Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 4, 2017

J. Klensin
June 2, 2017

DNS Privacy, Authorization, Special Uses, Encoding, Characters, Matching, and Root Structure: Time for Another Look? draft-klensin-dns-function-considerations-00

Abstract

The basic design of the Domain Name System was completed almost 30 years ago. The last half of that period has been characterized by significant changes in requirements and expectations, some of which either require changes to how the DNS is used or that can be accommodated only poorly or not at all. This document asks the question of whether it is time to either redesign and replace the DNS to match contemporary requirements and expectations (rather than continuing to try to design and implement incremental patches that are not fully satisfactory) or to draw some clear lines about functionality that is not really needed or that should be performed somewhere else.

Author's Note

This draft is intended to draw a number of issues and references together in one place and to start a discussion. It is obviously incomplete, particularly with regard to the list of perceived issues and deficiencies with that DNS. To avoid misunderstanding, I don't completely believe some of the deficiencies listed below but am merely providing information about claims of deficiencies. Input is welcome, especially about what is missing (or plain wrong) and would be greatly appreciated.

This document should be discussed on the IETF list or by private conversation with the author.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document explores contemporary expectations of the Internet's domain system (DNS) and compares them to the assumptions and properties of the DNS design. It is primarily intended to ask the question of whether the differences are causing enough stresses on the system, stresses that cannot be resolved satisfactorily by further patching, that the Internet community should be considering designing a new system, one that is better adapted to current needs and expectations, and developing a deployment and transition strategy for it. For those for whom actually replacing the DNS is too radical, the document may be useful in two other ways. It may provide a foundation for discussing what functions the DNS should not be expected to support and how those functions can be supported in other ways, perhaps via an intermediate system that then calls on the DNS. Or it may provide a basis for "better just get used to that and the way it works" discussions to replace fantasies about what the DNS might do in some alternate reality.

While this document does not assume deep technical or operational knowledge of the DNS, it does assume some knowledge and at least general familiarity with the concepts of RFC 1034 [RFC1034] and RFC 1035 [RFC1035] and the terminology discussed in RFC 7719 [RFC7719] and elsewhere.

2. Background and Hypothesis

The domain name system (DNS) [RFC1034] was designed starting in the early 1980s [RFC0799] [RFC0881] [RFC0882] with the main goal of replacing the flat, centrally-administered, host table system [RFC0810] [RFC0952] [RFC0953] with a hierarchical, administratively-distributed, system. The DNS design included some features that were judged to be unworkable and either replaced (e.g., the mail destination (MD) and mail forwarder (MF) approach [RFC0882] that were replaced by the MX approach [RFC0974]), abandoned (e.g., the mechanism for using email local parts as labels described in RFC 1034 Section 3.3), or deprecated (e.g., the WKS RR TYPE [RFC1123]. Newer ideas and requirements have identified a number of other features, some of which were less developed than others. Of course the original designers could not anticipate everything that has come to be expected of the DNS in the last 30 years.

In recent years, demand for new and extended services and uses of the DNS have, in turn, led to proposals for DNS extensions or changes of various sorts. Some have been adopted, including a model for negotiating extended functionality [RFC2671], others were found to be impracticable, and still others continue to be under consideration. A few features of the original DNS specification, such as the CLASS property and label types, have also been suggested to be so badly specified that they should be deprecated [Sullivan-Class].

Unlike earlier changes such as the IDNA mechanisms for better incorporating non-ASCII labels without modifying the DNS structure itself [RFC3490] [RFC5890], some recent proposals require or strongly suggest changes to APIs, formats, or interfaces by programs that need to retrieve information from the DNS or interpret that information. Requirements for such changes suggest that it may be time to stop patching the DNS or trying to extend it in small increments, but to consider development of a system that better meets today's needs and a transition strategy to it.

The next section of this document discusses a number of issues with the current DNS design that could appropriately be addressed by a different and newer design model. In at least some cases, changing the model and protocols could bring significant benefits to the Internet and/or its administration.

This document is not a proposal for a new protocol. It is intended to stimulate thought about how far we want to try to push the existing DNS, to examine whether expectations of it are already exceeding its plausible capabilities, and to start discussion of a redesign or alternatives to one if the time for that discussion has come.

