

DNSOP
Internet-Draft
Intended status: Standards Track
Expires: October 13, 2014

D. Migault (Ed)
Orange
April 11, 2014

DNS Search List Processing
draft-mglt-dnsop-search-list-processing-00.txt

Abstract

Domain Names can be Qualified or Unqualified Domain Names. Qualified Domain Names are resolved over the public DNS infrastructure, whereas Unqualified Domain Names are resolved using search lists. How search lists are generated and interpreted varies from one application to another and from one operating system to another. This makes Unqualified Domain Name resolution unpredictable, non deterministic, and as such neither reliable nor stable.

In addition, there is neither clear rules to define whether a name is a Qualified or an Unqualified Domain Name. This also contributes in making the naming resolution unreliable, as the resolution of a given name can result in different responses.

As a consequence, most resolution systems currently end with a "try and error" strategy. More specifically, according to some system dependent heuristics, a resolver initiates an Unqualified (resp. Qualified) Domain Name resolution, and, in case of a NXDOMAIN response, fails back in a Qualified (resp. Unqualified) Domain Name resolution. Such strategies were acceptable as the probability of collision between domains within search list and those published in the public DNS infrastructure remains low. In the context of the generalization of Top Level Domain, this assumption is not acceptable anymore, resulting in an unreliable and unstable naming resolution.

This document describes how search list should be generated and interpreted. Then, it describes how resolvers distinguish between Qualified and Unqualified Domain Names as well as how to resolve them.

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

Internet-Draft

DNS Search List Processing

April 2014

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 October 13, 2014.

Copyright Notice

Copyright (c) 2014 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

1.	Requirements notation	3
2.	Introduction	3
2.1.	Qualified and Unqualified Domain Names	3
2.2.	Domain Names Collision	4
2.3.	Structure of the Document	4
3.	Terminology	5
4.	Search List Generation	5
5.	Search List Interpretation	7
6.	Distinction of Unqualified and Qualified Domain Names	7
7.	Resolution of Unqualified and Qualified Domain Names	7
8.	IANA Considerations	8
9.	Security Considerations	8
10.	Acknowledgment	9
11.	References	9
11.1.	Normative References	9
11.2.	Informational References	10

Appendix A.	Document Change Log	11
Appendix B.	TLDs with A/AAAA	11
Author's Address		12

[1.](#) Requirements notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

[2.](#) Introduction

[2.1.](#) Qualified and Unqualified Domain Names

Until recently, the root zone had only a restricted number of well known Top-Level Domain (TLDs). These TLDs had a specific format of a few letters (generally two or three), and had remained almost unchanged for a long time. Simultaneously, end users and applications used for convenience Unqualified Domain Names for local scope resolutions. More specifically, suppose "host1.example" wants to establish a communication with "foo.example". As both host belong to the same Domain "example", simply specifying "foo" should be sufficient. The mechanism to auto-complete "foo" with "foo.example" is performed using search list mechanisms. In most cases, the use of Unqualified Domain Names was used in a local scope context, that is to say, when "example" was used for convenience, but not registered in the public DNS infrastructure.

As a result, there are two ways to express the names of the nodes within a Domain: A Fully Qualified Domain Name ("host2.example") and an Unqualified Domain Name ("host2"). Each of these names requires a different resolving mechanisms. Fully Qualified Domain Names (FQDN) are resolved on the public DNS infrastructure and Unqualified Domain Names are resolved using search lists.

As there are different ways to express a name, a resolver may assume the name is a Qualified (resp. Unqualified) Domain Name when in fact the name is an Unqualified (resp. Qualified) Domain Name. In our example, this would lead to a resolution of "foo." over the public DNS infrastructure. If "foo" is being registered in the root zone,

then "foo.example." and "foo." will most likely not provide the same responses. The confusion between Unqualified and Qualified Domain Names makes naming resolution unstable and unreliable.

