Honeywords in the Wild: A First Empirical Study of Deployment Readiness and Adoption

David Iliopoulos
Delft University of Technology
Delft, Netherlands
davidiliopoulos@gmail.com

Georgios Smaragdakis
Delft University of Technology
Delft, Netherlands
g.smaragdakis@tudelft.nl

Antreas Dionysiou✉
Delft University of Technology
Delft, Netherlands
a.dionysiou@tudelft.nl

Abstract

Credential database breaches represent a persistent and costly threat to authentication security, with breach identification delayed by an average of 241 days. Honeywords, decoy passwords stored alongside legitimate credentials, were proposed in 2013 as a mechanism for accelerating breach detection and have since attracted sustained scientific attention. Yet no empirical evidence exists that honeywords have been deployed in any real-world production authentication system. This absence is particularly informative: honeyword deployment leaves a structurally detectable signature in leaked credential data (multiple password digests per user account rather than one). This paper presents the first empirical investigation into honeyword deployment, structured around two questions: whether honeyword technology has reached sufficient readiness for production deployment, and whether observable evidence in real-world credential database leaks reveals adoption in practice. We assess deployment readiness through four independent signals interpreted via the NASA Technology Readiness Level (TRL) framework, which together place honeyword technology at TRL 6. We then analyze 487 leaked server-side credential databases obtained from dark web forums, finding that every breached service stores exactly one password digest per user, including 85 services breached after prior work had removed the primary architectural barrier to practical honeyword deployment, the need for an always-trusted honeychecker. The central finding is that the absence of honeyword adoption is not explained by low TRL, but by practitioner unawareness, incentive misalignment, and operational friction.

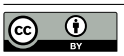
CCS Concepts

• **Security and privacy** → **Authentication**; *Intrusion detection systems*; Economics of security and privacy.

Keywords

honeywords, decoy passwords, credential database breach, password security, authentication, security measurement, dark web

ACM Reference Format:
David Iliopoulos, Georgios Smaragdakis, and Antreas Dionysiou. 2026. Honeywords in the Wild: A First Empirical Study of Deployment Readiness and Adoption. In *11th Workshop on Traffic Measurements for Cybersecurity (WTMC '26), November 15–19, 2026, The Hague, Netherlands.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3846375.3849099



WTMC '26, The Hague, Netherlands

ACM ISBN 979-8-4007-3017-7/2026/11
https://doi.org/10.1145/3846375.3849099

1 Introduction

Passwords remain one of the most widely deployed authentication mechanisms in online services, despite well-documented weaknesses [24]. Because verification requires storing a representation of the secret, credential databases accumulate sensitive authentication data at scale, making them persistent high-value targets. For example, a 2025 report documented 16 billion credentials exposed across major platforms including Google, Meta, and Apple [4]. Compounding this, breaches take 241 days on average to identify and contain [12]. Standard defenses (hashing, salting, and slow hash functions such as bcrypt [17]) protect stored passwords but do nothing to accelerate breach detection: by the time a breach is detected, attackers have had ample opportunity to sell credentials, take over accounts, or otherwise exploit the compromised data.

Honeywords, introduced by Juels and Rivest [13] in 2013, address this gap: a honeyword system stores several indistinguishable decoys alongside each real password, so an attacker who steals the database and authenticates with a honeyword triggers an alarm. Over the twelve years since, successive work has addressed the original proposal's architectural shortcomings, and honeyword implementations now exist across mainstream web authentication frameworks. Recent practitioner guidance reflects this growing relevance: the Dutch National Cyber Security Centre (NCSC) explicitly recommends honeywords for early intrusion detection [16].

Despite these advances, no empirical evidence exists that honeywords have been deployed in a production authentication system. This absence is not merely anecdotal: a service that deploys inline honeywords stores multiple password digests per user account, whereas a conventional service stores only one. This structural difference can be observed directly in leaked credential databases. However, interpreting the absence of such a signal first requires establishing that honeyword technology is sufficiently mature for deployment; otherwise, non-deployment would be unsurprising.

Accordingly, we investigate the following research questions. *RQ1:* Has honeyword technology reached a level of readiness sufficient for deployment in production authentication systems? *RQ2:* Do leaked credential databases contain observable structural evidence that honeywords have been deployed in practice? The two questions are sequential, not independent: *RQ1* asks whether non-deployment would even be meaningful, since low readiness would make it unsurprising, and only if readiness is sufficient does a negative answer to *RQ2* become informative.

This paper makes the following contributions:

- **The first empirical methodology for studying honeyword deployment in real-world authentication systems.** We introduce a two-stage methodology that first assesses whether honeyword technology has reached sufficient readiness for production

deployment and then examines leaked credential databases for the structural signature of adoption. This turns the previously theoretical honeyword deployment gap [5, 14] into an empirically testable question grounded in real-world evidence.

- **An evidence-based assessment of honeyword deployment readiness.** To answer *RQ1*, we evaluate four independent signals spanning academic literature, patent filings, source code repositories, and practitioner-facing discourse. Our analysis includes a publicly released Django implementation based on the Amnesia honeyword architecture [25], which we developed to minimize deployment friction and to serve as an instrumented probe of developer interest. Together, these signals place honeyword technology at TRL 6: demonstrated in relevant environments, but not yet supported by verifiable evidence of operational deployment.
- **Structural evidence of non-deployment in breached authentication systems.** To answer *RQ2*, we analyze 487 leaked credential databases spanning 11 industry sectors and find that all of them store a single password digest per user, the structural signature of non-deployment. This result holds across all sectors, including 85 breaches recorded after prior work reduced honeyword systems' reliance on an always-trusted honeychecker.

