IETF 116 hackathon: unilateral probing of TLS on DNS servers

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

First publication of this article on 27 March 2023

https://www.bortzmeyer.org/hackathon-ietf-116.html

On 25 and 26 March, it was the IETF 116 hackathon <https://www.ietf.org/how/runningcode/hackathons/116-hackathon/> in Yokohama. I worked on TLS probing of authoritative DNS servers.

The goal of the IETF hackathons is to test ideas that may become RFC later. "Running code" is an important part of IETF principles. It is both useful and pleasing to sit down together at big tables <https://social.secret-wg.org/@benno/110083188511633914> and to hack all weekend, testing many ideas and seeing what works and what does not.

A bit of context on this "TLS probing". The DNS is a critical infrastructure of the Internet. It is used for most activities on the Internet. Requests and responses were traditionally sent in the clear, and this is bad for privacy since they can be quite revealing ("this user queried www.aa.org", see RFC 7626 [1]). To improve DNS privacy, the IETF developed two sets of solutions : **minimizing** data (RFC 9156), to protect against the server you query, and **encrypting** data, to protect against third parties sniffing on the cable. For the second set, encryption, several techniques have been developed, I will focus here on DoT (DNS over TLS, RFC 7858) and DoQ (DNS over QUIC, RFC 9250) but mostly on DoT. These various techniques are standardized for the client-to-resolver link, the most critical for the privacy (see RFC 7626 for details). They are now deployed, although not universally. But this leaves the resolver-to-authoritative link in the clear. The current idea is therefore to use encryption for this link, what we call ADoT (authoritative DNS over TLS). Technically, it is the same DNS over TLS than for the client-to-resolver link. But in practice, there are some issues. A client talks to only one (or may be two) resolvers, which he or she knows. But a resolver talks to thousands of authoritative name servers. Some will have enabled ADoT, some not. How to know ?

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc7626.txt

There were some attempts in the DPRIVE working group <https://datatracker.ietf.org/wg/dprive/> to define a **signaling** mechanism, through which the operators of authoritative name servers could say they support ADoT. But these efforts all fail. A general problem of signalisation of services on the Internet is that the signal can be wrong, and often is (advertising a service which actually des not work). So, the approach on the draft we worked on, draft-ietf-dprive-unilateral-probing (later published as RFC 9539), was different : probing the authoritative name server to see if it really does ADoT.

The work was done by Yorgos Thessalonikefs (from NLnet Labs <https://www.nlnetlabs.nl/>) and myself. Yorgos worked on adding probing support to Unbound and I tested existing ADoT deployment, as well as adding DoT support to the Drink authoritative name server <https://framagit.org/bortzmeyer/drink/>.

First, let's see DoT in action, here with a local installation of Drink and the kdig client (part of Knot) :

```
% kdig +tls @localhost -p 8353 foobar.test
;; TLS session (TLS1.3)-(ECDHE-X25519)-(ECDSA-SECP256R1-SHA256)-(AES-256-GCM)
;; ->>HEADER<<- opcode: QUERY; status: NOERROR; id: 16994
;; Flags: qr aa rd; QUERY: 1; ANSWER: 0; AUTHORITY: 1; ADDITIONAL: 1

;; EDNS PSEUDOSECTION:
;; Version: 0; flags: ; UDP size: 1440 B; ext-rcode: NOERROR
;; Option (65023): 067265706F7274076578616D706C6503636F6D00
;; PADDING: 348 B

;; QUESTION SECTION:
;; foobar.test.         IN A

;; AUTHORITY SECTION:
foobar.test.         10 IN SOA ns1.test. root.invalid. 2023032512 1800 300 604800 86400
;; Received 468 B
;; Time 2023-03-25 21:43:26 JST
;; From 127.0.0.1@8353(TCP) in 44.7 ms
```

OK, it worked, a TLS connection was established (the TLS cryptographic algorithms used are displayed at the beginning of the response), and we got a DNS reply. Also, the answer was padded (RFC 7830) to 468 bytes (RFC 8467) in order to avoid traffic analysis.

