M³AAWG | Messaging Malware Mobile Anti-Abuse Working Group

Messaging, Malware and Mobile Anti-Abuse Working Group

M³AAWG Present and Future of the Public Suffix List

April 2023

Executive Summary

Though not well-known, the Public Suffix List (PSL)[1] is incorporated as a fundamental resource into some of the internet's most popular applications, including virtually all web browsers. The PSL includes the names of all the domains under which new (private) domains can be directly registered. When the PSL gets incorporated into applications, it is often used for security-sensitive purposes, including critical security decisions around domain ownership boundaries. As such, the PSL plays a vital and security-sensitive role for the internet.

The PSL is maintained on GitHub by a tiny (1–2 person) team of individual volunteers on a best-efforts basis. These volunteers have no service level agreement (SLA) and no formal contracts. Their work is focused on reviewing submissions, so planning for sustainment and succession is not a priority. If the PSL were to cease being maintained (perhaps as a result of volunteer burnout, health issues, legal action, or some other unforeseeable catastrophic event), the impact of its abandonment would be quite disruptive and could potentially affect the stability and security of the entire internet ecosystem.

This document describes the PSL, explains its current strengths and limitations, and outlines some possible future enhancements. Most importantly, though, the community must step up and help to make sure it continues to exist.

I. Background

The Public Suffix List (PSL) is a hand-curated list of top-level domains (TLDs) and "effective top-level domains" (eTLDs) under which new domains can be directly registered within the ICP-3[2] namespace. As of the date this document was written, the PSL contained 9,087 entries. Understanding the importance of TLD/eTLD boundaries requires a quick review of some domain name system (DNS) terms of art:

- **Domain names** consist of **labels** separated by dots. (In **www.example.com**, for example, **www**, **example**, and **com** are all labels).
- **TLDs** are the **rightmost label** of a domain. Familiar examples include **com**, **net**, and **org**. Most often, new domain names will be registered directly under a TLD – but not always.
- Sometimes new domain names may be registered under a more-hierarchical **effective top-level domain** (**eTLD**). Effective top-level domains **act like** top-level domains, but they have more than one label. A few examples of effective TLDs:

[1] https://publicsuffix.org/list/
[2] https://www.icann.org/resources/pages/unique-authoritative-root-2012-02-25-en

M³AAWG
Messaging, Malware and Mobile Anti-Abuse Working Group
P.O. Box 9125, Brea, CA 92822 USA ■ www.m3aawg.org

• Some countries allow registrations in geographic effective top-level domains such as **bj.cn** for Beijing, China, or **on.ca** for Ontario, Canada.

• In Oregon, primary and secondary schools often register domains under the **k12.or.us** effective top-level domain. For example, Springfield Public Schools uses springfield.**k12.or.us.**

• In the UK, companies will often register domains under the **co.uk** effective top-level domain. Examples of this include bbc.**co.uk** (the UK news company) and dege-skinner.**co.uk** (Dege & Skinner, a bespoke Savile Row tailor).

• Other examples of eTLDs include Mongolia's **gov.mn** and **edu.mn** eTLDs, as used by police.**gov.mn** and www.num.**edu.mn** (the National University of Mongolia).

Effective TLDs may be created by commercial entities. For example:

• Dynamic DNS providers may allow users on dynamically-assigned broadband IP addresses to register convenience hostnames (dyndns.org, freeddns.org, etc.).

• Cloud application providers (such as blogspot.com, github.io, etc.) have numerous different customers each with fully qualified domain names (FQDNs) under the same domain suffix.

• Cloud infrastructure providers such as Amazon and Azure, among others, place multiple customers on shared domains.

• Consumer network device providers may create domains for firewall devices, distributed backup systems, consumer NFS devices, etc. (e.g., myfritz.net).

• Content distribution networks (CDNs) such as Fastly and Cloudflare, among others, may place different CDN customers on the same domain suffix.