2 Background

Password security and honeywords. Standard practices (e.g., salted slow hashing, rate limiting, blocklists) reduce the risk that stolen credentials are immediately exploited, but do not address timely breach detection: a service may remain unaware its credential database has been exfiltrated for months. The 241-day mean time to identify and contain a breach [12] motivates mechanisms that turn otherwise silent breaches into detectable events at the moment of attempted account exploitation.

Honeywords [13] address this gap by storing k candidates per user, one real password and $k-1$ indistinguishable decoys (honeywords), so an attacker authenticating with a randomly chosen candidate risks an alarm with probability $(k-1)/k$. Honeywords are orthogonal to multi-factor authentication (MFA): MFA reduces the likelihood a stolen password grants access, whereas honeywords accelerate detection that the database was breached, addressing different stages of compromise.

Historical barriers to honeyword deployment. The original scheme [13] had two main shortcomings. First, it relied on a honeychecker: an external, always-trusted component that validates every authentication attempt and must remain unreachable to the attacker even after the credential database is breached. This requirement was widely recognized as a major deployment barrier [6, 25]. Amnesia [25] and Lethe [6] relax this reliance, making breach detection less dependent on a permanently trusted external component. Second, honeyword generation has largely relied on heuristics, such as modifying the real password [13] or sampling from password models [7], which must balance two competing goals: honeywords must be hard to distinguish from the real password for an attacker who breached the database, yet difficult for one without database access to guess and use to trigger false alarms. Huang et al. [11] showed such heuristics trade off poorly between false positives and negatives, and Bernoulli Honeywords [26] instead samples the password space via Bernoulli trials, enabling fine-grained control over false-positive rates. Together, these address the two main technical objections to deployment: dependence on an always-trusted honeychecker and reliance on heuristic honeyword generation techniques (HGTs).

The structural deployment signal. A key property exploited by this paper is that honeyword deployment leaves a directly observable signature in leaked credential databases. A service that deploys honeywords with parameter k stores k password digests per user account, whereas a service that does not deploy honeywords stores one. We capture this distinction using the *secrets-to-emails ratio*: the ratio between unique credential strings and unique user identifiers observed in a leaked database. A ratio near k indicates inline honeyword storage, while a ratio near one indicates conventional single-password storage. This signal can be measured without cooperation from the breached service, without active probing, and without access to any live authentication system.

3 Related Work

This paper builds on the tradition of large-scale empirical security measurement. Bonneau's analysis of 70 million passwords [3] established leaked credential databases as a legitimate empirical source for studying password structure without organizational cooperation; we extend that precedent from password distribution to deployment-signal analysis. Al Roomi and Li [1] showed that the gap between security recommendations and real-world deployment can be measured reproducibly at scale, and Internet-wide work such as ZMap [8] established the principle of inferring deployment state from externally observable artifacts without operator cooperation; our dark-web crawling applies similar logic to an environment conventional network scanning cannot reach. Szakály et al. [19] similarly found that 84% of EV charging stations still lack TLS a decade after standardization. No prior work combines these traditions to study honeyword deployment.

4 Methodology

Measuring honeyword deployment is hard: services have no incentive to disclose whether they use honeywords, so direct surveys are unreliable, and active probing fails because honeyword storage is a server-side property that cannot be inferred from external authentication behavior. We therefore proceed indirectly, through two complementary steps: whether honeywords could be deployed, and whether deployment is structurally detectable in the observable corpus of breached services. The pipeline is shown in Figure 1.

Step 1 assesses whether honeywords have reached a sufficient TRL for non-deployment to constitute a meaningful finding; if the technology were at a low readiness level, non-deployment would be unsurprising. Step 2 probes directly for structural deployment evidence in leaked credential databases.

4.1 Deployment Readiness Assessment

We assess deployment readiness through four complementary signals, each targeting a distinct channel of observable activity. The signals are distinct by design, since no single signal is sufficient on its own: a technology can be theoretically advanced but practically unknown, or commercially protected but never implemented.

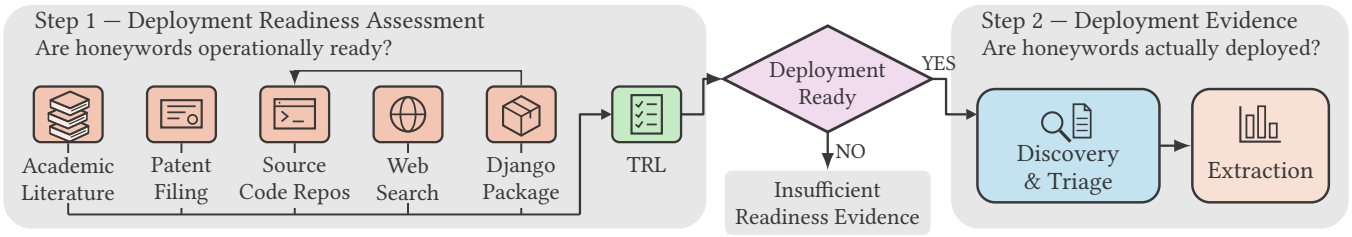


Figure 1: Two-step pipeline. Step 1 assesses deployment readiness through four independent signals interpreted via the NASA TRL framework. If readiness is established, Step 2 probes for structural deployment evidence in leaked credential databases.

