Domain Name System Operations
Internet-Draft
Updates: RFC8806 (if approved)
Intended status: Best Current Practice
Expires: 2 March 2026

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

29 August 2025

Making LocalRoot a Best Current Practice
draft-wkumari-dnsop-localroot-bcp-00

Abstract

RFC 8806 (often called "LocalRoot") defines a mechanism whereby a recursive resolver can fetch the contents of an entire zone and place this information into the resolver’s cache.

This has several benefits, including increased reliability, increased performance, improved privacy, and decreased or mitigating the effect of some types of DoS attacks.

While the majority of DNS resolver implementations natively support RFC 8806, it remains tricky to configure and maintain. This document recommends that DNS resolver software simplify this configuration, and further suggests that configuration becomes the default.

This document updates Section 2 of RFC8806 by relaxing the requirement that implementations MUST run an authoritative service.

/* Ed (WK): Open questions / ToDo / Notes (to be removed before publication):

1. I started writing this as rfc8806-bis, but as I did so I realized that it is likely better as a standalone document.

2. DONE - Add Zone Checksum

3. DONE - Look up BIND and Unbound support.

4. This document recommends ("Operation Considerations") using HTTP. Need to discuss the bootstrapping issue, and load balancing.

5. Security Considerations - flesh this out. I think that it just contains descriptions of the benefits from RFC8806, but I suspect there will be some other concerns too.

```
   */
```

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

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://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 2 March 2026.

Copyright Notice

Table of Contents

1. Introduction

[RFC8806] provides "a method for the operator of a recursive resolver to have a complete root zone locally, and to hide queries for the root zone from outsiders. The basic idea is to create an up-to-date root zone service on the same host as the recursive server, and use that service when the recursive resolver looks up root information."

While [RFC8806] behavior can be achieved by "manually" configuring software that acts as a secondary server for the root-zone (see [RFC8806] Section B.1. Example Configuration: BIND 9.12 and Section B.2 Example Configuration: Unbound 1.8), most resolver implementations now support simpler, and more robust, configuration mechanisms to enable this support. For example, ISC BIND 9.14 and above supports "mirror" zones, Unbound 1.9 supports "auth-zone", and Knot Resolver uses its "prefill" module to load the root zone information. See Appendix A for configuration details. In addition to providing simpler configuration of the LocalRoot mechanism, these mechanisms support "falling back" to querying the root-servers directly if they are unable to fetch the entire root zone.

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.

3. Making RFC8806 behavior be a Best Current Practice

[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:

1. promotes the behavior in [RFC8806] to be a Best Current Practice.

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. RECOMMENDS that [RFC8976] be used to validate the zone information before loading it.

4. Changes 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. Some resolver implementations achieve the behavior described in RFC8806 by fetching the zone information and "prefilling" their cache with this information. As the resulting behavior is (essentially) indistinguishable from the mechanism defined in RFC8806, this is viewed as being an acceptable implementation decision.

5. Applicability

This behavior should apply to all general-purpose recursive resolvers used on the public Internet.

6. Operational Considerations

In order for the [RFC8806] mechanism to be effective, a resolver must be able to fetch the contents of the entire root zone. This is currently usually performed through AXFR ([RFC5936]). In order for AXFR to work, the resolver must be able to use TCP (which is already required by [RFC7766]).

Resolvers MAY allow fetching this information via HTTPS. Where possible, HTTPS should be preferred as it will allow for compression as well as the possibility of using low-cost, well-distributed CDNs to distribute the zone files.

/* ED (WH): I don’t think we can get away without describing how/ where to pull this information from at some point. The ICANN https servers are one source, or should resolver code bases use their own defined CDNs? */

Resolvers MUST validate the contents of the zone before using it. This SHOULD be done using the mechanism in [RFC8976], but MAY be done by validating every signed record in a zone with DNSSEC [RFC9364].

/* Ed (WK): We might want to add some more discussions around failure handling, but, 1: [RFC8806] already covers much of this and 2: "don’t teach your grandmother to suck eggs" - implementations already handle this, so let’s not try to overspecify or overconstrain what they do. */

/* Ed (GH): As the NS records are unsigned the possibility of tampering with the root zone exists through these unsigned NS records. For this reason ZONEMD should be strongly recommended, or even MUST be used.*/

/* Ed (WH): I agree with GH, and said as much in [LOCALROOTPRIVACY] */

7. Security Considerations

There are three areas of potential concern that can be mitigated to some extent by using this mechanism, coupled with the use of [RFC8976].

The first is the potential to insert corrupted referral address records in response to queries to a root server. The referral addresses provided in a referral response is not a DNSSEC-signed record in the root zone, and thus there is the potential for an on-the-wire insertion attack by replacing this part of a referral response with a different address set. If ZONEMD is used to authenticate the the local copy of the root zone, such on-the-wire attacks are not feasible.

The second is the issue of leak 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].

The final issue solved with LocalRoot is that when information is always available locally, usage of it is no longer subject to DDoS attacks against the remote servers. By having the answers effectively permanently in cache, no queries to the upstream service provider (such as root servers) are needed since [RFC8806] resolvers effectively always have a cached set of data that is considered fresh longer than the typical TTL records within the zone [CACHEME] [LOCALROOTPRIVACY].

/* Ed (WK): Fill this in. I think that it just contains descriptions of the benefits from RFC8806, but I’m guessing that there are some other concerns too... */

Security requirements associated with the need to verify that the contents of the retrieved root zone are correct were discussed above, and mitigated by the usage of [RFC8976].

8. IANA Considerations

This document has no IANA actions.

9. References

9.1. Normative References

[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>.

[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>.

[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>.

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

9.2. 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>.

[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>.

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

[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>.

[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>.

[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.

The authors would like to thank Vint Cerf, John Crain, Puneet Sood, Robert Story, Suzanne Woolf.

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.

Appendix A: Example Configurations

These examples are provided to show how the LocalRoot mechanism can be configured in various resolver implementations. They are not intended to be exhaustive, and may not work with all versions of the software.

/* Ed (WK): These examples are just to get started. We would appreciate contributions from the resolver operators.

Yes, we are fully aware of the circular dependency of trying to resolve e.g www.internic.net when bootstrapping. More discussion on serving the root zone over HTTP by IP will be added later. */

ISC BIND 9.14 and above

See the BIND documentation for mirror zones (https://bind9.readthedocs.io/en/stable/reference.html#namedconf-statement-type%20mirror).

Example configuration using a "mirror" zone:

```
zone "." {
    type mirror;
};
```

Knot Resolver

See the Knot Resolver Cache prefilling (https://knot-resolver.readthedocs.io/en/v5.0.1/modules-prefill.html?highlight=cache%20prefilling) documentation for more information.

The following example configuration will prefill the root zone using HTTPS:

```
modules.load('prefill')
prefill.config({
      ['.'] = {
              url = 'https://www.internic.net/domain/root.zone',
              interval = 86400  -- seconds
              ca_file = '/etc/pki/tls/certs/ca-bundle.crt', -- optional
      }
})
```

Unbound 1.9 and above

See the Unbound documentation for Authority Zone Options (https://unbound.docs.nlnetlabs.nl/en/latest/manpages/unbound.conf.html#unbound-conf-auth-url) configuration.

The following example configuration will prefill the root zone using HTTPS:

```
   auth-zone:
     name: "."
     url: "https://www.internic.net/domain/root.zone"
     zonefile: "root.zone"
           fallback-enabled: yes
       for-downstream: no
       for-upstream: yes
       zonefile: "root.zone"
     prefetch: yes
```

Authors' Addresses

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

Wes Hardaker
USC/ISI
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