ADoT can also be tested with the check-soa <https://framagit.org/bortzmeyer/check-soa> tool. Here, all the authoritative name servers of the domain have ADoT :

```
% check-soa -dot aaa-cool.fr
ns01.one.com.
195.206.121.10: OK: 2023032101
2001:67c:28cc::10: OK: 2023032101
ns02.one.com.
185.10.11.10: OK: 2023032101
2001:67c:28cc:1::10: OK: 2023032101
```

And here only one has :

```
% check-soa -dot bortzmeyer.fr
ns2.bortzmeyer.org.
2400:8902::f03c:91ff:fe69:60d3: ERROR: dial tcp [2400:8902::f03c:91ff:fe69:60d3]:853: connect: connection refuse
172.104.125.43: ERROR: dial tcp 172.104.125.43:853: connect: connection refused
ns4.bortzmeyer.org.
2001:4b98:dc0:41:216:3eff:fe27:3d3f: ERROR: dial tcp [2001:4b98:dc0:41:216:3eff:fe27:3d3f]:853: i/o timeout
92.243.4.211: ERROR: dial tcp 92.243.4.211:853: i/o timeout
puck.nether.net.
2001:418:3f4::5: OK: 2023032600
204.42.254.5: OK: 2023032600
```

ADoT can also be tested with the RIPE Atlas probes <https://atlas.ripe.net/>, here with the Blaeu <https://labs.ripe.net/author/stephane_bortzmeyer/creating-ripe-atlas-one-off-measure> tool against a root name server, B.root-servers.net :

```
% blaeu-resolve --requested 100 --nameserver 199.9.14.201 --type SOA --tls --nsid .
Nameserver 199.9.14.201
[TIMEOUT] : 2 occurrences
[NSID: b4-ams; a.root-servers.net. nstld.verisign-grs.com. 2023032501 1800 900 604800 86400] : 54 occurrences
[NSID: b4-lax; a.root-servers.net. nstld.verisign-grs.com. 2023032501 1800 900 604800 86400] : 12 occurrences
[NSID: b4-iad; a.root-servers.net. nstld.verisign-grs.com. 2023032501 1800 900 604800 86400] : 15 occurrences
[NSID: b4-mia; a.root-servers.net. nstld.verisign-grs.com. 2023032501 1800 900 604800 86400] : 3 occurrences
[a.root-servers.net. nstld.verisign-grs.com. 2023032501 1800 900 604800 86400] : 1 occurrences
[NSID: b4-sin; a.root-servers.net. nstld.verisign-grs.com. 2023032501 1800 900 604800 86400] : 1 occurrences
Test #51316045 done at 2023-03-26T02:05:11Z
```

This is quite good, only two probes fail to connect over DoT, may be because they were on a network which blocks port 853 outgoing.

OK, so, some authoritative name servers have ADoT and some have not. Now, what should the resolver do ? Since there is no signaling, should it try port 853 ? How to do it without waiting forever if the server does not reply at all ? The idea behind the draft draft-ietf-dprive-unilateral-probing is indeed to always try ADoT and use it if it works **but** in an "intelligent" way. Instead of waiting for ADoT to succeed or fail, the resolver should do it on the "happy eyeballs" way (RFC 8305), trying both DoT and traditional DNS a the same time. Note that it is an unilateral choice : the resolver does all of the work and requires nothing from the authoritative server.

One resolver today does probing, although not in the way of the draft : PowerDNS Recursor <https://www.powerdns.com/recursor.html>. Let's try it (their DoT support is dscribed online <https://blog.powerdns.com/2022/06/13/probing-dot-support-of-authoritative-servers-just-try-it/>). We get the git, we follow the good compilation instructions <https://doc.powerdns.com/recursor/appendices/compiling.html>, we autoreconf, then ./configure, then make. We then can run it with the option --max-busy-dot-probes=100 (by default, it does not do probing). Note that probing in PowerDNS is lazy : it is not always tried, only of there is a "sufficient" traffic with the authoritative name server. You can see probing going on with tcpdump on port 853 and PowerDNS can report the results :

```
% rec_control dump-dot-probe-map /tmp/foo && fgrep Good /tmp/foo
dump-dot-probe-map: dumped 13962 records
45.54.76.1 qtmlabs.xyz. 14 Good 2023-03-29T04:27:32
157.53.224.1 qtmlabs.xyz. 39 Good 2023-03-29T04:28:37
77.55.125.10 abstractionleak.com. 1 Good 2023-03-29T04:25:39
77.55.126.10 abc-sport.fr. 3 Good 2023-03-29T04:25:39
185.10.11.11 one.com. 4 Good 2023-03-29T04:28:02
195.206.121.11 one.com. 4 Good 2023-03-29T04:28:03
```

```
51.77.156.11 BILLAUD.EU.ORG. 1 Good 2023-03-29T00:09:32
129.134.30.12 facebook.com. 5 Good 2023-03-29T04:29:26
129.134.31.12 facebook.com. 6 Good 2023-03-29T04:27:39
185.89.218.12 facebook.com. 14 Good 2023-03-29T04:27:39
185.89.219.12 facebook.com. 5 Good 2023-03-29T04:27:39
139.99.155.59 EU.ORG. 1 Good 2023-03-29T04:27:32
185.49.140.60 nlnetlabs.nl. 3 Good 2023-03-29T04:26:37
172.105.50.79 geekyboo.net. 1 Good 2023-03-29T04:22:37
23.234.235.93 obo.sh. 1 Good 2023-03-29T03:34:12
212.102.32.200 cdn77.org. 1 Good 2023-03-29T03:47:44
199.9.14.201 . 17 Good 2023-03-29T04:30:39
136.243.57.208 distos.org. 1 Good 2023-03-29T03:44:00
```