3. Warts and Tensions With The Current DNS

As suggested above, there are many signs that the DNS is incapable of meeting contemporary expectations of how it should work and functionality it should support. Some of those expectations are unrealistic under any imaginable circumstances; others are impossible (or merely problematic) in the current DNS structure but could be accommodated in a redesign. These are examples, rather than a comprehensive list, and do not appear in any particular order.

3.1. Multiple address types

While returning both TYPE A (IPv4 address) and AAAA (IPv6 address) records as additional information in response to any of several query types (see RFC 3596 [RFC3596]) was a useful patch, the better choice, except that it would have required DNS modifications, would almost

certainly have been to establish a single "address" query type (QTYPE) that could return whatever IPv4 and/or IPv6 addresses were available, perhaps with preference information if that were stored in the database, and without requiring the "ANY" be used.

3.2. Matching Part I: Case Sensitivity in Labels and Other Anomalies

The DNS specifications require that, when a domain name used in a query is matched to one stored in the database, ASCII characters be interpreted in a case-independent way, but they do not specify any matching rules other than simple bit string comparison for non-ASCII octets, i.e., octets of labels with the first bit turned on. Even though the current model for handling non-ASCII (i.e., "internationalized") domain name labels (IDNs) [RFC5890] (and see Section 3.3 below) encodes information so the DNS is not directly affected, the notion that some characters in labels are handled in a case-insensitive way and that others are case-sensitive (or that upper case must be prohibited entirely as IDNA does) has caused a good deal of confusion and resentment. Those concerns about inconsistent behavior and perceived discrimination against some languages have not been mitigated by repeated explanations that the relationships between "decorated" lower-case characters and their upper-case equivalent are often sensitive to language and locality and therefore not deterministic with information available to DNS servers.

3.3. Matching Part II: Non-ASCII ("internationalized") Domain Name Labels

Quite independent of the case-sensitivity problem, one of the fundamental properties of Unicode [Unicode] is that some abstract characters can be represented in multiple ways, such as by a single, precomposed, code point or by a base code point followed by one or more code points that specify combining characters. While Unicode Normalization can be used to eliminate many (but not all) of those distinctions for comparison (matching) purposes, it is best applied during matching rather than by changing one string into another. The first version of IDNA ("IDNA2003") made the choice to change strings during processing for either storage or retrieval [RFC3490] [RFC3491]; the second ("IDNA2008") required that all strings be normalized [RFC5891]. Neither is optimal, if only because transforming the strings themselves implies that the input string in an application may not be the same as the string used in processing and perhaps later display.

It would almost certainly be preferable, and more consistent with Unicode recommendations, to use normalization (and perhaps other techniques) at matching time rather than altering the strings at all,

even if there were still only a single matching algorithm, i.e., normalization were added to the existing ASCII-only case folding. However, even Unicode's discussion of normalization [Unicode-UAX15] indicates that there are special, language-dependent, cases (the most commonly-cited example is the dotless "i" (U+0131)). Not only does the DNS lack any information about languages that could be used in a mapping algorithm, but, as long as there is a requirement that there be only one mapping algorithm for the entire system, that information could not be used even if it were available. One could imagine a successor system that would use information stored at nodes in the hierarchy to specify different matching rules for subsidiary nodes (or equivalent arrangements for non-hierarchical systems). It is not clear whether that would be a good idea, but it certainly is not possible with the DNS as we know it.

3.4. Matching Part III: Label Synonyms, Equivalent Names, and Variants

As the initial phases of work on IDNs started to conclude, it became obvious that the nature and evolution of human language and writing systems required treating some names as "the same as" others. The first important example of this involved the relatively recent effort to simplify the Chinese writing system, thereby creating a distinction between "Simplified" and "Traditional" Chinese even though the meaning of the characters remained the same in almost all cases (in so-called ideographic character sets, characters have meaning rather than representing sounds). A joint effort among the relevant country code top level domain (TLD) registries and some other interested parties produced a set of recommendations for dealing with the issues with that script [RFC3743] and introduced the concept of "variant" characters and domain names.

