EPiC Series in Computing

Volume 111, 2026, Pages 66–78

Proceedings of The 21st Asia Joint Conference on Information Security

Keep It Lean: Longitudinal Maintenance of URL Blocklists Using Real-User Web Access Logs

Mizuho Hasegawa[1]*, Rui Tanabe[2], Kanta Shinbori[2], Yasuhiro Kunimine[2], and Katsunari Yoshioka[1]

[1] Yokohama National University, Yokohama, Japan
yoshioka@ynu.ac.jp
[2] Clwit, Inc., Tokyo, Japan
tanabe-r@clwit.co.jp, kunimine@clwit.co.jp

Abstract

The rapid proliferation of malicious websites poses a significant threat to Internet users. While URL-based blocklists are widely used as a practical defense mechanism, their effectiveness depends not only on identifying newly emerging malicious URLs but also on maintaining the blocklist appropriately over time. This study evaluates the long-term operational effectiveness of a URL blocklist generated from real-user web access logs.

From June 2024 to January 2026, we operated a URL blocklist generation method proposed in our previous work. We collected web access logs from real users and identified malicious URLs related to actual browsing activities. By adding these URLs and removing entries not accessed for more than two weeks, we maintained our blocklist every day and conducted a longitudinal cross-referencing experiment. Over 337 days, 6,866 malicious URLs were identified while maintaining an average blocklist size of 1,248 URLs. The blocklist matched 2,817 accesses from 102 users. We also found that 17% of the malicious URLs were removed soon after, whereas some URLs remained active for long periods. In addition, accesses to URLs after removal revealed limitations of simple lifetime-based maintenance. Finally, we conducted a real-user deployment in which the blocklist was updated daily and distributed weekly. Over 569 days, 13,208 malicious URLs were identified, while the average blocklist size was maintained at 1,330 URLs. During the 20-month evaluation, 84 users accessed URLs included in the blocklist 844 times. These results demonstrate that continuous removal of URLs can provide practical protection in real browsing environments while maintaining a relatively small blocklist size.

Keywords: URL Blocklist, Web Access Log

1 Introduction

In recent years, malicious websites, including malware distribution sites [1], phishing sites [2] and fake shopping sites [3] have become a serious threat to Internet users. According to the

*This work was done while researching at Yokohama National University.

APWG [4] (Anti-Phishing Working Group), phishing activity has continuously increased worldwide, with hundreds of thousands of phishing sites observed every quarter. Moreover, the threat posed by malicious websites is not limited to the increase in their number; attackers have also continuously evolved their techniques. In particular, attackers increasingly abuse search engine optimization (SEO) techniques [5] to manipulate search engine rankings and lure users to malicious websites through seemingly trustworthy search results.

One of the most widely adopted countermeasures against malicious websites is access control based on URL-based blocklist. For example, the Chrome web browser, which is used worldwide, incorporates Google Safe Browsing [6] (GSB), a mechanism that rapidly matches URLs accessed by users against known malicious URL lists and blocks users from visiting malicious websites. In 2022, Google reported that Chrome displayed more than 3 million warnings per day [7]. Similarly, security vendors have also developed browser-based protection mechanisms and URL reputation services as practical approaches to mitigating malicious websites. However, attackers frequently generate and discard large numbers of websites within short periods of time to evade detection [8, 9], requiring defenders to continuously collect and maintain blocklists [10, 11] .

A large body of research has focused on generating and expanding URL blocklists for protecting users from malicious websites [12–21]. Previous studies have proposed various techniques for automatically discovering malicious URLs using machine learning, search engines, web crawling, and threat intelligence services. Other studies empirically analyzed public blocklists and revealed characteristics such as update delays, URL lifetime, overlap among providers, and limitations in protection coverage over time [10, 11, 15–18]. However, prior work primarily focused either on malicious URL detection or on retrospective analyses of existing public blocklists. To the best of our knowledge, limited work has investigated how continuously adding newly observed malicious URLs and removing outdated entries affects the long-term operational effectiveness of a URL blocklist deployed to real users.

In this study, we investigate the long-term operational effectiveness of a URL-based blocklist generated from real-user web access logs. Moreover, we demonstrate how a URL blocklist can be continuously maintained and practically deployed in real browsing environments. For this evaluation, we operated a blocklist generation method proposed in our previous work [9, 22]. We first collected web access logs from real users and identified malicious domains related to actual browsing activities. We utilized a Google Chrome browser extension functioning as a sensor deployed to real users and collected logs everyday. Since the project web page [23] is written in Japanese, most participants are assumed to be located in Japan. We then utilized VirusTotal [24] to extract and analyze associated malicious URLs.

