An Empirical Measurement Study of Authentication Mechanisms in Cloud-Based Password Managers

Yanduo Fu, Qingxuan Wang✉, Ding Wang

College of Cryptology and Cyber Science, Nankai University, Tianjin 300350, China;
wangqingxuan@nankai.edu.cn
Key Laboratory of Data and Intelligent System Security (NKU), Ministry of Education, Tianjin 300350, China
Tianjin Key Laboratory of Network and Data Security Technology, Nankai University, Tianjin 300350, China

Abstract. A password manager (PM) is a widely recommended security tool that helps users manage online credentials. To support vault access from multiple devices, cloud-based PMs synchronize encrypted vaults and provide publicly reachable authentication entry points through websites or browser extensions. Remote attackers can also reach these entry points and submit candidate account identifiers and passwords.
In this paper, we conduct a black-box measurement of 15 representative cloud-based PMs across 20 authentication entry points, including separate account-password and master/vault-password entry points for five PMs. We probe account existence at the account-login entry point of each PM and find that six of 15 PMs expose a stable account-existence signal. Across the 20 entry points, we also compare responses to correct and incorrect credentials and record responses to up to 15 incorrect-password submissions spaced at least five seconds apart. Fifteen entry points expose a stable credential-validity signal. Among the remaining five, two require victim-controlled verification before password submission, one returns the same email-verification response to correct and incorrect credentials, and two require a master password together with an unavailable additional key, preventing the master password from being evaluated independently. For repeated incorrect-password submissions, 12 entry points show at least one visible intervention, six show none within our 15-attempt budget, and two are unreachable as victim-controlled verification precedes credential submission. Across the 12 affected entry points, five present a CAPTCHA, five show a temporary lockout, two reset the login interface, and one applies rate limiting; the NordPass account-password entry point shows both a CAPTCHA and rate limiting. These results show that PM login responses may reveal whether an account exists or a submitted credential is correct. Additionally, in our test flows, defenses (e.g., CAPTCHAs or UI resets) may add friction but do not always prevent further submissions.

Keywords: Password Manager · Authentication · Login Policy

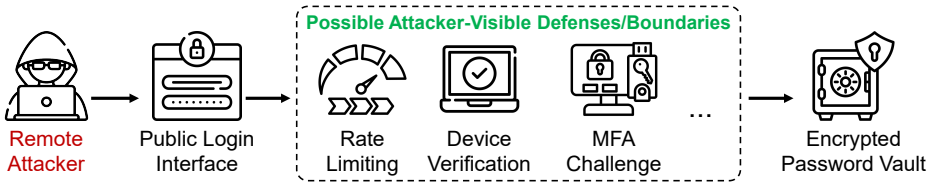


Fig. 1: A remote attacker can reach a cloud-based PM's public login interface through a website or browser extension. Depending on the product and authentication path, the attacker may encounter defenses triggered by repeated login failures or additional verification boundaries before vault access.

1 Introduction

Passwords remain the dominant authentication mechanism on the modern web [2, 44]. Despite the growing adoption of alternative authentication technologies, users still rely on passwords for numerous online accounts [46]. Managing this large set of credentials imposes a substantial cognitive burden and contributes to insecure practices such as weak passwords and password reuse across services [14, 47]. Such practices continue to pose substantial security risks [2, 35].

Password managers (PMs) have become an important security tool for alleviating this management burden [10, 18]. They can generate strong and unique passwords, store and autofill credentials, synchronize encrypted vaults across devices, and provide password-health checks, thereby reducing risks associated with weak passwords and password reuse [30]. Consequently, PMs are recommended by major standards bodies [15, 44] and security experts [8, 33].

Cloud synchronization allows users to access and unlock their vaults from multiple devices. This convenience also exposes publicly accessible authentication workflows, which means remote attackers can initiate these workflows through a PM's website or browser extension.

PM authentication mechanisms vary across products. Some PMs (e.g., LastPass, Bitwarden) use one master/vault password for login and vault unlock. Others separate an account password used for login from a master password used for vault unlock, such as NordPass [36] and Norton [37], or require a master password with an additional key, such as 1Password [5] and MultiPassword [4]. These additional keys are long, product-specific secrets generated by the PM. Victim-controlled verification steps (e.g., email verification, device approval) may also appear at different stages of these workflows. We refer to each stage at which a candidate credential can be submitted as an *authentication entry point*. A PM that separates login and vault-unlock credentials therefore has two such entry points. Reaching an entry point does not imply vault access, but the observable workflow may reveal whether an account exists, whether a candidate credential is correct, or how the service responds to repeated incorrect-password submissions.

Black-box differential probing has been applied to general website login policies [7, 22], but cloud-based PMs combine account login, vault unlock, victim-controlled verification, and product-specific keys. Prior PM studies have not jointly characterized account-existence signals, credential-validity signals, and

responses to repeated incorrect-password submissions in cloud-based PMs. We therefore analyze each authentication entry point under a black-box framework.

Threat Model. We consider a *fresh-device remote attacker* who has never authenticated as the victim and holds no trusted device or session (see Fig. 1). The attacker can access publicly reachable website or browser-extension authentication entry points, submit candidate identifiers (e.g., email addresses) and credentials, and observe network responses, UI feedback, and workflow transitions. The attacker does not possess victim-controlled email or SMS codes, MFA approval, trusted-device status, and required product-specific additional keys. A victim-controlled step before vault access may therefore form a *stopping boundary*. Human-solvable CAPTCHAs remain in scope and are completed manually during measurement, but victim-controlled verification is never bypassed.

Under this threat model, we conduct a black-box measurement study of 15 cloud-based PMs. These 15 PMs involve 20 authentication entry points as five PMs require an account password for login and a separate master/vault password for vault unlock. Rather than attempting to reconstruct undocumented server-side policies, we focus on externally observable authentication behavior. Specifically, we investigate the following research questions (RQs):

RQ1: Can a remote attacker distinguish registered from unregistered accounts through observable signals in cloud-based PM authentication workflows?

RQ2: Can an attacker determine whether a candidate credential is valid at each authentication entry point, and how far can the attacker progress afterward?

RQ3: What responses arise during a bounded sequence of incorrect-password submissions, and how do they affect a correct-credential submission?

Contributions. In summary, our work makes the following three contributions.

(1) We adapt black-box differential probing to authentication workflows in cloud-based PMs. The framework measures account-existence signals, candidate-credential validity signals, and attacker-visible responses during up to 15 incorrect-password submissions. For the five PMs that use an account password followed by a separate master/vault password, we test the two password entry points separately. We also record whether victim-controlled verification prevents credential submission or vault access, and whether an unavailable additional key prevents master-password validity from being isolated.

