SAC070

SSAC Advisory on the Use of Static TLD / Suffix Lists

An Advisory from the ICANN Security and Stability Advisory Committee (SSAC)
28 May 2015

Preface

This is an Advisory to the ICANN Board, the ICANN community, and the Internet community more broadly from the ICANN Security and Stability Advisory Committee (SSAC) on **the use of static TLD/suffix lists in applications**.

The SSAC focuses on matters relating to the security and integrity of the Internet's naming and address allocation systems. This includes operational matters (e.g., pertaining to the correct and reliable operation of the root zone publication system), administrative matters (e.g., pertaining to address allocation and Internet number assignment), and registration matters (e.g., pertaining to registry and registrar services). SSAC engages in ongoing threat assessment and risk analysis of the Internet naming and address allocation services to assess where the principal threats to stability and security lie, and advises the ICANN community accordingly. The SSAC has no authority to regulate, enforce, or adjudicate. Those functions belong to other parties, and the advice offered here should be evaluated on its merits.

A list of the contributors to this Advisory, references to SSAC members' biographies and disclosures of interest, and individual SSAC members' withdrawals and dissents with respect to the findings or recommendations in this Advisory are at the end of this document.

Table of Contents

Executive Summary

There is no uniform consensus definition of what is a public suffix. For the purposes of this Advisory, a public suffix is defined as "a domain under which multiple parties that are unaffiliated with the owner of the Public Suffix domain may register subdomains."[1] Examples of Public Suffix domains include "org", "co.uk", "k12.wa.us" and "uk.com".

There is no programmatic way to determine the boundary where a Domain Name System (DNS) label changes stewardship from a public suffix, yet tracking the boundary accurately is critically important for security, privacy, and usability issues in many modern systems and applications, such as web browsers. One method of determining this boundary is by use of public suffix lists (PSLs), which are static files listing the known public suffixes.

This advisory investigates the security and stability needs surrounding the growing use of PSLs on the Internet. In this Advisory, the Security and Stability Advisory Committee (SSAC) takes Mozilla's PSL as an archetype to study the current landscape. The SSAC finds varied uses of the Mozilla list. From this case study the SSAC derives various potential difficulties in two general areas: the content of a PSL generally; and operational and administrative concerns surrounding use and maintenance of a PSL.

It is important to note that this Advisory is not intended as a criticism of Mozilla or other PSL providers. The Mozilla volunteers are to be commended for successfully performing a vital service to the Internet community with no formal responsibility compelling them to do so.

The security and stability impacts of current PSL practices stem from the fact that while PSLs are now in the critical path for many Internet experiences and functions, including security controls, there is no broadly consistent application, accountability, or implementation for PSLs.

Specifically, the SSAC finds:

- The PSL is by its very nature a compromise between convenience of use and accuracy of its contents.

- There is no consensus definition of "public suffix" and associated terms, and in fact the PSL is used for several purposes having to do with administrative boundaries in the DNS.

- There is a lack of accountability mechanisms for ensuring PSLs are produced in a consistent, fair, unbiased manner with recourse for individuals or organizations that may have an issue.

[1] See https://tools.ietf.org/html/draft-pettersen-subtld-structure-10.

- A knowledge gap exists between registries and maintainers of the public suffix lists regarding the processes and responsibilities for changes and additions to the Mozilla PSL and other PSLs.

- There is no universal library, framework, tool, or mechanism for PSL use. Further, implementers do not use PSL entries consistently in software or other services. Registries cannot expect similar behavior across all devices or applications for their suffixes. Such behaviors contribute to an unstable user experience.

- There is great variation of latency for implementing PSL changes in software applications and Internet services. The update and distribution cycle for changes to entries in a PSL impact the usability and acceptance of new top level domains (TLDs) and/or policies in TLDs.

- There is a general lack of authentication and other standard security controls for the content and transmission of PSLs from maintainers to users.

- Due to the wide variety of use cases for PSLs, it may be difficult to create a one-size-fits-all PSL for all audiences covering any application or usage.

- If the new generic top-level domains (gTLDs) use public suffixes similarly to the existing generic TLDs, where typically there is one public suffix because the entire TLD is "public," there would be limited impact to the size of a PSL. However, if new gTLDs use public suffixes similarly to some country code TLD (ccTLDs), which may include more than one public subdomain, the impact to any PSL could be significant.

The SSAC makes the following recommendations in this advisory:

The SSAC first calls on the Internet Engineering Task Force (IETF) and application community to directly address these fundamental design compromises by designing, standardizing and adopting alternative solutions (see Recommendation 1). Second, because use of PSLs today is prevalent, and noting the time it takes for the IETF to standardize alternative solutions and the community to deploy them, the SSAC recommends a set of near-term measures to alleviate some of the higher risk issues with the current maintenance and use of PSLs (Recommendations 2-6).

1. Recognizing that alternatives to the PSL have been discussed (see Appendix A), the SSAC recommends the IETF and the applications community consider them for further specification and standardization through the IETF process.

2. The IETF should develop a consensus definition of "public suffix" and other associated terminology (e.g. "private suffix").

3. To close the knowledge gap between registry operators and popular PSL maintainers, ICANN and the Mozilla Foundation should collaboratively create

informational material that can be given to TLD registry operators about the Mozilla PSL.

4. The Internet Community should standardize the current approach to PSLs. Specifically:

    a. ICANN, as part of its initiatives on universal acceptance, should encourage the software development community (including the open source community) to develop and distribute programming and operating system libraries implementing robust (i.e. authenticated, timely, secure, accountable) distribution mechanisms for PSLs;

    b. Application developers should use a canonical file format and modern authentication protocols as specifications to this work;

    c. Application developers should also replace proprietary PSLs with well-known and widely accepted PSL implementations such as the Mozilla PSL and the proposed Internet Assigned Numbers Authority (IANA) PSL (Recommendation 5).

5. IANA should host a PSL containing information about the domains within the registries with which IANA has direct communication. Such a PSL at a minimum should include all TLDs in the IANA root zone and would be authoritative for those domains.

6. ICANN should explicitly include use and actions related to a PSL as part of the work related to universal acceptance of domain names.[2]

[2] See https://www.icann.org/resources/pages/universal-acceptance-2012-02-25-en.

1 Introduction

The Domain Name System (DNS) is a distributed system for hierarchically assigning names to Internet resources, such as Internet Protocol (IP) addresses, so those resources may be accessed using human-readable names, rather than numerical addresses. A Public Suffix List (PSL) is the result of an effort to identify DNS names that represent public namespace, from which administration of sub-namespace is delegated to registering entities. The maintenance of such a list is meant to aid with web security, privacy, and policy, as well as add convenience to a number of processes and tools in other applications and services. The best-known PSL is operated by volunteers in collaboration with the Mozilla Foundation.

It is important to note that this report is not intended as a criticism of Mozilla or other PSL providers. The Mozilla volunteers are to be commended for successfully performing a vital service to the Internet community with no formal responsibility compelling them to do so.

Questions of the effectiveness and scalability of the use of static top-level domain (TLD) lists or the Mozilla PSL have been raised, especially with the increased frequency of the creation of new generic TLDs (gTLDs). In this advisory the Security and Stability Advisory Committee (SSAC) outlines the known use cases of a PSL; inspects the scalability of the current Mozilla PSL system; describes the potential security, stability and usability concerns; and makes recommendations to Internet Corporation for Assigned Names and Numbers (ICANN) and the Internet community to improve the service.

This Advisory takes the PSL maintained by the Mozilla Foundation as an archetype of efforts by the Internet community to maintain and disseminate a list of public suffixes. There are many derivative or similar static lists of suffixes that may be proprietary or privately operated. This Advisory is also intended to inform those operating any such public suffix lists, and may contain important and potentially valuable considerations for privately maintained lists of suffixes other than the Mozilla PSL.