We see here that all these nameservers have a working ADoT. In order to probe many domains, I have fed the resolver with data from the .fr zone <http://opendata.afnic.fr/>, Cloudflare Domain Rankings <https://radar.cloudflare.com/domains> and a list of fediverse instances <http://demo.fedilist.com/instance>. A partial list of domains which currently have working ADoT for at least one name server is :

```
.               # Yes, the root
facebook.com
nlnetlabs.nl
powerdns.com
eu.org
billaud.eu.org
dyn.sources.org  # Drink name server
bortzmeyer.fr
nether.net
one.com # and the zones is hosts such as aaa-cool.fr
desec.io # and the zones it hosts such as qtmlabs.xyz
nazwa.pl # and the zones it hosts such as abc-sport.fr
psyopwar.com # and the zones it hosts such as obo.sh
cdn77.org
distos.org
```

As you can see, even the root has a working ADoT server (see the announcement <https://b.root-servers.org/news/2023/02/28/tls.html>), despite a prudent statement <https://root-servers.org/media/news/Statement_on_DNS_Encryption.pdf> of the root operators some time ago.

What about the work on Unbound ? Unfortunately, it was more complicated than it seemed, since "happy eyeballs" probing (sending the request twice) runs against some assumptions in Unbound code. But the work by Yorgos made possible to better understand the issues, which is an important point for the hackathon.

The work of this weekend raised several questions, to be carried to the working group <https://datatracker.ietf.org/wg/dprive/> :

— If the ADoT server replies but the reply indicates an error, such as SERVFAIL or REFUSED, should the resolver retry without DoT ? PowerDNS Recursor does it, but it seems it would make more sense to accept the reply, and just to remind system administrators that port 853 and 53 should deliver consistent answers.
— What should be the criteria to select an authoritative name server to query ? Should we prefer a fast insecure server or a slow encrypted one ? The draft does not mention it, because it is local policy. (PowerDNS Recursor has apparently no way to change its default policy, which is to use the fastest one, DoT or not.)

— Should we do lazy probing, like PowerDNS Recursor does, or use the more eager "happy eyeballs" algorithm ?

For the issue of consistency between port 53 and 853, this can go at odds with current practices. For instance, the only authoritative name server of `dyn.bortzmeyer.fr` hosts two different services, Drink for the autoritative service on port 53 and an open resolver, dnsdist <https://dnsdist.org/>, on port 853. So, the answers are different. Should we ban this practice ?

```
% check-soa -dot dyn.bortzmeyer.fr
ns1-dyn.bortzmeyer.fr.
2001:41d0:302:2200::180: ERROR: REFUSED
193.70.85.11: ERROR: REFUSED

% check-soa -tcp dyn.bortzmeyer.fr
ns1-dyn.bortzmeyer.fr.
2001:41d0:302:2200::180: OK: 2023032702
193.70.85.11: OK: 2023032702
```

(The REFUSED is because check-soa does not send the RD - recursion desired - bit.)

All the presentations done at the end of the hackathon are online <https://github.com/IETF-Hackathon/ietf116-project-presentations>, including our presentation <https://github.com/IETF-Hackathon/ietf116-project-presentations/blob/main/dot-unilateral-probing.pdf>.