ANALYSE DE L'IMPACT DE LA MISE EN OEUVRE D'IPSEC DANS LES ARCHITECTURES DE COMMUNICATION

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

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

SOMMAIRE

INTRODUCTION

L'IETF a commencé à acquérir une certaine inertie. L'âge des pionniers d'Internet est terminé : de nombreux protocoles ont maintenant envahi les noeuds et les feuilles du réseau. Certains de ces protocoles fonctionnent dans des dispositifs spécialisés (routeurs, pare-feux, etc.) qu'il est hors de question de remplacer, et d'autres sont intégrés au niveau des stations, mais demeurent indéracinables car leur fonctionnement est indispensable au réseau (ARP notamment). L'objet de ce document est la présentation des incompatibilités entre ces mécanismes et IPsec, incompatibilités qui sont sources de nombreux retards de déploiement pour la sécurité de l'Internet.

Au fil de ce document, nous serons amenés à examiner successivement les interactions entre IPsec et les caractéristiques propres d'IP, entre IPsec et les protocoles de transport, entre IPsec et les systèmes de résolutions d'adresses, entre IPsec et les translateurs d'adresses et de ports.

IPSEC ET IP

Cette partie traite des interactions entre IPsec et IP. Il est en effet important de savoir dans quelles mesures IPsec affecte le comportement d'IP, puisque de nombreux protocoles ou traitements effectués dans le réseau ont été conçus pour tirer parti des caractéristiques ordinaires d'IP.

Données Monotones

De nombreux protocoles envoient des données répétitives d'un paquet à l'autre, notamment les systèmes de diffusion d'informations (radio, webcam, mais aussi `Router Advertisement`, etc.). La distance de Hamming (nombre de bits différents) entre les paquets est alors réduite. Dans certains cas, cette distance est nulle (i.e. les données avant chiffrement des paquets sont identiques). Cela peut être fréquent avec des protocoles se basant sur UDP ou directement sur IP.

La similitude entre paquets constitue une menace importante pour la propriété de confidentialité assurée par ESP. En effet, avec des chiffrements classiques dans Z/pZ, ces similitudes sont exploitables pour casser la clef de session plus rapidement ; et si le temps de calcul demeure rédhibitoire, c'est-à-dire que la communication ne pourra vraisemblablement pas être déchiffrée pendant la durée de vie de la session, la pérennité de la confidentialité des informations est cependant compromise. Ainsi, si on considère deux paquets pour lesquels les données sont identiques ou peu différentes, sachant que les valeurs des vecteurs d'initialisation sont explicites et que les règles de construction du padding sont connues, les seules données inconnues pour l'attaquant sont les textes en clair et la clef de session. La distance entre messages étant faible, une analyse différentielle est facilitée (cf. §2.2 p.16 de [ADP99]). En mode CBC, une distance de Hamming faible entre vecteurs d'initialisation facilite aussi une telle analyse (cf. §1.2 de [KMS95], et §3 de [MD98]).

Ce problème de similitudes entre données est mentionné ici car, à défaut de constituer un problème d'interaction au sens de l'ingénierie des protocoles, il s'agit bien d'une interférence entre deux processus - les données de couche supérieure et les traitements d'IPsec - au sens de la théorie de l'information. Les solutions sont donc à rechercher dans cette théorie.

Il s'agit principalement d'augmenter la distance de Hamming entre paquets. Pour ce faire, plusieurs solutions (classiques) se présentent :

