Analyse Critique des Protocoles IPSec

Jean-Jacques Puig
Jean-Jacques.Puig@int-evry.fr
I.N.T

Maryline Laurent-Maknavicius
Maryline.Maknavicius-Laurent@int-evry.fr
I.N.T

SÉCURITÉ IP

Le premier document de l'IETF portant sur la sécurité d'IP date de 1988 [1]. Mais c'est en 1994 qu'un rapport de l'IAB [2] précipite les évènements et, en août 1995, on voit apparaître les premières versions ([3], [4], [5], [6], [7]) des standards actuels ([11], [13], [19], [6], [7]).

De nombreux documents sont venus compléter ce standard ([8], [9], [10], [12], [14], [15], [16], [17], [18], [20], [21], [22], [23]), mais **dès 1995, il était possible de l'implémenter et de l'utiliser** pour constituer des réseaux privés virtuels avec des clefs définies manuellement.

Pourtant, jusqu'à aujourd'hui, IPsec a connu un déploiement limité. Cet article en explorera les raisons au travers d'une présentation de l'infrastructure de sécurité pour IP et d'une étude des protocoles IPsec.

LES PIÈCES DU PUZZLE

Une des raisons qui expliquent le déploiement tardif d'IPsec est qu'**il ne constitue qu'une des parties de la stratégie de sécurité IP**.

Afin d'avoir une vue appropriée sur les besoins de cette stratégie, il convient de déterminer les traitements qui doivent être mis en jeux au niveau réseau. Pour cela, dans ce qui va suivre, on présente les arbres des décisions à prendre lors de l'envoi et lors de la réception d'un paquet. L'analyse de ces arbres permettra de dégager les pièces nécessaires ou quasi-indispensables du puzzle. [11] évoque un tel système de décisions.

Considérons le traitement d'un paquet sortant :

- Un paquet est présenté en file de sortie.
  La politique de filtrage ...
  - ... interdit les paramètres (destination...).
    - Politique par défaut (rejeter paquet).
  - ... autorise les paramètres (destination...).
    Une Association de Sécurité ...
    - ... existe avec la destination.
      - Protection et envoi du paquet.
    - ... doit être établie avec la destination.
      Négociation d'une Association de Sécurité :
      - Echec des négociations.
        - Politique par défaut (rejeter paquet).
      - Négociations réussies.
        Vérification des Certificats :
        - Echec de la vérification.
          - Politique par défaut (rejeter paquet).
        - Succès de la vérification.
          - Protection et envoi du paquet.

Traitement d'un paquet sortant

Et pour un paquet entrant :

- Un paquet est présenté en file d'entrée.
  La politique de filtrage ...
  - ... ne valide pas {SPI, source, protocole}.
    - Politique par défaut (rejeter paquet).
  - ... valide {SPI, source, protocole}.
    Vérification (identité, intégrité, anti-rejeu), Authentification, Déchiffrement.
    - ... erreur.
      - Politique par défaut (rejeter paquet).
    - ... réussi.
      - Paquet accepté (acheminement ou couche transport).

Traitement d'un paquet entrant

Un découpage par blocs donne la liste des éléments indispensables à une mise en oeuvre réaliste d'IPsec :

- Un processus de filtrage des paquets.
- Un processus de négociation d'associations de sécurité.
- Des systèmes de protection des communications.
- Une infrastructure de certification permettant l'authentification des entités.

Au sein de l'architecture, le filtrage consiste à vérifier la conformité des paquets vis à vis d'une politique de sécurité locale. Le filtrage peut porter sur les adresses du paquet, mais aussi l'utilisateur local impliqué ou toutes autres considérations appropriées. Définir ces règles est le rôle de la Base des Politiques de Sécurité ou SPD. Si le SPD autorise le paquet, le système détermine si une Association de Sécurité (SA) existe déjà. Le cas échéant, il faut la créer.

Cet SA est un contexte de communication entre des machines. Ce contexte définit les outils cryptographiques utilisés, les algorithmes et les clefs. Le SA est référencé par un Index de Paramètre de Sécurité, le SPI. Cet index est embarqué en clair dans les paquets protégés, ce qui permet de retrouver le SA impliqué à la réception du paquet. Mais tout cela nécessite un protocole pour établir les associations de sécurité : l'Internet Key Exchange, IKE.

IKE ne résoud qu'une partie du problème de négociation; définir des services, des algorithmes, échanger des clefs est insuffisant. Il faut en effet s'assurer de l'identité du correspondant. Quand les acteurs se connaissent a priori, ils peuvent disposer de clefs qu'ils ont échangées par un canal sûr (par exemple, par disquettes). Mais **quand ils ne se connaissent pas, des tiers de confiance doivent pouvoir garantir l'identité des correspondants**. C'est le rôle d'une Infrastructure à Clefs Publiques (PKI) ou d'un Réseau de Confiance.

Une fois l'association de sécurité établie, il faut disposer de protocoles de sécurité des communications pour mettre en application les services négociés. C'est aux protocoles AH (Authentication Header) et ESP (Encryption Security Payload) qu'est dévolue cette responsabilité.

AH et ESP ont été développés et déployés relativement vite, mais leur utilisation sans les autres pièces de l'architecture (un protocole d'échange de

clefs, un PKI...) est limitée. C'est ce qui explique que les VPNs entre réseaux d'entreprises avec une configuration manuelle se soient développés le plus rapidement.

L'utilisation des politiques de sécurité était initialement très limitée. Elles agissaient alors plutôt comme un processus de routage des paquets vers un tunnel pré-existant en fonction des destinations. Pourtant, **une approche plus complète des politiques de sécurité, reflètant la dynamique des relations de confiance entre organismes et le temps de vie des tunnels, constitue une étape critique pour le déploiement des éléments suivants**...

