A First Look at the Privacy Harms of the Public Suffix List

Stephen McQuistin
University of St Andrews
St Andrews, UK
sm@smcquistin.uk

Peter Snyder
Brave Software
San Francisco, USA
pes@brave.com

Colin Perkins
University of Glasgow
Glasgow, UK
csp@csperkins.org

Hamed Haddadi
Imperial College London & Brave Software
London, UK
h.haddadi@imperial.ac.uk

Gareth Tyson
Hong Kong University of Science & Technology
Guangzhou, China
gtyson@ust.hk

ABSTRACT

The public suffix list is a community-maintained list of rules that can be applied to domain names to determine how they should be grouped into logical organizations or companies. We present the first large-scale measurement study of how the public suffix list is used by open-source software on the Web and the privacy harm resulting from projects using outdated versions of the list. We measure how often developers include out-of-date versions of the public suffix list in their projects, how old included lists are, and estimate the real-world privacy harm with a model based on a large-scale crawl of the Web. We find that incorrect use of the public suffix list is common in open-source software, and that at least 43 open-source projects use hard-coded, outdated versions of the public suffix list. These include popular, security-focused projects, such as password managers and digital forensics tools. We also estimate that, because of these out-of-date lists, these projects make incorrect privacy decisions for 1313 effective top-level domains (eTLDs), affecting 50,750 domains, by extrapolating from data gathered by the HTTP Archive project.

CCS CONCEPTS

• **Security and privacy → Privacy protections**; **Domain-specific security and privacy architectures**; • **Information systems →** *Web applications.*

KEYWORDS

Web privacy; domain boundaries

ACM Reference Format:
Stephen McQuistin, Peter Snyder, Colin Perkins, Hamed Haddadi, and Gareth Tyson. 2023. A First Look at the Privacy Harms of the Public Suffix List. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (IMC '23), October 24–26, 2023, Montréal, QC, Canada.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3618257.3624836

IMC '23, October 24–26, 2023, Montréal, QC, Canada.

ACM ISBN 979-8-4007-0382-9/23/10…$15.00
https://doi.org/10.1145/3618257.3624836

1 INTRODUCTION

The public suffix list (PSL) is a community-maintained list of rules that can be applied to domain names to determine how they should be grouped into logical organizations or companies. These are referred to as *effective top-level domains* (eTLDs). The public suffix list is, for example, how web browsers know that `www.google.com` and `maps.google.com` are two domains run by a single organization "google.com" (since `google.com` is *not* a suffix on the list), but that `google.co.uk` and `yahoo.co.uk` are not (since `co.uk` *is* a suffix on the list). At present, the list contains nearly 9500 rules.

The public suffix list is a critical system for enforcing privacy boundaries on the Web in many different, and sometimes subtle, ways. Web browsers, for example, use the list for determining which origins can access cookies for another site (for example, why `google.co.uk` cannot access cookies set by `yahoo.co.uk`). The public suffix list is updated regularly and software using an outdated version can cause serious privacy harms. The more out-of-date the list is, the more often errors will be made.

With this in-mind, we present the first large-scale measurement study of how the public suffix list is used in open-source software projects. While some projects use the list correctly, by updating it regularly, we find a number that incorrectly use the public suffix list in several ways. This includes hard-coding versions of the list into a binary that is never updated, using libraries that rely on the developer of the parent project to manually update the list at build time, and attempting to automatically update the list but failing and continuing to function without an error, among others.

We find that these errors are common: 24.9% of the projects that we identify as using the list include a fixed, hard-coded list that is out-of-date (with a median age of 825 days), while only 12.8% include a version that is routinely updated. When determining the privacy boundaries in a recent representative sample of Web requests, we find that projects using hard-coded lists make incorrect decisions for 1313 eTLDs, affecting 50,750 domains. Many of the missing suffixes allow for the hosting of arbitrary content (*e.g.,* 27 projects are missing `digitaloceanspaces.com`), exposing users of those projects to significant potential privacy harms.

Our findings highlight the risks of using static lists to define privacy boundaries on the Web. We hope that they will encourage the safe use of the public suffix list, and provide data that motivates and influences the design of alternative approaches, such as the use of the DNS infrastructure to advertise domain boundaries [21].

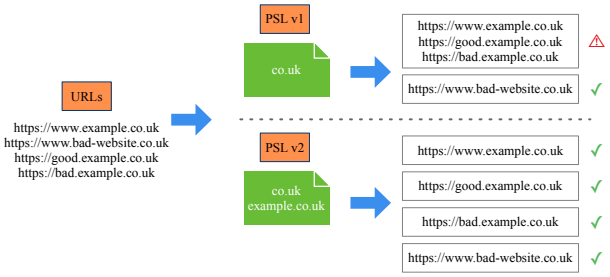

Figure 1: Illustrative example of the impact of an out-of-date Public Suffix List.

