# Faille BIND permettant une DoS via les mises à jour dynamiques

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 29 juillet 2009

https://www.bortzmeyer.org/bind-dos-update.html

---

La mise à jour, lors d'urgences frénétiques, de logiciels critiques, est une occupation courante sur l'Internet. Aujourd'hui, c'est BIND, un habitué de ce genre de distractions, qui nous offre une telle émotion, avec la faille de sécurité CVE-2009-0696 <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-0696>, alias VU#725188 <http://www.kb.cert.org/vuls/id/725188>, la possibilité de planter BIND à distance par le biais d'une requête DNS de mise à jour dynamique, même si la configuration du serveur n'autorise pas de telles mises à jour.

Annoncée le 28 juillet par l'ISC <https://www.isc.org/node/479>, cette faille, spécifique à BIND (contrairement à la faille Kaminsky <https://www.bortzmeyer.org/comment-fonctionne-la-faille-ka html>), vient d'une mauvaise analyse des requêtes de mise à jour dynamique ("*dynamic update*", normalisé dans le RFC 2136 [1]). Lorsqu'il reçoit une requête DNS spécialement fabriquée <https://www.isc.org/node/474>, où le pré-requis à la mise à jour, dans le paquet, a le type ANY (normalement jamais utilisé), le serveur plante immédiatement et écrit dans son journal :

```
Jul 29 09:10:57 lilith named[2428]: db.c:619: \
          REQUIRE(type != ((dns_rdatatype_t)dns_rdatatype_any)) failed
Jul 29 09:10:57 lilith named[2428]: exiting (due to assertion failure)
```

(On note que l'adresse IP de l'attaquant n'apparait pas dans ce message et que, de toute façon, un seul paquet UDP suffit et que son adresse source peut être mensongère.) Il n'y a plus qu'à redémarrer BIND et, entre temps, on n'a pas de résolution de noms. Si le domaine n'utilise que BIND (ce qui est hélas courant) et que l'attaquant vise tous les serveurs du domaine, il peut faire disparaitre le domaine de l'Internet. C'est d'ailleurs en raison de ce genre de risques qu'il faut que les serveurs Internet soient

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc2136.txt

de diverses origines. Par exemple, il ne faut pas utiliser que BIND, il est important d'avoir également d'autres logiciels comme NSD.

Pour que le serveur soit vulnérable, il faut aussi qu'il fasse autorité pour au moins une zone DNS de type `master` (`slave` ne suffit pas). Comme BIND, par défaut, est `master` pour `localhost` et `0.0.127.in-addr.arpa`, cela explique que la grande majorité des serveurs BIND sont vulnérables.

L'exploitation de cette faille a été rendue publique et peut-être aussi simple que ce programme (en ligne sur `https://www.bortzmeyer.org/files/dos-bind-du.pl`) écrit en Perl. Voici, vue par tshark, le paquet d'attaque (utilisant la zone `localhost` pour laquelle tous les BIND font autorité par défaut) :

```
Domain Name System (query)
    Transaction ID: 0x6142
    Flags: 0x2800 (Dynamic update)
        0... .... .... .... = Response: Message is a query
        .010 1... .... .... = Opcode: Dynamic update (5)
        .... ..0. .... .... = Truncated: Message is not truncated
        .... ...0 .... .... = Recursion desired: Don't do query recursively
        .... .... .0.. .... = Z: reserved (0)
        .... .... ...0 .... = Non-authenticated data OK: Non-authenticated data is unacceptable
    Zones: 1
    Prerequisites: 1
    Updates: 1
    Additional RRs: 0
    Zone
        localhost: type SOA, class IN
            Name: localhost
            Type: SOA (Start of zone of authority)
            Class: IN (0x0001)
    Prerequisites
        localhost: type ANY, class IN
            Name: localhost
            Type: ANY (Request for all records)
            Class: IN (0x0001)
            Time to live: 0 time
            Data length: 0
    Updates
        localhost: type ANY, class ANY
            Name: localhost
            Type: ANY (Request for all records)
            Class: ANY (0x00ff)
            Time to live: 0 time
            Data length: 0
```

On peut examiner le paquet complet <`http://www.pcapr.net/view/bortzmeyer+pcapr/2009/6/3/8/bind-dos-dynamic-update.pcap.html`> sur pcapr.net <`https://www.bortzmeyer.org/pcapr.html`>.

La solution la plus propre est de mettre à jour BIND vers une version sûre, et d'urgence, en suivant la liste <`https://www.isc.org/node/474`> donnée par l'ISC. Avec les versions officielles de l'ISC, vous pouvez voir si un serveur est à jour avec :

```
% dig @ns1.EXAMPLE.net  CH TXT version.bind.
```

et vous obtenez le numéro de version que vous pouvez comparer à la liste des versions sûres. **Attention** : avec beaucoup de paquetages BIND de différents systèmes d'exploitation, le numéro de version n'est pas forcément modifié lors de l'application de mises à jour de sécurité.

Si cette mise à jour est difficile, pour une raison ou pour une autre, le plus simple est de « démasteriser », c'est-à-dire de supprimer les zones de type `master` de la configuration (`named.conf`). Sur un serveur qui est uniquement à autorité, cela n'est pas un problème. Sur un serveur récursif, cela entrainera quelques disfonctionnements mais qui ne sont probablement pas trop graves, par rapport aux risques de la faille.

Une des raisons pour lesquelles on n'a pas toujours la possibilité de faire une mise à jour est que les fournisseurs réagissent plus ou moins rapidement. À l'heure où j'écris, Debian a fait une mise à jour <http://lists.debian.org/debian-security-announce/2009/msg00162.html> mais pas <https://bugzilla.redhat.com/show_bug.cgi?id=514292> RHEL, pourtant bien plus cher.

Une dernière solution pour protéger votre serveur est de demander au pare-feu, si vous n'utilisez pas les mises à jour dynamiques, de les bloquer. Avec Netfilter, on peut bloquer une partie de ces requêtes et donc les attaques du script cité plus haut avec :

```
iptables -A INPUT -p udp --dport 53 -j DROP -m u32 --u32 '30>>27&0xF=5'
```

Mais cela n'empêche pas tout : comme le filtre ci-dessus travaille avec des "*offsets*" fixes, il peut être contourné, par exemple, si l'attaquant rajoute des options dans le paquet. (La difficulté à écrire des règles u32 pour le DNS a été abordée dans un autre article <https://www.bortzmeyer.org/dns-netfilter-u32.html>.)

Si vous voulez capturer les requêtes DNS de mise à jour dynamique, pour analyse ultérieure, et que vous utiliser dnscap <https://www.dns-oarc.net/tools/dnscap>, une requête comme :

```
# dnscap -w updates -mu -i eth0
```

conviendra (`-mu` ne garde que les requêtes d'"*opcode*" `UPDATE`). Si vous voulez juste les voir et pas les enregistrer sur disque, remplacez `-w updates` par `-g`.