SAC064
SSAC Advisory on DNS “Search List” Processing

An Advisory from the ICANN Security and Stability Advisory Committee (SSAC)
13 February 2014

Preface

This is an advisory to the ICANN Board from the Security and Stability Advisory Committee (SSAC) concerning the security and stability implications of DNS search list processing. The SSAC advises the ICANN community and Board on matters relating to the security and integrity of the Internet's naming and address allocation systems. This includes operational matters (e.g., matters pertaining to the correct and reliable operation of the root name system), administrative matters (e.g., matters pertaining to address allocation and Internet number assignment), and registration matters (e.g., matters pertaining to registry and registrar services). SSAC engages in ongoing threat assessment and risk analysis of the Internet naming and address allocation services to assess where the principal threats to stability and security lie, and advises the ICANN community accordingly. The SSAC has no official authority to regulate, enforce, or adjudicate. Those functions belong to others, and the advice offered here should be evaluated on its merits.

A list of the contributors to this advisory, references to SSAC members' biographies and disclosures of interest, and SSAC members' objections to the findings or recommendations in this advisory are at end of this document.

Table of Contents

Executive Summary

A Domain Name System (DNS) "search list" (hereafter, simply "search list") is conceptually implemented as an ordered list of domain names. When the user enters a name, the domain names in the search list are used as suffixes to the user-supplied name, one by one, until a domain name with the desired associated data is found or the search list is exhausted.

Processing search lists was weakly standardized in early Requests For Comments (RFCs) and implemented in most operating systems. However, as the Internet has grown, search list behavior has diversified. Applications (e.g., web browsers and mail clients) and DNS resolvers process search lists differently. In addition, some of these behaviors present security and privacy issues to end systems, can lead to performance problems for the Internet, and might cause collision with names provisioned under the newly delegated top-level domains.

This advisory examines how current operating systems and applications process search lists. It outlines the issues related to the current search list behavior, and proposes both a strawman to improve search list processing in the long term and mitigation options for the Internet Corporation for Assigned Names and Numbers (ICANN) and the Internet community to consider in the short term. The purpose of these proposals is to help introduce new generic Top Level Domains (gTLDs) in a secure and stable manner with minimum disruptions to currently deployed systems. Specifically, the Security and Stability Advisory Committee (SSAC):

- Invites ICANN Supporting Organizations and Advisory Committees, and the DNS operations community to consider the proposed long term behavior for search list processing outlined in this advisory and comment on its correctness, completeness, utility and feasibility;

- Recommends ICANN to work with the DNS community to encourage the standardization of search list processing behavior; and

- Recommends ICANN, in the context of mitigating name collisions, to consider additional steps to address search list processing behavior.

1. Introduction

Many organizations create subdomains under their primary domain(s) to delegate or distribute management of the namespace, reduce the load on the authoritative DNS servers, and more easily distinguish a host's organizational and/or geographical affiliations.

As a convenience to users, many operating systems implement search list processing, a feature that allows a user to enter a partial name in an application, with the operating system expanding the name through entries in a search list. For example, if a user has a search list of "corp.example.com;berlin.example.com;example.com" and she types "system" into her browser's address box, the operating system would try "system.corp.example.com", "system.berlin.example.com", "system.example.com", and perhaps "system." in some order.

Search list processing, including order of operations for search list processing, was loosely specified in RFC 1123 (specifically, section 6.1.4.3 (2)), RFC 1535, and RFC 1536 and has been implemented in most operating systems. However, as the Internet has grown, search list behavior has diversified. Applications (e.g., web browser and mail clients) and DNS resolvers process search list suffixes differently. Some of these behaviors also present security and privacy issues to end systems, and performance problems both for the end system and the Internet.

This advisory examines how current operating systems and applications process search lists. It outlines the issues related to the current search list behavior, and proposes both a straw man to improve search list processing in the long term and mitigation options for ICANN and the Internet community to consider in the short term. The goal is to help introduce new gTLDs in a secure and stable manner, with minimum disruptions to currently deployed systems.

1.1 Terminology

A **Fully Qualified Domain Name (FQDN),** also known as an Absolute Domain Name, is a domain name that specifies its exact location (per RFC 1035) 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.corp.example.com. specifies an FQDN.

An **Unqualified Multi-label Domain Name,** also known as a Relative Multi-label Domain Name, is a domain name that consists of more than one label but does not have an unambiguous meaning in the public DNS. It is usually an internally used domain name (such as www.corp) that only becomes an absolute domain name once expanded as a result of search list processing.

An **Unqualified Single Label Domain Name**, also referred to as dotless domain name[1] 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 RFC 1035 (e.g., internal).

[1] See SAC 053: SSAC Advisory on Dotless Domain Names (23 February 2012) at: http://www.icann.org/en/groups/ssac/documents/sac-053-en.pdf.

A **Partial Domain Name is** an Unqualified Multi-label Domain Name or an Unqualified Single Label Domain Name.

2. Background

Search list is defined in RFC 1123, where it states:

> A search list is conceptually implemented as an ordered list of domain names. When the user enters a name, the domain names in the search list are used as suffixes to the user-supplied name, one by one, until a domain name with the desired associated data is found, or the search list is exhausted.

Search lists are configured locally on a host and are used by the host's resolver library to suffix unqualified names with domains common to the environment. Common entries for a search list include the name of the host's parent domain and related domains.