From March 2025 to January 2026, we conducted a longitudinal cross-referencing experiment to analyze the effectiveness of continuous URL addition and removal for blocklist maintenance. In this experiment, we detected new malicious URLs every day and were added to the blocklist daily, and URLs that had not been accessed for more than two weeks were removed from the blocklist. Over 337 days, 6,866 malicious URLs were identified while maintaining the average blocklist size at 1,248 URLs. The generated blocklist matched 2,817 accesses from 102 users. In addition, 1,157 URLs (17%) were removed through maintenance, indicating that several malicious URLs are short-lived and that continuous maintenance is effective for suppressing blocklist growth. Our longitudinal analysis also revealed several operational characteristics of malicious URLs. We observed that 16 URLs were accessed again after removal. This suggests that simple lifetime-based maintenance cannot completely prevent missed detections caused by URL reappearance. Furthermore, accesses were highly concentrated on a small number of URLs; the most frequently accessed URL was visited 1,567 times by 39 users. This indicates that although several malicious URLs are short-lived, some URLs remain active and repeatedly

affect users over long periods. Finally, we evaluated the practical effectiveness of the maintained blocklist through a long-term real-user deployment. During this evaluation, the blocklist was updated daily and distributed weekly to users. From June 2024 to January 2026 (569 days), 13,208 malicious URLs were identified while maintaining an average blocklist size of 1,330 URLs. The deployed blocklist blocked 844 accesses from 84 users in real browsing environments. These results demonstrate that continuously maintained URL blocklists can provide practical protection while maintaining a relatively small blocklist size.

Summarizing, the contributions of this paper are as follows:

- We show that a URL blocklist generated from real-user web access logs can provide long-term protection. We demonstrate that our continuous URL addition and removal can suppress blocklist growth while retaining URLs effective for user protection.
- In our longitudinal cross-referencing experiment, the blocklist maintained an average size of 1,248 URLs while matching 2,817 accesses from 102 users over 337 days. We reveal that accesses to malicious URL were concentrated, where a small number of URLs repeatedly affect multiple users over long periods. We analyzed limitations of simple lifetime-based maintenance by identifying accesses to malicious URLs after URL removal.
- We investigated the operational effectiveness of the maintained blocklist through a real-user deployment over 569 days. The deployed blocklist maintained an average size of 1,330 URLs and blocked 844 accesses from 84 users, demonstrating practical protection in real browsing environments using a relatively small number of URLs.

2 Generating and Maintaining URL Blocklist

In order to generate and maintain an URL-based blocklist, we utilized the malicious URL extraction method proposed in our previous work. Rather than re-proposing the detection method itself, we briefly describe the operational procedure and maintenance process.

First, web access logs from real users are collected. We use a security agent [23] to collect web access logs from real users. Anyone who agrees to the terms and conditions for participation can use the application freely. Note that it is still being distributed under a government project and is designed for users in Japan. Therefore most participants are expected to be from the country. The application works as an extension of the Google Chrome browser and uses Google Chrome APIs to collect web access data (e.g., user ID, URL and domain information, and accessed date). The data will be periodically sent to the central server for further analysis. From the collected logs, only domain information is extracted, and domains likely to be benign are excluded in order to generate a set of potentially malicious domains.

Next, the extracted domains are submitted to the online inspection service VirusTotal [24] to obtain related malicious URL information. The retrieved malicious URLs are then submitted again to VirusTotal to acquire the latest analysis results. Finally, the URL is registered into the blocklist only when the it is detected as malicious by more than a predefined threshold number of antivirus engines registered in VirusTotal. During the experiment, if more than eight security vendors detected a URL, we added the URL to the blocklist.

These two processes are executed every day, enabling newly observed malicious URLs to be continuously added to the blocklist. This operational framework allows the blocklist to adapt to newly emerging malicious websites. However, continuously adding URLs without removing outdated entries causes the blocklist size to increase indefinitely, potentially affecting distribution cost and matching performance. Therefore, this study also introduces a maintenance process

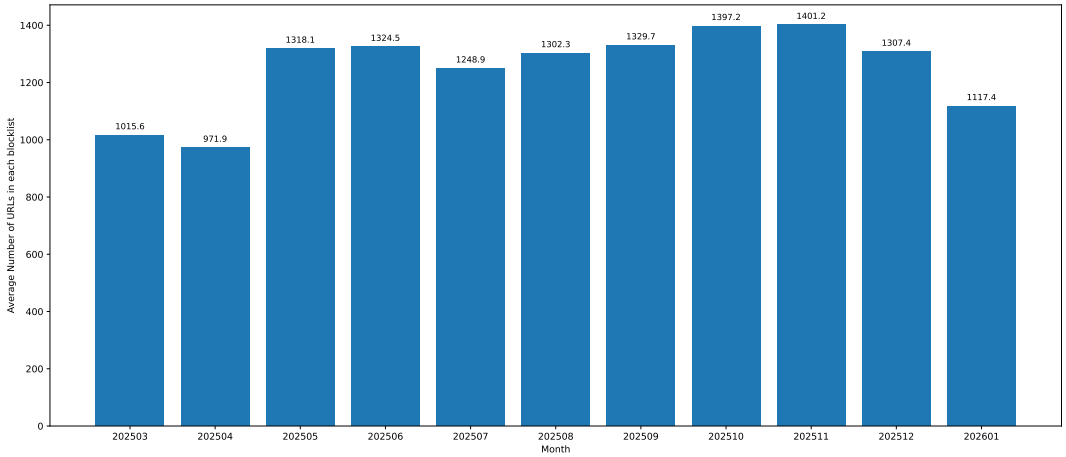

Figure 1: The number of URLs included in the blocklists from March 2025 to January 2026.

for removing stale entries. Specifically, the distributed blocklist follows three maintenance rules: (1) URLs that were classified as malicious within the previous two weeks are continuously redistributed, or (2) among URLs accessed by users, up to 100 URLs with the newest access timestamps are periodically re-analyzed using VirusTotal, and URLs still classified as malicious are continuously redistributed, or (3) URLs are re-analyzed by VirusTotal and still classified as malicious continue to be distributed. Note that during the experiment, due to the API quote, we re-analyzed up to 200 URLs per day and removed URLs that exhibited this limitation. All of this allow us to continuously eliminate outdated entries while maintaining the freshness.

