Les attaques par réflexion utilisant NTP

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 2 février 2014. Dernière mise à jour le 3 février 2014

https://www.bortzmeyer.org/ntp-reflexion.html

Sauf si vous n'êtes abonné à aucune liste de diffusion, aucun canal IRC, aucun rézosocio, aucun flux de syndication, aucun salon XMPP, bref, sauf si vous êtes coupé de tous les moyens d'informations modernes, vous savez que, depuis le mois de décembre 2013, les attaques par réflexion <https://www.bortzmeyer.org/attaques-reflexion.html> utilisant le protocole NTP ont connu une hausse spectaculaire et ont complètement chassé des médias les « vieilles » attaques DNS <https://www.bortzmeyer.org/jres-dos-2013.html>.

Un bon résumé de ces « nouvelles » attaques se trouve par exemple dans l'article de Dan Godin <http://arstechnica.com/security/2014/01/new-dos-attacks-taking-down-game-sites-deliver-c>. L'attaque est en fait connue depuis longtemps comme le documente un très bon article du Team Cymru <https://labs.ripe.net/Members/mirjam/ntp-reflections>. Le principe est le même que toutes les attaques par réflexion, l'Attaquant écrit à un tiers, le Réflecteur, en usurpant l'adresse IP de la Victime (ce qui est trop facile <https://www.bortzmeyer.org/usurpation-adresse-ip.html> aujourd'hui). Le Réflecteur va répondre à celui qu'il croit être l'émetteur mais qui est en fait la Victime. En soi, la réflexion ne change pas grand'chose sauf qu'elle est souvent accompagnée d'amplification : la réponse est plus grande que la question. Cela permet à l'Attaquant de ne pas « payer » toute l'attaque, c'est le Réflecteur qui travaille et, en prime, engage sa responsabilité, au moins civile (article 1382 du Code Civil - celui qui cause un dommage doit le réparer). Voici un exemple de trafic pendant une attaque NTP (merci à Benjamin Sonntag) :

NTP sert à synchroniser les horloges des machines et est normalisé dans le RFC 5905[1]. Voyons en pratique cette attaque. Le mécanisme le plus simple pour tester si un serveur NTP est vulnérable est fourni par la commande `ntpdc` (qui est livrée avec la mise en œuvre de référence de NTP <http://www.ntp.org/>) :

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5905.txt

```
% ntpdc -n -c monlist MACHINEÀTESTER
remote address          port local address      count m ver rstr avgint  lstint
===============================================================================
192.168.2.9            46302 192.168.2.4           25 3 4       0    171       41
192.134.4.12             123 192.168.2.4           45 4 4    1d0    101       54
192.168.2.1              123 192.168.2.4           49 3 4       0     85       65
192.168.2.7            37185 192.168.2.4           24 3 4       0    182      117
91.121.92.90             123 192.168.2.4           46 4 4    1d0     99      146
212.27.60.17             123 192.168.2.4           46 4 4    1d0     99      157
212.27.60.18             123 192.168.2.4           46 4 4    1d0     99      185
88.191.184.62            123 192.168.2.4           37 4 4    1d0    123      227
2001:648:2ffc:1225:a80   123 2a01:e35:8bd9:8bb0:ba27:ebff:feba:9094       47 4 4    1d0     96     243
```

Le serveur a répondu avec la liste de toutes les machines avec qui il a communiqué. C'est embêtant pour la vie privée. Mais c'est surtout embêtant pour la sécurité car la communication se passe sur UDP (qui n'offre aucune protection contre l'usurpation d'adresse IP) et la réponse est bien plus grande que la question :

```
% tcpdump ...
16:08:00.462739 00:1e:8c:76:29:b6 > b8:27:eb:ba:90:94, ethertype IPv4 (0x0800), length 234: 192.168.2.1.5744
16:08:00.466981 b8:27:eb:ba:90:94 > 00:1e:8c:76:29:b6, ethertype IPv4 (0x0800), length 482: 192.168.2.4.123
16:08:00.466999 b8:27:eb:ba:90:94 > 00:1e:8c:76:29:b6, ethertype IPv4 (0x0800), length 266: 192.168.2.4.123
```

