RFC 5387 : Problem and Applicability Statement for Better Than Nothing Security (BTNS)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 15 novembre 2008

Date de publication du RFC : Novembre 2008

https://www.bortzmeyer.org/5387.html

En matière de sécurité, le mieux peut être l'ennemi du bien. Un protocole très exigeant peut, en partie à cause de ces exigences, ne pas être déployé et laisser donc l'Internet aussi vulnérable qu'avant. C'est en partie ce qui est arrivé à IPsec et a justifié le travail sur le mécanisme « Mieux que rien » qui fait l'objet de plusieurs RFC dont notre RFC 5387[1] qui décrit les motivations et le domaine d'application de ce nouveau mécanisme ou dont le RFC 5386 qui normalise le nouveau protocole.

C'est une banalité que de dire que l'Internet n'est pas sûr. N'importe qui peut envoyer un paquet avec une fausse adresse IP, il est très facile d'écouter ce qui passe sur le réseau et il est facile de s'insérer dans une session déjà commencée. On peut se demander pourquoi le problème n'a pas encore été traité. Il l'a été, pourtant. Par exemple, IETF a depuis 1998 un protocole, IPsec (actuellement RFC 4301), qui permet de sécuriser n'importe quelle connexion Internet contre l'écoute ou la modification, en utilisant la cryptographie. IPsec est exigeant : pour établir une SA ("*Security Association*", association de sécurité entre deux machines, qui leur permettra de communiquer en toute sécurité), il faut s'authentifier, ce qui veut dire un secret partagé (un mot de passe) ou bien une signature reconnue par une autorité commune. Si IPsec peut utiliser plusieurs identités dans sa base , les adresses IP, noms de domaine et bien d'autres, toutes doivent être authentifiées (voir aussi la section 2.1). C'est lourd et compliqué et cela a pesé sérieusement dans le peu de déploiement d'IPsec (section 1 du RFC).

Alors, faut-il supprimer l'obligation d'authentification ? Après tout, si IPsec est utilisé entre deux pairs BitTorrent, quel besoin ont-ils de s'authentifier ? La plupart du temps, les pairs acceptent de servir n'importe quelle autre machine. Même chose pour un serveur Web public. Mais ce n'est pas si simple

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5387.txt

(section 1.1) car l'authentification sert aussi à empêcher les attaques de « l'Homme au Milieu », un attaquant situé entre les deux machines et qui se fait passer pour leur correspondant. Si un tel intermédiaire est présent, la cryptographie ne servira à rien puisque les clés utilisées sont connues du méchant.

C'est pourtant la voie que suit (avec prudence) le nouveau protocole BTNS ("*Better Than Nothing Security*" ou « Mieux vaut un peu de sécurité que pas de sécurité du tout - ce qui arrive si on impose des contraintes trop élevées. »).

Ainsi, le protocole telnet était bien connue pour son extrême vulnérabilité, puisque les mots de passe passaient en clair, mais, pendant longtemps, les seules solutions proposées étaient horriblement compliquées comme Kerberos (RFC 1411 pour un telnet « kerberisé »). Elles n'avaient donc jamais été déployées sérieusement avant la sortie de SSH en 1995. SSH, lui, remplacera telnet en quelques années, bien que certains experts en sécurité aient froncé le sourcil lors de son lancement en l'estimant insuffisamment blindé.

Une des solutions qu'utilise BTNS pour limiter les risques est de dépendre d'une authentification faite dans les couches hautes (IPsec travaille dans la couche 3). La section 1.3 décrit comment de tels mécanismes rendent BTNS moins dangereux qu'il n'en a l'air.

On peut se demander quel est l'intérêt de BTNS si un protocole comme TLS s'occupe de toute la sécurité. Mais BTNS est IPsec, donc protège également contre certaines attaques dans les couches basses, contre lesquelles TLS ne peut rien. Ainsi, un paquet TCP RST ("*Reset*", qui va couper la connexion) imité peut couper une session TLS, puisque TLS n'authentifie et ne protège que la couche application. Une telle attaque, pratiquée par exemple par Comcast contre ses propres clients `<http://www.eff.org/wp/packet-forgery-isps-report-comcast-affair>` ou par la censure chinoise contre ses citoyens `<http://www.cl.cam.ac.uk/~rnc1/ignoring.pdf>` (voir aussi la section 2.2.1), ne marcherait pas avec BTNS.

Les sections 2 et 3 du RFC décrivent plus en détail le problème que BTNS résout et la manière dont il le résoud (en permettant les connexions non authentifiées). La section 4, elle, revient sur le domaine d'application de BTNS. BTNS n'a pas d'intérêt pour les connexions très sécurisées, par exemple entre deux organisations qui veulent établir entre elles un VPN aussi solide que possible. Pour celles-ci, le IPsec actuel est une bonne approche. Mais BTNS est utile si :

— Un serveur public veut accepter tous les clients mais souhaite protéger les connexions après leur établissement,
— Un serveur sait authentifier ses clients après l'établissement de la connexion.

La section 4.1 explique les bénéfices de BTNS dans ce cas (pas d'infrastructure d'authentification à gérer) et la 4.2 ses faiblesses (le risque d'établir la connexion avec un imposteur).

Enfin, la section 5 analyse plus complètement les questions de sécurité parfois subtiles qui naissent de ce tournant dans IPsec. BTNS protège la connexion établie, mais pas l'établissement. Il est donc proche des autres protocoles TOFU ("*Trust On First Use*" ou, plus méchamment, "*leap of faith*", « acte de foi ») comme SSH. Contrairement à SSH, BTNS ne se souvient pas forcément des clés précédemment présentées, et ne peut donc pas détecter l'arrivée d'un imposteur (cf. section 5.7).

BTNS est donc vulnérable aux attaques de l'homme au milieu (section 5.3) mais aussi à certaines attaques DoS comme tout protocole de cryptographie. En effet (section 5.4), une machine qui établit une connexion IPsec avec vous (ce qui est plus facile avec BTNS puisqu'il n'y a plus d'authentification) peut vous faire exécuter des calculs de cryptographie assez coûteux.

À l'heure actuelle, je ne connais pas d'implémentation de BTNS dans les grands projets IPsec comme Openswan.