Network Working Group
Internet-Draft
Obsoletes: 8806 (if approved)
Intended status: Standards Track
Expires: 17 September 2026

P. Hoffman
ICANN

16 March 2026

RootCache: Filling Resolver Caches with Root Zone Records
draft-hoffman-rootcache-01

Abstract

Some DNS recursive resolver operators want to prevent snooping by third parties of requests sent to DNS root servers. Resolvers can reduce the number of queries sent to root server, and thus prevent observation of requests, by caching a copy of the full root zone. This document shows how a resolver can securely receive the full root zone and put it into the resolver's cache.

This document obsoletes RFC 8806.

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 17 September 2026.

Copyright Notice

Table of Contents

1. Introduction

This document describes RootCache, a method for a resolver to quickly and securely fill its cache with the entire root zone. RootCache is an enhancement to resolver operations, but does not change the DNS protocol used in those resolvers at all.

[[Copied from RFC 8806 with minor clarifications.]]

DNS recursive resolvers have to provide answers 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 root server of the global DNS to get the information for that TLD or to find out that the TLD does not exist. Research shows that the vast majority of queries going to the root are for names that do not exist in the root zone.

Many of the queries from recursive resolvers to root servers get answers that are referrals to other servers. Malicious third parties might be able to observe that traffic on the network between the recursive resolver and root servers.

A different approach to solving some of the problems discussed in this document is described as a standalone solution in [RFC8198] and [RFC9077]. This approach is included as part of RootCache.

Readers are expected to be familiar with [RFC9499].

1.1. RootCache Use Case

[[Copied from RFC 8806, but with changes to the goals.]]

The primary goals of RootCache are to provide more reliable answers for queries to the root zone. Using RootCache will probably have little effect on getting faster responses to the stub resolver for good queries on TLDs, because the TTL for most TLDs is usually long-lived (on the order of a day or two) and is thus usually already in the cache of the recursive resolver; the same is true for the TTL for negative answers from the root servers.

1.2. RootCache Design

RootCache is a method for the operator of a recursive resolver to have a complete root zone in their cache and to prevent queries going to the root servers. This in turn prevents outsiders from seeing what queries the resolver's clients have made because those queries are never sent to the root servers.

The basic idea is to create an up-to-date set of root zone answers in the cache of the recursive server, if possible. The recursive resolver validates all contents of the root zone before putting them in its cache, just as it would validate all responses from a remote root server. It only puts the root zone records in the cache if doing so would not force out other records.

RootCache adds records to a resolver's cache, but does not change the way the cache works. For example, if the operator stops running RootCache (either intentionally or accidentally), the cache acts exactly the same.

1.3. Differences Between This Document and RFC 8806

The core design of [RFC8806] was that resolvers would locally act as root servers. That design has many failure cases that need to be dealt with in by resolver sofware. The core design of RootCache is that resolvers will fill their cache with the contents of the root zone so that queries do not need to go to root servers unless the records in their cache time out. Failures to fill the cache do not cause any failure cases for the resolver as a whole.

[RFC8806] focused on getting the root zone by AXFR requests to root server operators. RootCache expands that by giving a standard way to get the root zone over HTTPS.

This document assumes that the vast majority of resolver operators will use the default configurations that come with their resolver software.

This document removes discussion of "faster responses", but will add it back in later versions if there is research to show that faster responses are measurably better for resolver operators or their users.

1.4. BCP 14 Language

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. RootCache Requirements and Operation

[[The following paragraph and the first three bullets are copied from RFC 8806.]]

In order to implement the RootCache mechanism described in this document:

* The system MUST be able to validate every signed record in a zone with DNSSEC [RFC9364].

* The system MUST have an up-to-date copy of the public part of the Key Signing Key (KSK) [RFC9364] used to sign the DNS root.

* The system MUST be able to retrieve a copy of the entire root zone (including all DNSSEC-related records).

* The system MUST be able to verify the contents of the root zone data using the ZONEMD record [RFC8976] from the root zone.

3. The system SHOULD implement aggressive use of the DNSSEC-validated cache as described in [RFC8198] and [RFC9077].

In order to enhance its cache with RootCache, a resolver performs the following steps in order.

3.1. Retrieval

The resolver periodically retrieves the entire root zone. It can do this from any source; see Section 4 for information on where the zone might be found and Section 5 for considerations on how often to refresh.

3.2. Verification

The contents of the retrieved root zone MUST be verified for completeness by checking it against the ZONEMD record in the zone, using the methods described in [RFC8976]. If the ZONEMD verification fails, the retrieved zone MUST be abandoned; the resolver SHOULD then try other configured sources.

3.3. Validation

After validating the contents of that root zone, every record in the root zone MUST be validated using DNSSEC [RFC9364]. If the DNSSEC validation fails, the retrieved zone MUST be abandoned; the resolver SHOULD then try other configured sources.

Performing this step is REQUIRED even for resolvers that are not configured to do DNSSEC validation for queries from clients.

3.4. Comparison

The resolver MUST compare the serial number in the SOA record in the retrieved root zone against the serial number in the SOA record for the root zone in its cache. If the serial number in the retrieved record is higher, or there is no SOA record for the root zone in the cache, the records of the retrieved root zone can be added to the resolver’s cache, replacing any records with the same name/class/type

triple that are already in the cache or adding new records.

However, if adding these newly-retrieved records to the cache would force other records out of the cache, the resolver SHOULD NOT add the new root records. Adding the new root records would cause one set of queries to not be sent on the wire, but would cause a different set of records to be forced out early and thus might expose a different set of queries on the wire. Implementations MAY choose to simply replace current root-level records in the cache with the ones received, or they MAY simply ignore the new records.

