This document states a definition of Domain Name beyond the use of the term within the Domain Name System. The document includes a survey of the diverse ways Domain Names have been interpreted within various protocols over time. The purpose of this is to give a solid foundation for work on Domain Names across all protocols making use of Domain Names.

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 http://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 March 30, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

0.  NOTE TO RFC EDITOR AND REVIEWERS .............................. 1
1.  Introduction .................................................. 1
2.  Reaching a definition of a Domain Name ...................... 1
3.  Subtypes of Domain Names ....................................... 1
4.  Interoperability Considerations ............................... 1
5.  Acknowledgements .............................................. 1
6.  IANA Considerations ............................................ 1
7.  Security Considerations ....................................... 1
8.  References .................................................... 1
9.  Authors’ Addresses ............................................ 1
0. NOTE TO RFC EDITOR AND REVIEWERS

A suitable venue for discussion of this document is the dnsop working group. Private comments may also be directed at the editor.

This document will be presented to the Dispatch WG as it relates to the use of identifiers across protocols, despite perceptions that the DNS defines Domain Names.

This section (and sub-sections) should be removed prior to publication.

1. Introduction

Two or three decades into the history of Domain Names, a popular notion has taken hold that Domains Names were defined and specified STD 13, the definition of the Domain Name System (DNS). The definitions within RFC 1034 and RFC 1035 have become the apparently-authoritative source for discussions on what is a Domain Name.

The truth is, RFC 1034 and RFC 1035 do not define Domain Names, those documents define only how Domain Names are used and processed in the DNS. But the way in which those RFCs are written seem to lend to the confusion.

Throughout this document the term "Domain Names" is capitalized to emphasize the concept of the names and DNS is used to describe the protocol and algorithms described in STD 13, including any applicable updates, related standards track documents and experimental track documents.

The term domain is a generic term. And there are many naming systems in existence. The use of the term Domain Names in this document refers to the roughly-defined set of protocols and their applications use of a naming structure that is prevalent amongst many protocols defined in IETF RFC documents.

If there is a use of Domain Names not listed here it is merely an omission. The goal in this document is to provide a survey that is sufficient to avoid hand-waving arguments, recognizing the dimishing return in trying to build a complete roster of uses of Domain Names. If there are omissions that ought to be included, please send references for the use case to the author (while this is an Internet Draft, that is).

1.1 Deconstructing RFC 1034’s Introduction

RFC 1034, section 2 begins with this text:

"This RFC introduces domain style names, their use for Internet mail and host address support, and the protocols and servers used to implement domain name facilities."

Which seems to indicate that RFC 1034 is the origin of Domain Names. Immediately following is section 2.1, entitled "The history of domain names" which includes the following text.

"The result was several ideas about name spaces and their management [IEN-116, RFC-799, RFC-819, RFC-830]. The proposals varied, but a
common thread was the idea of a hierarchical name space, with the hierarchy roughly corresponding to organizational structure, and names using "." as the character to mark the boundary between hierarchy levels. A design using a distributed database and generalized resources was described in [RFC-882, RFC-883]. Based on experience with several implementations, the system evolved into the scheme described in this memo.

The DNS as it is known today did not invent Domain Names (work on the Simple Mail Transfer Protocol did) and, for what it’s worth, wasn’t the first attempt at an Internet naming system (described in RFC 819 "The Domain Naming Convention for Internet User Applications").

One important phrase to keep in mind is:

"To simplify implementations,"

which appears in both RFC 1034 and RFC 1035 as well as their predecessors RFC 882 and RFC 883. This gives credence to the notion that Domain Names exist beyond the DNS.

1.2 Purpose of the Document

This document has a goal of establishing a definition of Domain Names for the purpose of continuing efforts to innovate in naming systems. The path by which this document aims to meet the goal is to describe in more detail the original definition of Domain Names (which never seems to have been documented in an RFC) and then to illustrate how Domain Names have been interpreted in the DNS, SMTP, X.509 Certificate (PKIX), URNs and other environments.