However, when names are seen as having meanings, rather than merely being mnemonics, and especially when they represent brands or the equivalent, or when spelling for a particular written language is not completely standardized, there is an immediate demand to treat different strings as exact equivalents. As a trivial English-language example, it is widely understood that "colour" and "color" represent the same word, so does that imply that, if they are used as DNS labels in domain names all of whose other labels are identical, should the two domain names be treated as identical? Examples for other languages or writing systems, especially ones in which some or all markings that distinguish characters by sound or that change the pronunciation of words are optional, are often more numerous and more problematic than national spelling differences in English, but they are harder to explain to those unfamiliar with those other languages or writing systems (and hard to illustrate in ASCII-only Internet-Drafts and RFCs). Although approximations are possible, the DNS cannot handle that requirement: not only do its aliasing mechanisms

(CNAME, DNAME, and various proposals for newer and different types of aliasing [DNS-Aliases] [DNS-BNAME], not provide a strong enough binding, but the ability to use those aliases from a subtree controlled by one administrative entity to that of another one, implies that there is little or no possibility of the owner (in either the DNS sense or the registrar-registrant one) of a particular name to control the synonyms for it. Some of that issue can be deal with at the application level, e.g., by redirects in web protocols, but taking that approach, which is the essential characteristic of "if both names belong to the same owner, everything is ok" approaches, results in names being handled in inconsistent ways in different protocols.

A different way of looking at part of this issue (and, to some degree, of the one discussed above in Section 3.3) is that these perceived equivalences and desired transformations are context-dependent, but the DNS resolution process is not [RFC6912].

Similar problems arise as people notice that some characters are easily mistaken with others and that might be an opportunity for user confusion and attacks [CACM-Homograph]. The most common proposed solution within the DNS context has been to treat these cases, as well as those involving orthographic variations, as "Variants" and either ban all but one (or a few) of the possible labels from the DNS (possibly on a first come first served basis) or by ensuring that any collection of such strings that are delegated as assigned to the same ownership (see above). Neither solution is completely satisfactory: if all but one string is excluded, users who guess at a different form, perhaps in trying to transcribe characters from written or printed form, don't find what they are looking for and, as pointed out above, "same ownerwhip" is sufficient only with carefully-designed and administered applications protocol support and sometimes not then.

Some of these issues are discussed at more length in an ICANN report [ICANN-VIP].

3.5. Query Privacy

There has been growing concern in recent years that DNS queries occur in clear text on the public Internet and that, if those queries can be intercepted, they can expose a good deal of information about interests and contacts that could compromise individual privacy. While a number of proposals, including query name minimization [RFC7816] have been made to mitigate that problem, it does not appear that any of them are as satisfactory as a system with query privacy designed in might be. More general tutorials on this issue have appeared recently [Huston-DNSPrivacy]

3.6. Alternate Name Spaces for Public Use in the DNS Framework: The CLASS Problem

The DNS standards include specification of a CLASS value to "identify a protocol family or instance of a protocol" RFC 1034, Section 3.6 and elsewhere [RFC1034]. While it was used effectively in the early days of the DNS to manage different protocol families within the same administrative environment, recent attempts to use it to either partition the DNS namespace in other ways such as for non-ASCII names (partially to address the issues in Section 3.2 Section 3.3) or to use DNS mechanisms for entirely different namespaces have exposed fundamental problems with the mechanism [Sullivan-Class], leading to recommendations that it be dropped entirely.

Whether either the function CLASS was originally intended to provide or the ones for which there have been attempts to use it more recently are actually needed is a separate question; it is clear that the current DNS technical and administrative model is unsuitable for either function.

3.7. Loose Synchronization

The DNS model of master and slave servers, with the latter initiating updates based on TTL values, together with more local caches, depends heavily on an approach that has come to be called "loose synchronization", i.e., that there can be no expectation that all of the servers that might reasonably answer a query will have exactly the same data unless those data have been unchanged for a rather long period. Put differently, if some or all of the records associated with a particular node in the DNS (informally, a fully-qualified domain name (FQDN)) change, one cannot expect those changes to be propagated immediately.