Historically, when attempting to resolve an unqualified domain name, some DNS resolvers used the domain name of the searching host for deriving the search list and did not distinguish the portion of that name that is in the locally administered scope from the part that is publically administered.[2] This created a security problem that is documented in RFC 1535. To mitigate this issue, the following guideline was proposed to handle search lists:

> At a minimum, DNS resolvers must honor the BOUNDARY between local and public administration, by limiting any search lists to locally-administered portions of the Domain Name space. [3]

The same RFC also proposed a more stringent set of guidelines for resolver software to process search lists:

- Any additional search alternatives should be configured into the resolver explicitly.
- DNS name resolver software should not use implicit search lists in attempts to resolve partial names into absolute FQDNs other than the host's immediate parent domain. Resolvers that continue to use implicit search lists must limit their scope to locally administered sub-domains.

[2] For example, when a user at machine.tech.aces.com makes a query, the implicit search list would be tech.aces.com, aces.com, and com.

[3] See RFC 1535 – "A Security Problem and Proposed Correction With Widely Deployed DNS Software," E. Gavron, 1993 at: http://www.ietf.org/rfc/rfc1535.txt.

- DNS name resolver software should not come pre-configured with explicit search lists.

- Where a "." exists in a specified name, it should be assumed to be a fully qualified domain name (FQDN) and should be tried as a rooted name first.

In addition to RFC 1535, RFC 1536 also proposes some guidelines to handle search lists. However, since both RFC 1535 and RFC 1536 are informational RFCs, not Internet Standards, these guidelines are weakly standardized.

3. Issues with Search List Processing

3.1. Non-Standardization

The SSAC observes that search list processing behaviors by applications and resolver libraries vary in the following areas:

- **Default behavior when search list is not specified:** Some resolver libraries use the domain name of the searching host to implicitly derive the search list, and iteratively remove labels in that name to form a custom search list.[4] Other resolvers only use the domain itself. Still others only honor explicit search lists set by the user or via Dynamic Host Configuration Protocol (DHCP).

- **Domains to apply the search list:** Resolver libraries are consistent in not applying the search list for names ending with ".". However, when resolving domain names without a terminating dot, some resolver libraries apply the search list only to unqualified single label domain names. Other resolvers also apply the search list to unqualified multi-label domain names.

- **Search order:** Some DNS resolvers apply the search list first, before trying the real QNAME (the domain name being queried); others try the real QNAME first, and apply the search list only when the real QNAME does not resolve.

These variations have been publicly documented recently by well-known researchers,[5] and confirmed (and extended) by the SSAC in the following empirical test with five operating systems and sixteen applications with respect to a lookup command on twelve domain names. The results and the detailed explanation of the methodology are included in Appendix B of this report. The general category of responses[6] are summarized below:

---

[4] For example, if host smith.corp.example.com makes a query, the search list order would be corp.example.com, followed by example.com.
[5] See Geoff Huston: https://labs.ripe.net/Members/gih/dotless-names.
[6] Acknowledgement of Geoff Huston for defining these categories in the following blog post: https://labs.ripe.net/Members/gih/dotless-names.

Table 1: Search List Behavior Observed in the SSAC's Empirical Testing

Name	Behavior	Example scenario
never	The search list is not applied, and the original name is queried in the DNS.	Ping command on Windows 7/8 for unqualified multi-label domain names (example.com).
always	The search list is always applied and the synthesized names are queried in the DNS, but the original name is never queried in the DNS.	Debian 7 sendmail (rcpt suffix and relay host) for unqualified single label domain names (example).
pre	The search list is applied to the original name in DNS queries, and if all permutations of the application of the search list generate a NXDOMAIN response then the original name is queried in the DNS.	Windows XP (service pack 2) ping command for unqualified single label (example)
post	The original name is queried in the DNS, and if this generates an NXDOMAIN response then the search list is applied to the original name in DNS queries.	Windows XP (service pack 2) ping command on unqualified multi-label domain names (example.com)
error	The QNAME results in an error, and the name is not queried at all.	Debian 7 postfix (relay host) for dot terminated unqualified multi-label domain names (example.com.)
www	"www" is prepended to the original QNAME.	OS X Safari for a multi-label domain name that does not resolve to an address (example.com). Note: the name exists in DNS, but not with an address record (such as A or AAAA)
search	The string is used as a search term to the default search engine of the browser.	Internet explorer 11 on Windows 7/8 for unqualified single label domain names (example)

In addition to variations in search suffix list handling by operating system (OS) resolver libraries, the SSAC has also observed that applications do not use the resolver library consistently with respect to search suffix lists. Such non-standard behavior might contribute to a degraded user experience.

3.2. Query Leakage

The current search list processing behavior observed in the previous section presents query leakage (that is, unintended and/or unnecessary queries that do not match the user's intent) in the following scenarios:

- The "post" search behavior results in the resolver first issuing a DNS query for the requested name as an FQDN. If no answer is yielded, then it iterates through the search list, appending each suffix in turn, attempting to resolve the newly formed FQDN in the DNS. Where users and applications use unqualified names and resolver libraries use "post" behavior, the result is a number of queries that are expected *not* to resolve prior to proper resolution of a name. In many cases leakage of such queries will result in them reaching the root servers.
- For resolvers in the "always" and "pre" category, when they move to different environments (e.g., from corporate to home network) where different search lists are set via DHCP, queries will be appended with the each search list entry that may not match the user's original intent, causing unintended and unnecessary queries.