This Advisory aims to inform the following audiences:

- Members of the ICANN community who are unaware of the issues surrounding PSL definitions and implementations that are affecting them (registries of TLDs, owners of domains in TLDs being prominent examples). This Advisory also aims to provide educational content to this community.
- Technologists and policy makers in standards and coordination bodies (IETF, World Wide Web Consortium (W3C,) ICANN, others) that are looking to address the inconsistencies and confusion created by the issues the SSAC has outlined here. This Advisory thus is similar to an IETF “problem statement”, but it also includes policy aspects of a PSL. For example the IETF does not typically address who decides what goes into a PSL and an accountable and sustainable process for those decisions.

- Software and service vendors that may be using a variety of public suffix and static TLD lists as part of their software/service. This Advisory also aims to identify issues with such usages.
- General Internet users who may be confused about why different answers, displays, effects and other responses are given for the same string depending on the situation, software, operating system, or service used.

2 Background and Terminology

Usage of a PSL is intimately tied to other Internet technologies. The following section provides an introduction to the relevant Internet technologies.

2.1 DNS

DNS *names* consist of a series of dot-separated labels, which describe the name's DNS ancestry[3] (Strictly speaking it's a sequence of nodes in the namespace tree). For example the ancestry of `foo.bar.baz` is: `foo.bar.baz`, then `bar.baz`, then `baz` (top-level domain or TLD), and finally the (implicit) *root* domain. Top-down delegation of subdomain space, beginning at the root domain, is the behavior enabling the necessary global system properties that distinguish the DNS, such as distributed authority and global uniqueness of names. A *zone* is an autonomously managed portion of namespace within the DNS. When domain namespace is delegated to a child zone, all subdomain namespace belongs to that child zone, unless explicitly further delegated to another zone, by the administrator of the child zone.

2.2 Public Suffix and Public Suffix Lists

There is no uniform consensus definition of what is a public suffix. For the purposes of this advisory, a public suffix is defined as "a domain under which multiple parties that are unaffiliated with the owner of the Public Suffix domain may register subdomains." Examples of Public Suffix domains include "`org`", "`co.uk`", "`k12.wa.us`" and "`uk.com`".

In general, TLDs (e.g., `com`) are used almost exclusively for delegating subdomain space to entities registering the domains (e.g., `example.com`), and commonly do not include hostnames directly in their zone, thus these TLDs are by definition public suffixes.

Many country-code TLDs (ccTLDs) (e.g., `uk`) and some gTLDs allow a delegation behavior, in which hostnames under certain second-level (e.g., `gov.uk`) or even third-level domains (e.g., `k12.pa.us`) can be registered and delegated to other entities, creating an "effective TLD" under their two-letter country code or gTLD name. These second- or third-level domains are public suffixes as well.

[3] Technically, the DNS name space is a tree, "The domain name of a node is the list of the labels on the path from the node to the root of the tree," per RFC 1034 Sec. 3.1, and the dots are a presentation convention.

A PSL is a static text file that lists all (or a subset) of the known public suffixes. The Mozilla Foundation maintains the most well-known PSL. The IANA TLD list can also be considered a PSL by this definition, as are a variety of TLD / PSL lists tailored for various applications. Some of these PSL lists are publicly available, but many are embedded in code or software configurations without user visibility or control.

Public suffix lists exist and are regularly updated because:

- There is currently no programmatic way to determine public suffixes for DNS entries with more than one label from a registered multi-level domain, yet tracking public suffix definition and use accurately is critically important for security, privacy, and usability issues in many modern systems and applications.
- Over time, new TLDs are added to the root zone and new suffixes are added to (and removed from) existing TLDs.
- Within a TLD the rules determining administrative boundaries may change over time. For example, in the UK policy for registrations directly under `.uk` has changed. This is especially likely as business models for new gTLDs evolve.
- TLDs are gradually introducing internationalized domain names (IDN). IDNs may lead to new public suffixes that take advantage of the expanded list of characters allowed in IDNs.

3 Use Cases for Public Suffix Lists

Using the Mozilla suffix list as an archetype, this section lists some common use cases for public suffix lists.

3.1 Setting Cookies

The question of reliably identifying effective TLDs affects the privacy and security of Hypertext Transfer Protocol (HTTP) cookies. An over-privileged cookie is the primary example, in which the Web server at `foo.bar.example` sends a cookie with a domain value of `bar.example`, which happens to be a public suffix. Without a public suffix list, the browser does not know that `bar.example` is a public suffix, and sends the cookie in subsequent requests to any other host in `bar.example`, not only to other hosts solely within the `foo.bar.example` domain. The browser typically sends these cookies without explicit user consent or action, yet cookies allow sensitive state information, including browsing history and login sessions, to be made known across independent entities. Any such sensitive information in such an over-privileged cookie will be sent to servers that are almost certainly not authorized by the user to view the information, creating significant security and privacy risks.

The public suffix list minimizes the potential for Web servers to inadvertently (or intentionally) set an over-privileged cookie. This security goal was the primary driver for developers of Mozilla Firefox who began the cross-browser effort in 2006 to develop a

list of public suffixes used in cookie policy.[4] Currently Firefox, Chromium/Chrome, Safari, and Opera use the Mozilla PSL to determine cookie settings. Internet Explorer will use the Mozilla PSL as well, starting with Windows 10.[5]

3.2 Highlighting Domains / Sorting Browser Histories

Some browsers use the Mozilla PSL to determine the highest (i.e., hierarchical) privately registered label of the hostname in a Uniform Resource Locator (URL) and highlight it in the address bar of the browser. This highlighting is an effort to reduce the social engineering effectiveness of such names as the unwieldy example: `www.victim-label.adhggj. fsddsaf.adfd.attacker.example`, which tempts users to think the relevant label is "`www.victim-label`" when it is in fact "`attacker.example`".

3.3 Use as Navigability Shortcuts

Google Chrome and Safari use the Mozilla PSL to determine whether text entered into the address bar of its browser is a hostname or a search term. For example, a term of "com" will be treated as a search for the phrase "com", because the term does not resolve to a registered domain (as it is a public suffix). However, the term for "foo.com" is treated as navigation, because it does contain a registered domain ("foo.com")

3.4 Restricting the Issuance of Wildcard Certificates

The Certification Authority (CA)/Browser Forum baseline requirements (11.1.3) require that before issuing a wildcard certificate, Certificate Authorities ensure that such a certificate is not issued for entries in the Mozilla PSL, e.g. `*.co.uk`, or that the entity actually owns the entirety of the public suffix.

3.5 Validating Top-level Domains

Numerous programming languages and web applications use a PSL or static TLD list to validate form entries or logic determining validity of TLDs, for example in user-submitted URLs or email addresses.

3.6 Anti-Spam

To reduce processing time, many mail gateways and/or spam filters review the rightmost label of the sender/return address for validity as a basic check using a match in a PSL as an initial pass/fail measure.

The Domain Based Message Authentication, Reporting and Conformance (DMARC) draft Request For Comments (RFC) uses the Mozilla PSL to determine the

[4] Much discussion leading to the first list of effective TLDs is documented in comments from a 2006 Bugzilla bug report for Mozilla Firefox.

[5] See http://blogs.msdn.com/b/ie/archive/2014/10/06/interoperable-top-level-domain-name-parsing-comes-to-ie.aspx.

"organizational domain". This is where the DMARC algorithm looks for DNS records relating to DMARC. DMARC is one of the use cases driving the IETF effort to find a programmatic or protocol-based way to determine such organizational boundaries.

Table 1 below summarizes the various use cases for the PSLs.

