RFC 8482 : Providing Minimal-Sized Responses to DNS Queries that have QTYPE=ANY

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 11 janvier 2019

Date de publication du RFC : Janvier 2019

https://www.bortzmeyer.org/8482.html

Lorsqu'un client DNS envoie une requête à un serveur DNS, le client indique le **type** de données souhaité. Contrairement à ce qu'on lit souvent, le DNS ne sert pas à « traduire des noms de domaine en adresses IP ». Le DNS est une base de données généraliste, qui sert pour de nombreux types de données. Outre les types (AAAA pour les adresses IP, SRV pour les noms de serveurs assurant un service donné, SSHFP ou TLSA pour les clés cryptographiques, etc), le DNS permet de demander **tous** les types connus du serveur pour un nom de domaine donné ; c'est ce qu'on appelle une requête ANY. Pour différentes raisons, l'opérateur du serveur DNS peut ne pas souhaiter répondre à ces requêtes ANY. Ce nouveau RFC spécifie ce qu'il faut faire dans ce cas.

Les requêtes comme ANY, qui n'utilisent pas un type spécifique, sont souvent informellement appelées « méta-requêtes ». Elles sont spécifiées (mais de manière un peu ambigüe) dans le RFC 1035 [1], section 3.2.3. On note que le terme « ANY » n'existait pas à l'époque, il est apparu par la suite.

Pourquoi ces requêtes ANY défrisent-elles certains opérateurs de serveurs DNS ? La section 2 de notre RFC explique ce choix. D'abord, quelle était l'idée derrière ces requêtes ANY ? La principale motivation était de déboguage : ANY n'est pas censé être utilisé dans la cadre du fonctionnement normal du DNS, mais lorsqu'on veut creuser un problème, vérifier l'état d'un serveur. Voici un exemple de requête ANY envoyée au serveur faisant autorité pour le nom de domaine `anna.nic.fr` :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc1035.txt`

```
% dig +nodnssec @ns1.nic.fr ANY anna.nic.fr
...
;; flags: qr aa rd; QUERY: 1, ANSWER: 3, AUTHORITY: 5, ADDITIONAL: 10
...
;; ANSWER SECTION:
anna.nic.fr. 600 IN A 192.134.5.10
anna.nic.fr. 172800 IN TXT "EPP prod"
anna.nic.fr. 600 IN AAAA 2001:67c:2218:e::51:41
...
;; Query time: 2 msec
;; SERVER: 2001:67c:2218:2::4:1#53(2001:67c:2218:2::4:1)
...
```

Le serveur `2001:67c:2218:2::4:1` a renvoyé les données pour trois types, A (adresse IPv4), AAAA (adresse IPv6) et TXT (un commentaire).

Certains programmeurs comprenant mal le DNS ont cru qu'ils pouvaient utiliser les requêtes ANY pour récupérer à coup sûr toutes les données pour un nom (ce fut par exemple une bogue de qmail). Voyons d'abord si ça marche, en essayant le même nom avec un autre serveur DNS :

```
% dig @::1 ANY anna.nic.fr
...
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
...
;; ANSWER SECTION:
anna.nic.fr. 595 IN AAAA 2001:67c:2218:e::51:41
...
;; Query time: 0 msec
;; SERVER: ::1#53(::1)
```

On n'a cette fois récupéré que l'adresse IPv6 et pas les trois enregistrements. C'est parce que cette fois, le serveur interrogé était un résolveur, pas un serveur faisant autorité. Il a bien répondu en donnant toutes les informations qu'il avait. Simplement, il ne connaissait pas tous les types possibles. Ce comportement est le comportement normal de ANY ; ANY (n'importe lesquelles) ne veut pas dire ALL (toutes). Il veut dire « donne-moi tout ce que **tu** connais ». Il n'y a jamais eu de garantie que la réponse à une requête ANY contiendrait toutes les données (c'est la bogue fondamentale de l'usage DNS de qmail). Si la réponse, comme dans l'exemple précédent, n'a pas de données de type A, est-ce que cela veut dire qu'il n'y a pas de telles données, ou simplement que le serveur ne les connaissait pas ? On ne peut pas savoir. Bref, il ne faut pas utiliser les requêtes ANY vers un résolveur ou, plus exactement, il ne faut pas compter sur le fait que cela vous donnera toutes les données pour ce nom. ANY reste utile pour le déboguage (savoir ce que le résolveur a dans le cache) mais pas plus.