• Some vendors may offer or sell subdomains of well-known TLDs (such as domains under br.com, cn.com, de.com, eu.com).

• Even some governmental suffixes are used as effective TLDs (e.g., homeoffice.gov.uk).

The point is that many diverse enterprises may rely on effective TLDs, not just on single-label true TLDs.

Effective TLDs can be helpful in that they tend to promote a hierarchical domain taxonomy, almost like a phylogenetic tree in the life sciences. Rather than just dumping everything into a single, monolithic flat namespace, a hierarchical organization can make it easier to distinguish between private companies, schools, government agencies, and so forth.

At the same time, eTLDs can also add complexity, including raising some fundamental questions: Where can someone register new domains? Where does one customer's domain (and subdomains) stop and a second customer's start?

You might assume that there must be some algorithm we could use to address that fundamental question, but you'd be wrong. Figuring out where domains can get registered in the DNS is a matter of consulting a manually curated list of suffixes. In most cases, the list that ends up getting checked is the Public Suffix List.

The Public Suffix List of TLDs and eTLDs identifies domain authority boundaries, which are used in some cases to indicate security and trust relationships. If those boundaries and trust relations aren't accurately delineated, you might end up with inadvertent access to someone else's domains or data – or they might end up with inadvertent access to yours.

SSL/TLS Certificate Issuance

For instance, consider wildcard SSL/TLS certificates. A domain owner can buy a wildcard SSL/TLS certificate covering all the hosts under their registered domain. For example, the Springfield Public Schools might purchase a certificate covering *.**springfield.k12.or.us**.

Every certificate authority follows its own unique process for certificate scoping review (within the guidelines established by the Certificate and Browser Forum[3]). Without the Public Suffix List as one guide point, however, the chances increase that a certificate authority might accidentally consider issuing an overly inclusive wildcard cert, such as a wildcard certificate covering *all* ***.k12.or.us** (that is, a certificate covering numerous K12 schools in Oregon), or – even worse – a wildcard certificate for *all* ***.or.us** (e.g., *all* domains registered under **or.us,** including private companies, schools, government agencies, and so on).

If this were to occur, it could enable widespread impersonation attacks and would be a huge potential security problem.

Web Authentication Cookies

A more subtle example of a security vulnerability involves web authentication cookies. You don't want web authentication cookies to be able to be set or read by unintended parties. Continuing with the k12.or.us example, establishing a **k12.or.us** entry in the Public Suffix List helps ensure that someone from one school district (perhaps the Salem Keizer Public schools, using hosts in salkeiz.**k12.or.us**) won't be able to set or read cookies from domains used by another school district (such as the Springfield Public Schools, using hosts in springfield.**k12.or.us**). The Public Suffix List helps keep those divisions of control – and much more – properly scoped.

These are just a few of the many reasons why the PSL is a critical security control mechanism, and why any proposed changes to the PSL are very carefully reviewed by the PSL maintainer team before they are deployed.

II. Format of the PSL

The PSL is freely available as a plain text file currently containing 9,087 (non-blank-line, non-comment) entries. Anyone can download a copy of the PSL using their web browser (or a Unix command-line client such as wget[4]). This format and distribution model is simple, but has some inherent limitations:

[3] https://cabforum.org/
[4] https://www.gnu.org/software/wget/

M³AAWG
Messaging, Malware and Mobile Anti-Abuse Working Group
P.O. Box 9125, Brea, CA 92822 USA ■ www.m3aawg.org

Scaling to Internet-Size Audiences While Ensuring Data Currency

The internet has billions of users. It is unsustainable for even a tiny fraction of them to routinely directly download a copy of the PSL.[5] (We've seen similar problems for other flat file download paradigms.[6])

As a result, many significant third-party projects that use the PSL work around this by (a) downloading one copy of the PSL and then (b) "embedding" and sharing that information as part of their code or compiled application. Doing this, however, means that those third-party software projects may potentially have out-of-date copies of the PSL hard-coded within their code or application unless a new and updated release of the third-party software project is generated whenever the PSL is updated. Some third parties include a downloaded copy of the PSL in the app directory and read it in at launch, allowing users to self-update their local file, but this is also challenging to keep current.

