An Analysis of Password Managers' Password Checkup Tools

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

Collins W. Munyendo
cmunyendo@gwu.edu
The George Washington University
USA

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

Peter Mayer
mayer@imada.sdu.dk
University of Southern Denmark
Denmark

ABSTRACT

Password managers (PMs) have been widely recommended to users to generate and store random, secure, and unique passwords across websites. Using a PM is often not enough however, especially if users store passwords that are guessable, or have been breached. To assist users in updating insecure passwords, PMs come with "checkup" features that report the strength of users' passwords. However, there has yet to be a systematic study of the features offered as part of these checkups, and the consistency of the checkup advice across different PMs. In this paper, we conduct a preliminary analysis of 14 PMs' password checkup features, recording how many passwords are reported weak and compromised. We find that many PMs fail to report breached credentials. Weak passwords were also under-reported by PMs. This analysis forms the basis for a larger study on the consistencies of PM checkup tools and how users perceive and use them.

CCS CONCEPTS

• **Security and privacy → Usability in security and privacy**.

KEYWORDS

password managers, authentication

ACM Reference Format:
Adryana Hutchinson, Collins W. Munyendo, Adam J. Aviv, and Peter Mayer. 2024. An Analysis of Password Managers' Password Checkup Tools. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3613905.3650741

1 INTRODUCTION

It is widely recommended that users employ a password manager (PM) for managing their online credentials due to the demands that passwords have to be strong, secure, and unique across accounts. Even still, password reuse is common [15, 32], and some users, even when using a PM, choose their own passwords that may be insecure [48, 49]. Furthermore, even strong, secure, and unique passwords stored in a PM may still be leaked through data breaches, necessitating password changes. PMs' password checkup tools can be helpful in nudging users to make these changes.

Despite their benefits, the accuracy and consistency of PMs' checkup tools' feedback across PMs has received little attention. In this paper, we perform an initial comparison of check tools across PMs. Through our analysis, we seek to answer the following two research questions:

RQ1 Do password checkup tools provide accurate reporting of weak and breached passwords?
RQ2 Are password checkup tools consistent in their feedback across different password managers? (e.g., reporting similar duplicate, weak, or breached passwords.)

In answering these questions, we built a test corpus of passwords to empirically measure the sophistication of checkup tools. In total, our test corpus consisted of 1990 passwords. We generated 315 passwords of different classes, and collected 349 known breached passwords from SecLists [22] and Have I Been Pwned [35]. An additional 1326 variants of breached passwords were also generated by RSmangler [21].

Based on user downloads, ratings, and reviews, we then selected 14 popular PMs that have password checkup features for evaluation, i.e. LastPass, Norton, Roboform, Enpass, SafeInCloud, Dashlane, 1Password, Sticky Password, Keeper, KeePassXC, Bitwarden, Google Chrome's PM, Safari's Keychain, and Microsoft Edge's PM. For each PM evaluated, we uploaded the password corpus and then extracted the password checkup tools' feedback for the corpus.

We find that PMs inconsistently report both weak and compromised passwords across their checkup tools. No single PM successfully marked all known breached passwords as compromised even when credentials were from publicly available databases. Additionally, we find that some PMs report less weak passwords compared to others, despite many using the same password strength algorithm. We also find that most PMs use the same database (Have I Been Pwned) of passwords, which can create a single point of failure for users of PMs.

These results offer a number of future directions, particularly around how users interpret and take action based on the feedback of different PM checkup tools and what design features of these tools lead to greater and more acute action on the part of users. We discuss future directions in conclusion to this preliminary work.

CHI EA '24, May 11–16, 2024, Honolulu, HI, USA

ACM ISBN 979-8-4007-0331-7/24/05
https://doi.org/10.1145/3613905.3650741

2 BACKGROUND

To alleviate the burden of having to remember complex passwords, password managers (PMs) generate and store strong, random, and unique passwords for users across accounts. PMs fall under three broad categories [25]:

(a) *Browser-based*: PMs that come automatically installed as part of a browser, such as Google Chrome and Firefox.
(b) *Third-party*: Separately installed applications that can be sold commercially on a fee-based subscription, but open source and free options are available as well. Users must install these PMs separately.
(c) *System*: PMs that are automatically installed as part of operating systems, such as Apple's iCloud Keychain.

Despite a wide variety of PMs, bad password habits still persist, with prior work finding that users of PMs, and particularly browser-based and system PM users, still reuse passwords across accounts [25, 33]. Other research [25] has additionally shown that most PM users generate passwords themselves, only using PMs to store them. Unfortunately, even among PM users, bad password practice seems to persist.

To increase the security of users' accounts, many PMs measure the strength of passwords in their vault using security audit tools. These audit tools (also called checkups) notify users of weak and reused passwords, with some reporting credentials found in breached datasets. While breach reporting is usually locked behind a subscription fee in third-party PMs [3], browser/system PMs generally include breach reports for free. Checkup tools vary in terms of how they report a weak password. For example, a password may be marked as "weak" because of the PM's password strength checker, or it could be marked as weak because it was found in a data breach. Some PMs also classify passwords as weak if they are reused, or because the password has an expiration date (set by the user). We explore how each PM reports weak/breached passwords in Section 5 of the results.