Avec un paquet de 234 octets, on a obtenu deux paquets, faisant en tout 748 octets. Cela fait une amplification d'un facteur 3 mais c'est parce qu'il s'agit d'un tout petit serveur ayant peu de correspondants. Sur un gros serveur public, on peut obtenir des dizaines de milliers d'octets en réponse, donc un facteur d'amplification qui dépasse largement celui du DNS <https://www.bortzmeyer.org/amplification-dns-combien.html>. (Sur votre réseau, pendant une attaque, regardez le port source des paquets : si c'est 123, c'est du NTP, si c'est du 53, c'est du DNS. Et attention, le DNS, contrairement à NTP, peut produire des paquets longs qui seront fragmentés. De toute façon, en pratique, il vaut mieux aussi analyser le contenu des paquets pour savoir si c'est vraiment une attaque par réflexion car les attaquants font des fois des choses compliquées pour se dissimuler.)

Et trouver des serveurs NTP publics permettant cette attaque est facile, il suffit de balayer tout l'espace IPv4. C'est ce que fait, pour le bien général, le projet Open NTP Project <http://openntpproject.org/>. On peut (c'est un exemple réel, j'ai juste modifié les adresses) indiquer son préfixe d'adresses IP et obtenir la liste des réflecteurs potentiels de son réseau :

```
Responding IP   Total bytes
192.0.2.10    44000
192.0.2.66    1104
192.0.2.238   1104
```

La première machine est un gros serveur NTP public, qui fournissait une énorme amplification (il a été corrigé depuis). À noter que, dans le cas réel, la dernière machine était un routeur Juniper, les routeurs de cette marque fournissant un réflecteur NTP par défaut <http://www.gossamer-threads.com/lists/nsp/juniper/49151>.

Bon, donc, il faut tester ses serveurs, avec `ntpdc -n -c monlist` depuis l'extérieur, et utiliser le Open NTP Project, pour détecter des machines auxquelles on ne penserait pas forcément. Attention toutefois en testant avec `ntpdc -n -c monlist`, surtout depuis un site distant : si `ntpdc` répond "*Timeout*", cela ne signifie pas forcément que vous êtes sécurisé. La réponse peut prendre beaucoup de paquets UDP et, si un seul est perdu (UDP ne garantit pas l'acheminement), `ntpdc`, n'arrivant pas à tout obtenir, dit "*Timeout*". Il faut donc utiliser tcpdump pour voir s'il y a eu des réponses. Ensuite, si les tests montrent que votre machine est bien un réflecteur potentiel, comment corriger ?

Pour des machines Unix qui utilisent la mise en œuvre de référence de NTP, il faut éditer le `/etc/ntp.conf` et, par défaut, bloquer les requêtes comme `monlist`, en mettant la directive `noquery` :

```
restrict default noquery
```

Testez avec `ntpdc -n -c monlist`, vous devez désormais récupérer un "*time out*" (rappelez-vous que NTP fonctionne sur UDP, pas TCP). Cette solution résout le problème de sécurité, votre machine n'est plus un réflecteur NTP. Mais elle fait perdre un outil de déboguage bien pratique. Par exemple, ntptrace va désormais être largement inutile, n'obtenant plus de réponses (sauf, ici, au premier pas) :

```
% ntptrace
localhost.localdomain: stratum 4, offset 0.000345, synch distance 0.010345
192.0.2.7: timed out, nothing received
***Request timed out
```

On peut toutefois autoriser au moins son réseau local (notez que ce logiciel, très ancien, n'accepte pas de notation CIDR) :

```
# Restriction par défaut
restrict default noquery

# Pas de restrictions pour les copains
restrict 192.0.2.0 mask 255.255.255.0
restrict 127.0.0.1
restrict ::1
restrict 2001:db8:cafe:42:: mask ffff:ffff:ffff:ffff::
```