Signal 1: Academic Literature. We collected honeyword-related papers through Semantic Scholar and arXiv using a three-tier keyword corpus (core terminology, extended technical vocabulary, and conceptual synonyms; full query counts in Appendix C), supplemented by forward citation chaining from four seed papers representing distinct architectural milestones [6, 13, 25, 26]. After deduplication and relevance filtering, the corpus contains 142 papers spanning 2013 to early 2026, classified into seven research-function categories derived inductively from the corpus (Appendix D).

Signal 2: Patent Filings. We queried Google Patents using Core Terminology only (six canonical terms — honeyword, honeychecker, sweetwords, and scheme-specific variants; full listing in Appendix C), applying a two-stage noise filter (an automated lexical filter followed by large language model (LLM)-assisted human review) to identify honeyword-specific filings.

Signal 3: Source Code Repositories. We searched GitHub across three complementary modes: repository search, code search restricted to Python, Java, Go, and Ruby, and commit-message search. This allowed us to capture honeyword implementations whether standalone or embedded within larger authentication systems. Included repositories were classified into four categories. As part of this signal, we developed and publicly released a pip-installable Django package implementing an Amnesia-based honeyword system [25], for two reasons. First, if integration friction were the primary barrier, a drop-in package should lower it, so its uptake directly tests that hypothesis. Second, its download and clone activity is a coarse, practitioner-facing engagement signal, adding a temporal, demand-side view to Signal 3's static count of published implementations.

Signal 4: Practitioner Awareness Sweep. We conducted a systematic sweep using a self-hosted SearXNG instance [18] across 31 queries, combining a direct tier (core and extended technical terminology) with an indirect tier of conceptual synonyms chosen in advance to avoid post hoc adjustment. Results were filtered by domain-level exclusion of known non-relevant sites, a negative-term filter removing unrelated deception and non-security content, and a co-occurrence filter requiring synonym matches to appear alongside a contextually paired term (e.g., *canary* only alongside *password*); survivors were reviewed for any evidence of real-world deployment or practitioner adoption. The two-tier design is intentional: if honeywords are known among practitioners, they should surface under either tier.

TRL Interpretation. Three of the four signals are interpreted through the NASA TRL framework [15], a nine-point scale from basic principle observation at TRL 1 to fully proven operational use at TRL 9, with each signal assigned the band its evidence type can substantiate (assignments and justification given alongside the verdict in Section 5). We use TRL as an interpretive lens rather than a precise audit: the analysis depends only on the coarse conclusion that readiness is high enough for non-deployment to be surprising, not on the exact level. The fourth signal, practitioner awareness, does not map onto a TRL level directly; instead it sets an upper bound, since visible discourse would be expected under either (direct or indirect) query tier if honeywords had reached practitioner-driven adoption. This upper bound is also the primary empirical basis for the deployment gap discussion in Section 6.

4.2 Deployment Evidence Collection

The dark web is a primary environment in which leaked credential databases circulate after a breach [20]. If a breached service had deployed honeywords, the structural signature (multiple password digests per user) should be observable in the leaked data. We designed a two-stage pipeline to surface this signal.

Discovery and Triage. We performed breadth-first crawling of onion services seeded from public index services (Ahmia, TorLinks, OnionLib), routing all traffic through Tor. A content-hashing mechanism collapsed mirror duplicates, and each node was fetched, scored, and its outbound links conditionally queued in a single pass, avoiding re-fetching in an unstable environment. Triage scoring applied a weighted keyword-indicator system with three tiers: structural database markers (weight 4), file extension indicators (weight 2), and breach-context vocabulary (weight 1); indicators are listed in Appendix E. The high-priority threshold of five requires corroboration across at least two tiers. An adaptive stopping condition terminated the crawl after a dynamically determined run of consecutive low-yield nodes. Figure 2 shows the crawl progression: distinct content plateaus well before node 12,001, indicating that further crawling would not have yielded additional relevant material. Sites showing law-enforcement or honeypot indicators were excluded regardless of triage score.

Structural Extraction. From the validated high-priority sites, we scraped approximately 6,000 forum posts and applied regex-based extraction to identify email addresses and credential-like strings across four format classes: JSON key-value pairs, email-password combination lines, keyword-prefixed password fields, and hexadecimal or password-hash strings. Only server-side database dumps preserve the per-service storage structure the secrets-to-emails ratio measures; the remaining categories were excluded from deployment signal analysis. Each post was classified into one of four leak types, applied in priority order from most to least

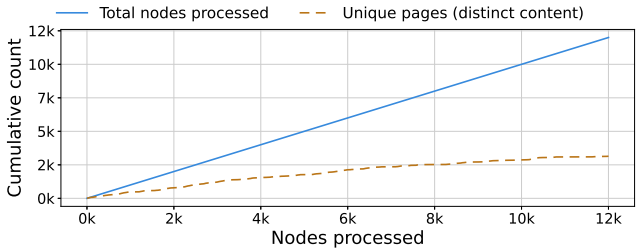


Figure 2: Cumulative count of total nodes processed and pages with distinct content during the onion service crawl. The unique-content curve plateaus well before crawl completion, indicating that continued crawling would not yield additional relevant pages.