Despite these tools' potential to nudge users to change vulnerable passwords, prior work has found that users often find these audits overwhelming [30], leading them to ignore them altogether. Additionally, if password strength is conveyed poorly, check-up tools may mislead users into thinking they have secure passwords when they do not. For example, previous work on password strength meters has shown that more stringent meters can increase the likelihood of stronger user-created passwords [43, 46], but poorly-configured ones may give users a false sense of security. Our study systematically evaluates password checkup tools across various PMs to show, for the first time, how these tools provide inconsistent alerts for various password classes.

3 RELATED WORK

Password Creation Habits. Users are advised to create long and complex passwords for every one of their online accounts. Unfortunately, users struggle with creating and remembering random, secure, and unique passwords across sites, especially because of the many accounts they have [34]. To alleviate the mental burden of remembering complex passwords, users often create and reuse memorable but guessable passwords [15, 32, 45, 47, 51]. For instance, user-created passwords often consist of predictable word sequences, with common phrases [48] and personal information (such as names, dates, and government IDs) [20, 49] frequently used. Even longer sentence-based password generation strategies often lead to predictable passwords [54]. Slightly altering existing passwords to meet specific password requirements is also common, such as adding a symbol to the end of a password [32, 47]. Password reuse across multiple websites can leave users susceptible to credential stuffing attacks, where previously breached credentials are used to attempt logins into different websites en masse. Attackers using password cracking tools like John the Ripper [31] can easily generate variants of breached passwords, leaving even heavily altered passwords highly vulnerable to attack. For PM checkup tools to work effectively, their strength meters must accurately report weak passwords, especially ones that are easily cracked by modern cracking tools. As part of our testing, we analyze how many PMs mark vulnerable (but altered) passwords as weak.

Motivation for Password Manager Usage. People use PMs for a variety of reasons. Pearman et al. found that users of pre-installed (system) and browser PMs used them out of convenience, while those who use 3rd-party PMs use them out of security concerns [33]. Unfortunately, PM adoption amongst the general population remains low. Common factors for low adoption rates are a lack of trust in PM software [4, 14, 24, 25, 36, 41], users' own threat assessment of password compromise [4, 5, 41, 49], and perceived ease-of-use and install time [2, 25, 36]. Stobert and Biddle also found that ease-of-use influenced which credentials security experts used PMs for [42]. Munyendo et al. also found that people who switch PMs do so out of usability concerns with the PM they use [29]. Nonetheless, Lyastani et al. found that when users used password generation tools, it significantly reduced the chance of them using weak passwords for their accounts [23]. To promote the use of password generation tools, it is important for checkup software to accurately (and consistently) report weak passwords in a users' vault. This is the focus of our study.

User Breach Awareness. PMs should notify users in a comprehensive way if a credential breach is detected during a checkup. Through an analysis of PM usage, Oesch et al. found that users were often overwhelmed with the amount of check-up notifications generated by audit tools [30]. While most users did find the notifications helpful, they recommended that audit tools should be more proactive in prompting users to change passwords, especially for high-value accounts. This is consistent with Karunakaran et al.'s findings, where participants expected proactive measures to be taken when credentials have been compromised in a data breach [17].

A recent study by Huang et al. found that Chrome's password breach notification was missing critical information, such as where a password breach was coming from, leading to confusion and mistrust from users. Participants also found that there was a lack of information on how to remediate leaked passwords [16]. Prior work has demonstrated that critical information is often not displayed in breach notifications. A study by Redmiles found that the lack of information in breach notifications caused users to be uncertain how to proceed in securing their online accounts [37]. Zou et al.'s study highlighted poor breach notification design, with the

Table 1: Popular third-party PMs identified through Google, Google Play, and iOS App Store.

Name of PM	Google Play Store Installs	Google Play Store Reviews	iOS App Store Ratings	Has Checkup Tool?	Free Trial?
LastPass	10,000,000+	22,900+	52,100+	Y	Y
Keeper	10,000,000+	96,800+	158,900+	Y	Y
Dashlane	5,000,000+	195,000+	3,500+	Y	Y
Norton PM	1,000,000+	71,800+	26,100+	P	-
Bitwarden	1,000,000+	45,000+	4,100+	Y	Y
McAfee True Key	1,000,000+	27,900+	1,500+	N	-
KeePass2Android	1,000,000+	33,700+	Not Available	Y	-
Kaspersky PM	1,000,000+	30,300+	913	Y	Y
Roboform	1,000,000+	14,000+	38,100+	Y	-
Enpass	1,000,000+	19,100+	1,200+	Y	-
NordPass	1,000,000+	14,000+	3,100+	P	Y
Password Safe	1,000,000+	51,300+	1,100+	N	-
1Password	500,000+	5,900+	1,500+	Y	Y
Avira	500,000+	8,120+	410	Y	N
SafeInCloud	100,000+	33,900+	1,900+	Y	-
Sticky Password	100,000+	8,920+	1,200+	Y	Y
mSecure	100,000+	5,340+	44,700+	P	-
Zoho Vault PM	50,000+	1,110+	685	Y	Y
Password Boss	10,000+	580	68	Y	Y
PassBolt	10,000+	351	1	N	-

Y = Both breach and password strength detection/Has free trial;
P = No breach detection;
N = No checkup tools/No free trial for premium features;
- = Not applicable

majority of notifications using overly-complex language. Phrasing that minimized risk was also observed in their study [55].

