

Unbound, un autre résolveur DNS

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 21 janvier 2009

<https://www.bortzmeyer.org/unbound.html>

Le résolveur DNS le plus courant, dans le monde du logiciel libre, est BIND. Mais il est toujours bon qu'il existe des alternatives sérieuses, pour faire face à toute éventualité. Le deuxième résolveur DNS le plus utilisé est sans doute Unbound.

Le résolveur est le logiciel serveur qui va répondre aux requêtes des clients DNS (ces clients sont typiquement des programmes ayant appelé `getaddrinfo`), en utilisant les informations qu'il va trouver chez les serveurs faisant autorité. Le résolveur est également typiquement un cache, pour éviter de refaire ces requêtes trop souvent.

Pendant de nombreuses années, seul BIND assurait cette fonction (il existait aussi quelques expériences peu fiables). Désormais, il est concurrencé par PowerDNS Recursor et Unbound. Ce dernier, après un prototype en Java <<https://www.bortzmeyer.org/unbound-dnssec.html>> est désormais, dans sa version de production, écrit en C.

Je teste Unbound sur une Debian 5.0 "lenny". Il existe un paquetage tout fait, mais d'une version un peu trop ancienne, à qui manque notamment le support de DLV (RFC 5074¹). Je vais donc créer un paquetage local en partant de celui de la version de développement de Debian, "sid") :

```
# Récupérer le source Debian. À faire sur une machine sid, puis à copier
% apt-get source unbound

# Récupérer la dernière version d'Unbound
% wget http://unbound.net/downloads/unbound-1.3.2.tar.gz

# La détacher
% tar xzvf unbound-1.3.2.tar.gz
% cd unbound-1.3.2
```

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc5074.txt>

```

# Copier le répertoire "debian", depuis le paquetage Debian
% cp -a -r ../unbound-sid/debian .

# Paquetages nécessaires pour compiler unbound (indiqués dans le champ
# Build-Depends: de debian/control)
% sudo aptitude install doxygen automake libldns-dev quilt

# Éditer le debian/changelog
% dch -i
# On y mettra:
# unbound (1.3.2-0.99) unstable; urgency=low
#
# * Local package of 1.3, to get DLV
#
# -- Stephane Bortzmeyer <bortzmeyer@nic.fr> Wed, 21 Jan 2009 15:31:23 +0100

# Créer le paquetage binaire
% dpkg-buildpackage -rfakeroot

# Voilà, le paquetage est prêt, on l'installe
% sudo dpkg -i ../unbound-host_1.3.2-0.99_i386.deb ../libunbound1_1.3.2-0.99_i386.deb ../unbound_1.3.2-0.99_

```

Notez que le démon ne démarre pas automatiquement en raison de la bogue #500176 <<http://bugs.debian.org/500176>> :

```

Starting recursive DNS server: unbound[1232549330] unbound[25036:0] error: bind: address already in use
[1232549330] unbound[25036:0] fatal error: could not open ports failed!

```

Je reconfigure BIND (dont j'ai besoin pour autre chose) pour ne **pas** écouter sur le réseau :

```

options {
listen-on-v6 { };
listen-on { };
...

```

et on reconfigure les paquetages :

```

% sudo dpkg --configure -a
Setting up unbound (1.3.2-0.99) ...
Starting recursive DNS server: unbound.

```

et tout est bon, comme on peut le vérifier dans le journal `/var/log/daemon.log` :

```

Jan 21 15:52:05 batilda unbound: [26056:0] notice: init module 0: validator
Jan 21 15:52:05 batilda unbound: [26056:0] notice: init module 1: iterator
Jan 21 15:52:05 batilda unbound: [26056:0] info: start of service (unbound 1.3.2).

```

ainsi qu'avec `dig`, en demandant à `::1` (la machine locale) :

```

% dig @::1 SOA af.
...
;; ANSWER SECTION:
af.                86400   IN      SOA     ns1.nic.af. hostmaster.nic.af. 2009012100 43200 3600 604800

```

<https://www.bortzmeyer.org/unbound.html>

Tout va bien, le résolveur est allé chercher l'information sur .af. On peut alors éditer /etc/resolv.conf et mettre :

```
% cat /etc/resolv.conf
...
# unbound local avec DLV 2009-01-21
nameserver ::1
```

Un des intérêts de Unbound (et d'un résolveur DNS sur sa propre machine) est de pouvoir valider les réponses DNS avec DNSSEC (RFC 4033 et suivants). J'édite /etc/unbound/unbound.conf pour activer la validation.

```
server:
...
dlv-anchor-file: "dlv.isc.org.key"
trust-anchor-file: "itar-trusted-anchors"
```

