Workgroup: Network Working Group
Internet-Draft: draft-ietf-dnsop-rfc8109bis-05
Obsoletes: 8109 (if approved)
Published: 5 June 2024
Intended Status: Best Current Practice
Expires: 7 December 2024
Authors: P. Koch DENIC eG
M. Larson ICANN
P. Hoffman ICANN

Initializing a DNS Resolver with Priming Queries

Abstract

This document describes the queries that a DNS resolver should emit to initialize its cache. The result is that the resolver gets both a current NS Resource Record Set (RRset) for the root zone and the necessary address information for reaching the root servers.

This document, when published, obsoletes RFC 8109. See Section 1.1 for the list of changes from RFC 8109.

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 7 December 2024.

Table of Contents

1. Introduction

Recursive DNS resolvers need a starting point to resolve queries. [RFC1034] describes a common scenario for recursive resolvers: they begin with an empty cache and some configuration for finding the names and addresses of the DNS root servers. [RFC1034] describes that configuration as a list of servers that will give authoritative answers to queries about the root. This has become a common implementation choice for recursive resolvers, and is the topic of this document.

This document describes the steps needed for this common implementation choice. Note that this is not the only way to start a recursive name server with an empty cache, but it is the only one described in [RFC1034]. Some implementers have chosen other directions, some of which work well and others of which fail (sometimes disastrously) under different conditions. For example, an implementation that only gets the addresses of the root name servers from configuration, not from the DNS as described in this document, will have stale data that could cause slower resolution.

This document only deals with recursive name servers (recursive resolvers, resolvers) for the IN class.

1.1. Changes from RFC 8109

This document obsoletes [RFC8109]. The significant changes from RFC 8109 are:

- Added section on the content of priming information.
- Added paragraph about no expectation that the TC bit in responses will be set.
- Added paragraph about RFC 9471 and requirements on authoritative servers and the TC bit. This clarified the role of glue records and truncation for responses from the root zone.
- Changed "man-in-the-middle" to "machine-in-the-middle" to be both less sexist and more technically accurate.
- Clarified that there are other effects of machine-in-the-middle attacks.
- Clarified language for root server domain names as "root server identifiers".
- Added short discussion of post-priming strategies.
- Added informative references to RSSAC documents.
- Added short discussion about this document and private DNS.
- Clarified that machine-in-the-middle attacks could be successful for non-signed TLDs.
- Added discussion of where resolvers that pre-fetch should get the root NS addresses.
- Elevated the expectations in "Expected Properties of the Priming Response" to MUST-level.
- Clarified that "currently" means at the time that this document is published.
- Added a note about priming and RFC 8806.
- Added a reference to research about discontinued root server addresses.

1.2. Terminology

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.

See [RSSAC026v2] for terminology that relates to the root server system.

2. Description of Priming

Priming is the act of finding the list of root servers from a configuration that lists some or all of the purported IP addresses of some or all of those root servers. In priming, a recursive resolver starts with no cached information about the root servers, and finishes with a full list of their names and their addresses in its cache.

Priming is described in Sections 5.3.2 and 5.3.3 of [RFC1034]. (It is called "SBELT", a "safety belt" structure, in that document.) The scenario used in that description, that of a recursive server that is also authoritative, is no longer as common.

The configured list of IP addresses for the root servers usually comes from the vendor or distributor of the recursive server software. This list is usually correct and complete when shipped, but may become out of date over time.

The domain names for the root servers are called the "root server identifiers". This list has been stable since 1997, but the IPv4 and IPv6 addresses for the root server identifiers sometimes change. Research shows that after those addresses change, some resolvers never get the new addresses; for example, see [OLD-J].

Therefore, it is important that resolvers be able to cope with change, even without relying upon configuration updates to be applied by their operator. Root server identifier and address changes are the main reasons that resolvers need to use priming to get a full and accurate list of root servers, instead of just using a statically configured list.

See [RSSAC023v2] for a history of the root server system.

Although this document is targeted at the global DNS, it also could apply to a private DNS as well. These terms are defined in [RFC8499].

Some systems serve a copy of the full root zone on the same server as the resolver, such as is described in [RFC8806]. In such a setup, the resolver primes its cache using the same methods as described in the rest of this document.

2.1. Content of Priming Information

As described above, the configuration for priming is a list of IP addresses. The priming information in software may be in any format that gives the software the addresses associated with at least some of the root server identifiers.

Some software has configuration that also contains the root server identifiers, sometimes as comments and sometimes as data consumed by the software. For example, IANA's "Root Hints File" at <https://www.internic.net/domain/named.root> is derived directly from the root zone and contains all of the addresses of the root server identifiers found in the root zone. It is in DNS zone file presentation format, and includes the root server identifiers. Although there is no harm to adding such information, it is not useful in the root priming process.

