Domain Name System Operations
Internet-Draft
Intended status: Standards Track
Expires: 22 April 2026

S. Huque
Salesforce
P. Vixie
SIE Europe, U.G.
W. Toorop
NLnet Labs

19 October 2025

Delegation Revalidation by DNS Resolvers
draft-ietf-dnsop-ns-revalidation-11

Abstract

This document describes an optional algorithm for the processing of Name Server (NS) resource record (RR) sets (RRsets) during iterative resolution, and describes the benefits and considerations of using this approach. When following a referral response from an authoritative server to a child zone, DNS resolvers should explicitly query the authoritative NS RRset at the apex of the child zone and cache this in preference to the NS RRset on the parent side of the zone cut. The (A and AAAA) address RRsets in the additional section from referral responses and authoritative NS answers for the names of the NS RRset, should similarly be re-queried and used to replace the entries with the lower trustworthiness ranking in cache. Resolvers should also periodically revalidate the delegation by re-querying the parent zone at the expiration of the TTL of either the parent or child NS RRset, whichever comes first.

About This Document

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

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-dnsop-ns-revalidation/.

Discussion of this document takes place on the DNSOP 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/shuque/ns-revalidation.

Status of This Memo

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

Copyright Notice

Table of Contents

1. Introduction

This document recommends improved DNS [RFC1034] [RFC1035] resolver behavior with respect to the processing of NS record sets during iterative resolution.

In Upgrading NS RRset Credibility (Section 3) we recommend that resolvers, when following a referral response from an authoritative server to a child zone, should explicitly query the authoritative NS RRset at the apex of the child zone and cache this in preference to the NS RRset on the parent side of the zone cut.

Upgrading NS RRset Credibility (Section 3) works most reliably with good quality child NS RRsets, where the name servers referenced in the RDATA of the NS RRset correspond to IP addresses of nameservers which correctly serve the child zone. We consider both the root, as well as the zones delegated from the root, to have good quality child NS RRset. We also note that there may be numerous zones further down the DNS delegation hierarchy that may not be competently administered, and may have incorrect or stale authoritative NS and associated address records. As a result they may require a resolver to fallback to data from the delegating side of the zone cut for successful resolution. Section 4 describes limiting the upgrading of NS RRset credibility to address this.

The address records in the additional section from the referral response (as glue) or authoritative NS response that match the names of the NS RRset should similarly be re-queried if they are cached non-authoritatively. The authoritative answers from those queries should replace the cached non-authoritative A and AAAA RRsets. This is described in (see Upgrading A and AAAA RRset Credibility (Section 5)).

In Strict and opportunistic revalidation (Section 6), we make a distinction between strict and opportunistic revalidation. Strict revalidation provides DNSSEC protection of infrastructure data (Section 9.1) with DNSSEC signed infrastructure data and validating resolvers, but for it to work correctly, good quality child NS RRsets are a pre-requisite. Opportunistic revalidation allows for fallback to the non-authoritative data returned in the referral responses, but therefore does not provide the same degree of protection as strict revalidation does.

Finally, in Delegation Revalidation (Section 7), we recommend that resolvers revalidate the delegation by re-querying the parent zone at the expiration of the TTL of the parent side NS RRset.

1.1. 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.

Throughout this document we will also use terminology with the meaning as defined below:

Triggering query:
: the DNS query that caused ("triggered") a referral response.

Infrastructure RRsets (data):
: the NS and address (A and AAAA) RRsets used by resolvers to contact the authoritative name servers

Revalidation:
: the process of obtaining the authoritative infrastructure data

Validation (validating) query:
: the extra query that is performed to get the authoritative version of infrastructure RRsets

Delegation revalidation:
: re-establishing the existence and validity of the parent-side NS RRset of a delegation

Revalidation point:
: a delegation under revalidation

Re-delegation:

the process of changing a delegation information to another set of authoritative name servers, potentially under different administrative control

2. Motivation

There is wide variability in the behavior of deployed DNS resolvers today with respect to how they process delegation records. Some of them prefer the parent NS set, some prefer the child, and for others, what they preferentially cache depends on the dynamic state of queries and responses they have processed [SOMMESE].

While this variability is expected to continue in deployed implementations, this document specifies an algorithm that can be adopted by resolvers to achieve several benefits.