It is paramount that checkup tools offer a comprehensive and accessible way for users to remediate weak and breached passwords. There must be careful consideration when designing notifications, as well as clear instructions on how to update insecure passwords, especially since prior work found that victims were more receptive of changing passwords than other actions [26]. Otherwise, users may not change the password at all, or change it to something equally insecure. Bhagavatula et al. found that when users changed passwords after a breach, the new passwords were generally the same strength as the old ones, and more similar to the other passwords they already use [6].

4 METHODOLOGY

4.1 Collecting Passwords

Mangled Passwords. To understand how checkup tools classify breached passwords, we gathered passwords from SecLists [22], a database of previously-leaked passwords. Previous work has noted that most users tend to slightly alter passwords after a data breach [10, 47, 50], often appending symbols or including substrings from other passwords. To understand how sophisticated each PM is at detecting slightly-altered (but still breached) passwords, we fed "mangled" versions of breached passwords into each PM and recorded how many passwords were reported as breached and weak. Importantly, a password can be marked as breached, but still be considered a "strong" password by a given PM (Tables 2, and 3 illustrate this divide); this is due to how each individual PM configures its password strength meter. We used RSmangler [21] to create variants of SecList's 200 most common passwords of 2020 [28]. We performed a broad range of modifications to each compromised password to account for ways a user **might** change a password after a breach [47].

Specifically, we performed the following modifications:

- *LEET passwords.* We slightly modified spellings of compromised passwords to generate 735 passwords.
- *Double passwords.* Each compromised passwords was concatenated with itself twice, generating 197 passwords.
- *Reversed passwords.* Each compromised passwords was reversed, generating 197 passwords.
- *Reverse case passwords.* We reversed the case of each compromised password, generating 197 passwords.

Breached passwords. Additionally, we collected 309 **unmodified** breached passwords from SecLists [1]. We collected passwords of diverse classes to determine if checkup tools detect breached passwords of different compositions more effectively than others. We also collected 40 breached passphrases by manually entering common phrases into Have I Been Pwned (HIBP) [35], a popular database of breached credentials. We collected a total of 349 vulnerable passwords.

We categorize the breached passwords as follows:

- *C-Class3 passwords.* Compromised passwords that must have 3 different character types: uppercase or lowercase letters, numbers, symbols. 112 passwords were found.
- *C-Class4 passwords.* Compromised passwords that must have 4 different character types: uppercase or lowercase letters, numbers, symbols. 197 passwords were found.
- *C-Passphrases.* Compromised passwords composed of 4-5 words, separated by spaces. 40 passwords were found.

Classed Passwords. Finally, to understand how different PMs classify passwords of different classes, we also generated 315 random passwords of various class and length.

We generated[2] the following:

- *Basic passwords.* Passwords with only alphabetical letters. 80 were generated.
- *Class3 passwords.* Passwords with 3 different character types: uppercase or lowercase letters, numbers, symbols. 80 were generated.

[1]We gathered passwords from SecList's rockyou-75, NordVPN, Honeynet, Hotmail, and MySpace breach corpuses.
[2]We generated passwords using https://passwordsgenerator.net/ and https://www.useapassphrase.com/.

- *Class4 passwords*. Passwords with 4 different character types: uppercase or lowercase letters, numbers, symbols. 80 were generated.
- *Passphrases*. Passwords composed of 4, 5, and 12 words, separated by hyphens. 75 were generated.

To get a complete perspective of PM checkup tools, we chose browser, system, and third-party PMs to test. For popular browser and system PMs, we tested Google Chrome's, Microsoft Edge's, and Safari's Keychain PM. We excluded Firefox because their PM checkup tool does not show the strength of individual passwords.

We identified third-party PMs by searching "password manager" on Google, Apple's app store, and Google Play's app store. We recorded the first page of results from each search, and narrowed down PMs using Google Play Store and iOS App Store reviews and ratings [3]. We then filtered out PMs that did not have free trials and checkup features. Finally, we prioritized PMs that have both breach detection and password strength checkers, though we did select one popular PM that did not have breach detection (Norton PM). The following third-party password managers are analyzed: LastPass, 1Password, Dashlane, Keeper, Bitwarden, Roboform, Enpass, Norton PM, SafeInCloud, and Sticky Password (Table 1). We also chose to look at KeePassXC (KeePass) since it is an open-source, longstanding PM that is supported on all platforms (such as KeePass2Android for Android). We chose not to test Kaspersky PM because its free version does not allow more than 5 credentials in its vault.

4.2 Testing Credentials

Users receive checkup reports in two ways: either by manually requesting a report, or the report is automatically generated by the PM. These reports list vulnerable passwords found in the users' vault, and users can then choose to edit/remove the credentials. We performed checkups through each PMs' interface. When the checkup was finished, all weak and compromised credentials were documented, and all passwords were deleted from the vault afterward. After collecting passwords and selecting PMs to test, we uploaded each Password Class corpus to every PM. We then recorded how many passwords were marked as weak and compromised by the PM, exporting the weak/compromised password datasets afterwards. Passwords were then deleted from each PM vault. We repeated this process until all Password Class corpuses were processed.