That model has worked rather well since the DNS was first deployed, protecting the system from requirements, that are typical where simultaneous update of multiple systems is needed, such as elaborate locking, complex update mechanisms, or journaling. As has often been pointed out with the Internet, implementation and operational complexity are often the enemy of stability, security, and robustness. Loose synchronization has helped keep the DNS as simple and robust as possible.

A number of recent ideas about using the DNS to store data that change very rapidly and where the changes are important are, however, largely incompatible with loose synchronization. Efforts to use very short (or zero) TTLs to simulate nearly-simultaneous updating may work up to a point but appear to impose very heavy loads on servers and distribution mechanisms that were not designed to accommodate

that style of working. Similar observations can be made about attempts to use dynamic, "server-push", updating rather than the traditional DNS mechanisms. While those might work better than ordinary short TTLs and update mechanisms as specified in RFC 1034 and 1035, they imply that a "master" server must know the identities of (and have real time access to all of) its slaves, defeating many of the advantages of caching, particularly those associated with reduction of query traffic across the Internet.

3.8. Private Name Spaces and Special Names

Almost since the DNS was first deployed, there have been situations in which it is desirable to use DNS-like names, and often DNS resolution mechanisms or modifications of them, with name spaces for which globally-available and consistent resolution using the public DNS is either unfeasible or undesirable (and for which the use of CLASS is not an appropriate mechanism). The need to isolate names and addresses on LANs from the public Internet, typically via "split horizon" approaches, is one example of this requirement although often not recognized as such. Another example that has generated a good deal of controversy involves "special names" -- labels or pseudo-labels, often in TLD positions, that signal that the full name should not be subject to normal DNS resolution or other processing [RFC6761].

Independent of troublesome policy questions about who should allocate such names and the procedures to be used, they almost inherently require either a syntax convention to identify them (there actually was such a convention, but it was abandoned many years ago and there is no plausible way to re-institute it) or tables of such names that are known to, and kept updated on, every resolver on the Internet, at least if spurious queries to the root servers are to be avoided.

If the DNS were to be redesigned and replaced, we could recognize this requirement as part of the design and handle it much better than it is possible to handle it today.

3.9. Alternate Response Encodings

The DNS specifies formats for queries and data responses, based on the state of the art and best practices at the time it was designed. Recent work has suggested that there would be significant advantages to supporting at least a description of the DNS messages in one or more alternate encodings, such as JSON [Hoffman-DNS-JSON]. While that work has been carefully done to avoid requiring changes to the DNS, much of the argument for having such a JSON-based description format could easily be turned into an argument that, if the DNS were

being revised, that format might be preferable as a more direct alternative to having DNS queries and responses in the original form.

3.10. Distribution and Managment of Root Servers

The DNS model requires a collection of root servers that hold, at minimum, information about top-level domains. Over the years, that requirement has evolved from a technically fairly minor function, normally carried out as a service to the broader Internet community and its users and systems, to a subject that is intensely controversial with regard to who should control those servers, how they should be distributed and where they should be located. While a number of mechanisms have been proposed and one (anycast [RFC7094]) is in very active use to mitigate some of the real and perceived problems, it seems obvious that a DNS successor, designed for today's perceived requirements, could handle these problems in a technically more appropriate and less controversial way.

3.11. Identifiers Versus Brands and Other Convenience Names

A key design element of the original network object naming systems for the ARPANET, largely inherited by the DNS, was that the names were identifiers and their being highly distinguishable and not prone to ambiguity was important. That led to very restrictive rules about what could appear in a name. In the case of the host table, the restrictions that came to the DNS (largely via SMTP) as the "preferred syntax" [RFC 1034 Section 3.5] or what we now often call the letter-digit-hyphen (LDH) rule. Similar rules to make identifiers easier to use, less prone to ambiguity, or less likely to interfere with syntax in more formal languages occur frequently. For example, almost every programming language has restrictions on what can appear in an identifier and Unicode provides general recommendations about identifier composition [Unicode-USA31]. Both are quite restrictive as compared to the number of characters and total number of strings that can be written using that character coding system.