It preferentially and predictably prefers the authoritative NS set at the apex of the child zone, which better comports with the the DNS protocol’s data ranking rules. (Note that the proposed redesign of the DNS delegation mechanism in [I-D.ietf-deleg] is expected to fundamentally alter where authoritative delegation information will reside. This document deals with the DNS delegation mechanism as currently deployed.)

It offers seeveral security advantages. The mechanisms described in this document help against GHOST domain attacks and a variety of cache poisoning attacks with resolvers that adhere to Sections 5.4.1 (Ranking data) and 6.1 (Zone authority) of [RFC2181]. Strictly revalidating referral and authoritative NS RRset responses, enables the resolver to defend itself against query redirection attacks, see Security and Privacy considerations (Section 9).

The delegation NS RRset at the bottom of the parent zone and the apex NS RRset in the child zone are unsynchronized in the DNS protocol. Section 4.2.2 of [RFC1034] says "The administrators of both zones should insure that the NS and glue RRs which mark both sides of the cut are consistent and remain so.". But for a variety of reasons they could not be [SOMMESE]. Officially, a child zone’s apex NS RRset is authoritative and thus has a higher cache credibility than the parent’s delegation NS RRset, which is non-authoritative (Sections 5.4.1 (Ranking data) and 6.1 (Zone authority) of [RFC2181]). Hence the NS RRset "below the zone cut" should immediately replace the parent’s delegating NS RRset in cache when an iterative caching DNS resolver crosses a zone boundary. However, this can only happen if (1) the resolver receives the authoritative NS RRset in the Authority section of a response from the child zone, which is not mandatory, or (2) if the resolver explicitly issues an NS RRset query to the child zone as part of its iterative resolution

algorithm. In the absence of this, it is possible for an iterative caching resolver to never learn the authoritative NS RRset for a zone, unless a downstream client of the resolver explicitly issues such an NS query, which is not something that normal end user applications do, and thus cannot be relied upon to occur with any regularity.

Increasingly, there is a trend towards minimizing unnecessary data in DNS responses. Several popular DNS implementations default to such a configuration (see "minimal-responses" in BIND and NSD). So, they may never include the authoritative NS RRset in the Authority section of their responses.

A common reason that zone owners want to ensure that resolvers place the authoritative NS RRset preferentially in their cache is that the TTLs may differ between the parent and child side of the zone cut. Some DNS Top Level Domains (TLDs) only support long fixed TTLs in their delegation NS sets. This inhibits a child zone owner’s ability to make more rapid changes to their name server configuration using a shorter TTL, if resolvers have no systematic mechanism to observe and cache the child NS RRset.

Similarly, a child zone owner may also choose to have longer TTLs in their delegation NS sets and address records to decrease the attack window for cache poisoning attacks. For example, at the time of writing, root-servers.net has a TTL of 6 weeks for the root server identifier address records, where the TTL in the priming response is 6 days.

A zone’s delegation still needs to be periodically revalidated at the parent to make sure that the parent zone has not legitimately re-delegated the zone to a different set of name servers, or even removed the delegation. Otherwise, resolvers that refresh the TTL of a child NS RRset on subsequent queries or due to pre-fetching, may cling to those name servers long after they have been re-delegated elsewhere. This leads to the second recommendation in this document, "Delegation Revalidation" - Resolvers should record the TTL of the parent’s delegating NS RRset, and use it to trigger a revalidation action. Attacks exploiting lack of this revalidation have been described in [GHOST1], [GHOST2].

3. Upgrading NS RRset Credibility

When a referral response is received during iteration, a validation query SHOULD be sent in parallel with the resolution of the triggering query, to one of the delegated name servers for the newly discovered zone cut. Note that DNSSEC validating resolvers today, when following a secure referral, already need to dispatch a query to one of the delegated name servers for the DNSKEY RRset, so this validation query could be sent in parallel with that DNSKEY query.

A validation query consists of a query for the child's apex NS RRset, sent to one of the newly discovered delegation's name servers. Normal iterative logic applies to the processing of responses to validation queries, including storing the results in cache, trying the next server on SERVFAIL or timeout, and so on. Positive responses to this validation query MAY be cached with an authoritative data ranking. Successive queries directed to the same zone SHOULD be directed to the nameservers listed in the child's apex, due to the ranking of this answer. If the validation query fails, the parent NS RRset SHOULD remain the one with the highest ranking and SHOULD be used for successive queries.

A response to the triggering query to the child may contain the NS RRset in the authority section as well. This NS RRset however has a lower trustworthiness than the set from the direct query (Section 5.4.1 of [RFC2181]), so regardless of the order in which the responses are received, the NS RRset from the answer section from the direct child's apex NS RRset query MAY be stored in the cache eventually.