3.5. Reducing Queries to the Root Servers Even Further

Resolvers implementing RootCache SHOULD also implement aggressive use of the DNSSEC-validated cache as described in [RFC8198] and [RFC9077]. This process, often called "aggressive NSEC", will prevent queries that would get negative replies from being sent to the root server system.

4. Sources of the Root Zone

[[Loosely copied from RFC 8806, with additions.]]

The root zone can be retrieved from anywhere as long as it comes with all the DNSSEC records needed for validation.

Currently, a resolver can get the root zone from ICANN by zone transfer AXFR (see [RFC5936]) over TCP from DNS servers at xfr.lax.dns.icann.org and xfr.cjr.dns.icann.org.

[[Should likely talk about [RFC9103].]]

Currently, there is a description of how the root zone file can be obtained from IANA (https://www.iana.org/domains/root/files).

Currently, the root can also be retrieved by AXFR over TCP from many of the root server operators at their service addresses. It can also be retrieved from the LocalRoot service (https://localroot.isi.edu/).

It is crucial to note that none of the above services are guaranteed to be available. It is possible that ICANN or some of the root server operators will turn off the AXFR capability on the DNS servers. Using AXFR over TCP to addresses that are likely to be anycast (as the ones above are) may conceivably have transfer problems due to anycast, but current practice shows that to be unlikely.

4.1. Root Zone Sources over HTTPS

[[Readers familiar with "/.well-known/" will want to review this carefully. The wording here is quite likely to change]]

Since the publication of [RFC8806], there has been an increased desire to be able to retrieve the root zone over HTTPS. This section shows a method for operators of web services that want to publish the root zone to make the zone easily findable, using the "/.well-known/" URL path prefix ([RFC8615]).

Web serververs that offer to serve the root zone, they SHOULD do so at an HTTPS URL whose path component is exactly "/.well-known/dns-root-zone/". Thus, a client who wants to get the root zone from the HTTPS web server at example.com would use the URL "https://example.com//.well-known/dns-root-zone/".

4.2. Configuration of Sources for RootCache

It seems likely that the vast majority of resolver operators will use the default configurations that come with their resolver software. Based on that, this document assumes that the list of sources for root zone information will be at least initially collected by the resolver software implementers. Those implementers are well-positioned to find and test sources, and to update their software when the list of good sources changes.

Resolver software that implements RootCache SHOULD come with a list of at least five sources of the root zone that are known at the time that the software is released. It SHOULD also allow the resolver operator to change the list of sources for the root zone.

It seems likely that some people will create and maintain lists of sources for RootCache zones. This is not a requirement for RootCache to be successful, but there might be a population of resolver operators who want to use a wider set of sources than what their resolver developer gives them in any particular software release.

5. Refresh Period

[[Discussion in the DNSOP WG in January 2026 indicates that determining this value will be contentious, at least initially. This is really just a placeholder, and it is likely that this section will have multiple orthogonal methods.]]

The resolver SHOULD get a new copy of the root zone from any of its configured sources approximately twice a day. This value is based on the TTLs of the records in the root zone in early 2026. These TTLs

have been the same for over a decade, but IANA could change in the future. Any such change in the root zone could change the values given here.

6. IANA Considerations

6.1. Registration in the the "Well-Known URIs" Registry

(This template is based on [RFC8615].)

URI suffix: dns-root-zone

Change controller: IETF

Specification document(s): This document

Status: permanent

Related information: N/A

7. Security Considerations

This document assumes that all sources of the root zone serve it unaltered, regardless of the protocol used to retrieve it. Further, it assumes that all such sources make a best-faith effort to serve quite fresh versions, and that those sources will stop service if they are unable to get fresh versions themselves.

If any of the MUST-level requirements in Section 3.2, Section 3.3, or Section 3.4 are not followed, a resolver can be tricked into serving bad data for records from the root zone.

A resolver that does not implement aggressive NSEC as described in Section 3.5 will leak negative queries to the root server system.

[[More to come]]

8. References

8.1. Normative References

[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/info/rfc8198>.

[RFC8615] Nottingham, M., "Well-Known Uniform Resource Identifiers (URIs)", RFC 8615, DOI 10.17487/RFC8615, May 2019, <https://www.rfc-editor.org/info/rfc8615>.

[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/info/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/info/rfc8976>.

[RFC9077] van Dijk, P., "NSEC and NSEC3: TTLs and Aggressive Use", RFC 9077, DOI 10.17487/RFC9077, July 2021, <https://www.rfc-editor.org/info/rfc9077>.

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

[RFC9499] Hoffman, P. and K. Fujiwara, "DNS Terminology", BCP 219, RFC 9499, DOI 10.17487/RFC9499, March 2024, <https://www.rfc-editor.org/info/rfc9499>.

[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/info/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/info/rfc8174>.

8.2. Informative References

[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/info/rfc5936>.

[RFC9103] Toorop, W., Dickinson, S., Sahib, S., Aras, P., and A. Mankin, "DNS Zone Transfer over TLS", RFC 9103, DOI 10.17487/RFC9103, August 2021, <https://www.rfc-editor.org/info/rfc9103>.

Appendix A. Acknowledgements

The discussions in the DNSOP Working Group led to many ideas here about improvements to [RFC8806]. Thanks go to Warren Kumari, Wes Hardaker, Jim Reid, and Geoff Huston for authoring and getting discussion going with a draft on "Populating resolvers with the root zone". Particular thanks go to Warren Kumari for co-authoring [RFC8806] (and RFC 7706 that preceded it).

Author’s Address

Paul Hoffman
ICANN
Email: paul.hoffman@icann.org