To indicate a name is a Fully Qualified Domain Name, one should end it with a dot, however, this has never been accurately be used, and the last dot is most of the time omitted. As a result it has always been confusing to distinguish between Qualified Domain Names and Fully Qualified Domain Names.

[2.2.](#) Domain Names Collision

Even though it has always been confusing, since the number of TLD was very limited, collision would not happen as "foo" differs from existing TLDs. As a result, evaluating "foo" as Fully Qualified Domain Name, would result in resolving "foo." over the public DNS. When the NXDOMAIN response is received, the resolver understands the name is an Unqualified, and use the search list. Overall the impact was quite limited.

Similarly, a company may use multiple Domains for its local scope Domain, and provisions its devices with the search list "example example.com". All devices, and network administrators have always considered that the resolution of "foo.example" is either of local scope or fails. As a result, if "foo.example" cannot be resolved, "foo.example.com" is resolved instead. As "example" was not part of the root zone, network administrators have never considered that "foo.example" could actually been resolved on the public DNS infrastructure and provide a response that is different from the one the private Domain. In the context of gTLDs, "example" is likely to be registered in the root zone, and this by another administrative entity than the company using "example" for its private network.

As the probability of collision was rather small, multiple ways have been implemented to handle the resolution of names including the way to handle search lists -- as with the implicit expansion mechanism. All of these non standard mechanisms provides a variety of ways a resolution is performed which differs from application to application and from operating systems to operating systems. The collision

between private domain names and public domain name makes naming resolution unstable and unreliable.

[2.3.](#) Structure of the Document

With the introduction of generic TLDs (gTLDs), one cannot assume anymore the probability of collision can be ignored. In order to guarantee the stability and reliability of the naming resolution, this document defines in [Section 4](#) how search lists MUST be generated and in [Section 5](#) how search lists MUST be interpreted. [Section 6](#) defines how Qualified and Unqualified Domain Names MUST be distinguished and [Section 7](#) defines how resolution for each category of Domain Name MUST be proceeded.

The expected outcome of such rules are 1) a more reliable and stable naming resolution and 2) a resolution process that is not impacted by the introduction of new gTLDs.

This document is largely based on [[SAC064](#)]

Migault (Ed)

Expires October 13, 2014

[Page 4]

Internet-Draft

DNS Search List Processing

April 2014

[3.](#) Terminology

- Qualified Domain Names or Fully Qualified Domain Name (FQDN) or Absolute Domain Name, is a domain name as defined in [[RFC1035](#)] that specifies its exact location in the DNS tree hierarchy, including the public top-level domain and the root zone. By convention, most operating systems treat domain names that include the terminating "." as an FQDN. For example, `www.corporation.example.com.` specifies an FQDN.
- Unqualified Domain Names is a Domain Name that is not expected to be resolved in the public DNS. In other words, such names is not a FQDN. It is usually an internally used domain name (such as `"www.corporation"`) that only becomes an absolute domain name once expanded as a result of search list processing. The ambiguity of such domains is to define whether it is a FQDN or an Unqualified Multi-label Domain Name. If `"example.com"` is in the search list and if `corporation` becomes a gTLD, `"www.corporation"` can be resolved on the public DNS and `"www.corporation.example.com"` can also be resolved.
- Multi-label Domain Name or Relative Multi-label Domain Name, is a

domain name that consists of more than one label.

- Single Label Domain Name or Dotless Domain Name in some contexts, is a domain name that consists of a single label that is 63 characters or less, starts with a letter, ends with a letter or digit, and has as interior characters only letters, digits, and hyphen as defined by [[RFC1035](#)].
- Generic Top Level Domain (gTLD) is a top level domain. If the list of these top level domain has been quite stable over the years, this list of top level is not any more restricted. As a result, resolving a name cannot be based anymore on the existence or non-existence of the top level domain as it may evolves over time.
- Domain part of a FQDN, is everything after the first dot.