1.3 Why Write This Now?

RFC 6761 defines "Special Use Domain Names" which has engendered much confusion of the role of Domain Names with respect to the DNS, particularly Top-Level Domains, which have come to have special meanings.

One of the outcomes from the recent discussion on RFC 6761 [this section is written informally] is an assumption that client software, that is an application receiving a string that looks like a name without any supplemental context, will be able to determine how to treat the string within the appropriate naming context. Within the familiar world of domain names today, certain top-level names are recognized as special (such as those whose last label is local). To date there are a limited number of special names, as the number scales up, there will be more work for client software developers. (See section 4.1 of this document.)

Note that whether or not the last label takes on this role is suggested here solely as an example. Whether it does or not is a something for "a solution" for which we don’t even have the starting point nailed down. Hence, it isn’t clear 4.1 should be in this document.

Given a lack of an explicit starting point, meaning a clear definition of what Domain Name means, this document is striving to provide a foundation.

2. Reaching a definition of a Domain Name
Domain Names emerged from the need to build a hierarchy around the growing number of identified hosts exchanging email. RFC 788, "SIMPLE MAIL TRANSFER PROTOCOL", explains, in its section 3.7:

"At some not too distant future time it might be necessary to expand the mailbox format to include a region or name domain identifier. There is quite a bit of discussion on this at present, and is likely that SMTP will be revised in the future to take into account naming domains."

2.1 Historical Perspective

Knowing the origins of a concept helps setting the correct boundaries for discussion. The past isn’t meant to restrict the future but meant to help provide a context, include forgotten ideas, and help identify rational for scope creep.

RFC 799 "Internet Name Domains" has (arguably) the first formation of what is a Domain Name:

"In its most general form, a standard internet mailbox name has the syntax

\(<user>\).<host>@<domain>\)

where \(<user>\) is the name of a user known at the host \(<host>\) in the name domain \(<domain>\)."

Prior to this, domain referred to principally an administrative domain, such as the initial organizations involved in networks at the time.

RFC 801 "NCP/TCP TRANSITION PLAN" contains this, indicating the passage from the host tables:

"It might be advantageous to do away with the host name table and use a Name Server instead, or to keep a relatively small table as a cache of recently used host names."

RFC 805 "Computer Mail Meeting Notes" contains this:

"The conclusion in this area was that the current "user@host" mailbox identifier should be extended to ‘user@host.domain’ where ‘domain’ could be a hierarchy of domains."

RFC 819 "The Domain Naming Convention for Internet User Applications" contains this:

"A decision has recently been reached to replace the simple name field, "<host>", by a composite name field, '<domain>'"

A domain name began to take on its current form:

"Internet Convention: Fred@F.ISI.ARPA"

In addition, "simple name" is defined as what we now call a label, and a "complete (fully qualified) name" is defined as "concatenation of the simple names of the domain structure tree nodes starting with its own name and ending with the top level node name". Noticeably absent is a terminating dot or any mention or representation of a
RFC 819 defines ARPA as a top-level name (as opposed to top-level domain name). This is an early mention of the role of top-level names.

This walk through history relies solely on the record left behind inside RFCs. The precise chain of events is likely slightly different and nuanced. The point of the exercise is to show that Domain Names are a concept the emerged over time, spawned the DNS with its domain names, a definition of host names derived from the host tables, and was heavily influenced by SMTP as the driving application. The definition of the FTP protocol, originally defined in RFC 959 "FILE TRANSFER PROTOCOL", never mentions hosts, domains or host names. But no formal definition of Domain Names has been written and recorded.

2.2 Newly Stated Definition of "Domain Name"

Looking through the early documents, and using the experience of the past decades, this new definition of Domain Names is stated:

A Domain Name is a sequence of labels concatenated by a designated separating character. The Domain Name Space is organized in a strict hierarchical manner with a recognized root Domain Name. The organization follows the rules of tree structure as defined by the field of graph theory in mathematics [Diestel].

