agent fields will be filled automatically. If they aren't, describe this in the Notes field. Then, you can manually add the data (e.g., by copying the password from within the password manager).

   c) Supply appropriate information to complete account registration. You might need to edit the password, confirm your email address, etc.
      - Use the email address and phone number corresponding to the password manager you are testing.
      - Enter authentic information whenever possible. For example, use your name, use our institution's address, etc.
      - If a username is required, try a username matching the email address (e.g., "FIRST.LAST.safari"). If it's rejected, remove the periods. If it's still rejected, describe the username which was accepted in the Notes field.
5) Log out of the newly created account
6) Log back into the newly created account, using the password manager's autofill
7) Complete documentation of the sample:
   a) Ensure you applied all relevant codes
   b) Add a link to your screen recording
   c) Describe other notes in the "Notes" field

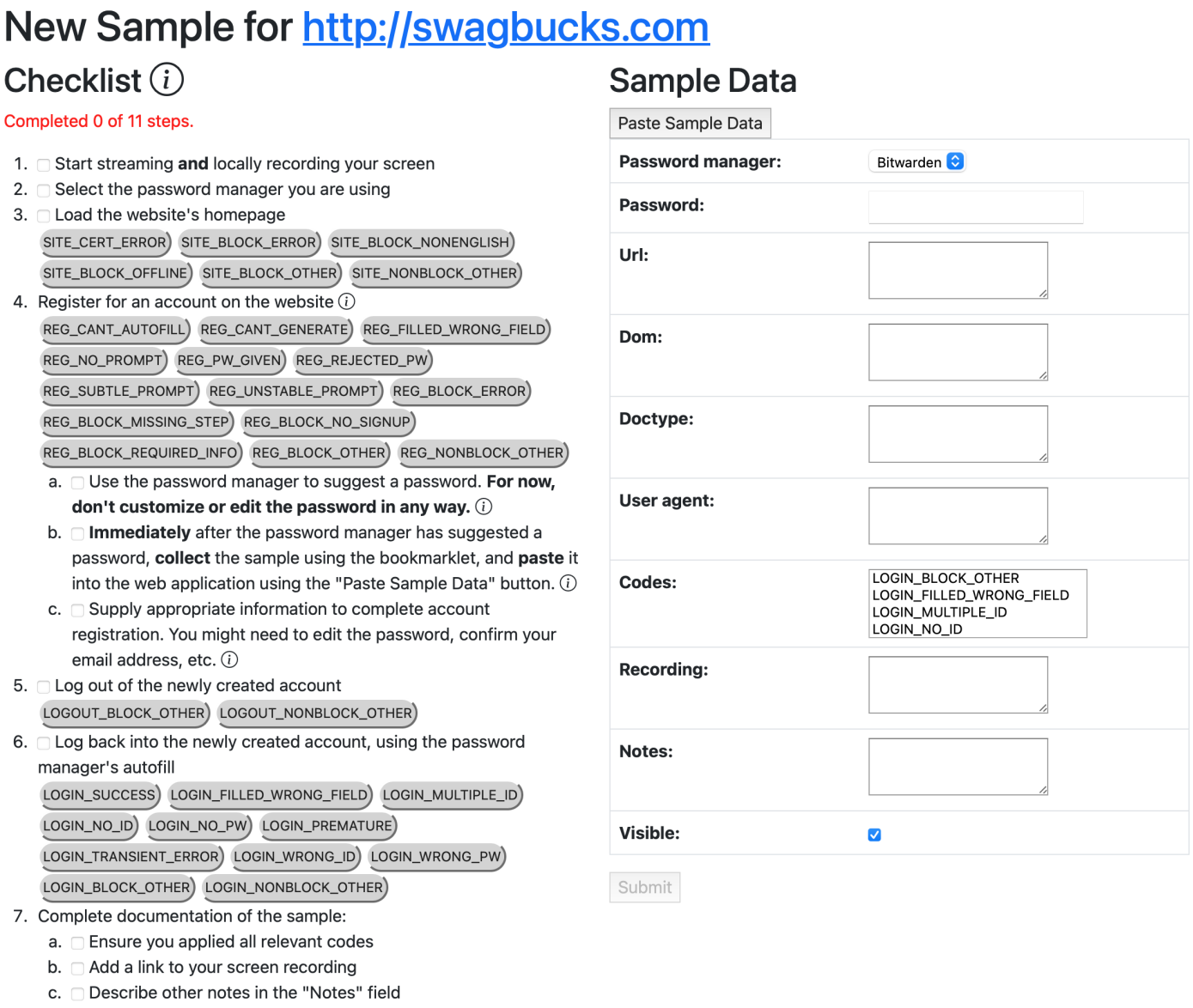

Fig. 6. A screenshot of the web form we used during data collection. The form included a checklist describing the data collection protocol. Clicking on the "info" icons showed more detailed instructions, which are included in Appendix A. Hovering over the codes showed the code descriptions listed in Table VII in Appendix B. We used a bookmarklet to record data from each webpage on which we generated passwords, including the URL of the page and the password suggested by the password manager.

APPENDIX B
SUPPLEMENTARY FIGURES AND TABLES

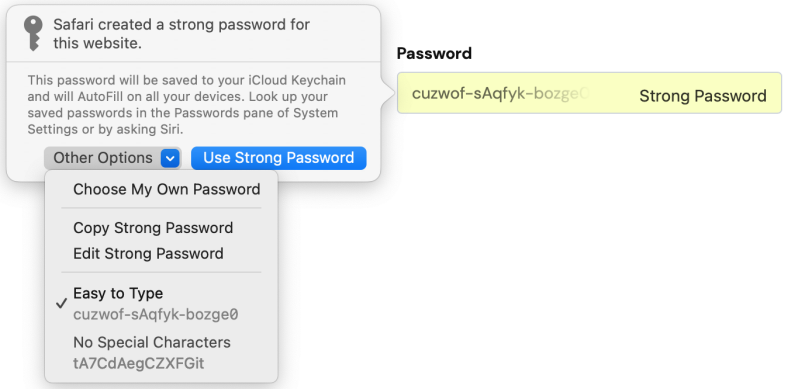

Fig. 7. We tested using Safari versions 15.5 to 15.6.1. Safari 15's user interface is depicted in Figure 4. This figure depicts Safari 16's new user interface for suggesting passwords, which makes it easier to customize password suggestions.