Importantly, we were unable to collect breach information for a number of PMs we tested. Norton PM, for example, does not have breach detection features. Google Chrome will only report compromised passwords when both a username and password pair is found in a breach [44]. Because we used a dummy username for every credential, we were unable to gather breach information from Chrome, Microsoft Edge [27], Dashlane [12], and LastPass [19].

5 RESULTS

5.1 RQ1: Accuracy

Weak Passwords. To understand how PMs measure the strength of passwords, we searched through each PMs' interface and installation website (including company forums) for available information. The majority of PMs (Enpass [13], Roboform [38], SafeInCloud [40], Dashlane [11], 1Password [1], KeePassXC [18], Bitwarden [7]) use zxcvbn, a popular password strength estimator that calculates how weak a password is by comparing it to several word dictionaries [52].

We further found that PMs were inconsistent in reporting weak passwords (Table 2), despite many using zxcvbn to calculate password strength. Alarmingly, no PM marked all compromised passwords as weak, though KeePassXC did come close, reporting all compromised **and** mangled passwords as weak, but did miss 4 compromised *Class4* passwords.

Despite KeePassXC's performance, end-users using the 13 additional PMs may experience issues. We found that some PMs do not mark breached passwords as weak, meaning that if a user does not perform a compromised credential checkup (or does not pay a premium subscription for one), they may mistakenly assume their compromised credential is a strong password. Similarly, if a user switches PMs to one that reports fewer passwords as weak (or vice versa), they may be mislead into thinking their passwords are (in)secure. This could exacerbate bad password practices, putting users' online accounts at risk.

Compromised Passwords. Similar to how we discovered how PM's measured password strength, we looked at each PMs' interface and website for any information on how they detect credential breaches. We found that the majority of PMs (RoboForm [38], Enpass [13], SafeInCloud [39], 1Password [1], Bitwarden [7]) use HIBP. Sticky Password uses a different breach detection service called ARC Database [9]. We could not find any information on what credential databases Google Chrome, KeePassXC, Safari, or Microsoft Edge use.

We found that compromised passwords were reported more consistently compared to weak passwords, which is likely due to many PMs using the same database of breached passwords. However, only one password corpus (*C-Class3*) had all breached passwords consistently marked as compromised by PMs. The fact that the checkup tools did not detect all compromised passwords in each corpus is highly concerning, especially considering that all compromised passwords we used can be easily accessed publicly (or generated using RSmangler). These passwords are known weak passwords that should not be used, even when heavily altered. As prior work has demonstrated, users will create predictable passwords [15, 32, 45, 47, 51]. We suggest that breach detection features in PMs compare users' credentials to more datasets, especially public ones (such as SecLists). Otherwise, PMs may miss obvious examples of insecure passwords.

5.2 RQ2: PM Comparison

Weak Passwords. We found that most PMs consistently marked mangled and compromised passwords weak at least 50% of the time. However, there are some outliers. Enpass reported a low number of *Double*, *Reversed*, *Reverse Case*, and *Compromised Class4* passwords as weak. Conversely, KeePassXC reported all *LEET*, *Double*, *Reversed*, *Reverse Case*, *Compromised Class3*, and *Compromised Passphrases* as weak (Table 2). KeePassXC also reported many generated, non-compromised passwords as weak, suggesting that KeePassXC has stricter password strength requirements compared to other PMs.

[3]Reviews and ratings were gathered in Dec. 2023.

Table 2: The number of passwords marked as weak by each PM.