Each label represents a node in a conceptual tree. The sequence of labels is concatenated from the deepest node in the tree up to the root node. "Fully qualified" refers to a sequence that ends with the root node.

When considering a fully qualified name, the first label of the name is the name of the deepest node in the tree, the last label is the name of the node is the root. The top-level label, top-level name, or top-level domain is the label just before the root (or last) label.

Excluded from the definition is the appearance or representation of the labels, the designated separator character’s representation, the ordering of the sequence in appearance, such as left-to-right or right-to-left, nor the written script nor encoding. The definition is purely conceptual.

In RFC 819 "Simple Mail Transfer Protocol", the designated separating character is the dot (‘.’) as represented in the ASCII [RFC 20] [ANSIX34] character set. This is the earliest application definition of how it represents Domain Names.

2.3 Limitation

There are many ways to build a name space, Domain Names are just one example. Domain Names are intended to build a name space that can scale tremendously as opposed to a name space for closed cluster of involved objects. Domain Names are used across many protocols defined inside and outside the IETF and have been defined to interoperate across implementations and protocols. This does not make Domain Names an official or required standard despite the name space’s widespread use.
2.4 Is This a Domain Name?

In the vein of questions like "but is it art?" as to whether an object is art worthy of display in a museum or just a ordinary piece of trap, one can question whether any string with a dot in it is a Domain name. For example, is this multi-sentence paragraph a domain name? It has characters and dots in it.

The important question is not whether a string is an example of a Domain Name based on its appearance. The use of a string is what makes it a Domain Name. A path name of a file with an extension looks like a Domain Name with the extension separated by a dot, if one allows the directory separating character (a '/' perhaps) as a legal member of a label. Within an OS, this is a file/path name. In a protocol it might be used as if it were a domain name.

3. Subtypes of Domain Names

Subtypes of Domain Names have come to be defined for different protocols, evolving and sometimes building on previous definitions.

3.1 Domain Names as Restricted for DNS

The DNS protocol place size restrictions on Domain Names and defines rules for matching domain names, treating sets of Domain Names as equivalent to each other. (This matching refers to treating upper case and lower case ASCII letters as equivalent.) The DNS defines the format used to transmit the names across the network as well as rules for displaying them inside text zone files. The DNS creates the notion that names are assigned by an authority per zone.

Placing size restrictions on Domain Names is significant in reducing the overall population of names that can be represented in the DNS. The matching rules have the effect of creating (to use a term from graph theory) cliques, distorting the tree-nature of the Domain Name graph. A clique is a completely connected sub-graph implying cyclic paths, a tree is a graph that is acyclic. In sum, the treatment of ASCII (and only ASCII) cases as equivalent is a distortion of the Domain Name hierarchy.

DNS defines two formats for domain names. One is the "on-the-wire" format used inside messages, a flags-and-length octet followed by some count of octets for each label with the final length of 0 representing the root. The other is a version that can be rendered in printable ASCII characters, complete with a means to represent other characters via an escape sequence. This does not alter the Domain Name concept but has implications when it comes to interoperating with other protocol definitions of their domain name use.

DNS assumes that there is, in concept, a central authority creating names within the DNS management structure (called a zone). Although the DNS does not define how a central authority is implemented nor how it coins names, the names have to come from a single point to appear in a zone. There are other means for claiming names, an example will be mentioned later.

A DNS domain name "192.0.2.1." can be configured and used in the protocol. The usefulness of this is limited by the concerns described later on in Interoperability Considerations. An outcome of that the convention of representing the Domain Name "192.0.2.1." as
DNS domain names have become the dominant definition of domain names due to the success (scale) of the DNS on the public Internet. Many protocols interact with the DNS but instead of supporting the complete definition of DNS domain names, the protocols rely on a subset more commonly called host names.

### 3.2 Host Names

Work on the definition of a host name began well before the issuance of the STD 13 documents defining DNS. The rules for the Preferred Syntax in RFC 1034 conform to the host name rules outlined in RFC 952. The host name definition was presented again in RFC 1123 "Requirements for Internet Hosts -- Application and Support" (which is part of STD 3). In section 2.1 of RFC 1123, one (of two mentions) definition of host name is presented, noting that the definition is a relaxation of what is in RFC 952.

