Domain Name System Operations
Internet-Draft
Updates: RFC8806 (if approved)
Intended status: Standards Track
Expires: 3 December 2026

W. Kumari
Google, Inc.
W. Hardaker
USC/ISI and Google, Inc.
J. Reid
RTFM llp
G. Huston
APNIC

1 June 2026

Populating resolvers with the root zone
draft-wkumari-dnsop-localroot-bcp-05

Abstract

DNS recursive resolver operators need to provide the best service possible for their users, which includes providing an operationally robust and privacy protecting service. Challenges to these deployment goals include difficulty of getting responses from the root servers (such as during a network attack), longer-than-desired round-trip times to the closest DNS root server, and privacy issues relating to queries sent to the DNS root servers. Resolvers can solve all of these issues by simply serving an already cached a copy of the full root zone.

This document shows how resolvers can fetch, cache and maintain a copy of the root zone, how to detect if the contents becomes stale, and procedures for handling error conditions.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://wkumari.github.io/draft-wkumari-dnsop-localroot-bcp/draft-wkumari-dnsop-localroot-bcp.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-wkumari-dnsop-localroot-bcp/.

Discussion of this document takes place on the Domain Name System Operations Working Group mailing list (mailto:dnsop@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/dnsop/. Subscribe at https://www.ietf.org/mailman/listinfo/dnsop/.

Source for this draft and an issue tracker can be found at https://github.com/https://github.com/wkumari/draft-wkumari-dnsop-localroot-bcp.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

DNS recursive resolvers have to provide responses to all queries from their clients, even those for domain names that do not exist. For each queried name that is within a top-level domain (TLD) that is not in the recursive resolver's cache, the resolver must send a query to a DNS root server to get the information for that TLD or to find out that the TLD does not exist. Many of the queries to root servers get answers that are referrals to other servers. But, research shows that the vast majority of queries going to the root are for names that do not exist in the DNS root zone [DNEROOTNAMES]. Regardless of whether the queries get positive or negative answers, there are privacy implications related to the eavesdropping of these queries as they are being transmitted to the DNS root servers.

1.1. Local Caching of Root Server Data

Caching the IANA root zone data locally, commonly referred to as running a "LocalRoot" instance, provides a method for the operator of a recursive resolver to use a complete copy of the IANA root zone locally instead of sending requests to the Root Server System (RSS). This goal can be implemented using a number of different techniques, including as described in this document. However, the net effect will be the same: few, if any, queries should be sent to the actual RSS.

Implementation techniques are documented herein for achieving LocalRoot functionality (see Section 3). At a high level, this involves a LocalRoot implementation pre-fetching the root zone at regular intervals and populating its resolver's cache with information, or by running an authoritative server in parallel that acts as a local, authoritative root server for its associated resolver. Other mechanisms for implementing LocalRoot functionality MAY be used. To a client, the net effect of using any technique SHOULD be nearly indistinguishable to that of a non-Localroot resolver.

Note that enabling LocalRoot functionality in a resolver should have little effect on improving resolver speed to its stub resolver clients for queries under Top Level Domains (TLDs), as the TTL for most TLDs is long-lived (two days in the current root zone). Thus, most TLD nameserver and address records are typically already in a resolver's cache. Negative answers from the root servers are also cached in a similar fashion, though potentially for a shorter time based on the SOA negative cache timing (one day in the current root zone).

Also note that a different approach to partially mitigating some of the privacy problems that a LocalRoot enabled resolver solves can be achieved using the "Aggressive Use of DNSSEC-Validated Cache" [RFC8198] functionality.

This document obsoletes [RFC8806].

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.1. Terminology used in this document

Readers are expected to be familiar with the terminology defined in [RFC8499]. In addition, the following terminology will be used in this document:

* IANA root zone: the Internet's globally unique DNS root zone as published by IANA [RFC2826]. This is the same source of root zone data used by the Root Server Operators [RSSAC055]. [RFC8499] describes the same root zone as "The zone of a DNS-based tree whose apex is the zero- length label. Also sometimes called ''the DNS root'."

* IANA root zone data: the complete set of records that makes up the IANA root zone.

* A LocalRoot enabled resolver: a recursive resolver that makes use of a local copy of the root zone data while performing its DNS resolution process.

* A LocalRoot implementation: the software or system of software responsible for implementing the functionality described in this specification. A LocalRoot implementation may be implemented as a singular component within a recursive resolver or within multiple components operating in coordination. Implementations may also vary significantly in how these tasks are performed, ranging from static configuration to more active systems. We refer to this entire system, regardless of implementation sytle, as a "LocalRoot implementation".

