Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 13, 2019

D. Crocker
Brandenburg InternetWorking
T. Adams
Proofpoint

June 11, 2019

DNS Perimeter Overlay
draft-dcrocker-dns-perimeter-01

Abstract

The Domain Name System (DNS) naming syntax provides no meta-data for indicating administrative transitions through the hierarchy. For example, it does not distinguish the higher-level portions that operate as public registries, versus those that operate as private organizations. This specification creates a basic overlay mechanism for defining a logical Perimeter between administrative entities through the naming hierarchy. The mechanism can then be applied for a variety of independent administrative indications.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 13, 2019.

Table of Contents

1. Introduction

Although some administrative structure can be inferred for the Domain Name System (DNS), there is no formalized syntax that distinguishes between the sequence of names in its referenced hierarchy. It does not mark any differentiating characteristics, such as transitions across administrative perimeters, as the sequence is followed. For example, it does not mark a change in administrative authority for subordinate names. A common example of needing such differentiation

is to indicate what part of a name belongs to a 'public' registry and what part belongs to a private registrant within that registry.

This specification defines a mechanism for marking perimeters in domain names, thereby permitting creation of logical overlays to the DNS. Various types of administrative distinctions could be useful. To facilitate creation of multiple, logical overlays, this specification only defines a basic, extensible mechanism for marking the presence of a Perimeter between administrations, and indicating where the semantics of the Perimeter are defined.

As a detailed example and to satisfy a real-world need, an overlay that emulates the established Public Suffix List ([PubSuff], [PubSuff-SSAC]) is provided in Appendix B.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. History

A number of Internet functions seek to discern a 'base' portion in a domain name, such as the basic organizational name like example.com, from a longer name, like marketing.west.example.com. An approach to accomplishing this is to distinguish the part that belongs to "public" registries, and consider the next node name below that as the base name.

The Public Suffix List has been used to satisfy this requirement. It has two kinds of domain names. One is for these 'public' names that operate through ICANN coordination. The other is 'private' which serves as a naming base in some cases [PubSuff], [PubSuff-SSAC]. The list is maintained as an independent effort producing a standalone document, with all of the challenges involved in such an operation. Entries are manually registered, which requires vetting of the source and on-going validation. Entries can be for a single name or can use a wildcard notation, to cover all names below the one that is registered. It is also possible to enter a name declared to be an exception to the wildcard cover. In keeping with the move towards support of non-ASCII names, entries are in UTF-8.

For 2015-2016, IETF's DBOUND working group explored possible DNS enhancements that would permit embedded information to support uses such as the Public Suffix List. The effort ultimately was unsuccessful. Several drafts were used as input to the working group discussions [DBOUNDwg].

NOTE: _The following summaries are intentionally terse and simplified. Suggestions for superior language that remains terse are eagerly sought. /dcrocker_

Two were considerations of underlying issues:

DABprob: _"DBOUND: DNS Administrative Boundaries Problem Statement"_ offers a "Problem Statement" and offered an extensive list of possible uses [DABprob].

DNRcon: _"Concepts for Domain Name Relationships"_ explores the general topic of "relationships" between different domain names. It considers structural choices, such as within the same naming hierarchy, versus across separate branches. It also considers types of relationships, such as public vs. private. Some use cases are considered, as are some solution considerations [DNRcon].

The proffered specifications were:

ODuse: _"Organizational Domains and Use Policies for Domain Names"_ proposes "an extensible system in which domain name policies can be discovered at various levels in the DNS tree." A policy record is stored under an underscored node name in a TXT record. The record can indicate that the current node is an organization name or that the name one level down is. Wildcards are permitted, to cover sub-domains, indicating a limit to the number of levels down. Usage policies are marked as allowed or not allowed. Initial types of policies were httpcookie and all (to indicate a default.) There is a mechanism for using URIs to retrieve parameters.[ODuse]

OBD: _"Publishing Organization Boundaries in the DNS"_ offers a specification "to publish in the DNS the boundaries between organizations that can be adapted to various policy models". Policies are expressed within a 16-bit bit-masked field. A demarcation point is indicated by a published record above the point, using a new DBOUND RR. The record can indicate that there are no boundaries lower than this name. The search algorithm is fully specified for all uses. It also indicates that "[d]ifferent sets of boundary rules can be published for different applications." The applicable application of a boundary is indicated by a numeric value in the record [OBD].