(2) We apply this framework to 15 cloud-based PMs with 20 authentication entry points. Account-existence probing at the account-login entry point of each PM shows that six of 15 PMs expose stable signals. Credential-validity probing across 20 entry points shows that 15 expose stable signals, three are blocked by victim-controlled verification, and two use compound credentials (i.e., master passwords and additional keys) that prevent master-password validity from being isolated. During repeated incorrect-password submissions, 12 reachable entry points show at least one visible intervention (e.g., temporary lockout, CAPTCHA), while six show none within our 15-attempt budget. The remaining two entry points are unreachable under our threat model as victim-controlled verification precedes credential submission.

(3) We analyze how additional verification and failed-login defenses affect online probing. A victim-controlled verification placed before the credential submission stage can prevent validity testing, whereas verification reached only after a correct credential may block vault access while still revealing credential validity. Similarly, CAPTCHA and UI reset allow probing to resume in our tested flows, whereas the observed temporary lockouts can block subsequent attempts, including correct credentials. These findings suggest opportunities to improve the security of PM authentication by reducing feedback that differs between correct and incorrect passwords before additional verification. Defenses against repeated incorrect-password submissions should also consider their usability effects on legitimate users.

Ethical Considerations. Our measurements target publicly accessible authentication entry points using researcher-controlled accounts. We never attempt to access accounts belonging to other users. Following prior work [7], we cap each repeated-failure probe at 15 incorrect-password submissions and space the submissions at least five seconds apart rather than using the one-second interval in [7]. When a blocking intervention appears, we stop submitting incorrect credentials and perform only the correct-credential and recovery checks defined in our methodology. We complete human-solvable CAPTCHAs manually because they fall within our attacker model, but never bypass victim-controlled email or SMS verification, MFA or device approval, trusted-device requirements, or product-specific additional keys unavailable to the default attacker. These limits reduce measurement load and the risk of escalating account or service state.

2 Background and Related Work

We first present the background of password managers (PMs). We then present related studies on measuring PM security and on website login policies.

2.1 Background of Password Managers

Our work focuses on online authentication behavior in cloud-based password managers (PMs). PMs can be categorized by integration model into built-in browser, built-in OS, or standalone PMs [32, 39, 40]. Authentication for synchronized built-in PMs is primarily mediated by the corresponding platform account, while local-only PMs expose no PM-operated remote authentication entry point to which our probes apply. We therefore study standalone PMs with independent cloud accounts, provider-operated cloud synchronization, and publicly reachable website or browser-extension login interfaces.

2.2 Security Studies of Password Managers

Prior work has examined the security of password managers (PMs) and has identified weaknesses in vault protection [48], metadata exposure [38], and credential sharing [27], as well as sensitive data exposure through client-side

mechanisms such as inter-process communication [13] and process memory [6]. Browser-focused work shows that malicious web content may exploit autofill behavior to expose stored credentials or sensitive data [19, 28]. Recent studies have examined phishing against PM users [9] and weaknesses in end-to-end encryption or zero-knowledge vault designs [16, 42]. These studies have primarily examined PM security at the storage, client-side, browser interaction, and cryptographic layers. Our study instead focuses on the provider-operated authentication behavior of cloud-based standalone PMs.

2.3 Authentication Mechanism Measurement in Websites

Authentication policies of websites have been extensively studied through black-box measurement. Hasegawa et al. [22] examined account-existence leakage across login, password recovery, and account-creation workflows in 87 websites, finding 98.9% of 87 websites vulnerable to this attack. Al Roomi and Li [7] developed a large-scale automated framework for measuring website login policies. Their framework compares login failures for registered and unregistered identifiers to detect user-enumeration signals, and probes rate limiting via 15 consecutive incorrect-password attempts in the same browser session followed by a correct-credential check. Across 31.2K evaluated domains, they found 19% exhibiting login-message-based user enumeration, while rate limiting appeared on only approximately 25-30% of the 18.0K domains evaluated for that policy.

Repeated-login defenses have also been examined in greater depth. Golla et al. [21] manually probed 12 online services with up to 25 failed login attempts and a subsequent correct-credential check, finding 11 of 12 services showing rate-limiting responses such as CAPTCHA, temporary blocking, and account lockout. Lu et al. [29] developed a more extensive black-box model that varied login intervals and client state to infer lockout thresholds and counting mechanisms, showing that rate-limiting behavior can depend on the targeted account, client IP, or browser state. Among 182 examined websites, 72% allowed frequent unsuccessful login attempts without lockout or throttling.

Our work builds on this black-box measurement perspective but targets the authentication workflows exposed by cloud-based PMs, which may expose distinct credential-submission entry points for account login and vault unlock with different credentials. Product-specific additional keys and victim-controlled verification further affect which entry points a remote attacker can reach. We therefore analyze each entry point separately and characterize externally observable behavior rather than reconstruct undocumented server-side policies.

3 Methodology

We first describe how we select cloud-based standalone password managers (PMs). Then, we describe our measurement setup, including test accounts, browser environment, and exception handling. Finally, we present our black-box methodology for measuring their authentication mechanisms.

Table 1: Basic Information of Our Selected 15 Password Managers (PMs)*.

Name@Version	Users	Credential†	Name@Version	Users	Credential†
LastPass@4.154.2	8M	MP	Avira PM@2.21.0.5015	8M	AP+MP
NordPass@7.9.3	7M	AP+MP	1Password@8.12.30.21	6M	MP+AK
Bitwarden@2026.7.0	6M	MP	Norton PM@8.3.1.1495	5M	AP+MP
Kaspersky PM@25.2.6.1	2M	AP+MP	Keeper PM@18.0.0	2M	MP
Dashlane@6.2631.0	1M	MP	MultiPassword@0.99.32	1M	MP+AK
Proton Pass@1.38.2	1M	MP	True Key@5.0.0.288	600K	MP
RoboForm@9.9.9.0	600K	MP	DualSafe@1.4.37	200K	MP
Zoho Vault@6.6.0	100K	AP+MP			

∗ "Users" is the Chrome extension user count reported by ChromeStats [3] on May 20, 2026 for PM candidate selection; 1K=1,000; 1M=1,000K; "Version" is the extension version reported by ChromeStats on August 1, 2026. Final measurements are conducted from August 1-8, 2026.
† Long-term credential structure: "MP"=master password only; "AP+MP"=account password followed by master/vault password; "MP+AK"=master password plus an additional key (e.g., the 1Password Secret Key [5] and the MultiPassword Private Key [4]). Additional verification or challenge steps (e.g., CAPTCHA and device verification) are reported in Tab. 2.

3.1 Selecting Password Managers

We select cloud-based standalone password managers (PMs) that provide public login interfaces as our measurement targets. This scope follows our threat model (see Sec. 1): A remote attacker without prior authenticated client state can reach the public login interface, submit online authentication attempts, and observe the responses without access to the victim's device or local vault.

To construct the candidate set, we collect PM candidates from recent academic studies (including [16, 17, 19, 24, 38]) and non-academic web sources, including Security.org [18], PasswordManager.com [25], and selected Reddit threads (see App. A for more details). Additionally, following prior work [19], we also search ChromeStats [3] using terms such as "password", "password-protected", "password vault", and "password manager". After merging duplicate products and renamed products, we obtain 85 unique PM candidates. We record product availability and Chrome extension user count according to ChromeStats [3] on May 20, 2026.

