RFC 4301 : Security Architecture for the Internet Protocol

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 9 février 2009

Date de publication du RFC : Décembre 2005

https://www.bortzmeyer.org/4301.html

L'Internet n'est pas sûr, on le sait bien. Un méchant peut lire les paquets IP qui circulent, il peut injecter de faux paquets, avec une fausse adresse IP source, etc. Il existe des tas de façons d'aborder ce problème et ce RFC décrit une des plus ambitieuses, une complète architecture de sécurité pour IP, IPsec.

Actuellement, le problème est en général traité au niveau des applications. Par exemple, SSH va complètement sécuriser une session, quelles que soient les vulnérabilités de la couche réseau sous-jacente. Un attaquant ne pourra pas faire passer sa machine pour une autre, même s'il peut envoyer de faux paquets IP.

Cette approche est relativement simple, assez facile à déployer et règle une grande partie des problèmes. Elle a deux limites : il faut modifier chaque application. Lorsque leur nombre augmente, cela devient vite pénible. Et elle ne protège pas contre certaines attaques. Par exemple, des faux paquets TCP `RST` ("*ReSeT*", c'est-à-dire remise à zéro de la connexion TCP) peuvent couper une session SSH puisque la couche transport n'est pas authentifiée. De tels faux paquets RST se trouvent dans le monde réel, par exemple, envoyés par la censure chinoise <http://www.cl.cam.ac.uk/~rnc1/ignoring.pdf>.

D'où l'idée de IPsec : sécuriser au niveau IP afin que toutes les applications soient automatiquement protégées, contre toutes les attaques. Ce RFC décrit l'architecture générale d'IPsec. Ce protocole est réputé très complexe à comprendre (et à déployer) et le RFC est gros et complexe (alors même qu'une partie des détails figure dans d'autres RFC comme le RFC 4303 [1] ou le RFC 4305, ce dernier normalisant les algorithmes de cryptographie obligatoires).

Comme le rappelle bien la section 1.1, IPsec est une architecture de sécurité pour IP, pas pour l'Internet. La securisation complète d'un réseau comme l'Internet dépend de bien d'autres choses.

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc4303.txt`

La section 2.1 décrit le cahier des charges d'IPsec : fournir des services d'authentification, d'intégrité et de confidentialité à IP (IPv4 ou IPv6).

Commençons par quelques généralités (section 3) : IPsec crée une barrière entre une zone non protégée et une zone protégée (section 3.1). Au passage de la barrière, IPsec traduit les paquets, d'une forme protégée vers une non-protégée, ou bien le contraire.

IPsec peut sécuriser la communication de deux façons : en authentifiant les paquets, sans chercher à assurer leur confidentialité, c'est le protocole **AH**, "*Authentication Header*" (section 3.2 et RFC 4302). Ou bien authentifier tout en chiffrant pour empêcher les méchants de lire le contenu, c'est le protocole **ESP**, "*Encapsulating Security Payload*" (section 3.2 et RFC 4303). Le second est de loin le plus utilisé, entre autre parce que AH ne passe pas à travers les routeurs NAT.

Pour ces deux protocoles, AH et ESP, IPsec dispose de deux modes, **tunnel** et **transport**. Dans le mode transport, la connexion est sécurisée de bout en bout entre les deux machines. Ce mode est donc le plus sûr mais nécessite IPsec sur les deux engins. Le mode tunnel, crée un VPN entre deux routeurs et les machines des deux réseaux locaux ainsi connectés sont protégées dans leurs communications entre ces deux réseaux, sans qu'elles aient besoin d'IPsec elles-mêmes.

Comment IPsec place ses informations dans le paquet ? Avec AH, le mode transport et IPv4, le champ Protocole de l'en-tête IPv4 indique non pas le protocole de la couche transport (typiquement TCP) mais AH (valeur 51 dans le registre IANA <`https://www.iana.org/assignments/protocol-numbers/protocol-numbers`>). On trouve donc un second en-tête après l'en-tête IP normal, contenant les données notamment :

— Le SPI, "*Security Parameter Index*", un nombre qui permet à la machine de retrouver les paramètres de la session, par exemple l'algorithme cryptographique utilisé (plusieurs algorithmes sont possibles, puisque la technologie en ce domaine ne cesse pas d'évoluer),
— Le résumé cryptographique des données qu'on veut authentifier, notamment l'adresse IP source,
— L'indication du « protocole suivant », c'est-à-dire le protocole de la couche transport.

Pour un routeur IP, un paquet protégé par AH est donc un paquet IP comme un autre, le champ Protocole d'IP n'étant lu qu'à l'arrivée.