Notamment pour la construction dynamique de SA, puisque le SPD définit les critères pour établir ces SA en fonction des caractéristiques de la communication. Pour établir ces SA, il a fallu attendre des implémentations de IKE, lesquelles (on le verra plus loin) n'apportent pas une satisfaction totale.

Bien que les services commerciaux à destination du public acceptent des négociations sans preuves d'identité, via SSL, la plupart des entreprises ne s'intéressent pas à l'établissement d'associations de sécurité avec des anonymes. Pour cette raison, **les acteurs doivent se prouver mutuellement leur identité pour entamer des négociations**. Répondre à ce problème est le but des réseaux de confiance et de l'infrastructure à clefs publiques. La lenteur de déploiement de ces systèmes à grande échelle a une répercussion directe sur le déploiement d'IPsec.

En résumé, **le déploiement d'IPsec dépend désormais de celui :**

1 **Des protocoles de négociation de SA**.
2 **Des autorités de certifications**.

LES PROTOCOLES D'IPSEC

Il est généralement admis que ces protocoles sont au nombre de trois :

- AH (Authentication Header)
- ESP (Encapsulating Security Payload)
- IKE (Internet Key Exchange)

Mais cela constitue une vision très simplifiée de la situation; hormis le fait que ces protocoles se déclinent en versions IPv4 et IPv6, les protocoles AH et ESP ont par ailleurs deux définitions, ou "modes", et IKE utilise d'autres protocoles (ISAKMP et OAKLEY).

Ce chapitre, dans un premier temps, introduit les types d'attaques rencontrées, puis, dans un deuxième temps, procède à la critique de ces trois protocoles.

Attaques typiques

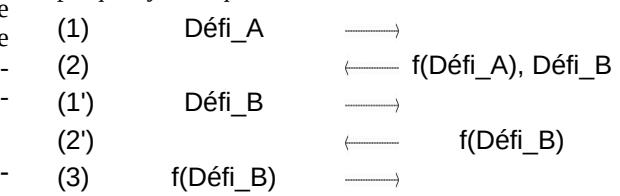
Attaque par Rejeux (Replay Attack)

Dans ce type d'attaque, des paquets sont sauvegardés par une personne malveillante, qui les renvoie plus tard sur le réseau. Cela permet de "dupliquer" les effets de ces paquets. Par exemple, considérons un paquet UDP (cela permet d'éviter la problèmatique des numéros de séquence TCP) contenant les informations (chiffrées, compressées...) suivantes: `INSERT INTO table_personnel VALUES(NULL, "François", "Perrin", 12)`. Dupliquer ce paquet peut corrompre la base de donnée `table_personnel` et y provoquer l'apparition d'un doublon ("`VALUES(NULL,`" autorise la base à générer une nouvelle clef) : Mr `François Perrin`, qui pour cette raison risque de recevoir son chèque en double à la fin du mois. Les attaques par rejeux peuvent s'effectuer sur des champs du paquet; on les appelle alors "Attaques par Copier/Coller" (Copy/Paste Attacks). Un nouveau paquet est généré par l'assaillant avec des champs copiés dans un ancien paquet. Un système d'authentification incomplet (par exemple qui n'authentifie que certaines informations utiles au traitement et qui n'assure pas de protection réelle contre le rejeu) est vulnérable à ce genre d'attaque.

Attaque par Réflexion (Reflection Attack)

Une attaque par réflexion est une forme d'attaque par Copier/Coller; elle peut prendre deux formes:

- La machine est attaquée par deux connexions (ou plus): les informations issues d'une connexion servent à duper la machine via l'autre connexion. Soient les échanges suivants, où f est un processus cryptographique symétrique:



Dans ce protocole, les correspondants s'échangent mutuellement des "challenges" (Défi_A et Défi_B); ces challenges les mettent au défi, mutuellement, de prouver qu'ils disposent du processus f (qui est généralement paramétré par une clef cryptographique). Comme l'assaillant n'est pas en mesure de calculer f(Défi_B) après le message 2, il ouvre une nouvelle connexion avec la victime, et lui demande (message 1') la réponse à son propre challenge (message 2') ! Il injecte alors cette réponse dans la première connexion, qu'il n'avait pas interrompue (message 3). A l'issue du message 3, la victime croit pouvoir faire confiance à l'assaillant, et par exemple lui accorde un accès.

- Dans certains cas, une seule connexion est suffisante. Considérons une machine qui authentifie les données envoyées par UDP au moyen d'un algorithme d'authentification symétrique. Si le protocole n'authentifie pas l'état de la transaction (le sens ou le numéro du message), l'assaillant peut copier le contenu des données du paquet et les renvoyer à la victime. Les données seront bien authentiques puisqu'elle les a authentifiées elle-même !

Il convient de bien insister sur le fait que c'est la **propriété de symétrie des algorithmes utilisés qui permet les attaques par réflexion**.

Détournement de Connexion (Connection Hijacking)

Cette attaque concerne les protocoles connectés; ces derniers s'appuient généralement, dans le modèle IP,

sur TCP. Au cours de cette attaque, l'assaillant attend que la victime passe avec succès un processus de contrôle d'accès, puis prend sa place.

Considérons par exemple l'utilisation de telnet. La victime se connecte à un serveur, s'identifie (`login: perrin`), et rentre son mot de passe (`password: ********`). Le système estime avoir authentifié l'utilisateur, et l'assaillant commence alors à envoyer des paquets forgés avec des commandes (par exemple `xhost +`). Notons que la faille tient plus, dans ce cas, au processus de contrôle d'accès (login + password à l'entrée du système) qu'au fait que telnet ne chiffre pas la communication. Cette attaque peut en effet être menée "en aveugle", c'est à dire sans être en mesure d'espionner le trafic sur le réseau. L'assaillant doit alors réussir à forger des paquets avec des numéros de séquence appropriés et tenir l'utilisateur légitime en échec (par exemple en coupant le courant d'une partie du bâtiment).

