Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 29, 2018

W. Mekking
Oracle Dyn
O. Gudmundsson
CloudFlare

March 28, 2018

Minimal Incremental Zone Transfer in DNS
draft-mekking-mixfr-02

Abstract

This document proposes extensions to the DNS protocol to provide an incremental zone transfer (IXFR) mechanism with dynamic update (UPDATE) capabilities, to keep IXFRs that deal with DNSSEC small.

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 http://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 September 29, 2018.

Table of Contents

1. Introduction

Incremental zone transfer (IXFR, [RFC1995]) was introduced to efficiently transfer changed portions of a zone. However, when a zone is signed with DNSSEC [RFC4033], [RFC4034], [RFC4035], the transfer can still become very large. For example, when many resource record sets (RRsets) need to be re-signed, or when the NSEC3 [RFC5155] salt is changed, an IXFR may become larger than a full zone transfer (AXFR, [RFC5936]). This is because the IXFR includes complete copies of both the deleted and replacement RRSIG records.

To keep the deltas small in zone transfers, we need to have a richer change syntax, for example like in Dynamic Update (DNS UPDATE, [RFC2136]. This document introduces a new query type MIXFR (minimal incremental zone transfer) that is able to express this richer syntax. The goal of this proposal is to allow small changes to be communicated over UDP, and remove as much redundant information from the zone transfer as possible.

An earlier proposal to keep the zone transfers small is IXFR-ONLY [IXFR-ONLY], by giving the client an oppurtunity to signal the server

that it prefers an error above a fall back to an AXFR in case the server is not able to send an IXFR. However IXFR-ONLY did not reduce the size of an IXFR.

2. Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Syntax

The syntax for MIXFR is a superset of IXFR. The richer syntax of MIXFR allows to add or delete multiple records with one resource record (RR). MIXFR is DNSSEC aware thus if there is a change to RRset it knows to delete the covering RRISG(s), this saves the transmission of old RRsigs.

3.1. Implicit RRSIG deletion

When an RRset is modified, the MIXFR client MUST also remove all existing RRSIG records on that RRset. This is valid for all RRtypes except RRSIG itself.

3.2. Add an RR

This works the same as with IXFR, with implicit RRSIG delete logic added.

3.3. Delete an RR

This works the same as with IXFR, with implicit RRSIG delete logic added.

3.4. Delete an RRset

Similar to DNS UPDATE. To delete an RRset, the MIXFR deletion list includes an RR whose NAME and TYPE are those of the RRset to be deleted. CLASS must be specified as ANY. RDLENGTH must be zero (0) and RDATA must therefore be empty. This also deletes the covering RRSIGs.

Note that a record with its CLASS set to ANY does _not_ mean to delete (or change) the record in all available classes: zone transfers are encapsulated in SOA records that determine the zone name and class (see Figure ()[#fig:a-MIXFR-response]). Only changes in the zone matching that name and class will be made.

3.5. Delete All RRsets on a Name

Similar to DNS UPDATE. To delete all RRSets at a name, the MIXFR deletion list includes an RR at that NAME, whose TYPE must be specified as ANY and CLASS must be specified as ANY. RDLENGTH must be zero (0) and RDATA must therefore be empty.

3.6. Replace an RRset

The MIXFR addition list includes an RR whose NAME and TYPE are those of the RRset to be replaced. CLASS must be specified as ANY. RDLENGTH must be non-zero and the RDATA is that of the first replacement record.

If an RRset is to be replaced with multiple records, the second and subsequent records MUST use the syntax for adding an RR.

The same syntax is used to delete an RRset and to replace an RRset with an RR whose RDLENGTH is zero. This is not ambiguous because the former appears in the deletion list (before the new SOA RR) and the latter appears in the addition list (after the new SOA RR).

4. Protocol Description

4.1. Client side

The client can send a MIXFR request. Just like with IXFR, it places a SOA RR in the authority section to signal the version of the zone it holds now. If the client does not want the server to fall back to AXFR, it MAY add another SOA RR in the additional section. This achieves MIXFR-only behavior, similar to IXFR-ONLY [IXFR-ONLY]. For example:

```
;; ->>HEADER<<- opcode: QUERY, rcode: NOERROR, id: 1337
;; flags: qr ; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
;; QUESTION SECTION:
;; example.    IN    MIXFR

;; AUTHORITY SECTION:
example.       IN    SOA    serial=1

;; ADDITIONAL SECTION:
example.       IN    SOA    serial=1
```

Figure 1: A MIXFR request for the "example." zone.