The blocklist is distributed to users via a security agent. Users attempting to access a registered URL are blocked and presented with a warning screen. This screen explains the reason for the block and provides "Proceed" and "Go Back" buttons, requiring the user to make a proactive decision.

3 Longitudinal Analysis

In this section, we investigate the long-term operational effectiveness of a URL-based blocklist generated from real-user web access logs. Section 3.1 presents a longitudinal cross-referencing experiment to analyze the effectiveness of continuous URL addition and removal for blocklist maintenance. Section 3.2 evaluates the practical effectiveness of the maintained blocklist through a long-term real-user deployment. Both experiments used the same blocklist generation and maintenance method described in the previous section.

3.1 Longitudinal Cross-Referencing Experiment

While our previous studies evaluated the blocklist maintenance method using short-term data [9, 22], this study further investigates its effectiveness over a long-term period. From March 2025 to January 2026 (11-months), we conducted a longitudinal cross-referencing experiment in which malicious URLs were detected daily and continuously added to the blocklist. Outdated URLs were removed according to the proposed maintenance policy. Over 337 days, 6,866 malicious

URLs were identified while maintaining the average blocklist size at 1,248 URLs. Figure 1 shows the monthly average number of URLs included in the blocklist.

During the observation period, the average number of daily active users who accessed more than one URL per day was 380. The generated blocklist matched 2,817 accesses to 243 unique malicious URLs from 102 users. Our longitudinal analysis revealed several operational characteristics of malicious URLs. In particular, 16 URLs were accessed again after being removed from the blocklist, indicating that simple lifetime-based maintenance cannot completely prevent missed detections caused by URL reappearance. We also observed that accesses were highly concentrated on a small number of URLs. The most frequently accessed URL was visited 1,567 times by 39 users and remained in the blocklist for an extended period.

To further analyze URL persistence, we investigated the lifetime distribution of URLs included in the blocklist. Figure 2 shows the number of URLs grouped by their lifetime. After being added to the blocklist, 1,157 URLs (17%) were removed within two weeks without receiving additional user accesses, being re-analyzed by VirusTotal, or remaining classified as malicious. Similarly, 45% of the URLs were removed within 30 days.

Under the proposed maintenance process, URLs that are not accessed within two weeks are re-analyzed using VirusTotal before removal. If the number of antivirus detections still exceeds the predefined threshold, the URL is redistributed in the blocklist for an additional two weeks. As a result, many URLs exhibited lifetimes of 28, 42, and 56 days due to repeated re-validation and redistribution cycles. Once URLs no longer receive accesses or fail re-validation, they are gradually removed from the blocklist on a daily basis. In contrast, URLs that continuously receive user accesses may remain in the blocklist for long periods; seven URLs remained in the blocklist throughout the entire observation period.

These findings indicate that many malicious URLs are short-lived and quickly disappear from practical browsing environments, whereas a subset of persistent malicious URLs continuously survives through repeated re-validation and redistribution. Furthermore, the proposed maintenance process effectively suppresses unlimited blocklist growth by removing inactive URLs while retaining URLs that continue to pose threats to users.

3.2 Real-User Deployment Evaluation

While our previous studies evaluated the blocklist maintenance method primarily through simulations using experimental data [9, 22], this study further investigates its effectiveness through a live deployment in a real-user environment. From June 2024 to January 2026 (20-months), we conducted a real-user deployment in which the blocklist was updated daily and distributed to users on a weekly basis. In this deployment, the same blocklist was used for protection over each seven-day distribution period. Over 569 days, 13,208 malicious URLs were identified while maintaining the average blocklist size at 1,330 URLs. Figure 3 shows the monthly average number of URLs included in the blocklist.

During the observation period, the average number of daily active users was 394, with 3,566 total active users. The deployed blocklist matched 844 accesses from 84 users. Figure 4 shows the monthly number of accesses to URLs included in the blocklist, the number of affected users, and the number of blocked accesses, including duplicate accesses. Although the number of accesses varied across months, malicious URL accesses were continuously observed throughout the deployment period, demonstrating that the blocklist provided practical protection in real browsing environments.

To further understand user exposure to malicious websites, we conducted a user-based analysis. Figure 5 shows the monthly number of newly affected users and previously affected users