We first verify whether each candidate is available on the collection date (see App. A). We exclude five products that have been discontinued or whose official clients are no longer obtainable, leaving 80 candidates. Among them, 64 PMs provide a Chrome extension, while 16 PMs do not provide Chrome extensions.

For the 64 candidates with Chrome extensions, we use the extension user count reported by ChromeStats [3] as an indicator of popularity. We use this indicator because browser extensions are an important means by which standalone PMs interact with websites, and Chrome has the largest share of the desktop browser market [43]. ChromeStats provides a consistent source for comparing Chrome extension user counts and has been used in prior measurement studies [19, 23]. To focus on products with a substantial user base, we retain 22 products whose official Chrome extensions have at least 100,000 users. For the 16 candidates that do not provide Chrome extensions and thus lack a directly comparable extension-based user count, we retain a product for further inspection if it either appears in a prior academic PM study or appears in at least two non-academic web sources. Five products meet this condition: AuthPass, Gokey, KeePass2Android,

KeePassium, and Vaultwarden, and the other 11 products are not retained. This step leaves 27 candidates for further manual inspection.

We then inspect the product scope and login process of each remaining candidate. To be included, a product must provide a cloud account and a publicly accessible account-registration process that allows us to create author-owned test accounts. Its public account-login process must use an account or master password. Its login process must provide at least one public authentication entry point to which our measurements can be applied. We exclude six local or file-based PMs that do not provide cloud accounts, four self-hosted or organization-specific products, and two PMs without a comparable public password-login process. Tab. 3 lists these 12 products and their exclusion reasons. After this screening, we retain 15 cloud-based standalone PMs for measurement in Tab. 1.

3.2 Measurement Setup

Account preparation. We create author-owned test accounts for each measured PM. We also prepare account identifiers (e.g., email addresses) that are controlled by the authors but not registered with the PM. The PM accounts are used only for this study, and their vaults contain no real user credentials or sensitive data. For PMs that require a compound credential (e.g., an additional key alongside a master password, as in 1Password [5] and MultiPassword [4]), we construct syntactically valid but incorrect additional keys for the submitted account identifier according to the product's input requirements, since our threat model assumes the attacker does not possess the real key but can construct one.
Probe Execution and Observation. Our work observes the feedback obtainable by remote attackers through publicly accessible authentication entry points. For login flows that can be reliably automated through authentication entry points, we use Playwright to launch a new browser context and execute the probes. We record UI feedback, flow transitions, screenshots, and network responses. A new browser context contains no prior cookies, local storage, or authenticated browser state and serves as our operational approximation of a fresh-client login state. This procedure removes client-side browser state but cannot reset server-side state, such as IP reputation, recent failed attempts, device trust, or automation-detection decisions.

When automated probing encounters a CAPTCHA, we switch to manual interaction rather than using automated solving methods. For some PMs (e.g., NordPass), automated input repeatedly triggers bot-detection challenges, making automated probing impractical under our procedure. We execute the corresponding flow manually. We also execute browser-extension login flows manually rather than building product-specific extension automation. For manual probes, we use a clean browser environment, such as incognito mode or a fresh virtual machine, and use DevTools, Chrome Net-Export, and screen recording to preserve observable workflow and network evidence. We do not modify browser fingerprints, use stealth plugins, or attempt to bypass victim-controlled verification mechanisms such as email verification or device approval. We report the probing method (automated or manual) for each PM in Tab. 2.

Throughout the paper, we use CAPTCHA as a generic term for CAPTCHA-based human-verification challenges, including reCAPTCHA and hCAPTCHA. Such challenges fall within our attacker model because they can be completed without access to victim-controlled factors. When encountered, we complete them manually and continue the probe. For the account-existence and credential-validity measurements, differences solely in CAPTCHA presentation or triggering are *not* used as evidence for classification, because such triggering may depend on auxiliary automation-detection or risk-control state. We complete the challenge and base the classification on the subsequent *UI feedback, workflow transitions, network responses, or stopping boundaries*. For repeated-failure probes, any CAPTCHA encountered during the bounded sequence is recorded as a visible intervention. This classification records its presence and does not imply that preceding failures trigger it. We complete it manually when possible to determine whether authentication attempts can resume. Challenges requiring access to victim-controlled out-of-band factors (e.g., email verification or device approval) are not bypassed and are recorded as blocking boundaries.

For each probe, we retain the necessary UI and network evidence. Before inclusion in the paper or shared evidence package, we redact account identifiers, passwords, authentication tokens, and other sensitive fields, following [7].

Black-box Probe Design, Repetition and Decision Rules. We measure each PM authentication flow using controlled black-box probes without attempting to reconstruct server-side authentication logic. For differential probes, we vary one security-relevant condition while holding the remaining controlled inputs fixed. We first conduct exploratory runs to map each PM's authentication workflow and identify the applicable measurement entry points. These runs are used for workflow characterization rather than as the sole basis for final classifications. For each account-existence or credential-validity probe, the final classification is based on three paired confirmation rounds, each starting from a clean client-side state. A signal is reported as *stable* within the measurement period only when each input across three paired rounds produces the same observable outcome and reaches the same login step. Observable outcomes include UI feedback, network responses, and workflow transitions. When a victim-controlled verification boundary prevents the target credential stage from being reached, we instead repeat the gate-reachability test in three clean contexts and classify the case according to the login step at which all three tests stop (e.g., email verification). If the three rounds disagree in observable outcome, reached login step, or classification, we perform one additional adjudication round under the same controlled setup. If the additional round confirms that the variation is associated with an identifiable condition, we report the result as *state-dependent*. Persistent disagreement without an identifiable cause is classified as inconclusive.

Repeated-failure probes follow separate repetition rules because incorrect-password submissions may alter server-side account, device, or risk state. Cases with no visible intervention or non-blocking interventions (e.g., CAPTCHA and UI reset) are repeated at least three times after the observable pre-probe state has been restored. Lockout-type interventions are confirmed in at least two

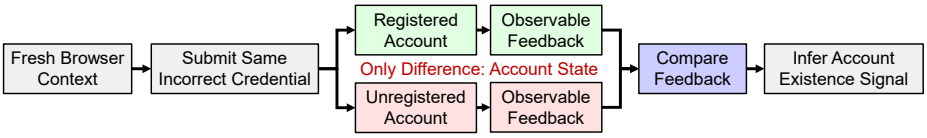


Fig. 2: Comparing registered and unregistered accounts under the same incorrect credentials to infer whether the PM exposes the existence of an account.

probes when the same observable pre-probe state can be restored without risking further account restriction. We consider the observable pre-probe state restored when the entry point returns to the same initial workflow with no residual intervention introduced by the preceding probe. Results of repeated-failure probes are interpreted within a budget of 15 incorrect-password submissions: no intervention means that no visible intervention is observed within this budget, rather than that no defense exists beyond this budget.