	LastPass	Norton	Roboform	Enpass	SafeInCloud	Dashlane	1Password	Sticky Password
LEET PWs (out of 735)	455 (61.9%)	665 (90.4%)	506 (68.8%)	447 (60.8%)	594 (80.8%)	458 (62.3%)	410 (55.7%)	365 (49.6%)
Double PWs (out of 197)	178 (90.3%)	183 (92.8%)	190 (96.4%)	13 (6.5%)	172 (87.3%)	178 (90.3%)	15 (7.6%)	133 (67.5%)
Reversed PWs (out of 197)	189 (95.9%)	192 (97.4%)	195 (98.9%)	8 (4%)	155 (97.4%)	190 (96.4%)	178 (90.3%)	133 (67.5%)
Reverse Case PWs (out of 197)	188 (95.4%)	192 (97.4%)	195 (98.9%)	8 (4%)	192 (97.4%)	183 (92.8%)	183 (92.8%)	177 (89.8%)
Passphrases (out of 75)	0	0	0	0	0	0	0	0
Basic PWs (out of 80)	0	19 (23.7%)	8 (10%)	20 (25.7%)	0	0	19 (23.7%)	39 (48.7%)
Class3 PWs (out of 80)	0	19 (23.7%)	1 (1.25%)	18 (22.5%)	0	0	0	3 (3.7%)
Class4 PWs (out of 80)	0	17 (21.25%)	0	15 (18.7%)	0	0	0	0
C-Class3 PWs (out of 112)	106 (94.6%)	112 (100%)	112 (100%)	112 (100%)	112 (100%)	108 (96.4%)	80 (71.4%)	83 (74.1%)
C-Class4 PWs (out of 197)	52 (26.3%)	108 (54.8%)	172 (87.3%)	14 (7.1%)	76 (38.5%)	50 (25.3%)	1 (.5%)	6 (3%)
C-Passphrase PWs (out of 40)	3 (7.5%)	0	38 (95%)	0	0	0	1 (2.5%)	0
	Keeper	**KeePassXC**	**Bitwarden**	**Google PM**	**Safari PM**	**Microsoft Edge PM**		
LEET PWs (out of 735)	161 (21.9%)	735 (100%)	665 (90.4%)	665 (90.4%)	595 (80.9%)	598 (81.3%)		
Double PWs (out of 197)	149 (75.6%)	197 (100%)	183 (92.8%)	183 (92.8%)	144 (73%)	140 (71%)		
Reversed PWs (out of 197)	175 (88.8%)	197 (100%)	192 (97.4%)	192 (97.4%)	148 (75.1%)	176 (89.3%)		
Reverse Case PWs (out of 197)	172 (87.3%)	197 (100%)	192 (97.4%)	192 (97.4%)	178 (90.3%)	176 (89.3%)		
Passphrases (out of 75)	1 (1.3%)	5 (6.6%)	0	0	0	0		
Basic PWs (out of 80)	19 (23.7%)	43 (53.2%)	19 (23.7%)	19 (23.7%)	1 (1.25%)	19 (23.7%)		
Class3 PWs (out of 80)	0	39 (48.7%)	19 (23.7%)	19 (23.7%)	5 (6.2%)	19 (23.7%)		
Class4 PWs (out of 80)	0	35 (43.7%)	17 (21.25%)	17 (21.25%)	2 (2.5%)	17 (21.25%)		
C-Class3 PWs (out of 112)	18 (16%)	112 (100%)	112 (100%)	112 (100%)	101 (90.1%)	88 (78.5%)		
C-Class4 PWs (out of 197)	0	193 (97.9%)	108 (54.8%)	108 (54.8%)	61 (30.9%)	67 (34%)		
C-Passphrase PWs (out of 40)	0	40 (100%)	0	0	0	0		

Table 3: The number of passwords marked as compromised. Note that some PMs do not report compromised passwords if there is not a username + password pair.

	Roboform	Enpass	SafeInCloud	1Password	Sticky Password	Keeper	KeePassXC	Bitwarden
LEET PWs (out of 735)	288 (39.1%)	288 (39.1%)	288 (39.1%)	288 (39.1%)	292 (39.7%)	348 (47.3%)	288 (39.1%)	288 (39.1%)
Double PWs (out of 197)	182 (92.3%)	182(92.3%)	182 (92.3%)	182 (92.3%)	183 (92.8%)	191 (96.9%)	182 (92.3%)	182 (92.3%)
Reversed PWs (out of 197)	189 (95.9%)	189 (95.9%)	189 (95.9%)	189 (95.9%)	183 (92.8%)	194 (98.4%)	189 (95.9%)	189 (95.9%)
Reverse Case PWs (out of 197)	189 (95.9%)	189 (95.9%)	189 (95.9%)	189 (95.9%)	190 (96.4)	196 (99.4%)	189 (95.9%)	189 (95.9%)
Basic PWs (out of 80)	0	0	0	0	0	19 (23.7%)	0	0
C-Class3 PWs (out of 112)	112 (100%)	112 (100%)	112 (100%)	112 (100%)	112 (100%)	112 (100%)	112 (100%)	112 (100%)
C-Class4 PWs (out of 197)	170 (86.2%)	170 (86.2%)	170 (86.2%)	170 (86.2%)	59 (29.9%)	105 (53.2%)	170 (86.2%)	170 (86.2%)
C-Passphrase PWs (out of 40)	38 (95%)	39 (97.5%)	39 (97.5%)	39 (97.5%)	38 (95%)	37 (92.5%)	39 (97.5%)	39 (97.5%)

It's likely that PMs using zxcvbn include additional factors when calculating password strength; zxcvbn itself reports a number between 0 (very guessable) to 4 (very unguessable) to reflect the strength of a password [52, 53]. PMs may interpret these numbers differently, which explains why the numbers each PM reports are inconsistent. However, we were only able to find additional information on how KeePassXC uses additional factors (such as password reuse) to factor into its strength report [18]. We were unable to find public documentation on how the other PMs generate its strength results, though Bitwarden's source code indicates that it does factor in whether a users' email address is present in the password [8].

Compromised Passwords. We saw a discrepancy between Roboform and other PMs that use HIBP, with Roboform only reporting 38 *Compromised Passphrases* while the others reported 39. It is unclear why Roboform failed to report the additional password, since entering the passphrase (*winner winner chicken dinner*) into HIBP shows that the password is indeed compromised. The fact that Roboform can fail to report passwords even when they are present in HIBP's database is worrisome. More testing should be performed to see how pervasive of a problem this is.

