Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: May 7, 2020

S. Huque
Salesforce
P. Vixie
Farsight Security

November 4, 2019

Delegation Revalidation by DNS Resolvers
draft-huque-dnsop-ns-revalidation-00

Abstract

This document recommends improved DNS [RFC1034] [RFC1035] resolver behavior with respect to the processing of Name Server (NS) resource record sets (RRset) during iterative resolution. 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. Resolvers should also periodically revalidate the child delegation by re-quering the parent zone at the expiration of the TTL of the parent side NS RRset.

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 May 7, 2020.

Copyright Notice

Table of Contents

1. Introduction

This document recommends improved DNS resolver behavior with respect to the processing of NS record sets during iterative resolution. The first recommendation is 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. The second recommendation is to revalidate the delegation by re-quering the parent zone at the expiration of the TTL of the parent side NS RRset.

2. Motivation

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. [RFC1034] Section 4.2.2 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. 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 glue ([RFC2181], Section 5.4.1. Ranking data). 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 enduser applications do.

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 Unbound).

Qname Minimisation [RFC7816], a more privacy preserving mode of iterative resolution, specifies the use of the NS query type at every step of the resolution process until the full query name has been reconstructed at the leaf zone. This would provide a way to definitively learn the child zone's authoritative NS RRset. In practice however, many (most?) implementations of Qname Minimisation currently employ the original query type or the A query type. Thus, regardless of whether Qname Minimisation is in use or not, this document recommends explicitly fetching the authoritative NS RRset at the child zone when following a referral.

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, and this inhibits the zone owner's ability to make more rapid changes to their nameserver configuration, if resolvers have no systematic mechanism to observe the child NS RRset.

A child 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 nameservers. Otherwise, resolvers that refresh the TTL of a child NS RRset on subsequent queries or due to pre-fetching, may cling to those nameservers long after they have been re-delegated elsewhere. This leads to the

second recommendation in this document, "Delegation Revalidation". Essentially, the resolver should record the TTL of the parent's delegating NS RRset, and use it to trigger a revalidation action. Technically, if both parent and child zone are DNSSEC [RFC4033] [RFC4034] [RFC4035] signed with a corresponding secure delegation between them, then expiration of the DS record will cause revalidation of the current child zone's DNSKEY set, so responses from the orphaned child nameservers would no longer be trusted. However, delegation revalidation is still necessary to locate the current nameserver addresses.

3. Upgrading NS RRset Credibility

- When a delegation response is received during iteration, a validation query should be sent in parallel with the forwarding of the triggering query to the delegated nameservers for the newly discovered zone cut. The response to the triggering query should be delayed until both the forwarded query and the validation query have been answered.

- A validation query consists of a query for the child's apex NS RRset, sent to the newly discovered delegation's nameservers. Normal iterative logic applies to the processing of responses to validation queries, including storing the results in cache, propagating NXDOMAIN back to the triggering query, trying the next server on SERVFAIL or timeout, and so on.

- If there are no nameserver names in common between the child's apex NS RRset and the parent's delegation NS RRset, then the responses received from forwarding the triggering query to the parent's delegated nameservers should be discarded after validation, and this query should be forwarded again to the child's apex nameservers.

- [TBD: There are a small but non trivial number of authoritative DNS services that timeout on explicit NS queries. Should we accommodate them by treating this practice lazily and opportunistically? Or do we expect the DNS Flag Day efforts will effectively banish them in the near future?]

4. Delegation Revalidation

- The lowest TTL found in a parent zone's delegating NS RRset should be stored in the cache and used to trigger delegation revalidation as follows: Whenever a cached RRset is being considered for use in a response, the cache should be walked upward toward the root, looking for expired delegations. At the first expired delegation encountered while walking upward toward the root, revalidation should be triggered, putting the processing of dependent queries on hold until validation is complete.

- To revalidate a delegation, the iterative caching DNS resolver will forward the query that triggered revalidation to the nameservers at the closest enclosing zone cut above the revalidation point. While searching for these nameservers, additional revalidations may occur, perhaps placing a chain of dependent queries on hold, unwinding in downward order as revalidations closer to the root must be complete before revalidations further from the root can begin.

- If a delegation can be revalidated at the same node, then the old apex NS RRset should be deleted from cache and then the new delegating NS RRset should be stored in cache. The minimum TTL from the new delegating NS RRset should also be stored in cache to facilitate future revalidations. This order of operations ensures that the RRset credibility rules do not prevent the new delegating NS RRset from entering the cache. It is expected that the child's apex NS RRset will rapidly replace the parent's delegating NS RRset as soon as iteration restarts after the revalidation event.

- If the new delegating NS RRset cannot be found (RCODE=NXDOMAIN) or if there is a new zone cut at some different level of the hierarchy (insertion or deletion of a delegation point above the revalidation point) or if the new RRset shares no nameserver names in common with the old one (indicating some kind of redelegation, which is rare) then the cache should be purged of all names and RRsets at or below the revalidation point. This facilitates redelegation or revocation of a zone by a parent zone administrator, and also conserves cache storage by deleting unreachable data.

- To make the timing of a revalidation event unpredictable from the

point of view of a potential cache-spoof attacker, the parent's delegating NS RRset TTL should be reduced by a random fraction of its value before being stored for use in revalidation activities.

- This section describes a precise algorithm for delegation revalidation. A simpler resolver implementation may choose alternative methods, e.g. it may choose to perform the upward search for expired delegations at scheduled intervals rather than for every response decision. Or it may have a fixed maximum TTL for child zones before they are expired from the cache and requeried at the parent.

5. Acknowledgements

The practices described in this document were originally proposed in [I-D.vixie-dnsext-resimprove], by Vixie, Joffe, and Neves.

6. IANA Considerations

This document includes no request to IANA.

7. Security Considerations

Upgrading NS RRset Credibility (Section 3) allows resolvers to cache and utilize the authoritative child apex NS RRset in preference to the non-authoriative parent NS RRset. However, it is very important

to implement the steps described in Delegation Revalidation (Section 4) 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 redelegated zone. This problem has been seen in the wild [include reference to Ghost Domains paper here].

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>.

[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>.

[RFC7816] Bortzmeyer, S., "DNS Query Name Minimisation to Improve Privacy", RFC 7816, DOI 10.17487/RFC7816, March 2016, <https://www.rfc-editor.org/info/rfc7816>.

8.2. Informative References

[I-D.vixie-dnsext-resimprove]
Vixie, P., Joffe, R., and F. Neves, "Improvements to DNS Resolvers for Resiliency, Robustness, and Responsiveness", draft-vixie-dnsext-resimprove-00 (work in progress), June 2010.

[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>.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <https://www.rfc-editor.org/info/rfc4034>.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <https://www.rfc-editor.org/info/rfc4035>.

Authors' Addresses

Shumon Huque

Salesforce

Email: shuque@gmail.com

Paul Vixie
Farsight Security

Email: paul@redbarn.org