3.3 Account-Existence Probing Measurement

We measure whether an attacker can determine if an account identifier (i.e., an email address) is registered with the target PM. For each PM, we compare one registered and one author-controlled unregistered identifier under the same incorrect login credentials (Fig. 2). We probe only the account-login entry point of the five two-layer PMs because the identifier is already established before vault unlock. Under the repetition and decision rules above, a stable difference attributable only to account state is classified as an exposed account-existence signal. Otherwise, no signal is observed under our probe.

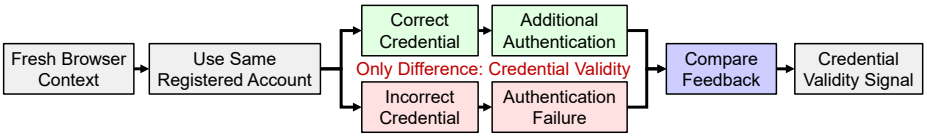


Fig. 3: Comparing correct/incorrect credentials for the same registered account.

3.4 Credential-Validity Probing Measurement

We measure whether an attacker can determine whether a candidate credential is valid before vault access. At each credential submission stage, we compare correct and incorrect candidate credentials for the same registered account while holding other inputs fixed (Fig. 3). Under the repetition and decision rules described above, a stable difference before vault access is classified as an exposed validity signal, including when only the correct credential reaches later verification (e.g., email verification or device approval). We report *Blocked* when victim-controlled verification either prevents the target credential from being submitted or makes correct and incorrect candidates observationally indistinguishable, and *Compound* when an unavailable additional key prevents master-password validity from being isolated. Otherwise, no signal is observed under our probe.

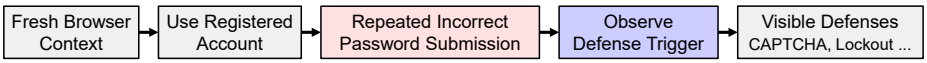


Fig. 4: Observing responses during repeated incorrect-password submissions.

3.5 Bounded Repeated-Failure Probe

We characterize the attacker visible responses that arise during repeated incorrect-password submissions (see Fig. 4). Following prior work [7], for each applicable entry point, we use a budget of 15 incorrect-password submissions to balance the likelihood of observing failed-login defenses against the measurement load imposed on the service. Unlike one-second intervals in prior work [7], we wait at least five seconds between submissions to reduce burstiness and the likelihood that our measurement itself unnecessarily escalates account or risk state.

We report CAPTCHA whenever it is observed during the bounded probe. We complete the challenge manually and continue the probe when further authentication attempts are permitted. Other non-blocking interventions are likewise recorded, and we continue the probe when further authentication attempts remain possible. If an intervention prevents further attempts, such as a temporary lockout state, we stop submitting incorrect credentials and use a correct credential to verify whether authentication is indeed blocked. When the blocking state later disappears, we check whether the authentication flow has returned to its observable pre-probe state by submitting the correct credential. If a victim-controlled out-of-band verification step (e.g., an email OTP, SMS code, or device approval) prevents further probing, we record it as a blocking boundary and terminate the sequence. If no blocking intervention occurs, we continue until the 15-attempt budget is reached. We then submit the correct credential once to determine whether it is still accepted.

No intervention means only that no visible intervention is observed within our 15-attempt budget. This bounded probe does not reconstruct exact triggering thresholds, determine whether an intervention is bound to an account, IP address, device, browser session, or another risk signal, or characterize behavior beyond the 15-attempt budget or under high-rate or distributed guessing.

4 Evaluation and Analysis

We now present the results of our measurement and conduct a security analysis.

4.1 RQ1: Results of Account-Existence Probing

Column "Account Existence" in Tab. 2 summarizes the account-existence probing results for 15 selected password managers (PMs). At the account-login entry point, six of 15 PMs expose stable account-existence signals under our probes. We observe three kinds of signals. LastPass allows registered identifiers to reach email verification, whereas unregistered identifiers cannot. DualSafe, True Key, and Zoho Vault display distinguishable on-page messages. 1Password and MultiPassword expose network-level differences. For 1Password, the webpage shows

Table 2: Account-Existence and Credential-Validity Signals, Repeated-Failure Responses, and Stopping Boundaries Across the Tested Entry Points*.

Password Manager	Entry Point	Account Existence†	Credential Validity†	Failure‡ Response(≤15)	Stopping Boundary◇	Process Mode⋆
LastPass	Website	✓	Blocked	Lockout	Email	A
Avira-AP	Website	×	✓	CAPTCHA	None	A+M
Avira-MP	Website	-	✓	Not Observed	None	M
NordPass-AP	Website	×	✓	CAPTCHA + Rate Limit	None	M
NordPass-MP	Extension	-	✓	UI Reset	None	M
1Password	Website	✓	Compound	Not Observed	Compound	A
Bitwarden	Website	×	✓	Not Observed	Email	A
Norton-AP	Website	×	✓	CAPTCHA	None	M
Norton-MP	Extension	-	✓	Lockout	None	M
Kaspersky-AP	Website	×	✓	CAPTCHA	None	A+M
Kaspersky-MP	Extension	-	✓	Lockout	None	M
Keeper	Website	×	Blocked	-	Email	A
Dashlane	Extension	×	Blocked	-	Email	M
MultiPassword	Website	✓	Compound	Not Observed	Compound	A
Proton Pass	Website	×	✓	Not Observed	None	A
True Key	Extension	✓	✓	Not Observed	Email	M
RoboForm	Extension	×	✓	Lockout	Email	M
DualSafe	Extension	✓	✓	Lockout	None	M
Zoho Vault-AP	Website	✓	✓	CAPTCHA	Email^{s}	A+M
Zoho Vault-MP	Extension	-	✓	UI Reset	None	M

∗ AP and MP denote account and master/vault password entry points.
† ✓: stable distinguishable signal; ×: no stable signal observed; −: not applicable because account existence is tested only at the account-login entry point. For Credential Validity, **Blocked**: victim-controlled verification prevents credential validity being exposed; **Compound**: an unavailable additional key prevents validity from being isolated.
‡ Responses observed during the bounded probe. **CAPTCHA**: a bot-detection or human-verification challenge; **Rate Limit**: a temporary request restriction whose activation or effect on a subsequent correct credential may vary across observed paths; **Lockout**: an explicit sustained locked or suspended state that blocks subsequent submissions, including a correct credential; **UI Reset**: a client-side reset after which attempts can resume; **Not Observed**: no visible intervention within the 15-attempt budget; **-**: credential submission is unreachable.
◇ Earliest additional stop before vault access: **Email**: victim-controlled email verification; **Compound**: an unavailable additional key; **None**: no stopping boundary observed. ^{s} denotes state-dependent email verification documented for unusual Zoho sign-ins [31].
⋆ A=automated; M=manual; A+M=automated with manual interaction.