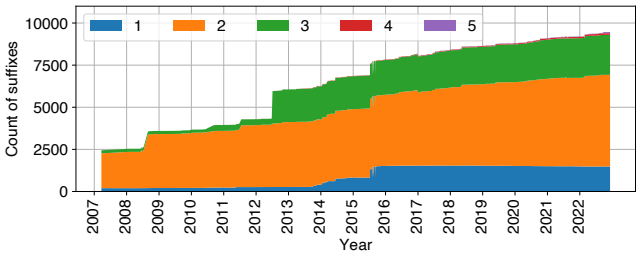

Figure 2: Growth of the Public Suffix List, and number of suffix components, over time.

2 WEB PRIVACY AND THE PSL

Web applications are affected by (at least) two different privacy boundaries: (i) the *domain*, the human readable name that, through the DNS, describes the IP address(es) of the server(s) that host the application; and (ii) the *site*, or the group of (sub-)domains that are controlled by the same entity and that together provide some service. For example, the **domains** `www.google.com`, `maps.google.com`, and `calendar.google.com` are all part of the same **site** since they're all controlled by the same organization. The domain `www.yahoo.com`, on the other hand, is controlled by a different organization and is part of a different site.

Browsers allow different web pages running on the same site to read to and write from the same state, even if they are on different domains. Code running on different sites is prevented from accessing common state, and so is, in principle, prevented from tracking a user across site boundaries.

This raises the question of how browsers can determine which domains belong to which sites. In simple cases, a heuristic such as "*the site is the first name to the left of the first period*" may work. In practice though, heuristics for mapping domain names to sites quickly fail, and the diversity of domain names (and patterns of domain names) uses make generalizing impossible. For example, the domains `amazon.co.uk` and `google.co.uk` belong to different organizations despite having the same "*name to the left of the first period*". Similarly, some organizations allow users to register their own sub-domains (e.g., `digitaloceanspaces.com`), and browsers should maintain boundaries between these.

To overcome these problems, in 2007, Mozilla initiated the Public Suffix List [5]. The public suffix list is a plain text list of *effective top level domains* (eTLDs). These are the parts of a domain name that are shared by multiple sites owned by different organizations. The PSL describes how domain names should be grouped together into sites (*i.e.,* eTLD+1s), and so fall within the same privacy boundary. Prominent examples of eTLDs include `com`, `co.uk`, and `blogspot.com`.

Any software can retrieve the PSL to identify trust boundaries between domain names, and there are several well-documented uses of the list. These include filtering "*supercookies*" (*e.g.,* attempts to set cookies across `.com`), grouping domains into sites, and finding DMARC policy records for email subdomains. Importantly, the public suffix list is maintained as a community effort on GitHub, whereby any domain owner that allows sub-domain registration by third-parties can submit name suffixes for inclusion. A new list is published several times each month, and users of the list are encouraged to periodically retrieve the latest list. Since the public suffix list is used to construct privacy boundaries, using an incorrect version of the public suffix list potentially allows users to be tracked across unrelated domains.

A common, real-world way that a web browser, or other application, could use an incorrect public suffix list is by using an out-of-date version of the list, such that it is not aware of newly added rules. For example, a browser using a version of the PSL from before `github.io` was added would not know to treat each of its sub-domains as separate sites, and would instead treat them all as a single site, allowing them to share cookies and other state. This would be a serious privacy risk, since each sub-domain can be registered and controlled by a separate organization. Figure 1 illustrates this scenario. In this example, *PSL v1* does not include the `example.co.uk` eTLD, resulting in the domains `example.co.uk`, `good.example.co.uk`, and `bad.example.co.uk` being grouped together within the same site. *PSL v2* includes this suffix, so these subdomains are appropriately separated.

Beyond preventing cross-site cookie access and tracking, another common application of the public suffix list is in grouping domains together in the user interfaces of web applications. However, this too can present a privacy risk to users. For example, consider a password manager that has stored credentials for the domain name `good.example.co.uk`. When the user visits that domain, they will be prompted by the password manager to autofill their credentials. However, if, as in the example scenario above, the password manager is using *PSL v1*, then they will also be prompted to autofill their credentials on `bad.example.co.uk`. While there is no direct sharing of data, the user interface, by prompting users and incorrectly indicating that domain names may be related, can potentially expose users to privacy harms.

3 DATASETS

The Public Suffix List. The PSL is hosted on GitHub [2] and, at the time of writing, has 1,294 commits. We extract all versions of the list, covering 1,142 versions of the list dated from 22nd March 2007 through to 20th October 2022.