IPsec, en sécurisant la couche réseau (IP), permet de protéger la couche transport, donc TCP, contre ce type d'attaques.

Déni de Service (Denial of Service)

Le déni de service est un type d'attaque particulier dont le but premier n'est pas l'intrusion mais plutôt d'affecter la disponibilité d'un système ou la qualité d'un service.

Si un assaillant est en mesure de lancer ce type d'attaque, c'est à dire s'il est en mesure de communiquer facilement avec la machine victime et de lui envoyer des paquets forgés, alors **la victime ne peut qu'essayer de limiter les effets de l'attaque.**

En disposant de plusieurs machines, l'assaillant peut lancer une déni de service distribué. Par exemple, ses n machines établissent m connexions à un serveur web et maintiennent un trafic juste suffisant pour garder les connexions. Si toutes les connexions peuvent être occupées de cette manière, personne d'autre ne pourra se connecter. Cette technique serait très pratique pour réserver des tickets d'avions ou de trains aux dates critiques (si elle n'était pas illégale).

Les protocoles de sécurité sont malheureusement un peu plus vulnérables que les protocoles standards à cette attaque. En effet, un processus de sécurité implique des calculs, notamment, dans le pire des cas, des exponentiations (calculs de puissances d'un nombre), ce qui est coûteux en temps de traitement et affecte la disponibilité du service. Une technique classique pour combattre le déni de service consiste à obliger la personne qui se connecte à "prouver son intérêt" dans l'échange, par exemple en lui demandant de confirmer certaines valeurs ou d'effectuer des calculs simples. Ce jeu de questions-réponses est mis en oeuvre par ce qu'on appelle des "cookies". Une fois convaincu de l'intérêt et de la bonne foi de son correspondant, le serveur peut prendre le risque d'effectuer des calculs plus coûteux.

Les parties suivantes vont nous montrer dans quelles mesures IPsec permet de contrer les attaques présentées précédemment.

Authentication Header

Les documents [4], [6], [8], [9], [13], [16], [17], [23] constituent les éléments officiels de AH.

AH fournit les **services cryptographiques** suivants :

- Authentification
- Intégrité
- Non-répudiation de l'origine

La conception de AH permet de **contrer les attaques** suivantes :

- Attaque par rejeux
- Attaque par réflexion
- Détournement de connexion

Accessoirement, AH **limite les effets** des attaques :

- Déni de service

Mais toutes ces assertions demeurent théoriques ! La pratique, notamment un usage inconsidéré des options du protocole, peut réduire drastiquement ses qualités.

La **configuration manuelle des clefs d'authentification constitue l'abus le plus répandu**. En effet, dans un tel contexte, les clefs ne seront pas raffraichies régulièrement, ce qui va permettre les attaques par rejeux. Par ailleurs, si l'administrateur définit le même index de paramètre de sécurité et la même clef dans les deux sens (par exemple, s'il construit un script unique pour les deux machines), les attaques par réflexions deviennent aussi possibles.

De façon générale, **une configuration manuelle menace, sur le long terme, les propriétés cryptographiques du protocole** : les paquets rejoués ou réflèchis corrompent les propriétés d'authentification et d'intégrité. De plus, une durée de vie importante des clefs laisse aux attaquants le temps de les casser.

Le **processus d'authentification** défini dans le standard est la conséquence d'un consensus. Des modes d'authentification symétriques sont en effet recommandés, ce qui ne donne pas une preuve totale de la provenance. Cependant, ce choix a un avantage en terme de ressources de calculs requises, ce qui est une problèmatique de premier plan vis à vis des attaques de type "déni de service".

Par ailleurs, la portée de l'authentification pose un problème d'ordre sémantique : un processus d'authentification et d'intégrité doit porter sur des données dotées d'une valeur sémantique ([25], [27], [35]). L'authentification de l'adresse IP a-t'elle un sens ? Le mécanisme d'authentification ne prouve-t'il pas déjà l'origine du paquet, même compte tenu de l'utilisation de mécanismes d'authentification symétrique ?

Enfin, AH pose un **problème d'interprétation du modèle en couches** des réseaux. En effet, l'interprétation d'une option d'IP, telle que AH, est intimement dépendante de la vision que l'on a de la couche réseau :

- Si la couche réseau est unique et indivisible, une option peut affecter l'intégralité de la couche. Cette approche, qui est celle de AH, nécessite un développement concerté des options afin d'éviter les problèmes d'incompatibilités. L'avantage est que la portée efficace de l'option est élargie; c'est ainsi qu'AH prouve

l'intégrité des adresses source et destination, ainsi que des autres options qui précèdent son entête.

- Si la couche réseau est elle-même divisée en sous-couches, les options sont encapsulées les unes dans les autres, et la dernière est encapsulée par l'entête IP. Cette vision est plus flexible puisque chaque option se comporte comme une boîte noire. AH n'a pas cette vision en sous-couches de la couche réseau, et est donc incompatible avec d'autres traitements de niveau réseau. Exemple : Si AH authentifie un paquet dont les adresses vont être modifiées par un autre mécanisme (NAT...), le paquet sera rejeté à la destination, puisque la vérification d'intégrité échouera.

Dans la pratique, cela pose surtout un problème de choix. **Dans certains cas, AH ne pourra tout simplement pas être utilisé**.

