RFC 4987 : TCP SYN Flooding Attacks and Common Mitigations

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 1 septembre 2007

Date de publication du RFC : Août 2007

https://www.bortzmeyer.org/4987.html

Curieusement, malgré le nombre d'attaques ayant visé le protocole TCP, aucun RFC n'avait documenté lesdites attaques et les solutions couramment utilisées comme les "*SYN cookies*".

TCP (RFC 793[1]) est de très loin le plus utilisé des protocoles de transport sur Internet. Même les applications qui font passer l'essentiel de leur trafic en UDP comme la téléphonie, dépendent de TCP pour l'établissement et la coupure de la liaison. Une attaque sur TCP a donc typiquement des effets dévastateurs et la première à avoir été largement médiatisée, l'attaque par "*SYN flooding*" contre Panix en 1996, a provoqué une prise de conscience. De nombreuses défenses contre les différentes attaques ont été développées dans les années qui ont suivi et ce RFC est le premier à les documenter.

La section 2 de notre RFC décrit en détail le "*SYN flooding*" qui consiste à envoyer des paquets de demande d'ouverture de connexion (des paquets **SYN**) sans jamais répondre aux accusés de réception (packets **ACK**). Les mises en œuvre typiques de TCP ont un tableau de taille fixe pour leurs connexions TCP et le seul paquet SYN suffit à réserver une case de ce tableau, jusqu'à l'expiration d'un délai qui est typiquement de plusieurs minutes.

Sans avoir besoin de révéler son adresse IP, avec un trafic relativement faible, un attaquant peut donc empêcher la machine cible d'accepter des nouvelles connexions TCP, empêchant ainsi tout service.

Un outil comme hping permet d'automatiser cette attaque, sans que l'attaquant aie même à savoir programmer (non, je ne donnerai pas les détails gratuitement).

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc793.txt

La section 3 du RFC s'attaque ensuite aux mesures de défense comme les "*SYN cookies*", qui consiste à ne **pas** allouer du tout d'**état** à la réception du paquet SYN, mais à encoder les informations contenues dans ce paquet dans le numéro de séquence TCP initial. La réception du paquet ACK, le deuxième paquet envoyé par le client, permettra de vérifier qu'un SYN avait bien été envoyé et que l'ACK du serveur avait bien été reçu. Cette idée, mise au point par Dan Bernstein, est conforme au principe « ne pas allouer d'état sans authentification » posé dans la section 4.1.1 du RFC 4732.

Les "*SYN cookies*" posent quelques problèmes à TCP (décrits dans la section 3.6 du RFC ou, par exemple, dans le manuel sur FreeBSD). C'est pour cela qu'ils ne sont typiquement pas activés par défaut. Par exemple, sur une Debian, il faut les configurer à `yes` dans le fichier `/etc/network/options` (qui indique au script de démarrage qu'il doit mettre à jour la variable Linux `/proc/sys/net/ipv4/tcp_syncookies`).

Aussi, d'autres solutions existent comme le "*SYN cache*", qui alloue un état, mais de petite taille ou comme l'abandon, lorsque la table des connexions en cours est pleine, des connexions non actives les plus anciennes.