"We weren't able to sign in to your account." for both account identifiers, while the response for registered accounts contains the `userUuid` and `signInAddress` fields, which are absent for unregistered identifiers (see Figs. 5(a) and 5(b)). MultiPassword similarly returns password-authentication parameters for the registered account and "User not found" for the unregistered one.

The other nine PMs expose no stable account-existence signals. Eight reach the same categorical outcome for both account states. Any auxiliary differences are input-derived, unstable, or CAPTCHA-only and therefore do not satisfy our classification rule. Bitwarden is discussed separately below.

For Bitwarden, an initial test comparing one registered and one unregistered account shows differing KDF parameters and master-password salt, appearing to reveal account existence. However, testing additional unregistered identifiers reveals varying plausible configurations that overlap with values returned for the registered account, indicating the difference is not specific to account state.

We further inspect Bitwarden's source code [11,12] to explain this behavior. For a registered account, the pre-login endpoint returns the account's actual KDF information and master-password salt. For an unregistered account, Bitwarden applies an explicit enumeration-protection mechanism that selects a synthetic KDF configuration from a pool of plausible configurations and independently sets the salt to either the normalized email address or `null`. The client implementation further shows that the master-password salt for the conventional email-based flow is derived by lowercasing and trimming the user's email address. Consequently, the synthetic responses for unregistered accounts overlap with values that can legitimately occur for registered accounts. Under our tested default configuration, the observed KDF parameters fall within the synthetic candidate pool and the salt does not provide a stable distinction between registered and unregistered accounts. We therefore observe no stable account-existence signal in Bitwarden's pre-login response. Accounts using cryptographic configurations outside the synthetic response space could potentially produce distinguishable responses, which we leave to future work.

4.2 RQ2: Results of Credential-Validity Probing

Columns "Credential Validity" and "Stopping Boundary" in Tab. 2 report whether a remote attacker can distinguish correct from incorrect candidate credentials and the earliest additional stopping boundary observed before vault access. Fifteen of 20 authentication entry points across 10 of 15 PMs expose a stable credential-validity signal: a correct credential reaches a later verification, device confirmation, or vault-related step, whereas an incorrect credential fails earlier (see the Bitwarden examples in Figs. 6(a) and 6(b)).

DualSafe, Zoho Vault, and True Key also expose account-existence signals, whereas probing the other seven PMs requires a registered account identifier obtained through another channel. For five two-layer PMs, passing the account-login stage additionally exposes a stable validity signal at the vault-unlock entry point. In Bitwarden, RoboForm, and True Key, a correct candidate reaches a victim-controlled device, email, or MFA verification, whereas an incorrect candidate fails earlier. The Zoho Vault account-password entry point exhibits this pattern in some tested states, while in other states a correct account password proceeds without an additional verification step. These verification steps can prevent vault access without hiding credential validity.

The other five entry points do not expose an independently interpretable validity signal. Dashlane and Keeper place victim-controlled verification (e.g., email verification and device approval) before the target credential submission stage (see Figs. 6(d), 6(e)). LastPass accepts credential submissions but returns the same email-verification response for correct and incorrect credentials (see Fig. 6(c)). We classify these three cases as *Blocked*. 1Password and MultiPassword use compound credentials consisting of a master password and an additional key. Because the correct additional key is not assumed available to the attacker, a failed authentication attempt does not reveal whether the master password or

the additional key is incorrect. We therefore classify these cases as *Compound*, since master-password validity cannot be isolated under our threat model.

Across these cases, credential validity is not exposed when victim-controlled verification (e.g., email verification or device approval) precedes password submission, when correct and incorrect credentials lead to the same subsequent verification step (e.g., email verification), or when a master password must be submitted with an unavailable additional key and therefore cannot be evaluated independently. When only a correct credential reaches a later step, the transition reveals that the credential is correct even though vault access remains blocked.

4.3 RQ3: Results of Bounded Repeated-Failure Probing

Column "Failure Response($\leq$15)" in Tab. 2 summarizes the attacker-visible responses to repeated incorrect-password submissions within our measurement budget. Across the 20 entry points, 12 show at least one visible intervention, six show no visible intervention within our 15-attempt budget, and two are unreachable because victim-controlled verification precedes credential submission.

The six *Not Observed* cases are Bitwarden, Avira master password, Proton Pass, 1Password, MultiPassword, and True Key. In each case, the probe reaches the 15-attempt budget without a visible intervention, and the subsequent correct-credential check does not reveal a failure-induced block. This bounded result does not imply that no defense exists beyond the tested budget or under other server-side risk states. For 1Password and MultiPassword, repeated submissions remain possible in the tested sequence. Keeper and Dashlane stop the default attacker before the credential submission stage through victim-controlled verification.

The 12 entry points with a visible intervention yield 13 non-exclusive mechanism observations: five CAPTCHAs, five lockouts, one rate limit, and two UI resets. The total exceeds the number of affected entry points because NordPass account-password is the only entry point at which we observe both a CAPTCHA and rate limiting. CAPTCHAs appear at the account-password entry points of Avira, NordPass, Norton, Kaspersky, and Zoho Vault during the bounded probes (see Figs. 7(e)-7(f)). Their presentation varies with the tested state, and thus we report the observed mechanism rather than a fixed triggering threshold. At the master/vault-password entry points, NordPass and Zoho Vault instead reset or close the extension interface after repeated failures. Reopening the interface permits further attempts, and the correct master password remains usable in the tested sequences. These CAPTCHA and UI-reset responses introduce visible friction, but they do not stop the attacker from continuing the tested path.

Five entry points (i.e., LastPass, DualSafe, RoboForm, and the master password entry points of Kaspersky and Norton) enter a visible temporary lockout after repeated incorrect-password submissions (see Figs. 7(c) and 7(d) for examples of the lockout and an immediate correct-password attempt at the Kaspersky master-password entry point). During the observed blocking window, these entry points also block the correct credential. At the next checkpoint, approximately five minutes later, the correct credentials are accepted again. We use this five-minute checkpoint for all five entry points because Kaspersky

displays a five-minute lockout and we do not test other waiting intervals. We therefore describe these observations as lockouts without inferring that every restriction is enforced at the server or bound to the account.

The NordPass account-password entry point exhibits a state-dependent combination of CAPTCHA (see Fig. 7(a)) and rate limiting (see Fig. 7(b)). In one observed path, a correct account password is accepted after the first webpage-level "too many requests" state. In a second path, continued wrong-password submissions lead to a later "too many failed attempts" rate-limited state that blocks subsequent correct-password submissions. The next tested correct-password submission succeeds after waiting approximately five minutes. These paths establish a temporary, state-dependent restriction, but they do not establish a fixed trigger threshold or recovery duration. We classify this entry point as Rate Limit because the correct credential remains accepted after the initial restriction, and we report the later blocking as a state-dependent path.