D'un point de vue protocole, AH est en réalité plutôt simple, ce qui peut surprendre puisque l'on souligne souvent la complexité des protocoles IPsec. AH ne requiert que peu d'options pour fonctionner, et se décline seulement en deux modes pour chacune des deux versions d'IP. C'est plutôt la configuration des éléments qui permettent d'utiliser AH de façon optimale (en changeant de clefs régulièrement) qui est complexe.

De même pour la documentation [13] : cette dernière s'avère très claire et suffisamment précise. En revanche, aucune justification n'y est apportée en ce qui concerne les objectifs du protocole, et notamment le "pourquoi" d'une preuve d'intégrité et d'authenticité aussi large sur le paquet. En conclusion, **il n'est pas évident que les "plus" apportés par AH par rapport à ESP suffisent à justifier l'existence de ce protocole** (cela peut expliquer pourquoi AH est progressivement délaissé au profit de ESP uniquement). Par ailleurs, les choix techniques n'ont pas été justifiés non plus.

Encapsulating Security Payload

Les documents [5], [6], [7], [8], [9], [10], [15], [16], [17], [18], [19], [23] constituent les éléments officiels de ESP.

ESP fournit les **services cryptographiques** suivants :

- Authentification
- Intégrité
- Non-répudiation de l'origine
- Confidentialité

La conception de ESP permet de **contrer les attaques** suivantes :

- Attaque par rejeux
- Attaque par réflexion
- Détournement de connexion

Accessoirement, ESP **limite les effets** des attaques :

- Déni de service

ESP souffre aussi des configurations manuelles. **La complexité de l'installation manuelle d'un tunnel rend plus que probable une configuration maladroite**, avec des SPI identiques dans les deux sens, des clefs identiques dans les deux sens, etc. Dans ce cas, les attaques réflexives deviennent possibles. Et même si les associations de sécurité dans chaque sens sont suffisamment différentes, les rejeux ne peuvent être contrés. Encore une fois, une durée de vie trop importante des éléments cryptographiques peut permettre aux attaquants de déterminer les clefs utilisées. Soulignons notamment que beaucoup de protocoles de niveau transport ont des comportements prévisibles (http, ftp...), et qu'il est donc possible de constituer des couples (texte clair, texte chiffré) en analysant les caractéristiques des transactions au niveau réseau (tailles des paquets, nombre de paquets échangés, etc.); c.f. [32], [34].

Exemple : Un malveillant espionne le trafic ESP à destination d'un serveur web d'entreprise (il a réussi à déterminer qu'il s'agit d'un serveur web grâce à du "social engineering", des politiques de sécurité laxistes, des messages ICMP issus d'anciennes communications...). Il attend un certain laps de temps, puis voit arriver un premier paquet ESP depuis un client. Ce paquet est vraisemblablement un SYN TCP. La réponse du serveur sera un SYN/ACK. Le prochain paquet du client sera un ACK, et le suivant contiendra des données http prévisibles ("`GET / HTTP/1.1\r\n...`"). Diverses techniques de cryptanalyse à partir de fragments en clair existent pour exploiter ces informations. Des attaques par textes clairs choisis peuvent aussi être menées en envoyant des mails avec des documents attachés caractéristiques vers la boîte d'une personne qui les consultera à travers le tunnel.

En définitive, l'utilisation prolongée des mêmes clefs peut provoquer une dégradation irrémédiable des services cryptographiques !

L'**utilisation de l'authentification au sein d'ESP pose de nombreux problèmes.** Tout d'abord, son utilisation est facultative. Or, la confidentialité n'a pas de valeur sans authenticité ([25], [27], [29], [35]); il est en effet facile d'envoyer des données aléatoires sous l'identité de quelqu'un d'autre, en respectant les contraintes de bourrage des algorithmes de chiffrement. Ces données seront déchiffrées avant d'être transférées à la couche transport, dont la réaction ne sera pas forcément adéquate. De plus, sans authentification, des rejeux peuvent être mis en oeuvres.

Par ailleurs, quand ESP assume l'intégrité et l'authenticité du paquet, le fait-il correctement ? D'un point de vue cryptographique, ESP est faillible dans la mesure où il chiffre avant d'authentifier: ce n'est pas le sens de l'information qui est authentifié, mais un ensemble de caractères sans signification issus du chiffrement; l'étape d'authentification peut donc être validée même si la clef de chiffrement a été corrompue entre temps. [24] et [35] décrivent cela plus en détail pour le protocole SSL.

En revanche, d'un point de vue sécurité du service, être en mesure de vérifier l'authenticité avant de déchiffrer économise une opération coûteuse contre les paquets falsifiés, ce qui permet de lutter efficacement contre le déni de service. On pourrait authentifier les données de la couche transport - pour préserver le sens -, chiffrer l'ensemble (données de transport, preuve d'authenticité) - pour assurer la confidentialité -, et authentifier ce texte chiffré - pour lutter contre le déni de service. Toutes les applications ne peuvent se

permettre un tel coût de calcul, et la complexité de l'ensemble augmenterait dangeureusement.

Une alternative est d'authentifier la clef utilisée par l'étape de chiffrement. Cette alternative a été choisie dans ESP: le couple (clef d'authentification, clef de chiffrement) est indissociable, car ces deux clefs doivent être dérivées à partir d'une clef unique. L'utilisation d'une clef du couple pour le chiffrement force l'utilisation de l'autre clef du couple pour l'authentification et vice-versa. Malheureusement, cette condition doit être prise en compte par les protocoles d'échanges de clefs, et ils doivent parvenir à ce résultat d'une façon sécurisée.

Donc : **compte-tenu de la situation, ESP utilise au mieux l'authentification, mais au prix de contraintes fortes sur les protocoles d'échange de clefs**.

