Le résolveur DNS sécurisé de FDN

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 25 avril 2023. Dernière mise à jour le 26 avril 2023

https://www.bortzmeyer.org/fdn-dot-doh.html

L'association FDN a récemment annoncé <https://www.fdn.fr/ouverture-des-services-dot-doh/> que son service de résolveur <https://www.bortzmeyer.org/resolveur-dns.html> DNS public était désormais accessible avec les protocoles de sécurité DoT et DoH, ce qui améliore grandement ce service.

Un peu de contexte pour commencer. Le résolveur <https://www.bortzmeyer.org/resolveur-dns.html> DNS est un composant tout à fait critique de nos activités sur l'Internet. L'essentiel de ce que l'on fait sur l'Internet impliquera une ou plusieurs requêtes DNS, et ces requêtes sont envoyées à notre résolveur. S'il est en panne, c'est à peu près l'équivalent de pas d'Internet du tout. S'il ment, s'il ne renvoie pas sincèrement ce que les serveurs faisant autorité <https://www.bortzmeyer.org/serveur-dns-faisant-a html> lui ont dit, on peut être emmené n'importe où. Or, les résolveurs mentent souvent, par exemple à des fins de censure, motivée par la politique ou par les finances.

Pour contourner cette censure, il existe plusieurs méthodes, une des plus simples étant de changer de résolveur DNS. Beaucoup de gens utilisent donc des résolveurs DNS **publics**, c'est-à-dire accessibles par tous et toutes. Le plus connu est sans doute Google Public DNS. Pour tester, nous allons utiliser la commande dig, d'abord avec un résolveur de Free, qui ment :

```
% dig @212.27.40.240 sci-hub.se
...
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1575
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
...
;; ANSWER SECTION:
sci-hub.se. 3600 IN A 127.0.0.1

;; Query time: 4 msec
;; SERVER: 212.27.40.240#53(212.27.40.240) (UDP)
;; WHEN: Tue Apr 25 18:43:07 CEST 2023
;; MSG SIZE  rcvd: 55
```

Il prétend que `sci-hub.se` a comme adresse IP `127.0.0.1`, ce qui est faux (c'est l'adresse de la machine locale). Vérifions en demandant à Google :

```
% dig @8.8.8.8 sci-hub.se
...
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63292
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
...
;; ANSWER SECTION:
sci-hub.se. 60 IN A 186.2.163.219

;; Query time: 156 msec
;; SERVER: 8.8.8.8#53(8.8.8.8) (UDP)
;; WHEN: Tue Apr 25 18:44:39 CEST 2023
;; MSG SIZE  rcvd: 55
```

Google dit la vérité, Sci-Hub est bien accessible à l'adresse IP `186.2.163.219`. On peut aussi vérifier avec les sondes RIPE Atlas <`https://atlas.ripe.net/`> :

```
% blaeu-resolve --requested 100 --country FR --type A sci-hub.se
[186.2.163.219] : 44 occurrences
[127.0.0.1] : 40 occurrences
[ERROR: NXDOMAIN] : 6 occurrences
Test #52634681 done at 2023-04-25T16:33:30Z
```

On voit que la plupart des sondes en France voient la vraie adresse, les autres étant derrière un résolveur menteur, qui donne la fausse adresse `127.0.0.1`, ou même prétendent que le nom n'existe pas (`NXDOMAIN`).

Évidemment, si Google Public DNS ne ment pas, il pose d'autres problèmes, par exemple en matière de vie privée, mais aussi de souveraineté numérique. C'est pour cela qu'il est bon qu'il existe de nombreux autres résolveurs. Ainsi, l'association FDN fournit depuis longtemps un résolveur DNS public non menteur (et j'en profite pour passer à IPv6) :

```
% dig @2001:910:800::12 sci-hub.se
...
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59004
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
...
;; ANSWER SECTION:
sci-hub.se. 60 IN A 186.2.163.219

;; Query time: 40 msec
;; SERVER: 2001:910:800::12#53(2001:910:800::12) (UDP)
;; WHEN: Tue Apr 25 18:47:09 CEST 2023
;; MSG SIZE  rcvd: 55
```

Jusqu'à présent, j'ai utilisé le transport par défaut du DNS, UDP (RFC 768 [1]). Ce transport a plusieurs défauts très sérieux, qui font qu'il ne devrait jamais être utilisé pour accéder à un résolveur DNS public :