TABLE IV. We identified 57 standalone password managers using Google searches and app store searches.

Password Manager	**Google Play Store: Installs**	**Google Play Store: Ratings**	**iOS App Store: Ratings**	**macOS App Store: Ratings**	**Website URL**
Microsoft Authenticator	50,000,000+	910,535	172,858	Not available	https://www.microsoft.com/en-us/security/mobile-authenticator-app
LastPass	10,000,000+	217,131	46,282	Not available	https://www.lastpass.com/
Keeper	10,000,000+	95,505	158,672	1,284	https://www.keepersecurity.com
Dashlane	5,000,000+	174,227	68,467	671	https://www.dashlane.com
Norton Password Manager	1,000,000+	51,911	16,783	174	https://my.norton.com/extspa/passwordmanager
KeePassDroid	1,000,000+	37,375	Not available	Not available	https://www.keepassdroid.com
1Password	1,000,000+	34,164	28,211	2,657	https://1password.com
Bitwarden	1,000,000+	34,041	2,623	379	https://bitwarden.com
KeePass2Android	1,000,000+	31,668	Not available	Not available	https://philipp.crocoll.net/keepass2android/index.php
McAfee True Key	1,000,000+	26,430	1,342	Not available	https://www.truekey.com
Kaspersky Password Manager	1,000,000+	25,498	560	33	https://usa.kaspersky.com/password-manager
Password Manager SafeInCloud	1,000,000+	24,680	1,035	370	https://www.safe-in-cloud.com/en/
RoboForm Password Manager	500,000+	25,377	30,581	21	https://www.roboform.com
Enpass Password Manager	500,000+	16,641	1,041	249	https://www.enpass.io
NordPass Password Manager	500,000+	5,800	390	Not available	https://nordpass.com
Sticky Password	100,000+	7,897	992	Not available	https://www.stickypassword.com
Avira Password Manager	100,000+	5,630	278	2	https://www.avira.com/en/password-manager
mSecure	100,000+	4,713	38,812	11,468	https://www.msecure.com