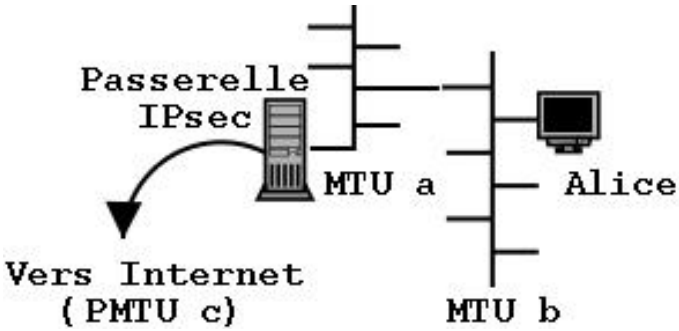
- Utiliser CBC entre les paquets, i.e. sans donner de façon explicite la valeur du vecteur d'initialisation dans chaque paquet. Seul le premier paquet contient alors un vecteur d'initialisation, et le vecteur d'initialisation utilisé pour le chiffrement suivant est la dernière valeur calculée pour le chiffrement en cours. Cette pratique est exclue sur IP, puisque les probabilités de pertes de paquets ne sont pas négligeables. De plus, si l'attaquant est capable d'intercepter un paquet, il est souvent en mesure de les intercepter tous, ce qui rend cette technique inutile.
- Effectuer un brouillage des données avant chiffrement, à l'aide d'une clef qui varierait par paquet et qui serait concaténée aux données brouillées avant chiffrement. Actuellement, aucun travail n'envisage d'ajouter une nouvelle couche de complexité comme celle-ci.
- La compression de données peut avoir pour effet secondaire d'augmenter la distance de Hamming entre deux clairs présentant de petites différences. Cette technique se prête bien au contexte d'IPsec et plus généralement de IP, de par les gains de performances qu'elle apporte et son intégration dans la gestion des SA d'IPsec (intégration dans le DOI; cf. §3.1 de [KA98-1] et §3.4.5 de [DP98]).
- Certains algorithmes de chiffrement ont la capacité de créer des chiffres différents pour des clairs identiques, avec une clef de session identique. Ils brouillent ainsi les similitudes entre messages. Pour cela, des valeurs aléatoires, des compteurs ou des séries de nombres interviennent dans le processus de chiffrement. Lors du déchiffrement, ces valeurs aléatoires se simplifient dans les calculs (elles se divisent ou s'annulent d'elles-mêmes).

IPsec et la Fragmentation

IPsec soutient plusieurs positions face à la fragmentation :

- En mode transport, IPsec travaille uniquement sur des paquets (cf. §15.2.B.2 de [KA98-1], ainsi que [KA98-2] et [KA98-3]).
- En mode tunnel, IPsec peut encapsuler des fragments.

Ce comportement induit plusieurs problématiques; considérons le schéma suivant :



Dans ce schéma, on imagine que Alice communique avec une machine quelconque située sur Internet. La machine située avant l'accès à Internet est la passerelle de sécurité IPsec. Comme les paquets émis par Alice traversent deux réseaux de MTU différentes avant d'arriver à la passerelle, de la fragmentation peut advenir. Le routeur à l'interface des deux réseaux peut en effet prendre la décision de fragmenter ou renvoyer un message `ICMP_PMTU`. Si un réseau radio et/ou un réseau ad-hoc interviennent dans les réseaux a ou b, la probabilité de fragmentation est élevée (l'envoi du message `ICMP_PMTU` est soumis à d'autres problématiques : voir plus

bas).

En mode tunnel, cela ne pose aucun problème : IPsec encapsule les paquets IP indépendamment du fait qu'ils soient des fragments ou non.

En mode transport, le réassemblage des paquets est nécessaire (cf. §3.1 de [KA98-3] et de [KA98-1]) avant d'appliquer la protection. Cela pose un problème de performance et la rentabilité est tout de suite remise en cause : le paquet sera sans doute découpé à nouveau en fragments pour pouvoir être envoyé sur Internet (En général, le PMTU sur un chemin en dehors du réseau est inférieur à celui du réseau).

Les considérations précédentes ont mis en exergue des limitations qui sont tout à fait acceptables pour les passerelles de sécurité : mode tunnel obligatoire. Par ailleurs, si une fragmentation est requise postérieurement au traitement opéré par IPsec, celle-ci se produit dans le contexte des mécanismes classiques d'IP et demeure totalement transparente pour IPsec (un réassemblage a lieu à l'arrivée avant vérification par IPsec).

En revanche, ces limitations constituent un inconvénient majeur pour les dispositifs BITW (Bump In The Wire) ou BITS (Bump In The Stack). Si ces équipements travaillent en mode transport, des pertes de performances sont à craindre, du fait du réassemblage et de la refragmentation des paquets. S'ils travaillent en mode tunnel, une adresse IP doit leur être allouée - ce qui réduit la « transparence » du dispositif -, et ils doivent informer leurs clients que le PMTU est pour eux inférieur à celui du lien (puisque le tunnel rajoutera de l'overhead), ou alors ils doivent fragmenter en sortie du tunnel.