La portée de l'authentification est limitée à ce que ESP peut encapsuler : tout ce qui précède l'entête ESP est ignoré, ce qui garantit une bonne compatibilité avec d'autres options de la couche réseau (par exemple, IP dans IP). Cependant, ESP ne constitue pas systématiquement une bonne alternative à AH quand ce dernier ne peut être utilisé : par exemple, les mécanismes de NAT/PAT peuvent nécessiter des modifications des données au niveau transport, et rentrer en conflit avec ESP. Ainsi, si la valeur du port du protocole de transport est modifiée durant le transit, ESP échouera lors de la vérification d'intégrité et/ou lors du déchiffrement du paquet.

A l'instar de AH, ESP est un protocole assez simple, son utilisation nécessitant cependant un configuration très complexe. Par ailleurs, la documentation de ESP, si elle est calquée sur celle de AH, laisse la porte ouverte à un nombre important d'interprétations divergentes, notamment en ce qui concerne les documents décrivant l'utilisation des algorithmes de chiffrement : les tailles de clefs, les modes de "chaining" constituent des pôles d'interrogations fréquentes; les vecteurs d'initialisation pour le chiffrement par blocs font l'objet de propositions qui sont très controversées : [26], [29], [32].

Internet Key Exchange

Les protocoles d'échanges de clefs constituent le point de convergence de processus complexes et nécessitent d'effectuer des choix difficiles. Afin de mieux comprendre l'ensemble, il est malheureusement nécessaire de considérer, au moins dans un premier temps, chaque composante séparément à l'image d'une boîte noire. Nous serons donc amenés à aborder dans cette partie ISAKMP, IKE, les certificats...

ISAKMP

Les documents [20], [21] constituent les éléments officiels de ISAKMP. [11] apporte un complément d'informations.

ISAKMP, Internet Security Association and Key Management Protocol, permet la négociation de paramètres de sécurité, comme les services cryptographiques, les algorithmes... Il établit ainsi les associations de sécurité entre les machines.

Ce protocole est très complexe, et la granularité trop fine autorisée dans les propositions échangées ne fait pas bon ménage avec la grande quantité d'options cryptographiques disponibles. Il en résulte une documentation entâchée de nombreuses erreurs, d'explications manquantes et de contradictions internes (voir [25]). L'ensemble est particulièrement ambigüe, notamment en ce qui concerne le positionnement des attributs, ou les raisons pour lesquelles les "payloads" portent la description du "payload" suivant (champ "next payload") au lieu de leur propre description; ce dernier point peut sembler anodin, mais est très déconcertant à l'usage et peu consistant dans certains échanges. Dans cet état, des recommandations techniques et des précisions sur les traitements à effectuer, auraient été les bienvenues.

De plus, certains aspects cryptographiques étaient mal maîtrisés par le Working Group à l'époque de la rédaction, ce qui aboutit à des notations confuses, notamment en ce qui concerne les MAC, qui constituent des preuves d'authenticité, et les HASH, qui constituent des preuves d'intégrité; [8] tente de clarifier ces points. Cette absence de maîtrise explique la présence d'échanges dont l'utilité n'est pas significative ("Auth Only Exchange"), ainsi qu'une utilisation inadaptée des "cookies". Des techniques de sécurité sont employées, mais de façon non concertée. Un autre exemple concerne la génération des SPI (index de paramètres de sécurité : voir plus haut), qui doit être aléatoire d'après ISAKMP [21], alors que le document "Internet Security Architecture" [11] insiste bien sur le fait que la gestion du SPI est une problèmatique locale. Des SPI aléatoires n'apportent rien en terme de sécurité et peuvent en compliquer la gestion (par exemple, pour la construction d'un arbre de recherche des SPI). Il pourrait être divertissant (et utile pour un "cracker") d'analyser l'évolution des SPI sur une machine qui les génère aléatoirement afin de déterminer le "fingerprint" du système d'exploitation ou les caractéristiques (périodes, "seed"...) du générateur de nombres aléatoires.

Enfin, si ISAKMP supporte l'échange de certificats en précisant une autorité de certification, il laisse par contre de côté la gestion de listes d'autorités de certifications. Or, en réalité, il y a une probabilité non négligeable pour que les machines soient certifiées par plusieurs autorités, et ISAKMP ne permet pas de trouver facilement le "set" commun d'autorités, et encore moins de faire un choix intelligent dans ce "set".

IKE

IKE ne constitue que la partie "échange de clefs" de l'ensemble. Il ne s'intéresse pas aux aspects négociations de la communication. Le document [22] décrit en profondeur ce protocole.

IKE rajoute beaucoup à la complexité d'ISAKMP. IKE a trois types de défauts:

- Les "vides" de la documentation.
- Les incohérences de la documentation.
- Les failles du protocole.

Dans les "vides" de la documentation, on "ne trouve pas" :

- Des champs et des valeurs qui devraient intervenir

dans le calcul des "hash" (voir les failles plus bas).

- Les propriétés requises pour les "HMAC" et "PRF" utilisés. Les passer sous silence augmente les risques de voir apparaître dans le futur des combinaisons affaiblies de fonctions. Il serait pourtant possible de définir des interfaces entre IKE et ces fonctions [25].

- Les justifications de l'algorithme de dérivation des clefs. En l'absence de justifications, les choix paraissent aléatoires et peu sûrs. Cependant, ils respectent les contraintes énoncées par ESP, bien que ces dernières ne soient pas rappelées dans le document.

Les incohérences de la documentation sont notamment dues aux interdépendances des documents [11], [12], [14], [20], [21] et [22]. Il y a cependant aussi des incohérences internes : la représentation des éléments d'un groupe est différente selon qu'ils sont utilisés pour un échange de clefs (payload "KE") ou pour la définition d'un groupe ("New Group Mode").

