# MinimaLT, un remplaçant réaliste pour TCP, TLS et IPsec ?

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 26 mai 2013

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

---

Il y a des chercheurs ambitieux. En voici qui proposent un protocole qui veut remplacer à la fois TCP **et** TLS **et** IPsec. Ils exagèrent ? Pas complètement. Le projet est prometteur (mais encore à ses tout débuts) mais il y a quelques points noirs qu'ils « oublient » de mentionner.

Le projet MinimaLT (à ne pas confondre avec l'application Apple qui s'écrit minimALT) ne semble pas avoir encore publié de code. Il y un article assez détaillé <http://www.ethos-os.org/~solworth/minimalt-20130522.pdf> (lien alternatif <http://cr.yp.to/tcpip/minimalt-20130522.pdf>), un joli poster publicitaire <http://datasys.cs.iit.edu/events/GCASR13/docs/poster216.pdf> et c'est tout. Donc, pour l'instant, tout est à étudier avec prudence.

À quel problème s'attaque MinimaLT ? Les auteurs estiment que du TCP sans chiffrement est vraiment trop dangereux dans l'Internet d'aujourd'hui et qu'il faudrait du chiffrement partout. Des protocoles comme TLS (RFC 6347 [1]) ou IPsec (RFC 4301) visent à fournir de la cryptographie mais souffrent de différentes faiblesses, soit de performance (ce qui amène certains à ne pas imposer TLS, de peur des conséquences sur les délais pour l'utilisateur) ou de complexité (qui utilise réellement IPsec, à part des tunnels point-à-point entre deux sites de la même organisation ?).

MinimaLT vise donc à rendre le chiffrement moins coûteux et plus simple. Le principe de base est de créer à la demande un tunnel chiffré dès que deux machines communiquent. Ensuite, tout le trafic MinimaLT entre ces deux machines (même s'il provient de deux applications sans aucun rapport entre elles) passe par ce tunnel. On ne paiera donc qu'une fois certains coûts cryptographiques, alors qu'avec TLS on les paie à chaque connexion.

Et pour authentifier la machine en face, afin d'être sûr qu'on n'est pas en train de parler au vilain Homme du Milieu ? MinimaLT utilise pour cela des certificats X.509 trouvés dans le DNS. J'en parlerai plus longuement plus loin car c'est le principal point noir de MinimaLT.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6347.txt

Quel est l'intérêt de ces tunnels ? C'est que l'établissement d'un contexte cryptographique (les clés utilisées, notamment) nécessite pas mal d'échanges entre les deux machines et introduit donc une latence <https://www.bortzmeyer.org/latence.html> importante. MinimaLT (dont le nom veut dire "*Minimal Latency Tunneling*") ne peut pas complètement faire disparaître ces échanges mais il les **amortit** sur un plus grand nombre de connexions. La première fois qu'Alice parle à Bob, il faudra attendre. Mais le tunnel reste ouvert ensuite et pourra servir aux communications suivantes. L'établissement de ces connexions dans le tunnel pourra être très rapide puisque l'essentiel du boulot a déjà été fait.

Cela a des conséquences dont l'article parle peu : en cas de communications intermittentes, il faut maintenir un état entre deux sessions, et en cas de communications intensives (pensez à Gmail utilisant MinimaLT), il y a beaucoup d'état à maintenir. Mais, surtout, cela rend les comparaisons de performances plus délicates : lorsqu'on mesure le nombre de connexions qu'on peut ouvrir par seconde et qu'on compare MinimaLT avec TCP ou TCP+TLS, on compare des pommes et des oranges... Les résultats présentés dans l'article sont d'autant plus difficiles à lire que les auteurs ne disent pas à chaque fois clairement s'ils mesurent en partant d'un état « froid » (rigoureusement aucun état) ou « chaud » (tunnel déjà établi).

On notera que le tunnel fonctionne sur UDP, afin de réussir à passer les "*middleboxes*" qui infestent l'Internet. C'est dommage mais c'est réaliste : la solution architecturalement propre (créer un nouveau protocole de transport) ne serait pas déployable, avec toutes les machines sur le trajet qui se croient autorisées à bloquer tous les protocoles qu'elles ne connaissent pas (SCTP avait le même problème et a fait le même choix, cf. RFC 6951.). MinimaLT doit ensuite réinventer toutes les fonctions de TCP (délivrance des messages garantie, et dans l'ordre, contrôle de congestion) ce qui est beaucoup de travail et offre beaucoup de possibilités d'erreur.

Un gros problème de toute solution de chiffrement est d'établir l'authenticité du pair situé en face. Si Alice croit parler à Bob mais qu'elle parle en fait à Mallory, le chiffrement ne servira à rien. Il faut donc s'assurer de l'identité du pair. Ce point, pourtant crucial, est très vite traité dans l'article. On comprend qu'il existe un annuaire des caractéristiques des machines, comprenant leurs clés cryptographiques et des certificats qui lient ces clés aux noms des machines. Le DNS est utilisé pour réaliser cet annuaire. DNSSEC n'est pas mentionné car la sécurité repose sur la signature des certificats récupérés dans le DNS. MinimaLT reprend donc tous les problèmes de sécurité de X.509, notamment l'absence de lien entre une AC et un domaine (n'importe quelle AC peut signer pour n'importe quel domaine, sans que le titulaire du domaine ne puisse s'y opposer, cf. RFC 6394). À noter que l'idée de mettre des certificats dans le DNS est ancienne mais que l'article ne cite pas un seul des travaux antérieurs (par exemple, DANE - RFC 6698 - n'est même pas mentionné en passant).

(Un expert en sécurité anonyme me fait aussi remarquer que MinimalLT ne permet apparemment que d'authentifier le répondant, pas l'initiateur de la connexion, contrairement à TLS et IPsec.)

Enfin, il faut préciser que l'interface de MinimaLT avec les applications n'est guère discutée. L'article note à juste titre que les mises en œuvre de TLS sont typiquement trop difficiles à utiliser par les applications, avec trop de possibilités d'erreur entraînant des failles de sécurité (voir l'article « "*The most dangerous code*" <http://seenthis.net/messages/101863> »). Mais utiliser une autre API veut dire qu'il faudra adapter toutes les applications. MinimaLT est conçu pour un nouveau système d'exploitation, sans base installée, Ethos <http://www.ethos-os.org/>, donc ce cas ne sera peut-être pas trop un problème. Mais il y a aussi un portage sur Linux où la question se posera.

L'article originel est riche et je n'ai pas parlé de tout, je vous encourage donc à le lire. Une discussion a lieu <http://www.reddit.com/r/netsec/comments/1evqqn/new_minimalt_network_protocol_by_petullo_zhang/> sur Reddit mais pour l'instant avec peu de contenu. Merci à l'expert sécurité anonyme pour sa relecture de cet article.

---