Table 1: Uses cases of PSL (adapted from https://wiki.mozilla.org/Public_Suffix_List/Use_Cases)

Use Case	Description	Question	Example Applications
Cookie-Setting	Deciding whether a cookie should be allowed to be set for a suffix of a given domain	Are this domain and its suffix controlled by the same entity?	Mozilla Firefox, Google Chrome, Safari, Opera
'Responsible Domain' Highlighting/ Browser History Sorting	Deciding which parts of a domain to highlight or sort on in a UI - "Public Suffix + 1"	Are this domain and its suffix controlled by the same entity?	Mozilla Firefox
Navigability	Deciding whether a browser should attempt to navigate to a given URL without consulting DNS	Is there (likely to be) an A record for this domain?	Google Chrome
Secure Sockets Layer (SSL) Wildcards	Deciding whether to issue or accept an SSL wildcard certificate for `*.public.suffix`.	Are the servers of this domain and its suffix operated by the same entity?	Certificate Authorities
TLD Validation	Numerous programming languages use the PSL to validate form entries, or in logic determining the validity of TLDs in domain names generated in various ways.	Does this human-generated URL submitted have a valid TLD?	Web forms, programming language libraries
Anti-Spam	The TLD in a domain name is reviewed for validity on return/sender addresses as a basic check, using match in PSL as initial pass/fail to reduce processing time.	Should I quickly drop this email FROM: <perp@scam.example> if .example isn't in the list of TLDs?	Anti-spam software

4 Issues Concerning the Generation and Maintenance of a PSL

The following sections cover concerns the SSAC has that are specific to generating and maintaining the content of a PSL. These are generic concerns that apply independent of the intended use of the PSL.

4.1 Lack of Consensus Over the Definition of "Public Suffix"

Differences of opinion exist over what a "public suffix" is, and a variety of stakeholders base their definitions upon their precise need or solution. Although there are many that express with vigor an authoritative position about appropriate definitions, there is variation and inconsistency.

One clear area of delineation that has been made in the Mozilla Public Suffix List has been to split the horizon of the list between two areas: Public/ICANN and Private. In Mozilla's definition:

- The Public/ICANN section includes gTLD and ccTLD suffixes entries that comply with Internet Coordination Policy-3 (ICP-3)[6] and are directly delegated by IANA or are associated to them.
- The *private* section includes entries from many subdomain registration services[7] such as CentralNic (owner of e.g. eu.com and us.org), as well as companies such as DynDNS, Amazon, Google, GitHub, Heroku[8], Microsoft and Red Hat, who provide DNS resolution and cloud services. This section exists because some registered domain owners wish to delegate subdomains to parties with no relationship to each other.

Although such categorization may be necessary and helpful for improving decision making, it raises the following issues both for the volunteer community that maintains the Mozilla PSL and its consumers:

- *Registry status confusion*. The public/private demarcation in Mozilla PSL is not clear enough that a third party can discern the difference without the markings. The presence of the suffix in the Mozilla PSL, even though it is marked "private", might imply that it has the equivalent status of a registry directly delegated by IANA or with clear transfers in administrative authority from IANA and the TLD.
- *Potential consumer misunderstanding of trust relationships*. People may infer trust based on the structure of DNS itself. One might be able to argue in the ICANN domain section of the Mozilla PSL such trust has some validity as the

[6] See https://www.icann.org/resources/pages/unique-authoritative-root-2012-02-25-en.

[7] Defined by Anti-Phishing Working Group, a subdomain registration service is a provider that gives customers subdomain "hosting accounts" beneath a domain name that the provider owns. See http://docs.apwg.org/reports/APWG_Global_Phishing_Report_1H_2014.pdf.

[8] Hiroku Website Security note https://devcenter.heroku.com/articles/cookies-and-herokuapp-com.

operator responsible for setting the policy for the TLDs presumably also sets policies for the Second Level Domains (SLDs) in the PSL and one can expect there are at least some consistency (e.g. `com.au` and `.au`, `gov.cn` and `.cn`). However, such trust may be misplaced with the private domains section, as these often are not operated by or in relationship with the registry for the TLD. An example of this is that `.de` (the ccTLD for Germany) is operated by DENIC, but `com.de` is operated by CentralNic, an entity not affiliated with DENIC. Since the Mozilla PSL has both of them in one file, this distinction may not be clear. The opposite situation may occur if an entity were to use the IANA list of TLDs as the basis for trust relationships, as many registries, particularly ccTLDs, subdivide their zones and don't use the top level at all as a legitimate public suffix portion of a registered domain. For example no domain should be registered under `.au` (Australia), which is in the IANA list of TLDs, but rather under `com.au`, `net.au`, `org.au`, etc. This is where the limits on the Mozilla concept of "public" or "private" as a full description of administrative relationships become clear.

- *Confusion of public and private suffixes*. Most applications use both ICANN Domains and Private Domains without distinction. However, sometimes there is a need to distinguish these two, and the fact that these two sections are listed in a single file make such a distinction difficult. For example, The DMARC specification[9] uses the Mozilla PSL to determine the "organizational domain". This is where the DMARC algorithm looks for DNS records relating to DMARC policy. This usage should probably exclude the private entries in the Mozilla PSL, but the DMARC specification does not currently say that it should. Having these two sections together requires implementers to have the knowledge of the Mozilla PSL in order to make these decisions, which may or may not be true per implementer.

The lack of a consistent industry-wide definition of "public suffix" contributes to such confusion. People do not understand what a public or private suffix is, the differences between the two, or who has the right to update each type. It's also entirely possible that finer distinctions are required regarding administrative boundaries than simply "public" vs. "private" for some use cases.

4.2 Lack of Accountability for Operators of PSLs

Since there is no universally agreed-upon definition of a PSL, no official standards, and no body with formal standing publishing PSLs, there are no accountability mechanisms for ensuring PSLs are produced in a consistent, fair, unbiased manner with recourse for individuals or organizations that may have an issue. One notable exception to that is IANA with respect to its very limited PSL list. The Mozilla PSL could be considered accountable to some extent given its public nature and fairly transparent process.

[9] See https://www.rfc-editor.org/rfc/rfc7489.txt.

Other PSL maintainers are typically private companies supporting their own software, services, or systems. To a certain extent, market pressures will provide some accountability and incentive to maintain accurate PSL information in their products. However relying on large-scale complaints or major interoperability problems to induce change in a private PSL introduces many risks and does not support the full community of public suffix applicants, who may not provide enough usage or a compelling case for some PSL producers to incur costs to incorporate them. Thus goals like universal acceptance of TLDs are difficult to achieve in such a diverse environment.

In the volunteer-maintained Mozilla PSL example, no major security, fairness, or process issues have arisen, and the Mozilla PSL has achieved the widest adoption as a publicly maintained PSL despite lacking a formal accountability process. However, regardless of this successful track record, risks remain with the lack of oversight, accountability, and policy process inherent in the nature of the organization. Some of these risks include:

- *Capture by corporate or government entity.* Any volunteer organization without formal or limited membership/leadership could be subverted by a determined effort. With control, that entity could make decisions to list or de-list suffixes to further its own vested interests rather than the interests of all Internet stakeholders. Such self-interested decisions could include not allowing the inclusion of suffixes of rival companies or censoring suffixes to which it is philosophically opposed. If there were concerns about subversion or list policy, people could choose to forgo the current open source project to establish something more trustworthy or applicable to their needs. The operational and programming costs or educational effort required to abandon the Mozilla PSL in this way have not been studied but can be assumed to be non-trivial, thus there would be a "switching cost" to consider when weighing alternatives.
- *Continuity.* If a major portion of the volunteers were to leave the effort, without a strong successor plan, the operation itself may not be sustainable.
- *Introduction of errors or malicious entries.* While not a problem to date, a lack of overall accountability framework or oversight makes it more difficult to enforce best practices to address errors or to stand-up security/review processes to ensure a determined adversary is not able to insert a malicious change.
- *Legal standing to allow companies and organizations to rely upon the list/process.* With no formal organization to work with, no published review, appeals process, or stakeholder involvement, entities may be precluded by their risk management teams from utilizing a PSL provided by a volunteer group such as the Mozilla PSL team. Some corporations have decided to maintain their own PSLs and this may be a key factor in their decisions, which leads to lack of uniformity in PSL entries across the ecosystem.

4.3 Knowledge Gap for Adding Entries to a PSL

There is often a knowledge gap between registries and maintainers of a PSL regarding the processes and responsibilities for adding entries. This gap has been evidenced even with

the well-publicized Mozilla PSL, and likely is much bigger for other PSLs, particularly private ones.