Les failles du protocole permettent les attaques par réflexion, d'affecter les paramètres des associations de sécurité et de faire du copier-coller sur des éléments authentifiés. Ces failles sont présentées dans [25]; voyons cela plus en détail :

- Une attaque par réflexion est possible dans le cas où l'authentification repose sur un secret partagé. Dans ce scénario, l'attaquant renvoie HASH_I (la valeur d'authentification de l'initiateur) avec la valeur publique de Diffie-Hellman de l'initiateur. Ceci est rendu possible par le fait que HASH_I et HASH_R sont calculés de la même manière. Cette faille est décrite plus précisément dans [25], paragraphe 5.3.

- Une attaque par altération est aussi possible : quand un des participants fait des propositions cryptographiques, la proposition choisie dans la réponse n'est pas incluse dans la preuve d'intégrité, ce qui permet à un attaquant de forcer le choix d'une proposition dont les propriétés sont plus faibles. Cette attaque est aussi décrite dans [25] au paragraphe 5.3.

- D'un point de vue général, les valeurs d'intégrité et d'authentification n'incluent pas suffisamment d'éléments du contexte, comme par exemple le numéro du message, son sens (Initiateur -> Répondeur...), etc. Il est donc possible d'exploiter ces valeurs dans des messages forgés.

En conclusion, IKE est trop obscur et complexe pour pouvoir être implémenté correctement. D'ailleurs, la majorité du trafic de la liste de bugs de FreeSwan, l'implémentation linux de IPsec, concerne Pluto, le "daemon" IKE de FreeSwan.

Son Of Ike

Le Working Group IPsec, agacé par les défauts, et surtout par l'échec de IKE, a décidé de lui trouver un successeur. Le document [36], qui sera sans doutes remis à jour prochainement, précise ce que doit être un protocole d'échange de clefs pour pouvoir succéder à IKE.

Cette fois, on a donc bien une présentation des objectifs, une description des fonctionnalités, etc. Malheureusement, les candidatures sont multiples, et les fonctionnalités attendues ne font pas l'unanimité. Notamment :

- Le support de l'anonymat.
- L'échange de données politiques, comme la négociation d'AS ou la description de politiques de la SPD.

Enfin, la recommandation, qui se voulait décrire un set minimal de fonctionnalités pour un protocole léger, tend à s'épaissir avec le temps.

Les Certificats

De nombreuses questions soulevées par la mise en place d'une architecture de sécurité Internet ne pourront trouver de solutions sans une vue plus précise de ce que sera le développement et l'utilisation des certificats. Il a été vu que les mécanismes de IKE pour supporter des certificats issus de différentes autorités sont insuffisants. Par ailleurs, il n'est pas évident que cela soit nécessaire : les utilisations actuelles d'IPsec impliquent dans la majorité des cas une seule autorité, et il ne semble pas y avoir un grand désir d'interopérabilité à ce niveau.

Les certificats posent aussi des problèmes vis à vis de la valeur qu'on leur donne : identifient-ils une personne, une machine ou un utilisateur ? Que faire quand plusieurs certificats sont présentés et peuvent être requis pour des accès différents ?

Tant que ces questions ne seront pas résolues, ce qui peut nécessiter une expérience de déploiement à grande échelle, le Working Group est condamné à travailler sur des suppositions. Un exemple récurrent est le support de l'anonymat : on peut considérer que c'est au protocole d'échange de clefs de protéger l'identité, ou à l'autorité de certification de générer des certificats dont les identités n'ont de sens que pour elle.

Limitations de l'Architecture

Certaines limitations n'apparaissent pas lors d'une étude en isolation des protocoles. On a jugé utile d'en mentionner quelques-unes ici.

Tout d'abord, IPsec peut rentrer en conflit avec des mécanismes d'IP, comme la fragmentation ([25]); [11] décrit à ce sujet tout un ensemble de scénarios en appendice B, expliquant où la fragmentation peut avoir lieu, et où les paquets doivent être reconstruits. Le sujet est en réalité très complexe, et les boitiers autonomes qui chiffrent les paquets en transit sont au coeur de cette problèmatique. Ensuite, [11], le document de référence pour IPsec, propose des mécanismes dont l'emploi n'est pas (actuellement) réaliste. Notamment :

- Si les messages ICMP ne sont pas authentifiés, il est recommandé de ne pas les prendre en compte. Cela nécessite une infrastructure à clefs publiques pour vérifier les messages issus des routeurs sur les chemins empruntés par les paquets [25]. Par ailleurs, cela ne suffirait pas à prouver qu'un routeur est "de bonne foi"; ce dernier devra inclure des éléments du paquet fautif dans son message, ou le source-routing devra être actif.

- Le document présente une technique dangereuse

pour retrouver une règle de politique de sécurité à partir d'une association de sécurité : il s'agit des SPD "back-pointers". Le risque de cette méthode est de permettre une rupture de séquence dans le parcours des politiques de sécurité, provoquant un comportement inconsistant, source de failles de sécurité.

Par ailleurs, de nombreux détails de l'architecture restent inconnus :

- Malgré les propositions de Cisco, il n'existe toujours pas de protocole standardisé pour découvrir les passerelles de sécurité, ce qui est pourtant nécessaire pour établir des associations de sécurité entre deux sites sans informations a-priori.

- Les durées de vie des associations de sécurité peuvent être définies en octets et/ou en secondes, mais aucun lien n'est encore fait avec la durée de vie des paramètres cryptographiques engagés. Pour certains "ciphers", la durée de vie de la clef peut être inférieure à celle de l'association de sécurité.