A la réception, que ce soit en mode transport ou en mode tunnel, un réassemblage est nécessaire en cas de fragmentation entre les systèmes mettant en jeu IPsec. Or, IPv4 n'impose absolument pas de désactiver le drapeau `MORE_FRAG` et de rendre nulle la valeur du champ `OFFSET` à l'issue du traitement de réassemblage (voir §3.4.1 de [KA98-2] et [KA98-3]). Cela implique la modification de toute pile IPv4 sur laquelle on désire implémenter IPsec, de façon à ce que ces champs aient des valeurs avant le traitement de la sécurité.

Par ailleurs, de par le fait que de nombreux routeurs sur Internet prennent d'eux-mêmes la décision de modifier le bit `DF` (Don't Fragment), AH ne protège pas ce champ, ni le `Fragmentation_Extension_Header` dans le cadre d'IPv6.

La perte de fragments signifie souvent la perte des données protégées. Dans les cas où seules l'authentification et l'intégrité sont assurées (AH et ESP sans chiffrement), une analyse des fragments récupérés pourrait fournir des renseignements exploitables par le destinataire (dans les cas graves où l'information est vitale). En revanche, quand du chiffrement est mis en place, certains fragments sont critiques afin de retrouver le contexte de sécurité (clefs, algorithmes) à appliquer pour analyser les fragments qui ont pu arriver jusqu'à destination.

Un attaquant peut exploiter les mécanismes de fragmentation/réassemblage pour provoquer des dénis de service et empêcher ainsi la mise en place de nouvelles associations de sécurité. Dans toute implémentation d'IP, des ressources (mémoire surtout) sont réquisitionnées afin de conserver les fragments dans des buffers. IPsec ne fait pas exception, et si un attaquant inonde une passerelle de sécurité avec des fragments, ce type de déni de service peut advenir. En effet, déterminer si un fragment provient d'un homologue de confiance n'est pas évident : les informations peuvent être incomplètes, ou l'association de sécurité peut être en cours de création. Notamment, IKE, SIGMA, JFK et IKEv2, utilisent UDP et envoient des messages de taille relativement importante (supérieure à 500 octets, taille minimale à partir de laquelle une fragmentation est légale), et sont donc particulièrement sensibles à cette attaque : les fragmentations « légitimes » peuvent arriver, et les implémentations sont donc obligées de conserver l'état de l'échange; voir §2.4 p 5 de [H3KP02-1], §3.1 de [Hof02].

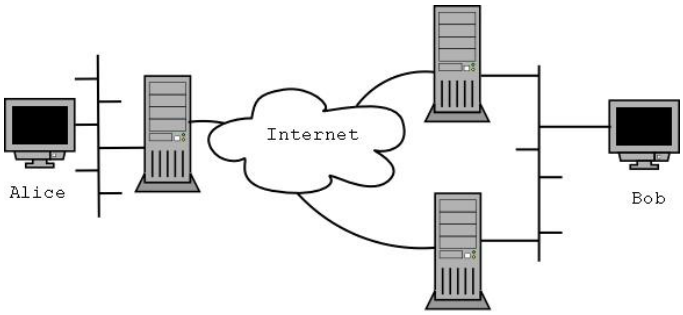
Une solution est décrite §2.6 de [H3KP02-2] : si la probabilité d'être sous le feu d'une attaque est élevée, un mécanisme doit pouvoir informer le système de réassemblage de n'accepter que les fragments UDP issus d'adresses pour lesquelles un cookie valide a été reconnu. Cela implique que les messages d'initialisation (`IKE_SA_init`) soient d'une taille inférieure à 500 octets (i.e. ils ne peuvent être fragmentés) afin de pouvoir servir un cookie aux nouveaux arrivants.