PSLs rely on a requestor to submit entries to the PSL maintainer, or for the maintainer of the PSL to determine that new PSL entries should be added based on new TLDs or new TLD registry policies. If registries do not request an update to a PSL entry, their TLDs or entries would not be added in a timely manner or at all, unless the PSL maintainer is particularly diligent. Browsers and/or other applications that rely on a PSL do not recognize an un-added public suffix or TLD. However, the registry managers often do not know where to submit their request, even for the Mozilla PSL. This situation is exacerbated further when there are many versions of PSLs maintained by a variety of organizations or individuals.

Besides registry managers, individual domain owners and software reporters can also submit entries to some PSLs. In these cases, the PSL maintainer needs to verify the submission with the entity or organization responsible for the relevant public suffix. However, the PSL maintainer often does not have existing relationships with these entities. Establishing such relationships, and maintaining trust takes time. If registries (or entities) responsible do not answer the validating questions for the request, further delay is introduced.

Anecdotal evidence suggests that these gaps do exist. For example:

- The gTLD `.post` was entered into the DNS root in August 2012 and the first second-level domain names went live in October 2012. However, `.post` was not entered into the Mozilla PSL until April 2013. The reason for the delay is that the registry operator did not submit them.

- The ccTLDs `.sx` and `.cw` (for Sint Maarten and Curacao respectively) were delegated in October 2011. Second-level domains in these TLDs have been live since at least July 2012. However the TLDs were not added to the Mozilla PSL until February 2013. The reason for the delay is that the registry operator did not submit them.

- According to Mozilla volunteers, a significant factor for TLDs not being entered into its PSL in 2010 and 2011 was the fact that some ccTLD operators did not respond to requests for verification from Mozilla.

4.4 Latencies in Adding Entries to PSL

Beyond the education issues of requesters understanding that PSLs exist and understanding their policies for adding or editing entries, as outlined in Section 4.3, the maintainers of PSLs may have processes that introduce delays for updating PSL entries. For example, the Mozilla PSL is subject to time donated by volunteers. Other PSLs are subject to the release schedule of software or software libraries where they are maintained if the PSL is deployed as a static file. Other PSL operators may provide online updates, but again, may have other priorities that delay updating such resources.

According to analysis performed by the SSAC, from January 2008 to July 2014 there were 172 confirmed and resolved requests to update the Mozilla PSL.[10] The median time it took to resolve a request is 23.9 days, with 75 percent of the requests resolved within 77.6 days, and 90 percent of the request resolved within five months. Table 3 displays detailed statistics.

Table 3: Statistics on processing speed of the PSL (days)

	Times to resolve the request (days)
Mean	65.9
Median	23.9
Min	0.0
Max	885.9
25% percentile	3.4
50%	23.9
75%	77.6
90%	155.7

The cause of such delays could be combination of the following factors:

- Knowledge gap for adding entries to PSL as explained in section 4.3
- Lack of time from Mozilla PSL volunteers to process the request

Similar issues have been seen with other software reliant on various PSLs but data on such delays is more difficult to obtain due to the private nature of most such PSLs.

4.5 Formats of PSL Entries and Files

There are no standards body guidelines, RFCs or industry standard for what a PSL should look like or how it should be stored and distributed. Distribution issues will be covered in a subsequent section. There are several possible list types, such as a static file, live database, central repository, or distributed system solution. The actual entries into a PSL present interesting challenges as well, as different policies for different levels of the DNS "tree," or depending upon specific zones, are difficult to do consistently. Single entries for each suffix seem straightforward, and would work if all possible siblings and parents at a given level of the tree had similar policies. That is not the case however, causing the need to represent "exceptions" and variances.

[10] Prior to 2012 there existed an IDN whitelist inclusion request within the same ticket profile, which included an extensive review of code points published by the registry, in the interests of reduction of homograph domains. SSAC analysis manually separates the IDN whitelist requests from PSL requests to keep data comparisons consistent.

The closest thing to a standard is arguably the Mozilla PSL, which is a static file accessible via the Internet. The current format of the Mozilla PSL supports simple entries (e.g., `bar.example`), wildcard entries (e.g., `*.bar.example`), and exceptions to wildcards (e.g., `!brown.bar.example`). IDNs are among the entries, and are encoded as UTF-8[11] (U-LABEL) in the file. This format has worked thus far, but the Mozilla PSL may benefit from a more specialized or optimized format.

Beyond formatting, other meta-data would be useful for various use cases. For example, timestamps, times to live (TTLs) or other indicators of the provenance of each entry in a PSL would be beneficial for applications like caching or version control. Similarly validation of authenticity and integrity could be useful for security applications. Although there is a version control system within the process of generation of the Mozilla PSL, the file itself lacks indicators of version or generation date in a clear, human-readable or machine-readable manner, making it difficult to know the freshness of the data in hand.

4.6 Inclusion of Private Namespaces in a PSL

What suffixes are included on a public suffix list deserves close scrutiny, as suffixes for non-public namespaces[12] or alternative root TLDs can be included, depending upon the policy of the list maintainer. As a best practice, any widely deployed PSL should support the long-term security and stability of the Internet ecosystem and not introduce ambiguity or confusion. As a useful guideline, ICP-3 describes the importance of adhering to an Internet identifier structure with a unique, authoritative root.[13] For example, the Mozilla PSL complies with ICP-3[14] and in this way the Mozilla PSL supports the stability of the Internet.

If a PSL were to ignore ICP-3, it could lead to conflicting and mutually exclusive decisions on which suffixes were public and which were not. This situation could lead to indeterminate behavior within applications or between applications on the same machine using different PSLs, particularly when implemented on networks with their own private namespaces.

The content of a PSL also needs to consider a consensus definition of what a "public suffix" is. This concern is discussed in detail in Section 4.1; the definition must, of course, inform the content of a PSL.

---

[11] U from Universal Coded Character Set + Transformation Format—8-bit. See: http://en.wikipedia.org/wiki/UTF-8.

[12] For example, a private/corporate domain space is only known to name servers within the corporate network.

[13] ICANN. A Unique, Authoritative Root for the DNS. ICP-3. July 2001. https://www.icann.org/resources/pages/unique-authoritative-root-2012-02-25-en

[14] For more information, please see wiki.mozilla.org/Public_Suffix_List, and the actual list at http://publicsuffix.org/list/effective_tld_names.dat.

5 Issues Concerning the Use of a PSL

The following issues concern the technical usage of a PSL. The SSAC finds concerns in the implementation, latency, content, and variable use cases of modern PSLs.

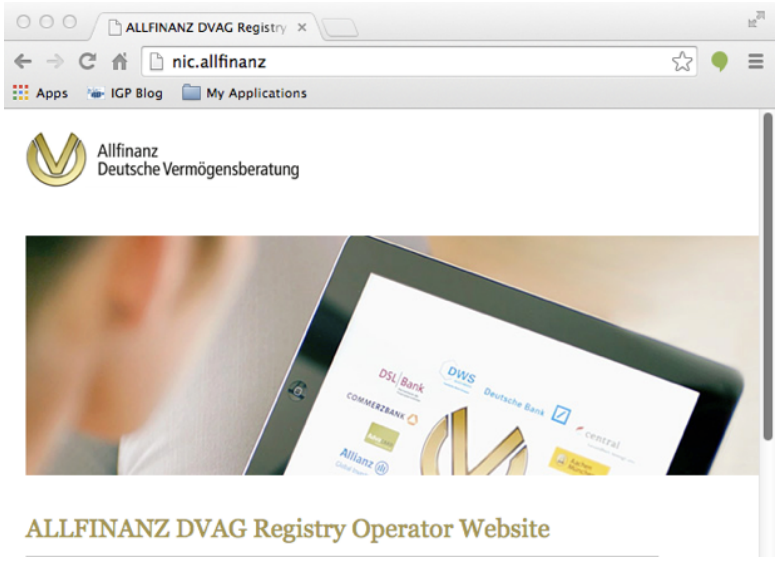
5.1 Inconsistent Suffix List Use and Processing

