

Moving from home grown to open source

A thrilling tale of RFC non-compliance, wildcard hell and scaling issues

Robin Geuze

2023-02-04

How it started for me

- ▶ TransDNS, the homebrew DNS authoritative software, was originally written in 2003-2004 and extended to support DNSSEC in 2012.
- ▶ I started working at TransIP in 2013 as a PHP (backend) developer.
- ▶ After a little while I was asked to take a look at TransDNS because it was crashing and I knew (some) C/C++.
- ▶ Figured out what was wrong, made a really quick fix, since the proper fix would take too much, and that was all the time I spend on it for a while .
- ▶ DISCLAIMER: While I was involved in most of the work described hereafter there are some things mentioned which I merely advised on rather than actively took part in. I'll try to be clear on that.

Starting situation

- ▶ Very basic setup, 3 servers, all running TransDNS. No loadbalancing
- ▶ Signing stack using dnssec-tools (1.x) and a lot of automation in PHP.
- ▶ DNS propagation through various cron's, which was really slow.
- ▶ Roughly 1M zones most of which are DNSSEC signed.
- ▶ Very few people had the knowledge to work on it.
- ▶ RFC compatibility was lacking.
- ▶ Adding new record types was a lot of work.
- ▶ Initial bug was fixed, but there were a bunch of other ones still in there

Initial steps

- ▶ Deploying new TransDNS versions was problematic due to long startup time. During the startup queries would get dropped.
- ▶ So we implemented load balancing. We were running FreeBSD so our first attempt was using relayd.
- ▶ We had lots of weird, hard to debug issues with relayd, so we eventually switched to haproxy for TCP and a homebrew forwarder for UDP.
- ▶ Worked quite well and allowed us to iterate on TransDNS a bit more, fixing some glaring issues like a lack of bounds checking and various ednscomp issues.
- ▶ Eventually we switched to dnsmdist to prevent having to maintain two pieces of software and allow for more flexibility.
- ▶ In the meantime we improved the TCP stack from a thread per connection model to a polling model based on kqueue to deal with high influx TCP connections.

RFC compatibility (1)

- ▶ Thanks to SIDN's validation monitoring we kept getting reminded about certain glaring RFC violations.
- ▶ The most important ones actually had the same case and we resolved them in one set of fixes.
- ▶ The first issue was incorrect handling of wildcards.
- ▶ It would still return results for a *.c wildcard for a.b.c if b.c also exists. According to the RFC's this should be an NXDOMAIN.
- ▶ The second issue was incorrect handling of Empty Non Terminals (ENT's).
- ▶ If a record exists at a.b.c this means b.c also exists. TransDNS however would return NXDOMAIN for queries of b.c rather than NODATA.

RFC compatibility (2)

- ▶ Difficult to detect what would be broken if behaviour was fixed.
- ▶ For DNSSEC enabled domains it would already be broken on DNSSEC resolvers, primarily 8.8.8.8 at that point.
- ▶ Decided to fix it in two steps, fix the behaviour for +do queries first, then for all queries.
- ▶ In between we captured a large quantity of queries, roughly 2 days, and compared the dnssec versus non-dnssec results to find any problem domains and contacted a few dozen affected customers.
- ▶ One other smaller issue was that the NSEC implementation in TransDNS was broken. This was fixed by a complete rewrite of the NSEC code.

Moving to PowerDNS

- ▶ SIDN announced that domains using DNSKEY algorithm 7 would no longer receive the DNSSEC incentive.
- ▶ At this point we decided to bite the bullet and overhaul our setup,
- ▶ We decided to build a new setup based on PowerDNS to fix various other problems as well.
- ▶ We first needed to pick a PowerDNS Backend
- ▶ However, TransDNS is fully memory based, leading to very fast response times and high throughput
- ▶ PowerDNS SQL backend was too slow
- ▶ PowerDNS bind backend didn't support the API

LMDB backend

- ▶ PowerDNS Imdb backend seemed like a good solution.
- ▶ It is fast, and it has support for the API.
- ▶ It did have the problem that record size could not be larger than 512 bytes.
- ▶ This was caused by the way the backend stored data in LMDB itself
- ▶ In the end I decided to fix the Imdb backend.
- ▶ PowerDNS 4.4.0 featured a patch to fix the record size limitation, and we moved forward based on the LMDB backend after that.

Issues when moving

- ▶ Once we started moving domains over we started running into weird issues however.
- ▶ Every thursday we saw a huge bump in the AXFR queue and all our updates took hours to propagate
- ▶ We figured out it was caused by DNSSEC signature renewal in PowerDNS
- ▶ After some discussion with the PowerDNS folks I came up with a solution, "AXFR priority levels"
- ▶ Basically user initiated AXFR's get more priority than NOTIFY's, SOA refresh or signature refresh
- ▶ Patch was included in the PowerDNS auth 4.5.0, and once rolled out we still saw huge AXFR queues, but zones updates would still propagate swiftly

Assorted issues

- ▶ We ran into some other issues which were either minor, solved in the loadbalancer layer, or just fixed in PowerDNS updates
- ▶ PowerDNS TCP performance was somewhat lackluster due to the one thread per client model
- ▶ Various small bugs in the LMDB backend were encountered due to the relatively new code
- ▶ One CVE was discovered in PowerDNS within a day of rolling out a new version (CVE-2021-36754)
- ▶ We saw some query patterns that seemed to be specifically designed to overload PowerDNS and did not affect TranDNS before. Resolved this by adding detection and mitigation at the load balancing layer.

Closing thoughts

- ▶ Migrating a homebrew setup can be a lot of work, especially if there is a long history
- ▶ However, it is worth it because it yields more flexibility and generally improves the quality of the setup going forward
- ▶ Rolling out new record types and features is now much easier, and the setup is even build in such a way that the secondaries could also run on other authoritative software
- ▶ Doing the migration in easy to control steps keeps things manageable and keeps the project from feeling to daunting.
- ▶ DNSSEC incentives work when trying to ensure implementation quality if you enforce that.

Question?

References

- ▶ TransDNS source code: <https://github.com/transip/transdns>
- ▶ LMDB record size change:
<https://github.com/PowerDNS/pdns/pull/9389>
- ▶ AXFR priority queue change:
<https://github.com/PowerDNS/pdns/pull/10196>