According to analyses of data collected by the Domain Name System Operations, Analysis, and Research Center (DNS-OARC) and reported by the Day in The Life of the Internet (DITL) project certain strings repeatedly appear at the root level of the DNS in queries seeking to resolve TLD labels. Figure 1 and 2 depicts the traffic to some proposed TLDs using 2012 and 2013 DITL data.

Figure 1: NXDOMAIN Traffic for some proposed TLDs (source: 2012, 2013 DITL)

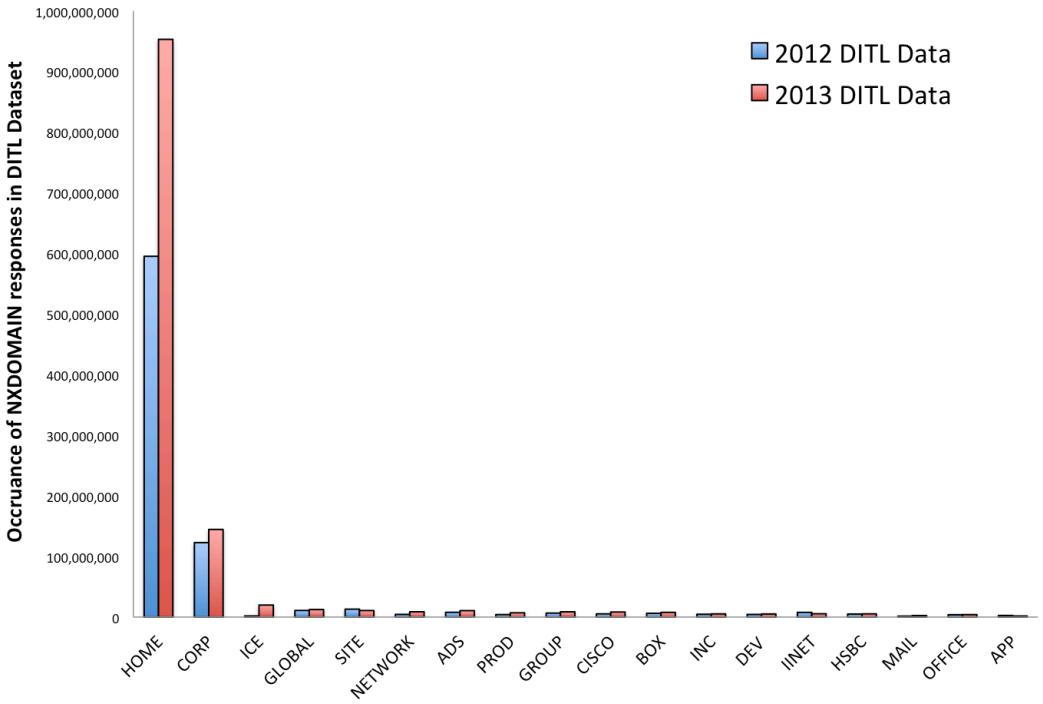
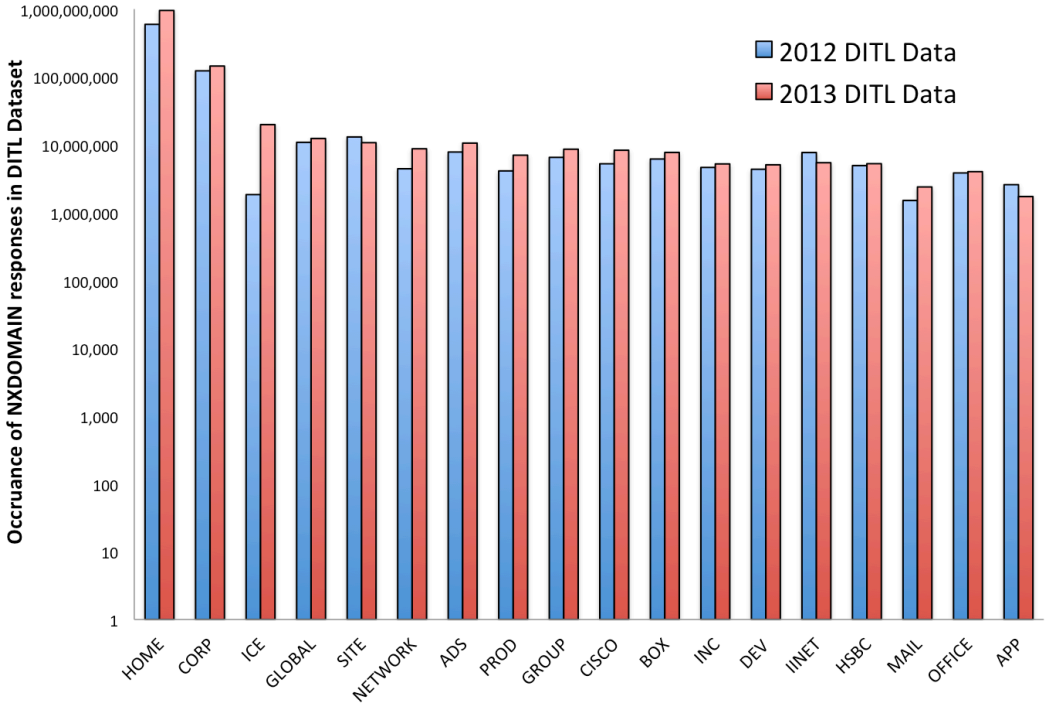


Figure 2: NXDOMAIN traffic for some proposed TLDs (source: 2012, 2013 DITL) Note the log scale of the Y – Axis.