Except for *Compromised Passphrase* and *Compromised Class4* passwords, Keeper reported more compromised passwords than the other 13 PMs. Interestingly, Keeper reported 19 *Basic* (**not** compromised) credentials as breached; this is surprising considering those credentials were generated using a password generator, suggesting that Keeper has a larger database of breached credentials compared to others we tested. Sticky Password also reported more passwords in some corpuses compared to PMs that use HIBP, though it did underperform when detecting *Compromised Class4* credentials, only detecting 29.9% of passwords when PMs that use HIBP detected 86.2%. While HIBP is able to report a fair amount of compromised passwords, some databases were **better** at detecting more breached passwords. Even when passwords are detected by HIBP, sometimes they are not reported by a PM. PMs should look into using a combination of databases to find compromised credentials and ensure that their PMs accurately report this information.

6 DISCUSSION & FUTURE WORK

Discussion. In this paper, we looked at 14 password managers (PMs) and evaluated their checkup features. Using multiple different

password corpuses that contained weak, compromised, and strong credentials, we analyzed how many passwords each checkup tool marked as breached and weak. We found that most checkup tools do not accurately report breached passwords, even when using passwords found in public databases. Weak passwords that are variants of known compromised passwords were also under-reported.

We also found that most PMs' checkup features we tested use Have I Been Pwned (HIBP) to report breached passwords. Despite HIBP's popularity, some breached passwords still went unreported, even when a password was verifiably in HIBP's database (such as the case with Roboform). Additionally, as prior work has pointed out, many checkup tools with breach detection are locked behind premium subscriptions [3]. This is concerning, especially since PMs may not mark computationally "strong" breached passwords as weak, meaning users will have to pay a premium subscription to see if they actually need to change their passwords. Consequently, this could also incentivize users to keep using their breached passwords. We also saw inconsistencies in how PMs report weak passwords, despite many using zxcvbn to measure password strength. This may cause PMs to report passwords as strong, when they are actually not. Switching to a PM that reports fewer passwords as weak or compromised can cause additional uncertainty in users.

To prevent confusion, we recommend that PMs are more transparent in how they determine password strength, especially on how they weigh password characteristics (such as password composition or length). To better protect users' online accounts, PMs should mark all breached passwords as weak, even if they are strong by a PM's strength metric. We also recommend that PMs use multiple databases (chiefly public ones) to detect more breached passwords, especially if a user is paying for a premium subscription.

Future Work. Prior work has found that users find breach detection features useful, but also overwhelming, with many choosing to ignore them altogether [30]. Still, it is unclear how many PM users are aware of these tools, or how they use them. Is unawareness the result of these tools often being locked behind premium features [3], or can PMs do a better job at letting users know about these features? How do users adjust checkup tool settings, determine which passwords they need to change, and other remediation strategies? Are there more effective ways to nudge people to remediate weak/breached passwords? To fill this research gap, we will conduct a user study for the next iteration of this work.

ACKNOWLEDGMENTS

This material is based upon work supported by the US National Science Foundation under Grant No. 1845300.