Ainsi, le réseau local, au moins, peut poser des questions et ntptrace est un peu plus sympa :

```
% ntptrace
localhost.localdomain: stratum 4, offset 0.000594, synch distance 0.007256
192.0.2.7: stratum 3, offset -0.002068, synch distance 0.014179
193.55.167.2: timed out, nothing received
***Request timed out
```

Mais attention, authentifier par l'adresse IP n'est pas terrible, une telle configuration permet toujours à un méchant d'attaquer une de vos machines. Il faut encore déployer des bonnes pratiques comme de bloquer sur le pare-feu les paquets entrants qui prétendent venir de chez vous. Ce problème avec ntptrace est un problème courant en sécurité : les mesures de sécurité rendent le déboguage plus compliqué. Notez que, si vous utilisez l'excellent programme de test Nagios-compatible check_ntp_time <https://www.monitoring-plugins.org/doc/man/check_ntp_time.html> (si vous ne l'utilisez pas, vous devriez <http://word.bitly.com/post/74839060954/ten-things-to-monitor>), il n'y a pas de problème, il n'utilise pas les commandes de contrôle et marche avec tout serveur NTP :

```
% /usr/share/nagios/libexec/check_ntp_time -H ntp.nic.fr
NTP OK: Offset -0.001630425453 secs|offset=-0.001630s;60.000000;120.000000;
```

Tous les détails sur la configuration sont en ligne <http://support.ntp.org/bin/view/Support/AccessRestrictions>. J'ai simplement mis `noquery` car il bloque l'attaque par réflexion+amplification. Mais il peut y avoir d'autres <http://support.ntp.org/bin/view/Support/AccessRestrictions#Section_6.5.2.> et l'excellente documentation du Team Cymru <http://www.team-cymru.org/ReadingRoom/Templates/secure-ntp-template.html> recommande :

```
restrict default kod nomodify notrap nopeer noquery
```

Je n'ai parlé ici que de la mise en œuvre de référence de NTP. Si vous utilisez le OpenNTPD d'OpenBSD <http://www.openntpd.org/>, il n'y a rien à faire, il ne répond pas aux requêtes de déboguage, de toute façon (et, par défaut, il ne répond pas du tout aux requêtes du réseau tant que vous n'avez pas mis un `listen on quelquechose`). Mais, de toute façon, n'oubliez pas de **tester** votre réseau, avec `ntpdc -n -c monlist SERVER` et avec le Open NTP Project (attention, il lui faut quelques jours pour ajuster ses données donc, si vous changez votre configuration, le résultat ne se verra pas tout de suite).

Quelques bonnes lectures supplémentaires :

— Un exemple d'analyse du pcap d'une vraie attaque <https://www.bortzmeyer.org/attaque-ntp-en-v
html>,
— L'avis officiel de vulnérabilité est le CVE-2013-5211 <http://cve.mitre.org/cgi-bin/cvename.
cgi?name=CVE-2013-5211>.
— Un des premiers articles synthétisant le problème était celui du CERT lituanien <https://
cert.litnet.lt/en/docs/ntp-distributed-reflection-dos-attacks>.
— Un bon article bien écrit et expliquant tout est celui de CloudFlare <http://blog.cloudflare.
com/understanding-and-mitigating-ntp-based-ddos-attacks>.
— En français, notez qu'il y a eu un avis du CERT-FR (ex-CERTA) <http://www.certa.ssi.
gouv.fr/site/CERTA-2014-ACT-003/index.html>.
— Et l'alerte du CERT RENATER <https://services.renater.fr/ssi/cert/avis?avis_
id=8024&filters=YTo0OntzOjQ6InllYXIiO2k6MDtzOjQ6InR5cGUiO047czo3OiJ0ZXh0X2luIjthO
page=1#avis_cert/list/40cd750bba9870f18aada2478b24840a/list/>,
— Sur les solutions présentées, voir l'annonce officielle de ntp.org <http://support.ntp.org/
bin/view/Main/SecurityNotice#DRDoS_Amplification_Attack_using>.
— Et, pour le OpenNTPD, l'article de Calomel <https://calomel.org/ntpd.html>.