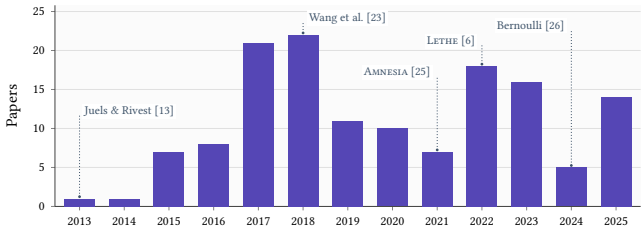


Figure 3: Annual publication volume of the 142-paper honeyword corpus (2013–2025), with foundational barrier-removal milestones annotated.

structurally distinctive: infostealer logs (system metadata alongside credential pairs), targeted combolists (high pair density with a named service in the title), compiled combolists (high pair density without a named service), and server-side database dumps (SQL structural markers such as `INSERT INTO` and WordPress-specific fields, classified last since they are only unambiguous once the other categories are excluded). We validated the pipeline's ability to detect inline honeyword storage using synthetic relational dumps seeded with $k \in \{2, 3\}$ digests per user. In every case, the classifier routed the dump to the server-side category and recovered a ratio equal to k, confirming that an inline honeyword deployment would register a ratio of at least 2.0 in the corpus.

5 Results

5.1 Deployment Readiness

We assess each signal in turn before stating the collective verdict.

Signal 1: Academic Literature. The collection pipeline returned 142 papers spanning 2013 to early 2026. Applying the seven-category classification (Appendix D), honeyword generation frameworks form the largest category, accounting for 55 papers (46.6% of substantive output; Figure 3). Five foundational papers trace a progression of barrier removal that has now run its course: Wang et al. [23] established the flatness requirement as the central generation-quality criterion, and Amnesia [25], Lethe [6], and Bernoulli [26] subsequently addressed the principal architectural barriers identified in the original 2013 proposal [13]: Amnesia and Lethe removed the need for an always-trusted honeychecker, and Bernoulli avoided the downsides of heuristic generation techniques. Crucially, 2025 marks the first year in which deployment-oriented

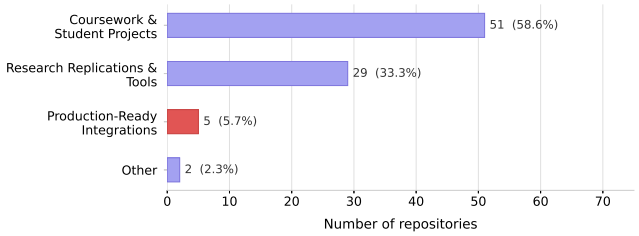


Figure 4: Functional classification of 87 honeyword-related GitHub repositories across four categories.

output reached meaningful concentration, with two of the eight deployment-focused papers in the twelve-year corpus appearing in a single year [5, 14], confirming that the community has begun treating adoption as the central unsolved problem.

Signal 2: Patent Filings. The pipeline identified 16 honeyword patents filed across the United States (6), China (9), and Taiwan (1) between 2013 and 2025. Filing activity is continuous with no multi-year gap, and half the corpus was filed from 2020 onwards, consistent with sustained research interest. A pronounced geographic shift is visible: every core patent since 2019 has been filed exclusively by Chinese institutions across four major research universities, activity not reflected in Western practitioner channels.

Signal 3: Source Code Repositories. The pipeline returned 87 repositories after deduplication, spanning 2013 to early 2026. The functional breakdown of these repositories is shown in Figure 4. Five production-ready integration repositories cover the authentication infrastructure of the majority of web applications: Ruby on Rails via Devise (2013), Linux PAM in C (2017), PHP CodeIgniter (2018), Python Django (2019), and a modular honeyword server (2024). That developers across four independent ecosystems produced integrations meeting production-readiness criteria (packaged releases, dependency management, or installation documentation) suggests they judged the mechanism suitable for real-world integration, not just academic demonstration. Our own implementation, the django-amnesia-honeywords package, released in February 2026, accumulated ≈600 mirror-filtered downloads over four months (February–June 2026) without advertising beyond a single public repository listing, and over 100 unique GitHub clones (Figure 5). These metrics indicate measurable developer interest in an accessible implementation, but they do not provide evidence of sustained operational adoption. In absolute terms, these figures describe a technology that is repeatedly patented and re-implemented yet whose production-ready integrations remain in the single digits.

Signal 4: Practitioner Awareness Sweep. The SearXNG sweep retained 77 results after filtering, of which only 4 (5.2%) originate from practitioner-authored content. Zero results survived filtering for any of the 14 conceptual synonym queries (Appendix C), including *decoy password*, *canary password*, and *trap password*, indicating that honeywords have no visible footprint in practitioner-facing channels even under informal vocabulary. The most notable result is a 2025 LastPass blog post framed explicitly around introducing honeywords to readers unfamiliar with the term[1], twelve years after their original proposal. The absence of practitioner discourse

[1] https://blog.lastpass.com/posts/honeywords-and-ai-detection

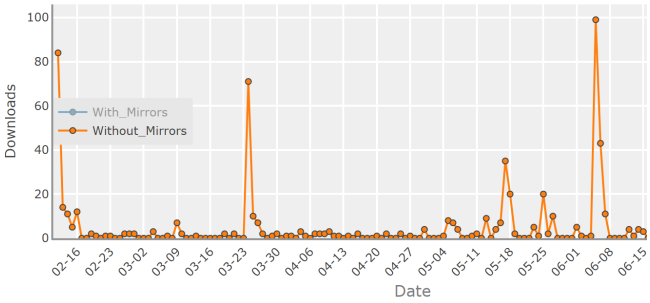


Figure 5: Cumulative `django-amnesia-honeywords` engagement over the four-month dissemination period (February–June 2026): PyPI downloads with and without mirror/CI traffic, and GitHub clones.

