RFC 6978 : A TCP Authentication Option NAT Extension

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 20 juillet 2013

Date de publication du RFC : Juillet 2013

https://www.bortzmeyer.org/6978.html

Le protocole d'authentification des paquets TCP **AO** ("*Authentication Option*"), normalisé dans le RFC 5925 [1], a une limite : il ne fonctionne pas du tout lorsque la session est établie à travers au moins un routeur NAT. Ce nouveau RFC propose une extension qui lui permet d'authentifier quand même les paquets dans ce cas, au prix d'une légère baisse de la sécurité.

En effet, l'authentification AO range et génère ses clés en tenant compte d'un ensemble de paramètres de la session (RFC 5925, sections 3.1 et 3.2), parmi lesquels se trouvent les adresses IP de source et de destination, ainsi que les ports de source et de destination. Un routeur NAT modifie en général l'adresse IP source et les routeurs existants (qui font souvent du NAPT et pas du NAT à proprement parler, cf. RFC 2663) modifie également le port source. AO ne retrouve donc plus ses clés (RFC 5925, section 9.2). Ce n'était pas un gros problème pour le premier client d'AO, le protocole BGP : on met rarement des routeurs BGP derrière du NAT. Mais cette limite est gênante pour généraliser AO à d'autres utilisations, sans compter les futures améliorations de TCP, par exemple le "*multipath*" du RFC 6824, où la même connexion TCP utilise plusieurs adresses IP.

Donc, la nouvelle option se nomme TCP-AO-NAT et consiste (section 4 de notre RFC) à ajouter au MKT ("*Master Key Tuple*", cf. RFC 5925, section 3.1) deux booléens, `localNAT` et `remoteNAT` indiquant si un routeur NAT est présent en local ou en distant. Si `localNAT` est vrai, AO va mettre à zéro l'adresse IP source et le port source avant de calculer les clés. Si c'est `remoteNAT` qui est vrai, ce sera l'adresse et le port de destination qu'on ignorera.

Le MKT n'est pas transmis à la machine distante (et n'apparait donc pas dans l'option TCP AO), il est typiquement configuré à la main des deux côtés. La valeur à donner à `localNAT` et `remoteNAT` est déterminée manuellement, mais elle peut aussi être découverte par les méthodes habituelles de

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc5925.txt`

détection de NAT. Le client situé immédiatement derrière un routeur NAT (par exemple la machine de M. Michu à la maison) va mettre `localNAT` à 1. Le serveur en face va mettre `remoteNAT` à 1. Si les deux partenaires sont derrière un NAT (et utilisent une technique comme celle du RFC 8489), les deux booléens peuvent être à 1.

Notez bien qu'AO ne protège que TCP et ne tient pas compte du contenu des paquets. Si un ALG sur le trajet modifie les adresses contenues, mettons, dans une connexion FTP, TCP-AO-NAT n'y pourra rien.

Attention, rappele la section 6 : TCP-AO-NAT revient à ignorer certaines valeurs qui identifient la connexion et donc à diminuer l'entropie. Si `localNAT` est vrai, on passe d'une source de hasard pour la KDF qui est composée de deux adresses IP, deux ports et deux ISN (l'"*Initial Sequence Number*" de TCP) à une seule adresse, un seul port et les deux ISN. Si les deux booléens `localNAT` et `remoteNAT` sont vrais, il ne reste plus que les deux ISN comme source d'entropie. Bref, l'extension normalisée dans ce RFC **diminue** la sécurité. Toutefois, comme les ports ne sont pas toujours très aléatoires (malgré le RFC 6056) et les adresses encore moins, l'essentiel de l'entropie venait des deux ISN, de toute façon.