The SSAC believes that search list processing is a contributor to the invalid queries seen at the root servers. Our preliminary analysis shows that Windows XP (OS used by 30 percent of desktop computers as of the fourth quarter of 2013[7]) and Linux (OS used by around 60 percent of the servers[8]) exhibit "post" behavior by default, and queries for unqualified names to resolver libraries on these systems result in potential query leakage. Such leakage not only poses a privacy problem for the end users, it might also result in performance degradation.

3.3. Security Risks From Collisions with Newly Delegated Names

Certain search list behaviors implemented in Windows XP and on Linux systems also present security risks from collision with names provisioned under the newly delegated top-level domains. We use the following example to illustrate.

Fred works for example.com, with offices in multiple countries and an extensive intranet. Their internal system uses Windows XP for desktop/laptop and Linux for server environments. Example.com has created subdomains under their primary domain, one for their main corporate infrastructure and one for each of their 3 remote offices (corp.example.com, paris.example.com, sydney.example.com and chicago.example.com)

Much of their documentation (including a list of holidays, the phone-directory, etc.) is stored on an internal server called www.corp.example.com. Users go to this address for all sorts of things, and have learned that while in the office they can enter www.corp to find information on the server. Similarly, the MTAs (mail transfer agents) are configured to handle non-FQDN names. Thus if they want to send mail to someone in the Chicago office they can email bob@chicago.

Fred is in the Chicago office, and when he connects his laptop to the corporate wireless network his search list is set (using DHCP) to: corp.example.com, chicago.example.com, example.com.

Fred enters www.corp into his browser and presses enter.

Today (prior to the delegation of .CORP), the DNS resolution processes in the stub resolver in Fred's machine will try to resolve this address lookup in the following manner:

[7] Desktop OS Market Share (4Q 2013) by NetMarketShare, available at: http://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustomd=0&qptimeframe=Q

[8] See Usage statistics and market share of Unix for websites. Available at: http://w3techs.com/technologies/details/os-unix/all/all.

```
User > Resolver:  A? www.corp.
Resolver > User:  NXDomain q: A? www.corp.
User > Resolver:  A? www.corp.corp.example.com.
Resolver > User:  NXDomain q: A? www.corp.corp.example.com.
User > Resolver:  A? www.corp.chicago.example.com.
Resolver > User:  NXDomain q: A? www.corp.chicago.example.com.
User > Resolver:  A? www.corp.example.com.
Resolver > User:  A? www.corp.example.com. 192.0.2.10
```

Figure 3: DNS resolution interaction with search list configured.

As shown above, Fred's machine first looks up www.corp (because it contains a '.' it is first treated by his local Windows XP resolver as a fully qualified domain name), and then tries appending each item in the search list until the name finally resolves (www.corp.example.com., 192.0.2.10). All of these queries may be visible at the root server system if an NXDOMAIN is not served from the negative cache[9] on an intermediate resolver.

Before corp is delegated, the lookup process as illustrated in section 2.2 works correctly and Fred reaches his intranet server.

If, however, the .CORP TLD exists (that is, has been delegated), and www.corp is registered with proper A/AAAA records, the first lookup (www.corp) will now succeed and the search list processing will exit. This means that Fred will no longer be reaching his corporate intranet server, and will instead reach a machine in a newly delegated domain name under the corp gTLD.

This change in expected process will both break Fred's connectivity to his internal system and potentially expose him to data loss/interception that does not exist while the .CORP TLD is not delegated. As an example, information leakage might happen if the host that Fred actually connects to presents itself as if it were the host he intended to connect to, and Fred starts sharing information without properly authenticating the party he is talking to. In addition to normal user interactions, many such resolution functions are embedded in applications and software in enterprises and not necessarily initiated by (or even visible to) humans behind a keyboard.

Specifically together with the issues presented in SAC057 about Internal Name Certificates,[10] this result presents a significant attack vector regarding information leakage, direct or in the form of a man-in-the-middle attack. A normal user cannot detect the leakage, even in some cases (i.e. those without serve certificate

[9] That is, the cache of NXDOMAIN responses.
[10] See SAC057: SSAC Advisory on Internal Name Certificates (15 March 2013) at: http://www.icann.org/en/groups/ssac/documents/sac-057-en.pdf.

authenticate) with the help of digital certificates and current cryptographic PKI protections.[11]

However, in order for this to be a significant attack vector, the following conditions must be met:

1) users must be in an environment where unqualified multi-label links have some meaning; and
2) users must be in an environment where Windows XP (and Linux-based OS) does not simply exist but also has a significant number of users. There must be enough users to make it worth advertising multi-label unqualified names, since to any non-Windows XP/Linux users such domain names would simply appear to be unresolvable.

4. A Straw Man to Improve Search List Processing

In this section, the SSAC proposes a modified search list processing algorithm that would mitigate many of the issues identified in this advisory. Some of these rules have been proposed in previous informational RFCs, while other rules proposed here depart from previous RFCs. All of these rules are in agreement with the recent application software behaviors (e.g., Windows 7/8, Mac OS X) with respect to search list processing.

4.1. Proposal

4.1.1. No Automatically Generated Search Lists

Administrators (including DHCP server administrators) should configure the search list explicitly, and must not use implicit search lists (as defined in Section 2).

Where DNS parameters such as the domain search list have been manually configured, these parameters should not be overridden by DHCP.

These are suggested default processing rules. Operating system / resolver vendors may provide configuration options to override these.