When a resolver detects that the child's apex NS RRset contains different name servers than the non-authoritative version at the parent side of the zone cut, it MAY report the mismatch using DNS Error Reporting [RFC9567] on the Report-Channel for the child zone, as well as on the Report-Channel for the parent zone, with an extended DNS error code of TBD (See Section 8).

A No Data response (see Section 2.2 of [RFC2308]) for the validating NS query should be treated the same as a failed validating NS query. The parent NS RRset SHOULD remain the one with the highest ranking and SHOULD be used for successive queries. All resolution failures MUST be cached as directed in [RFC9520], to prevent aggressive requeries.

4. Limiting upgrading NS Credibility

Upgrading NS RRset Credibility (Section 3) works most reliable with good quality child NS RRsets, where the name servers referenced in the RDATA of the NS RRset do result in IP addresses which serve the child zone. We consider both the root, as well as the zones delegated from the root, to have good quality child NS RRset, but recognize that, because of the less transparently administered further delegations (among others by parties that are perhaps less devoted to DNS administration), some of those further delegations may be sub-optimal for upgrading NS RRset credibility.. An implementation MAY limit revalidation to delegations that cross administrative boundaries such as anywhere in ".ip6.arpa" and ".in-addr.arpa" as well as any so-called "public suffix" such as the root zone, top level zones such as ".com" or ".net", and effective top level zones such as ".ad.jp" or ".co.uk".

5. Upgrading A and AAAA RRset Credibility

5.1. Upgrading glue

Additional validation queries for the "glue" address RRs of referral responses (if not already authoritatively present in cache) SHOULD be sent with the validation query for the NS RRset as well. Positive responses SHOULD be cached with authoritative data ranking. The non-authoritative "glue" MAY be cached with non-authoritative data ranking for fallback purposes. Successive queries directed to the same zone SHOULD be directed to the authoritative nameservers denoted in the referral response.

The names from the NS RRset in a validating NS response may differ from the names from the NS RRset in the referral response. Outstanding validation queries for "glue" address RRs that do not match names in a newly discovered authoritative NS RRset may be discarded, or they may be left running to completion. Their result MUST no longer be used in queries for the zone. Outstanding validation queries for "glue" address RRs that do match names in the authoritative NS RRset MUST be left running to completion. They do not need to be re-queried after reception of the authoritative NS RRset (see Section 5).

Validated "glue" may result in unreachable destinations if they are obtained from poorly managed zones with incorrect address records. A resolver MAY choose to keep the non-authoritative value for the "glue" next to the preferred authoritative value for fallback purposes. Such a resolver MAY choose to fallback to use the non-authoritative value as a last resort, but SHOULD do so only if all other authoritative "glue" led to unreachable destinations as well.

5.2. Upgrading additional address from authoritative NS responses

Authoritative responses for a zone's NS RRset at the apex can contain nameserver addresses in the Additional section. An NS RRset validation response is an example of such a response. A priming response is another example of an authoritative zone's NS RRset response [RFC8109].

When additional addresses in authoritative NS RRset responses are DNSSEC verifiable (because the complete RRset is included, including a verifiable signature for the RRset) and DNSSEC secure, they MAY be cached authoritatively immediately without additional validation queries. DNSSEC validation is enough validation in those cases. Otherwise, the addresses cannot be assumed to be complete or even authoritatively present in the same zone, and additional validation queries SHOULD be made for these addresses.

Note that there may be outstanding address validation queries for the names of the authoritative NS RRset (from referral address validation queries). In those cases no new validation queries need to be made.

6. Strict and opportunistic revalidation

6.1. Strictly revalidating referrals and authoritative NS RRset responses

Resolvers may choose to delay the response to a triggering query until it can be verified that the answer came from a name server listening on an authoritatively acquired address for an authoritatively acquired name. This would offer the most trustworthy responses with the least risk for forgery or eavesdropping, however without fallback to lower ranked NS RRsets and addresses, there is no failure mitigation and a failed NS RRset validation query, due to a broken child NS RRset or to malfunctioning child zone's authoritative servers, will then lead to a hard failure to query the referred to child zone.

If the resolver chooses to delay the response, and there are no nameserver names in common between the child's apex NS RRset and the parent's delegation NS RRset, then any responses received from sending the triggering query to the parent's delegated nameservers SHOULD be discarded, and this query should be sent again to one of the child's apex nameservers.