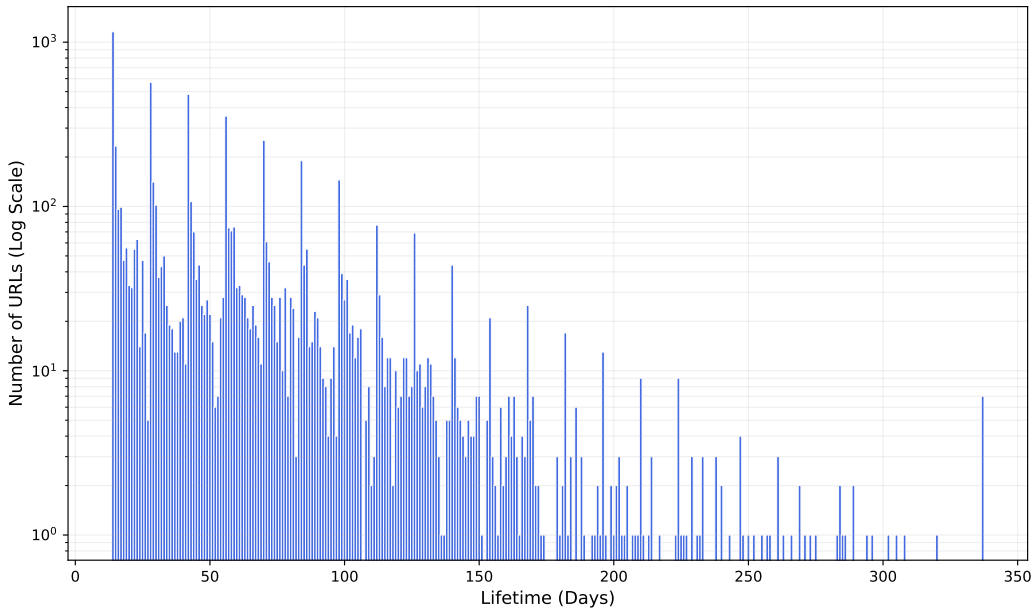

Figure 2: Lifetime distribution of URLs in the blocklists from March 2025 to January 2026.

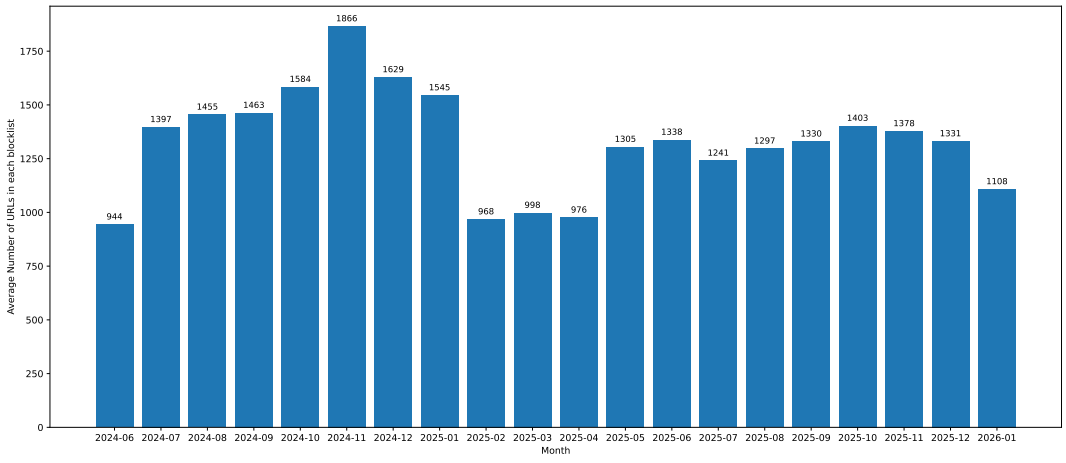

Figure 3: The number of URLs included in the blocklists from June 2024 to January 2026.

who accessed URLs included in the blocklist. Newly affected users continued to appear throughout the entire observation period, indicating that malicious websites continuously emerged and affected different users over time. At the same time, a subset of users repeatedly accessed malicious websites across multiple months, suggesting persistent exposure and the need for continuous protection.

We then analyzed the distribution of malicious website accesses on a per-user basis. Figure 6 shows the number of malicious website accesses per user. The majority of affected users accessed

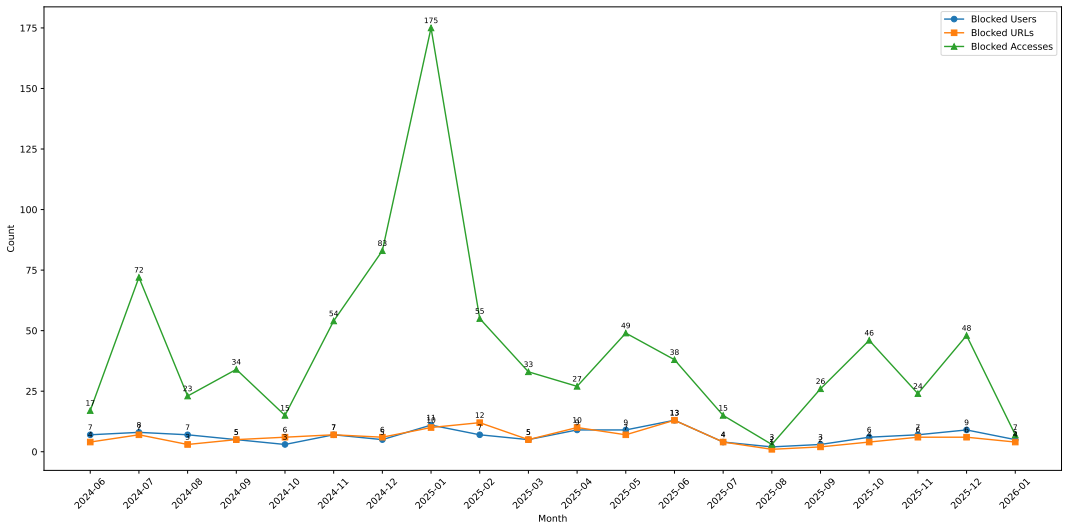

Figure 4: The number of affected users, and the number of blocked accesses, including duplicate accesses, by the blocklists from June 2024 to January 2026.