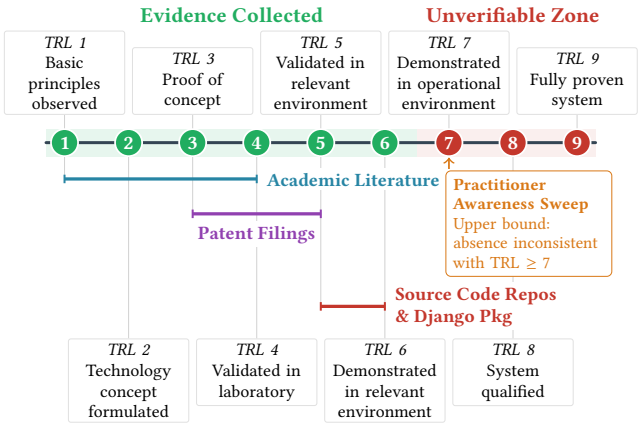


Figure 6: Four deployment readiness signals mapped onto the NASA TRL scale. Green indicates confirmed positive evidence; orange indicates a bounding or corroborating role rather than direct TRL contribution; the red zone marks levels unverifiable from surface sources.

makes practitioner-driven adoption at TRL 7 difficult to substantiate. We therefore use this signal as an upper bound on the readiness assessment, while examining actual deployment separately in Section 5.2.

TRL Verdict. Three signals place honeyword technology at TRL 6: the main architectural limitations identified in prior work have been addressed (academic literature, TRL 1–4), 16 patents confirm sustained recognition across multiple jurisdictions (patent filings, TRL 3–5), and five production-ready framework integrations across four languages, distinct from the coursework majority by the criteria in Appendix F, demonstrate deployability in a relevant environment (source code repositories, TRL 5–6). The practitioner awareness sweep bounds this verdict from above: the complete absence of practitioner discourse is inconsistent with community-driven adoption at TRL 7, which we classify as unverifiable rather than unmet, since operators have rational incentives not to disclose deployment publicly. Figure 6 maps each signal's evidential role onto the nine-point scale, and Table 2 (Appendix B) summarizes all four signals' contribution to the verdict; whether the structural signature of deployment is nonetheless detectable through other means is addressed directly in the following section.

5.2 Deployment Evidence

With TRL 6 established, we turn to whether honeywords have actually been deployed: whether any breached service in the observable corpus exhibits the structural signature of honeyword adoption. The remainder of this section describes how this question was investigated and what the resulting evidence shows.

Discovery and Triage. The crawler processed 12,001 nodes across the onion ecosystem. Of these, 52% were unreachable at crawl time. This reflects both the structural ephemerality of onion services and the disruption caused by a sustained law enforcement campaign that dismantled several major forums during the study period, including RaidForums, BreachForums, and Cracked/Nulled [9, 21, 22]. Content hashing collapsed 1,975 duplicate pages into mirror groups, yielding 3,780 unique pages. Triage scoring identified 31 high-priority domains; after manual validation and exclusion of link directories, document archives, paywalled sites, and CSAM-adjacent content (excluded for ethical and legal reasons), two validated general-purpose breach-trading forums remained, together supplying the raw forum posts analyzed in the next step.

Structural Extraction. Applied to the posts from the two validated sites, this step scraped approximately 6,000 forum posts, of which 1,343 matched a recognized credential-dump format. Of these, 490 produced a secrets-to-emails ratio of zero (PII breaches with no credential strings) and 366 were classified as infostealer logs, combolists, or produced corrupted ratios (implausibly high or low values caused by non-credential tokens being misparsed as passwords, or credential strings being missed), leaving 487 server-side database dumps spanning 11 industry sectors (Table 1, Appendix A).

Ratio Distribution. Figure 7 shows the secrets-to-emails ratio (defined in Appendix G) for the 487 server-side database dumps. Nearly all posts produce a ratio consistent with single-hash storage: 436 (89.5%) exhibit a ratio of exactly 1.0, and the remaining 51 deviate only marginally (42 between 1.01–1.10, 9 between 0.90–0.99). We manually inspected all 51: ratios below 1.0 came mostly from accounts with no stored password hash (e.g., OAuth, SSO) or truncated credential exports, and ratios above 1.0 from non-password strings (API keys, session or CSRF tokens) miscounted as credentials; none reached a ratio of 2.0, the signature of inline honeyword storage (Section 4.2), so we find no structural deployment signal among them. No post exhibits a ratio consistent with honeyword deployment (maximum observed: 1.10), uniformly across all 11 sectors (Table 1, Appendix A). Using the rule of three [10], zero detections in a sample of size n give an approximate 95% upper confidence bound of $3/n$; treating the corpus as a sample of breached services with comparable dump structure and detectability, this bounds the prevalence of detectable inline honeyword deployment below approximately 0.6% at 95% confidence ($3/487 \approx 0.6\%$).