It is RECOMMENDED, to scope *strict* upgrading of NS, A and AAAA RRset credibility, to the root zone and zones delegated from the root zone only (see Section 4).

6.2. Opportunistic revalidating referral and authoritative NS RRset responses

In practice, we expect many implementations may answer the triggering query in advance of the validation query for performance reasons. An additional reason is that there are unfortunately a number of nameservers in the field that (incorrectly) fail to properly answer explicit queries for zone apex NS records, and thus the revalidation logic may need to be applied lazily and opportunistically to deal with them. In cases where the delegated nameservers respond incorrectly to an NS query, the resolver SHOULD abandon this algorithm for the zone in question and fall back to using only the information from the parent’s referral response.

One may consider to only limit _strict_ upgrading of NS, A and AAAA RRset credibility to the root zone and zones delegated from the root zone, and perform _opportunistic_ revalidations for further delegations (see Section 4).

7. Delegation Revalidation

The essence of this mechanism is revalidation of all delegation metadata that directly or indirectly supports an owner name in cache. This requires a cache to remember the delegated name server names for each zone cut as received from the parent (delegating) zone’s name servers, and also the TTL of that NS RRset, and the TTL of the associated DS RRset (if seen).

A delegation under revalidation is called a "revalidation point" and is "still valid" if its parent zone’s servers still respond to an in-zone question with a referral to the revalidation point, and if that referral overlaps with the previously cached referral by at least one name server name, and the DS RRset (if seen) overlaps the previously cached DS RRset (if also seen) by at least one delegated signer.

If the response is not a referral or refers to a different zone than before, then the shape of the delegation hierarchy has changed. If the response is a referral to the revalidation point but to a wholly novel NS RRset or a wholly novel DS RRset, then the authority for that zone has changed. For clarity, this includes transitions between empty and non-empty DS RRsets.

If the shape of the delegation hierarchy or the authority for a zone has been found to change, then currently cached data whose owner names are at or below that revalidation point MUST NOT be used. Such non-use can be by directed garbage collection or lazy generational garbage collection or some other method befitting the architecture of the cache. What matters is that the cache behave as though this data was removed.

Since revalidation can discover changes in the shape of the delegation hierarchy it is more efficient to revalidate from the top (root) downward (to the owner name) since an upper level revalidation may obviate lower level revalidations. What matters is that the supporting chain of delegations from the root to the owner name be demonstrably valid; further specifics are implementation details.

Revalidation MUST be triggered when delegation meta-data has been cached for a period at most exceeding the delegating NS or DS (if seen) RRset TTL. If the corresponding child zone’s apex NS RRset TTL is smaller than the delegating NS RRset TTL, revalidation MUST happen at that interval instead. However, resolvers SHOULD impose a sensible minimum TTL floor they are willing to endure to avoid potential computational DoS attacks inflicted by zones with very short TTLs.

In normal operations this meta-data can be quickly revalidated with no further work. However, when re-delegation or take-down occurs, a revalidating cache SHOULD discover this within one delegation TTL period, allowing the rapid expulsion of old data from the cache.

8. IANA Considerations

IANA is requested to assign a value to the "Extended DNS Error Codes" registry [RFC8914].

INFO-CODE	Purpose	Reference
TBD	referral NS RRset mismatch	[this document]

Table 1

9. Security and Privacy Considerations

9.1. DNSSEC protection of infrastructure data

Referral response NS RRsets and glue, and the additional addresses from authoritative NS RRset responses (such as the root priming response), are not protected with DNSSEC signatures. An attacker that is able to alter the unsigned A and AAAA RRsets in the additional section of referral and authoritative NS RRset responses, can fool a resolver into taking addresses under the control of the attacker to be authoritative for the zone. Such an attacker can redirect all traffic to the zone (of the referral or authoritative NS RRset response) to a rogue name server.

A rogue name server can view all queries from the resolver to that zone and alter all unsigned parts of responses, such as the parent side NS RRsets and glue of further referral responses. Resolvers following referrals from a rogue name server, that do not revalidate those referral responses, can subsequently be fooled into taking addresses under the control of the attacker to be authoritative for those delegations as well. The higher up the DNS tree, the more impact such an attack has. An attacker controlling a rogue name server for the root has potentially complete control over the entire domain name space and can alter all unsigned parts undetected.