ODUP: _"Resource Record for DNS Administrative Boundaries"_ specifies a method for "judging domain name administrative boundaries" and considers the records within a boundary to be

related and those across a boundary to be unrelated. The specification defines a different DBOUND RR, from that of [OBD]. If supports assorted flags plus an "Anchor Name/Name Collection" field. The Anchor Name usage "build a connection between the owner name and the anchor name which is a FQDN", where"owner name" is defined as "some names' anchor name" in a different DBOUND RR. The Name Collection usage lists "names which are supposed to share the same DNS boundaries under the same anchor name" [ODUP].

SOPA: _"Asserting DNS Administrative Boundaries Within DNS Zones"_ defines "...a way to assert that two domains lie in the same policy realm..." or that they do not [SOPA].

In general terms, it's important for any effort in this space to carefully consider the guidance in both [RFC5507] and [RFC6950]. Of particular concern to the current draft are the caveats highlighted in Section 3.3.1 of [RFC6950], about synchronization, authorization and delegation.

3. Perimeter Overlay Overview

A Domain Name Perimeter (DNS Perimeter) distinguishes a logical separation, occurring between two adjacent nodes in the DNS hierarchy. The name that is lower in the hierarchy marks the beginning of its portion (identified by "BEGIN"), and the name higher marks the end of its portion (identified by the term "END"). As such, a Perimeter is the interface between segments along a domain name branch, for which there can be different administrative authorities and to which different policies can be applied.

Because the DNS does not permit associating information with the graph connector 'between' names, information about a Perimeter needs to be associated with one or both of the nodes adjacent to the Perimeter. One possible advantage of this requirement is permitting flexibility in the operational management of marking a Perimeter. The organization 'above' the Perimeter might have more or less incentive to mark the Perimeter than the organization 'below' it. In this way, the Perimeter can be marked by the organization with the greater incentive (or by both organizations, depending on the use case.)

Definition of a DNS Perimeter:

> A logical demarcation between two, adjacent DNS nodes, where one node is the parent of the other, and the child is part of a branch spanning one or more subdomains.

The metadata that is associated with such a node name needs to indicate:

Position: Whether this node name is at the 'end' of am administrative sub-hierarchy, before a Perimeter transition -- and therefore the final node name 'above' the Perimeter; whether it 'begin's a portion of administrative sub-hierarchy, immediately after a Perimeter transition; or whether it is a node name that is an internal 'part' of a sub-hierarchy.

The 'part' construct might be useful for defining a place to hold parametric detail specific to that node within the hierarchy. It might also be useful to hold a pointer to the 'begin' node name.

Schema: The registered name of the perimeter definition. The Schema name identifies the semantic discipline for the record containing the reference. This permits multiple Schemas to share the same perimeter.

Parameters: Any schema-specific information required by the schema definition.

Note: DNS Perimeter Overlay uses a TXT RRset to an _underscored node name (_perim). This constrains queries for TXT records to only Perimeter records. Still, a query to a Perimeter Overlay node will return all of the TXT records stored there, and there might be multiple 'users' (schemas) using the same DNS node name. So, the client will need to do a simple search of the returned TXT RRs, for the one that is desired. It is expected that there will never be a large number of such records; so the burden of distinguishing among multiple records is expected to be small.

4. DNS Perimeter Overlay Syntax

A node that is immediately above or below a DNS Perimeter indicates itself with TXT DNS RR, in an _underscore-labeled sub-branch under that node [RFC8552].

4.1. Perimeter Branch Indication

The scoped use of the Perimeter TXT RR is indicated with a subordinate, leaf node name of:

"_perim."