- Quid des intéractions avec les firewalls ou les systèmes de détections d'intrusions ? Lorsque le paquet est chiffré, tout système exploitant les "pattern-matching" voit son efficacité réduite à néant.

- Les DNS font encore l'objet de nombreuses attaques. Quel nouveau paradigme pourra équilibrer les besoins de sécurité et de performance de la propagation des espaces de nommage ? Comment, à l'image de la révocation de certificats, révoquer un nom corrompu dans un cache DNS ?

- Les politiques de sécurité ne présentent pas encore des critères de sélections adaptés à certains protocoles, comme SCTP, et les points de vues s'affrontent sur le sujet de la sélection multiple.

Enfin, IPsec interopère avec plus ou moins de bonheur avec d'autres protocoles, notamment le NAT et DHCP, qui feront l'objet d'un prochain rapport.

CONCLUSION

L'architecture IPsec est complexe. Souvent, les standards sont mal documentés ou font des assomptions gratuites. Cela se répercute inévitablement sur les implémentations. De plus, la finesse de configuration met le système hors de portée du néophyte et empêche une détermination du niveau de sécurité réel.

Le déploiement de l'architecture est lent, et le demeurera sans une vision plus précise de l'intégration des infrastructures de certification, des politiques de sécurité et des processus d'échanges de clefs.

Cependant, il ne faudrait pas, sous prétexte que nous venons de montrer du doigt les lacunes d'IPsec, juger trop durement l'architecture. IPsec est ancien, et cependant, de bonnes décisions ont été prises dès le départ (concernant les SPI, l'anti-rejeu...). De plus, IPsec évolue, certes lentement, mais sûrement : tous les travaux sur "Son Of Ike" ne vont pas être abandonnés du jour au lendemain. Enfin, l'utilisation d'IPsec pour la construction de VPNs ou pour sécuriser des accès distants est toute à fait réaliste; il s'agit d'ailleurs de son marché actuel.

Si IPsec est bien loin de constituer une solution idéale, il supporte par contre plutôt bien la comparaison avec les autres solutions de sécurité pour Internet. Ecarter IPsec et miser sur un protocole (ssl, tls...) de certes plus souple, plus facile à maîtriser et à configurer, mais limité à une utilisation particulière, est un risque qui nécessite une évaluation en profondeur.

RÉFÉRENCES

[1] DRAFT REVISED IP SECURITY OPTION
Date: 1er Janvier 1988
Status: RFC 1038
Auteur: M. St. Johns
(Source: http://www.ietf.org/rfc/rfc1038.txt)

[2] REPORT OF IAB WORKSHOP ON SECURITY IN THE INTERNET ARCHITECTURE
Date: 8-10 Février 1994
Status: RFC 1636
Auteur: R. Braden, D. Clark, S. Crocker, C. Huitema
(Source: http://www.ietf.org/rfc/rfc1636.txt)

[3] SECURITY ARCHITECTURE FOR THE NTER-INET PROTOCOL
Date: Août 1995
Status: RFC 1825
Auteur: R. Atkinson
(Source: http://www.ietf.org/rfc/rfc1825.txt)

[4] IP AUTHENTICATION HEADER
Date: Août 1995
Status: RFC 1826
Auteur: R. Atkinson
(Source: http://www.ietf.org/rfc/rfc1826.txt)

[5] IP ENCAPSULATING SECURITY PAYLOAD (ESP)
Date: Août 1995
Status: RFC 1827
Auteur: R. Atkinson
(Source: http://www.ietf.org/rfc/rfc1827.txt)

[6] IP AUTHENTICATION USING KEYED MD5
Date: Août 1995
Status: RFC 1828
Auteur: P. Metzger, W. Simpson
(Source: http://www.ietf.org/rfc/rfc1828.txt)

[7] THE ESP DES-CBC TRANSFORM
Date: Août 1995
Status: RFC 1829
Auteur: P. Karn, P. Metzger, W. Simpson
(Source: http://www.ietf.org/rfc/rfc1829.txt)

[8] HMAC: KEYED-HASHING FOR MESSAGE AUTHENTICATION
Date: Février 1997
Status: RFC 2104
Auteur: H. Krawczyk, M. Bellare, R. Canetti
(Source: http://www.ietf.org/rfc/rfc2104.txt)

[9] HMAC-MD5 IP AUTHENTICATION WITH REPLAY PREVENTION
Date: Février 1997
Status: RFC 2085
Auteur: M. Oehler, R. Glenn
(Source: http://www.ietf.org/rfc/rfc2085.txt)

[10] THE NULL ENCRYPTION ALGORITHM AND ITS USE WITH IPSEC
Date: Novembre 1998
Status: RFC 2410
Auteur: R. Glenn, S. Kent
(Source: http://www.ietf.org/rfc/rfc2410.txt)

[11] SECURITY ARCHITECTURE FOR THE NTER-INET PROTOCOL
Date: Novembre 1998
Status: RFC 2401
Auteur: S. Kent, R.Atkinson
(Source: http://www.ietf.org/rfc/rfc2401.txt)

[12] IP SECURITY DOCUMENT ROADMAP
Date: Novembre 1998
Status: RFC 2411
Auteur: R. Thayer, N. Doraswamy, R. Glenn
(Source: http://www.ietf.org/rfc/rfc2411.txt)

[13] IP AUTHENTICATION HEADER
Date: Novembre 1998
Status: RFC 2402
Auteur: S. Kent, R. Atkinson
(Source: http://www.ietf.org/rfc/rfc2402.txt)

[14] THE OAKLEY KEY DETERMINATION RO-PTOCOL
Date: Novembre 1998
Status: RFC 2412
Auteur: H. Orman
(Source: http://www.ietf.org/rfc/rfc2412.txt)