Figure 2 shows the growth of the list since its creation in 2007. The list began life with 2447 entries, quickly growing to 8062 by 2017, with slower growth towards 9368 suffixes by October 2022. The plot also breaks down the suffix rules based on the number of suffix components (*i.e.,* the number of elements separated by

dots). We see that 17% of entries cover a single component, 57.5% of all entries cover two components, 25.3% have three components, and a small fraction (~0.1%) have four or more components. There are several notable spikes in growth. In mid-2012, a significant number of suffixes (~1623) are added to support 4th-level name registrations within the Japanese domain name registry, which allows for city-level registrations [7].

GitHub Repositories. To determine what type of open-source software relies on the PSL, we search `github.com` for repositories that contain the list. To do this, we make use of the Sourcegraph API [6], and perform a search for files named `public_suffix_list.dat` in public GitHub repositories. We find 273 repositories. We note the limitations of this approach in identifying projects using the public suffix list: we will not identify closed-source projects, those that aren't hosted on GitHub, or that make use of the public suffix list, but with a different filename. As a result, we identify a subset of the projects using the list.

As we will discuss in Section 5, we identify 43 projects that use the list in potentially privacy harming ways. We sought to notify the maintainers of those projects of our findings, either privately, where contact details were available, or where this was not possible, by opening a GitHub issue explaining the correct use of the public suffix list.

HTTP Archive. To characterise the privacy risk of using an old or out-of-date version of the public suffix list, we retrieve web browsing data from the HTTP Archive [3]. This dataset contains millions of URLs that are gathered from the Chrome User Experience Report [1]. We look at the 498M desktop web requests gathered in the July 2022 snapshot. We then determine the suffix of the domain name in every request using each version of the public suffix list.

IANA Root Zone Database. We split suffix entries into two categories: (*i*) top-level domains, and (*ii*) private domains. To further categorise top-level domains, we label them using the IANA Root Zone Database [4] as *generic TLDs* (*e.g.,* `.com`, `.google`), *country-code TLDs* (*e.g.,* `.uk`, `.de`), *sponsored TLDs* (*e.g.,* `.edu`, `.aero`), and *infrastructure TLDs* (*e.g.,* `.arpa`).

Reproducibility and data access. We make available our code for gathering, processing, and analyzing the data discussed in this paper. This, and our full labelled dataset of repositories that we identify as using the public suffix list, is available from https://doi.org/10.17630/50e596c3-7537-4f74-b503-e9bcc5c8b95a.

4 HOW PROJECTS INTEGRATE THE PSL

While we have discussed two common use cases of the public suffix list – managing cookies in web browsers, and prompting password autofill within sites – there is a wide range of other applications where determining the administrative boundaries between sites is necessary. This includes cosmetic uses (such as grouping domains together in the web browser UI), and validation systems (such as SSL wildcard issuance). Using the methodology described in Section 3, we find 273 GitHub repositories using the public suffix list.

We first characterise the use of the public suffix list in open-source projects hosted on GitHub. To characterise how these projects are using the list, we manually examine each of the projects, and classify them as integrating the list in one of three ways:

Category	Number of projects
Fixed (F)	68 (24.9%)
Production (Prd.)	43 (15.8%)
Test (T)	24 (8.8%)
Other (O)	1 (0.4%)
Updated (U)	35 (12.8%)
Build	24 (8.8%)
User	8 (2.9%)
Server	3 (1.1%)
Dependency (D)	170 (62.3%)
Java: `jre`	113 (41.4%)
Shell: `ddns-scripts`	15 (5.5%)
Python: `oneforall`	12 (4.4%)
Python: `python-whois`	10 (3.7%)
Ruby: `domain_name`	10 (3.7%)
Other	10 (3.7%)

Table 1: Open-source projects using the Public Suffix List by usage type.

Fixed Incorporation. One way projects incorporate the public suffix list is by hard-coding a version of the list into their code, without any mechanism for updating the list. This is, in general, the most risky way a project could integrate the list. We find that nearly 25% of the GitHub projects use the list in this way. We further classify these projects into one of three sub-categories: (i) "*production*" (15.8% of projects) denotes projects that use hard-coded, outdated versions of the list in production code (i.e.,, code that runs as part of the project's normal use); (ii) "*test*" (8.8%) denotes projects that use an outdated list as part of a test suite, and (iii) "*other*" (0.4%) denotes projects that use a fixed list for any other reason; we identify one such project, where a hard-coded list is included but not used in the codebase. Of these, "*production*" is the most privacy-harming, as users are interacting with software that is using an out-of-date version of the list. However, the other uses may indirectly expose users to privacy harms, by, for example, not highlighting mismanaged privacy boundaries during testing.