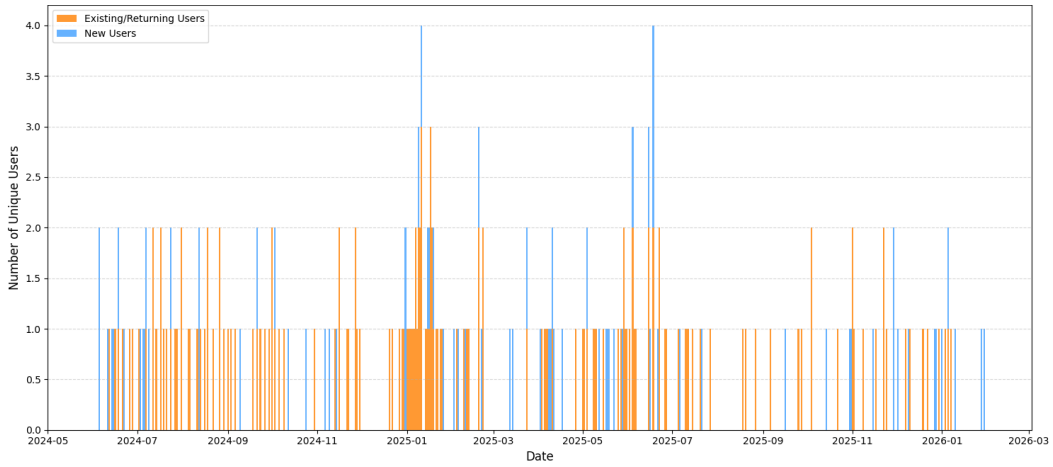

Figure 5: The number of newly and previously affected users who accessed URLs included in the blocklists from June 2024 to January 2026.

malicious websites only once, whereas a small number of users repeatedly accessed malicious websites many times. Specifically, 36.9% (31 out of 84) of affected users accessed malicious websites only once, while several users accessed malicious websites more than 30 times, and one user accessed malicious websites 130 times during the observation period. While many users encounter malicious websites only occasionally, some users appear to be persistently exposed to malicious content, potentially due to risky browsing behavior, repeated visits to compromised websites, or continuous exposure through malicious advertisements and search engine poisoning.

Finally, we analyzed the lifetime distribution of URLs included in the blocklist. After being

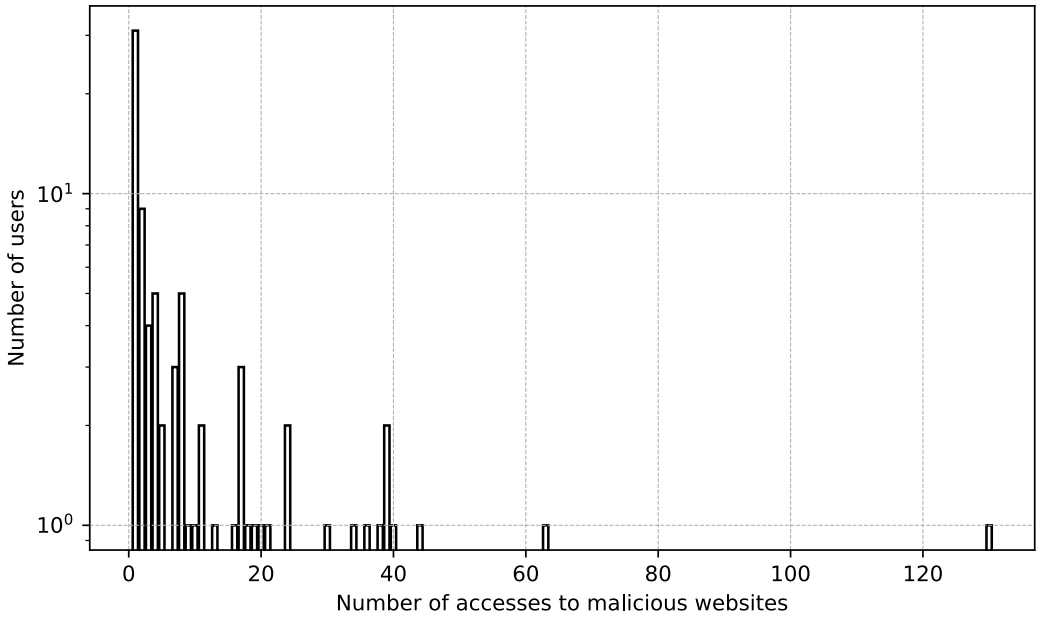

Figure 6: The number of malicious website accesses per user from June 2024 to January 2026.

added to the blocklist, 2,667 URLs (20%) were removed within two weeks. In contrast, the longest-lived URL remained in the blocklist for 453 days.

Overall, these findings demonstrate that continuously maintained URL blocklists can provide practical and sustained protection against malicious websites in real-world browsing environments while maintaining a manageable blocklist size.