[MM] Adding a whole record is quite some overhead in bits while we only signal one bit of information: to fall back or not to fall back.

[OG] Can we use a bit from header or OPT record? Or can we just use "Class | 0x8000" to signal that?

4.2. Server side

A server receiving a minimal incremental zone transfer (MIXFR) request will reply with a MIXFR. A MIXFR looks exactly like an IXFR, except there may be zero or more of the new introduced syntax RRs that can add or delete more records. For the zone "example.", the following zone transfer can be sent that will replace all signatures in the zone with new signatures for the names "example.", "a.example.", "b.example." and "c.example.":

```
;; ->>HEADER<<- opcode: QUERY, rcode: NOERROR, id: 1337
;; flags: qr ; QUERY: 1, ANSWER: 9, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;; example.    IN    MIXFR

;; ANSWER SECTION:
example.       IN    SOA    serial=3
example.       IN    SOA    serial=1
example.       ANY   RRSIG
example.       IN    SOA    serial=3
example.       IN    RRSIG  rdata
a.example.     IN    RRSIG  rdata
b.example.     IN    RRSIG  rdata
c.example.     IN    RRSIG  rdata
example.       IN    SOA    serial=3
```

Figure 2: A MIXFR response for the "example." zone.

The server MAY reply with an IXFR or AXFR instead. If the server does not implement MIXFR it MUST return a response with NOTIMPL rcode. The client MUST fallback to request IXFR or AXFR.

4.3. Future zone transfer improvements

In many cases DNS servers have many zones in common, and there are many changes in the zones each hour, in this case having a long lived TCP connection or an out-of-band protocol where the primary server can push changes to the secondary.

The size of the zone transfer can be reduced even more if the syntax on the wire is changed, i.e. the RR wire format is abandoned. A different grammar may add operators, remove duplicate RRset owner names, and use standard compression algorithms.

These kind of improvements will require more drastic changes, and may be covered in a separate, future document.

5. IANA Considerations

IANA is requested to assign the OPCODE value [TBD] (decimal) for MIXFR, in sub-registry "DNS OpCodes" of registry "Domain Name System (DNS) Parameters".

6. Security Considerations

This document does not introduce additional security considerations. Or does it?

Should we explain what the security implications are, because descriptions from old RFC's are not good enough?

Any MIXFR transactions should use secure channels such as IPSEC or SSH tunnel, and use TSIG for authentication.

7. Acknowledgements

Johan Ihren, Tony Finch, Bob Harold.

8. References

8.1. Informative References

[IXFR-ONLY] Sury, O. and S. Kerr, "IXFR-ONLY to Prevent IXFR Fallback to AXFR", February 2010, <https://tools.ietf.org/html/draft-kerr-ixfr-only-01>.

[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>.

[RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, DOI 10.17487/RFC5155, March 2008, <https://www.rfc-editor.org/info/rfc5155>.

[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>.

8.2. Normative References

[RFC1995] Ohta, M., "Incremental Zone Transfer in DNS", RFC 1995, DOI 10.17487/RFC1995, August 1996, <https://www.rfc-editor.org/info/rfc1995>.

[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>.

[RFC2136] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, DOI 10.17487/RFC2136, April 1997, <https://www.rfc-editor.org/info/rfc2136>.

Appendix A. Changelog

A.1. Version 02

- Removed 'Delete All RRsets of a Type' because it had the same syntax as 'Delete an RRset' [Olafur].
- Clarify ANY CLASS [#5, Bob Harold].
- Sleep for 3 years.
- Remove IXFR Gone Wild section.

A.2. Version 01

- Split document in trivial and 'more wild' ideas.

A.3. Version 00

- Initial version

Authors' Addresses

W. (Matthijs) Mekking
Oracle Dyn
Hertogswetering 163-167
Utrecht 3543 AS Utrecht
NL

EMail: matthijs.mekking@oracle.com
URI: https://www.dyn.com

Olafur Gudmundsson
CloudFlare
San Francisco, CA 94107
USA

EMail: olafur@cloudflare.com