En mode tunnel, le « protocole suivant » de l'en-tête AH sera IP, tout simplement, et c'est un paquet IP complet qui sera encapsulé.

L'un des principaux problèmes avec AH concerne les routeurs NAT. Comme AH vise à garantir l'intégrité du paquet et donc l'absence de modifications, il rentre directement en conflit avec le NAT qui modifie au moins l'adresse source. Le RFC 3715 discute plus en détail ce problème. À noter que AH n'est plus désormais obligatoire dans IPsec (section 3.2), ESP le remplaçant en mieux dans presque tous les cas.

Et ESP ? Cette fois, le paquet est chiffré et l'indication du protocole suivant indique comment analyser le contenu, après déchiffrement. Le SPI joue le même rôle qu'avec AH, il indique le contexte dans lequel a été fait le chiffrement et donc notamment l'algorithme utilisé. À noter que l'en-tête IP circule donc en clair et peut, à lui seul, fournir des indications à un indiscret : qui communique avec qui et même quel genre d'applications (par les bits DSCP).

Contrairement à AH, ESP ne garantit pas l'intégrité de l'en-tête IP mais celui lui permet de passer le NAT.

Comme AH, ESP peut fonctionner en mode tunnel ou en mode transport.

En pratique, aujourd'hui, rares sont les utilisateurs de AH, la grande majorité des sessions IPsec sont protégées par ESP.

Je reviens maintenant un peu sur le SPI ("*Security Parameter Index*"). Il fait partie d'un contexte plus large, la SA ("*Security Association*", section 4). Ce contexte est ce qui permet de retrouver les paramètres d'une session IPsec notamment l'algorithme de chiffrement. Stocké dans une SAD ("*Security Associations Database*", section 4.4.2), la SA est indexée par le SPI et, selon les cas, également par les adresses IP et le protocole (AH ou ESP). À chaque arrivée d'un paquet IP entrant, c'est le SPI dans ce paquet qui servira à trouver la politique à appliquer (section 5.2).

Puisque j'ai mentionné la SAD, c'est l'occasion de revenir sur les bases de données d'IPsec (section 4.4). Leur forme exacte dépend évidemment de l'implémentation mais, conceptuellement, toute mise en œuvre d'IPsec doit avoir les bases suivantes :

— SPD ("*Security Policy Database*") qui stockera les préférences de l'administrateur réseau : quels paquets doivent être protégés par IPsec, notamment, en fonction de **sélecteurs** qui sont en général des adresses IP et numéros de port. Une politique sera par exemple, « Protéger avec ESP tous les paquets échangés avec `2001:DB8:1:27::/64` et laisser passer le reste non protégé ».
— SAD ("*Security Association Database*"), qui indique les SA en cours.
— PAD ("*Peer Authorization Database*"), liée à la gestion des clés.

La SPD est présentée en détail dans la section 4.4.1. Sur FreeBSD, on la trouve typiquement dans `/etc/ipsec.conf` et elle contient des appels à `setkey`. Les politiques possibles sont `PROTECT` (protéger le trafic avec IPsec), `BYPASS` (laisser passer, même non protégé) et `DISCARD` (ne pas laisser passer). Les sélecteurs possibles sont décrits en section 4.4.1.1. Toute mise en œuvre d'IPsec doit au moins accepter les adresses IP source et destination comme sélecteurs. Les sélecteurs pouvant se recouvrir partiellement, l'ordre des directives dans cette base est crucial (voir l'annexe B pour un algorithme permettant de supprimer ce recouvrement).

Une fois la SPD configurée, elle sera consultée à chaque paquet IP, pour déterminer l'action à entreprendre (section 5 pour le traitement des paquets IP). Par défaut, en l'absence d'une mention dans la SPD, le paquet sera mis à la poubelle.

La SAD est décrite en section 4.4.2. Elle garde mémoire des options de chaque association de sécurité, comme l'algorithme cryptographique utilisé (la section 4.4.2.1 détaille le contenu des entrées de cette base). Pour les paquets sortants, IPsec utilise typiquement un cache associant une prise réseau à une association de sécurité présente dans la SAD. Pour les paquets entrants, le SPI ("*Security Parameter Index*") dans le paquet sert d'index dans la SAD.

La gestion des clés est traitée dans la section 4.5. C'est de loin la partie la plus complexe d'IPsec (comme de beaucoup d'autres protocols de sécurité). En pratique, la grande majorité des utilisateurs d'IPsec ne comptent que sur des clés gérés manuellement (section 4.5.1), IKE (RFC 4306) étant très peu déployé.