Temporal Analysis. Of the 487 posts, 141 carry attributable breach timestamps. Of these, 85 (60.3%) correspond to breaches occurring in 2021 or later, after the publication of Amnesia [25], which eliminated the primary architectural deployment barrier (i.e., the need for a honeychecker). Every one of these 85 post-Amnesia records exhibits a secrets-to-emails ratio of exactly 1.0, including

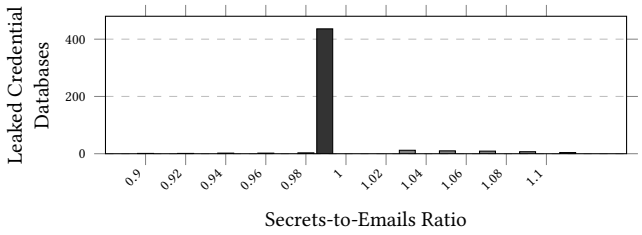


Figure 7: Distribution of secrets-to-emails ratios across 487 server-side database dumps; a ratio of exactly 1.0 is the signature of single-hash storage, exhibited by 436 posts (89.5%).

those from the Technology & Software sector, the organizations most likely to possess the in-house authentication infrastructure required for honeyword integration.

6 Discussion

Technology Readiness–Deployment Barriers. The TRL 6 verdict and the uniformly negative structural signal together suggest that non-deployment cannot be explained by technical unreadiness alone. We identify three barriers that help explain this readiness–deployment gap.

Barrier 1: Practitioner Unawareness. The practitioner-facing web sweep found little footprint for honeywords outside academic channels: twelve years of academic output have generated only four practitioner-authored web results, the most notable a 2025 LastPass blog post whose title presupposes readers do not know what honeywords are. A Reddit thread retrieved via the indirect query *honeypot password* even shows a user reinventing the concept without reference to the literature. The NCSC's recent honeyword deployment guidance [16] is, to our knowledge, the first instance of a national cybersecurity authority incorporating the mechanism into practitioner-facing guidance.

Barrier 2: Incentives and Private Verifiability. Honeywords create an incentive problem: they impose integration and monitoring costs, but public disclosure of their benefits would undermine the defensive uncertainty they depend on, while non-disclosure makes deployment hard to reward through ordinary signals such as customer-facing security claims. Private verification by insurers or auditors would require trusted intermediaries and standardized evidence that do not yet exist, so organizations may rationally deprioritize honeywords even when the technology is mature enough for operational evaluation. Nanda and Reiter [14] identify this economic barrier and propose incentive mechanisms to address it; our paper provides empirical grounding for that theoretical argument.

Barrier 3: Operational Friction and False-Positive Risk. Engagement with our Django package indicates interest in an accessible implementation, not verifiable deployment, suggesting friction persists at the integration level even after architectural barriers are removed. A second barrier is the asymmetric risk of false positives: Bernoulli Honeywords [26] bounds the false-positive probability per marking campaign, but the cumulative probability across a deployment's lifetime can grow substantially, and the literature lacks a per-lifetime bound, a gap operators bearing the cost of each alert may treat as disqualifying.

Limitations. Several caveats bound the negative result. The corpus derives from only two sites that specialize in reposting previously circulated dumps rather than sourcing breaches directly, so the uniform null may partly reflect what these sites host. The ratio signal also requires a complete per-user record; partial exports and combolists fall outside the server-side category by design and are excluded rather than analyzed. The corpus is further skewed toward organizations that suffered a full database exfiltration severe enough to be traded on these forums, likely over-representing services with weaker security practices. A ratio of one does not rule out deployment with certainty for honeychecker-based schemes, since an attacker who compromised both the database and the honeychecker could repost only the verified real password, a pattern indistinguishable from non-deployment by our structural signal alone. Relatedly, our structural signal assumes inline honeyword storage manifests as multiple password digests per user record; it therefore cannot detect decoys encoded within a single field, nor schemes such as ErsatzPasswords [2], which store a single machine-dependent digest backed by a hardware security module rather than additional stored secrets. Such deployments yield a secrets-to-emails ratio of 1.0 and are indistinguishable from non-deployment under our method. Nor can we independently verify the authenticity of dark web postings, some of which are fabricated or relabeled to attract buyers. Finally, two factors that we do not measure plausibly shape adoption. Retrofitting honeywords onto legacy credential stores carries integration cost that greenfield deployment does not, and such systems are plausibly over-represented among breached databases. Separately, federated authentication (OAuth 2.0, SSO) concentrates credential storage in identity providers; this reduces the number of services maintaining their own password databases, but not the underlying need, since identity providers remain high-value credential stores to which honeywords apply.

Future Work. The methodology is inherently longitudinal: repeated application of the same pipeline can track whether practitioner guidance [16] or incentive mechanisms [14] lead to detectable deployment over time. We will continue monitoring the Django package and aim to complement surface-level measurement with confidential evidence from practitioners, insurers, or national authorities, where deployment can be verified without disclosure.