IPsec et le Routage

Les modules IPsec BITS (``Bump In The Stack'') occupent une position délicate. Ces implémentations d'IPsec viennent s'intercaler entre les pilotes de carte réseau et la couche IP, dans des systèmes pour lesquels les sources de la couche IP ne sont pas disponibles ou librement modifiables. En mode transport, l'utilisation de tels dispositifs ne pose aucun problème de routage, de même en mode tunnel si l'extrémité du tunnel correspond à la destination des paquets. En revanche, toujours en mode tunnel, si la destination finale n'est pas l'extrémité du tunnel ou si l'hôte effectue du multi-homing, l'emplacement de ces modules dans la pile peut rendre difficile les décisions d'acheminement : comment ces dispositifs pourraient-ils prendre une décision de routage et déterminer la bonne interface et la bonne passerelle pour acheminer le paquet en sortie du tunnel ? Ce rôle revient plutôt à la couche IP, mais cette dernière n'est pas sensée savoir qu'un traitement de type tunnel a eu lieu, puisqu'il s'agit d'un module BITS. Etablir un diagnostic peut alors être difficile, car si la destination finale est la machine locale ou si le couple (interface, next hop) requis pour l'acheminement correspond à celui par défaut, tout fonctionnera correctement ! En §15.2.B.2.b de [KA98-1], deux « moindres maux » pour faire face à ce problème sont présentés :

- Encapsuler les paquets en sortie du tunnel dans de nouveaux paquets, sans protection cette fois, et les renvoyer à la couche IP pour que celle-ci les soumette à nouveau au traitement d'IPsec, qui les laissera transiter (puisque les paquets ne convoient plus d'informations impliquant des traitements cryptographiques). Ce mécanisme semble très compliqué et peu performant. De plus il nécessite l'autorisation de l'acheminement des paquets sans protection issus de la machine locale et à destination de l'hôte final.
- Décapsuler les données du paquet ``tunnelé'' et les envoyer à la couche IP de façon à ce qu'elle crée elle-même un nouveau paquet vers la bonne destination. Il est à craindre que des paquets protégés par des mécanismes comme AH ne survivent pas à ce traitement (par exemple dans le cas où le tunnel acheminait des paquets protégés par AH, la reconstruction du paquet en sortie du tunnel risque d'introduire des erreurs pour la preuve d'intégrité du paquet IP).

Un autre problème d'acheminement des paquets se pose en mode tunnel...



Sur le schéma précédent, quand Alice désire envoyer un paquet à Bob, la passerelle de sécurité d'Alice analyse la destination du message. La politique de sécurité impose la création d'un SA pour toute communication avec le domaine d'où est issu Bob... Mais comment la passerelle détermine-t'elle les adresses IP des passerelles du domaine de Bob, et comment choisit-elle avec quelle passerelle établir un SA ? Il n'existe pas, actuellement, de mécanisme permettant la découverte de passerelles IPsec ou de résoudre la situation de multihoming. Il est nécessaire que la passerelle d'Alice connaisse celle de Bob. Le draft [BKRS02] (août 2002) énonce en §3.2.2 qu'un protocole permettant la découverte des passerelles de sécurité est une nécessité pour le groupe IPSP (``IP Security Policy''). Diverses propositions apparaissent épisodiquement (elles ont maintenant expiré), mais aucune ne semble s'imposer. L'utilisation du ``Remote Discovery Protocol'' (RDP) de Cisco - à ne pas confondre avec ``Cisco Discovery Protocol'' (CDP), qui fonctionne uniquement au niveau 2 - avait été présentée lors du 52ème congrès IETF, sans grand succès.

IPsec et le Filtrage

D'un point de vue général, les mécanismes de confidentialité cohabitent mal avec ceux de monitoring ou de filtrage. Les systèmes de détection d'intrusion (IDS) voient ainsi leur utilité réduite. L'information pertinente qu'ils peuvent renvoyer sont les éléments de protocoles (adresses, options...) et les caractéristiques des échanges (volume total de communication, direction de la communication). Ces caractéristiques mesurables peuvent cependant être faussées, par exemple via l'utilisation du ``Traffic Flow Confidentiality'' (TFC), qui consiste à faire du bourrage dans les paquets avec des octets inutiles avant chiffrement. L'utilisation conjointe d'IPsec et de systèmes de monitoring nécessite une évaluation des besoins (confidentialité nécessaire ?) et une analyse du réseau (où placer les machines IPsec et où placer les sondes ?).