[15] THE ESP CBC-MODE CIPHER ALGORITHMS
Date: Novembre 1998
Status: RFC 2451
Auteur: R. Pereira, R. Adams
(Source: http://www.ietf.org/rfc/rfc2451.txt)

[16] THE USE OF HMAC-MD5-96 WITHIN ESP AND AH
Date: Novembre 1998
Status: RFC 2403
Auteur: C. Madson, R. Glenn
(Source: http://www.ietf.org/rfc/rfc2403.txt)

[17] THE USE OF HMAC-SHA-1-96 WITHIN ESP AND AH
Date: Novembre 1998
Status: RFC 2404
Auteur: C. Madson, R. Glenn
(Source: http://www.ietf.org/rfc/rfc2404.txt)

[18] THE ESP DES-CBC CIPHER ALGORITHM WITH EXPLICIT IV
Date: Novembre 1998
Status: RFC 2405
Auteur: C. Madson, N. Doraswamy
(Source: http://www.ietf.org/rfc/rfc2405.txt)

[19] IP ENCAPSULATING SECURITY PAYLOAD (ESP)
Date: Novembre 1998
Status: RFC 2406
Auteur: S. Kent, R. Atkinson
(Source: http://www.ietf.org/rfc/rfc2406.txt)

[20] THE INTERNET IP SECURITY DOMAIN OF INTERPRETATION FOR ISAKMP
Date: Novembre 1998
Status: RFC 2407
Auteur: D. Piper
(Source: http://www.ietf.org/rfc/rfc2407.txt)

[21] INTERNET SECURITY ASSOCIATION AND KEY MANAGEMENT PROTOCOL (ISAKMP)

Date: Novembre 1998
Status: RFC 2408
Auteur: D. Maughan, M. Schertler, M. Schneider, J. Turner
(Source: http://www.ietf.org/rfc/rfc2408.txt)

[22] THE INTERNET KEY EXCHANGE (IKE)
Date: Novembre 1998
Status: RFC 2409
Auteur: D. Harkins, D. Carrel
(Source: http://www.ietf.org/rfc/rfc2409.txt)

[23] THE USE OF HMAC-RIPEMD-160-96 WITHIN ESP AND AH
Date: Juin 2000
Status: RFC 2857
Auteur: A. Keromytis, N. Provos
(Source: http://www.ietf.org/rfc/rfc2857.txt)

[24] THE ORDER OF ENCRYPTION AND AUTHENTICATION FOR PROTECTING COMMUNICATIONS
Date: 6 Juin 2001
Status: Publication Crypto'2001
Auteur: H. Krawczyk
(Source: http://eprint.iacr.org/2001/045.ps)

[25] A CRYPTOGRAPHIC EVALUATION OF IPSEC
Status: Publication
Auteur: N. Ferguson, B. Schneier
(Source: http://www.counterpane.com/ipsec.pdf)

[26] ATTACKING PREDICTABLE IPSEC ESP NI-ITIALISATION VECTORS
Date: Mars 2002
Status: Publication
Auteur: A. Nuopponen, S. Vaarala
(Source: http://www.hut.fi/~svaarala/espiv.pdf)

[27] CRYPTOGRAPHIE APPLIQUÉE
Date: 1996
Status: Ouvrage
Auteur: B. Schneier
(Source: ISBN: 2-84180-036-9)

[28] SKEME: A VERSATILE SECURE KEY EXCHANGE MECHANISM FOR INTERNET
Date: 30 Novembre 1995
Status: Publication IEEE
Auteur: H. Krawczyk
(Source: http://www.isoc.org/conferences/ndss96/krawczyk.ps)

[29] CRYPTOGRAPHIC MODES OF OPERATION FOR THE INTERNET
Date: Août 2001
Status: Publication NIST
Auteur: S.M. Bellovin, M. Blaze
(Source: http://www.research.att.com/~smb/papers/internet-modes.ps)

[30] NETWORK ADDRESS TRANSLATORS: EFFECTS ON SECURITY PROTOCOLS...
Date: Novembre 2000
Status: Publication IEEE
Auteur: S.P. Shieh, F.S. Ho, Y.L. Huang, J.N. Luo

[31] IPSEC: SECURING VPNS
Date: 2001
Status: Ouvrage
Auteur: C.R. Davis
(Source: ISBN: 0-07-212757-0)

[32] PROBLEM AREAS FOR THE IP SECURITY PROTOCOLS
Date: 22 Juillet 1996
Status: Publication Usenix UNIX Security Symposium
Auteur: S.M. Bellovin
(Source: http://www.research.att.com/~smb/papers/badesp.ps)

[33] CRYPTOGRAPHY AND THE INTERNET
Date: Août 1998
Status: Publication Crypto'1998
Auteur: S.M. Bellovin
(Source: http://www.research.att.com/~smb/papers/inet-crypto.ps)

[34] PROBABLE PLAINTEXT CRYPTANALYSIS OF THE IP SECURITY PROTOCOLS
Date: 1997
Status: Publication IEEE / SNDSS
Auteur: S.M. Bellovin
(Source: http://www.research.att.com/~smb/papers/probtxt.ps)

[35] ANALYSIS OF THE SSL 3.0 PROTOCOL
Date: 1997
Status: Publication Usenix UNIX Security Symposium
Auteur: B. Schneier, D. Wagner
(Source: http://www.counterpane.com/ssl-revised.pdf)

[36] PROTOCOL REQUIREMENTS FOR SON-OF-IKE
Date: Novembre 2001
Status: draft-ietf-ipsec-son-of-ike-protocol-reqts-00.txt
Auteur: C. Madson
(Source: http://www.ietf.org/internet-drafts/draft-ietf-ipsec-sonofike-rqts-00.txt)