4.1.2. Unqualified Single-Label Domain Names Are Never Queried Directly

When a user enters a single label name, that name may be subject to search list processing if a search list is specified, but must never be queried in the DNS in its original single-label form.

[11] This issue could be solved with the use of technologies such as DNS-based Authentication of Named Entities *(DANE),* however they are not yet well supported.

4.1.3. Unqualified Multi-label Domain Names Never Use Search Lists

When a user queries a hostname that contain two or more labels separated by dots, such as www.server, applications and resolvers must query the DNS directly. Search lists must not be applied even if such names do not resolve.

4.2. Negative Consequences For The Change

There are administrators that today rely on both automatic generation of search lists, and the automatic propagation of search lists to clients via DHCP. The proposed change would require reconfiguration of systems.

There are users, and links in web pages, that use partially qualified names (such as www.corp) instead of either just a single token or a fully qualified domain name. Changing search list behavior of unqualified multi-label domain names would reduce the utility of these names.

Finally, not all applications currently in use treat these categories of domain names in the same way. Incompatibilities and operational problems, specifically in BYOD (Bring Your Own Device) environments, already exist.

5. Short Term Mitigation Options for Search Lists

It is desirable in the long run to change search list processing behavior, and the SSAC proposes one approach in section 4. However, in the short run it is likely that existing search list behavior will interfere with the introduction of new gTLDs. Thus, the SSAC advises ICANN, as the global coordinator for the DNS, to consider additional measures to mitigate the impact of search lists in the overall context of name collision.

These measures would include:

- Commission additional research studies to further understand the cause of invalid queries to the root zone and the significance of search list processing as a contributor to those queries.
- Communicate to system administrators that search list behaviors currently implemented in some operating systems will cause collision with names provisioned under the newly delegated top-level domains. Such communication should complement the current ICANN effort in this area[12] with findings and recommendations from this report.[13]

[12] See, "Guide to Name Collision Identification and Mitigation for IT Professionals." Section 5 addressed issues related to search list processing. https://www.icann.org/en/about/staff/security/ssr/name-collision-mitigation-05dec13-en.pdf.

[13] This SSAC document proposes greater clarity for algorithms that specify how to handle different strings received by resolver libraries. This work complements the existing ICANN document.

6. Findings

Finding 1: There is variance on how search lists are processed. The variation is large between applications and operating systems.

Finding 2: RFC 1535 is ambiguous in how search list processing should take place. How to process a (specifically unqualified) domain name can be interpreted in multiple ways.

Finding 3: Deployed operating systems and applications violate RFC 1535 today. Application developers and providers of operating systems today already have started to implement search list algorithms that differ from RFC 1535. This leads to incompatibilities and might contribute to a degraded user experience.

Finding 4: Some search list algorithms deployed today will create problems when new TLDs are delegated. Search list processing according to RFC 1535 can today result in local resolution of a name if a TLD is not delegated. However after that TLD is delegated, global resolution will occur (see SAC062[14]).

7. Recommendations

Given the variance in implementation of search lists, the use of shortened domain names is non-deterministic and, as a result, can result in negative consequences regardless of whether TLDs are delegated. Thus, the SSAC makes the following recommendations:

Recommendation 1: The SSAC invites all ICANN Supporting Organizations and Advisory Committees, the Internet Engineering Task Force (IETF), and the DNS operations community to consider the following proposed behavior for search list processing and comment on its correctness, completeness, utility and feasibility.

a. Administrators (including DHCP server administrators) should configure the search list explicitly, and must not rely on or use implicit search lists; Where DNS parameters such as the domain search list have been manually configured, these parameters should not be overridden by DHCP.

b. When a user enters a single label name, that name may be subject to search list processing if a search list is specified, but must never be queried in the DNS in its original single-label form.

c. When a user queries a hostname that contain two or more labels separated by dots, such as www.server, applications and resolvers must query the DNS

[14] See SAC062: SSAC Advisory Concerning the Mitigation of Name Collision Risk (07 November 2013) at: http://www.icann.org/en/groups/ssac/documents/sac-062-en.pdf.

directly. Search lists must not be applied even if such names do not resolve to an address (A/AAAA). Therefore www.server is always a FQDN.

Recommendation 2: The SSAC recommends ICANN staff to work with the DNS community and the IETF to encourage the standardization of search list processing behavior.

Such an effort should begin with ICANN staff submitting an Internet-Draft to the IETF, and advocating for its standardization within the IETF process. The effort should update RFC 1535 and other applicable RFCs to address the Findings and Recommendations in this document.

Recommendation 3: In the context of mitigating name collisions, ICANN should consider the following steps to address search list processing behavior.

a. Commission additional research studies to further understand the cause of invalid queries to the root zone and the significance of search list processing as a contributor to those queries.

b. Communicate to system administrators that search list behaviors currently implemented in some operating systems will cause collision with names provisioned under the newly delegated top-level domains. Such communication should complement the current ICANN effort in this area with findings and recommendations from this report.[15]

8. Acknowledgments, Disclosures of Interests, and Objections and Withdrawals

In the interest of greater transparency, these sections provide information on three aspects of our process. The Acknowledgments section lists the SSAC members and other individuals who contributed to this particular document. The Disclosures of Interest section points to the biographies of the Committee members and any conflicts of interest, real, apparent or potential, that may bear on the material in this document. The Objections and Withdrawals section provides a place for individual members to disagree with the content of this document or the process for preparing it.