REFERENCES

[1] 1Password. 2022. 1Password password quality algorithm. https://1password.community/discussion/130158/1password-password-quality-algorithm
[2] Nora Alkaldi and Karen Renaud. 2019. Encouraging Password Manager Adoption by Meeting Adopter Self-Determination Needs. (01 2019). https://doi.org/10.24251/HICSS.2019.582
[3] Sabrina Amft, Sandra Höltervennhoff, Nicolas Huaman, Yasemin Acar, and Sascha Fahl. 2023. "Would You Give the Same Priority to the Bank and a Game? I Do Not!" Exploring Credential Management Strategies and Obstacles during Password Manager Setup. (Aug. 2023), 171–190.
[4] Salvatore Aurigemma, Thomas Mattson, and Lori N. K. Leonard. 2017. So Much Promise, So Little Use: What is Stopping Home End-Users from Using Password Manager Applications?. In *Hawaii International Conference on System Sciences*. https://api.semanticscholar.org/CorpusID:11088576
[5] Ramakrishna Ayyagari, Jaejoo Lim, and Olger Hoxha. 2019. Why Do Not We Use Password Managers? A Study on the Intention to Use Password Managers. *Contemporary Management Research* 15, 4 (Dec. 2019), 227–245. https://doi.org/10.7903/cmr.19394 Number: 4.
[6] Sruti Bhagavatula, Lujo Bauer, and Apu Kapadia. 2020. (How) Do people change their passwords after a breach? http://arxiv.org/abs/2010.09853 arXiv:2010.09853 [cs].
[7] Bitwarden. Last Accessed: 2024-01-11. Vault Health Reports | Bitwarden Help Center. https://bitwarden.com/help/reports/
[8] Bitwarden. Last Accessed: 2024-03-12. *password-strength.service.spec.ts*. https://github.com/bitwarden/clients/blob/9e8f20a8731a16a06d165b4744ddeabc2ed5b84d/libs/common/src/tools/password-strength/password-strength.service.spec.ts#L10
[9] Crossword. Last Accessed: 2024-01-11. Arc. https://www.crosswordcybersecurity.com/arc
[10] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and XiaoFeng Wang. 2014. The Tangled Web of Password Reuse. In *Proceedings 2014 Network and Distributed System Security Symposium*. Internet Society, San Diego, CA. https://doi.org/10.14722/ndss.2014.23357
[11] Dashlane. Last Accessed: 2024-01-11. Manage the Password Health of your Starter, Team, or Business plan. https://support.dashlane.com/hc/en-us/articles/360016225300-Manage-the-Password-Health-of-your-Starter-Team-or-Business-plan
[12] Dashlane. Last Accessed: 2024-01-17. Security alerts and Dark Web Monitoring in Dashlane. https://support.dashlane.com/hc/en-us/articles/360000038180-Security-alerts-and-Dark-Web-Monitoring-in-Dashlane
[13] Enpass. Last Accessed: 2024-01-11. Miscellaneous — Enpass Security Whitepaper documentation. https://support.enpass.io/docs/security-whitepaper-enpass/miscellaneous.html#password-strength-estimation
[14] Michael Fagan, Yusuf Albayram, Mohammad Maifi Hasan Khan, and Ross Buck. 2017. An investigation into users' considerations towards using password managers. *Human-centric Computing and Information Sciences* 7, 1 (March 2017), 12. https://doi.org/10.1186/s13673-017-0093-6
[15] Dinei Florencio and Cormac Herley. 2007. A large-scale study of web password habits. In *Proceedings of the 16th international conference on World Wide Web - WWW '07*. ACM Press, Banff, Alberta, Canada, 657. https://doi.org/10.1145/1242572.1242661
[16] Yue Huang, Borke Obada-Obieh, and Konstantin Beznosov. 2022. Users' Perceptions of Chrome's Compromised Credential Notification. (Aug. 2022).
[17] Sowmya Karunakaran, Kurt Thomas, Elie Bursztein, and Oxana Comanescu. 2018. Data Breaches: User Comprehension, Expectations, and Concerns with Handling Exposed Data. (Aug. 2018).
[18] KeePassXC. Last Accessed: 2024-01-11. How KeePassXC's Password Health Check Feature Works – KeePassXC. https://keepassxc.org/blog/2020-08-15-keepassxc-password-healthcheck/
[19] LastPass. Last Accessed: 2024-01-17. Dark Web Monitoring & Alerts - LastPass. https://www.lastpass.com/features/dark-web-monitoring
[20] Yue Li, Haining Wang, and Kun Sun. 2017. Personal Information in Passwords and Its Security Implications. *IEEE Transactions on Information Forensics and Security* 12, 10 (Oct. 2017), 2320–2333. https://doi.org/10.1109/TIFS.2017.2705627
[21] Kali Linux. Last Accessed: 2024-01-05. rsmangler | Kali Linux Tools. https://www.kali.org/tools/rsmangler/
[22] Kali Linux. Last Accessed: 2024-01-05. Seclists Kali Linux Tools. https://www.kali.org/tools/seclists/
[23] Sanam Ghorbani Lyastani, Michael Schilling, Sascha Fahl, Michael Backes, and Sven Bugiel. 2018. Better managed than memorized? Studying the Impact of Managers on Password Strength and Reuse. (Aug. 2018).
[24] Raymond Maclean and Jacques Ophoff. 2018. Determining Key Factors that Lead to the Adoption of Password Managers. In *2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC)*. IEEE, Plaine Magnien, 1–7. https://doi.org/10.1109/ICONIC.2018.8601223
[25] Peter Mayer, Collins W Munyendo, Adam J Aviv, and Michelle L Mazurek. 2022. Why Users (Don't) Use Password Managers at a Large Educational Institution. (2022).
[26] Peter Mayer, Yixin Zou, Florian Schaub, and Adam J Aviv. 2021. "Now I'm a bit angry:" Individuals' Awareness, Perception, and Responses to Data Breaches that Affected Them. (2021).
[27] Microsoft. Last Accessed: 2024-01-17. Protect your online accounts using Password Monitor - Microsoft Support. https://support.microsoft.com/en-au/topic/protect-your-online-accounts-using-password-monitor-6f660aae-65aa-476c-871a-7fe2bcb0c4c1
[28] Daniel Miessler. 2020. SecLists/Passwords/2020-200mostusedpasswords.txt at master · danielmiessler/SecLists. https://github.com/danielmiessler/SecLists/blob/master/Passwords/2020-200-most-used-passwords.txt
[29] Collins W. Munyendo, Peter Mayer, and Adam J. Aviv. 2023. "I just stopped using one and started using the other": Motivations, Techniques, and Challenges When

Switching Password Managers. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Copenhagen Denmark, 3123–3137. https://doi.org/10.1145/3576915.3623150

[30] Sean Oesch, Scott Ruoti, James Simmons, and Anuj Gautam. 2022. "It Basically Started Using Me:" An Observational Study of Password Manager Usage. In *CHI Conference on Human Factors in Computing Systems*. ACM, New Orleans LA USA, 1–23. https://doi.org/10.1145/3491102.3517534

[31] Openwall. 2023. John the Ripper password cracker. https://www.openwall.com/john/

[32] Sarah Pearman, Jeremy Thomas, Pardis Emami Naeini, Hana Habib, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Serge Egelman, and Alain Forget. 2017. 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*. ACM, Dallas Texas USA, 295–310. https://doi.org/10.1145/3133956.3133973

[33] Sarah Pearman, Shikun Aerin Zhang, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2019. Why people (don't) use password managers effectively. (Aug. 2019).