Host names are subsets of DNS domain names in the sense that the character set is limited. In particular, only "let" (i.e., presumably letters a-z), "digits" and "hyphen" can be used, with hyphen only internal to a label. (This description is meant to be illustrative, not normative. See the grammar presented on page 5 of RFC 952 for specifics.) RFC 1945 "Hypertext Transfer Protocol -- HTTP/1.0", Section 3.2.2 "http URL" specifically references section 2.1 of RFC 1123. The reference is explicit.

RFC 5321 "Simple Mail Transfer Protocol" refers to RFC 1035 for a definition of domain names but includes text close to what is in the previous paragraph, noting that domain names as used in SMTP refer to both hosts and to other entities. RFC 5321 updates RFC 1123, but does not cite the latter for a definition of host names. RFC 5321 additionally requires brackets to surround address literals, referring to the use case as an "alternative to a domain name."

### 3.3 URI Authority and Domain Names

In RFC 3986, also known as STD 66, "Uniform Resource Identifier (URI): Generic Syntax", mentions in its section 3.2.2 (page 20) that the host subcomponent of the URI Authority (section 3.2) "should conform to the DNS syntax". This comes after discussion that the host subcomponent is not strongly tied to the DNS, i.e., names can be managed via concept other than the DNS. There’s no discussion on the rationale but this enables the reuse of code parsing and marshalling the host subcomponent between different Domain Name environments.

This reinforces the notion that there’s a need to understand how Domain Names interoperate amongst protocols and applications. And reinforces the need to derive or make explicit a way for client software to know how to resolve a name, that is, convert a name into a network address.

### 3.4 Internet Protocol Address Literals

The above definition includes address literals such as 192.0.2.1 for IPv4 and even IPv6 literals such as ::ffff:192.0.2.1. Yes, these qualify as Domain Names. In some protocols, these domain names are specified as being preceded by a "#" (find this and cite) or encased in square brackets "[" and "]" (SMTP mentioned already).
3.5 Internationalized Domain Names in Applications

The original definitions of Domain Names (such as DNS domain names and host names) assumed the ASCII character set. Specifically, making the labels case insensitive prohibited a straightforward use of any method of representation of non-ASCII characters.

RFC 5890 "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", with associated other documents, defines IDNA2008 as a convention for handling non-ASCII characters in DNS domain names. In figure 1 of that document, the sets of legal DNS domain name formats are defined. Noted in the footnotes of the figure, applications unaware of IDNA2008 cannot distinguish the sub sets defined by the document meaning this definition is not an alteration of Domain Names, but, like host names, yet another subset of DNS domain names.

3.6 Restricted for DNS Registration

RFC 4290 "Suggested Practices for Registration of Internationalized Domain Names (IDN)" presents reasons why registration of DNS domain names is restricted, in the context of IDN. (That RFC refers to an older form than IDNA2008, but the concepts still apply.) This is yet another convention related to DNS domain names, excluding names that would lead to undesirable outcomes.

3.7 Tor Network Names

The Tor network is an activity organized by the Tor Project, Inc., described on its main web page "https://www.torproject.org/index.html.en". One component of the network are Domain Names ending in ".onion". (There are other suffixes in use, but it isn’t very clear how they are used, defined or whether they are active.)

The way in which Domain Names are used in Tor is described in two web documents "Tor Rendezvous Specification" [RENDEV] and "Special Hostnames in Tor" [OHOST] available from the project’s website.

Syntactically, a Tor domain name fits within the DNS domain name definition but the manner of assignment is different in a manner incompatible with the DNS. (Not better or worse, still significantly different.) Tor domain names are derived from cryptographic keys and organized by distributed hash tables, instead of assigned by a central authority per zone.

3.8 X.509

In RFC 5280 "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", section 4.2.1.6 "Subject Alternative Name" a dNSName is defined to be a host name, with the further restriction that the name " " cannot be used.