8.1 Acknowledgments

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

SSAC members
Jaap Akkerhuis
Don Blumenthal
James Galvin

[15] See footnote 12 and 13 concerning efforts that have occurred.

Patrik Fältström
Warren Kumari
Danny McPherson
Paul Vixie
Suzanne Woolf

ICANN staff
Francisco Arias
Casey Deccio (ICANN research fellow)
Steve Sheng (editor)

During the production of this report, the SSAC consulted a broader technical community. For their time and contributions, the SSAC wants to specifically thank the following person(s):

Geoff Huston (APNIC)

8.2 Disclosures of Interest

SSAC member biographical information and Disclosures of Interest are available at: http://www.icann.org/en/groups/ssac/biographies-13feb14-en.htm.

8.3 Objections and Withdrawals

There were no objections or withdrawals.

Appendix A: Search List in the RFCs – Research Note

In this research note, the SSAC compiled relevant past Requests for Comments (RFCs) that provide guidance to search list behavior. For each RFC, it summarized the relevant sections on search list, as well as key points that would be of interest to readers.

RFC 1034: Domain names - concepts and facilities P.V. Mockapetris [November 1987] (TXT = 129180) (Obsoletes RFC0973, RFC0882, RFC0883) (Updated-By RFC1101, RFC1183, RFC1348, RFC1876, RFC1982, RFC2065, RFC2181, RFC2308, RFC2535, RFC4033, RFC4034, RFC4035, RFC4343, RFC4035, RFC4592, RFC5936) (Also STD0013) (Status: INTERNET STANDARD) (Stream: Legacy)

> 3.1. Name space specifications and terminology
>
> When a user needs to type a domain name, the length of each label is omitted and the labels are separated by dots ("."). Since a complete domain name ends with the root label, this leads to a printed form which ends in a dot. We use this property to distinguish between:
>
> - a character string which represents a complete domain name (often called "absolute"). For example, "poneria.ISI.EDU."
>
> - a character string that represents the starting labels of a domain name which is incomplete, and should be completed by local software using knowledge of the local domain (often called "relative"). For example, "poneria" used in the ISI.EDU domain.
>
> Relative names are either taken relative to a well known origin, or to a list of domains used as a search list. Relative names appear mostly at the user interface, where their interpretation varies from implementation to implementation, and in master files, where they are relative to a single origin domain name. The most common interpretation uses the root "." as either the single origin or as one of the members of the search list, so a multi-label relative name is often one where the trailing dot has been omitted to save typing.
>
> 4.3.4. Negative response caching (Optional)
>
> The DNS provides an optional service which allows name servers to distribute, and resolvers to cache, negative results with TTLs. For example, a name server can distribute a TTL along with a name error indication, and a resolver receiving such information is allowed to assume that the name does

not exist during the TTL period without consulting authoritative data. Similarly, a resolver can make a query with a QTYPE which matches multiple types, and cache the fact that some of the types are not present.

This feature can be particularly important in a system which implements naming shorthands that use search lists beacuse a popular shorthand, which happens to require a suffix toward the end of the search list, will generate multiple name errors whenever it is used.

RFC 1123: Requirements for Internet Hosts - Application and Support R. Braden [October 1989] (TXT = 245503) (Updates RFC0822, RFC0952) (Updated-By RFC1349, RFC2181, RFC5321, RFC5966) (Also STD0003) (Status: INTERNET STANDARD) (Stream: Legacy)

6.1.4.3 Interface Abbreviation Facilities
...

Search Lists

A search list is conceptually implemented as an ordered list of domain names. When the user enters a name, the domain names in the search list are used as suffixes to the user-supplied name, one by one, until a domain name with the desired associated data is found, or the search list is exhausted. Search lists often contain the name of the local host's parent domain or other ancestor domains. Search lists are often per-user or per-process.

It SHOULD be possible for an administrator to disable a DNS search-list facility. Administrative denial may be warranted in some cases, to prevent abuse of the DNS.

There is danger that a search-list mechanism will generate excessive queries to the root servers while testing whether user input is a complete domain name, lacking a final period to mark it as complete. A search-list mechanism MUST have one of, and SHOULD have both of, the following two provisions to prevent this:

(a) The local resolver/name server can implement caching of negative responses (see Section 6.1.3.3).

(b) The search list expander can require two or more interior dots in a generated domain name before it tries using the name in a query to non-local domain servers, such as the root.

DISCUSSION:

The intent of this requirement is to avoid excessive delay for the user as the search list is tested, and more importantly to prevent excessive traffic to the root and other high-level servers.

For example, if the user supplied a name "X" and the search list contained the root as a component, a query would have to consult a root server before the next search list alternative could be tried. The resulting load seen by the root servers and gateways near the root would be multiplied by the number of hosts in the Internet.

The negative caching alternative limits the effect to the first time a name is used. The interior dot rule is simpler to implement but can prevent easy use of some top-level names.

RFC 1535: A Security Problem and Proposed Correction With Widely Deployed DNS Software E. Gavron [October 1993] (TXT = 9722) (Status: INFORMATIONAL) (Stream: Legacy)

Abstract: This document discusses a flaw in some of the currently distributed name resolver clients. The flaw exposes a security weakness related to the search heuristic invoked by these same resolvers when users provide a partial domain name, and which is easy to exploit (although not by the masses). This document points out the flaw, a case in point, and a solution.

Research Note: This entire RFC is relevant.

