DNS hackathon 2025 in Stockholm

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

First publication of this article on 23 March 2025

https://www.bortzmeyer.org/hackathon-dns-2025.html

On 15 and 16 March 2025, I was at the DNS hackathon 2025 <https://labs.ripe.net/author/becha/join-the-dns-hackathon-2025/> in Stockholm, organised by RIPE NCC, DNS-OARC <https://www.dns-oarc.net/> and Netnod. I worked on a mechanism to synchronize the caches of DNS resolvers.

Hackathons are meant to be a collective work. After all, if you just code alone, you can as well stay at home/office. The organisers insist that you make big groups, with people of various profiles. (Speaking of diversity, there was apparently two women for more than thirty participants, which is typical for hackathons.) The subject I championed, implementation and interoperability of DELEG <https://www.afnic.fr/en/observatory-and-resources/expert-papers/dns-delegation-and-its-possible-evolu>, did not raise sufficient interest so I went to another project, Poisonlicious <https://github.com/DNS-Hackathon/Poisonlicious>. The idea for this project came from Quad9, a big public DNS resolver. Their network of resolvers is made of many sites, each with several physical machines, each physical machines hosting several virtual machines. When a DNS client asks the resolution of a domain name, each of the virtual machines has to do it on its own, without sharing work with the others, even if they were very close. The idea is therefore to synchronize the caches : when a machine finishes the resolution work, it sends a copy of the responses it got to other machines.

The practical work included :

— Patching the Unbound resolver to be able to both send and receive these data,
— Writing an Internet draft to document the idea, and the work to do (I worked mostly on this task).

In the current iteration of the Internet-Draft, the data is sent as an ordinary DNS message in UDP, authenticated by TSIG (otherwise, poisoning other machines with bad data is a risk). In the future, techniques like MQTT may be used for efficient synchronization.

The work done by Willem Toorop on Unbound is in this pull request <https://github.com/NLnetLabs/unbound/pull/1250> (it required to add TSIG support in Unbound, which did not need it before). The Internet draft is `draft-bortzmeyer-dnsop-poisonlicious`. It will be discussed in the dnsop IETF working group <https://datatracker.ietf.org/wg/dnsop/>. I also developed a small program in Python, using the excellent dnspython <https://www.dnspython.org/> library to resolve a domain name and send it, following the protocol, to the receiving machine : . Reading its source code gives you a good idea about how the mechanism works. You can also get a pcap of the packet sent : (the command was `python poisonlicious.py www.afnic.fr`). But nothing extraordinary, it is an ordinary DNS packet, with the TSIG signature.

There were other interesting projects during this hackathon :

— resviz <https://github.com/DNS-Hackathon/resviz> (DNS resolution visualizer). The goal is to be pedagogical (how resolvers work by showing the authoritative servers queried).
— Babies <https://github.com/DNS-Hackathon/Stork-DNS-Zone-Viewer> (from the Stork project <https://stork.isc.org/>, I let you find the reason for the name Babies). Monitoring / controlling multiple DNS servers from different vendors by proxying to their different APIs. NSD proved to be a bit tough during the hackathon. This project published its own report in a detailed article <https://www.isc.org/blogs/2025-dns-hackathon/>.
— LHB <https://github.com/DNS-Hackathon/LHB> ("Lookup Hostname Best practices") addressed the problem with getaddrinfo. Basically, the function `getaddrinfo` is available everywhere but very limited (no other types than the IP addresses, no information about whether the resolution was validated, for instance with DNSSEC, etc). Daniel Stenberg was not at the hackathon but was often quoted since he wrote a lot about getaddrinfo issues <https://daniel.haxx.se/blog/2024/06/06/bye-bye-hosting-c-ares-web/>. Also, there was a great talk at the last FOSDEM on this : "getaddrinfo sucks, everything else is much worse <https://fosdem.org/2025/schedule/event/fosdem-2025-4229-getaddrinfo-sucks-everything-else>". The hackathon project added some code in Ladybird.
— Canned DNS <https://gitlab.com/canneddns/canneddns> was about breaking the DNS on purpose, in order to exercise testing tools like Zonemaster <https://zonemaster.fr/>. They created a DNS server giving bad answers (two SOA, discrepancy between section count and the actual section, NXDOMAIN with data, etc). Unlike IBDNS <https://www.afnic.fr/en/observatory-and-resources/news/afnic-launches-ibdns-the-intentionally-broken-dns>, it will be specific to a test program. It is written in Go (not everyone subscribed yet to the Rust cult). I specially appreciated the fact that responses are not hardwired in the code but configured in TOML, which allows you to configure responses at will. Also, this project got the prize "most complete project".
— idIOT <https://github.com/DNS-Hackathon/SupportingDocs/blob/main/IdiOT.pdf>: DNS for Internet of stuff that spies on you. Current implementations of DNS on things are often broken. They do not respect TTL, they hardwired resolver IP addresses, etc. idIOT documented that. This hackathon project got the prize for "best project name".
— DohoT or Donion <https://github.com/DNS-Hackathon/DoHot-or-Donion>. An ordinary resolver knows both the IP address of its client and the question asked, which is bad for privacy. The idea is to add an intermediary (client [Caractère Unicode non montré [1]] proxy [Caractère Unicode non montré] resolver, with encryption), the intermediary will know the client address but not the question and the resolver will only know the question. If they don't collude, it is safe. Tor is cool but too slow (long circuits, may be with high latency, which is really bad for the DNS). Oblivious DNS (RFC 9230 [2]) may be the solution but is a new protocol. Six solutions were tested during the hackathon, ordinary DoH is the fastest (of course), improved Tor with shorter directed circuits may be the best solution.

Thanks to Vesna Manojlovi[Caractère Unicode non montré] who convinced me to come, to Johanna Eriksson and Denesh Bhabuta for the organisation, and to my nice project group, Willem Toorop, Babak Farrokhi and Moin Rahman.

1. Car trop difficile à faire afficher par LaTeX
2. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc9230.txt`