CAPTCHAs and UI resets can increase interaction cost while still allowing attempts to continue, whereas the observed lockouts and one NordPass rate-limited path temporarily block even the correct credential.

4.4 Comparison with Website Measurements

The measurement framework of Al Roomi and Li [7] is the closest to ours in methodology, particularly in its black-box measurement of user enumeration signals and bounded repeated-failure probing. They found login-message-based user-enumeration signals on 19% of 31.2K evaluated domains and observed rate limiting on approximately 25-30% of the domains evaluated for that policy. In our measurements, six of 15 PMs expose account-existence signals, while 12 of 18 reachable entry points show at least one visible repeated-failure intervention. Both studies observe account-existence differences and entry points that permit repeated incorrect-password submissions. In our PM measurement, these include account-login and vault-unlock entry points. However, these results are not directly comparable because the samples and probing procedures differ.

5 Discussion

We first present the security implications of exposed signals and defensive mechanisms. Then, we discuss the limitations and future work of our study.

5.1 Attempts and Credential-Validity Feedback in Online Guessing

The feasibility of online password guessing depends on two separate factors: how many candidate passwords an attacker can submit and whether the responses reveal which candidates are correct. Our bounded repeated-failure probes provide evidence about whether further submissions remain possible within 15 attempts, while our credential-validity probes examine whether the

responses distinguish correct from incorrect credentials. Neither factor alone determines the feasibility of online guessing.

These two factors can produce different outcomes. A verification step reached only after a correct credential is submitted may block vault access while still revealing that the credential is correct. Conversely, the unavailable additional key prevents independent master-password evaluation for 1Password and Multi-Password, even though repeated submissions remain possible during our tested sequences. PM login flows should therefore be assessed by both the opportunities for continued submission and the information returned after each submission.

5.2 Repeated Failures and Legitimate Access

Repeated-failure responses affect continued password testing and legitimate access. In our tested flows, completing a CAPTCHA or reopening a reset interface permits another submission, although both actions add work for legitimate users. Progressively increasing delays or temporary lockouts can restrict continued submissions, but a restriction tied to an account may also allow an attacker to deny a legitimate user access to the vault [29]. Repeated email or MFA requests may create notification fatigue [20]. Thus, a public authentication entry point should not generate an unlimited sequence of victim-facing requests. Temporary restrictions should state their duration and provide recovery through a previously trusted device or another verified channel. Kaspersky's displayed remaining-attempt count at the master-password entry point (see Figs. 7(c) and 7(d)) can alert a legitimate user to earlier failed attempts, but it also tells an untrusted client how many attempts remain before lockout. Appropriate attempt thresholds and lockout durations require product-specific evaluation because our study does not measure legitimate-user error rates.

For the five PMs with separate account-login and vault-unlock entry points, repeated-failure responses differ between the two entry points. In our tests, all five account-password entry points present a CAPTCHA, with NordPass additionally showing rate limiting. At the vault-unlock entry points, Kaspersky and Norton show temporary lockouts, NordPass and Zoho Vault reset the UI, and Avira shows no visible intervention within the 15-attempt budget. These responses have different security and usability effects, although our work does not establish why the vendors select them. The behavior and usability cost at one entry point should therefore not be assumed to apply to the other.

5.3 Limitations and Future Work

Our work has the following limitations. First, our black-box method observes only feedback at public login entry points and cannot reconstruct server-side authentication or risk-control logic. It cannot determine whether restrictions are bound to an account, IP address, device, session, or another signal. It also cannot fully reset server-side risk state. Our results therefore only characterize behavior under the tested conditions. They have not been independently validated by PM vendors. Repeated sessions and available source code corroborate some

interpretations but do not establish longitudinal stability. The measurements remain a snapshot of the tested product versions and service states. Future longitudinal measurements can assess the stability of these findings over time.

Second, we study 15 selected cloud-based standalone PMs using controlled test accounts and configurations. The results do not cover built-in browser or OS password managers, local-only products, or every subscription, MFA, trusted-device, and account-history configuration. Future work can extend the framework to additional PMs, account configurations, and later product versions.

Third, the repeated-failure probe is limited to 15 incorrect-password submissions spaced at least five seconds apart and does not evaluate high-rate, long-running, or distributed guessing. "Not Observed" means only that no visible intervention appeared within this budget, and our observations do not establish exact trigger thresholds, recovery durations, or behavior beyond the tested window. Because our classification relies on client-visible evidence, it may miss defenses that leave no stable UI or network marker, and dynamic CAPTCHA or risk state may not be reproducible. Finally, we identify possible interaction and usability costs but do not measure their effects on legitimate users. Future user studies can examine these security-usability trade-offs.

6 Conclusion

In this paper, we have conducted a black-box measurement of 15 cloud-based password managers (PMs) across 20 public authentication entry points. Six PMs expose stable account-existence signals, while 15 of the 20 entry points expose stable credential-validity signals. During repeated-failure probing, 12 entry points exhibit visible interventions, six show none within our 15-attempt budget, and two are unreachable because victim-controlled verification precedes credential submission. Our results show that victim-controlled verification reached only after a correct credential may prevent vault access without hiding credential validity, while visible intervention responses such as CAPTCHA or UI reset may still allow continued probing. PM authentication should therefore be evaluated per entry point, considering what information is exposed, where verification occurs, and whether a failed-login defense restricts subsequent attempts. PM providers can reduce differences between correct and incorrect credential responses before additional verification and deploy failed-login defenses that balance resistance to continued probing against availability for legitimate users. Future research should track PM authentication policies longitudinally, determine whether failed-login defenses are bound to accounts, devices, sessions, or network addresses, and examine how rate limits and lockouts affect user access and denial-of-service risk.

Acknowledgments. We appreciate anonymous reviewers for invaluable comments. Qingxuan Wang is the corresponding author. This research was supported in part by the National Natural Science Foundation of China under Grant Number 62502240 and in part by the China Postdoctoral Science Foundation under Grant Number 2026T190415.