Strictly revalidating referral and authoritative NS RRset responses (see Section 6), enables the resolver to defend itself against the above described attack with DNSSEC signed infrastructure RRsets. Unlike cache poisoning defences that leverage increase entropy to protect the transaction, revalidation of NS RRsets and addresses also provides protection against on-path attacks.

Since December 6, 2023, the root zone contains a DNSSEC signed cryptographic message digest [RFC8976][ROOT-ZONEMD], covering all root zone data. This includes all non-authoritative data such as the A and AAAA RRsets for the IP addresses of the root server identifiers, as well as the NS RRsets and glue that make up the delegations. A root zone local to the resolver [RFC8806] with a verified and validated ZONEMD RRset, would provide protection similarly strong to strictly revalidating the root and the top level domains.

9.2. Cache poisoning protection

In [DNS-CACHE-INJECTIONS] an overview is given of 18 cache poisoning attacks of which 13 can be remedied with delegation revalidation. The paper provides recommendations for handling records in DNS responses with respect to an earlier version of the idea presented in this document [I-D.wijngaards-dnsext-resolver-side-mitigation].

Upgrading NS RRset Credibility (Section 3) allows resolvers to cache and utilize the authoritative child apex NS RRset in preference to the non-authoritative parent NS RRset. However, it is important to implement the steps described in Delegation Revalidation (Section 7) at the expiration of the parent's delegating TTL. Otherwise, the operator of a malicious child zone, originally delegated to, but subsequently delegated away from, can cause resolvers that refresh TTLs on subsequent NS set queries, or that pre-fetch NS queries, to never learn of the re-delegated zone [GHOST1], [GHOST2].

9.3. Other considerations

Some resolvers do not adhere to Sections 5.4.1 and 6.1 of [RFC2181], and only use the non-authoritative parent side NS RRsets and glue returned in referral responses for contacting authoritative name servers [I-D.fujiwara-dnsop-resolver-update]. As a consequence, they are not susceptible to many of the cache poisoning attacks enumerated in [DNS-CACHE-INJECTIONS] that are based upon the relative trustworthiness of DNS data. Such resolvers are also not susceptible to the GHOST domain attacks [GHOST1], [GHOST2]. Such resolvers will however never benefit from DNSSEC protection of infrastructure RRsets and are susceptible to query redirection attacks.

Revalidating referral and authoritative NS RRset responses will induce more traffic from the resolver to the authoritative name servers. The traffic increase may be substantial if the address RRsets for the names in the NS RRset's RDATA were provided non-authoritatively (as glue or as additional addresses) and need revalidation too [REDIRECTED-QUERY-TRAFFIC]. Resolvers SHOULD take care to limit the amount of work they are willing to do to resolve a query to a sensible amount.

10. References

10.1. 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>.

[RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, DOI 10.17487/RFC2181, July 1997, <https://www.rfc-editor.org/info/rfc2181>.

[RFC2308] Andrews, M., "Negative Caching of DNS Queries (DNS NCACHE)", RFC 2308, DOI 10.17487/RFC2308, March 1998, <https://www.rfc-editor.org/info/rfc2308>.

[RFC8109] Koch, P., Larson, M., and P. Hoffman, "Initializing a DNS Resolver with Priming Queries", 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>.

[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>.

[RFC8914] Kumari, W., Hunt, E., Arends, R., Hardaker, W., and D. Lawrence, "Extended DNS Errors", RFC 8914, DOI 10.17487/RFC8914, October 2020, <https://www.rfc-editor.org/info/rfc8914>.

[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>.

[RFC9520] Wessels, D., Carroll, W., and M. Thomas, "Negative Caching of DNS Resolution Failures", RFC 9520, DOI 10.17487/RFC9520, December 2023, <https://www.rfc-editor.org/info/rfc9520>.

[RFC9567] Arends, R. and M. Larson, "DNS Error Reporting", RFC 9567, DOI 10.17487/RFC9567, April 2024, <https://www.rfc-editor.org/info/rfc9567>.

10.2. Informative References

[DNS-CACHE-INJECTIONS]
Klein, A., Shulman, H., and M. Waidner, "Internet-Wide Study of DNS Cache Injections", n.d., <https://ieeexplore.ieee.org/abstract/document/8057202>.

[GHOST1] Jiang, J., Liang, J., Li, K., Li, J., Duan, H., and J. Wu, "Ghost Domain Names: Revoked Yet Still Resolvable", n.d., <https://www.ndss-symposium.org/ndss2012/>.