[4.](#) Search List Generation

A search lists is an ordered lists of Domains. When a naming resolution involves a search list for a given name, a resolution is performed for each Domain. Suppose "D1.example.com", "D2.foo", "D3.foo" is a search list and "X" is the name to be resolved. Then, the resolver attempts to resolve successively "X.D1.example.com", "X.D2.foo" and "X.D3.foo" until a response is provided.

To guarantee a reliable and stable way to resolve names, one must also determine a deterministic way to build the search list as well as a deterministic way to handle the search list. To our knowledge the search list may be populated by:

- 1) An explicit manual search list configuration by the end user. Typically this means the user has manually edit the /etc/resolv.conf file on a Linux platform, or the suffix field in various applications.
- 2) The Domain part of the FQDN assigned to the host. More specifically, the FQDN assigned to the host consists in a Domain appended to a hostname. With DHCPv4 [[RFC2131](#)] the Domain is assigned using the DHCP Domain Name Option [[RFC2132](#)]. With DHCPv6 [[RFC3315](#)], the Domain is derived from the DHCPv6 Client FQDN

Option [[RFC4704](#)].

- 3) A search list assigned to the host via the DHCPv4 Domain Search Option [[RFC3397](#)] or the DHCPv6 Domain Search Option [[RFC3646](#)].
- 4) Implicit expansion of the search list which consists in expanding the search prefix "corporation.example.com" into the list "corporation.example.com" "example.com" "com".

In order to make systems end up with the same search list, here are our recommendations:

- 1) If the search list results from a manual configuration, then DHCP Options MUST NOT automatically affect the search list. More specifically, Domain Name derived from DHCPv4 Domain Name Option [[RFC2132](#)] or DHCPv6 Client FQDN Option [[RFC4704](#)] and DHCP Domain Search Option [[RFC3397](#)], [[RFC3646](#)] are ignored for the concerned of search list generation. This follows the recommendations of [[RFC3397](#)] and [[RFC3646](#)].
- 2) If the search list is not manually configured, then DHCP Options MAY be considered. DHCP Domain Search Option [[RFC3397](#)], [[RFC3646](#)] are considered. If considered, the search list is only defined by these options and only these options.
- 3) In the absence of DHCP Domain Search Options, the search list is derived from the Domain that is the DHCPv4 Domain Name Option [[RFC2132](#)] or DHCPv6 Client FQDN Option [[RFC4704](#)]. If so, the search list is only constituted of the Domain name of the host.
- 4) If none of these options are provided, then the search list is empty and resolution are directly performed over the public DNS.

[5.](#) Search List Interpretation

Here is our recommendations to make search list be handled in the same way across systems:

- 1) Implicit expansion of search domain name MUST NOT be performed. In fact implicit expansion exposes the resolver to security flaws as described in [[RFC1535](#)] and [[RFC1536](#)]. As a consequence of not

using implicit expansion of search list, search list MUST be explicitly expressed. Suppose a resolver is expected to resolve a hostname within "paris.corporation.example.com" and then "corporation.example.com". In this case, the associated search list MUST be "paris.corporation.example.com" and MUST NOT be "paris.corporation.example.com". Avoiding implicit expansion addresses the [\[RFC1535\]](#) requirements of indicating the BOUNDARIES of the local scope. Note that indicating explicitly the search list does not significantly increase the size of the DHCP Domain Search Option if the option follows the compression method of domain name encoding in [section 4.1.1 of \[RFC1035\]](#). However, if the Option length exceeds 255 bytes, [\[RFC3396\]](#) describes how to use long options.

6. Distinction of Unqualified and Qualified Domain Names

This section defines how the resolver unequivocally considers a name is an Unqualified Domain Name or a Fully Qualified Domain Name. This distinction leads to different resolution process described in [Section 7](#).

Any name - that is to say a Single-Label Domain Name or a Multi-Label Domain Names - ending with a dot "." is considered as a Fully Qualified Domain as defined in [\[RFC1035\]](#) and [\[RFC1535\]](#).

