OpenDNSSEC et les états des clés

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 10 février 2010

https://www.bortzmeyer.org/opendnssec-states.html

Une des plaies de DNSSEC est la gestion des clés. Cette gestion comprend plusieurs aspects, garder les clés disponibles (mais aussi empêcher les méchants de mettre la main dessus) et assurer leur remplacement régulier (pour faire face à certaines attaques par cryptanalyse, et pour s'assurer que les procédures fonctionnent), ainsi que le remplacement d'urgence en cas de problème. Un tel travail peut être fait à la main mais est très pénible donc le logiciel libre OpenDNSSEC <http://www.opendnssec.org/> a été développé pour automatiser une partie de ces tâches.

J'ai déjà parlé d'OpenDNSSEC dans un article « OpenDNSSEC, ou comment faciliter l'utilisation de DNSSEC <https://www.bortzmeyer.org/opendnssec-debut.html> ». Je me concentre dans ce nouvel article sur les **états** que peuvent prendre les clés dans OpenDNSSEC.

La clé passe en effet par plusieurs états au cours de sa vie. Ces différents états sont nécessaires car on ne peut pas publier une clé et espérer qu'elle soit instantanément utilisable partout, en raison des caches des résolveurs qui peuvent garder, par exemple, des « vieilles » signatures. Avec DNSSEC, il faut penser en quatre dimensions... Les termes utilisés par OpenDNSSEC viennent du RFC 7583 [1] (section 3.1). Ce sont :

— "*Generated*" : la clé a été créée (OpenDNSSEC permet de créer des clés à l'avance, par exemple à des fins de sauvegarde, avec `ods-ksmutil key generate --policy test --interval 1Y`, où 1Y veut dire de générer des clés pour l'année qui vient),
— "*Published*" : la clé est publiée dans le DNS, sous la forme d'un enregistrement `DNSKEY` mais elle n'a pas encore forcément atteint tous les résolveurs (car ils peuvent avoir une vieille version de l'ensemble des `DNSKEYs` dans leur cache),
— "*Ready*" : la clé est publiée et a atteint tous les résolveurs,
— "*Active*" : la clé est utilisée pour signer,
— "*Retired*" : la clé n'est plus utilisée pour signer mais est encore publiée car des vieilles signatures faites avec cette clé sont peut-être encore dans des caches,

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc7583.txt`

— "*Dead*" : la clé est encore publiée mais elle ne devrait plus servir à rien, toutes les signatures ont quitté les caches,
— "*Removed*" : la clé est complètement supprimée.
— "*Revoked*" : non décrit dans `draft-morris-dnsop-dnssec-key-timing`, il désigne une clé qui a été annulée « à la main ».

On le voit, le principal trajet suivi par les clés est "*PUBLISHED*" -¿ "*READY*" -¿ "*ACTIVE*" -¿ "*RETIRED*" mais il existe des tas d'autres possibilités. Par exemple, si on configure OpenDNSSEC pour exiger la sauvegarde (`<RequireBackup/>` dans `conf.xml`), les clés commencent toutes leur vie dans l'état `GENERATED` et pas `PUBLISHED`.

L'outil `ods-ksmutil` permet de voir l'état des clés (pour tous les exemples ici, la politique de signature, dans `kasp.xml`, a utilisé des durées de vie très courtes, pour faciliter les tests, par exemple `<Lifetime>P1D</Lifetime>` soit une journée seulement pour la ZSK) :

```
% sudo ods-ksmutil key list
SQLite database set to: /var/opendnssec/kasp.db
Keys:
Zone:                           Keytype:      State:    Date of next transition:
...
bortzmeyer.fr                   ZSK           active    2010-02-10 16:09:10
bortzmeyer.fr                   ZSK           ready     next rollover
...
```