3. Components of a LocalRoot enabled resolver

To implement the goals described in Section 1.1 and meet the requirements described in Section 4, a LocalRoot enabled resolver will need to perform three fundamental tasks:

1. Identify locations from where root zone data can be obtained (Section 3.1).

2. Downloading and refreshing the root zone data from one of the publication points (Section 3.2).

3. Integrating and serving the data while performing DNS resolutions (Section 3.3).

Implementing these tasks entirely alleviates the need for sending any (other) DNS requests to the RSS.

Each of these tasks are described in greater detail in the subsections below.

3.1. Identifying locations from where root zone data can be obtained

For a LocalRoot enabled resolver to serve up to date data, an implementation must be able to fetch the contents of the entire IANA root zone on a regular basis from at least one publication source. Implementations can find sources of root zone data in a number of ways, including but not limited to:

1. An operationally configured list of sources (for example a file of URLs) that can be used to fetch a copy of the IANA root zone.

2. A list of sources distributed with the resolver software itself, (akin to how the root hints file is distributed with many resolvers today).

3. Downloading a list of available sources from IANA. The mechanism and list format for doing so is described in [draft-hardaker-dnsop-iana-root-zone-publication-points], which asks IANA to aggregate, publish and maintain a list of IANA DNS root zone sources at _TBD-URL_ Guidance to IANA (or for other entities wishing to collect and redistribute a list of sources) for how to collect and maintain a list of IANA root data publication sources is also discussed separately in [draft-hardaker-dnsop-root-zone-pub-list-guidelines].

3.2. Downloading and refreshing root zone data

Once a list of available publication points of IANA root zone data have been configured or obtained, a LocalRoot implementation MAY be use the following steps to obtain and maintain an up to date copy of the IANA root zone data. Note that as long as the desired effect of performing normal DNS resolution remains stable when combined with LocalRoot functionality, other implementation strategies MAY be used.

If a local copy of the IANA root zone data is unavailable for use in DNS resolution at any point in these steps, resolvers SHOULD fall back to performing DNS resolution by issuing queries directly to the RSS instead. If a resolver is unable to do so, it MUST respond to client requests with a SERVFAIL response code.

1. A LocalRoot implementation SHOULD use a list of root zone sources identified in Section 3.1 for obtaining a copy of the IANA root zone.

2. A LocalRoot implementation SHOULD select one of the available sources from step 1, and from it retrieve a current copy of the IANA root zone. Resolvers SHOULD prioritize sources that can be fetched the most efficiently. For example, when supported, https sources should be preferred as it allows for compression negotiation as well as the use of low-cost, well-distributed Content Delivery Networks (CDNs).

   When sending requests to a source of IANA root zone data, the resolver SHOULD minimize its impact on the source by querying at a rate no faster than specified by the SOA refresh timer and SHOULD use data freshness protocol checks instead of downloading the entire contents at each refresh (example checks include the HEAD method [RFC9110] when using HTTP(s) or by querying the root zone’s SOA over DNS first when using AXFR, IXFR or XoT). Once

fetched, an implementation MUST NOT make use of the obtained IANA root zone data with a SOA serial number older than any previously obtained copy [RFC1982].

3. If the LocalRoot implementation failed to retrieve the IANA root zone data in step 2, or the SOA serial number was deemed to be older than the already cached data, then it SHOULD attempt to retrieve the IANA root zone data from another source. If the LocalRoot implementation resolver has exhausted the list of sources, it SHOULD stop attempting to download the IANA root zone data and SHOULD wait another refresh time length until retrying sources again.

4. Having successfully downloaded a copy of the IANA root zone, the LocalRoot implementation MUST verify the contents of the IANA root zone data using the ZONEMD [RFC8976] record contained within it. Note that this REQUIRES verification of the ZONEMD record using DNSSEC [BCP237] with the configured IANA root zone trust anchor. The contents of the fetched zone MUST NOT be used until after ZONEMD verification, including its DNSSEC verification, is complete and successful. Once the IANA root zone data has been verified, the LocalRoot implementation can begin LocalRoot enabled DNS resolution, potentially using the steps defined in Section 3.3.

5. The resolver MUST check at least one the sources in step 1 at a regular interval to identify when a new copy of the IANA root zone data is available. This frequency MAY be configurable and SHOULD default to the IANA root zone's current SOA refresh value. When a resolver has detected that a new copy of the IANA root zone data is available, the resolver SHOULD start at step 2 to obtain a new copy of the IANA root zone data. Resolvers MAY check multiple sources to ensure one source has not fallen significantly behind in its copy of the IANA root zone.

