Une bogue amusante de BIND avec les enregistements NSEC3

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 8 septembre 2010

http://www.bortzmeyer.org/bogue-bind-nsec3.html

Pratiquement chaque TLD qui a décidé de déployer DNSSEC a trouvé une nouvelle bogue de BIND, car ce TLD utilisait une combinaison d'options non encore testée. Lors des essais préalables à la signature de .FR, le cas d'une zone NSEC3 ne comportant qu'un seul nom signé a révélé une nouvelle bogue.

Faites l'expérience vous-même avec toutes les versions de BIND jusqu'à la 9.6.2 ou 9.7.2 incluses. Créez une zone où un seul nom contient des enregistrements faisant autorité, par exemple :

```
example.       IN      SOA     ns1 root (
                2010083009              ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                         2419200        ; Expire
                          86400 )       ; Negative Cache TTL

        IN      NS      ns1.nic.fr.
        IN      NS      ns2.nic.fr.

        IN      TXT "Test"

toto    IN   NS ns1.bortzmeyer.org.
        IN   NS ns2.bortzmeyer.org.

foobar    IN   NS ns1.bortzmeyer.org.
        IN   NS ns2.bortzmeyer.org.

baz    IN   NS ns1.example.net.

$INCLUDE "Kexample.+008+31414.key"
```

Cette zone contient plusieurs noms (`example`, `toto.example`, etc). Signée avec NSEC ou même avec le NSEC3 du RFC 5155[1], elle ne pose pas de problème. Mais si on ajoute à NSEC3 l'option *"opt-out"* (section 6 du RFC 5155, option `-A` de `dnssec-signzone` par exemple `dnssec-signzone -3 0BADDCAF -H 1 -A -P -z example`), il n'y a qu'un seul nom signé, l'apex, `example`. En effet, les autres noms comme `foobar.example` ne contiennent pas d'enregistrement faisant autorité, seulement des délégations. La chaîne NSEC3 boucle alors sur elle-même, il n'y a qu'un seul condensat cryptographique, pour l'apex :

```
P9UQ7AKGQS75E10PPBQ2FRKUSSNEKID6.example. 86400 IN NSEC3 1 1 1 \
             0BADDCAF P9UQ7AKGQS75E10PPBQ2FRKUSSNEKID6 NS SOA TXT
```

Comme le savent tous les programmeurs, les bogues surviennent en général aux limites, pour des tailles de 0 ou de 1 ou de $2\hat{\}32$... Ici, lorsque la chaîne des NSEC3 est de longueur 1, BIND n'est plus capable de servir la zone correctement. Lançons-le avec ce fichier de configuration :

```
options {
        directory "/tmp/bind";
        recursion no;
        dnssec-enable yes;
        listen-on port 9053 {any;};
};

zone "example" {
        type master;
        file "example.signed";
};
```

puis, `named -g -c /mon/fichier.conf`. Si on interroge ce BIND avec dig sur l'apex, tout va bien :

```
% dig +dnssec @localhost -p 9053 A example

; <<>> DiG 9.6-ESV-R1 <<>> +dnssec @localhost -p 9053 A example
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33585
;; flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 1
...
;; AUTHORITY SECTION:
example.                600     IN      SOA ns1.example. root.example. 2010083009 604800 86400 2419200 8640
example.                600     IN      RRSIG   SOA 8 1 600 20100929110548 20100830110548314 example. pdY
P9UQ7AKGQS75E10PPBQ2FRKUSSNEKID6.example. 86400 IN NSEC3 1 1 1 0BADDCAF P9UQ7AKGQS75E10PPBQ2FRKUSSNEKID6 NS
...
```

Si on interroge sur un nom qui n'existe pas, tout va également bien, le NSEC3 est bien renvoyé pour prouver la non-existence :