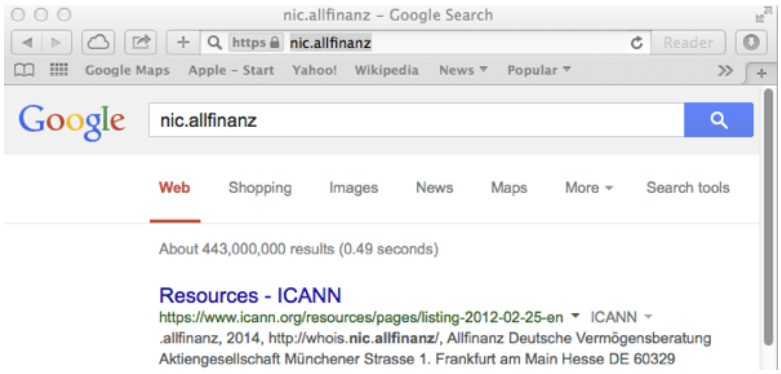
Different applications on the same computer may have different results for the user since different applications do not use any sort of unified library, standard, file structure, or methodology for determining suffixes. This is a fundamental problem that has arisen in other specific features needed for reliable Internet functionality. In these other instances Internet standards were usually created and continue to be updated. Such efforts have not been codified for PSLs; software developers, web services, and other parties must make decisions on how to create, maintain, update, and use PSLs.

While many applications have adopted the Mozilla PSL, not all actually use it the same way. It is also likely that different software packages on a given computer will use completely different PSLs, making software interoperability difficult to ensure or debug. For example, Microsoft has announced it will transition to using the Mozilla PSL starting in IE 11 in Windows 10. However older versions will continue to use the current Microsoft private PSL and PSL format. This transition is a positive development, however the transition period will evidence all the above problems in variable list content, updating, and structure until all IE 10 and earlier deployments are upgraded.

In time the IETF DBound Working Group (see Appendix A) may be able to improve on this situation, but there is no immediate prospect of fixing it.

The following screenshots of a new gTLD domain name (nic.allfinanz) rendered by Google Chrome (version 37.0.2062.124) and Safari (version 7.0.6 9537.78.2) highlight this difference.





Such inconsistency in handling PSL processing can create user confusion and reduce the usability of new gTLDs, IDNs, private suffix spaces or anything else reliant upon a PSL.

5.2 Latency of Implementing PSL Changes in Software Applications and Internet Services

As an example to illustrate issues with latency of PSL changes in software applications, web services, and other tools that use a PSL, we consider the Mozilla PSL. Currently, the Mozilla PSL relies on a bug reporting system to handle requests for updates combined with an ad-hoc publishing schedule via the web, which introduces at least three types of delay. These challenges are representative of those any text-based, manual PSL would face. The first type, propagation from bug report to code commit, is covered in section 4.2 as a delay inherent in creating the PSL. In addition to updating the content of the PSL, there are two potential delay factors:

- *Propagation from web publication to adoption of the new Mozilla PSL in software applications where PSL updates is performed independently by the software.* Software varies in the way it may update configurations to incorporate updates to the Mozilla PSL. Typically software will poll for updated configurations of a resource like a PSL, as there is no mechanism to push updates out to all clients. Latency is affected by factors such as the update-polling interval built into the software and access to the Internet or the source repository for the PSL. Access could be restricted due to local policy (e.g. firewall rules) or lack of connectivity to the Internet (e.g. mobile device roaming).
- *Propagation from web publication to downstream software.* Software products or services that rely heavily upon accurate PSLs, for example web browsers, but do not provide a polling mechanism built into the deployed software, will typically rely on a rapid update cycle. In these cases, updates include changes to configurations of items like approved Certificate Authorities or PSLs. For example, Google Chrome and Mozilla Firefox regularly release an updated version at least every six weeks. Other browsers such as Safari can take longer for the changes of the PSL to be reflected in configurations. Due to the high number of security updates many browsers release, the update cycle for PSLs may be shortened. An important liability is that such updates are dependent upon the end-user enabling update functionality or manually upgrading their software.

Many third-party PSL-based software libraries may not include dynamic update capabilities nor frequent update cycles as browsers do, and continue to use older copies of the Mozilla PSL (or whatever PSL they incorporate). Unless a manual update is performed on such software products, they may continue to have an outdated copy indefinitely. This lack of upgrade path is the largest problem identified with PSL updates. Most application developers do not release updates based solely on updates to the PSL they utilize, including those leveraging the Mozilla PSL. Thus, unless the application automatically updates user settings where it is installed by querying a public PSL source like the Mozilla PSL, updates to the PSLs used by applications will rely solely upon update cycles for the software itself including bug fixes and upgrades. For many users this may mean that they rarely, if ever, get updates to the PSLs used in their applications, particularly if they do not update regularly or must follow a long approval process before adopting new versions of software.

A slow pace was acceptable historically, as TLDs only changed infrequently. However multiple factors make modern change more frequent: increased pace of the introduction of more TLDs from ICANN's new gTLD program, additions of IDNs, further subdivision of ccTLDs, and continued growth in effective TLDs in private namespaces such as those maintained by vendors of assorted cloud services. This increased rate of change poses a challenge for Mozilla volunteers and software developers, and ultimately impacts usability and acceptance of new TLDs and/or changed policies in existing ones.

At the time of this writing, the Mozilla PSL is being downloaded approximately one million times a day. It is not directly clear from this data how often the same entity is accessing the file, which makes prediction of future use difficult.

There are other issues outside applications that utilize the Mozilla PSL for changes to the public top-level namespace. Users of applications that utilize a "custom" PSL are fully reliant on those software developers to maintain an up-to-date list of TLDs along with policies of all registries as to which child domains should be considered public suffixes. Even for large developers (e.g. Microsoft updating Internet Explorer[15]) this is a challenge, but they persist as it is necessary to satisfy their extensive user bases. For other applications it may pose an insurmountable challenge.

5.3 Authentication of PSL Contents

As with any data disseminated across the Internet, it is not enough for the user to have a readable, timely PSL; the user must have a correct and authentic PSL. Since a PSL is often used in security-sensitive decisions, it is important for the file or the entries to be authenticated before use. Otherwise it could become a target for subversion. Some PSLs secure distribution using standard technologies, such as Transport Layer Security (TLS), and this works well for transit directly from the provider of the PSL. In addition, authentication measures should be considered for protecting PSL content while it is at rest and for distribution not directly from the PSL provider, as these are also standard

[15] See http://blogs.msdn.com/b/ie/archive/2014/10/06/interoperable-top-level-domain-name-parsing-comes-to-ie.aspx

security risks for data required to successfully negotiate the Internet. This is particularly relevant in the case of the distributor, as subversion of that data scales rapidly to all users of that PSL. Usual best practices for information authentication, such as digital signatures, should be sufficient if integrated successfully. The authentication issue is compounded by the lack of agreement on PSL format or processing, as discussed in section 4.5. Ideally, an authentication solution would be integrated into these other solutions.

5.4 Different Use Cases of PSLs

People who create PSLs typically do so to solve a particular problem where knowing what is and is not a public suffix is important. Once built, such a PSL potentially can be used to address other issues. For example, although the original intent of the Mozilla PSL was to solve cookie issues for browsers, it has evolved to be used by many entities to solve many problems. These different use cases have different requirements for which a part of the Mozilla PSL may be used. Section 3 lists some of the common use cases. While this works fine for many use cases, different threat models may argue for different types of PSL listings.

- For cookies and CAs, the threat is that the adversary's domain will really be under different management from the target domain, but he'll pretend it's the same. A simple example would be `foo.example` asking for a certificate or cookie for `*.example` or `.example` (note this is a toy example as the CA is forbidden from granting a certificate for *.TLD but could if they were acting maliciously).