4 Related Work

Malicious URL detection and URL-based blocklisting have long been studied as fundamental approaches for protecting users from web-based attacks. Early studies focused on automatically identifying malicious URLs from the Internet using machine learning, search-engine analysis, and web crawling techniques [25–30]. Ma *et al.* [25] proposed a supervised machine learning approach to classify URLs as benign or malicious. Akiyama *et al.* [26] improved URL blocklisting by discovering malicious URLs located in the neighbourhood of already known malicious URLs through search engines. Subsequent studies targeted malware distribution and compromised websites. For example, Zhang *et al.* proposed *PoisonAmplifier* [27], which identifies compromised websites by analyzing poisoned webpages and their links, while Stokes *et al.* proposed *WebCop* [28] to detect malware distribution sites and their related landing pages. Invernizzi *et al.* proposed *EvilSeed* [29], which expands known malicious URLs by generating search queries to discover similar malicious webpages. More recently, Bronjon *et al.* [30] proposed a hybrid approach combining traditional URL blocklists with machine learning techniques to improve the detection of previously unseen malicious URLs.

Based on these detection techniques, many free and commercial URL blocklists have been developed and publicly released [31–33]. Akiyama *et al.* proposed *AutoBLG* [12], a framework that automatically collects new malicious URLs starting from existing URL blocklists. As

phishing attacks became a major threat on the Internet, researchers also proposed phishing-specific detection methods. Corona *et al.* proposed *DeltaPhish* [13], which detects phishing webpages by comparing their HTML structure and visual appearance with those of legitimate websites. Likewise, researchers have been detecting and collecting fraudulent shopping sites. Existing studies have proposed a wide range of technical approaches, including detection methods based on HTML structure, textual features, and image characteristics, as well as techniques leveraging machine learning and large language models (LLMs) [34–37].

Several studies have investigated the operational characteristics and effectiveness of URL blocklists themselves. Sheng *et al.* [14] evaluated phishing blocklists by measuring phishing campaign durations as well as the update speed and coverage of blocklists. Kührer *et al.* [15] analyzed the growth and size of phishing blocklists over time. Bell *et al.* [16] investigated the uptake, dropout, lifetime, and overlap of URLs among Google Safe Browsing, OpenPhish, and PhishTank. Mitsuhiro *et al.* [17] surveyed public blocklists and reported characteristics such as lifespan, update frequency, geolocation bias, and the existence of invalid URLs. Feal *et al.* [18] conducted a transparency and content analysis of open-source blocklists, revealing their structure, maintenance processes, and relationships with other providers. These studies demonstrated that different blocklists have relatively small overlaps and varying update characteristics, raising questions regarding how effectively continuously maintained blocklists protect users in real-world environments.

Researchers have also explored the use of external threat intelligence services such as VirusTotal for malicious URL analysis and blocklist generation. Tanaka *et al.* [38] proposed a method to efficiently collect malicious URL candidates from open-source intelligence (OSINT) and generate URL blocklists using web crawlers and sandbox analysis. Le *et al.* [39] utilized VirusTotal as an anti-virus aggregation platform and demonstrated a method to reduce malware analysis costs while maintaining low false-negative rates. Pritom *et al.* [40] used VirusTotal threat intelligence to label malicious or compromised domains for machine learning-based detection. Peng *et al.* [41] deployed artificial phishing websites to evaluate how VirusTotal and its anti-phishing vendors detect phishing content over time, revealing substantial differences in detection performance across vendors.

While previous studies extensively analyzed malicious URL detection methods and the characteristics of public blocklists, limited work has investigated how continuously maintained blocklists should be operated in practical environments, including the addition of newly observed malicious URLs, the removal of outdated entries, and long-term deployment to real users. In contrast, our study evaluates a continuously maintained URL blocklist generated from real-user browsing activities over a long-term period. We focus not only on blocklist effectiveness but also on operational aspects such as URL lifetime, maintenance policies, blocklist size suppression, and repeated exposure to malicious websites in real browsing environments.

5 Discussion

5.1 Ethical Consideration

In conducting this study, we adhered closely to ethical standards. We consulted with the project's Internal Review Board (IRB) to ensure that our use of the logs was ethical and respectful of users' privacy. Before installation, all users were required to agree to the browser extension's terms and conditions and license agreement, which specifies that the collected information will be used for research purposes. Data were collected under the specified terms and conditions, and we analyzed only the data relevant to the objectives of our experiments. As the

Software License Agreement outlined, the dataset used to identify unwanted accesses consisted solely of accessed URLs with anonymized user IDs.

5.2 Limitations

During the cross-reference experiment, URLs in the blocklist that matched with real-user web access logs were 243. Among these URLs, we observed that 41 URLs had been accessed before their initial registration and that our blocklist generation method is not perfect.

During the malicious URL detection process, we decided that if more than six security vendors detected a URL, we added the URL to the blocklist. This conservative threshold was chosen based on prior studies to robustly minimize false positives in a live user environment. The real user used in the experiment was around one thousand per day and that we used the commercial VirusTotal API under the quota of 1,000 API calls per day, which determined our daily query capacity. Adjusting the threshold for identifying malicious URLs or altering the API quota will result in different outcomes. Another key feature is that, adjusting the period for removing malicious URLs will result in different outcomes. These consideration is crucial depending on the target use case. While these preset values represent a balanced and practical configuration for our target use case, we leave this evaluation for future work.

