RFC 8968 : Babel Routing Protocol over Datagram Transport Layer Security

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 12 janvier 2021

Date de publication du RFC : Janvier 2021

https://www.bortzmeyer.org/8968.html

Le protocole de routage Babel, normalisé dans le RFC 8966[1], n'offre par défaut aucune sécurité. Notamment, il n'y a aucun moyen d'authentifier un routeur voisin, ni de s'assurer que les messages suivants viennent bien de lui. Sans même parler de la confidentialité de ces messages. Si on veut ces services, ce RFC fournit un moyen : sécuriser les paquets Babel avec DTLS.

La sécurité est évidemment souhaitable pour un protocole de routage. Sans elle, par exemple, une méchante machine pourrait détourner le trafic. Cela ne veut pas dire qu'on va toujours employer une solution de sécurité. Babel (RFC 8966) est conçu entre autres pour des réseaux peu ou pas gérés, par exemple un événement temporaire où chacun apporte son PC et où la connectivité se fait de manière ad hoc. Dans un tel contexte, déployer de la sécurité ne serait pas facile. Mais, si on y tient, Babel fournit désormais deux mécanismes de sécurité (cf. RFC 8966, section 6), l'un, plutôt simple et ne fournissant pas de confidentialité, est décrit dans le RFC 8967, l'autre, bien plus complet, figure dans notre RFC et repose sur DTLS (qui est normalisé dans le RFC 9147). DTLS, version datagramme de TLS, fournit authentification, intégrité et confidentialité.

L'adaptation de DTLS à Babel est décrite dans la section 2 du RFC. Parmi les problèmes, le fait que DTLS soit très asymétrique (client-serveur) alors que Babel est pair-à-pair, et le fait que DTLS ne marche qu'en "*unicast*" alors que Babel peut utiliser "*unicast*" ou "*multicast*". Donc, chaque routeur Babel doit être un serveur DTLS, écoutant sur le port 6699 <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml> (qui n'est pas le 6696 du Babel non sécurisé). Le routeur Babel continue à écouter sur le port non chiffré, et, lorsqu'il entend un nouveau routeur s'annoncer en "*multicast*", il va tenter de communiquer en DTLS avec lui. Puisque

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc8966.txt

DTLS, comme TLS, est client-serveur, il faut décider de qui sera le client et qui sera le serveur : le routeur de plus faible adresse sera le client (par exemple, `fe80::1:2` est inférieure à `fe80::2:1`. Le serveur différencie les sessions DTLS par le port source ou par les "*connection identifiers*" du RFC 9146. Comme avec le Babel non chiffré, les adresses IPv6 utilisées doivent être des adresses locales au lien. Client et serveur doivent s'authentifier, en suivant les méthodes TLS habituelles (chacun doit donc avoir un certificat).

Une fois que la session DTLS est en marche, les paquets Babel "*unicast*" passent sur cette session et sont donc intégralement (du premier au dernier octet) protégés par le chiffrement. Pour que la sécurité fournie par DTLS soit ne soit pas compromise, le routeur ne doit plus envoyer à son voisin de paquets sur le port non chiffré, à part des paquets `Hello` qui sont transmis en "*multicast*" (pour lequel DTLS ne marche pas) et qui permettent de découvrir d'éventuels nouveaux voisins (la règle exacte est un peu plus compliquée, regardez le RFC). On ne peut donc plus envoyer d'information en "*multicast*" (à part les `Hello`). En réception, un nœud Babel sécurisé doit ignorer les paquets non protégés par DTLS, sauf l'information de type `Hello`.

Une stricte sécurité nécessite qu'un routeur Babel sécurisé n'accepte plus de voisins qui ne gèrent pas DTLS, et ignore donc tous leurs paquets sauf ceux qui contiennent un TLV `Hello`. Autrement, les communications non sécurisées ficheraient en l'air la confidentialité de l'information. Si on veut une sécurité plus relaxée, un nœud peut faire tourner les deux versions du protocole, sécurisée et non sécurisée, mais alors, exige le RFC, uniquement sur des interfaces réseau différentes.

Comme toutes les fois où on rajoute des données, on risque de dépasser la MTU. La section 3 du RFC nous rappelle qu'il faut éviter d'envoyer des paquets qui seront fragmentés et donc qu'il faut éviter de fabriquer des paquets qui, après chiffrement, seraient plus gros que la MTU.

Rien n'est parfait en ce bas monde et l'ajout de DTLS, s'il résout quelques problèmes de sécurité, ne fait pas tout, et même parfois ajoute des ennuis. La section 5 du RFC rappelle ainsi qu'un méchant pourrait tenter des négociations DTLS nombreuses avec une machine afin de la faire travailler pour rien (une variante de Slowloris). Limiter le nombre de connexions par client n'aiderait pas puisqu'il pourrait toujours mentir sur son adresse IP.

Même en faisant tourner Babel sur DTLS, les messages de type `Hello` peuvent toujours être envoyés sans protection, et donc manipulés à loisir. Un routeur Babel doit donc toujours être capable d'utiliser des `Hello` "*unicast*" et protégés.

Et puis bien sûr, limitation classique de TLS, authentifier un routeur voisin et sécuriser la communication avec lui ne signifie pas que ce voisin est honnête ; il peut toujours mentir, par exemple en annonçant des routes vers des préfixes qu'il ne sait en fait pas joindre.

À ma connaissance, il n'existe pas encore de mise en œuvre de ce RFC.