In the last decade or two, another perspective has emerged, largely without being explicitly understood or acknowledged. In it, the DNS is really (and primarily) a system for expressing thoughts and concepts. Those include free expression of ideas in as close to natural language as possible as well as representation of product names and brands. That view requires letter-like characters that might not be reasonable in identifiers along with a variety of symbols and punctuation and might require indicators of preferred type styles to provide information in a form that exactly matches personal or legal preferences. That perspective would argue for standardizing word and sentence separators, removing the 63 octet per

label limit and probably the limit of 255 octets on the total length of a domain name, and maybe even eliminating the hierarchy or allowing separators for labels in presentation form (now fixed at "." for the DNS) to be different according to context. At least it suggests that the original design was defective in not prioritizing those uses over support for unique and unambiguous identifiers.

So we have two, or, depending on how one counts, three very different use cases. The historical one is support for unique identifiers. The other is expression of ideas and, if one considers it separate, presentation of brand and product names. Because they inherently involve different constraints, priorities, and success criteria, these perspectives are, at best, only loosely compatible.

We cannot simultaneously optimize both the identifier perspective and either or both of the others in the same system. At best, there are some complex trade-offs involved. Even then, it is not clear that the same DNS (or other system) can accommodate all of them. Until we come to terms with that, the differences manifest themselves with friction among communities, most often with tension between "we want to do (or use or sell) these types of labels" and "not good for the operational Internet or the DNS".

3.12. A Single Hierarchy with a Centrally-controlled Root

A good many Internet policy discussions in the last two decades have revolved around such questions of how many top level domains there should be and what they should be, who should control them and how, how (or if) their individual operations and policy decisions should be accountable to others, and what processes should be used (and by what entities or organizational structures) to make those decisions. Several people have pointed out that, if we were designing a next-generation DNS using today's technology, it should be possible to remove the technical requirement for a central authority over the root (some people have suggested that blockchain approaches would be helpful for this purpose). Whether that would be desirable on not is fairly obviously a question of perspective and priorities

3.13. Scaling of Reputation and Other Ancillary Information

The original design for DNS administration, reflected in RFC 1591 [RFC1591] and elsewhere, assumed that all domains would exhibit a very high level of responsibility toward and for the community and that level of responsibility would be enforced if necessary. More recent decisions have taken things in the direction of "registrant beware" and even "user and applications beware". One possible approach to the problems, especially security problems, that are enabled by the new environment is to establish reputation systems

associated with clearly-defined administrative boundaries and with warnings to users.

The IETF DBOUND WG [IETF-DBOUND] addressed ways to establish and document boundaries more precise than simple dependencies on TLDs but it was not successful in producing a standard. A TLD reputation-based approach was adopted by some web browsers after IDNs and a growing number of gTLDs were introduced; that approach was based on a simple list and does not scale to the current size of the DNS or even the DNS root.

4. Searching and the DNS - An Historical Note

Some of the issues identified above might reasonably be addressed, not by changing the DNS itself but by changing our model of what it is about and how it is used. Specifically, one key assumption when the DNS (and the host table system before it) was designed was that it was a naming system for network resources, not, e.g., digital content. As such, exact matching was important, it was reasonable to have labels treated as mnemonics that did not necessarily have linguistic or semantic meaning except to those using them, and so on. A return to that model, presumably by having user-facing applications call on an intermediate layer to disambiguate user-friendly names and map them to DNS names (network object locators) would significantly reduce stress on the DNS and would also allow dealing with types of matching and similar or synonymous strings that cannot be handled algorithmically no matter how much DNS matching rules were altered.

In the early part of the last decade, the IETF explored that approach a little bit in the context of IDNs and what were then called "Internet keywords" [DNS-search]. It may be time to look at that approach again and more deeply in the context of more recent developments.

It is worth noting that, while that "search" approach, or some other approach that abstracted and separated several of the issues identified in Section 3 from the DNS protocol and database themselves, it does not address all of them. At least some elements of several of those issues, such as the synchronization ones described in Section 3.7, are inherent in the DNS design and, if we are not going to replace the DNS, we had best get used to them.