The IANA registration information for the _perim DNS scoped attribute name is in Section 8.1.

4.2. Perimeter TXT RR

A TXT RR that is used to indicate a Perimeter is composed of an initial identifier, followed by three fields, as described in Section 3.

The ABNF [RFC5234] for the Perimeter TXT RR is:

```
Perim TXT:  "perim" sp Pos sp Schema [sp Params]
            ; ISSUE: the 'perim' string is arguably redundant, given that the
            ; _underscored node naming approach already defines this as a
            ; perimeter record.
            ; I encourage keeping it, so interpretation of the record can stand
            ; on its own. /dcrocker

Pos:        "begin" / "end" / "part"
            ; begin = first in the perimeter hierarchy sub-sequence
            ; part = within the hierarchy sub-sequence
            ; end = last in the hierarchy sub-sequence

Schema:     { Entry from DNS Perimeter Registry }

Params:     Param *("," Param)

Param:      attr [eq val]

attr:       1*alpha
            ; what is a better choice than <alpha>? /dcrocker

eq:         "="

val:        1*alpha
            ; what is a better choice than <alpha>? /dcrocker
```

Perimeter TXT RR ABNF

Schema is the registered name for a specific use of the DNS Perimeter Overlay mechanism. The IANA registration information for the _perim DNS scoped attribute name is in Section 8.2.

That is, a TXT record under _perim has a series of space-separated fields:

1. Identifies this as a _perim TXT record.

2. Indicates whether the record 'begins' an administrative area, by appearing as the first node after a Perimeter, or whether it 'ends' an administrative area, by appearing as the last node before a Perimeter.

3. Indicates the controlling Schema.

4. Optional to the syntactic mechanism, this is a series of one or more comma-separated (with no white space) parameters, as defined by the particular Schema specification, where a parameter can be a simple string or an attribute/value pair.

4.3. Syntax ExampleS

Therefore, an organization might indicate the top of its naming hierarchy with:

```
_perim.company.pubregistry.example
   /
  TXT  "perim begin suffix private"
```

Suffix BEGIN Example

while the parent registry for this organization's name might also indicate the name above it is the bottom of the delegating organization's naming branch:

```
_perim.pubregistry.example
   /
  TXT  "perim end suffix public"
```

Public Suffix END Example

and a node within a private organization's branch might point to its 'organizational domain' that begins this private suffix:

```
_perim.dept.company.pubregistry.example
   /
  TXT  "perim part suffix private od=company.pubregistry.example"
```

Suffix PART Example

5. Discussion

5.1. End/Begin Interaction

The occurrence of either a 'begin' or an 'end' _perim TXT resource record defines the Perimeter, in terms of basic Perimeter existence. The presence of both _perim TXT records both above and below the Perimeter is redundant.

For this core mechanism, a 'begin' _perim TXT record MAY occur in a top-level domain, immediately under the DNS root. It would, of course, have no corresponding 'end' parameter "above" the Perimeter. Beyond specification of the technical details, actual usage of a Perimeter record for a name administered through a "public" registry is a matter of registry policy and is, therefore, outside the scope of this specification.

A particular Schema might define specific requirements or constraints on the occurrence of its Perimeter records. The Schema might mandate only one type of record. Or it might permit policy parameters that could conflict. Such issues are entirely within the purview of the Schema specification and are invisible to this core DNS Perimeters Overlay mechanism.

5.2. Schema/Schema Interaction

For simplicity and commonality, the core DNS Perimeter Overlay mechanism defers policy and usage detail up to the Schema specifications that rely on that detail.

> The semantics and extended syntax of a Perimeter are defined by a specific, registered Schema that is referenced in a _perim TXT RR. In terms of the core Perimeter Overlay mechanism, a Perimeter that is defined by one Schema is invisible to other Schemas by default, even if they share the same node.
>
> However a Schema specification MAY define its own rules regarding the occurrence of different Perimeter Schemas and/or the relationship of this Schema to another. For example, one Schema's Perimeter Overlay might create dependencies and interactions with another Schema Perimeter Overlay.