3. Priming Queries

A priming query is a DNS query whose response provides root server names and addresses. It has a QNAME of ".", a QTYPE of NS, and a QCLASS of IN; it is sent to one of the addresses in the configuration for the recursive resolver. The priming query can be sent over either UDP or TCP. If the query is sent over UDP, the source port SHOULD be randomly selected (see [RFC5452]). The Recursion Desired (RD) bit MAY be set to 0 or 1, although the meaning of it being set to 1 is undefined for priming queries.

The recursive resolver SHOULD use EDNS0 [RFC6891] for priming queries and SHOULD announce and handle a reassembly size of at least 1024 octets [RFC3226]. Doing so allows responses that cover the size of a full priming response (see Section 4.2) for the current set of root servers. See Section 3.3 for discussion of setting the DNSSEC OK (DO) bit (defined in [RFC4033]).

3.1. Repeating Priming Queries

The recursive resolver SHOULD send a priming query only when it is needed, such as when the resolver starts with an empty cache or when the NS RRset for the root zone has expired. Because the NS records for the root zone are not special, the recursive resolver expires those NS records according to their TTL values. (Note that a recursive resolver MAY pre-fetch the NS RRset before it expires.)

If a resolver chooses to pre-fetch the root NS RRset before that RRset has expired in its cache, it needs to choose whether to use the addresses for the root NS RRset that it already has in its cache or to use the addresses it has in its configuration. Such a resolver SHOULD send queries to the addresses in its cache in order to reduce

the chance of delay due to out-of-date addresses in its configuration.

If a priming query does not get a response, the recursive resolver MUST retry the query with a different target address from the configuration.

3.2. Target Selection

In order to spread the load across all the root server identifiers, the recursive resolver SHOULD select the target for a priming query randomly from the list of addresses. The recursive resolver might choose either IPv4 or IPv6 addresses based on its knowledge of whether the system on which it is running has adequate connectivity on either type of address.

Note that this recommended method is not the only way to choose from the list in a recursive resolver's configuration. Two other common methods include picking the first from the list, and remembering which address in the list gave the fastest response earlier and using that one. There are probably other methods in use today. However, the random method listed above SHOULD be used for priming.

3.3. DNSSEC with Priming Queries

The root NS RRset is signed and can be validated by a DNSSEC validating resolver. At the time this document is published, the addresses for the names in the root NS RRset are in the "root-servers.net" zone. All root servers are also authoritative for the "root-servers.net" zone, which allows priming responses to include the appropriate root name server A and AAAA RRsets. However, because at the time this document is published the "root-servers.net" zone is not signed, the root name server A and AAAA RRsets cannot be validated. An attacker that is able to provide a spoofed priming response can provide alternative A and AAAA RRsets and thus fool a resolver into considering addresses under the control of the attacker to be authoritative for the root zone.

A rogue root name server can view all queries from the resolver to the root and alter all unsigned parts of responses, such as the parent side NS RRsets and glue in referral responses. A resolver can be fooled into trusting child (TLD) NS addresses that are under the control of the attacker as being authoritative if the resolver:

*follows referrals from a rogue root server,

*and does not explicitly query the authoritative NS RRset at the apex of the child (TLD) zone,

*and does not explicitly query for the authoritative A and AAAA RRsets for the child (TLD) NS RRsets.

With such resolvers, an attacker that controls a rogue root server effectively controls the entire domain name space and can view all queries and alter all unsigned data undetected.

An attacker controlling a rogue root name server also has complete control over all unsigned delegations, and over the entire domain name space in case of non DNSSEC validating resolvers.

If the "root-servers.net" zone is later signed, or if the root servers are named in a different zone and that zone is signed, having DNSSEC validation for the priming queries might be valuable. The benefits and costs of resolvers validating the responses will depend heavily on the naming scheme used.

4. Priming Responses

A priming query is a normal DNS query. Thus, a root server cannot distinguish a priming query from any other query for the root NS RRset. Thus, the root server's response will also be a normal DNS response.

4.1. Expected Properties of the Priming Response

The priming response MUST have an RCODE of NOERROR, and MUST have the Authoritative Answer (AA) bit set. Also, it MUST have an NS RRset in the Answer section (because the NS RRset originates from the root zone), and an empty Authority section (because the NS RRset already appears in the Answer section). There will also be an Additional section with A and/or AAAA RRsets for the root servers pointed at by the NS RRset.

Resolver software SHOULD treat the response to the priming query as a normal DNS response, just as it would use any other data fed to its cache. Resolver software SHOULD NOT expect 13 NS RRs because, historically, some root servers have returned fewer.