7 Conclusion

We present the first empirical study of honeyword deployment. Four readiness signals place the technology at TRL 6, yet none of 487 breached services across 11 sectors shows the structural signature of inline honeyword storage. This does not rule out private or non-inline deployments, but adoption leaves little observable trace, pointing to limited practitioner visibility, weak incentives, and unresolved false-positive handling as likely barriers.

Acknowledgments

This research has received funding from the European Union's Horizon Europe research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101206668, and under projects SafeHorizon (grant agreement No 101168562) and RECITALS (grant agreement No 101168490). Views and opinions

expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency (REA). Neither the European Union nor the granting authority can be held responsible for them.

References

[1] Suood Al Roomi and Frank Li. 2023. A Large-Scale Measurement of Website Login Policies. In *USENIX Security*. Anaheim, CA, 2061–2078.

[2] Mohammed H. Almeshekah, Christopher N. Gutierrez, Mikhail J. Atallah, and Eugene H. Spafford. 2015. ErsatzPasswords: Ending Password Cracking and Detecting Password Leakage. In *ACSAC*. 449–458.

[3] Joseph Bonneau. 2012. The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords. In *S&P*. 538–552. doi:10.1109/SP.2012.49

[4] CyberNews. 2025. 16 Billion Passwords Exposed in Record-Breaking Data Breach: What Does It Mean for You? https://cybernews.com/security/billions-credentials-exposed-infostealers-data-leak/.

[5] Sudiksha Das and Ashish Kundu. 2025. Advancing Honeywords for Real-World Authentication Security. https://arxiv.org/abs/2510.22971. arXiv:2510.22971 [cs.CR]

[6] Antreas Dionysiou and Elias Athanasopoulos. 2022. Lethe: Practical data breach detection with zero persistent secret state. In *EuroS&P*. IEEE, 223–235.

[7] Antreas Dionysiou, Vassilis Vassiliades, and Elias Athanasopoulos. 2021. HoneyGen: Generating Honeywords Using Representation Learning. In *AsiaCCS*. 265–279.

[8] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. 2013. ZMap: Fast Internet-wide Scanning and Its Security Applications. In *USENIX Security*. Washington, D.C., 605–620.

[9] Europol. 2025. Law Enforcement Takes Down Two of the Largest Cybercrime Forums in the World. Accessed: 2026-07-13. https://www.europol.europa.eu/media-press/newsroom/news/law-enforcement-takes-down-two-largest-cybercrime-forums-in-world

[10] James A. Hanley and Abby Lippman-Hand. 1983. If Nothing Goes Wrong, Is Everything All Right? Interpreting Zero Numerators. *JAMA* 249, 13 (1983), 1743–1745. doi:10.1001/jama.1983.03330370053031

[11] Zonghao Huang, Lujo Bauer, and Michael K Reiter. 2024. The impact of exposed passwords on honeyword efficacy. In *USENIX Security*. 559–576.

[12] IBM Security. 2025. Cost of a Data Breach Report 2025. https://www.ibm.com/reports/data-breach. Accessed: 2026-6-2.

[13] Ari Juels and Ronald L Rivest. 2013. Honeywords: Making password-cracking detectable. In *CCS*. 145–160.

[14] Mridu Nanda and Michael K. Reiter. 2025. Incentivizing Collaborative Breach Detection. arXiv:2506.04634 [cs.CR] doi:10.48550/arXiv.2506.04634

[15] NASA. 2012. Technology Readiness Level. https://www.nasa.gov/directorates/somd/space-communications-navigation-program/technology-readiness-levels/. Accessed: 2025-12-15.

[16] Nationaal Cyber Security Centrum. 2025. Digitaal Struikeldraad: Misleiding-gebaseerde Detectie. https://www.ncsc.nl/detectie/digitaal-struikeldraad.

[17] Niels Provos and David Mazières. 1999. A Future-Adaptable Password Scheme. In *USENIX ATC*. Monterey, CA, 81–91.

[18] SearXNG Contributors. 2021. SearXNG: A Privacy-respecting, Hackable Metasearch Engine. https://github.com/searxng/searxng.

[19] Marcell Szakály, Sebastian Köhler, and Ivan Martinovic. 2025. Current Affairs: A Security Measurement Study of CCS EV Charging Deployments. In *USENIX Security*. 7997–8015.

[20] Kurt Thomas, Frank Li, Ali Zand, Jacob Barrett, Juri Ranieri, Luca Invernizzi, Yarik Markov, Oxana Comanescu, Vijay Eranti, Angelika Moscicki, Daniel Margolis, Vern Paxson, and Elie Bursztein. 2017. Data Breaches, Phishing, or Malware? Understanding the Risks of Stolen Credentials. In *CCS*. 1421–1434.

[21] U.S. Department of Justice. 2022. United States Leads Seizure of One of the World's Largest Hacker Forums and Arrests Administrator. Accessed: 2026-07-13. https://www.justice.gov/archives/opa/pr/united-states-leads-seizure-one-world-s-largest-hacker-forums-and-arrests-administrator

[22] U.S. Department of Justice. 2023. Justice Department Announces Arrest of the Founder of One of the World's Largest Hacker Forums and Disruption of Forum's Operation. Accessed: 2026-07-13. https://www.justice.gov/archives/opa/pr/justice-department-announces-arrest-founder-one-world-s-largest-hacker-forums-and-disruption

[23] Ding Wang, Haibo Cheng, Ping Wang, Jeff Yan, and Xinyi Huang. 2018. A Security Analysis of Honeywords. In *NDSS*. 1–16.