RFC 1536: Common DNS Implementation Errors and Suggested Fixes A. Kumar, J. Postel, C. Neuman, P. Danzig, S. Miller [October 1993] (TXT = 25476) (Status: INFORMATIONAL) (Stream: IETF, Area: int, WG: dns)

6. Name Error Bugs:

This bug is very similar to the Zero Answer bug. A server returns an authoritative NXDOMAIN when the queried name is known to be bad, by the server authoritative for the domain, in the absence of negative caching. This authoritative NXDOMAIN response is usually accompanied by the SOA record for the domain, in the authority section.

Resolvers should recognize that the name they queried for was a bad name and should stop querying further.

Some resolvers might, however, not interpret this correctly and continue to query servers, expecting an answer record.

Some applications, in fact, prompt NXDOMAIN answers! When given a perfectly good name to resolve, they append the local domain to it e.g., an application in the domain "foo.bar.com", when trying to resolve the name "usc.edu" first tries "usc.edu.foo.bar.com", then "usc.edu.bar.com" and finally the good name "usc.edu". This causes at least two queries that return NXDOMAIN, for every good query. The problem is aggravated since the negative answers from the previous queries are not cached. When the same name is sought again, the process repeats.

Some DNS resolver implementations suffer from this problem, too. They append successive sub-parts of the local domain using an implicit searchlist mechanism, when certain conditions are satisfied and try the original name, only when this first set of iterations fails. This behavior recently caused pandemonium in the Internet when the domain "edu.com" was registered and a wildcard "CNAME" record placed at the top level. All machines from "com" domains trying to connect to hosts in the "edu" domain ended up with connections to the local machine in the "edu.com" domain!

GOOD/BAD IMPLEMENTATIONS:

Some local versions of BIND already implement negative caching. They typically cache negative answers with a very small TTL, sufficient to answer a burst of queries spaced close together, as is typically seen.

The next official public release of BIND (4.9.2) will have negative caching as an ifdef'd feature.

The BIND resolver appends local domain to the given name, when one of two conditions is met:

i. The name has no periods and the flag RES_DEFNAME is set.
ii. There is no trailing period and the flag RES_DNSRCH is set.

The flags RES_DEFNAME and RES_DNSRCH are default resolver options, in BIND, but can be changed at compile time.

Only if the name, so generated, returns an NXDOMAIN is the original name tried as a Fully Qualified Domain Name. And only if it contains at least one period.

FIXES:

a. Fix the resolver code.

b. Negative Caching. Negative caching servers will restrict the traffic seen on the wide-area network, even if not curb it altogether.

c. Applications and resolvers should not append the local domain to names they seek to resolve, as far as possible. Names interspersed with periods should be treated as Fully Qualified Domain Names.

In other words, Use searchlists only when explicitly specified. No implicit searchlists should be used. A name that contains any dots should first be tried as a FQDN and if that fails, with the local domain name (or searchlist if specified) appended. A name containing no dots can be appended with the searchlist right away, but once again, no implicit searchlists should be used.

RFC 3397: Dynamic Host Configuration Protocol (DHCP) Domain Search Option B. Aboba, S. Cheshire [November 2002] (TXT = 15446) (Status: PROPOSED STANDARD) (Stream: IETF, WG: NON WORKING GROUP)

Abstract: This document defines a new Dynamic Host Configuration Protocol (DHCP) option which is passed from the DHCP Server to the DHCP Client to specify the domain search list used when resolving hostnames using DNS.

Security Considerations

The degree to which a host is vulnerable to attack via an invalid domain search option is determined in part by DNS resolver behavior. [RFC1535] discusses security weaknesses related to implicit as well as explicit domain searchlists, and provides recommendations relating to resolver searchlist processing. [RFC1536] section 6 also addresses this vulnerability, and recommends that resolvers:

[1] Use searchlists only when explicitly specified; no implicit searchlists should be used.

[2] Resolve a name that contains any dots by first trying it as an FQDN and if that fails, with the local domain name (or searchlist if specified) appended.

[3] Resolve a name containing no dots by appending with the searchlist right away, but once again, no implicit searchlists should be used.

In order to minimize potential vulnerabilities it is recommended that:

[a] Hosts implementing the domain search option SHOULD also implement the searchlist recommendations of [RFC1536], section 6.

[b] Where DNS parameters such as the domain searchlist or DNS servers have been manually configured, these parameters SHOULD NOT be overridden by DHCP.

[c] Domain search option implementations MAY require DHCP authentication [RFC3118] prior to accepting a domain search option.

RFC 3646: DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6) R. Droms [December 2003] (TXT = 13312) (Status: PROPOSED STANDARD) (Stream: IETF, Area: int, WG: dhc)

Abstract: This document describes Dynamic Host Configuration Protocol for IPv6 (DHCPv6) options for passing a list of available DNS recursive name servers and a domain search list to a client.

Appendix B: Testing Methodology and Result for Search List Processing

Strings tested: The strings tested exhibit the following characteristics:

1. Whether the string contains a single label or multiple labels. (1 or 2)
2. Whether the string is dot terminated (explicit FQDN) (Y or N)
3. Whether the SLD tested exists (Y, N)
4. Whether the TLD tested exists (Y, N)
5. Whether the name resolves to address (Y, N)

Test Case	Examples	
1,N,N,N,N	foo-test	(a.k.a unqualified single label)
1,Y,N,N,N	foo-test.	
1,N,Y,Y,Y	dk	
1,Y,Y,Y,Y	dk.	
2,N,N,N,N	foo-test.baz	(a.k.a unqualified multi-label)
2,Y,N,N,N	foo-test.baz.	
2,N,Y,N,N	foo-test.com	
2,Y,Y,N,N	foo-test.com.	
2,N,Y,Y,N	nasa.gov	
2,Y,Y,Y,N	nasa.gov.	
2,N,Y,Y,Y	test.com	
2,Y,Y,Y,Y	test.com.	