Comme souvent en sécurité, le diable est dans les détails. Par exemple, la section 6 couvre le traitement des paquets ICMP. Que faire d'eux ? Les ignorer lorsqu'ils ne sont pas protégés par IPsec ? Du point de vue sécurité, cela serait cohérent mais, dans l'état actuel du déploiement d'IPsec, presque tous les paquets ICMP émis par des routeurs intermédiaires seraient alors ignorés, bien qu'ils apportent des informations essentielles. Notre RFC recommande donc (section 6.1.1) de laisser le choix à l'administrateur réseau de les accepter ou pas.

Si la section 7 traite un autre détail délicat, la gestion de la fragmentation, la section 8 parle des problèmes de MTU, une des plaies de l'Internet. AH et ESP ajoutent tous les deux des octets au paquet qu'ils protègent, et ces octets peuvent faire que le paquet dépasse la MTU du lien. Le problème est donc similaire à celui de tous les autres tunnels. Aux autres difficultés de déploiement d'IPsec s'ajoute donc le problème de MTU.

Le déploiement réel d'IPsec est très faible. Conçu pour sécuriser peu ou prou toutes les communications sur Internet, sa principale utilisation aujourd'hui est pour faire des tunnels entre deux réseaux de la même organisation. Comme un protocole normalisé est moins vital dans ce cas, IPsec est concurrencé par d'autres protocoles de tunnel (comme celui d'OpenVPN).

Quelles sont les implémentations pour les systèmes d'exploitation Unix ? Sur Linux, la plus répandue est Openswan mais il y a aussi StrongSwan <http://www.strongswan.org/> (les deux dérivant du même code). Sur NetBSD, IPsec est inclus dans le système et bien documenté dans la "*NetBSD IPsec FAQ*" <http://www.netbsd.org/docs/network/ipsec/>. Avec NetBSD, la SPD peut être par prise : "*Per-socket : configured via setsockopt(2) for a certain socket. You can specify like "encrypt outgoing packets from this socket". This works well when you would like to run IPsec-aware server program.* ". Sinon, elle est par paquet, la SPD se configure (c'est pareil sur FreeBSD par exemple dans `/etc/ipsec.conf`) avec `setkey(8)` :

```
#! /bin/sh
#
# packet will look like this: IPv4 ESP payload
# the node is on 10.1.1.1, peer is on 20.1.1.1
setkey -c <<EOF
add 10.1.1.1 20.1.1.1 esp 9876 -E 3des-cbc "hogehogehogehogehogehoge";
add 20.1.1.1 10.1.1.1 esp 10000 -E 3des-cbc 0xdeadbeefdeadbeefdeadbeefdeadbeefdeadbeefdeadbeef;
spdadd 10.1.1.1 20.1.1.1 any -P out ipsec esp/transport//use;
EOF
```

9876 et 10000 sont le SPI.

Ce RFC décrit « IPsec version 3 » (le RFC 2401 étant la version 2), même si ces numéros de version ne sont pas mentionnés dans le RFC (mais ils apparaissent dans d'autres documents). La section 13 liste les différences avec la version 2, notamment le fait qu'AH n'est plus obligatoire (ce qui reflète son faible usage), des SPD plus souples, etc.

Pour une excellente introduction à IPsec, je recommande l'article de Steve Friedl « "*An illustrated guide to IPsec*" <http://www.unixwiz.net/techtips/iguide-ipsec.html> », qui contient notamment de bons schémas des paquets tels qu'ils sont présents sur le câble. Parfait si on veut comprendre comment IPsec marche.

IPsec a toujours suscité des passions. Une des critiques les plus fouillées a été écrite par Niels Ferguson et Bruce Schneier dans un article très vivant et très pédagogique, "*A Cryptographic Evaluation of IPsec*" <http://www.schneier.com/paper-ipsec.html>. Ferguson et Schneier font remarquer que IPsec semble avoir été conçu par un comité, où l'important était de faire plaisir à tout le monde plutôt que de produire un protocole sûr. Ainsi, ils reprochent à IPsec d'avoir deux protocoles (AH et ESP) alors qu'ESP seul suffirait. De même, ils contestent les deux modes, tunnel et transport, puisque le mode transport assurerait l'équivalent des fonctions des deux modes. La sécurité d'un protocole dépendant largement de sa simplicité, IPsec, pour eux, ne peut pas être vraiment sûr. Cette critique couvre la version 2 mais la plupart des remarques s'appliquent encore à la version actuelle. (Notez tout de même que le RFC actuel reconnait que AH n'est pas vraiment indispensable.)