Continued on the next page

Password Manager	Google Play Store: Installs	Google Play Store: Ratings	iOS App Store: Ratings	macOS App Store: Ratings	Website URL
Password Safe	100,000+	3,937	4,291	273	https://pwsafe.org
KeePassDX	100,000+	1,807	Not available	Not available	https://www.keepassdx.com
RememBear	100,000+	1,514	2,383	Not available	https://www.remembear.com
Zoho Vault Password Manager	50,000	907	479	5	https://www.zoho.com/vault/
Password Boss	10,000+	504	65	Not available	https://www.passwordboss.com
LogmeOnce	10,000+	410	164	Not available	https://www.logmeonce.com
Passwarden	10,000+	299	70	63	https://www.keepsolid.com/passwarden/
Password Manager Data Vault	10,000+	1,095	1,608	661	https://ascendo.co
LessPass	5,000+	132	2	Not available	https://www.lesspass.com/
Solarwinds Passportal	5,000+	89	44	Not available	https://www.passportalmsp.com
PassHub WWPass Key	5,000+	40	7	Not available	https://www.wwpass.com/passhub-enterprise-password-manager
Padloc	1,000+	Not given	4	Not available	https://padloc.app
Passbolt - password manager	1,000+	Not given	4	Not available	https://www.passbolt.com
Psono	1,000+	Not given	3	Not available	https://psono.com
PassCamp	1,000+	Not given	2	0	https://www.passcamp.com
KeepShare	500+	55	Not available	Not available	https://play.google.com/store/apps/details?id=com.hanhuy.android.keepshare
Passwork	500+	Not given	0	Not available	https://passwork.pro
Intuitive Password	100+	Not given	1	0	https://www.intuitivepassword.com
Password Manager - Safe Lock	Not available	Not available	16,738	Not available	https://www.approver-studio.com/password-tos
Password Manager ’	Not available	Not available	16,471	Not available	https://apps.apple.com/us/app/password-manager/id998953281
Strongbox	Not available	Not available	3,360	128	https://strongboxsafe.com
KeePassium	Not available	Not available	640	Not available	https://keepassium.com
KyPass	Not available	Not available	366	9	https://www.kyuran.be/software/kypass/
SyncPass	Not available	Not available	60	Not available	http://www.simpleanywhere.com/syncpass/
Minimalist: Password Manager	Not available	Not available	39	45	https://minimalistpassword.com
iKeePass	Not available	Not available	34	Not available	http://www.ikeepass.de
PassDrop 2	Not available	Not available	23	Not available	https://apps.apple.com/app/id1206056096
SamuraiSafe - Password Manager	Not available	Not available	15	20	https://samarama.net
Clipperz	Not available	Not available	Not available	Not available	https://clipperz.com
Encryptr	Not available	Not available	Not available	Not available	https://spideroak.com/personal/encryptr?ref=producthunt
Hypervault	Not available	Not available	Not available	Not available	https://www.hypervault.com
KeePass	Not available	Not available	Not available	Not available	https://keepass.info
KeePassXC	Not available	Not available	Not available	Not available	https://keepassxc.org

Continued on the next page

Password Manager	Google Play Store: Installs	Google Play Store: Ratings	iOS App Store: Ratings	macOS App Store: Ratings	Website URL
KyPass Companion	Not available	Not available	Not available	Not available	https://www.kyuran.be/software/kypass4mac/
MacPass	Not available	Not available	Not available	Not available	https://macpassapp.org
Pass (GPG)	Not available	Not available	Not available	Not available	https://www.passwordstore.org
Passwd.Team	Not available	Not available	Not available	Not available	https://passwd.team
Passwordix	Not available	Not available	Not available	Not available	https://apps.apple.com/app/id488134069
RatticDB	Not available	Not available	Not available	Not available	https://github.com/tildaslash/RatticWeb

TABLE V. We tested four password managers on these 100 websites. Data collection could not be completed on all websites, for the reasons noted in this table. We initially tried to create accounts on the domains in the SBO data, but in some cases we created accounts on related domains. For example, we created accounts on Costco's U.S. website, instead of its Mexican website. Websites are ordered by the Tranco popularity rank of their SBO domain.