Updated Incorporation. Second, a project can include an outdated version of the public suffix list, but attempt to update the list periodically. Yet, in these cases, the project's code falls back to using the hard-coded list if the project was not able to fetch an updated copy. We find that 12.8% of relevant projects on GitHub incorporate the list in this way. We further classify projects in this category into one of three sub-categories: (i) "*build*" (8.8% of projects), meaning the project attempts to the update the list as part of its build step, and then continues using the same version of the list when the resulting application is run, (ii) "*user*" (2.9% of projects), meaning the project attempts to update the list on bootstrap, and it is intended to be restarted often (e.g., a user application), or (iii) "*server*" (1.1% of projects), meaning the project attempts to update the list on bootstrap, but it is a project that is unlikely to be restarted often (e.g., a server daemon). Naturally, these 1.1% of service projects are most at risk, as they rarely obtain updated versions.

Dependency Incorporation. Finally, projects can indirectly incorporate the public suffix list by using an open source library that incorporates the list. We observe that 62.3% of GitHub projects that use the list do so through a third-party library. Because of ambiguities in which version of which library would be used at build time, we do not classify projects by how the dependency library manages the public suffix list (i.e., fixed or updated). We instead classify projects in this category by the library used for fetching the list.

Table 1 presents a full breakdown of our taxonomy. Of the three approaches, *fixed* is the most harmful: here, projects will never include new additions to the list. However, those projects that attempt to update their lists are also exposed: these updates might fail, resulting in the use of the out-of-date versions of the list that they incorporate. In the next section, we explore ways to quantify the associated risk.

5 ESTIMATING PRIVACY HARM

Having identified projects that use the list, and characterised the nature of their usage, we next turn to estimating the privacy harm that comes from misuse of the public suffix list. We perform this estimation in three ways, looking at (i) the age of (out-of-date) incorporated lists; (ii) the popularity of projects that use the list; and (iii) how recent, real-world HTTP requests would be interpreted by the lists used by projects.

List Age. Intuitively, the more out-of-date the list is, the more severe, and more frequent, the misclassified privacy boundaries will be. Figure 3 plots the distribution of observed list ages per repository (where it can be obtained). We break down the repositories based on the update strategy they use. For example, an age of 500 indicates that a repository uses the list from $t - 500$ days ago (t = 8 December 2022, where t is when we performed our measurements).

We observe a wide range of list ages. Across all repositories, we find a median list age of 871 days. This suggests that use of extremely out-of-date lists is commonplace. For repositories that use the *updated* strategy, we find a median list age of 915 days. While this is higher than the median across all repositories, this will only have a negative impact if the automatic (*e.g.,* build or run time) update fails, and the code base falls back to the static copy of the list. Otherwise, this out-of-date version will be replaced by the latest version of the list; maintaining a more up-to-date version of the list would limit the potential harm when the automatic update fails. Of the projects with a fixed copy of the list (*i.e.,* that do not automatically update it), we find a lower median list age of 825 days. While this indicates that maintainers of these projects are at least somewhat aware that the list should be updated, it suggests that a significant fraction of repositories add a copy of the PSL and do not subsequently update it.

Github Repository Popularity. We next inspect the number of stars repositories receive as a basic proxy of their "popularity". Intuitively, more popular repositories will have more widespread privacy risks. Stars are used on GitHub to bookmark or save a reference to a repository; we argue that repositories with many stars will indicate a widely used package. As shown in Table 3, star counts strongly correlate with fork counts, another potential measure of popularity, with a Pearson correlation coefficient of 0.96 for the listed repositories.

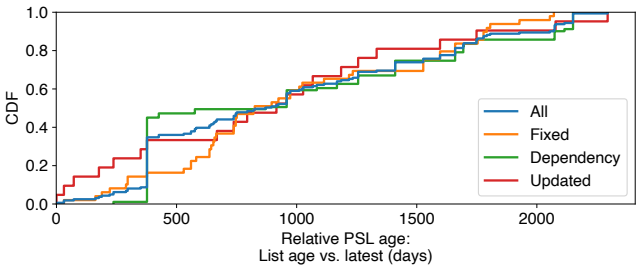

Figure 3: Age of lists stored in GitHub projects.

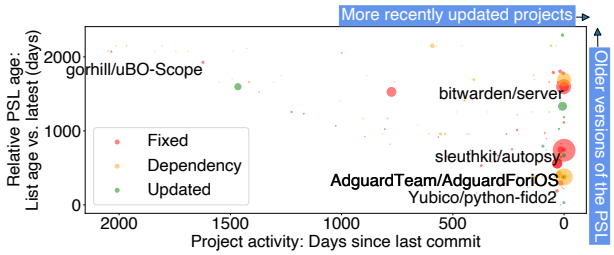

Figure 4: PSL age vs. days since last commit, sized by star count (popularity), and colored by PSL usage type, with selected *fixed* and *production* projects labelled.