Les mécanismes actifs effectuant un filtrage selon des caractéristiques des paquets sont aussi tenus en échec. Les pare-feux sont bien sûr les premiers acteurs concernés; ils nécessitent l'autorisation des protocoles AH et ESP, et l'ouverture du port 500 pour IKE, tout cela sans pouvoir déterminer le contenu utile du trafic ! Cela nécessite une certaine confiance; il est donc plus judicieux de combiner le firewall et la passerelle de sécurité, ou de le placer après la sortie du tunnel (une fois la tâche d'IPsec accomplie). Il existe cependant d'autres mécanismes actifs dont on peut ignorer l'existence jusqu'au moment où l'incident survient : ponts (bridges) et autres systèmes assurant la transition entre plusieurs réseaux gérés par des opérateurs différents. Notamment, dans le cadre d'un accès à Internet, les messages ICMP `Fragmentation Needed` sont souvent bloqués par les FAI, et il est alors nécessaire de fixer le `Maximum Segment Size` (MSS) de TCP par rapport à la MTU. De la même manière, une adaptation de la taille des paquets est nécessaire quand un routeur fait la transition entre une interface IP/802.3 et une interface IP/PPPOE/802.3. Ces traitements peuvent affecter les preuves d'intégrité d'IPsec (cf. [Bou02]).

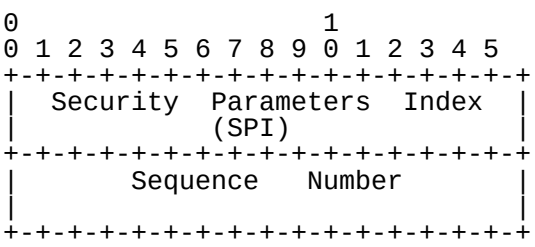
On pourrait objecter à l'utilisation d'un firewall les capacités de filtrage d'IPsec, plus précisemment les SAD ``Traffic Selectors''. Malheureusement, ce système fait preuve d'un comportement peu cohérent (voir plus bas).