Simple Format

The PSL is just a plain text file with a very simple format. There's no clear way of associating attributes with listings (as one could if the PSL were to use a real database, or even just a structured, machine-readable format such as JSON Lines[7] format). The PSL also only supports four types of entries:

- **Comments** (e.g., lines beginning with //), such as:

  // gr : https://grweb.ics.forth.gr/english/1617-B-2005.html

  // Submitted by registry <segred@ics.forth.gr>

- **Exact match entries** (containing a TLD or eTLD, one per line), such as:

  gr

  com.gr

  edu.gr

  net.gr

  org.gr

  gov.gr

  blogspot.gr

  simplesite.gr

- **Wildcard entries** (containing a domain name with a left-hand leading asterisk, one per line), such as:

  *.kawasaki.jp

  *.kitakyushu.jp

  *.kobe.jp

  *.nagoya.jp

[5] This is sometimes referred to as the "hosts file" problem, referring to the way host name information was shared in the early days of the internet prior to the creation of the Domain Name System. See, for example, https://en.wikipedia.org/wiki/Hosts_(file).

[6] "EasyList is in trouble and so are many ad blockers," *Adguard*, October 2022 https://adguard.com/en/blog/easylist-filter-problem-help.html

[7] https://jsonlines.org/

*.sapporo.jp

*.sendai.jp

*.yokohama.jp

- **Exception entries** (containing a domain name prefixed with an exclamation point, one per line), such as:

!city.kawasaki.jp

!city.kitakyushu.jp

!city.kobe.jp

!city.nagoya.jp

!city.sapporo.jp

!city.sendai.jp

!city.yokohama.jp

That's a very simple syntax to parse and support[8] – which is good – but that simple format means that some entries which could potentially be very elegantly and succinctly expressed as regular expressions[9] might instead need to be explicitly enumerated, thereby increasing the size of the PSL file.

It should be noted that there is diversity to how the wildcard and exception entries are interpreted (or even used) by those who integrate, parse, or otherwise utilize the PSL.

III. Competing PSL Uses and Design Objectives

The needs and objectives of PSL users vary widely.

- **Accuracy.** Some users prioritize having absolutely accurate public suffix data (even for complex, little-used, or short-lived eTLDs). These users might accept a higher level of "churn" – or a much-larger-size PSL file – if that's required in order to get to a current and perfectly accurate list of public suffixes.

- **Compactness**. Other users may be space-constrained. These users may prefer a compact version of the PSL that is "close enough" for the most popular eTLDs, even if some "corner cases" (or new/short-lived eTLDs) may be imperfectly handled. Some of these users may be incorporating copies of the PSL into popular products where keeping program size modest is a top need.

- **File stability**. A third category of users may seek PSL file stability. These users may have a painful local review and vetting process, or may incur substantial costs every time they have to build and distribute a newly-updated copy of a product that incorporates the PSL. Keeping those PSL users happy may be best accomplished by keeping the PSL stable and only updating it when it's absolutely essential.

- **Ease of parsing**. As previously mentioned, the PSL's format is quite simple, and thus easily parsed. If we were willing to forgo some of that simplicity, and perhaps begin supporting full regular expressions, we might be able to simultaneously improve the fidelity of the PSL while also reducing its size. Supporting regular expressions in diverse access libraries, however, is not a trivial uplift.