References

1. Dropbox Passwords discontinuation (Aug 2025), https://bit.ly/4fUElVW
2. 2026 Data Breach Investigations Report (2026), https://vz.to/4goy06w
3. Compare and analyze Chrome extensions (2026), https://chrome-stats.com/
4. What is a MultiPassword private key? (2026), https://bit.ly/4bL3Qb8
5. 1Password: About your Secret Key (Apr 2026), https://bit.ly/4o3bliu
6. Agarwal, A., O'Connell, S., Kim, J., Yehezkel, S., Genkin, D., Ronen, E., Yarom, Y.: Spook.js: Attacking Chrome Strict Site Isolation via Speculative Execution. In: Proc. IEEE S&P 2022. pp. 699–715
7. Al Roomi, S., Li, F.: A Large-Scale Measurement of Website Login Policies. In: Proc. USENIX SEC 2023. pp. 2061–2078
8. Amft, S., Höltervennhoff, S., Panskus, R., Marky, K., Fahl, S.: Everyone for Themselves? A Qualitative Study about Individual Security Setups of Open Source Software Contributors. In: Proc. IEEE S&P 2024. pp. 1065–1082
9. Anliker, C., Daniele, L., Capkun, S.: Phishing Attacks against Password Manager Browser Extensions. In: Proc. USENIX SEC 2025. pp. 7857–7876
10. Bitwarden: World Password Day 2025 Survey (2025), https://bit.ly/4fciGH7
11. Bitwarden: AccountController (2026), https://bit.ly/4xujeRl
12. Bitwarden: master-password.service.ts (2026), https://bit.ly/4q6gU0r
13. Bui, T., Rao, S.P., Antikainen, M., Bojan, V.M., Aura, T.: Man-in-the-Machine: Exploiting Ill-Secured Communication Inside the Computer. In: Proc. USENIX SEC 2018. pp. 1511–1525
14. Charnsethikul, P., Fattepurkar, A., Desai, D., Lucas, G., Mirkovic, J.: Replication: A Study on How Users (Don't) Use Password Managers. In: Proc. USEC 2026
15. CISA: Require Strong Passwords (2024), https://bit.ly/48OVkGV
16. Duan, Y., Wang, D., Fu, Y.: Security Analysis of Master-Password-Protected Password Management Protocols. In: Proc. IEEE S&P 2025. pp. 664–682
17. Duan, Y., Wang, D., Li, Y.: Credential Extraction Attacks Against Compromised Credential Checking Services of Password Managers. In: Proc. IEEE S&P 2026
18. Frew, P., Petrino, G.: 2024 Password Manager Industry Report and Statistics (Mar 2026), https://bit.ly/3LVdB9A
19. Fu, Y., Wang, D.: Leaky Autofill: An Empirical Study on the Privacy Threat of Password Managers' Autofill Functionality. In: Proc. ACSAC 2024. pp. 288–303
20. Golla, M., Ho, G., Lohmus, M., Pulluri, M., Redmiles, E.M.: Driving 2FA adoption at scale: Optimizing Two-Factor authentication notification design patterns. In: Proc. USENIX SEC 2021. pp. 109–126
21. Golla, M., Schnitzler, T., Dürmuth, M.: "Will Any Password Do?" Exploring Rate-Limiting on the Web. In: Proc. WAY 2018. pp. 1–5
22. Hasegawa, A.A., Watanabe, T., Shioji, E., Akiyama, M.: I know what you did last login: inconsistent messages tell existence of a target's account to insiders. In: Proc. ACSAC 2019. pp. 732–746
23. Hsu, S., Tran, M., Fass, A.: What is in the Chrome Web Store? In: Proc. AsiaCCS 2024. pp. 785–798
24. Huaman, N., Amft, S., Oltrogge, M., Acar, Y., Fahl, S.: They would do better if they worked together: The case of interaction problems between password managers and websites. In: Proc. IEEE S&P 2021. pp. 1367–1381
25. Kallstrom, G.: The Best Password Managers of 2026 (Apr 2026), https://www.passwordmanager.com/best-password-managers/
26. Lenovo: Lenovo Common Preloaded Software (2026), https://bit.ly/45YR2dL

27. Li, Z., He, W., Akhawe, D., Song, D.: The Emperor's New Password Manager: Security Analysis of Web-based Password Managers. In: Proc. USENIX SEC 2014
28. Lin, X., Ilia, P., Polakis, J.: Fill in the Blanks: Empirical Analysis of the Privacy Threats of Browser Form Autofill. In: Proc. ACM CCS 2020. pp. 507–519
29. Lu, B., Zhang, X., Ling, Z., Zhang, Y., Lin, Z.: A Measurement Study of Authentication Rate-Limiting Mechanisms of Modern Websites. In: Proc. ACSAC 2018. pp. 89–100
30. Lyastani, S.G., Schilling, M., Fahl, S., Backes, M., Bugiel, S.: Better managed than memorized? Studying the Impact of Managers on Password Strength and Reuse. In: Proc. USENIX SEC 2018. pp. 203–220
31. Manirathinam G: Why am I being asked for additional verification during sign-in? (2026), https://bit.ly/3RI2VBw
32. Mayer, P., Munyendo, C.W., Mazurek, M.L., Aviv, A.J.: Why Users (Don't) Use Password Managers at a Large Educational Institution. In: Proc. USENIX SEC 2022. pp. 1849–1866
33. Mayer, P., Zou, Y., Schaub, F., Aviv, A.J.: "Now I'm a bit angry:" Individuals' Awareness, Perception, and Responses to Data Breaches that Affected Them. In: Proc. USENIX SEC 2021. pp. 393–410
34. Microsoft: Microsoft Authenticator FAQs (2026), https://bit.ly/4wAuET7
35. Nisenoff, A., Golla, M., Wei, M., Hainline, J., Szymanek, H., Braun, A., Hildebrandt, A., Christensen, B., Langenberg, D., Ur, B.: A Two-Decade Retrospective Analysis of a University's Vulnerability to Attacks Exploiting Reused Passwords. In: Proc. USENIX SEC 2023. pp. 5127–5144
36. NordPass: NordPass Master Password and Account Password (2026), https://bit.ly/4wDRb2s
37. Norton: Secure your passwords and other personal information in your vault using Norton Password Manager (2026), https://bit.ly/3PMIbra
38. Oesch, S., Ruoti, S.: That Was Then, This Is Now: A Security Evaluation of Password Generation, Storage, and Autofill in Browser-Based Password Managers. In: Proc. USENIX SEC 2020. pp. 2165–2182
39. Pearman, S., Zhang, S.A., Bauer, L., Christin, N., Cranor, L.F.: Why people (don't) use password managers effectively. In: Proc. SOUPS 2019. pp. 319–338
40. Ray, H., Wolf, F., Kuber, R., Aviv, A.J.: Why older adults (Don't) use password managers. In: Proc. USENIX SEC 2021. pp. 73–90
41. SaferPass: SaferPass Is Closing July 31 — Please Export Your Data (2025), https://saferpass.com/help/contact-us
42. Scarlata, M., Torrisi, G., Backendal, M., Paterson, K.G.: Zero Knowledge (About) Encryption: A Comparative Security Analysis of Three Cloud-based Password Managers. In: Proc. USENIX SEC 2026. pp. 1–18
43. Statcounter: Desktop Browser Market Share Worldwide (2026), https://bit.ly/3LUNy2m
44. Temoshok, D., et al.: NIST SP 800-63B-4 digital identity guidelines: Authentication and authenticator management. Tech. rep. (July 2025), https://bit.ly/4t8lXy8
45. Trezor: Trezor Password Manager (2023), https://bit.ly/4xEj9Lf
46. Viezelyte, K.: Juggling security: How many passwords does the average person have in 2026? (May 2026), https://bit.ly/47YkyAr
47. Wang, D., Wang, P., He, D., Tian, Y.: Birthday, name and bifacial-security: Understanding passwords of Chinese web users. In: Proc. USENIX SEC 2019
48. Zhao, R., Yue, C.: All your browser-saved passwords could belong to us: A security analysis and a cloud-based new design. In: Proc. CODASPY 2013. pp. 333–340

A Details of Password Manager Selection

Candidate collection from Reddit. We search for "best password manager" and select five recommendation threads from `r/PasswordManagers` and `r/cybersecurity`, which contain general discussions of PM selection and usage. We do not include threads from `r/privacy` or `r/degoogle`, whose discussions may prioritize privacy-specific or de-Googling considerations. More specifically, we consider four threads from `r/PasswordManagers` (`https://bit.ly/4h5bIYy`, 25 PMs; `https://bit.ly/4hDVUfh`, 18 PMs; `https://bit.ly/4ySlrI8`, 18 PMs; `https://bit.ly/3S35nm4`, 18 PMs) and one thread from `r/cybersecurity` (`https://bit.ly/4fIsiLc`, 27 PMs). After deduplicating PM names, we obtain 45 unique PMs for subsequent screening.