IPsec et ICMP

ICMP est un protocole essentiel pour le fonctionnement des protocoles ou des applications qui s'appuient sur IP. En acheminant une description sommaire des erreurs rencontrées sur le réseau, il permet d'obtenir des informations importantes sur cette boîte noire (et peu bavarde) qu'est le réseau IP. Malheureusement, le [Pos81-1], qui décrit ICMP, est relativement ancien et n'a absolument pas été conçu dans un contexte de sécurité. Certaines fonctionnalités de ICMP semblent compatibles avec IPsec, plus précisément toutes celles qui impliquent un échange entre deux hôtes : (`ECHO`,`ECHO_REPLY`) et (`TIMESTAMP`,`TIMESTAMP_REPLY`). Les messages `INFORMATION_REQUEST` et `INFORMATION_REPLY` peuvent être protégés dans le cas d'une communication d'hôte à hôte, mais aucune méthode standard n'existe (la nature de cet échange proscrit l'usage d'un AS classique). Une variante de `INFORMATION_REQUEST` est l'interrogation du réseau, qui ne peut être sécurisée par des moyens standardisés (la réponse reste un message d'hôte à hôte, et peut être protégée). Tous les messages pouvant être générés en un point quelconque du réseau ont une compatibilité faible avec IPsec : `DESTINATION_UNREACHABLE`, `SOURCE_QUENCH`, `REDIRECT`, `TIME_EXCEEDED`, `PARAMETER_PROBLEM`; compatibilité faible, car seul un système ayant la possibilité de partager un SA avec le destinataire peut envoyer de façon sécurisée ces messages. Cela implique, d'un point de vue général, qu'il est impossible de faire confiance aux routeurs intermédiaires sur lesquels transitent les paquets protégés (cela nécessiterait une PKI à laquelle souscriraient les routeurs intermédiaires : absurde et dangereux à grande échelle pour le pouvoir que cela donnerait au maître d'une telle PKI). L'usage du routage de source pourrait peut être permettre de déterminer si un routeur a légitimité pour envoyer un paquet ICMP, mais cela ne prouvera pas pour autant son honnêteté. En effet, les 64 bits d'informations obligatoires (en plus de l'en-tête IP) que ICMP doit retourner sont insuffisants pour prouver cette légitimité : cela correspond aux informations suivantes dans les protocoles AH et ESP :

```
0                   1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Header   |  Payload Len  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          RESERVED             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Security  Parameters  Index  |
|             (SPI)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

64 premiers bits du header AH



64 premiers bits du header ESP

Parmi ces informations, le SPI apporte un renseignement intéressant. Cependant, il est insuffisant pour déterminer l'hôte à qui acheminer le message si IPsec est utilisé sur une passerelle de sécurité, ou pour acheminer au bon protocole de niveau transport (par exemple, pour un message `DESTINATION_UNREACHABLE` de type `PORT_UNREACHABLE`). Si ESP est utilisé, une table de correspondance peut associer les adresses sources des paquets qui ont été récemment émis dans le tunnel et le numéro de séquence avec lequel ils ont été envoyés (le numéro de séquence n'est pas chiffré par ESP). Cela permet de retrouver l'émetteur interne du paquet problématique. Cependant, encore une fois, tout cela ne prouve pas la « bonne foi » du routeur émetteur du message.

[KA98-1], §6 suggère, en l'absence de preuve d'authentification, d'ignorer les messages ICMP autres que `REDIRECT` et `ICMP_PMTU`, qui sont critiques au fonctionnement. Dans tous les cas, les politiques de sécurité locales doivent permettre l'interdiction/l'autorisation par type de message ICMP. En l'absence de mécanismes de substitutions à ces messages de ICMP, cette stratégie semble la meilleure, bien qu'elle ouvre la porte à des dénis de service relativement simples à mettre en oeuvre (redirection illicite ou réduction extrême de la bande passante). Curieusement, le traitement spécifique de `ICMP_PTMU` fait l'objet d'un développement important dans le §6.1.2 de [KA98-1], alors que plus de précisions auraient aussi été bienvenues sur le message `REDIRECT`. Des stratégies identiques peuvent cependant être définies pour ces deux messages. Si la passerelle peut déterminer la source du paquet problématique, ou réduire les sources *potentielles (``possible originating hosts'')* à un nombre *raisonnable* (*``manageable number''*), elle doit relayer les messages ICMP directement. Sinon, elle mémorise l'information et la relaye à la prochaine source qui utilise l'AS en question dans les conditions fautives (MTU du paquet trop importante, etc.). Cela implique de recalculer le PMTU renvoyé à la source en fonction de l'overhead occupé par l'encapsulation par IPsec en mode tunnel. Par conséquent, les messages ICMP arrivant du tunnel déclenchent la construction de nouveaux messages ICMP. Ces mesures « frôlent » les limites imposées dans le §1 de [Pos81-1] : *Un message ICMP ne doit pas être envoyé à propos d'un message ICMP*; de plus la granularité des messages peut varier suivant la nature des implémentations (BITS, BITW, native, etc.), et un mécanisme doit définir une durée de vie sur les informations mémorisées (afin que le PMTU ne reste pas figé dans le temps).