- For mail, the threat is that the adversary's domain is really under the same management but claims to be different. An example would be spam purporting to be from `abc.xyz.bigbank.example` but the DMARC record is at `bigbank.example`.

Depending on one's threat model, one would prefer a scheme that failed open (allowed access to continue) or failed closed (blocked access) when a PSL lookup fails. Due to these different preferences based on intended use case, it may be difficult to unify a one-size-fits-all PSL process for a wider audience and any application or usage.

A complicating issue is that different protocols have different administrative boundaries meaning that PSL content may need to be protocol specific. Current PSLs may not be able to accommodate such variety in uses. For example, `.name` has a use case whereby one could have *firstname.lastname.name* as the domain but *firstname@lastname.name* as the email, so there is a different separation of policy control between the namespace used for mail and the namespace used for web and other applications. Such a discrepancy would cause problems with, for example, the DMARC standard.

6 Architectural Considerations

PSLs represent a convenience for software engineers. They encapsulate in a single resource a summary of information that is not conveniently available elsewhere: namely, an enumeration of public suffixes in the DNS and the associated semantics of organizational boundaries within the label hierarchy of the DNS. However, reliance upon such a summary carries with it certain risks.

This approach has led to inevitable shortcomings. As third parties maintain these lists they are challenged to ensure that the lists are accurate, complete and authoritative. As described in Section 4, these shortcomings include the lack of a common precise semantic definition of a "public suffix", the lack of accountability in the production of such lists, the incompleteness of these lists, the differences in format and encroachment of private name spaces into these lists in certain cases.

In addition, these lists are not maintained by the entities that are responsible for the operation of the policy settings of the names enumerated in the list. They represent a number of third party commentaries on the contents of the DNS. Such a third-party attempt inherently suffers certain weaknesses when attempting to glean comprehensive policy information from the DNS.

A further consideration is the changing nature of the public suffix space of the DNS itself. At a time when the numbers of public suffixes were growing slowly, and changes to the policy settings of existing public suffixes were infrequent, progressive maintenance of a public suffix list could only improve the accuracy and completeness of the list. The completeness and accuracy of this list was essentially a static target.

This environment has now irrevocably changed. The growing list of TLDs, and the policy changes to these domains in order to match their offerings to customer demand, implies that the public suffix set is no longer even approximately static. The continuing lag of the PSLs in tracking the underlying public suffix set raises legitimate questions relating to universal acceptance issues with the more recent public suffixes, and the potential for unintended use of cookies or unintended scope of domain name certificates as a consequence of this mismatch.

This raises the question as to whether such lists could ever achieve these essential objectives of accuracy, completeness, timeliness and authority given their inherent shortcomings. It is difficult to see how these objectives could be achieved without making considerable changes to the way in which this information is defined and produced. Due to variable design requirements, even if these issues were solved, it may not be possible to maintain this information conveniently in a single source.

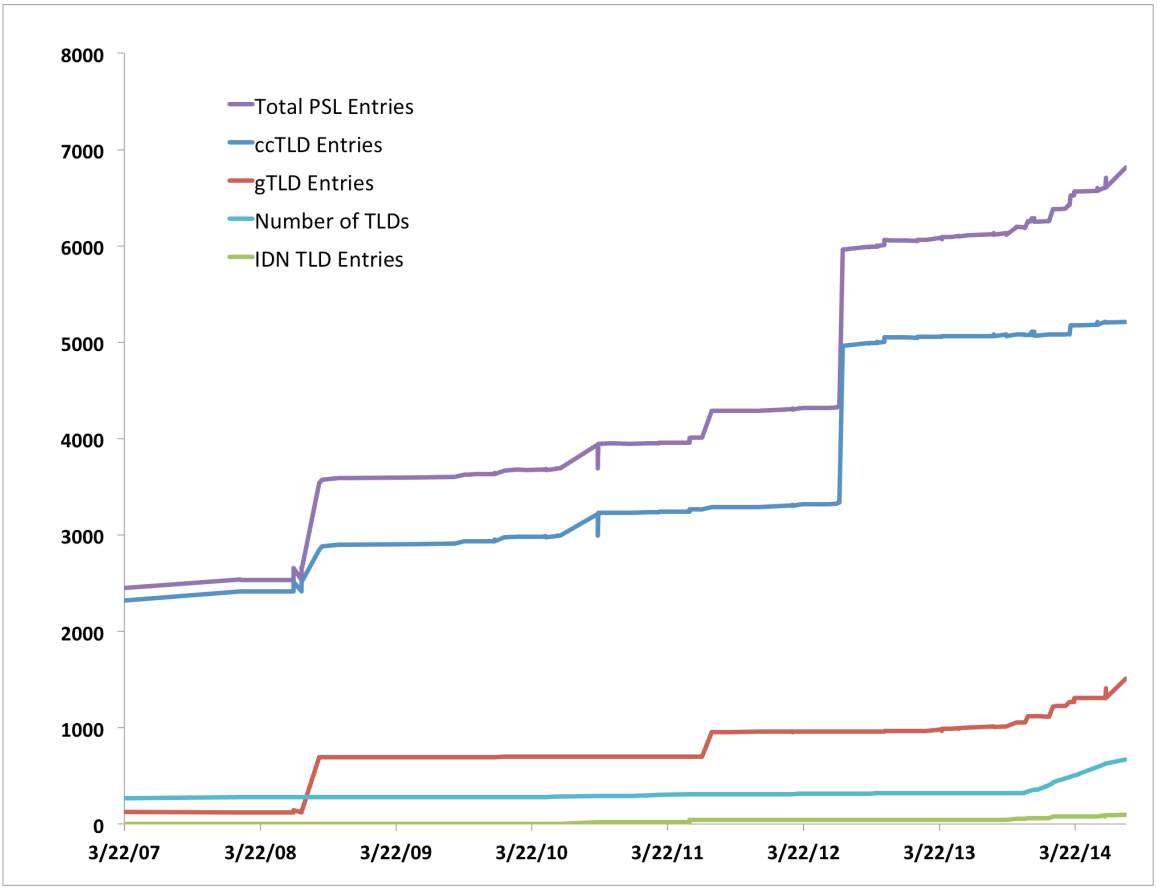
If a manual PSL inevitably suffers from such shortcomings, and represents at best a partial view of the DNS, then a PSL may not be able to achieve the necessary goals of its use. In particular, it is not clear that the Internet community should rely on such a PSL for security functions knowing its limitations.

Conflating two distinct concepts is often unwise. The concept of a PSL may conflate the boundary point of the public to private name spaces and the enumeration of public DNS suffixes into a single summary. As noted in Table 1, there are cases where the incorrect assumption of organizational boundaries in the DNS hierarchy can lead to information compromise, and third parties are forced into the position of making some form of assumption as to the location of such boundaries in the DNS. It may be a more prudent measure that such information as organizational boundaries in the DNS should be maintained and published by the entity responsible for the zone. This strategy eliminates a third-party errors, liability, and delays, however it is not without its own costs.

7 Scalability Issues with New gTLDs

As of 11 August 2014, the Mozilla PSL is a text file containing 6,763 entries. Over time, the number of gTLD entries has increased from just over 100 to about 1000 in 2014. However, most gTLDs in the Mozilla PSL (457) only have one entry: the TLD itself. The median number of entries for current gTLDs is 1, and the 75th percentile of PSL entries is 8.5, even though the mean is 43 because of the outliers. Overall gTLD entries in the PSL seem to increase linearly, and gTLD entries are only 1/5 of the PSL entries, as displayed in Figure 1 below.

Figure 1: Mozilla PSL Entries by Categories over time



From these historical statistics, one could conclude that the addition of gTLDs produces limited overhead for the Mozilla PSL in terms of content, as they are initially flat and presumably only offer second-level registration. However, there are several pending Registry Service Evaluation Requests that could significantly change this ratio. The most prominent example is Atgron, Inc, (operator of the new `.wed` registry) who has made a Request[16] to ICANN to offer third-level domain name registrations. In this scheme, the registry would reserve 11,000 second-level domains and offer third-level registrations. Other registries have indicated similar desires as well. Indeed, many of the applicants in the new gTLD program have indicated that they will operate their registries in ways more consistent with ccTLD spaces, particularly those focused on specific geographies. Some ccTLDs have shown a propensity to add numerous publicly usable sub-delegations to their primary TLD, so historical trends in gTLD expansions may not hold going forward.