[24] Ding Wang, Zijian Zhang, Ping Wang, Jeff Yan, and Xinyi Huang. 2016. Targeted online password guessing: An underestimated threat. In *CCS*. 1242–1254.

[25] Ke Coby Wang and Michael K Reiter. 2021. Using amnesia to detect credential database breaches. In *USENIX Security*. 839–855.

[26] Ke Coby Wang and Michael K. Reiter. 2024. Bernoulli Honeywords. In *NDSS*.

Ethics and Privacy Statement

Access was restricted exclusively to publicly reachable content without payment or circumvention of access controls — no paid memberships were purchased, no access controls were bypassed, and no registration information was falsified. The extraction pipeline retains no email addresses, passwords, hashes, or any other personal identifiers; only aggregate structural metrics are written to disk. Sites showing indicators of law enforcement involvement or honeypot infrastructure were excluded regardless of triage score. This study was conducted in accordance with institutional research ethics guidelines. Generative AI tools were used for classification assistance across four signals: academic literature categorization, patent filing relevance filtering, source code repository classification, and web search result filtering.

A Sector Distribution

Table 1 breaks down the 487 server-side database dumps analyzed in Section 5.2 by industry sector.

Table 1: Distribution of database dumps across sectors.

Sector	Posts	%
E-commerce & Retail	99	20.3
Technology & Software	94	19.3
Gaming & Entertainment	64	13.1
Finance & Crypto	52	10.7
Education	39	8.0
Government & Public	36	7.4
Social & Community	28	5.7
Telecommunications	27	5.5
Travel & Hospitality	26	5.3
Healthcare	8	1.6
Other	14	2.9
Total	487	100.0

B TRL Verdict

Table 2 summarizes the four readiness signals and the TRL band each supports, as discussed in Section 5.1.

Table 2: Summary of the four deployment readiness signals and their contribution to the TRL assessment.

Signal	Key Finding	TRL
Academic literature	142 papers; architectural barriers addressed	1–4
Patent filings	16 patents; US, China, Taiwan; continuous since 2013	3–5
Source code repos	87 repos; 5 production-ready; ~600 downloads	5–6
Practitioner sweep	4 results; 0/14 synonym hits	≤6 (bound)
Verdict		**TRL 6**

C Keyword Corpus

Table 3 lists the six canonical terms used to seed all four collection pipelines (Section 4.1).

Table 3: Core Terminology: six canonical terms applied across all four collection pipelines.

Term	Source
honeyword(s)	[13]
honeychecker	[13]
sweetwords	[13]
Amnesia honeyword(s)	[25]
Lethe honeyword(s)	[6]
Bernoulli honeyword(s)	[26]

Table 4 reports the number of queries issued per pipeline, by keyword tier.

Table 4: Query counts per pipeline by keyword tier.

Pipeline	Keyword Tier	Queries
Academic literature	Core + Extended	17
Patent filings	Core only	6
Source code (GitHub)	Core only	6
Web search	Core + Extended + Synonyms	31

Table 5 lists the conceptual synonyms used only in the practitioner awareness sweep (Signal 4).

Table 5: Eight conceptual synonyms used in 14 practitioner-awareness queries.

Term	Variants
decoy password	decoy passwords, decoy credential
canary password	canary passwords
trap password	trap passwords
honeypot password	honey password
fake credential detection	decoy credential detection
credential canary	—
password deception	—
honeyword deployment	—

D Research-Function Categories

Table 6 defines the seven categories used to classify the academic literature corpus (Signal 1).

Table 6: Seven research categories used to classify the academic literature corpus.

Category
Foundational works
Generation schemes
Attack and analysis
Deployment and practical considerations
Hybrid and extension works
Descriptive overviews
Surveys and systematization of knowledge

E Triage Scoring System

Table 7 details the weighted indicator tiers used to score onion domains during crawling (Section 4.2).

Table 7: Weighted keyword-indicator tiers for triage scoring.

Tier	Example indicators	Weight
Critical	sql dump, insert into, create table, wp_users	4
File extension	.sql, .csv, .rar, .zip, .tar.gz	2
Contextual	database leak, data breach, hacked, exposed	1

F GitHub Repository Categories

Table 8 defines the four categories used to classify GitHub repositories (Signal 3).

Table 8: Four mutually exclusive categories of honeyword-related GitHub repositories, ordered by deployment intent.

Category	Classification criteria
Coursework & Student Projects	Course assignments and student projects.
Research Replications & Tools	Independent replications of published schemes or standalone research tools.
Production-Ready Integrations	At least one of: package published on a registry, versioned release, dependency management files, CI/CD configuration, or installation docs aimed at end users.
Others	Curated lists aggregating honeyword resources alongside other mechanisms.

G Secrets-to-Emails Ratio

For each server-side database dump, the secrets-to-emails ratio is defined as $r = \frac{|C|}{|E|}$ where $|C|$ is the count of unique credential-like strings extracted from the post after blacklist filtering, and $|E|$ is the count of unique email addresses extracted from the same post. A service storing exactly one password hash per user produces $r \approx 1$; a service storing k honeyword candidates per user produces $r \approx k$.