---

[8]Over 30 popular programming languages have libraries to consume the PSL, see https://publicsuffix.org/learn/

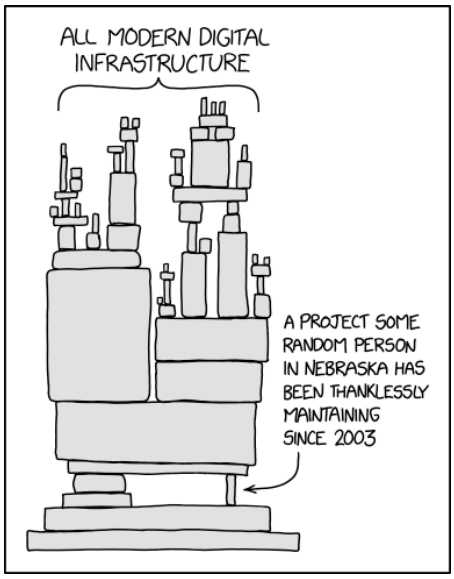
[9]https://en.wikipedia.org/wiki/Regular_expression

• **Access model considerations**. It's hard to get much simpler than using a central plain text file that users can download over the web. If we're willing to consider a more sophisticated data access model, we might be able to make the leap from a centralized distribution model to a distributed one (akin to going from the old "hosts" file model to today's DNS).

• **Web Browser and General Application Performance**: The primary use case, where the PSL originated and evolved as a web standard, saw web browsers enjoying improved performance by locally consulting the PSL instead of checking a central registry over the internet. The PSL has evolved beyond just web use, as exemplified by applications' growing utilization of the PSL for similar latency benefit considerations. This trend should not be overlooked, as it weighs heavily on why the PSL's existing format has not evolved into the DNS or other solutions.

IV. The Public Suffix List as a Precarious Dependency

The PSL is hosted on a CDN via Google Cloud by the Mozilla Foundation. However, the PSL is maintained by a very small team of individual volunteers, working on a best-efforts basis, with no service level agreement (SLA), no formal contract, and no planning for sustainment or succession. Its continued existence is precarious, as captured in this XKCD:



Source: https://xkcd.com/2347

That cartoon was inspired by the once-precarious and unsupported status of OpenSSL.[10] The PSL is another example of that sort of precarious existence.

Internet users all appreciate the PSL volunteer maintainers' efforts (and they have done a great job to date, for which the present writers are eternally grateful), but critical infrastructure (and the PSL should be considered critical infrastructure) **must be *sustainable* and *resilient***. This implies that the PSL needs sustainable funding and support from a foundation (or from multiple businesses whose products rely on the PSL).

[10] "Tech giants, chastened by Heartbleed, finally agree to fund OpenSSL," *Ars Technica*, April 2014, https://arstechnica.com/information-technology/2014/04/tech-giants chastened-by-heartbleed-finally-agree-to-fund-openssl/

There also needs to be succession planning, so that when current volunteers become tired of selflessly volunteering, or get ill, or retire, there will be someone ready to step up and assume responsibility for maintaining the PSL. We cannot allow the PSL, which is now a critical component in many internet systems and applications, to potentially become an abandoned open-source project.

V. What M³AAWG-Related Entities Should Do

- If you use the PSL, make sure your copy is *current*, and that someone is responsible for keeping it that way.
- If you are responsible for a TLD or major eTLD, review your PSL entry(ies) and make sure it is properly represented. If it isn't, review the submission guidelines and submit an appropriately formatted pull request along with the DNS-based submitter validation records to update your entry or entries.
- If you are someone who does protocol development, please consider engaging with the IETF DBOUND effort.[11] DBOUND is engaged in considering alternatives to the current non-scalable, simple-flat-file-download-from-the-web model.
- If you are in a position to help the PSL, whether in terms of working as a volunteer or ensuring the PSL is organizationally sustainable, please do what you can! M³AAWG hopes to see the following headline appear prominently in an industry publication someday soon: "Industry coalition to advise, support, rally sponsors, and define a path forward for the Public Suffix List."

[11] https://www.ietf.org/mailman/listinfo/dbound

As with all documents that we publish, please check the M3AAWG website (www.m3aawg.org) for updates.