Où obtient-on ces fichiers ? Le premier est la clé publique du registre DLV <<https://secure.isc.org/ops/dlv:>>, trouvée sur le site de l'ISC et vérifiée par PGP :

```
% sudo wget http://ftp.isc.org/www/dlv/dlv.isc.org.key
% sudo wget http://ftp.isc.org/www/dlv/dlv.isc.org.key.asc
% gpg --verify dlv.isc.org.key.asc
gpg: Signature made Sun Sep 21 06:30:51 2008 CEST using RSA key ID 1BC91E6C
gpg: Good signature from "Internet Systems Consortium, Inc. (Signing key, 2006) <pgpkey2006@isc.org>"
...
```

Le second est un fichier de KSK ("*Key Signing Key*") comme on en trouve sur le site Web des registres qui utilisent DNSSEC. Pour éviter de toutes les récupérer une par une, on va télécharger l'ITAR <<https://itar.iana.org/>> ("*Interim Trust Anchor Repository*") de l'IANA :

```
% sudo wget https://itar.iana.org/anchors/anchors.mf
# On ne télécharge pas la signature PGP, on fait confiance à TLS
% sudo mv anchors.mf itar-trusted-anchors
```

Et on recharge Unbound :

```
% sudo /etc/init.d/unbound force-reload
```

Est-ce que tout marche ? Testons avec un nom dans un TLD **non** signé :

```
% dig +dnssec SOA com.
...
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 13, ADDITIONAL: 1
...
;; ANSWER SECTION:
com.                900      IN       SOA      a.gtld-servers.net. nstld.verisign-grs.com. 1232550840 1800 900
```

<https://www.bortzmeyer.org/unbound.html>

On a une réponse mais pas de "flag" AD ("Authentic Data"). En l'absence de signature, Unbound n'a rien pu vérifier.

Maintenant, testons avec un domain signé mais pour lequel ni le registre DLV de l'ISC ni l'ITAR ne contient de "trusted anchor" (ou "secure entry point"), de clé publique :

```
% dig +dnssec SOA ripe.net
...
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 5, ADDITIONAL: 5
...
;; ANSWER SECTION:
ripe.net.          172419  IN      SOA      ns-pri.ripe.net. dns-help.ripe.net. 2009012101 3600 7200 12
ripe.net.          172419  IN      RRSIG    SOA 5 2 172800 20090220060006 20090121060006 13992 ripe.net.
```

Il y a bien une signature (enregistrement RRSIG) mais aucun moyen de la valider.

Maintenant, prenons un domaine qui est signé et existe dans le registre DLV de l'ISC :

```
% dig +dnssec SOA mondomaineamoi.fr
...
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 5
...
;; ANSWER SECTION:
mondomaineamoi.fr. 21600  IN      SOA      ns.mondomaineamoi.fr. postmaster.mondomaineamoi.fr. 20090107
```

C'est parfait, on a le "flag" AD ("Authentic Data"), la signature a pu être vérifiée. On est désormais protégé contre un empoisonnement DNS de ce domaine.

Et avec ITAR? Le TLD suédois n'est pas dans le registre DLV mais sa clé est dans ITAR :

```
% dig +dnssec SOA iis.se
...
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 1
...
;; ANSWER SECTION:
iis.se.           3600  IN      SOA      ns.nic.se. hostmaster.iis.se. 1232358302 10800 3600 1814400
```

Ça marche aussi.

Un dernier test, que se passe t-il si les données sont invalides, par exemple parce qu'un méchant a essayé d'empoisonner notre cache DNS? Utilisons le service de test `test.dnssec-tools.org`. Je mets leurs clés dans un autre fichier de clés (Unbound peut en gérer plusieurs, ce qui est très pratique pour les mises à jour) :

```
trust-anchor-file: "manual-trusted-anchors"
```

et je teste :

<https://www.bortzmeyer.org/unbound.html>

```
% dig +dnssec A baddata-A.test.dnssec-tools.org
...
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 20526
```

et Unbound, fort logiquement, refuse de me donner ces données incorrectes et répond SERVFAIL (*"Server Failure"*). Si on trouve ce comportement trop sévère, on peut l'adoucir avec l'option `val-permissive-mode`.

Unbound n'est pas intéressant que pour DNSSEC. Il a plein d'autres options. Par exemple, supposons que votre entreprise utilise des TLD bidons comme `.local` (c'est une violation des RFC 2606 et RFC 2826 mais c'est une autre histoire). Pour résoudre les noms dans ces TLD, vous devez vous adresser à un serveur qui les connaisse. Mettons que cela soit `192.0.2.129` et `2001:DB8:63::1`. On écrit dans `unbound.conf`:

```
forward-zone:
  name: "local"
  forward-addr: 192.0.2.129
  forward-addr: 2001:DB8:63::1
```

et on peut alors résoudre les noms en `.local`. (La configuration ci-dessus suppose que `192.0.2.129` et `2001:DB8:63::1` soient récursifs, sinon il faudrait utiliser la directive `stub-zone`.)