6. Sample Overlay Templates

Here are some notional use cases, for abstract usage models using DNS Perimeter Overlays. They are provided as basic discussions, rather than detailed specifications, to serve both as simple examples and as guidance for possible adaption to specific needs. Other models are certainly plausible.

NOTE: This section might be appropriate to move into an independent document, as a larger repertoire of examples is developed and specified. As this document develops, suggestions for additional samples is encouraged. /dcrocker

A Schema specification needs to make clear what operational and policy models it is using, to distinguish it from other Schemas that might seem similar.

CAVEAT: There is a basic (but easily-forgotten) reality that the registry for a parent domain has ultimate control over the descendant domains. All sorts of anomalies are possible (and likely) when a descendant is a different organization, but ultimately, that's the type of issue that isn't directly discernible via DNS. Concern for such issues is internal to the administration of that DNS node hierarchy. responsible

6.1. Default/Override 'Convenience' Overlay

An organization might want to have a Perimeter early in the DNS hierarchy that defines a basic set of parameters and policies, as defaults for names within the Perimeter. It might then permit nodes under this to override any of these defaults. The default record, therefore, serves as a convenience, to reduce the amount of detail that needs to be provided at lower levels in the DNS hierarchy.

Specifying the details that can be provided as defaults is straightforward.

The basic operational model is for the client to start with the full DNS name, down to the lower level and then look up to the higher-level 'base' name. There needs to be a simple, efficient means for the client to determine what that 'base' name is, so that it can deterministically query it for the default information.

6.2. Master/Addition 'Control' Overlay

An organization might want to have a Perimeter early in the DNS hierarchy that defines a rigorous set of mandatory parameters and policies. Within its administrative purview, these would be global details, enforced for all subordinate names.

As for the Convenience model, the overlay specification here needs to make clear what operational model applies. The remaining technical details are the same as for the Convenience model. What differs is the semantics of using the superior/subordinate overlay records.

Note that most of the operational details of the 'Control" model are the same as the 'Convenience' model, although their semantics have a basic difference.

6.3. Vendor/Customer Overlay

A vendor that services customers via subdomains under their corporate domain might opt to publish DNS Perimeter declarations as clear demarcations between their "enterprise" and "customer" nodes. The Schema might define semantics that enable third parties to support the customers, potentially applying different rules per customer node. In this case, each "begin" _perim TXT RR associated with a node will define the policies that apply to that customer, while the "end" _perim DNS TXT will act as the demarcation line between the customer(s) and the vendor.

6.4. Organizational Alias

There are various relationships that might exist between two domain names in different DNS branches. One example is complete equivalence. That is, the two names are aliases for the same organizational unit. A DNS Perimeter Overlay Schema could support this construct by having a Schema parameter that specifies a the domain name of organizational alias. Each name could point to the other. (The 'part' example in Section 4.3 demonstrates the simpler case of merely pointing to a name earlier in the branch, but a Scheme could define a similar construct that instead points to names in other branches.) Concerns for authorization and accuracy would be internal to the Schema.

7. Propogating 'Begin' Location for Search Efficiency

NOTE: This section is currently offered as a discussion, to consider the plausibility of an approach at efficiently finding a 'begin' record, given a name farther down its branch. /dcrocker

One concern for the pragmatics of DNS operation is being able to easily populate records into a large number of sub-domains. Another is producing a useful response for names that are not registered, such as for communicating policies related to an organization's sub-domains. In both cases, the information can be stored in a higher-level name.

However it is one thing to list data in the DNS -- somewhere up the branch of the hierarchy -- and quite another to find it, when its location is not already known.

> _Given a longer domain name, what is the process of finding the shorter portion containing a _perim TXT 'begin' declaration?_

In the worst case, a tree-walk is required, querying each, next-higher portion of the DNS, or starting at the root and querying each node down. For a name with many components, this can be expensive and slow, while essentially creating a vector for a denial of service attack.