5. Acknowledgements

Many of the concerns and ideas described in this document reflect conversations over a period of many years, some rooted in DNS "keyword" and "search" discussions that paralleled the development of Internationalized Domain Names (IDNs). Conversations with, or

writing of, Rob Austein, Christine Borgman, Vint Cerf, Lyman Chapin, Patrik Faltstrom, Geoff Huston, Xiaodong Lee, Karen Liu, Yaqub Mueller, Andrew Sullivan, Paul Twomey, Suzanne Woolf, Jiankang Yao, other participants in the circa 2003 "DNS Search" effort and in the ICANN SSAC Working Party on IDNs, and some others whose names were sadly forgotten were particularly important to either the content of this document or the motivation for writing it even though they may not agree with the conclusions I have reached and bear no responsibility for them.

Many of the subsections of Section 3 were extracted from comments first made in conjunctions with recent email discussions. Comments from Suzanne Woolf about an early draft were particularly important.

6. IANA Considerations

[[CREF1: RFC Editor: Please remove this section before publication.]]

This memo includes no requests to or actions for IANA.

7. Security Considerations

From both security and privacy perspectives, a replacement for the DNS would not have to go very far to be a significant improvement.

8. References

8.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <http://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <http://www.rfc-editor.org/info/rfc1035>.

8.2. Informative References

[CACM-Homograph] Gabrilovich, E. and A. Gontmakher, "The Homograph Attack", Communications of the ACM 45(2):128, February 2002, <http://www.cs.technion.ac.il/~gabr/papers/homograph_full.pdf>.

[DNS-Aliases]
Woolf, S., Lee, X., and J. Yao, "Problem Statement: DNS Resolution of Aliased Names", March 2011, <https://datatracker.ietf.org/doc/draft-ietf-dnsext-aliasing-requirements/>.

[DNS-BNAME]
Yao, J., Lee, X., and P. Vixie, "Bundled DNS Name Redirection", May 2016, <https://datatracker.ietf.org/doc/draft-yao-dnsext-bname/>.

[DNS-search]
IETF, "Internet Resource Name Search Service", 2003, <https://datatracker.ietf.org/wg/irnss/about/>.

While it met several times informally and as one or more BOFs, this effort never really got off the ground. That was due in part to the IETF decision to go forward with the IDNA approach and in part by signs that the "keyword" efforts were beginning to fall apart.

[Hoffman-DNS-JSON]
Haffman, P., "Representing DNS Messages in JSON", May 2017, <https://datatracker.ietf.org/doc/draft-hoffman-dns-in-json/>.

[Huston-DNSPrivacy]
Huston, G. and J. Silva Dama, "DNS Privacy", Internet Protocol Journal Vol 20, No 1, March 2017, <http://ipj.dreamhosters.com/wp-content/uploads/issues/2017/ipj20-1.pdf>.

[ICANN-VIP]
ICANN, "IDN Variant Issues Project: Final Integrated Issues Report Published and Proposed Project Plan for Next Steps is Now Open for Public Comment", February 2012, <https://www.icann.org/news/announcement-2012-02-20-en>.

[IETF-DBOUND]
IETF, "Domain Boundaries (dbound)", 2017, <https://datatracker.ietf.org/wg/dbound/about/>.

[RFC0799] Mills, D., "Internet name domains", RFC 799, DOI 10.17487/RFC0799, September 1981, <http://www.rfc-editor.org/info/rfc799>.

[RFC0810] Feinler, E., Harrenstien, K., Su, Z., and V. White, "DoD Internet host table specification", RFC 810, DOI 10.17487/RFC0810, March 1982, <http://www.rfc-editor.org/info/rfc810>.

[RFC0881] Postel, J., "Domain names plan and schedule", RFC 881, DOI 10.17487/RFC0881, November 1983, <http://www.rfc-editor.org/info/rfc881>.

[RFC0882] Mockapetris, P., "Domain names: Concepts and facilities", RFC 882, DOI 10.17487/RFC0882, November 1983, <http://www.rfc-editor.org/info/rfc882>.

[RFC0952] Harrenstien, K., Stahl, M., and E. Feinler, "DoD Internet host table specification", RFC 952, DOI 10.17487/RFC0952, October 1985, <http://www.rfc-editor.org/info/rfc952>.

[RFC0953] Harrenstien, K., Stahl, M., and E. Feinler, "Hostname Server", RFC 953, DOI 10.17487/RFC0953, October 1985, <http://www.rfc-editor.org/info/rfc953>.