— Aucune confidentialité, les requêtes circulent en clair (c'est bien sûr pareil quand vous parlez au résolveur de votre FAI ou de l'organisation où vous travaillez, mais la distance est plus courte alors que, en accédant à un résolveur public, vous passez par plusieurs AS dont chacun peut vous espionner).
— Aucune authentification, vous avez peut-être confiance dans Google ou dans FDN mais vous ne pouvez pas être sûr que c'est bien à eux que vous parlez, puisque UDP ne protège absolument pas contre l'usurpation d'adresse IP <https://www.bortzmeyer.org/usurpation-adresse-ip.html> et que, de toute façon, quelqu'un a pu détourner le routage, comme cela s'est déjà fait <https://www.bortzmeyer.org/turquie-dns-frnog.html>.
— En outre, du fait de cette absence de protection contre l'usurpation d'adresse IP permet des attaques par réflexion <https://www.bortzmeyer.org/attaques-reflexion.html> utilisant le résolveur DNS public. Donc, un résolveur DNS public permettant UDP n'est pas seulement dangereux pour ses utilisateurs mais aussi pour tout l'Internet. (C'est entre autres pour cela que mon résolveur public <https://doh.bortzmeyer.fr/policy> n'offre pas d'UDP du tout.)

La première solution serait d'utiliser TCP qui lui, au moins, protège contre l'usurpation d'adresse IP. Mais il ne résout pas les questions d'authentification et de confidentialité. D'où l'existence de protocoles qui résolvent ces questions, en utilisant la cryptographie : DoT ("*DNS over TLS*", RFC 7858) et DoH ("*DNS over HTTPS*", RFC 8484). Les deux fournissent les mêmes services, la différence principale est que DoH est plus difficile à bloquer, car il utilise un port habituel, le 443 de HTTPS. Avec DoT (ou DoH), les trois problèmes mentionnés plus haut disparaissent. Voyons ce que cela donne pour les résolveurs de FDN, nous allons utiliser cette fois le client DNS kdig, qui fait partie de Knot <https://www.knot-dns.cz/> :

```
% kdig +tls @2001:910:800::12 sci-hub.se
;; TLS session (TLS1.3)-(ECDHE-SECP256R1)-(ECDSA-SECP384R1-SHA384)-(CHACHA20-POLY1305)
;; ->>HEADER<<- opcode: QUERY; status: NOERROR; id: 30070
;; Flags: qr rd ra; QUERY: 1; ANSWER: 1; AUTHORITY: 0; ADDITIONAL: 1

;; EDNS PSEUDOSECTION:
;; Version: 0; flags: ; UDP size: 1232 B; ext-rcode: NOERROR

;; QUESTION SECTION:
;; sci-hub.se.          IN A

;; ANSWER SECTION:
sci-hub.se.          60 IN A 186.2.163.219

;; Received 55 B
;; Time 2023-04-25 20:04:34 CEST
;; From 2001:910:800::12@853(TCP) in 84.4 ms
```

C'est parfait, tout marche. Comme vous le voyez au début, la session TLS a été authentifiée avec ECDSA et elle est chiffrée avec ChaCha20. Et avec DoH ?

```
% kdig +https=/dns-query @2001:910:800::12 sci-hub.se
;; TLS session (TLS1.3)-(ECDHE-SECP256R1)-(ECDSA-SECP384R1-SHA384)-(CHACHA20-POLY1305)
;; HTTP session (HTTP/2-POST)-([2001:910:800::12]/dns-query)-(status: 200)
```

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc768.txt

```
;; ->>HEADER<<- opcode: QUERY; status: NOERROR; id: 0
;; Flags: qr rd ra; QUERY: 1; ANSWER: 1; AUTHORITY: 0; ADDITIONAL: 1

;; EDNS PSEUDOSECTION:
;; Version: 0; flags: ; UDP size: 1232 B; ext-rcode: NOERROR

;; QUESTION SECTION:
;; sci-hub.se.          IN A

;; ANSWER SECTION:
sci-hub.se.          60 IN A 186.2.163.219

;; Received 55 B
;; Time 2023-04-25 20:07:21 CEST
;; From 2001:910:800::12@443(TCP) in 105.4 ms
```

On voit qu'une requête HTTPS a été faite à `https://[2001:910:800::12]/dns-query` et qu'elle a marché.

J'ai un peu triché, car, en fait, kdig n'avait pas authentifié. Puisque DoT et DoH reposent tous les deux sur TLS, utilisons un client TLS pour regarder le certificat, ici celui de GnuTLS :

```
% gnutls-cli -p 853 2001:910:800::12
...
Connecting to '2001:910:800::12:853'...
- Certificate type: X.509
- Got a certificate list of 3 certificates.
- Certificate[0] info:
 - subject `CN=resolver0.fdn.fr', issuer `CN=R3,O=Let's Encrypt,C=US', serial 0x049a52af0442ad4f67002cf0364c
 ...
```

Le certificat a été émis par Let's Encrypt et porte sur le serveur `resolver0.fdn.fr` (ainsi que, plus loin, `ns0.fdn.fr`). Dans les deux tests avec kdig, je n'indiquais pas le nom, juste l'adresse IP, et l'authentification avec le certificat n'était pas possible. Dans un cas plus réel, on doit aussi indiquer au client DoT ou DoH le nom qu'on veut vérifier.