The existing `.name` registry offers a large number of third level registrations under a very large number of surname second levels, though the registry has not put forth requests for these to be added to the Mozilla PSL. Such voluminous changes could increase PSL length significantly.

Taken together, these trends may indicate a PSL that is an order of magnitude or more larger than the current one. Large, static files like this may introduce performance issues into software that requires quick, low-overhead transactions with their governing PSL. The current situation is akin to the early days of the Internet, where all domain entries were ensconced in the "hosts.txt" file in the operating system. DNS was developed as a way to handle this problem, distributing the system and not requiring client machines to store all possible domains or TLDs. The size of the anticipated PSL may be larger than what is feasible in a fixed text file loaded onto all clients.

8 Findings

Finding 1: The PSL is a design compromise between convenience of use and accuracy of its contents.

As detailed in Section 6, while a PSL is intended to be a convenient summary of the policy and organizational boundaries in the DNS, the nature of the maintenance and administration of these lists by third parties effectively preclude these lists ever achieving an appropriate level of authority and timeliness to fulfill a foundational role in any system where security is a priority.

The conversation that underlies this design compromise between convenience of use and accuracy of its contents illustrates a common theme in the practical application of security to operational systems: there is always a tradeoff between the desire for convenience and efficiency, and due prudence in accepting a third party's attestations as being equivalent to an authoritative statement from the original source.

[16] See ICANN Registry Request Service by Atgron.Inc. https://www.icann.org/en/system/files/files/atgron-wed-request-08oct13-en.pdf.

Finding 2: There is no consensus definition of "public suffix" and associated terms, and in fact the PSL is used for several purposes having to do with administrative boundaries in the DNS.

As mentioned in Section 4.1, differences of opinion exist over what a "public suffix" is, and there is variation among stakeholders that base their definition upon their precise need or solution. This is exacerbated by the inclusion of "private suffix" names in many PSL applications due to many identical needs and use cases for such suffixes. The lack of a consistent industry-wide definition of "public suffix" and related terms contributes to such confusion, and as subsequent issues exposed show, is central to many aspects of the issues discussed in this report with regard to creation and use of PSLs.

Finding 3: There is a lack of accountability mechanisms for ensuring PSLs are produced in a consistent, fair, unbiased manner with recourse for individuals or organizations that may have an issue.

As mentioned in Section 4.2, the Mozilla PSL could be considered accountable to some extent given its public nature and fairly transparent process. However, gaps exist in accountability. In general, there is no organization or entity with formal authority or accountability for determining content of any PSL today.

Finding 4: A knowledge gap exists between registries and maintainers of the public suffix lists regarding the processes and responsibilities for changes and additions to the Mozilla PSL and other PSLs.

Finding 5: There is no universal library, framework, tool, etc. for PSL use. Further, implementers do not use PSL entries consistently in software or other services. Registries cannot expect similar behavior across all devices or applications for their suffixes. Such behaviors contribute to unstable user experience.

Finding 6: There is great variation of latency for implementing PSL changes in software applications and Internet services. The update and distribution cycle for changes to entries in a PSL impact the usability and acceptance of new TLDs and/or policies in TLDs.

As illustrated in Section 5.2, in best-case scenarios, latency can be around 12 weeks. Common cases of latencies last much longer, and some PSLs never get updated.

Finding 7: There is a general lack of authentication and other standard security controls for the content and transmission of PSLs from maintainers to users.

Finding 8: Due to the wide variety of use cases for PSLs, it may be difficult to create a one-size-fits-all PSL for all audiences covering any application or usage.

Finding 9: If the new gTLDs use public suffixes similarly to the existing generic

TLDs, where typically there is one public suffix because the entire TLD is "public," there would be limited impact to the size of a PSL. However, if new gTLDs use public suffixes similarly to some ccTLDs, which may include more than one public subdomain, the impact to any PSL could be significant.

9 Recommendations

The SSAC first calls on the IETF and application community to directly address this fundamental design compromise by designing, standardizing and adopting alternative solutions (see Recommendation 1). Second, because use of PSLs today is prevalent, and noting the time it takes for the IETF to standardize alternative solutions and the community to deploy them, the SSAC recommends a set of near-term measures to alleviate some of the higher risk issues with the current maintenance and use of PSLs (Recommendations 2-6).

Recommendation 1: Recognizing alternatives to the PSL have been discussed (see Appendix A), the SSAC recommends the IETF and the applications community consider them for further specification and possible standardization through the IETF process.

These efforts should consider the issues raised by this Advisory as inputs to the problem statements to be addressed by new standards and solutions.

The DBOUND working group, chartered in April 2015 to consider issues arising from the observation that "Various Internet protocols and applications require some mechanism for determining whether two domain names are related," is addressing a more general version of this problem, having to do with other kinds of administrative boundaries as well. But technologists interested in the "public suffix designation" problem should consider participation there.

Recommendation 2: The IETF should develop a consensus definition of "public suffix" and other associated terminology (e.g. "private suffix").

The DBOUND charter and draft problem statement suggest some useful distinctions to include.

Recommendation 3: To close the knowledge gap between registries and popular PSL maintainers, ICANN and the Mozilla Foundation should collaboratively create informational material that can be given to TLD registry operators about the Mozilla PSL.

Recommendation 4: The Internet community should standardize the current approach to PSLs. Specifically:

Recommendation 4a: ICANN, as part of its initiatives on universal acceptance, should encourage the software development community (including the open source community) to develop and distribute programming and operating system libraries implementing

robust (i.e. authenticated, timely, secure, accountable) distribution mechanisms for PSLs. These libraries should be written across all common platforms and operating systems in a way as to ensure consistent and standard interpretation of a given PSL across all platforms.

Recommendation 4b: Application developers should use a canonical file format and modern authentication protocols as specifications to this work.

Recommendation 4c: Application developers should also replace proprietary PSLs with well-known and widely accepted PSL implementations such as the Mozilla PSL and the proposed IANA PSL (Recommendation 5).

Recommendation 5: IANA should host a PSL containing information about the domains within the registries with which IANA has direct communication. Such a PSL would be authoritative for those domains.

Such a list should include, at a minimum, all TLDs in the IANA root zone.

Recommendation 6: ICANN should explicitly include use and actions related to a PSL as part of the work related to universal acceptance.

10 Acknowledgments, Disclosures of Interest, Dissents, and Withdrawals

In the interest of transparency, these sections provide the reader with information about four aspects of the SSAC process. The Acknowledgments section lists the SSAC members, outside experts, and ICANN staff who contributed directly to this particular document. The Disclosures of Interest section points to the biographies of all SSAC members, which disclose any interests that might represent a conflict—real, apparent, or potential—with a member's participation in the preparation of this Report. The Dissents section provides a place for individual members to describe any disagreement that they may have with the content of this document or the process for preparing it. The Withdrawals section identifies individual members who have recused themselves from discussion of the topic with which this Report is concerned. Except for members listed in the Dissents and Withdrawals sections, this document has the consensus approval of all of the members of SSAC.

10.1 Acknowledgments

The committee wishes to thank the following SSAC members and external experts for their time, contributions, and review in producing this Advisory.

SSAC members

Jaap Akkerhuis
Patrik Fältström
Geoff Huston

Warren Kumari
Danny McPherson
Ram Mohan
Rod Rasmussen
Suzanne Woolf

Invited Guest Participants:
Casey Deccio[17]
Jothan Frakes

ICANN staff

Patrick Jones
Julie Hedlund
Kathy Schnitt
Steve Sheng (editor)
Jonathan Spring

10.2 Disclosures of Interest

SSAC member biographical information and Disclosures of Interest are available at: https://www.icann.org/resources/pages/biographies-2014-10-08-en.