3.3. Integrating and serving root zone data during resolution

Any mechanism a LocalRoot implementation uses to integrate the IANA root zone data obtained in Section 3.2 to perform DNS resolution tasks is sufficient if it is virtually indistinguishable to the DNS resolver's clients. Two example implementation strategies are included below.

3.3.1. Pre-caching the root zone data

Once the IANA root zone data has been collected and verified as complete and correct (Section 3.2), a resolver MAY simply update its cache with the newly obtained records. Note that it is RECOMMENDED that such implementations also perform aggressive DNSSEC caching [RFC8198], otherwise significant traffic will still be sent to the Root Server System.

3.3.2. Running a local authoratative copy of the root zone in parallel

[RFC8806] described an implementation mechanism where a copy of the IANA root zone could be run in an authoratative server running in parallel to the recursive resolver. The recursive resolver could then be configured to simply point at this parallel server for obtaining data related to the root zone instead of the RSS itself.

Note that [RFC8806] required that the parallel server be running on a loopback address, but this specification removes that requirement. Instead, implementations MAY run the parallel service on any service address it can legitimately use. However, such a server MUST NOT use an address of one of the official root server addresses in the root zone.

4. LocalRoot enabled resolver requirements

The following requirements are to be followed when creating and/or deploying a LocalRoot implementation:

* A LocalRoot implementation MUST have a configured DNSSEC trust anchor such as an up-to-date copy of the public part of the Key Signing Key (KSK) [RFC4033] or used to sign the DNS root or its DS record.

* A LocalRoot implementation MUST retrieve or be provisioned with a copy of the entire current root zone (including all DNSSEC-related records) (see Section 3.2).

* A LocalRoot implementation MUST validate the contents of the root zone using ZONEMD [RFC8976], and MUST check the validity of the ZONEMD record using DNSSEC.

* A LocalRoot implementation MUST use and serve records from the root zone without modification.

* A LocalRoot enabled resolver SHALL return identical answers about the DNS root, or any other part of the DNS, as if it would if it were not operating as a LocalRoot enabled resolver.

* A LocalRoot implementation SHOULD be able to fall back to querying the authoritative RSS servers whenever the local copy of the root zone data is unavailable or has been deemed stale (see Section 3.2).

* A LocalRoot implementation MUST have an upper time limit beyond which if a new copy of the IANA root zone data is not available it will revert to sending regular DNS queries to the RSS for performing DNS resolutions on behalf of its clients. This upper limit value MAY be configurable and SHOULD default to the root zone's current SOA expiry value. It MUST NOT be longer than the root zone's current SOA expiry value. Once the LocalRoot implementation's copy of the IANA root zone has been successfully refreshed and is no longer considered expired, the resolver may resume LocalRoot enabled resolution operations.

* A LocalRoot implementation MUST revert to using regular DNS for querying the root server when the downloaded zone contains RRTYPEs or cryptographic algorithm types it does not understand.

* A LocalRoot implementation SHOULD use the EDNS EXPIRE Option [RFC7314].

5. Operational Considerations

TBD

6. Security Considerations

There are areas of potential concern that are mitigated to some extent by using this mechanism.

6.1. IANA root zone data security

Secure DNS verification of an obtained copy of the IANA root zone is possible because of the use of the RSS's ZONEMD [RFC8976] record. This allows for the entire zone to be fetched and subsequently verified before being used within recursive resolvers resolvers. DNSSEC provides the same assurance for individual signed resource records sourced from the root zone, including of the ZONEMD record itself.

6.2. Leakage of potentially sensitive information

One privacy concern with the use of DNS is the leakage of potentially sensitive information that may be contained in the query name used in DNS queries. Most root servers (except b.root-servers.net) do not currently support queries over encrypted transports, resulting in query names that are visible to on-the-wire eavesdroppers, and may also be held in any operational logs maintained by root server operators. Such concerns may be mitigated by Query Name Minimization [RFC9156], but common implementations of this mechanism appear to only minimize query names of four or fewer labels, and the uptake rate of query name minimization appears to be quite low [QNAMEMIN]. Furthermore, even with Query Name Minimization, queries for non-existent names (generated from keyword searches and mis-configurations) can cause additional privacy leaks. [RFC8806] eliminates the need for the resolver to perform specific queries to any root nameserver, and obviates any such consideration of query name leakage [LOCALROOTPRIVACY].

6.3. Local resiliency of the DNS

Another issue solved with LocalRoot is that when information is always available locally, usage of it is no longer subject to DDoS attacks against dependent networks and remote servers. By having the answers effectively permanently in cache, no queries to the upstream service provider (such as root servers) are needed since LocalRoot enabled resolvers effectively always have a cached set of data that is considered fresh longer than the typical TTL records within the zone [CACHEME] [LOCALROOTPRIVACY].

