An IXFR Fallback to AXFR Case
draft-song-dnsop-ixfr-fallback-00

Abstract

This memo introduces an IXFR issue observed during a multiple signers experiment conducted in Yeti DNS project. In the experiment IXFR client is designed to pull the zone from three IXFR servers who used their own key to sign the zone and produce different RRSIG records intentionally. The configuration of multiple signers cause the failure of IXFR in client side.

REMOVE BEFORE PUBLICATION: The source of the document is currently placed at GitHub [xml-file]. Comments and pull request are welcome.

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 November 17, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect
1. Introduction

In DNS specifications authoritative name server uses full zone transfer (AXFR) [RFC5936], incremental Zone Transfer (IXFR)[RFC1995], and NOTIFY [RFC1996] to achieve coherency of the zone contents. IXFR is an optimization for large DNS zone transfer, which allows server only transfer the changed portion(s) to client.

AXFR fallback usually happens at server side by simply returning IXFR client the entire new zone in condition that IXFR server cannot fulfill the given delta-update request. It is because an IXFR client may has multiple IXFR servers for a single zone. It is not a protocol defect but do stimulate people to find optimization avoiding full zone transfer [I-D.kerr-ixfr-only] and trying to make a new IXFR protocol [I-D.ietf-dnsext-rfc1995bis-ixfr].

[RFC5936] suggests that if its upstream servers have different ideas of the zone contents with the same (zone, serial) pair, the client can stop adding records that already exist or deleting records that do not exist. However, if an IXFR incoherence error is spotted by that client, it is not clear whether the client should stop IXFR process and ask for AXFR as a fallback. To the author's knowledge, there is such recommendation so far for AXFR fallback initiated by client in formal document.

This memo introduces an IXFR problem observed during a DNS root experiment in Yeti project[Yeti-DNS-Project] which involves multiple root zone distribution master (DM). It is designed that three DMs do have different "ideas" on how their keys are managed and produce different RRSIG records. In this scenario, it is observed that different DNS implementations have different behaviors due to the ambiguity in understanding of IXFR and fallback to AXFR.
REMOVE BEFORE Publication: The motivation of this memo is to ask for discussion in the community whether this specific fallback or non-fallback is viewed as a problem. Is it worthwhile developing IXFR protocol further towards rfc1996bis. Or this memo can serve providing some information and guidance to operators who do run authoritative servers in similar situation as it is done in Yeti experiment.

2. The IXFR issues observed in MZSK experiment

As a background for this memo, the introduction of Yeti testbed and experiments can be found in [I-D.song-yeti-testbed-experience], and section 3.1 and 4.2 are relevant. Conceptually, Yeti DNS intends to break one signer/DM role into three which buys some properties in loosely cooperative environment, such as resilience to single point of failure, more independent choice for slave server, and certain degree of transparency and management coordination for important zone (root zone in Yeti’s case).

One experiment in Yeti is designed to test multiple signers with Multiple ZSKs (MZSK). It is required that all public ZSKs used by DMs are included in the zone as a key set; and resolver can validate the message by picking one key from the key set. From DNSSEC point of view, it is technically workable. However, different signers do produce different RRSIG RR which introduces zone inconsistency from beginning in this case. In current setting of Yeti experiment, it is possible that one client does AXFR/IXFR from one server and later asks for IXFR from another server.

It is observed that when the IXFR client switched from one IXFR server to another, it received a IXFR response deleting RRSIG record that does not exist. One IXFR client running NSD 4.1.7 rejected IXFR response, made a log indicating a bad data and then asked for full zone transfer. Luckily, Yeti root zone is relatively small (691K), so the fallback to AXFR does not cause significant performance degeneration. But if operator does host big zone with MZSK model, it will cause problem based on current IXFR.

Another observation is that another IXFR client running Knot 2.1.0 in similar situation just accepts the IXFR response, ignores the differences and generates a merged zone with two RRSIG RRs. It not only produces larger response, but also causes DNSSEC failure when a new zone is generated given that old RRSIG is the signature of old zone RRs.
3. Possible solution

Generally, there are three considerations to this issue.

- Asking for development of RRSIG-aware IXFR format in which the RRSIG is treated as a special and RRSIG RR should always be transferred in full (like it does in AXFR). In the case of MZSK experiment, the old RRSIG record(s) is replaced by the new RRSIG record(s) and no specific deleted RRSIG is sent. Compared to the first case in NSD 4.1.7, it is helpful to reduce the cost for full zone transfer if the zone is fairly large.

- Adopting the behavior of NSD 4.1.7 as a improvement for IXFR protocol in which an IXFR client should fall back to AXFR automatically in the event of an IXFR incoherence error. To avoid unnecessary full zone transfer, it is desirable that the IXFR client is more "sticky" to the server who transfers the zone to the client last time.

- Without modification of IXFR protocol, asking each IXFR client (root slave in Yeti case) to be tied to a specific IXFR server(Yeti DM). This would decrease redundancy and resiliency but would allow normal IXFR, and may make debugging easier. But the MZSK operation introduce regional DM which is not desirable for Yeti case.

4. Acknowledgments

Specially thanks to Stephane Bortzmeyer who first spotted the IXFR issues from his NSD and Knot resolvers. Acknowledgment to Paul Vixie and Shane Kerr who contributed a lot to this technical finding and possible solutions in this memo. Thanks to Antonio Prado who helped to make the language more readable.

5. References

[I-D.ietf-dnsext-rfc1995bis-ixfr]

[I-D.kerr-ixfr-only]
[I-D.song-yeti-testbed-experience]
Song, D., Kerr, S., and D. Liu, "Experiences from Root Testbed in the Yeti DNS Project", draft-song-yeti-testbed-experience-01 (work in progress), December 2015.


Author’s Address

Linjian Song
Beijing Internet Institute
2508 Room, 25th Floor, Tower A, Time Fortune
Beijing 100028
P. R. China

Email: songlinjian@gmail.com
URI: http://www.biigroup.com/