[RFC0974] Partridge, C., "Mail routing and the domain system", STD 10, RFC 974, DOI 10.17487/RFC0974, January 1986, <http://www.rfc-editor.org/info/rfc974>.

[RFC1123] Braden, R., Ed., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, DOI 10.17487/RFC1123, October 1989, <http://www.rfc-editor.org/info/rfc1123>.

[RFC1591] Postel, J., "Domain Name System Structure and Delegation", RFC 1591, DOI 10.17487/RFC1591, March 1994, <http://www.rfc-editor.org/info/rfc1591>.

[RFC2671] Vixie, P., "Extension Mechanisms for DNS (EDNS0)", RFC 2671, DOI 10.17487/RFC2671, August 1999, <http://www.rfc-editor.org/info/rfc2671>.

[RFC3490] Faltstrom, P., Hoffman, P., and A. Costello, "Internationalizing Domain Names in Applications (IDNA)", RFC 3490, DOI 10.17487/RFC3490, March 2003, <http://www.rfc-editor.org/info/rfc3490>.

[RFC3491] Hoffman, P. and M. Blanchet, "Nameprep: A Stringprep Profile for Internationalized Domain Names (IDN)", RFC 3491, DOI 10.17487/RFC3491, March 2003, <http://www.rfc-editor.org/info/rfc3491>.

[RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", RFC 3596, DOI 10.17487/RFC3596, October 2003, <http://www.rfc-editor.org/info/rfc3596>.

[RFC3743] Konishi, K., Huang, K., Qian, H., and Y. Ko, "Joint Engineering Team (JET) Guidelines for Internationalized Domain Names (IDN) Registration and Administration for Chinese, Japanese, and Korean", RFC 3743, DOI 10.17487/RFC3743, April 2004, <http://www.rfc-editor.org/info/rfc3743>.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, DOI 10.17487/RFC5890, August 2010, <http://www.rfc-editor.org/info/rfc5890>.

[RFC5891] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, DOI 10.17487/RFC5891, August 2010, <http://www.rfc-editor.org/info/rfc5891>.

[RFC6761] Cheshire, S. and M. Krochmal, "Special-Use Domain Names", RFC 6761, DOI 10.17487/RFC6761, February 2013, <http://www.rfc-editor.org/info/rfc6761>.

[RFC6912] Sullivan, A., Thaler, D., Klensin, J., and O. Kolkman, "Principles for Unicode Code Point Inclusion in Labels in the DNS", RFC 6912, DOI 10.17487/RFC6912, April 2013, <http://www.rfc-editor.org/info/rfc6912>.

[RFC7094] McPherson, D., Oran, D., Thaler, D., and E. Osterweil, "Architectural Considerations of IP Anycast", RFC 7094, DOI 10.17487/RFC7094, January 2014, <http://www.rfc-editor.org/info/rfc7094>.

[RFC7719] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", RFC 7719, DOI 10.17487/RFC7719, December 2015, <http://www.rfc-editor.org/info/rfc7719>.

[RFC7816] Bortzmeyer, S., "DNS Query Name Minimisation to Improve Privacy", RFC 7816, DOI 10.17487/RFC7816, March 2016, <http://www.rfc-editor.org/info/rfc7816>.

[Sullivan-Class]
Sullivan, A., "The DNS Is Not Classy: DNS Classes Considered Useless", July 2016, <https://datatracker.ietf.org/doc/draft-sullivan-dns-class-useless/>.

[Unicode] The Unicode Consortium, "The Unicode Standard, Version 9.0.0,", ISBN 978-1-936213-13-9, 2016, <http://www.unicode.org/versions/Unicode9.0.0/>.

[Unicode-UAX15]
Davis, M. and K. Whistler, "Unicode Normalization Forms", February 2016, <http://unicode.org/reports/tr15/>.

[Unicode-USA31]
Davis, M., "Unicode Identifier and Pattern Syntax", May 2016, <http://unicode.org/reports/tr31/>.

Author's Address

John C Klensin
1770 Massachusetts Ave, Ste 322
Cambridge, MA 02140
USA

Phone: +1 617 245 1457
Email: john-ietf@jck.com