4.2. Completeness of the Response

At the time this document is published, there are 13 root server operators operating a total of more than 1500 root server instances. Each has one IPv4 address and one IPv6 address. The combined size of all the A and AAAA RRsets exceeds the original 512-octet payload limit from [RFC1035].

In the event of a response where the Additional section omits certain root server address information, re-issuing of the priming query does not help with those root name servers that respond with a fixed order

of addresses in the Additional section. Instead, the recursive resolver needs to issue direct queries for A and AAAA RRsets for the remaining names. At the time this document is published, these RRsets would be authoritatively available from the root name servers.

If some root server addresses are omitted from the Additional section, there is no expectation that the TC bit in the response will be set to 1. At the time that this document is written, many of the root servers are not setting the TC bit when omitting addresses from the Additional section.

Note that [RFC9471] updates [RFC1035] with respect to the use of the TC bit. It says "If message size constraints prevent the inclusion of all glue records for in-domain name servers, the server must set the TC (Truncated) flag to inform the client that the response is incomplete and that the client should use another transport to retrieve the full response." Because the priming response is not a referral, root server addresses in the priming response are not considered glue records. Thus, [RFC9471] does not apply to the priming response and root servers are not required to set the TC bit if not all root server addresses fit within message size constraints. There are no requirements on the number of root server addresses that a root server must include in a priming response.

5. Post-Priming Strategies

When a resolver has a zone's NS RRset in cache, and it gets a query for a domain in that zone that cannot be answered from its cache, the resolver has to choose which NS to send queries to. (This statement is as true for the root zone as for any other zone in the DNS.) Two common strategies for choosing are "determine the fastest name server and always use it" and "create buckets of fastness and pick randomly in the buckets". This document gives no preference to any particular strategy other than to suggest that resolvers not treat the root zone as special for this decision.

6. Security Considerations

Spoofing a response to a priming query can be used to redirect all of the queries originating from a victim recursive resolver to one or more servers for the attacker. Until the responses to priming queries are protected with DNSSEC, there is no definitive way to prevent such redirection.

An on-path attacker who sees a priming query coming from a resolver can inject false answers before a root server can give correct answers. If the attacker's answers are accepted, this can set up the ability to give further false answers for future queries to the resolver. False answers for root servers are more dangerous than,

say, false answers for Top-Level Domains (TLDs), because the root is the highest node of the DNS. See Section 3.3 for more discussion.

In both of the scenarios above, a validating resolver will be able to detect the attack if its chain of queries comes to a zone that is signed, but not for those that are unsigned.

7. IANA Considerations

This document does not require any IANA actions.

8. References

8.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[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>.

[RFC3226] Gudmundsson, O., "DNSSEC and IPv6 A6 aware server/resolver message size requirements", RFC 3226, DOI 10.17487/RFC3226, December 2001, <https://www.rfc-editor.org/info/rfc3226>.

[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/info/rfc4033>.

[RFC5452] Hubert, A. and R. van Mook, "Measures for Making DNS More Resilient against Forged Answers", RFC 5452, DOI 10.17487/RFC5452, January 2009, <https://www.rfc-editor.org/info/rfc5452>.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <https://www.rfc-editor.org/info/rfc6891>.

[RFC8109] Koch, P., Larson, M., and P. Hoffman, "Initializing a DNS Resolver with Priming Queries", BCP 209, RFC 8109, DOI

10.17487/RFC8109, March 2017, <https://www.rfc-editor.org/info/rfc8109>.

[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>.

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

[RFC9471] Andrews, M., Huque, S., Wouters, P., and D. Wessels, "DNS Glue Requirements in Referral Responses", RFC 9471, DOI 10.17487/RFC9471, September 2023, <https://www.rfc-editor.org/info/rfc9471>.

8.2. Informative References

[OLD-J] Wessels, D., "Thirteen Years of 'Old J Root'", 2015, <https://indico.dns-oarc.net/event/24/contributions/378/>.

[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>.

[RSSAC023v2] "History of the Root Server System", 2016, <https://www.icann.org/en/system/files/files/rssac-023-17jun20-en.pdf>.

[RSSAC026v2] "RSSAC Lexicon", 2020, <https://www.icann.org/en/system/files/files/rssac-026-lexicon-12mar20-en.pdf>.

Appendix A. Acknowledgements

RFC 8109 was the product of the DNSOP WG and benefitted from the reviews done there. This document also benefitted from review by Duane Wessels.

Authors' Addresses

Peter Koch
DENIC eG
Kaiserstrasse 75-77
60329 Frankfurt
Germany

Phone: +49 69 27235 0
Email: pk@DENIC.DE

Matt Larson

ICANN

Email: matt.larson@icann.org

Paul Hoffman
ICANN

Email: paul.hoffman@icann.org