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

```
% dig +dnssec @localhost -p 9053 A doesnotexistatall.example

; <<>> DiG 9.6-ESV-R1 <<>> +dnssec @localhost -p 9053 A doesnotexistatall.example
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 8774
;; flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 1
...
;; AUTHORITY SECTION:
example.                600     IN      SOA ns1.example. root.example. 2010083009 604800 86400 2419200 86400
example.                600     IN      RRSIG   SOA 8 1 600 20100929110548 20100830110548314l4 example. pdY7ja1e
P9UQ7AKGQS75E10PPBQ2FRKUSSNEKID6.example. 86400 IN NSEC3 1 1 1 0BADDCAF P9UQ7AKGQS75E10PPBQ2FRKUSSNEKID6 NS SOA
...
```

Mais, si on interroge le serveur sur un nom qui existe, patatras :

```
% dig +dnssec @localhost -p 9053 A foobar.example

; <<>> DiG 9.6-ESV-R1 <<>> +dnssec @localhost -p 9053 A foobar.example
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63947
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 1
...
;; AUTHORITY SECTION:
foobar.example.         600     IN      NS      ns1.bortzmeyer.org.
foobar.example.         600     IN      NS      ns2.bortzmeyer.org.
...
```

Pas de NSEC3 pour prouver l'absence d'enregistrements (`ANSWER: 0`) ! Un tel comportement est tout à fait contraire aux normes et conduit les résolveurs validants à refuser de transmettre la réponse (`SERVFAIL`, pour "*Server Failure*").

Si vous voulez voir le comportement normal d'un serveur de noms DNSSEC, regardez des zones signées avec NSEC3 et "*opt-out*" comme `.ORG` ou `.PM`. Un test comme `dig +dnssec @a2.org.afilias-nst.info A bortzmeyer.org` (nom existant mais non signé) ou `dig +dnssec @d.nic.fr A nic.pm` (même chose) doit renvoyer des NSEC3.

Alors, pourquoi BIND n'en renvoyait-il pas, alors que NSD, par exemple, le faisait bien ? C'est à cause de ce cas limite, un seul nom signé et donc une chaîne NSEC3 bouclant immédiatement, cas qui n'était pas prévu. En ajoutant un nom bidon (le `_` est là pour éviter une collision avec un nom délégué) :

```
; Adding this name workarounds the bug
_test    IN TXT "Test again"
```

Alors, tout marche :

```
% dig +dnssec @localhost -p 9053 A foobar.example

; <<>> DiG 9.6-ESV-R1 <<>> +dnssec @localhost -p 9053 A foobar.example
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41292
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 6, ADDITIONAL: 1
...
;; AUTHORITY SECTION:
foobar.example.         600     IN      NS      ns2.bortzmeyer.org.
foobar.example.         600     IN      NS      ns1.bortzmeyer.org.
P9UQ7AKGQS75E10PPBQ2FRKUSSNEKID6.example. 86400 IN NSEC3 1 1 1 0BADDCAF 50P7FK09A20A8BATD9DPV7NQPP1QM4OB NS
...
```

Notez bien que le NSEC3 pointe désormais vers un second condensat cryptographique puisqu'il y a deux noms signés.

La bogue a été signalée à l'ISC et corrigée quelques jours après. Vous ne pouvez pas encore la voir en ligne car BIND 9 (contrairement à son successeur <http://www.bortzmeyer.org/bind10-avance.html>) a un mode de développement assez fermé <http://www.isc.org/software/bind/news>, où les rapports de bogue et le VCS ne sont pas publics. Mais voici ce qui apparaitra dans la liste des changements :

```
2951.   [bug]           named failed to generate a correct signed response
                        in a optout, delegation only zone with no secure
                        delegations. [RT #22007]
```

Et voici le patch complet (en ligne sur http://www.bortzmeyer.org/files/bind-rt22007.patch). Notez que la correction elle-même ne fait que quelques lignes de C mais que le patch a nécessité le développement d'un test de non-régression, destiné à empêcher cette bogue de réapparaître dans le futur. La livraison d'une nouvelle version de BIND est en effet précédée de tests automatiques du serveur ainsi compilé.