A feature embedded in the basic DNS specification is the wildcard, as defined in Section 4.3.3 of [RFC1034]. This permits server-side configuration into a higher-level domain name and delivers the information for queries to subordinate names. Unfortunately, this feature cannot be used for records that are stored under a specialized naming branch such as those using underscored scoping, since they are in an adjacent branch under the name and cannot propagate.

> _So how can a user process that has a fully qualified domain name, find Perimeter information from some upper level in the hierarchy, such as the base "organizational domain", when the classic DNS wildcard feature cannot be used?_

In some cases, the queried name will exist and might have a 'part' record to provide the information, or it might exist and not have the information, or it might not exist. The latter two cases requires some additional means for obtaining information about the containing Perimeter.

Absent additional mechanism, finding a DNS Perimeter requires some sort of tree walk, which has the problems cited above. Use of a purpose-built RR -- rather than underscore-scoped naming -- would permit employing wildcards, but new RRs continue to suffer deployment and use barriers.

Having a tree-walk done offline and publishing a list is a possibility. That is, publish a table that shows the entries which were found by a background searching process. When there are relatively few entries and the search space is relatively small and the rate of change is relatively slow, this approach can be useful. However it requires consulting an external table and requires an effort to maintain it.

Another approach is use of the DNS Additional section in the server response:

> Query for a Perimeter node; the server will return would include the associated Perimeter BEGIN record from earlier in the

hierarchy, if the queried node is within that hierarchy -- that is, is above the actual or virtual END record. (As for any information supplied through the Additional section, the responding server will need to be modified to provide this enhanced information for specific kinds of queries.)

It might be reasonable to constrain this behavior only to a Perimeter record that requests it, by adding a wildcard construct to the basic Perimeter BEGIN syntax.

A Perimeter-aware client -- or recursive server -- could cache these results, building an incremental portion of the overall table for this type of Perimeter.

8. IANA Considerations

8.1. _perim Registration in DNS Underscore Global Scoped Entry Registry

The following entry is to be added to the DNS Underscore Global Scoped Entry Registry:

RR Type	_NODE NAME	REFERENCE
TXT	_perim	{this document}, Section 4

Table 1: _perim Registration in Global Scoped Entry Registry

8.2. DNS Perimeter Overlay Registry

The DNS Perimeter Overlay Registry lists specific uses of the DNS Perimeter Overlay mechanism.

The registration table for the DNS Perimeter Overlay Registry will contain two columns:

SCHEMA	REFERENCE

Table 2: DNS Perimeter Overlay Registry Table

- This registry is to operate under the IANA rules for "Expert Review" registration; see Section 8.2.1.

- The detail to be provided by a DNS Perimeter Overlay entry's referenced Schema specification is defined in Section 4.2.

- The specification referenced in a DNS Perimeter Overlay registration MUST contain values for all of the fields specified in Section 4.2.

- Within the registry, each Schema name must be unique.

- The table is to be maintained with entries sorted by the Schema name.

- The required Reference for an entry MUST have a stable resolution to the organization controlling that registry entry.

8.2.1. Guidance for Expert Review

This section provides guidance for expert review of registration requests in the DNS Perimeter Overlay Registry.

> This review is solely to determine adequacy of a requested entry in this Registry, and does not include review of other aspects of the document specifying that entry. For example such a document might also contain a definition of the resource record type that is referenced by the requested entry. Any required review of that definition is separate from the expert review required here.

The review is for the purposes of ensuring that:

- The details for creating the registry entry are sufficiently clear, precise and complete

- The Schema name is unique in the table

For the purposes of this Expert Review, other matters of the specification's technical quality, adequacy or the like are outside of scope.

8.3. Suffix Entry in DNS Perimeter Overlay Registry

NOTE: As a formality, this section is in the IANA section for this document. However it is expected that the Public Suffix use of DNS Perimeter Overlay will be moved to a separate specification document, before this document is published. /dcrocker

SCHEMA	REFERENCE
suffix	{this document}, Appendix B

Table 3: DNS Perimeter Overlay Registry Table

9. Security Considerations

This memo defines a mechanism for signaling information about administrative perimeters. The mechanism itself introduces no security issues. However specific uses of the mechanism might define transitions in authority that offer new attack surfaces.