Figure 4 presents a scatterplot of projects identified as using *fixed* versions of the list in *production* code, with the age of the list, activity of the project, and project popularity shown. The majority of repositories have few stars: of the repositories with fixed public suffix lists, that use this in live code, only 5 repositories have 500 or more stars, with a median of 60 stars. There are, however, a notable subset of extremely popular repositories that still rely on outdated lists. For example, the *bitwarden* suite includes two-factor authentication and password management applications. Two open-source projects that support these applications appear in our dataset: the *server* project (10,959 stars) and the *mobile* project (4,059 stars) both include hard-coded copies of the PSL. *Autopsy* (1,720 stars) is a digital forensics tool written in Java, designed for use by law enforcement agencies. It is notable that popular, actively-maintained projects, including those with a security focus, do not routinely update their copies of the PSL.

Estimating Privacy Boundary Risks. Having shown that there are popular projects using significantly out-of-date lists, we next quantify the potential privacy harms that this poses. While Figure 2 shows the growth of the list over time, it does not capture the real-world use of the suffixes that are added. If, for instance, use of the suffixes is scarce (e.g., few or no registered domain names use them) or the domain names are not actually visited by users, then the privacy harms of using out-of-date lists would be lessened. To do quantify the real-world risk, we simulate how a recent, representative sample of Web requests would be interpreted by applications using each previous version of the public suffix list.

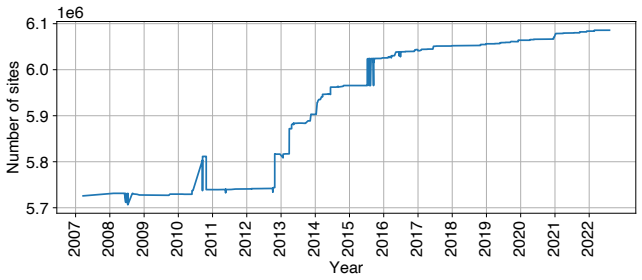

Figure 5: Number of sites formed in HTTP Archive July 2022 snapshot by different versions of PSL.

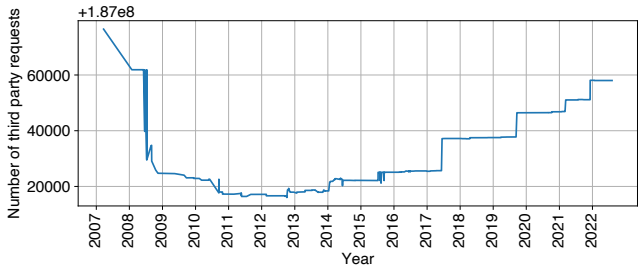

Figure 6: Number of requests that are categorised as third party by different versions of PSL.

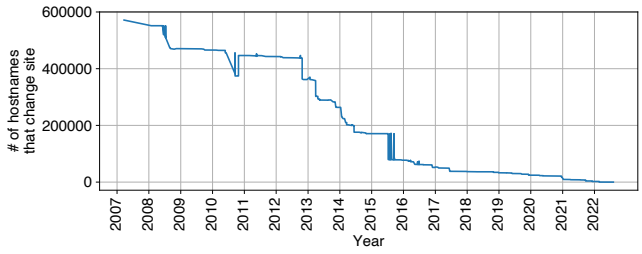

Figure 7: Number of hostnames that are in different sites vs. the most recent public suffix list.

Specifically, we use each version of the public suffix list to determine the privacy boundaries (i.e., site membership) of the hostnames requested in a recent snapshot of the HTTP Archive project. To calculate this, we extract the suffix for each unique hostname in the HTTP Archive snapshot, and then calculate the size and composition of the sites that are formed. We do this using all versions of the PSL. This shows how different privacy boundaries are constructed as different versions of the public suffix list are used, and, crucially, allows us to quantify the impact of using out-of-date lists.

To determine which URL in the HTTP Archive belongs to which site, as determined by the public suffix list, we:

(1) Strip each URL to the domain name component; for example, the URL `https://www.example.com/page.html` becomes `www.example.com`;
(2) Determine the suffix for each *unique* domain name in the dataset using each version of the PSL;
(3) Group domain names by suffix, forming *sites* (a site is sometimes known as eTLD+1).

These steps allow us to capture two metrics: (i) the number of unique *sites* contained within our HTTP Archive snapshot for a given version of the public suffix list, and (ii) the number of *domains* that make-up those sites. This allows us to measure the impact of using an out-of-date version of the list. If the HTTP Archive snapshot is evaluated using an old version of the PSL and the number of sites decreases while their size increases, this suggests that some privacy boundaries are not being preserved. For example, PSL v1 in Figure 1 creates 3 sites (with an average of 1.33 domains in each site), while PSL v2 creates 4 sites (with 1 domain in each site) – the latest version of the list produces more granular (and correct) privacy boundaries, leading to more sites, with a smaller number of domains mapped to each.

