Unbound, un résolveur DNSSEC libre

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 2 avril 2007. Dernière mise à jour le 18 février 2008

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

Tout le monde se met à DNSSEC ce mois-ci, essentiellement grâce aux intentions clairement exprimées par le gouvernement US de maintenir son contrôle sur la racine du DNS y compris via les signatures DNSSEC. Le monde bruit donc de nouvelles et de projets sur DNSSEC, souvent plus virtuels que réels et souvent motivés par le désir de faire un beau communiqué de presse plutôt que par celui de déployer un service largement utilisé. Mais peut-on tirer profit des rares domaines signés, dès aujourd'hui ?

Pour celui qui voudrait limiter le risque d'empoisonnement des données DNS grâce à DNSSEC, il existe deux résolveurs DNS en logiciel libre, BIND et Unbound <http://www.rfc.se/unbound/prototype-resolver.html>. Ce dernier est écrit en Java et, comme la plupart des programmes Java, ne pouvait pas être utilisé dans un environnement 100 % logiciel libre, peu de programmes Java réels acceptant de tourner avec les environnements libres comme gcj. Désormais, l'implémentation de référence étant libre <http://www.sun.com/2006-1113/feature/>, on peut utiliser des programmes Java.

Unbound est encore assez beta et l'installation se fait à la main, en copiant les fichiers (dans mon cas, en `/local/share/unbound` et j'ai modifié le script de lancement `unbound-resolver.sh` pour indiquer le `CLASSPATH` Java).

Unbound se configure ensuite via le fichier `iterative-config.properties`. Les clés des domaines qu'on veut vérifier sont dans `trust_anchors`. En effet, la racine du DNS n'est pas signée à l'heure actuelle. En attendant l'aboutissement de projets comme DLV (décrit dans le RFC 4431 [1]), il faut donc configurer à la main les zones sécurisées et indiquer leurs clés. Pour récupérer celles-ci, il n'y a pas de moyen propre et standard, il faut aller à la pêche un peu partout. Dans mon fichier (en ligne sur `https://www.bortzmeyer.org/files/trust_anchors`), j'ai indiqué en commentaires les URL des sites où récupérer les clés ; question sécurité, cela vaut ce que cela vaut mais il est difficile de faire mieux aujourd'hui. En tout cas, cela vaut mieux que de simplement les récupérer dans le DNS (`dig DNSKEY domain.example.`), ce qui n'offrirait aucune sécurité.

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc4431.txt`

On peut ensuite tester Unbound avec dig. Si un nom a une signature invalide, Unbound renvoie "*Server Failure*" (SERVFAIL). Cette politique n'est pas configurable (le résolveur DNSSEC de Microsoft est apparemment tout aussi peu configurable mais, lui, il renvoie "*No such domain*" (NXDOMAIN).

Dans le fichier `iterative-config.properties`, on peut aussi indiquer, pour chaque domaine, des serveurs de noms à interroger directement si les « vrais » serveurs ne font pas de DNSSEC. Ainsi, `.net` avait des serveurs DNSSEC (mais qui ne semblent plus accessibles). L'AFNIC en a aussi pour `.fr`, accessibles de l'extérieur. Si la zone `.example` est signée sur un tel jeu de serveurs experimentaux, on peut dire à Unbound de l'utiliser ainsi :

```
dns.stub = example
dns.stub.example = 192.0.2.34 192.0.2.35
```

On peut trouver plein de noms valides et invalides (à des fins de test) en <http://www.dnssec-tools.org/testzone/>.

Après quelques jours, j'ai arrêté. Unbound plante très souvent, avec de magnifiques piles d'exceptions Java comme :

```
java.lang.IllegalArgumentException: Entry was invalid
        at se.rfc.unbound.cache.CacheCore.removeFromLRU(CacheCore.java:139)
        at se.rfc.unbound.cache.CacheCore.get(CacheCore.java:212)
        at se.rfc.unbound.cache.StandardHostCache.lookup(StandardHostCache.java:94)
        at se.rfc.unbound.DelegationPoint.addTarget(DelegationPoint.java:216)
        at se.rfc.unbound.DelegationPoint.addTargets(DelegationPoint.java:245)
        at se.rfc.unbound.DelegationPoint.addTargets(DelegationPoint.java:259)
        at se.rfc.unbound.DelegationPoint.messageToDelegationPoint(DelegationPoint.java:411)
        at se.rfc.unbound.iter.IterativeResolver.processInitRequest(IterativeResolver.java:923)
        at se.rfc.unbound.iter.IterativeResolver.handleEvent(IterativeResolver.java:743)
        at se.rfc.unbound.iter.IterativeResolver.processRequest(IterativeResolver.java:679)
        at se.rfc.unbound.validator.ValidatingResolver.processRequest(ValidatingResolver.java:873)
        at se.rfc.unbound.server.Server.processRequest(Server.java:226)
        at se.rfc.unbound.server.Server.handleReadRequest(Server.java:763)
        at se.rfc.unbound.server.Server.handleRead(Server.java:698)
        at se.rfc.unbound.server.Server.serve(Server.java:383)
        at se.rfc.unbound.server.Server.main(Server.java:1098)
```

Le problème est parfois lié à un nom particulièr (résoudre `www.ripe.net` entraine une `java.lang.NullPointe` alors que résoudre `ns-pri.ripe.net` ne crée aucun problème) et parfois pas. Dans tous les cas, Unbound-Java n'est pas prêt pour un fonctionnement en production.

Son développement semble désormais stoppé. Un autre logiciel du même nom a été publié <http://www.nominet.org.uk/news/releases/?contentId=5085>, écrit cette fois en C (ce qui règle les problèmes de licence), Unbound est donc désormais utilisable en production <https://www.bortzmeyer.org/unbound.html>. Il a une API donc on peut développer ses propres applications DNSSECisées. Ici, voici son utilisation depuis la ligne de commande :

```
% unbound-host -F ~/etc/trusted-keys -v www.ripe.net
www.ripe.net is an alias for kite-www.ripe.net. (secure)
kite-www.ripe.net has address 193.0.0.214 (secure)
kite-www.ripe.net has IPv6 address 2001:610:240:0:a843::8 (secure)
kite-www.ripe.net has no mail handler record (secure)

% unbound-host -F ~/etc/trusted-keys -v www.afnic.fr
www.afnic.fr is an alias for rigolo.nic.fr. (insecure)
rigolo.nic.fr has address 192.134.4.20 (insecure)
rigolo.nic.fr has IPv6 address 2001:660:3003:2::4:20 (insecure)
rigolo.nic.fr has no mail handler record (insecure)
```