Ici, deux ZSK sont publiées, une seule servant à signer. Voyons leurs identificateurs :

```
% sudo ods-ksmutil key list --verbose
...
Zone:            State:    Date of next transition:  CKA_ID:                            Keytag:
bortzmeyer.fr    active    2010-02-10 16:09:10       1d117886e98f80b3dd8516d9a975c877    24243
bortzmeyer.fr    ready     next rollover             980d24646af64877ac20aa051b4d29cb    48961
```

24243 est actuellement active. Cela peut se vérifier dans le fichier de zone qu'a signé OpenDNSSEC :

```
bortzmeyer.fr.  86400   IN      RRSIG   TXT 8 2 86400 \
            20100210202914 20100210082904 24243 ... ;{id = 24243}
; Pas de RRSIG avec 48961
```

mais les deux clés sont publiées :

```
bortzmeyer.fr. 600 IN DNSKEY 256 3 8 Aw...R ;{id = 48961 (zsk), size = 1024b}
bortzmeyer.fr. 600 IN DNSKEY 256 3 8 Aw...F ;{id = 24243 (zsk), size = 1024b}
```

Et l'évolution d'une clé ? Si on fait des `ods-ksmutil key list` à intervalles réguliers, on voit :

```
active    2010-02-08 16:08:57       0be66578e00401781l0646f3f62a8eef  41398
ready     next rollover             759d2847a33e80326007a4a2587b7a98  53104
```

puis :

```
active    2010-02-08 16:08:57       0be66578e0040178110646f3f62a8eef  41398
ready     next rollover             759d2847a33e80326007a4a2587b7a98  53104
publish   2010-02-08 15:34:03       1d117886e98f80b3dd8516d9a975c877  24243
```

On a vu l'apparition d'une nouvelle clé, la 24243, qui a été générée automatiquement par OpenDNSSEC et publiée. On voit ensuite :

```
retire    2010-02-09 04:24:04       0be66578e0040178110646f3f62a8eef  41398
active    2010-02-09 16:09:04       759d2847a33e80326007a4a2587b7a98  53104
ready     next rollover             1d117886e98f80b3dd8516d9a975c877  24243
```

41398 ne sert plus à signer mais est encore publiée. 53104 la remplace comme clé principale. 24243 est publiée depuis assez longtemps pour être utilisable. Enfin, on n'a plus que :

```
active    2010-02-09 16:09:04       759d2847a33e80326007a4a2587b7a98  53104
ready     next rollover             1d117886e98f80b3dd8516d9a975c877  24243
```

41398 est passée en état "*Dead*". `ods-ksmutil` ne l'affiche plus (pour éviter d'être noyé sous les nombreuses clés mortes). La version 1.0.0 de OpenDNSSEC ne fournit pas d'option pour voir ces clés (même chose pour celles en état "*Generated*").

Si on veut voir quand même cette clé, on peut attaquer directement le HSM (ici, softHSM <`http://trac.opendnssec.org/wiki/SoftHSM`>):

```
% sudo sqlite3 /var/opendnssec/kasp.db
sqlite> SELECT id, state, publish, active, dead, keytype, algorithm, location \
             FROM Keydata_view \
             WHERE location = '0be66578e0040178110646f3f62a8eef' OR ...;
85|6|2010-02-06 15:08:49|2010-02-07 16:08:57|2010-02-09 05:09:06|256|8|0be66578e0040178110646f3f62a8eef
86|6|2010-02-07 15:08:56|2010-02-08 16:09:04|2010-02-10 05:09:13|256|8|759d2847a33e80326007a4a2587b7a98
87|4|2010-02-08 15:09:03|2010-02-09 16:09:10||256|8|1d117886e98f80b3dd8516d9a975c877
```

(La commande a été tapée une journée plus tard donc, entre temps, `759d2847a33e80326007a4a2587b7a98` (53104) a également été supprimée.)

Depuis la première rédaction de cet article, les états de OpenDNSSEC ont été mieux documentés <`https://wiki.opendnssec.org/display/DOCS/Key+States`>.