Tranco Rank	SBO Domain	Completed Data Collection?	Sampled Domain
1	google.com	Yes	accounts.google.com
4	facebook.com	Yes	www.facebook.com
8	amazon.com	Yes	www.amazon.com
9	yahoo.com	Yes	login.yahoo.com
11	linkedin.com	Yes	www.linkedin.com
14	reddit.com	Yes	www.reddit.com
16	live.com	Yes	signup.live.com
17	taobao.com	No, website wasn't available in English	
27	ebay.com	Yes	signup.ebay.com
28	vimeo.com	Yes	vimeo.com
34	paypal.com	Yes	www.paypal.com
165	indeed.com	Yes	secure.indeed.com
239	steampowered.com	Yes	store.steampowered.com
248	samsung.com	Yes	account.samsung.com
306	xfinity.com	No, required info we couldn't supply	
308	capitalone.com	No, required info we couldn't supply	
363	messenger.com	No, didn't offer the option to sign up	
520	adp.com	No, required info we couldn't supply	
572	hilton.com	Yes	www.hilton.com
709	t-mobile.com	No, required info we couldn't supply	
829	fidelity.com	No, required info we couldn't supply	
832	nyu.edu	No, required info we couldn't supply	
863	delta.com	Yes	www.delta.com
879	norton.com	Yes	login.norton.com
884	irctc.co.in	Yes	www.irctc.co.in
898	aa.com	Yes	www.aa.com
991	pandora.com	Yes	www.pandora.com
996	nintendo.com	Yes	accounts.nintendo.com
1,060	uber.com	Yes	auth.uber.com
1,112	comcast.net	No, required info we couldn't supply	
1,343	rei.com	Yes	www.rei.com
1,535	pitt.edu	Yes	cfopitt.taleo.net
1,578	syf.com	No, required info we couldn't supply	
2,097	monster.com	Yes	identity.monster.com
2,217	swagbucks.com	Yes	www.swagbucks.com
2,643	23andme.com	Yes	auth.23andme.com
2,702	icims.com	No, required info we couldn't supply	
2,980	easyjet.com	Yes	www.easyjet.com
3,049	citibankonline.com	No, required info we couldn't supply	
3,363	match.com	Yes	www.match.com
3,534	creditsesame.com	No, required info we couldn't supply	
3,715	wikibuy.com	Yes	capitaloneshopping.com