3.9 Multicast DNS

Multicast DNS uses a name space ending with ".local." as described in RFC 6762, "Multicast DNS". The rules for Multicast DNS domain names differ from DNS domain names. Multicast DNS domain names are encoded as Net-Unicode as defined in RFC5198 " Unicode Format for Network Interchange" with the DNS domain name tradition of case folding the
ASCII letters when matching names. Appendix F of RFC 6762 gives an explanation of why the punycode algorithm is not used.

3.10 /etc/hosts

The precursor to DNS, host tables, still exists in remnants in many operating systems. There are library functions, used by applications to resolve DNS domain names, that can return names of arbitrary length (meaning, for example longer than what DNS domain names are defined to be).

RFC 3493, "Basic Socket Interface Extensions for IPv6", addresses this in Section 6, further documentation can be found as part of The Open Group Base Specifications Issue 7 [IEEE1003.1] and Microsoft Winsock Functions [WINSOCK].

3.11 Other Protocols

This section is used to enumerate other protocols that use Domain Names but in general do not impose any other restrictions that what has been mentioned above.

SSH [RFC 4251 "The Secure Shell (SSH) Protocol Architecture"] uses host names, using the name when storing public keys of hosts. SSH clients, not necessarily the protocol, illustrate how applications juggle the different forms of Domain Names. SSH can be invoked to open a secure shell with a host via its DNS domain name/host name or it can be used to open a secure shell with a host via its Multicast DNS domain name.

FTP [RFC 959 " FILE TRANSFER PROTOCOL (FTP)"] is silent on domain names but client implementations of the protocol behave as SSH clients, being aware the differences between definitions of Domain Names (DNS vs. MulticastDNS).

DHCP [RFC 2131 "Dynamic Host Configuration Protocol"] includes domain names in it’s Domain Search Option [RFC 3397 "Dynamic Host Configuration Protocol (DHCP) Domain Search Option"]. The encoding of Domain Names used is the on-the-wire format of the DNS, using DNS-defined message compression. DHCP handles Domain Names in other options such as in RFC 4702 defined "The DHCP Client FQDN Option", in the same format. The significance of this is that most other protocols represent DNS domain names or host names in a human readable form, DHCP is using the machine-friendly format.

4. Interoperability Considerations

Any single protocol is able to define a format for a conceptual Domain Name. Examples given above show that many protocol have done so. From the examples it is clear that the way in which protocols have interpreted Domain Names has varied, leading to, at least, user interfaces having to have built-in intelligence when handling names and, at worst, a growing confusion over how the Domain Name space is to be managed.

When protocols having different formats and rules for Domain Names interact, software implementing the protocols translate one protocol’s domain name format to another’s format. Even when the translation is straightforward, software often fails to handle error conditions well. (Is there a citation for that?)
Often times the clash of definitions impacts the design of a new protocol and/or an extension of a protocol. For example, adding non-ASCII domain names has to be done with backwards compatibility with an installed base of ASCII-assuming code. This clash can inhibit new uses of Domain Names.

Search lists are a Domain Name mechanism studied in SAC064 "SSAC Advisory on DNS ‘Search List’ Processing". One of the particular use cases related to this topic is the issuance of search lists via DHCP and then used by any user-client protocol implementation. This emphasizes an interoperability consideration for how Domain Names are treated in different protocols, not just among implementations of one protocol.

The definition of a Fully Qualified Domain Name has two forms. The discussion over FQDN involved human-readable names. The principle question is whether to require the terminating dot or to assume it when the end of an input string is hit. Many protocol clients will silently add a dot when a user types in a name to a command line. But some definitions, such as the one in SAC064 require the terminating dot to be included before a name is considered to be fully qualified.

The Special Use Domain Names registry defined in RFC 6761 lists Domain Names that are to be treated in a manner inconsistent with the DNS normal processing rules. This registry contains Domain Names regardless of whether the name is a DNS domain name and regardless whether the name is a top-level (domain) name [RFC 819] or is positioned elsewhere in the tree structure.