6 Conclusion

Over a 20-month experiment, we analyzed web access logs from real users and queried their domains using VirusTotal to obtain related URLs. Investigating the number of security vendor's detection results, we identified 13,208 unique malicious domains. Our longitudinal analysis revealed that 84 users accessed URLs in the blocklist 844 times, and demonstrated the continuously maintained URL blocklists can provide practical protection in real-world browsing environments. While maintaining a relatively small blocklist size, we analyzed limitations of simple lifetime-based maintenance by identifying accesses to malicious URLs before blocklist registration and after URL removal. Although this study treated malicious URLs as a single undifferentiated category, the lifespans and access frequencies may vary depending on specific threat types. In future work, we plan to optimize dynamic maintenance strategies tailored to the unique characteristics of each threat category, and to conduct comparative evaluations against other existing blocklist maintenance strategies. While our longitudinal analysis revealed that a portion of malicious URLs reappear after being removed from the blocklist, developing mechanisms to dynamically handle these recurring threats remains a key open challenge that we leave this for future works.

Acknowledgements. This work was supported by JSPS KAKENHI Grant Number 21H03444 and 21KK0178. This work was supported in part by the collaborative research with Clwit, Inc.

References

[1] Kaspersky, 2024. [Online]. Available: https://www.kaspersky.com/resource-center/definitions/drive-by-download

[2] CloudFlare, "What is a phishing attack?" 2026. [Online]. Available: https://www.cloudflare.com/learning/access-management/phishing-attack/

[3] P. Kotzias, "Why Fake Online Stores Are Surging — And How to Spot Them," https://us.norton.com/blog/online-scams/fake-e-shops.

[4] APGW, "Phishing activity trends reports," 2026. [Online]. Available: https://apwg.org/trendreports

[5] C. Carpineto and G. Romano, "Learning to detect and measure fake ecommerce websites in search-engine results," in *Proceedings of the international conference on web intelligence*, 2017, pp. 403–410.

[6] (2022) "google safe browsing". Google. [Online]. Available: https://safebrowsing.google.com/

[7] Google, "Google safe browsing - google transparency report," 2024. [Online]. Available: https://transparencyreport.google.com/safe-browsing/overview

[8] A. Oest, P. Zhang, B. Wardman, E. Nunes, J. Burgis, A. Zand, K. Thomas, A. Doupé, and G.-J. Ahn, "Sunrise to sunset: Analyzing the end-to-end life cycle and effectiveness of phishing attacks at scale," in *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, Aug. 2020, pp. 361–377. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/oest-sunrise

[9] K. Takao, C. Hiraishi, R. Tanabe, K. Takada, A. Fujita, D. Inoue, C. Gañán, M. van Eeten, K. Yoshioka, and T. Matsumoto, "Vt-sos: A cost-effective url warning utilizing virustotal as a second opinion service," in *NOMS 2024-2024 IEEE Network Operations and Management Symposium*. IEEE, 2024, pp. 1–5.

[10] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang, "An empirical analysis of phishing blacklists," 2009.

[11] A. Oest, Y. Safaei, P. Zhang, B. Wardman, K. Tyers, Y. Shoshitaishvili, and A. Doupé, "{PhishTime}: Continuous longitudinal measurement of the effectiveness of anti-phishing blacklists," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 379–396.

[12] B. Sun, M. Akiyama, T. Yagi, M. Hatada, and T. Mori, "Automating url blacklist generation with similarity search approach," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 4, pp. 873–882, 2016.

[13] I. Corona, B. Biggio, M. Contini, L. Piras, R. Corda, M. Mereu, G. Mureddu, D. Ariu, and F. Roli, "Deltaphish: Detecting phishing webpages in compromised websites," in *European Symposium on Research in Computer Security*. Springer, 2017, pp. 370–388.

[14] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang, "An empirical analysis of phishing blacklists," in *6th Conference on Email and Anti-Spam (CEAS 09)*. Carnegie Mellon University, 2009, pp. 1–10.

[15] M. Kührer and T. Holz, "An empirical analysis of malware blacklists," *PIK-Praxis der Informationsverarbeitung und Kommunikation*, vol. 35, no. 1, p. 11, 2012.

[16] S. Bell and P. Komisarczuk, "An analysis of phishing blacklists: Google safe browsing, openphish, and phishtank," in *Proceedings of the Australasian Computer Science Week Multiconference*, ser. ACSW '20. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3373017.3373020

[17] M. Umizaki, T. Morikawa, A. Fujita, T. Takahashi, T. Lin, and D. Inoue, "Understanding the characteristics of public blocklist providers," in *2022 IEEE Symposium on Computers and Communications (ISCC)*, 2022.

[18] Á. Feal, P. Vallina, J. Gamba, S. Pastrana, A. Nappa, O. Hohlfeld, N. Vallina-Rodriguez, and J. Tapiador, "Blocklist babel: On the transparency and dynamics of open source blocklisting," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1334–1349, 2021.

[19] J. Zhang, P. A. Porras, and J. Ullrich, "Highly predictive blacklisting." in *USENIX security symposium*, 2008, pp. 107–122.

[20] A. Kachole and S. Jondhale, "Unwanted message filtering system from osns user's wall using customizable filtering rules and black list techniques," *International Journal of Emerging Technology and Advanced Engineering (IJETAE)*, vol. 4, no. 2, pp. 752–755, 2014.

[21] J. Mahesh, P. Subhash, and T. Ravindra, "Customizable content based filtering method to filter texts and images from social network user wall," *International Journal of Computer Science and*

Information Technologies (IJCSIT), vol. 6, no. 4, pp. 3611–3614, 2015.

