Le problème DNSSEC venu du froid

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 20 juin 2023

https://www.bortzmeyer.org/gl-dnssec.html

Nous sommes en 2023, tous les logiciels libres DNS font correctement du DNSSEC depuis longtemps, les bavures des débuts semblent bien oubliées. Mais il y a des logiciels bizarres qui continuent à être utilisés, et qui ont des bogues subtiles, ce qui frappe en ce moment le TLD du Groenland.

Commençons par les observations : `.gl` a un problème pour ses sous-domaines comme `com.gl` (mais il y en a d'autres). Ces sous-domaines sont délégués, mais au même jeu de serveurs que le TLD (ce qui est une mauvaise pratique, voir à la fin). On voit le problème sur DNSviz <https://dnsviz.net/d/com.gl/ZJEMOQ/dnssec/>, avec le message « *"NSEC3 proving non-existence of com.gl/DS : No NSEC3 RR matches the SNAME (com.gl)"* ». On voit aussi avec les sondes RIPE Atlas <https://atlas.ripe.net/> qu'un certain nombre de résolveurs n'arrive pas à résoudre les noms :

```
% blaeu-resolve --requested 100 --displayvalidation --type NS com.gl
[d.nic.gl. ns1.anycastdns.cz. ns2.anycastdns.cz.] : 73 occurrences
[ERROR: SERVFAIL] : 23 occurrences
[ERROR: NXDOMAIN] : 2 occurrences
[ (TRUNCATED - EDNS buffer size was 4096 )  d.nic.gl. ns1.anycastdns.cz. ns2.anycastdns.cz.] : 1 occurrences
Test #55841896 done at 2023-06-20T09:20:58Z
```

Ça veut dire quoi ?

Déboguons avec dig. Le TLD a trois serveurs de noms, testons avec `d.nic.gl` :

```
% dig @d.nic.gl NS com.gl
...
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 23986
;; flags: qr aa rd; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1
...
;; ANSWER SECTION:
com.gl. 86400 IN NS ns2.anycastdns.cz.
com.gl. 86400 IN NS d.nic.gl.
com.gl. 86400 IN NS ns1.anycastdns.cz.
```

OK, ça marche. Mais tout déconne quand un résolveur **validant** essaie d'accéder à ce nom, et demande un enregistrement de type DS ("*Delegation Signer*") :

```
% dig @d.nic.gl DS com.gl
...
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 55694
;; flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
...
;; AUTHORITY SECTION:
gl. 900 IN SOA a.nuuk.nic.gl. gl-admin.tele.gl. 2022119306 900 1800 6048000 3600
```

Zut et zut, cette fois, cela ne marche pas, on a un NXDOMAIN ("*No Such Domain*") alors que la requête précédente, **pour le même nom**, répondait positivement. Le serveur faisant autorité <https://www.bortzmeyer.org/serveur-dns-faisant-autorite.html> dit donc n'importe quoi.

Est-ce que cette réponse erronée est validable ? Oui ! Le serveur renvoie des enregistrements de type NSEC3 (RFC 5155 [1]) :

```
% dig +dnssec @d.nic.gl DS com.gl
;; Truncated, retrying in TCP mode.
...
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 43612
;; flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 8, ADDITIONAL: 1
...
;; AUTHORITY SECTION:
...
S2UOJG57GTBJ0M12ECAU9CSFD38EJNDN.gl. 3600 IN NSEC3 1 1 10 504D114B SAGKR73F41OMFFI8TDE1CGHOQM502SIH NS SOA R
...
BBTTMJM743SRPQ6J4KQDIUC73E3C1HOA.gl. 3600 IN NSEC3 1 1 10 504D114B BSHTF866A32E02RJ617EUE8CCP45A6V4 NS DS RR
...
```

Mais, comme le dit DNSviz plus haut, ce ne sont pas les bons. Comme, avec NSEC3, la preuve de non-existence n'utilise pas les noms de domaine mais un condensat de ces noms, il va falloir faire quelques calculs. On va utiliser le programme `knsec3hash`, distribué avec Knot. D'abord, trouvons les paramètres de condensation :

```
% dig NSEC3PARAM gl
...;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30244
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 1
...
;; ANSWER SECTION:
gl. 0 IN NSEC3PARAM 1 0 10 504D114B
```

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc5155.txt`

1 est l'algorithme de condensation <https://www.iana.org/assignments/dnssec-nsec3-parameters/dnssec-nsec3-parameters.xml#dnssec-nsec3-parameters-3>, ici SHA-1, 0 une série d'options (inutilisée ici), 10 est le nombre de condensations successives à effectuer, 504D114B est le sel. Nous pouvons donc calculer le condensat :

```
% knsec3hash 504D114B 1 10  gl
s2uojg57gtbj0m12ecau9csfd38ejndn (salt=504D114B, hash=1, iterations=10)
```

C'est bien le nom s2uojg57gtbj0m12ecau9csfd38ejndn vu plus haut. (On savait de toute façon qu'il s'agissait de l'apex de la zone, en raison des types SOA, DNSKEY, NSEC3PARAM, etc, qu'elle contient.) Bon, et le com.gl :

```
% knsec3hash 504D114B 1 10  com.gl
biavqpkequ599fc57pv9d4sfg1h0mtkj (salt=504D114B, hash=1, iterations=10)
```

Le second enregistrement NSEC3 nous dit qu'il n'y a rien entre bbttmjm743srpq6j4kqdiuc73e3c1hoa et bshtf866a32e02rj617eue8ccp45a6v4. Donc, pas de biavqpkequ599fc57pv9d4sfg1h0mtkj, ce qui implique que com.gl n'existe pas. Réponse cohérente mais fausse (le domaine existe bien).

Notez que les deux autres serveurs faisant autorité pour .gl donnent une autre réponse, toute aussi fausse mais différente :

```
% dig +short NS gl
ns2.anycastdns.cz.
d.nic.gl.
ns1.anycastdns.cz.

% dig +dnssec @ns1.anycastdns.cz DS com.gl
...
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36229
;; flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 1
...
;; AUTHORITY SECTION:
s2uojg57gtbj0m12ecau9csfd38ejndn.gl. 3600 IN NSEC3 1 1 10 504D114B SAGKR73F41OMFFI8TDE1CGHOQM502SIH NS SOA RRSIG
```

Cette fois, on a un NOERROR, moins grave que le faux NXDOMAIN, mais le NSEC3 est pour gl, pas pour le nom demandé (d'où le message d'erreur de DNSviz).

Comment cela a pu se produire ? Je ne connais pas de logiciel ayant ce comportement, mais je note que le domaine a pris un sérieux risque : com.gl est délégué (il y a un ensemble d'enregistrements NS) mais aux mêmes serveurs que .gl. C'est une mauvaise pratique car elle rend plus difficile le déboguage (les données de la zone fille peuvent masquer celles de la zone parente). Ce n'est pas forcément la cause primaire, mais elle n'aide pas à déblayer le problème.

(Merci à Viktor Dukhovni, inlassable détecteur et débogueur de problèmes DNSSEC dans les TLD. Il vient d'ailleurs d'en trouver un autre pour le .scot.)

---

https://www.bortzmeyer.org/gl-dnssec.html