Number of Sites. Figure 5 plots the number of sites that are formed by processing the HTTP Archive snapshot with each version of the list. As shown, the number of sites found in the dataset is broadly flat in the early years of the suffix list, before growing rapidly from 2013 through 2016, and then plateauing more recently. This broadly reflects the trends seen in the development of the list (as shown in Figure 2) and follows the intuition that a greater number of suffixes leads to a greater number of sites. This confirms that use of out-of-date lists *does* lead to incorrect privacy boundaries: the latest list in our dataset creates an additional 359,966 sites, when compared to the first.

Third-Party Resources. Figure 6 shows the number of requests that are categorised as third-party, for each version of the public suffix list. These are domains that the PSL considered *outside* of the first-party organization. As the public suffix list determines the site boundaries, whether or not a request is third-party will change as those boundaries change. We see that in the early years of the list there is a significant drop in the number of requests that are categorised as third party. This is because the PSL formalizes ownership boundaries, and reduces the number of incorrectly classified third parties. However, after plateauing, this has steadily risen from 2014 through to 2022. Third-party requests are a greater privacy risk, as they are usually for resources that are shared across a significant number of third-party domains. Our results indicate a significant potential privacy harm: more requests are erroneously treated as first-party when using out-of-date lists.

Estimating Harm Based on Age. Figure 7 shows the number of hostnames in the HTTP Archive snapshot that are members of different sites in a previous version of the public suffix list, versus the most recent version.

This shows that the older a list is, the greater the number of hostnames that are mapped to the wrong site. This is measured by the number of hostnames in the HTTP Archive snapshot that are in a different site in the most recent PSL, when compared with each prior version. As shown, most of the significant suffix rules (*i.e.,* rules that caused the largest shifts) were added to the list in 2007 through 2016, with less significant shifts in more recent years. This results in part from older sites (and suffixes) having a longer time to accumulate more users and traffic.

eTLD (Hostnames)	Projects			
	D	Fixed		U
		Prd.	T/O	
myshopify.com (7848)	44	23	7	13
digitaloceanspaces.com (3359)	46	27	12	14
smushcdn.com (3337)	44	23	7	13
r.appspot.com (3194)	34	15	3	7
sp.gov.br (2024)	13	2	0	2
altervista.org (1954)	32	14	3	7
readthedocs.io (1887)	23	13	2	4
netlify.app (1278)	35	15	5	9
mg.gov.br (1153)	13	2	0	2
lpages.co (1067)	23	13	2	4
pr.gov.br (891)	13	2	0	2
web.app (871)	28	13	2	5
carrd.co (776)	28	13	2	5
rs.gov.br (747)	13	2	0	2
sc.gov.br (714)	13	2	0	2

Table 2: Largest eTLDs in the HTTP Archive snapshot that are created by subsequent rule additions to the PSL, where at least one project is labelled as *fixed* and *production* and has the rules missing. Labels as in Table 1.

Estimating Harm of Open-Source Project Use. To estimate the potential privacy harm that the open-source projects that we have identified may be exposed to by using an out-of-date suffix list, we combine our findings. Specifically, we check the number of domains in the HTTP Archive that are misclassified on the wrong site when using out-of-date lists.

Table 2 shows the largest (in terms of hostnames impacted) eTLDs in the HTTP Archive snapshot, where there is at least one project that is using a fixed version of the public suffix list in production code, and where that version does not include the eTLD. We also show the total number of projects missing the rules, based on the taxonomy discussed in Section 5: these projects do *not* correctly enforce privacy boundaries for the hostnames in the HTTP Archive snapshot. While the top 15 such eTLDs are shown, we identify 1,313 in total, affecting 50,750 hostnames. This is a significant number of eTLDs, and it includes popular services, including Shopify (`myshopify.com`), a commerce platform, and Digital Ocean spaces (`digitaloceanspaces.com`), a CDN. The impact of this depends on the application's use of the list. Password managers, for example, might incorrectly suggest autofilling a password on domains that are operated by different organizations; similarly, web browsers may allow cookies and other state to be set and read inappropriately. More broadly, applications using out-of-date lists will misinterpret hostnames as being under the same administrative control, when the latest version of the public suffix list explicitly indicates that this is not the case. While we do not attempt to categorise the domains that are incorrectly mapped to the same site, many of the missing eTLDs that we identify (e.g., Digital Ocean spaces) allow for the arbitrary hosting of content: projects using lists that do not have these suffixes are exposed to significant potential privacy harms.