Continued on the next page

Tranco Rank	SBO Domain	Completed Data Collection?	Sampled Domain
3,858	bc.edu	Yes	bc.csod.com
4,947	geico.com	No, required info we couldn't supply	
7,369	elvenar.com	Yes	us.elvenar.com
8,228	clarivate.com	No, unable to complete registration	
8,509	niche.com	Yes	www.niche.com
8,675	neopets.com	Yes	www.neopets.com
9,905	confirmit.com	No, required info we couldn't supply	
10,608	fnb-online.com	No, required info we couldn't supply	
10,896	slb.com	Yes	www.slb.com
13,761	eurostar.com	Yes	login.eurostar.com
21,923	playerauctions.com	Yes	account.playerauctions.com
24,148	roadrunnersports.com	Yes	www.roadrunnersports.com
24,746	stanfordchildrens.org	No, required info we couldn't supply	
25,481	splitwise.com	Yes	www.splitwise.com
26,140	erieinsurance.com	No, required info we couldn't supply	
30,084	mercer.edu	No, didn't offer the option to sign up	
42,440	gofobo.com	Yes	gofobo.com
42,853	costco.com.mx	Yes	signin.costco.com
45,886	cybercoders.com	Yes	www.cybercoders.com
47,282	hmfusa.com	No, required info we couldn't supply	
49,011	decluttr.com	Yes	account.decluttr.com
50,886	universitytickets.com	No, required info we couldn't supply	
53,502	f1000.com	Yes	f1000research.com
62,203	rue21.com	Yes	www.rue21.com
75,872	duquesnelight.com	No, required info we couldn't supply	
76,894	aladtec.com	Yes	secure17.aladtec.com
88,340	helpowl.com	Yes	www.helpowl.com
95,775	socialmedialink.com	Yes	smiley.socialmedialink.com
97,055	urbansitter.com	Yes	www.urbansitter.com
98,821	engageme.tv	Yes	hideout.co
103,494	cajungrocer.com	Yes	www.cajungrocer.com
111,701	convergentcare.com	No, required info we couldn't supply	
116,753	vocellipizza.com	Yes	weborder.vocellipizza.com
131,261	loyaltyplus.aero	Yes	ffkp.loyaltyplus.aero
132,479	peoples-gas.com	No, required info we couldn't supply	
133,621	mod.gov.il	No, didn't offer the option to sign up	
141,708	userinterviews.com	Yes	www.userinterviews.com
152,079	userlytics.com	Yes	www.userlytics.com
164,949	smartpanel.io	No, error on homepage or wouldn't load	
183,717	springboardonline.org	No, required info we couldn't supply	
185,528	markten.com	No, didn't offer the option to sign up	
201,199	orderup.com	Yes	pizzademo.orderup.com.au
207,344	adpserviceedge.com	No, error on homepage or wouldn't load	
213,513	fond.co	No, required info we couldn't supply	
215,449	malakye.com	Yes	www.malakye.com
294,975	reviewr.com	No, required info we couldn't supply	
374,674	joinbain.com	No, required info we couldn't supply	
444,553	versatilephd.com	Yes	versatilephd.com

Continued on the next page

Tranco Rank	SBO Domain	Completed Data Collection?	Sampled Domain
445,628	gametame.com	Yes	gametame.com
689,640	pittplusme.org	Yes	pittplusme.org
761,514	rewardshopping.com	No, error on homepage or wouldn't load	
844,307	free1040taxreturn.com	No, required info we couldn't supply	
855,904	reportez.com	No, error on homepage or wouldn't load	
882,626	recipeshubs.com	No, error on homepage or wouldn't load	
1,925,217	turkernation.com	No, error on homepage or wouldn't load	
Undefined	butteredcatlabs.com	Yes	gist.butteredcatlabs.com
Undefined	REDACTED.com	No, didn't offer the option to sign up	
Undefined	seabournacademy.com	Yes	seabournacademy.com

TABLE VI. During our testing, 19 websites rejected passwords generated by at least one password manager. Websites' stated reasons for rejecting our passwords are shown in this table. Sometimes a website only rejected one researcher's password, so we also report the number of passwords rejected.

Sampled Domain	**Passwords Rejected**	**Stated Password Policy**
signup.ebay.com	Keeper (1/2)	After filling the password field: "Please remove the symbol you entered and try a different one."
vimeo.com	Bitwarden (2/2)	After submitting the form: "Password must be at least 8 characters long and must contain at least one number and at least one symbol"
account.samsung.com	Bitwarden (2/2)	Submission is disabled until you click the password field: "Passwords must: Use 8 or more characters with a mix of letters, numbers, and symbols."
www.delta.com	Keeper (2/2)	After clicking the info icon: "MUST BE • Between 8 and 20 characters • At least 1 number • At least 1 uppercase letter • At least 1 lowercase letter CANNOT CONTAIN • The "@" or "¡" symbols • Your SkyMiles number, email or username • More than 3 special characters"
login.norton.com	Bitwarden (2/2)	After filling the password field: "Your password must be 8 characters or longer and include the following: • Upper and lower case letters • At least 1 number • At least 1 symbol"
www.irctc.co.in	Keeper (2/2), Safari (1/2)	After filling the password field: "Password is invalid. Min 8 character & Max 15 character. Password must contain at least one small & one capital alphabet and numeric digit."
www.aa.com	Keeper (2/2)	After clicking the info icon: "Password requirements • 6-16 characters • Any combination of special characters, letters and numbers • No spaces before the first, or after the last characters"
cfopitt.taleo.net	Keeper (1/2)	After submitting the form: "The password you entered is not valid Please note that the password must respect the following rules: • It must contain between 6 and 32 characters. Use only characters from the following set: ! # $ % & ( ) * + , - . / 0123456789 : ; ¡ = ¿ ? @ ABCDEFGHIJKLMNOPQRSTUVWXYZ [\] _ ` abcdefghijklmnopqrstuvwxyz { — } - • It must contain at least 1 letter(s) (ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz). • It must contain at least 1 numeric character(s) (0123456789). • It must not contain more than 2 identical consecutive characters (AAA, iiii, $$$$$ ...). • It must not contain your user name."