- A basic opportunity for concern is authorization to make a particular assertion, using a DNS Perimeter Overlay. The basic mechanism defined here offers no means for validating an assertion. So any detailed specification for a particular use needs to consider the potential of unauthorized assertions.

- Conflicting Perimeter entries for adjacent 'begin' and 'end' assertions could be problematic. That is, information in the _perim TXT RR for the parent name might conflict with information in the _perim TXT RR for the child. Consideration of such a conflict is left to the individual Schema specifications that use the DNS Perimeter Overlay mechanism.

10. References

10.1. References - Normative

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 5234, January 2008.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", RFC 8162, May 2017.

[RFC8552] Crocker, D., "Scoped Interpretation of DNS Resource Records through "Underscored" Naming of Attribute Leaves", RFC 8552, ISSN 2070-1721, March 2019.

10.2. References - Informative

[DABprob] Sullivan, A., Hodges, J., and J. Levine, "DBOUND: DNS Administrative Boundaries Problem Statement", I-D draft-sullivan-dbound-problem-statement-02, August 2016.

[DBOUNDwg] IETF, "Domain Boundaries (dbound)", 2016.

[DNRcon] Deccio, C. and J. Levine, "Concepts for Domain Name Relationships", I-D draft-deccio-dbound-name-relationships-00, July 2015.

[OBD] Levine, J., "Publishing Organization Boundaries in the DNS", I-D draft-levine-dbound-dns-01, September 2016.

[ODUP] Yao, J., Kong, N., and X. Li, "Resource Record for DNS Administrative Boundaries", I-D C. Deccio, January 2016.

[ODuse] Deccio, C., "Organizational Domains and Use Policies for Domain Names", I-D draft-deccio-dbound-organizational-domain-policy-03, July 2016.

[PubSuff] Foundation, M., "Public Suffix List", URL https://publicsuffix.org.

[PubSuff-SSAC] Committee, I. S. A. S. A., "SAC070: SSAC Advisory on the Use of Static TLD / Suffix Lists", URL https://www.icann.org/en/system/files/files/sac-070-en.pdf, May 2015.

[PubSuffSyn] Foundation, M., "Public Suffix List Format", URL https://publicsuffix.org/list/.

[RFC5507] IAB, Faltstrom, P., Austein, R., and P. Koch, "Design Choices When Expanding the DNS", RFC 5507, April 2009.

[RFC6950] Peterson, J., Kolkman, O., Tschofenig, H., and B. Aboba, "Architectural Considerations on Application Features in the DNS", RFC 6950, October 2013.

[SOPA] Sullivan, A. and J. Hodges, "Asserting DNS Administrative Boundaries Within DNS Zones", I-D draft-sullivan-domain-policy-authority-02, February 2016.

[SubTLD] Pettersen, Y., "The Public Suffix Structure file format and its use for Cookie domain validation", I-D draft-pettersen-subtld-structure-10, February 2014.

Appendix A. Acknowledgements

Appendix B. DNS Suffix Perimeter

ISSUE: _Specification of Perimeter use to replicate Public Suffix List functionality. This section needs careful review and revision by the PSL community. _ /dcrocker

It appears that there are a number of adjunct uses of Domain Names that get merged with the PSL. These probably are candidates for other Perimeter Overlay encodings. /d

ISSUE: _The basic usage mode for PSL information is for an application that has a fully qualified domain name to 'find' the portion that is public, as distinct from the remaining portion that is assigned by a private registry. The 'finding' process is not facilitated by the DNS, which only queries for an exact name, rather than doing "searching". Worse, this impedes building a table by brute-force testing of the tree._

So an open issue is the method for either real-time or background use of PSL information through DNS Perimeter Overlay. /dcrocker

The Public Suffix List describes itself as [PubSuff]:

> "A "public suffix" is one under which Internet users can (or historically could) directly register names."