Et vers un serveur faisant autorité, est-ce que là, au moins, on a une garantie que cela va marcher ? Même pas car certains serveurs faisant autorité ne donnent qu'une partie des données, voire rien du tout, pour les raisons exposées au paragraphe suivant.

En effet, les requêtes ANY ont des inconvénients. D'abord, elles peuvent être utilisées pour des attaques par réflexion, avec amplification <`https://www.bortzmeyer.org/attaques-reflexion.html`>. La réponse, si le serveur envoie toutes les données possibles, va être bien plus grosse que la question, assurant une bonne amplification <`https://www.bortzmeyer.org/amplification-dns-combien.`

html> (cf. RFC 5358, et section 8 de notre nouveau RFC). Bien sûr, ANY n'est pas le seul type de requête possible pour ces attaques (DNSKEY ou NS donnent également de « bons » résultats.)

Ensuite, les requêtes ANY peuvent permettre de récupérer facilement toutes les données sur un nom de domaine, ce que certains opérateurs préféreraient éviter. C'est à mon avis l'argument le plus faible, le même effet peut être obtenu avec des requêtes multiples (il y a 65 536 types possibles, mais beaucoup moins en pratique) ou via le "*passive DNS*" <https://www.bortzmeyer.org/dnsdb.html>.

Enfin, avec certaines mises en œuvre des serveurs DNS, récupérer toutes les informations peut être coûteux. C'est par exemple le cas si le dorsal du serveur est un SGBD où les données sont accessibles uniquement via la combinaison {nom, type}.

Bref, il est légitime que le gérant d'un serveur DNS veuille bloquer les requêtes ANY. Mais que doit-il répondre dans ce cas ? Ne pas répondre du tout, comme le font certains pare-feux programmés et configurés avec les pieds n'est pas une solution, le client réémettra, gaspillant des ressources chez tout le monde. Notre RFC suggère un choix de trois méthodes (section 4) :

— Envoyer un sous-ensemble non-vide des données connues. Le client ne saura jamais si on lui a envoyé toutes les données, seulement celles connues du serveur, ou bien un sous-ensemble. Mais rappelez-vous qu'il n'y a jamais eu aucune garantie qu'ANY renvoie tout. Cette technique ne change donc rien pour le client.
— Variante du précédent, essayer de deviner ce que veut le client et le lui envoyer. Par exemple, en renvoyant tous les A, AAAA, MX et CNAME qu'on a et en ignorant les autres types d'enregistrement, on satisfera sans doute la grande majorité des clients.
— Envoyer un enregistrement HINFO, solution décrite en détail plus loin.

Toutes ces réponses sont compatibles avec le protocole existant. Le RFC 1035, section 3.2.3, est relativement clair à ce sujet : ANY n'est pas la même chose que ALL (section 7 de notre RFC). Notez que notre nouveau RFC n'impose pas une politique particulière ; ce RFC ne dit pas qu'il faut renvoyer la réponse courte, il dit « si vous le faites, faites-le avec une des trois méthodes indiquées ».

Notez que le comportement du serveur peut dépendre de si la question était posée sur UDP ou sur TCP (section 4.4 du RFC). En effet, avec TCP, le risque d'attaque par réflexion est très faible.

Voici un exemple chez Cloudflare, la société qui a le plus « poussé » pour ce RFC <https://blog.cloudflare.com/what-happened-next-the-deprecation-of-any/> :

```
% dig +nodnssec @ns5.cloudflare.com. ANY cloudflare.com
...
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54605
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
...
;; ANSWER SECTION:
cloudflare.com. 3789 IN HINFO "ANY obsoleted" "See draft-ietf-dnsop-refuse-any"

;; Query time: 4 msec
;; SERVER: 2400:cb00:2049:1::a29f:209#53(2400:cb00:2049:1::a29f:209)
;; WHEN: Wed Dec 19 16:05:12 CET 2018
;; MSG SIZE  rcvd: 101
```

Le client recevant ces réponses les mémorise de la manière classique. S'il trouve un HINFO, il peut décider de l'utiliser pour répondre aux requêtes ANY ultérieures. (C'est un changement de la sémantique du type HINFO mais pas grave, ce type étant très peu utilisé.)