Continued on the next page

Sampled Domain	Passwords Rejected	Stated Password Policy
www.easyjet.com	Keeper (2/2)	Before interacting with the page: "Password minimum requirements: • Be between 10-20 characters • Contain at least 1 number • Contain at least 1 uppercase letter • Contain at least 1 lowercase letter • Not start with 0 • Not contain # & + = or space"
www.match.com	Keeper (2/2)	Before interacting with the page: "You can use letters, numbers & underscores."
capitaloneshopping.com	Bitwarden (2/2), Safari (2/2)	After clicking or filling the password field: "Passwords must: • be between 8 and 64 characters long • contain at least 1 uppercase and 1 lowercase letter • contain at least 1 special character (ie. !()@#$%^&*) • not repeat the same character more than twice • not be similar to your email address"
www.niche.com	Keeper (2/2), Safari (2/2)	After filling the password field or submitting the form: "Passwords must contain only letters, numbers, and !@#$%^&*"
www.neopets.com	Keeper (1/2), Safari (2/2)	After submitting the form: "You may only use the letters A through Z, the numbers 0 through 9, or !, @, #, %, ^, &, *, $, +, ., _, (, or). Your password must have at least 2 numbers in it."
account.playerauctions.com	Keeper (2/2), Safari (2/2)	After submitting the form: "Your password must contain a combination of letters and numbers, special characters can be used, and the length is between 8-16"
gofobo.com	Bitwarden (2/2), Keeper (2/2), Safari (2/2)	After filling the password field: "Please enter no more than 18 characters." "Password needs to be a minimum length of 8, and it must contain uppercase and lowercase letters, numbers, and one of the following special characters: @ $! # * ? &."
signin.costco.com	Bitwarden (2/2), Keeper (2/2)	After clicking or filling the password field: "Password must include the following: • Use between 8 and 16 characters • Include at least one lowercase (a-z) and one uppercase letter (A-Z) • Include at least one special character (e.g. !@#$&) • Does not contain blank spaces or the following special characters: ¡ ¿ , • Include at least one digit (0-9) • Passwords match" Although the password policy is shown after clicking the password field, Keeper's user-interface covers most of the policy.

Continued on the next page

Sampled Domain	Passwords Rejected	Stated Password Policy
www.urbansitter.com	Bitwarden (2/2)	After submitting the form: "Passwords must meet the following requirements: • 8 character minimum • 1 lowercase letter • 1 uppercase letter • 1 number • 1 symbol"
ffkp.loyaltyplus.aero	Keeper (2/2), Safari (2/2)	After submitting the form: "Please specify a password between 6 and 15 characters which contains at least one alpha and one numeric character."
pittplusme.org	Bitwarden (2/2), Chrome (2/2)	After clicking the password field: • "Must be at least 8 characters • Must have at least one special character • Must have at least one number" After submitting the form: "Passwords must have at least one special character. (E.g. '$ % ^ ! _ etc.')"

TABLE VII. The number of websites on which either researcher encountered each code. BLOCK codes explain why data collection on a website couldn't be completed. The "Any PWM" column gives the total number of websites on which any researcher encountered each code when using any password manager.