6 RELATED WORK

The public suffix list is one of many lists used to define and enforce privacy boundaries on the Web. Another category of privacy-affecting lists used on the Web are filter lists, used by ad-blockers and other content filtering tools. Filter lists are extremely popular on the Web, and have been the target of a wide range of research, including studies by Garimella et al. [12], Pujol et al. [19], Merzdovniket al. [17], Gervaiset al. [13], Liet al. [15], and Zarraset al. [22]. They found that crowd-sourced filter lists positively (and significantly) improve the privacy, security, and performance of Web browsers. Our findings support this, highlighting similar benefits from using an up-to-date public suffix list. Our work differs in that we focus on the public suffix list, and, to the best of our knowledge, we are the first to characterize it and the risks of using out-of-date versions of the list.

A large amount of existing work has explored the trade-offs between different privacy boundaries on the Web. An enormous amount of work (e.g., [8–11, 14, 18]) has documented the privacy harm of the profile-as-boundary approach to managing privacy on the Web, largely starting with foundational studies by Mayeret al. [16] and Roesneret al. [20]. We contribute to the wider space by exploring how out-of-date public suffix list usage can also negatively impact the definition of appropriate privacy boundaries.

7 CONCLUSION

This paper has investigated the use of the public suffix list in open-source software. The use of out-of-date lists is common: only 12.8% of projects we identify use a version that is routinely updated; and 24.9% include a hard-coded and outdated version, with a median age of 825 days. Vulnerable software includes popular, security-orientated projects, like *Bitwarden* and *Autopsy*.

We estimate the potential privacy harm that results from projects using these out-of-date lists by interpreting a recent snapshot of Web requests through each version of the public suffix list. Using data from the HTTP Archive, we found that those projects that use hard-coded, outdated versions of the list construct incorrect privacy boundaries for 1313 eTLDs, affecting 50,750 domains. This includes for services, like Digital Ocean spaces (`digitaloceanspaces.com`), a content delivery network, that allow their users to host arbitrary content, resulting in significant potential privacy harms.

Our results indicate that application developers need to be more aware of the privacy implications of using out-of-date versions of the public suffix list. As the risks that we have identified are inherent to any list-based approach to the definition of Web privacy boundaries, we hope that our findings will raise awareness about the safe use of such lists, and motivate the design of alternative solutions (*e.g.,* integrating boundaries within the DNS infrastructure [21]).

ACKNOWLEDGEMENTS

This work was supported in part by the UK Engineering and Physical Sciences Research Council under grant EP/S036075/1. In order to meet institutional and research funder open access requirements, the authors have applied a CC BY public copyright licence to any Author Accepted Manuscript version arising.

REFERENCES

[1] Chrome UX report. https://web.dev/chrome-ux-report.
[2] GitHub: Public suffix list. https://github.com/publicsuffix/list.
[3] HTTP Archive. https://httparchive.org.
[4] IANA: Root zone database. https://www.iana.org/domains/root/db.
[5] Public suffix list. https://publicsuffix.org.
[6] Sourcegraph. https://sourcegraph.com.
[7] Technical bylaws on attribute type (organization type type), regional type jp domain name registration, etc. https://jprs.jp/doc/rule/saisoku-1.html.
[8] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juarez, Arvind Narayanan, and Claudia Diaz. The web never forgets: Persistent tracking mechanisms in the wild. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 674–689, 2014.
[9] Claude Castelluccia, Lukasz Olejnik, and Tran Minh-Dung. Selling off privacy at auction. In *Network and Distributed System Security Symposium (NDSS)*, 2014.
[10] Steven Englehardt and Arvind Narayanan. Online tracking: A 1-million-site measurement and analysis. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 1388–1401, 2016.
[11] Imane Fouad, Nataliia Bielova, Arnaud Legout, and Natasa Sarafijanovic-Djukic. Missed by filter lists: Detecting unknown third-party trackers with invisible pixels. In *PETS*, 2020.
[12] Kiran Garimella, Orestis Kostakis, and Michael Mathioudakis. Ad-blocking: A study on performance, privacy and counter-measures. In *Proceedings of the 2017 ACM on Web Science Conference*, pages 259–262, 2017.
[13] Arthur Gervais, Alexandros Filios, Vincent Lenders, and Srdjan Capkun. Quantifying web adblocker privacy. In *EuroSys*, pages 21–42. Springer, 2017.
[14] Tai-Ching Li, Huy Hang, Michalis Faloutsos, and Petros Efstathopoulos. Trackadvisor: Taking back browsing privacy from third-party trackers. In *International Conference on Passive and Active Network Measurement*, pages 277–289. Springer, 2015.
[15] Zhou Li, Kehuan Zhang, Yinglian Xie, Fang Yu, and XiaoFeng Wang. Knowing your enemy: understanding and detecting malicious web advertising. In *CCS*, pages 674–686, 2012.
[16] Jonathan R Mayer and John C Mitchell. Third-party web tracking: Policy and technology. In *2012 IEEE symposium on security and privacy*, pages 413–427. IEEE, 2012.
[17] Georg Merzdovnik, Markus Huber, Damjan Buhov, Nick Nikiforakis, Sebastian Neuner, Martin Schmiedecker, and Edgar Weippl. Block me if you can: A large-scale study of tracker-blocking tools. In *EuroS&P*, pages 319–333. IEEE, 2017.
[18] Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos Markatos. Cookie synchronization: Everything you always wanted to know but were afraid to ask. In *The World Wide Web Conference*, pages 1432–1442, 2019.
[19] Enric Pujol, Oliver Hohlfeld, and Anja Feldmann. Annoyed users: Ads and ad-block usage in the wild. In *Proceedings of the 2015 Internet Measurement Conference*, pages 93–106, 2015.
[20] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. Detecting and defending against third-party tracking on the web. In *Presented as part of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, pages 155–168, 2012.
[21] Andrew Sullivan, Jeff Hodges, and John R. Levine. DBOUND: DNS Administrative Boundaries Problem Statement. Internet-Draft draft-sullivan-dbound-problem-statement-02, Internet Engineering Task Force, February 2016. Work in Progress.
[22] Apostolis Zarras, Alexandros Kapravelos, Gianluca Stringhini, Thorsten Holz, Christopher Kruegel, and Giovanni Vigna. The dark alleys of madison avenue: Understanding malicious advertisements. In *IMC*, pages 373–380, 2014.