An advisory report by the ICANN Security and Stability Advisory Committee uses a definition of PSL from [SubTLD]:

> "A domain under which multiple parties that are unaffiliated with the owner of the Public Suffix domain may register subdomains."

The basic semantics of the list are quite simple, only marking the Perimeter between the portion of a domain name -- it's suffix -- administered by a public registry and the remaining portion of the name administered by a registrant. Some uses of the list have more elaborate semantics, but these really are value-added features beyond the basic mechanism -- even though some are encoded in the published list. The details of the Public Suffix list are not amenable to algorithmic derivation, because the criteria for determining whether

a suffix is 'public' varies significantly from one DNS naming branch to another.

The goal in defining a Public Suffix Perimeter within the DNS itself is to permit the owner of a name at a Public Suffix Perimeter to mark its presence directly, rather than having to go through an independent registration service. Anyone can then discern the Perimeter directly, without needing access to a separate list. Further much, or all of, the compiled list can be developed by a rigorous DNS tree walk, rather than by relying on additions and deletions each being submitted to the Public Suffix registration service.

A particular efficiency and convenience in this direct publication method is that the public registry can have a single entry for the 'end' name in the public suffix and implicitly thereby mark all of the children names as the 'begin' of the private part of the name.

NOTE: The details provided here are a bare minimum to define Public Suffix Perimeters. As this specification is reviewed by subject matter experts, it is expected that the details will be enhanced. /dcrocker

B.1. IANA DNS Suffix Registration

The IANA registration information for the Suffix Perimeter entry is at Section 8.3.

B.2. Suffix Perimeter TXT Syntax

This specification for DNS Suffix information, stored in a _perim TXT record, is meant to approximate what is specified in [PubSuffSyn]. Each DNS Perimeter Overlay Suffix Schema TXT RR serves as a 'rule' in the Public Suffix table. Some accommodations have been made, to the constraints of fitting this within a TXT value segment.

Given the variety of uses of information called "Public Suffix List", there could reasonably be different specifications offered. Two possibilities are listed here:

B.2.1. Core PSL

This provides a simple capability for marking a Perimeter, without labeling their 'type'.

```
Perim Params: extra SP  comment

extra:        ["!"] *("*.")
              ; ! = exception
              ; * = wildcard for node name field(s),
              ;     creating prefix to current name.

comment:      "//" *CHAR
```

'Core' DNS Suffix Params ABNF

A simple entry will have no parameters; the existence of the TXT record defines the DNS node containing it as an entry in the Public Suffix List. If wildcard fields are specified, they are added as a prefix to the current node's name. An 'exception' indicator marks this name as overriding a higher-level rule.

B.2.2. PubPrivPSL

This permits distinguishing between portions of the namespace that are public and those, below this, that are private. In order to prevent a private entry from claiming that it is public, a private registry can declare that it is the lowest-level (final) public registry

```
Perim Params: pubpriv extra SP  comment

pubpriv:      "pub" [", fin"] / "priv"
              ; distinguish between public vs. private registry
              ; public registry can indicate it is the final (lowest) one

extra:        ["!"] *("*.")
              ; ! = exception
              ; * = wildcard for node name field(s),
              ;     creating prefix to current name.

comment:      "//" *CHAR
```

'Public/Private' DNS Suffix Params ABNF

There can be layers of public registries and layers of private registries, for a single, fully qualified domain name. This version of the specification permits multiple boundaries; an explicit indication of the type of registry is required. A simple entry will have no <extra> parameters; the existence of the TXT record defines the DNS node containing it as an entry in the Public Suffix List. If wildcard fields are specified, they are added as a prefix to the

current node's name. An 'exception' indicator marks this name as overriding a higher-level rule.

Authors' Addresses

D. Crocker
Brandenburg InternetWorking
675 Spruce Dr.
Sunnyvale, CA 94086
USA

Phone: +1.408.246.8253
Email: dcrocker@bbiw.net
URI: http://bbiw.net/

T. Adams
Proofpoint

Email: tadams@proofpoint.com