Mais pourquoi HINFO, type qui était normalement prévu pour donner des informations sur le modèle d'ordinateur et sur son système d'exploitation (RFC 1035, section 3.3.2) ? La section 6 du RFC traite cette question. Le choix de réutiliser (en changeant sa sémantique) un type existant était dû au fait que beaucoup de boitiers intermédiaires bogués refusent les types DNS qu'ils ne connaissent pas (un nouveau type NOTANY avait été suggéré), rendant difficile le déploiement d'un nouveau type. HINFO est très peu utilisé, donc considéré comme « récupérable ». Ce point a évidemment fait l'objet de chaudes discussions à l'IETF, certains étant choqués par cette réutilisation sauvage d'un type existant. Le type NULL avait été proposé comme alternative mais l'inconvénient est qu'il n'était pas affiché de manière lisible par les clients DNS actuels, contrairement à HINFO, qui permet de faire passer un message, comme dans l'exemple Cloudflare ci-dessus.

Un HINFO réel dans une réponse peut être mémorisé par le résolveur et empêcher certaines requêtes ANY ultérieures. De la même façon, le HINFO synthétique généré en réponse à une requête ANY peut masquer des vrais HINFO. Attention, donc, si vous avez des HINFO réels dans votre zone, à ne pas utiliser ce type dans les réponses aux requêtes ANY.

Mais les HINFO réels sont rares. En janvier 2017, en utilisant la base DNSDB `<https://www.bortzmeyer.org/dnsdb.html>`, je n'avais trouvé que 54 HINFO sur les trois millions de noms de `.fr`, et la plupart n'étaient plus dans le DNS. Les meilleurs étaient :

```
iiel.iie.cnam.fr. IN HINFO "VS3100" "VMS-6.2"
wotan.iie.cnam.fr. IN HINFO "AlphaServer-1000" "OSF"
```

Il y a peu de chance qu'une VaxStation 3100 soit encore en service en 2017 :-) Autres HINFO utilisés de façon créative :

```
www-cep.cma.fr. IN HINFO "bat. B" ""
syndirag.dirag.meteo.fr. IN HINFO "VM Serveur Synergie 1 operationnel" "RHEL 5.4"
www.artquid.fr. IN HINFO "Artquid" "ArtQuid, La place de marche du Monde de l'Art (Antiquites, Objets d'art,
```

Le HINFO de `syndirag.dirag.meteo.fr` est toujours en ligne et illustre très bien une raison pour laquelle les HINFO sont peu utilisés : il est pénible de les maintenir à jour (la machine n'est probablement plus en RHEL 5.4).

Notons que d'autres solutions avaient été étudiées à l'IETF pendant la préparation de ce RFC (section 3) :

— Créer un nouveau code de retour (au lieu de l'actuel NOERROR). Le nommer, par exemple, NOTALL. Mais les résolveurs actuels, recevant un code inconnu, auraient simplement renvoyé la question à d'autres serveurs.
— Utiliser une option EDNS, mais l'expérience prouve que les options EDNS inconnues ont du mal à passer les boitiers intermédiaires.

— Simplement décider que ANY devenait un type d'enregistrement comme les autres, au lieu de rester un « méta-type » avec traitement spécial. Dans ce cas, comme il n'y a pas de données de type ANY dans la zone, la réponse aurait été NODATA (code de retour NOERROR mais une section de réponse vide). Cela s'intégre bien avec DNSSEC par exemple. Mais cela casserait les attentes des logiciels clients, et cela ne leur laisserait rien à mémoriser (à part la réponse négative).

Le choix a donc été fait de renvoyer quelque chose, afin que le client s'arrête là, et qu'il puisse garder quelque chose dans sa mémoire (cache).

On notera que cela laisse entier le problème du client qui voudrait récupérer, par exemple, adresse IPv4 (A) et IPv6 (AAAA) avec une seule requête. Plusieurs approches ont été proposées à l'IETF mais aucune adoptée.

Les techniques de ce RFC sont-elles disponibles ? NSD a depuis sa version 4.1 une option `refuse-any`, mais pas conforme au RFC (elle répond avec le bit TC indiquant la troncature, ce que le RFC refuse explicitement). BIND a depuis la version 9.11 une option `minimal-any` qui, elle, est conforme. En mettant `minimal-any yes;` dans la configuration, BIND répondre aux requêtes ANY avec un seul enregistrement. BIND n'utilise donc pas la solution « HINFO » mais le RFC permet ce choix.