A PROJECTS USING THE PUBLIC SUFFIX LIST

Table 3 lists the GitHub projects that we have identified as using fixed versions of the public suffix list, where the age of the list can be obtained. The list shows the name, star count, and fork count for each repository, alongside the age of the list (vs. 8th December 2022), and the number of hostnames that are missing from the list (vs. the HTTP Archive snapshot discussed in Section 5).

Repository name	Star count	Fork count	List age (days)	# of missing hostnames
Production				
bitwarden/server	10959	1087	1596	36326
bitwarden/mobile	4059	635	1596	36326
sleuthkit/autopsy	1720	561	746	21494
alkacon/opencms-core	473	384	1778	36936
firewalla/firewalla	434	117	746	21494
SAP/SapMachine	397	79	376	3966
Yubico/python-fido2	324	102	188	1
gorhill/uBO-Scope	222	20	1927	37739
fgont/ipv6toolkit	222	66	1791	36966
LeFroid/Viper-Browser	164	22	529	8166
Keeper-Security/Commander	145	67	1113	27685
nabeelio/phpvms	134	116	644	9228
coreruleset/ftw	104	36	750	21576
gorhill/publicsuffixlist.js	79	12	289	2236
Twi1ight/TSpider	68	21	2070	49581
j3ssie/go-auxs	60	22	664	9236
Intsights/PyDomainExtractor	59	5	31	0
alterakey/trueseeing	47	13	296	2249
BenWiederhake/domain-word	40	3	1233	30080
timlib/webXray	27	22	1659	36329
mecsa/mecsa-st	20	6	1659	36329
amphp/artax	20	4	2054	49197
dicekeys/dicekeys-app-typescript	15	4	825	21729
netarchivesuite/netarchivesuite	14	22	1778	36936
mallardduck/php-whois-client	11	3	657	9232
kee-org/keevault2	10	4	895	21961
AdaptedAS/url_parser	9	3	924	21970
h-j-13/WHOISpy	9	3	1527	36307
oaplatform/oap	9	7	1527	36307
amphp/http-client-cookies	7	5	162	1
hrbrmstr/psl	6	2	1753	36933
szepeviktor/unique-email-address	6	0	819	21675
WebCuratorTool/webcurator	6	4	973	22977
Test				
ClickHouse/ClickHouse	26127	5725	737	21494
win-acme/win-acme	4620	770	560	8178
yasserg/crawler4j	4336	1923	1527	36307
jeremykendall/php-domain-parser	1021	121	296	2249
rockdaboot/wget2	365	61	1805	36988
DNS-OARC/dsc	94	23	1010	24294
rushmorem/publicsuffix	90	17	636	9164
park-manager/park-manager	49	7	653	9229
addr-rs/addr	40	11	636	9164
datablade-io/daisy	32	7	737	21494
elliotwutingfeng/go-fasttld	10	3	221	4
m2osw/libtld	9	3	581	8178
Komposten/public_suffix	8	2	1217	29974
Other				
du5/gfwlist	29	16	1023	24298

Table 3: Open-source projects identified as having *fixed* usage of the public suffix list.