10.3 Dissents

There were no dissents.

10.4 Withdrawals

There were no withdrawals.

[17] For his contribution as the ICANN fellow researching on this topic.

Appendix A: Alternatives to Public Suffix Lists

The SSAC recognizes the following alternatives to the Mozilla PSL and believes the Internet community should consider them for specification and adoption.

Public Suffix Structured File Format
A PSL design has been proposed to standardize the file format and its retrieval methods.[18] Extended markup language (XML) is the proposed format of the list in this design (practically, JavaScript Object Notation (JSON) or another structured language could be used). Hypertext Transfer Protocol (HTTP) is the proposed method for retrieving it. The format allows for flexible definition of public suffixes at a Top Level Domain (TLD), descending multiple levels, if desired. It can be retrieved securely with means, such as Secure Sockets Layer (SSL).

Standardized Public Suffix Distribution Method
This proposal is dependent on a structured file format. If the file format is standardized, the maintenance and distribution of the list could also be standardized. One benefit of this system, compared to the current Mozilla PSL, is that the maintenance is distributed to the TLD registries, such that a single entity doesn't need to be responsible for updating the entire list. Rather, each TLD assumes responsibility for maintaining its own tree. However, there is no delegation process defined for sub-entities to be able to recursively maintain their own public suffixes. In complex public suffix scenarios this might be unpalatable.

Enhanced Validation of Domains Using the Domain Name System (DNS)
Another proposal suggests using the DNS and HTTP to determine whether a DNS name is a public suffix.[19] In this proposal a browser performs a DNS lookup of type "A" (address) for the suffix in question, the response of which is used to help make a determination as to whether the suffix is public. The idea is that typically a public suffix wouldn't be expected to have an address associated with it. In cases where the client is unable to perform such a DNS lookup, it can perform instead an HTTP HEAD request to the suffix, of which a successful response indicates that the domain is not a public suffix.

The simplicity of this proposal is its strength. However, simplicity is also a weakness.

Related Internet Engineering Task Force (IETF) Work
DBOUND (*D*omain *Bound*aries) is a newly chartered working group within the IETF for developing some solution for determining whether two domain names are related.[20] The January 2015 draft of the proposed DBOUND problem statement (http://datatracker.ietf.org/doc/draft-sullivan-dbound-problem-statement/) is still rough but indicates that defining a "policy realm" is one mechanism of determining if domains are related, however current methods for determining a policy realm are considered insufficient—namely, the Mozilla PSL. A "policy realm" is roughly a group of domains

[18] See https://tools.ietf.org/html/draft-pettersen-subtld-structure-10.
[19] See https://tools.ietf.org/html/draft-pettersen-dns-cookie-validate-05.
[20] See https://datatracker.ietf.org/wg/dbound/charter/.

under the control of a single administrator.[21] There are two distinct proposed solutions from the DBOUND working group that are still in early discussions: asserting DNS policy realm boundaries and publishing organizational boundaries.

Asserting DNS Policy Realm Boundaries[22]
One proposed solution for defining which domains within the same DNS subtree are related is to create a "start of policy authority" (SOPA) record in the DNS. The proposed SOPA record would allow a domain administrator to explicitly state whether a target domain is included or excluded from the policy realm of the domain. The authors acknowledge that without using DNSSEC the usual problems of spoofing and cache poisoning arise. This solution is only viable for target domains that are an ancestor, a descendent, or a sibling of the owner name; it is not considered wise for cross-linkages across DNS subtrees.

Publishing Organization Boundaries in the DNS[23]
One proposed solution for defining which domains are under the same organizational management is to create a new subdomain "`_ob`" and insert this domain with a TXT record to mark organizational boundaries. The proposal includes measures for handling multiple organizational boundary transitions; each lookup would require two or three DNS queries until a query for a `_ob` domain returned a NXDOMAIN or NSEC result. This solution also has the usual problems without DNS Security Extensions (DNSSEC), and is not able to provide cross-linkages of related domains across DNS subtrees.

[21] See https://datatracker.ietf.org/doc/draft-sullivan-dbound-problem-statement/?include_text=1.
[22] See http://tools.ietf.org/html/draft-sullivan-domain-policy-authority-01.
[23] See https://tools.ietf.org/html/draft-levine-orgboundary-02.

Appendix B: Mozilla Public Suffix List

Entries in the Mozilla PSL
As of 11 August 2014, the Mozilla PSL is a text file containing 6,763 entries, arranged into two sections:

- The *ICANN* domains section, which includes 6333 generic Top Level Domains (gTLD) and country code TLD (ccTLD) suffixes entries. This section is intended to include suffixes that comply with Internet Coordination Policy (ICP-3)[24] and are directly delegated by IANA or are associated to them.
- The *private* domains section, which includes 430 entries from many subdomain registration services such as CentralNic (owner of e.g. eu.com and us.org), as well as and companies such as DynDNS, Amazon, Google, GitHub, Heroku,[25] Microsoft and Red Hat, who provide Domain Name System (DNS) resolution and cloud services. This section exists because some registered domain owners wish to delegate subdomains to parties not trusting each other.

An analysis of the label distribution among PSL entries in the ICANN domains section and Private domains section provides some quantifiable statistics on how the PSL is being used. The distribution is shown according to three variables – number of labels, section (ICANN or Private), and TLD type.

Table 3: Label Distribution of Entries in the PSL by categories of entries

	gTLD	ccTLD	IDN Total	Total
1 Label – ICANN	457	225	87	769
2 Labels – ICANN	651	2,986	6	3,643
3 Labels – ICANN	0	1,918	0	1,918
4 Labels - ICANN	0	3	0	3
1 Label – Private	0	0	0	0
2 Labels – Private	308	70	0	378
3 Labels – Private	27	9	0	36
4 Labels - Private	15	1	0	16
Total	1,458	5,212	93	6763

One-label entries (i.e., TLDs) are by definition public suffixes. Two- and three-level entries under ccTLDs account for 72.5% of the PSL entries. The majority of these entries correspond to geographic regions within the country represented by the ccTLD,[26] and a few TLDs contribute to the majority of the list.

[24] See https://www.icann.org/resources/pages/unique-authoritative-root-2012-02-25-en.

[25] See Hiroku Website Security note at https://devcenter.heroku.com/articles/cookies-and-herokuapp-com.

[26] For example, many legacy entries in `.us` follow RFC1480, such as for each state in the United States being represented under the `us` ccTLD, e.g., `id.us` for Idaho.

Table 4: TLDs with >50 entries in the ICANN Domain Section of the PSL

TLD	PSL Entries
JP	1748
NO	754
MUSEUM	549
IT	369
US[27]	225
PL	180
RU	133
AERO	90
UA	79
BR	71

Success Factors of Mozilla PSL

The Mozilla PSL project began with a limited remit and was only intended to improve browser security and user privacy. Since then, it has become very successful. The Mozilla PSL is used by the majority of browsers and has lots of derivative uses. Such widespread adoption speaks to the demand for such "suffix" lists. Although there are other similar initiatives that have existed, most such initiatives have either not gained traction or abandoned the effort and used the Mozilla PSL instead, since it met the need. There are a few factors that contribute to the Mozilla PSL's success within Mozilla Firefox, and the various derivative users, integrators, developers and projects:

- Single entity manages the list: single point-of-contact for changes, single process for approving changes.
- Maintainer is a reputable organization, knowledgeable about HTTP usage, privacy and security.
- Version controlled with bug tracking system and follows a release cycle so that it can be tightly integrated with the system in which it is used.
- Database of entries can be embedded in the application. This reduces latency for potentially frequent lookups so that the record level lookups are fast and efficient.
- As it is an existing solution to widespread need for reasonable "TLD" logic, it allows developers to focus on developing.

[27] Typically, descending 3ld+ entries under .US are Legacy RFC1318 or RFC1480 entries.