7. IANA Considerations

This document contains no requests to IANA, although its companion documents do.

8. References

8.1. Normative References

[BCP237] Best Current Practice 237,
<https://www.rfc-editor.org/info/bcp237>.
At the time of writing, this BCP comprises the following:

Hoffman, P., "DNS Security Extensions (DNSSEC)", BCP 237, RFC 9364, DOI 10.17487/RFC9364, February 2023, <https://www.rfc-editor.org/info/rfc9364>.

[RFC1982] Elz, R. and R. Bush, "Serial Number Arithmetic", RFC 1982, DOI 10.17487/RFC1982, August 1996, <https://www.rfc-editor.org/rfc/rfc1982>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <https://www.rfc-editor.org/rfc/rfc4033>.

[RFC7314] Andrews, M., "Extension Mechanisms for DNS (EDNS) EXPIRE Option", RFC 7314, DOI 10.17487/RFC7314, July 2014, <https://www.rfc-editor.org/rfc/rfc7314>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC8198] Fujiwara, K., Kato, A., and W. Kumari, "Aggressive Use of DNSSEC-Validated Cache", RFC 8198, DOI 10.17487/RFC8198, July 2017, <https://www.rfc-editor.org/rfc/rfc8198>.

[RFC8499] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", RFC 8499, DOI 10.17487/RFC8499, January 2019, <https://www.rfc-editor.org/rfc/rfc8499>.

[RFC8806] Kumari, W. and P. Hoffman, "Running a Root Server Local to a Resolver", RFC 8806, DOI 10.17487/RFC8806, June 2020, <https://www.rfc-editor.org/rfc/rfc8806>.

[RFC8976] Wessels, D., Barber, P., Weinberg, M., Kumari, W., and W. Hardaker, "Message Digest for DNS Zones", RFC 8976, DOI 10.17487/RFC8976, February 2021, <https://www.rfc-editor.org/rfc/rfc8976>.

8.2. Informative References

[BIND-MIRROR] "BIND 9 Mirror Zones", n.d., <https://bind9.readthedocs.io/en/stable/reference.html#namedconf-statement-type%20mirror>.

[CACHEME] "Cache Me If You Can: Effects of DNS Time-to-Live", n.d., <https://ant.isi.edu/~johnh/PAPERS/Moura19b.pdf>.

[DNEROOTNAMES]
"NoError vs NxDomain by-week", n.d., <https://rssac002.root-servers.org/rcode_0_v_3.html>.

[draft-hardaker-dnsop-dns-xfr-scheme]
"The DNS XFR URI Schemes", n.d., <https://datatracker.ietf.org/doc/draft-hardaker-dnsop-dns-xfr-scheme/>.

[draft-hardaker-dnsop-iana-root-zone-publication-points]
"A format for publishing a list of sources of IANA root zone data", n.d., <https://datatracker.ietf.org/doc/draft-hardaker-dnsop-iana-root-zone-publication-points>.

[draft-hardaker-dnsop-root-zone-pub-list-guidelines]
"Guidelines for IANA DNS Root Zone Publication List Providers", n.d., <https://datatracker.ietf.org/doc/draft-hardaker-dnsop-root-zone-pub-list-guidelines>.

[KNOT-PREFILL]
"Knot Resolver Prefill", n.d., <https://knot-resolver.readthedocs.io/en/stable/modules-prefill.html>.

[LOCALROOTPRIVACY]
"Analyzing and mitigating privacy with the DNS root service", n.d., <http://ant.isi.edu/~hardaker/papers/2018-02-ndss-analyzing-root-privacy.pdf>.

[NOROOTS] "On Eliminating Root Nameservers from the DNS", n.d., <https://www.icir.org/mallman/pubs/All19b/All19b.pdf>.

[QNAMEMIN] "DNS Query Privacy", n.d., <https://www.potaroo.net/ispcol/2019-08/qmin.html>.

[RFC2826] IAB, "IAB Technical Comment on the Unique DNS Root", RFC 2826, DOI 10.17487/RFC2826, May 2000, <https://www.rfc-editor.org/rfc/rfc2826>.

[RFC5936] Lewis, E. and A. Hoenes, Ed., "DNS Zone Transfer Protocol (AXFR)", RFC 5936, DOI 10.17487/RFC5936, June 2010, <https://www.rfc-editor.org/rfc/rfc5936>.