Code	Description	Bitwarden	Chrome	Keeper	Safari	Any PWM
SITE BLOCK ERROR	The website's homepage gives an error that we cannot resolve.	2	2	2	3	3
SITE BLOCK NONENGLISH	The website isn't in English, and doesn't offer an English version.	1	1	1	1	1
SITE BLOCK OFFLINE	The website's homepage wouldn't load.	4	4	4	4	4
SITE CERT ERROR	The website gives a certificate error. In this case, ignore the error and proceed to the website.	1	1	1	1	1
REG BLOCK MISSING STEP	The registration process ends prematurely, the registration process cannot be completed because a necessary email isn't received, etc.	1	1	1	1	1
REG BLOCK NO SIGNUP	The website doesn't offer the option to sign up for an account.	5	5	5	5	5
REG BLOCK REQUIRED INFO	Registering for the website requires information that we cannot supply.	26	26	26	26	26
REG CANT AUTOFILL	The password manager can't fill the password field, even when explicitly clicking the "Fill" button. Doesn't apply if the main password field is filled, but the "Confirm Password" field isn't filled.	0	0	3	0	3
REG CANT GENERATE	The password manager can't generate a password for the website, even when performing non-standard interactions to try to do so (e.g., right-clicking, etc.).	0	1	0	0	1
REG FILLED WRONG FIELD	When registering for a website, the password manager puts the username and/or password into field(s) that aren't appropriate for that data.	6	0	6	0	11
REG NO PROMPT	The password manager doesn't prompt you to use an automatically generated password unless you perform a non-standard interaction. If this occurs, you should manually kickstart the generation process (e.g., by right-clicking in the password field, clicking an already active password field, etc.).	Not applicable, since Bitwarden never prompts users to generate passwords.	12	7	3	15

Continued on the next page

Code	Description	Bitwarden	Chrome	Keeper	Safari	Any PWM
REG PW GIVEN	The password is generated by the website. In this case, we give the password manager a chance to store the password by logging in. Then, we test that the password manager can autofill the password by logging out and back in again.	1	1	1	1	1
REG REJECTED PW	The website rejected the automatically generated password. Don't apply this code if the website warns that the password won't be accepted, but actually does accept it.	8	1	13	7	19
REG SUBTLE PROMPT	The password manager doesn't prompt you to use an automatically generated password in its usual way. Instead, it displays a more subtle prompt.	0	0	0	5	5
REG UNSTABLE PROMPT	The password manager's password generation or credential registration prompt disappears in a way that makes it difficult to use.	0	0	9	0	9
LOGIN FILLED WRONG FIELD	When logging in to a website, the password manager puts the username and/or password into field(s) that aren't appropriate for that data.	1	0	0	0	1
LOGIN MULTIPLE ID	The password manager offers to autofill the username, email, etc. with multiple choices, some correct, some incorrect. Only applies when the user must choose between these options.	8	1	2	0	10
LOGIN NO ID	At the time of login, the password manager hadn't stored a username, email, etc. for the account.	0	2	0	3	5
LOGIN NO PW	At the time of login, the password manager hadn't stored a password for the account.	0	1	0	1	2
LOGIN PREMATURE	The password manager prematurely submits the login form (e.g., before an additional identifier is typed by the user), causing an unsuccessful login.	0	0	1	0	1
LOGIN SUCCESS	You were able to successfully log back into your account.	61	61	61	61	61
LOGIN TRANSIENT ERROR	At first, the website doesn't accept the credentials autofilled by the password manager, but after retrying (e.g., refreshing the login page, logging out and back in again, etc.), the credentials are accepted.	0	1	6	0	7

Continued on the next page

Code	Description	Bitwarden	Chrome	Keeper	Safari	Any PWM
LOGIN WRONG ID	The password manager fills an incorrect username, email, etc. for the account.	8	5	13	4	18
LOGIN WRONG PW	The password manager fills an incorrect password for the account.	5	1	14	1	18