A Single-Label Domain Name not ending with a dot is considered as an Unqualified Domain Name as recommended by [\[RFC3397\]](#) and [\[RFC3646\]](#).

A Multi-Label Domain Names is considered as a Qualified Domain Name as recommended by [\[RFC3397\]](#) and [\[RFC3646\]](#).

7. Resolution of Unqualified and Qualified Domain Names

This section defines how the resolver MUST proceed for a resolution for Qualified Domain Names and Unqualified Domain Names.

For Qualified Domain Names, the resolver MUST proceed to the resolution over the DNS public infrastructure. If the resolution

fails, returning a NXDOMAIN, no attempt SHOULD be done to resolve it

as an Unqualified Domain Name.

For Unqualified Domain Names, the resolver MUST proceed to the resolution using search list. If the resolution fails, returning a NXDOMAIN, no attempt SHOULD be done to resolve it as an Qualified Domain Name.

Rules defined above to differentiate Unqualified and Qualified Domain Names are similar as in [[RFC3397](#)] and [[RFC3646](#)]. However, the resolution process described in this document differs as we do not permit fall backs to resolution on Qualified or Unqualified Domain Names. In fact, [[RFC3397](#)] and [[RFC3646](#)] defines the resolution as a best guess whether the name is an Unqualified (resp. Qualified) Domain Name. Then, if the resolution fails with an NXDOMAIN, response, the resolver falls back and considers the name as a Qualified (resp. Unqualified) Domain Name.

The main purpose at that time was to limit the number of round trips. Processing resolution this way is not any more acceptable in a gTLD context, as it affects the stability and reliability of the naming resolution. Our primary goal in defining how resolution proceeds is to guarantee resolution remains independent of the newly inserted or removed TLD. More specifically, a name that is considered Unqualified must be resolved using search lists, and if the resolution fails, no fall back to Qualified name should be performed. If fall backs are permitted, then the output of the resolution depends on the content of the root zone. Similarly, if a name is considered qualified, no fall back to unqualified should be done.

These rules do not make possible the resolution of TLD as Single-Label Domain Name. In this case, the TLD to be resolved SHOULD explicitly mention the resolution MUST be performed over the DNS public infrastructure by appending a dot at the end. [Appendix B](#) shows that some TLDs have already associated A/AAAA records.

[8.](#) IANA Considerations

There are no IANA consideration for this document.

[9.](#) Security Considerations

The whole document is about security, more especially naming reliability and stability.

The document defines rules to handle search list so a naming resolution remains stable over time. This is done in different ways. First by defining how search lists are generated, and how search

lists are interpreted by resolver. Then we designates rules to define in a deterministic manner whether a name to be resolved SHOULD be considered as a Qualified Domain Name or as a Unqualified Domain Name. Each kind of Domain name has its associated resolution process, and we do not permit resolution fall backs.

These rules are intended to address the flaws described in [[RFC1535](#)] and [[RFC1536](#)]. The reason for the late fixing is the gTLD program of the ICANN that make now possible to insert new TLDs in the root zone.

As these rules are not currently deployed, most devices will not have clearly defined boundaries between Qualified and Unqualified resolutions. In addition, fall backs resolution between these two categories will happen and MUST be address by administrator before any new gTLD.

DNSSEC [[RFC4033](#)], [[RFC4034](#)] and [[RFC4035](#)] is not designed to distinguish Qualified and Unqualified Domain Names. In fact DNSSEC has been designed to provide a proof of integrity and a proof of ownership. In the case of name collision, if "foo." is in the signed root zone and "foo.example.com" is also signed with DNSSEC, then DNSSEC validates both names. DNSSEC can however help to distinguish between "foo." and "foo.exmaple.com" if the application knows the Key Signing Key (KSK) associated to the expected Domain "example.com". In other words, the KSK will be considered as the Trust Anchor for the requested names.