These are reasons this document is needed. The reason for the confusion over what’s a legal domain name stems from application-defined restrictions. For example, using a one-label domain name ("dotless") for sending email is not a problem with the DNS nor the name in concept, but is a problem for mail implementations that expect more than one label. (One-label names may be assumed to be in ARPA host table format.)

4.1(*) Use of Top-Level Domains as Protocol Identifiers

(Would like to have "dns" and "dns<digit>" added to the Special Use Domain Names registry. As well as the digits [address literals].)

(*) - I am not sure this belongs in this document. The idea is what one can select a name space by the top-level domain (label). I believe this would be scope creep for this document. Leaving it here for consideration.

5. Acknowledgements

Input received from Andrew Sullivan and Paul Hoffman. Not to imply endorsements of the text.

Comments from George Michaelson, Kevin Darcy, Joe Abley, Jim Reid, Tony Finch, Robert Edmonds, hellekin, Stephane Bortzmeyer, Ray Bellis, Bob Harold, Alec Muffett. Not to imply endorsement.

6. IANA Considerations

None.

7. Security Considerations
Nothing direct. This document proposes a definition of the term "Domain Name" and surveys how it has been variously applied. In some sense, loosely defined terms give rise to security hazards. Beyond that, there is no impact of "security."

8. References

Many references are in-line throughout the text with titles to ease comprehension of the prose. All documents cited are listed here. Whether there is a normative/informative split will depend what, if any, track this document is processed. For now, consider this a reading list on the topic.


RFC 5280

RFC 5321

RFC 5890

RFC 6761

RFC 6762

Diestel

SAC064

RENDDEV

OHOST
Nick Mathewson, "Special Hostnames in Tor", Undated, <https://gitweb.torproject.org/torspec.git/tree/address-spec.txt>

IEEE1003

WINSOCK

9. Author’s Address

Edward Lewis
ICANN
801 17th Street NW
Suite 401
Washington DC, 20006 US
edward.lewis@icann.org

Appendix A.

This is lifted from an email message expanding on why there’s a problem. The email refers to the proposal to add "alt" to the Special Use Domain Names registry as well as a ficticious proposal to add "carrot". This message can be found here:
Here’s the problem I see.

Let’s say I want to write a very basic SSH client (just to make the story simple). Someone can then type “eds-ssh computer-name” and open up a secured connection.

If computer-name ends in .local, I open TCP to an IP address from the lookup in mDNS.

If computer-name ends in .onion, I open TCP to an IP address I get via Tor (assuming that .onion supports remote shell).

If computer-name ends in a digit, I suppose it’s an address literal and open TCP accordingly.

If computer-name ends in whatever is in the DNS root zone, I find the address in DNS.

If computer-name ends in something not in the DNS root zone, I return an error.

The gotchas include, what if the latter two are indistinguishable because the DNS resolver sent back a landing page — or the latter three if the redirection service didn’t recognize .onion as special.

What if, in a year from now, .carrot becomes yet another way to resolve names?

What if, in the future, .alt is defined as having special meaning? (Note — the fact that .alt is in an actual ID and .carrot is purely fictional means .carrot is closer to being an RFC. ;)) (Joke)

It seems to me that a new layer of software is emerging between the UI and the stub resolver, one that will need to know where to send a name resolution query. (Perhaps even amongst DNS stub resolvers on different interfaces.) This emerging layer needs to know how to direct its work flow. The Special Use Domain Names Registry would be the place to start (but as it’s written now, the emerging layer can’t be future proofed).

These are just TLD examples, perhaps a simplification.

I see a fork in the codepath ahead regarding "whether the DNS is above Domain Names" (like .alt) or whether "Domain Names are broader than what was conveniently defined for a DNS". It’s important to know which of those two statements are true so we can get on with Special Use Domain Names, and perhaps to find ways to objectively assign new names for new uses.

I think defining —whether— name.onion is a Domain Name will make us re-think how Domain Names interoperate amongst protocols beyond the DNS.