[GHOST2] Li, X., Liu, B., Bai, X., Zhang, M., Zhang, Q., Li, Z., Duan, H., and Q. Li, "Ghost Domain Reloaded: Vulnerable Links in Domain Name Delegation and Revocation", n.d., <https://www.ndss-symposium.org/ndss-paper/ghost-domain-reloaded-vulnerable-links-in-domain-name-delegation-and-revocation/>.

[I-D.fujiwara-dnsop-resolver-update] Fujiwara, K., "Updating Resolver Algorithm", Work in Progress, Internet-Draft, draft-fujiwara-dnsop-resolver-update-00, 31 October 2016, <https://datatracker.ietf.org/doc/html/draft-fujiwara-dnsop-resolver-update-00>.

[I-D.ietf-deleg] April, T., paek, P., Weber, R., and Lawrence, "Extensible Delegation for DNS", Work in Progress, Internet-Draft, draft-ietf-deleg-04, 16 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-deleg-04>.

[I-D.vixie-dnsext-resimprove] Vixie, P. A., Joffe, R., and F. Neves, "Improvements to DNS Resolvers for Resiliency, Robustness, and Responsiveness", Work in Progress, Internet-Draft, draft-vixie-dnsext-resimprove-00, 23 June 2010, <https://datatracker.ietf.org/doc/html/draft-vixie-dnsext-resimprove-00>.

[I-D.wijngaards-dnsext-resolver-side-mitigation] Wijngaards, W., "Resolver side mitigations", Work in Progress, Internet-Draft, draft-wijngaards-dnsext-resolver-side-mitigation-01, 24 February 2009, <https://datatracker.ietf.org/doc/html/draft-wijngaards-dnsext-resolver-side-mitigation-01>.

[REDIRECTED-QUERY-TRAFFIC] Toorop, W., Thessalonikefs, Y., Overeinder, B., Mller, M., and M. Davids, "The reduced risk of redirected query traffic with signed root name server data", n.d., <https://www.icann.org/en/system/files/files/reduced-risk-redirected-query-traffic-signed-root-name-server-data-22may24-en.pdf#h.8mh7wvmas7vi>.

[ROOT-ZONEMD]
Wessels, D., "Root zone operational announcement: introducing ZONEMD for the root zone", n.d., <https://lists.dns-oarc.net/pipermail/dns-operations/2023-December/022388.html>.

[SOMMESE] Sommese, R., Moura, G. C. M., Jonker, M., van Rijswijk-Deij, R., Dainotti, A., Claffy, K. C., and A. Sperotto, "When parents and children disagree: Diving into DNS delegation inconsistency", n.d., <https://par.nsf.gov/servlets/purl/10186683>.

Appendix A. Acknowledgements

Wouter Wijngaards proposed explicitly obtaining authoritative child NS data in [I-D.wijngaards-dnsext-resolver-side-mitigation]. This behavior has been implemented in the Unbound DNS resolver via the "harden-referral-path" option. The combination of child NS fetch and revalidating the delegation was originally proposed in [I-D.vixie-dnsext-resimprove], by Paul Vixie, Rodney Joffe, and Frederico Neves.

The authors would like to thank Ralph Dolmans who was an early collaborator on this work, as well as the many members of the IETF DNS Operations Working Group for helpful comments and discussion.

Appendix B. Implementation status

Note to the RFC Editor: please remove this entire appendix before publication.

* The Unbound resolver software has opportunistic revalidating of referral and authoritative NS RRset responses, as described in Section 3, Section 5 and Section 6.2 in this document, implemented since version 1.1 (released August 29, 2008). It is enabled with a configuration option harden-referral-path: yes which is disabled by default.

  "Redhat Enterprise Linux has been running Unbound with the harden-referral-path: option set to yes for years without problems", as mentioned by Paul Wouters during dnsop workgroup session at the IETF 119.

* The Knot Resolver software revalidates the priming response as part of priming the root zone since version 1.5.1 (released December 12, 2017)

* Section 7 has been implemented in the Unbound resolver since version 1.4.17 (released May 24, 2012).

Authors' Addresses

Shumon Huque
Salesforce
415 Mission Street, 3rd Floor
San Francisco, CA 94105
United States of America
Email: shuque@gmail.com

Paul Vixie
SIE Europe, U.G.
Email: paul@redbarn.org

Willem Toorop
NLnet Labs
Science Park 400
1098 XH Amsterdam
Netherlands
Email: willem@nlnetlabs.nl