[34] Denise Ranghetti Pilar, Antonio Jaeger, Carlos F. A. Gomes, and Lilian Milnitsky Stein. 2012. Passwords Usage and Human Memory Limitations: A Survey across Age and Educational Background. *PLoS ONE* 7, 12 (Dec. 2012), e51067. https://doi.org/10.1371/journal.pone.0051067

[35] Have I Been Pwned. Last Accessed: 2024-01-05. Have I Been Pwned: Pwned Passwords. https://haveibeenpwned.com/Passwords

[36] Hirak Ray, Flynn Wolf, Ravi Kuber, and Adam J. Aviv. 2021. Why Older Adults (Don't) Use Password Managers. (Aug. 2021), 73–90.

[37] Elissa M. Redmiles. 2019. "Should I Worry?" A Cross-Cultural Examination of Account Security Incident Response. http://arxiv.org/abs/1808.08177 arXiv:1808.08177 [cs].

[38] Roboform. Last Accessed: 2024-01-11. How Secure Is My Password? https://www.roboform.com/how-secure-is-my-password

[39] SafeInCloud. 2022. Compromised passwords. //safeincloud.ladesk.com/767071-Compromised-passwords

[40] SafeInCloud. Last Accessed: 2024-01-11. What is the crack time for password. https://safeincloud.ladesk.com/709198-What-is-the-crack-time-for-password

[41] Sunyoung Seiler-Hwang, Patricia Arias-Cabarcos, Andrés Marín, Florina Almenares, Daniel Díaz-Sánchez, and Christian Becker. 2019. "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*. ACM, London United Kingdom, 1937–1953. https://doi.org/10.1145/3319535.3354192

[42] Elizabeth Stobert and Robert Biddle. 2016. Expert Password Management. In *Technology and Practice of Passwords*, Frank Stajano, Stig F. Mjølsnes, Graeme Jenkinson, and Per Thorsheim (Eds.). Vol. 9551. Springer International Publishing, Cham, 3–20. https://doi.org/10.1007/978-3-319-29938-9-1 Series Title: Lecture Notes in Computer Science.

[43] Joshua Tan, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2020. 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*. ACM, Virtual Event USA, 1407–1426. https://doi.org/10.1145/3372297.3417882

[44] Kurt Thomas, Jennifer Pullman, Kevin Yeo, Ananth Raghunathan, Patrick Gage Kelley, Luca Invernizzi, Borbala Benko, Tadek Pietraszek, Sarvar Patel, Dan Boneh, and Elie Bursztein. 2019. Protecting accounts from credential stuffing with password breach alerting. (2019).

[45] Blase Ur, Jonathan Bees, Sean M. Segreti, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2016. Do Users' Perceptions of Password Security Match Reality?. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, San Jose California USA, 3748–3760. https://doi.org/10.1145/2858036.2858546

[46] Blase Ur, Patrick Gage Kelley, Saranga Komanduri, Joel Lee, Michael Maass, Michelle L Mazurek, Timothy Passaro, Richard Shay, Timothy Vidas, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2012. How Does Your Password Measure Up? The Effect of Strength Meters on Password Creation. (2012).

[47] Blase Ur, Fumiko Noma, Jonathan Bees, Sean M Segreti, Richard Shay, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2015. "I Added '!' at the End to Make It Secure": Observing Password Creation in the Lab. (July 2015).

[48] Rafael Veras, Christopher Collins, and Julie Thorpe. 2014. On the Semantic Patterns of Passwords and their Security Impact. In *Proceedings 2014 Network and Distributed System Security Symposium*. Internet Society, San Diego, CA. https://doi.org/10.14722/ndss.2014.23103

[49] Rafael Veras, Julie Thorpe, and Christopher Collins. 2012. Visualizing semantics in passwords: the role of dates. In *Proceedings of the Ninth International Symposium on Visualization for Cyber Security*. ACM, Seattle Washington USA, 88–95. https://doi.org/10.1145/2379690.2379702

[50] Rick Wash and Emilee Rader. 2021. Prioritizing security over usability: Strategies for how people choose passwords. *Journal of Cybersecurity* 7, 1 (Feb. 2021), tyab012. https://doi.org/10.1093/cybsec/tyab012

[51] Rick Wash, Emilee Rader, and Ruthie Berman. 2016. Understanding Password Choices: How Frequently Entered Passwords are Re-used Across Websites. (June 2016).

[52] Daniel Lowe Wheeler. 2016. zxcvbn: Low-Budget Password Strength Estimation. *25th USENIX Security Symposium* (2016), 157–173.

[53] Daniel Lowe Wheeler. 2024. dropbox/zxcvbn. https://github.com/dropbox/zxcvbn original-date: 2012-02-28T03:25:54Z.

[54] Weining Yang, Ninghui Li, Omar Chowdhury, Aiping Xiong, and Robert W. Proctor. 2016. An Empirical Study of Mnemonic Sentence-based Password Generation Strategies. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Vienna Austria, 1216–1229. https://doi.org/10.1145/2976749.2978346

[55] Yixin Zou, Shawn Danino, Kaiwen Sun, and Florian Schaub. 2019. You 'Might' Be Affected: An Empirical Analysis of Readability and Usability Issues in Data Breach Notifications. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–14. https://doi.org/10.1145/3290605.3300424