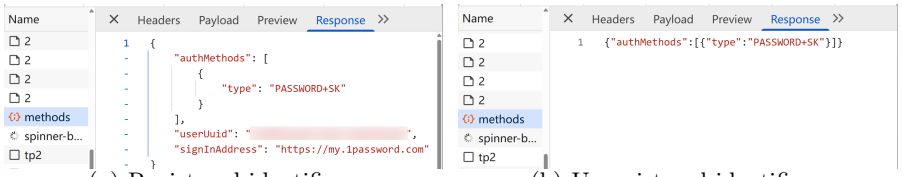
Additional exclusion details. After the initial screening described in Sec. 3.1, we manually inspect the remaining 27 candidates. Tab. 3 summarizes PMs excluded due to incompatibility with our measurement scope. We also report the five candidates excluded earlier because they are unavailable or discontinued.

Table 3: PM candidates excluded for reasons other than popularity.

PM name	Primary Reason
PMs unavailable or no longer maintained at the time of data collection (May 20, 2026).	
Dropbox Passwords	'Dropbox Passwords will be fully discontinued on October 28, 2025.' [1]
ThinkVantage PM	Lenovo lists Password Manager as end of life [26].
Microsoft Autofill	'Microsoft Autofill Chrome Extension was retired on December 14, 2024.' [34]
Trezor PM	'Trezor Password Manager has been deprecated.' [45]
SaferPass PM	'SaferPass will be shutting down on July 31, 2025.' [41]
PMs incompatible with our measurement scope.	
iCloud Passwords	No public login interface; relies on iCloud Keychain verification.
KeePassXC	No vendor-hosted vault; stores local encrypted databases.
ExpressKeys	Account authentication uses an emailed verification code.
Passbolt	Requires specific instance URL during registration for access.
Delinea Web	Enterprise PM with organization-specific deployment.
Enpass	No vendor-hosted vault; uses third-party cloud storage.
Passwordstate	Enterprise PM with administrator-configured deployment.
AuthPass	Local PM; cloud service only syncs encrypted databases.
Gokey	Stateless password generator without account or vault.
KeePass2Android	KeePass client without vendor-hosted vault service.
KeePassium	KeePass client without vendor-hosted vault service.
Vaultwarden	Self-hosted server with deployment-specific authentication.

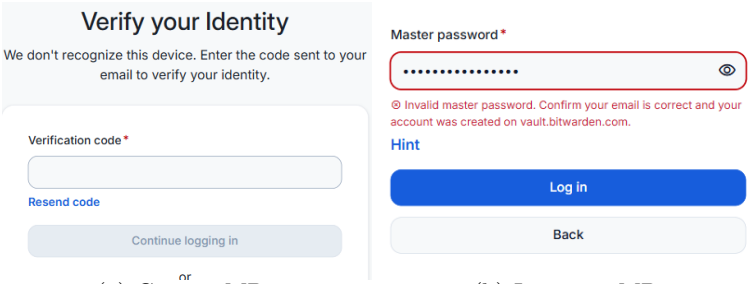
B Representative Evidence from Our Measurements

We present representative redacted evidence for the three measurement dimensions summarized in Tab. 2.

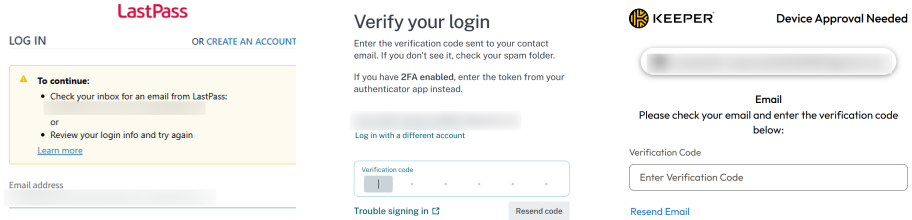


(a) Registered identifier. (b) Unregistered identifier.

Fig. 5: Representative redacted RQ1 evidence of an account-existence signal in 1Password. Both probes receive HTTP 200 responses to the same authentication method request, but only the registered-identifier response contains the `userUuid` (value redacted) and `signInAddress` fields.

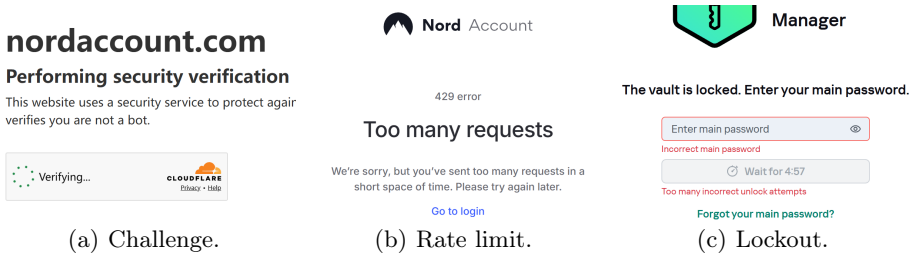


(a) Correct MP. (b) Incorrect MP.



(c) LastPass. (d) Dashlane. (e) Keeper.

Fig. 6: Representative redacted RQ2 evidence. (a)-(b) For Bitwarden, a correct master password (MP) reaches device verification, whereas an incorrect MP is blocked at the password step. (c) LastPass accepts the tested password but returns the same email-verification response for correct and incorrect credentials. (d)-(e) Dashlane and Keeper require victim-controlled verification before the target password can be submitted. These cases are classified as *Blocked*.



(a) Challenge. (b) Rate limit. (c) Lockout.

(d) Correct MP. (e) Kaspersky CAPTCHA. (f) Avira CAPTCHA.

Fig. 7: Representative redacted RQ3 evidence: (a)-(b) a Cloudflare challenge and HTTP 429 rate limiting at the NordPass account-password entry point; (c)-(d) a lockout at the Kaspersky master-password entry point that persists when the correct master password is submitted immediately after lockout; and (e)-(f) additional CAPTCHA examples at the account-password entry points.