DANE [[I-D.ietf-dane-ops](#)] uses DNSSEC to provide the cryptographic material, used by the above application or transport layer. If the applications know the certificate or the key used by the layers above, then DANE can be used to distinguish between the expected Names, and the one returned by the resolver.

[10.](#) Acknowledgment

[11.](#) References

[11.1.](#) Normative References

- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, [RFC 1035](#), November 1987.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", [RFC](#)

[2131](#), March 1997.

- [RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", [RFC 2132](#), March 1997.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", [RFC 3315](#), July 2003.
- [RFC3396] Lemon, T. and S. Cheshire, "Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)", [RFC 3396](#), November 2002.
- [RFC3397] Aboba, B. and S. Cheshire, "Dynamic Host Configuration Protocol (DHCP) Domain Search Option", [RFC 3397](#), November 2002.
- [RFC3646] Droms, R., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", [RFC 3646](#), December 2003.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", [RFC 4033](#), March 2005.
- [RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", [RFC 4034](#), March 2005.
- [RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", [RFC 4035](#), March 2005.
- [RFC4704] Volz, B., "The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Client Fully Qualified Domain Name (FQDN) Option", [RFC 4704](#), October 2006.

[11.2.](#) Informational References

[I-D.ietf-dane-ops]

Dukhovni, V. and W. Hardaker, "DANE TLSA implementation

and operational guidance", [draft-ietf-dane-ops-03](#) (work in progress), February 2014.

- [RFC1535] Gavron, E., "A Security Problem and Proposed Correction With Widely Deployed DNS Software", [RFC 1535](#), October 1993.

Migault (Ed)

Expires October 13, 2014

[Page 10]

Internet-Draft

DNS Search List Processing

April 2014

- [RFC1536] Kumar, A., Postel, J., Neuman, C., Danzig, P., and S. Miller, "Common DNS Implementation Errors and Suggested Fixes", [RFC 1536](#), October 1993.
- [SAC064] "SAC064: SSAC Advisory on DNS "Search List" Processing", An Advisory from the ICANN Security and Stability Advisory Committee, URL: <http://www.icann.org/en/groups/ssac/documents/sac-064-en.pdf>, February 2014.

[Appendix A](#). Document Change Log

[[draft-mglt-dnsop-search-list-processing-00.txt](#)]: First version published.

[Appendix B](#). TLDs with A/AAAA

This section provides a small command line that tests which TLD has an A or a AAAA RRset.

```
wget http://data.iana.org/TLD/tlds-alpha-by-domain.txt
for i in `cat tlds-alpha-by-domain.txt`;
do
a=`dig +short -t A $i.`;
aaaa=`dig +short -t AAAA $i.`;
if [ "${a}" != "" ] || [ "${aaaa}" != "" ];
then
echo $i - A:${a}, AAAA:${aaaa};
fi;
sleep 1;
done
```

Figure 1: Command Line to test TLD with A/AAAA

AC - A:193.223.78.210, AAAA:
AI - A:209.59.119.34, AAAA:
CM - A:195.24.205.60, AAAA:
DK - A:193.163.102.24, AAAA:2a01:630:0:40:b1a:b1a:2011:1
GG - A:87.117.196.80, AAAA:
IO - A:193.223.78.212, AAAA:
JE - A:87.117.196.80, AAAA:
PN - A:80.68.93.100, AAAA:
SH - A:193.223.78.211, AAAA:
TK - A:217.119.57.22, AAAA:
TM - A:193.223.78.213, AAAA:
TO - A:216.74.32.107, AAAA:
UZ - A:91.212.89.8, AAAA:
WS - A:64.70.19.33, AAAA:

Figure 2: output

Author's Address

Daniel Migault
Orange
38 rue du General Leclerc
92794 Issy-les-Moulineaux Cedex 9
France

Phone: +33 1 45 29 60 52
Email: daniel.migault@orange.com