Search list configuration: icann.org, lax.icann.org, with domain set to icann.org.
Operating Systems tested: Windows XP, Windows 7, Windows 8, Mac OS X 10.9, Debian 7
Applications tested:

- **Command tools:** nslookup (special resolver library), ping (uses standard OS resolvers library);
- **Browsers:** Internet Explorer 6, Internet Explorer 8, Internet Explorer 11, Safari, Firefox, Chrome, Iceweasel (Firefox);
- **Mail clients and servers:** Thunderbird (outgoing mail server setting), outlook (outgoing mail server setting), Apple Mail (outgoing mail server setting), postfix (both as relay host and as mail recipient suffix), exim4 (both as relay host and as mail recipient suffix), sendmail (both as relay host and as mail recipient suffix), qmail (both as relay host and as mail recipient suffix).

Capture tools: Wireshark and/or tcpdump.

Result:
Note, to avoid caching, every string tested are different. In general, there are seven types of responses, the fist four are general behaviors, named by the order the search

list is applied in relation to the original query name[16], and the last two are special behaviors of certain applications.

Name	Description	Example
never	the search list is not applied, and the original name is queried in the DNS	*User > Resolver: A? www.corp.* *Resolver > User: NXDomain q: A? www.corp.*
always	the search list is always applied and the synthesized names are queried in the DNS, but the original name is never queried in the DNS	*User > corp.* *User > Resolver: A? corp.corp.example.com.* *Resolver > User: NXDomain q: A? corp.corp.example.com.* *User > Resolver: A? corp.chicago.example.com.* *Resolver > User: NXDomain q: A? corp.chicago.example.com.* *User > Resolver: A? corp.example.com.* *Resolver > User: NXDomain A? corp.example.com.*
pre	the search list is applied to the original name in DNS queries, and if all permutations of the application of the search list generate a NXDOMAIN response then the original name is queried in the DNS.	*User > corp.* *User > Resolver: A? corp.corp.example.com.* *Resolver > User: NXDomain q: A? corp.corp.example.com.* *User > Resolver: A? corp.chicago.example.com.* *Resolver > User: NXDomain q: A? corp.chicago.example.com.* *User > Resolver: A? corp.example.com.* *Resolver > User: NXDomain A? corp.example.com.* *User > Resolver: A? corp.* *Resolver > User: NXDomain q: A? corp.*
post	the original name is queried in the DNS, and if this generates an NXDOMAIN response then the search list is applied to the original name in DNS queries.	*User > Resolver: A? www.corp.* *Resolver > User: NXDomain q: A? www.corp.* *User > Resolver: A? www.corp.corp.example.com.* *Resolver > User: NXDomain q: A? www.corp.corp.example.com.* *User > Resolver: A? www.corp.chicago.example.com.* *Resolver > User: NXDomain q: A? www.corp.chicago.example.com.* *User > Resolver: A? www.corp.example.com.* *Resolver > User: A? www.corp.example.com. 192.0.2.10*
error	the QNAME results in an error, and the name is not queried at	

[16] Credit to Geoff Houston, used with permission. https://labs.ripe.net/Members/gih/dotless-names.

	all.	
WWW	"www" is prepended to the original QNAME.	*User: A? corp.* User > Resolver: www.corp ...
search	the string is used as a search term to the default search engine of the browser.	*User: A? corp.* Application: https://www.google.com/search?q=corp

Appendix C: How to Configure Search lists Behavior in Operating Systems

Windows:

Windows XP:
When a Windows XP machine attempts to resolve an unqualified multi-label name, the DNS client will attempt to resolve the name as specified, then will append the domains that are listed in the DNS suffix search order.

Windows Vista (7 and 8).
When a Windows Vista machine attempts to resolve an unqualified multi-label name, the DNS client will attempt to resolve the name as specified. The DNS suffix search order will NOT be used.

The following registry entry works for both Windows XP and Windows Vista
HKLM\Software\Policies\Microsoft\WindowsNT\DNSClient\AppendToMultiLabelName
Type = DWORD

Data:
0 (Do not Append Suffix)
1 (Append suffix)

If the registry entry is not present, the default in Windows XP is 1, and 0 in Windows Vista.

This registry changes and its effects apply only to the ping command, they do not apply to the Nslookup tool. This is because Nslookup contains its own DNS resolver and does not rely on the resolver built into the operating system (DNS Client). The DNS (multi-label) query packets sent by the nslookup tool will append the domains listed in the suffix search order irrespective of the registry key settings mentioned here.

Microsoft TechNet. (2009) DNS Client Name Resolution behavior in Windows Vista vs. Windows XP. Available at:
http://blogs.technet.com/b/networking/archive/2009/04/16/dns-client-name-resolution-behavior-in-windows-vista-vs-windows-xp.aspx
Microsoft (2012). DNS Processes and Interations. Available at:
http://technet.microsoft.com/en-us/library/dd197552(v=ws.10).aspx

Mac OS X

In Mac OS X, you can use search domains (configured in the Network pane of System Preferences) to help you auto-complete long host names in Safari and other applications. In OS X Lion, name lookups using search domains are completed differently than in previous versions of Mac OS X.