Notez également qu'il n'est pas du tout indispensable d'utiliser un résolveur public, vous pouvez avoir votre propre résolveur. Vous maximisez ainsi le contrôle sur la résolution de noms. Mais attention : si avoir votre propre résolveur résout le problème de la censure, il ne protège pas votre vie privée, les requêtes sortant en clair (tout au plus peut-on dire que, pour le FAI, regarder le trafic de son résolveur est plus facile que d'analyser le trafic réseau). Si on veut en plus protéger sa vie privée, une solution intéressante est de combiner les deux méthodes : avoir son propre résolveur **et** faire suivre ("*to forward*") les requêtes à un résolveur extérieur, par exemple un résolveur public. (Et, au passage, n'oubliez pas de valider localement avec DNSSEC, ce qui vous protégera contre des mensonges si le résolveur à qui vous faites suivre n'est pas si digne de confiance que ça.) Avec Knot-resolver <`https://www.knot-resolver.cz/`>, cela se fait ainsi (notez qu'on indique le nom, pour l'authentification) :

```
policy.add(policy.all(policy.TLS_FORWARD({{'2001:910:800::12', hostname='ns0.fdn.fr'}})))
```

Et avec Unbound, c'est ainsi :

```
forward-zone:
  name: "."
  forward-addr: 2001:910:800::12@853
        forward-tls-upstream: yes
```

Normalement, Unbound authentifie le serveur DoT distant si on ajoute son nom (`forward-addr: 2001:910:800::12@853#ns0.fdn.fr`) mais je n'arrive pas à le faire fonctionner (« *"error : ssl handshake failed crypto error :0A000086 :SSL routines : :certificate verify failed"* »).

On peut aussi tester les résolveurs de FDN avec les sondes RIPE Atlas <https://atlas.ripe.net/>, puisqu'elle savent faire du DoT (mais pas du DoH) :

```
% blaeu-resolve --tls --nameserver=2001:910:800::40 --requested 100 --country FR --type A sci-hub.se
Nameserver 2001:910:800::40
[186.2.163.219] : 92 occurrences
Test #52639363 done at 2023-04-25T18:41:38Z
```

C'est parfait, tout a marché.

Outre la solution d'avoir un résolveur local qui fait suivre à un résolveur public comme celui de FDN, vous pouvez aussi utiliser directement celui de FDN sur les terminaux comme expliqué dans la documentation d'ARN <https://arn-fai.net/fr/internet-alternatif/dns>. Voici par exemple une copie d'écran sur Android 11 :

Une fois qu'il a été configuré ainsi, l'ordiphone contacte bien FDN, comme vu ici avec tcpdump :

```
20:15:28.914148 IP6 2a01:e34:dada:e1d0:7f67:9eba:979e:eb57.37376 > 2001:910:800::16.853: Flags [S], seq 24076408
20:15:28.917330 IP6 2001:910:800::16.853 > 2a01:e34:dada:e1d0:7f67:9eba:979e:eb57.37376: Flags [S.], seq 3248346
20:15:28.918686 IP6 2a01:e34:dada:e1d0:7f67:9eba:979e:eb57.37376 > 2001:910:800::16.853: Flags [.], ack 1, win 1
20:15:28.919074 IP6 2a01:e34:dada:e1d0:7f67:9eba:979e:eb57.37376 > 2001:910:800::16.853: Flags [P.], seq 1:518,
20:15:28.921301 IP6 2001:910:800::16.853 > 2a01:e34:dada:e1d0:7f67:9eba:979e:eb57.37376: Flags [.], ack 518, win
20:15:28.921923 IP6 2001:910:800::16.853 > 2a01:e34:dada:e1d0:7f67:9eba:979e:eb57.37376: Flags [P.], seq 1:230,
20:15:28.922704 IP6 2a01:e34:dada:e1d0:7f67:9eba:979e:eb57.37376 > 2001:910:800::16.853: Flags [.], ack 230, win
```

tcpdump voit qu'il y a une connexion, pourrait découvrir que c'est du TLS, mais évidemment pas savoir quel(s) était(ent) le(s) nom(s) de domaine demandé(s).