[22] M. Hasegawa, Y. Kawaguchi, H. Takayama, Y. Kunimine, H. Tanaka, S. Kawamura, A. Fujita, S. Yasuda, D. Inoue, N. Fujiwara *et al.*, "A case study of url blocklist maintenance," in *2024 IEEE Conference on Dependable and Secure Computing (DSC)*. IEEE, 2024, pp. 100–101.

[23] NICT, "Warpdrive," 2024. [Online]. Available: https://warpdrive-project.jp/

[24] VirusTotal, "Virustotal," 2024. [Online]. Available: https://www.virustotal.com

[25] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious urls," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '09. New York, NY, USA: Association for Computing Machinery, 2009, pp. 1245—-1254. [Online]. Available: https://doi.org/10.1145/1557019.1557153

[26] M. Akiyama, T. Yagi, and M. Itoh, "Searching structural neighborhood of malicious urls to improve blacklisting," in *2011 IEEE/IPSJ International Symposium on Applications and the Internet*. New York, NY, USA: Initialism of Institute of Electrical and Electronics Engineers, 2011, pp. 1–10.

[27] J. Zhang, C. Yang, Z. Xu, and G. Gu, "Poisonamplifier: A guided approach of discovering compromised websites through reversing search poisoning attacks," in *Research in Attacks, Intrusions, and Defenses*, D. Balzarotti, S. J. Stolfo, and M. Cova, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 230–253.

[28] J. W. Stokes, R. Andersen, C. Seifert, and K. Chellapilla, "WebCop: Locating neighborhoods of malware on the web," in *3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET 10)*. San Jose, CA: USENIX Association, Apr. 2010, pp. 1–8. [Online]. Available: https://www.usenix.org/conference/leet-10/webcop-locating-neighborhoods-malware-web

[29] L. Invernizzi, P. M. Comparetti, S. Benvenuti, C. Kruegel, M. Cova, and G. Vigna, "Evilseed: A guided approach to finding malicious web pages," in *2012 IEEE Symposium on Security and Privacy*. New York, NY, USA: Initialism of Institute of Electrical and Electronics Engineers, May 2012, pp. 428–442.

[30] B. Gogoi, T. Ahmed, and A. Dutta, "A hybrid approach combining blocklists, machine learning and deep learning for detection of malicious urls," in *2022 IEEE India Council International Subsections Conference (INDISCON)*, 2022, pp. 1–6.

[31] Zeltser, "Free blocklists of suspected malicious ips and urls," Zeltser, 2024. [Online]. Available: https://zeltser.com/malicious-ip-blocklists/

[32] (2024) Phishtank — join the fight against phishing. PhishTank. [Online]. Available: https://phishtank.org/

[33] OpenPhish. (2024) Phishing intelligence. OpenPhish. [Online]. Available: https://openphish.com/index.html

[34] K. Sakai, K. Takeshige, K. Kato, N. Kurihara, K. Ono, and M. Hashimoto, "An automatic detection system for fake japanese shopping sites using fasttext and lightgbm," *IEEE Access*, vol. 11, pp. 111 389–111 401, 2023.

[35] M. Bitaab, H. Cho, A. Oest, Z. Lyu, W. Wang, J. Abraham, R. Wang, T. Bao, Y. Shoshitaishvili, and A. Doupé, "Beyond phish: Toward detecting fraudulent e-commerce websites at scale," in *2023 IEEE Symposium on Security and Privacy (S&P)*, 2023, pp. 2566–2583.

[36] M. Sánchez-Paniagua, E. Fidalgo, E. Alegre, and F. Jáñez Martino, "Fraudulent e-commerce websites detection through machine learning," in *Hybrid Artificial Intelligent Systems: 16th International Conference, HAIS 2021, Bilbao, Spain, September 22–24, 2021, Proceedings*. Berlin, Heidelberg: Springer-Verlag, 2021, p. 267–279. [Online]. Available: https://doi.org/10.1007/978-3-030-86271-8_23

[37] M. Bitaab, A. Karimi, Z. Lyu, A. Oest, D. Kuchhal, M. Saad, G.-J. Ahn, R. Wang, T. Bao, Y. Shoshitaishvili *et al.*, "Scammagnifier: Piercing the veil of fraudulent shopping website campaigns." in *32nd Annual Network and Distributed System Security Symposium, NDSS 2025, San Diego, California, USA, February 24-28, 2025.*, 2025.

[38] T. Yasuyuki and K. Shingo, "Seedsminer: Accurate url blacklist-generation based on efficient osint seed collection," in *Proc of the IEEE/WIC/ACM International Conference on Web Intelligence*, ser. WI '19, 2019, pp. 250–255.

[39] S. Le Blond, C. Gilbert, U. Upadhyay, M. Gomez-Rodriguez, and D. Choffnes, "A broad view of the ecosystem of socially engineered exploit documents," Feb. 2017.

[40] M. M. A. Pritom and S. Xu, "Supporting law-enforcement to cope with blacklisted websites: Framework and case study," in *2022 IEEE Conference on Communications and Network Security (CNS)*, 2022, pp. 181–189.

[41] P. Peng, L. Yang, L. Song, and G. Wang, "Opening the blackbox of virustotal: Analyzing online phishing scan engines," in *Proceedings of the Internet Measurement Conference*, ser. IMC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 478–485. [Online]. Available: https://doi.org/10.1145/3355369.3355585