[RFC7766] Dickinson, J., Dickinson, S., Bellis, R., Mankin, A., and D. Wessels, "DNS Transport over TCP - Implementation Requirements", RFC 7766, DOI 10.17487/RFC7766, March 2016, <https://www.rfc-editor.org/rfc/rfc7766>.

[RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/rfc/rfc9110>.

[RFC9156] Bortzmeyer, S., Dolmans, R., and P. Hoffman, "DNS Query Name Minimisation to Improve Privacy", RFC 9156, DOI 10.17487/RFC9156, November 2021, <https://www.rfc-editor.org/rfc/rfc9156>.

[RSSAC055] "Principles Guiding the Operation of the Public Root Server System", n.d., <https://itp.cdn.icann.org/en/files/root-server-system-advisory-committee-rssac-publications/rssac-055-07jul21-en.pdf>.

[UNBOUND-AUTH-ZONE] "Unbound Auth Zone", n.d., <https://nlnetlabs.nl/documentation/unbound>.

Acknowledgments

The authors have discussed this idea with many people, and have likely forgotten to acknowledge and credit many of them. If we discussed this with you, and you are not listed, please please let us know and we'll add you.

This work has been founded upon previous documents. Most importantly, [RFC8806], authored by Warren Kumari and Paul Hoffman, and "On Eliminating Root Nameservers from the DNS" [NOROOTS] by Mark Allman.

The authors would like to thank Joe Abley, Vint Cerf, John Crain, Marco Davids, Peter Koch, Matt Larson, Florian Obser, Swapneel Patnekar, Puneet Sood, Robert Story, Ondrej Sury, Suzanne Woolf, and many many others for their comments, suggestions and input to both past and current versions of this document.

In addition, one of the authors would like to once again thank the bands "Infected Mushroom", "Kraftwerk", and "deadmau5" for providing the soundtrack to which this was written. Another author recently discovered the band "Trampled by Turtles" while working on this document and is submitting it as a nomination for the best-band-name-ever award.

History of the LocalRoot concept

Note: DNSOP needs to discuss whether to publish this as a BCP or as a proposed standard. Currently this is listed as STD track based on a number of preliminary conversations the authors had with both operators and IETF participants.

[RFC8806] is an Informational document that describes a mechanism that resolver operators can use to improve the performance, reliability, and privacy of their resolvers. This document concludes the concept of [RFC8806] was a success, but that actual implementation of it has varied according to the needs of various code bases and operational environments. Thus, this document houses many of the original concepts of [RFC8806] but is largely a complete rewrite to match modern expectations based on recent implementation and deployment experiences.

This document differs in a number of critical ways (TBD: this list is incomplete):

1. promotes the behavior in [RFC8806] to be either a Proposed standard or a Best Current Practice, depending on what the WG decides.

2. RECOMMENDS that resolver implementations provide a simple configuration option to enable or disable functionality, and

3. RECOMMENDS that resolver implementations enable this behavior by default, and

4. REQUIRES that [RFC8976] be used to validate the IANA root zone information before loading it.

5. Adds a mechanism for priming the list of places for fetching root zone data.

6. Adds protocol steps for ensuring resolution stability and resiliency.

An important change from RFC8806

[RFC8806] Section 2 (Requirements) states that:

> The system MUST be able to run an authoritative service for the root zone on the same host. The authoritative root service MUST only respond to queries from the same host. One way to assure not responding to queries from other hosts is to run an authoritative server for the root that responds only on one of the loopback

addresses (that is, an address in the range 127/8 for IPv4 or ::1 in IPv6). Another method is to have the resolver software also act as an authoritative server for the root zone, but only for answering queries from itself.

This document relaxes this requirement. Resolver implementations can achieve the desired behavior of directly serving the contents of the root zone via multiple implementation choices, beyond those listed in [RFC8806]. This can include the implementation guidance described in RFC8806, but this document allows for implementations to select any mechanism for fetching and re-distributing the contents of the root zone on their resolver service addresses as long as the other requirements specified in this document are still followed (see Section 4).

For example, an implementation can simply "prefill" the resolver's cache with the current contents of the root zone. As the resulting behavior is (essentially) indistinguishable from the mechanism defined in RFC8806, this is viewed as being an acceptable implementation decision.

Authors' Addresses

Warren Kumari
Google, Inc.
Email: warren@kumari.net

Wes Hardaker
USC/ISI and Google, Inc.
Email: ietf@hardakers.net

Jim Reid
RTFM llp
St Andrews House
382 Hillington Road, Glasgow Scotland
G51 4BL
United Kingdom
Email: jim@rfc1035.com

Geoff Huston
APNIC
6 Cordelia St
South Brisbane QLD 4101
Australia
Email: gih@apnic.net