Indépendamment d'IPsec, une mise à jour globale de ICMP s'impose, notamment en ce qui concerne les 64 bits de données transportés, issus du paquet original : dans le RFC, cette quantité de données est exacte, alors que dans la réalité, elle est considérée comme un minimum. Il est donc très classique de trouver des messages ICMP avec 96 bits de données ou plus. Ces informations supplémentaires (si elles ne sont pas chiffrées) peuvent soulager IPsec pour les traitements de ICMP (§15.3.1.B.3.1 de [KA98-1]. Malheureusement, il n'est pas vraisemblable de faire évoluer toutes les implémentations actuelles de ICMP, étant donné le nombre de machines impliquées. L'apparition de passerelles avec des comportements spécifiques est donc à envisager. [FS00] présente le problème comme insoluble, compte tenu du travail de mise à jour à effectuer, et de l'échelle des interactions.

Intégrité et Acheminement

Comme cela a été évoqué précédemment, de nombreux traitements sont susceptibles de modifier les paquets sur le réseau, ce qui limite l'intérêt des preuves d'intégrité, notamment celle apportée par AH, bien que le [KA98-2] classe très clairement les champs de IP en `mutable`, `immutable`, `predictable`. Avec le développement de nouveaux types de traitements ou de nouvelles options pour IPv4, des incidents sont envisageables. Cela s'est notamment produit avec l'ajout de la notification explicite d'engorgement (ECN - [3168]) : le §9.2 (``IPsec Tunnels'') de ce RFC décrit une quantité non négligeable de modifications à apporter à IPsec, notamment au traitement des entêtes, aux bases d'associations de sécurité, aux bases de politiques de sécurité, aux protocoles de négociations, au mode tunnel, etc. Le sujet étant particulièrement pointu et l'adoption de ECN semblant des plus aléatoires (voir le paragraphe sur ECN dans Documentation/Configure.help des sources du noyau linux), il n'en sera pas plus question ici.

IPsec et la Couche Transport

Le modèle en couches des réseaux est une abstraction nécessaire pour une meilleure conception des protocoles et des systèmes réseaux, et pour un meilleur développement du code qui vient « imprimer » dans la réalité les spécifications papier. C'est aussi un modèle très intéressant d'un point de vue pédagogique. Cependant, la réalité des spécifications et des implémentations est plus floue. De nombreuses incompatibilités peuvent notamment se révéler entre IPsec et des mécanismes orientés « couche transport ».

Les « Sélecteurs de Trafic »

De nombreuses confusions réapparaissent à intervalles réguliers sur les listes liées à IPsec (ipsec, freeswan, ipsp, etc.) entre les différents filtrages opérés par IPsec. Il convient de distinguer :

- L'utilisation du triplet (SPI, Protocole_{AH/ESP}, Adresse de Destination) qui permet de retrouver un SA dans le SAD. Notons qu'avec 2^{32} SPI possibles, la mention du protocole n'est pas réellement utile, et dans bien des cas, l'adresse de destination ne sert pas non plus. La prochaine version du standard (slide 9 de [Ken02]) ne rendra obligatoire la sélection d'un SA que par (SPI) pour l'unicast et par (SPI, Adresse de Destination) pour le multicast.
- Les ``Traffic Selectors'' permettent la mise en application de politiques de sécurité. Les conséquences de ces politiques sont au nombre de trois :
  1 Rejet du paquet
  2 Protection par IPsec
  3 Acheminement sans protection

  Les éléments nécessaires pour prendre la décision sont alors ceux contrôlés par les sélecteurs : Adresses source et destination, mais aussi nom de l'utilisateur et de la machine, protocole de transport, niveau de sensibilité des données,

ports sources et destination; plus de précisions sont données au §4.4.2 de [KA98-1].

Le standard laisse une certaine latitude pour les implémentations des SPD, SAD et sélecteurs, du moment qu'un ensemble de caractéristiques minimales observables sont respectées. En conséquence, les implémentations montrent de subtiles différences de comportements dans certains cas oubliés par le standard : certaines acceptent de regrouper les trafics UDP et TCP à destination d'un port identique (par exemple pour le dns) avec un seul SA. D'autres nécessitent deux SA (ce qui est un peu rigide mais qui correspond bien à l'esprit du standard). Ces considérations ont été évoquées sur la liste IPsec en Octobre 2002. Dans cette discussion Andrew Krywaniuk (Alcatel) soutient que les sélecteurs de trafic outrepassent déjà leur rôle en analysant le protocole de niveau transport... Cette position est cohérente; le filtrage opéré par IPsec devrait pouvoir se faire avec les seules données normalement accessibles au niveau IP. Un pré-filtrage pourrait être effectué aux niveaux application et transport pour les paquets sortants avec l'aide d'un pare-feux, et un post-filtrage pourrait de même être effectué après traitement par IPsec pour les paquets entrants. On délègue ainsi la responsabilité de ces filtrages a un dispositif conçu en ce but, plus modulaire et plus versatile que les ``Traffic Selectors''. Le défaut principal des sélecteurs est en effet leur manque de flexibilité, ce qui justifie de confier plutôt cette tâche à un pare-feux, et un autre défaut est qu'IPsec requiert, par l'usage des sélecteurs, des informations issues des couches supérieures qui ne sont pas toutes liées aux données, notamment le nom de l'utilisateur ou de la machine. La partie suivante montre comment les sélecteurs classiques définis pour IPsec ont été rendus en partie obsolètes par des protocoles de transport particuliers.

Stream Control Transmission Protocol (SCTP)

SCTP [SCTP00] est un protocole de transport fiable conçu pour fonctionner sur des réseaux de type paquets, et notamment sur IP. Il est prévu de l'utiliser principalement, mais pas exhaustivement, dans les réseaux de téléphonie publique commutés (PSTN).

La caractéristique de SCTP qui pose problème peut se présenter aussi dans les futurs protocoles de transport : les sessions SCTP associent un groupe d'émetteurs à un groupe de destinataires.

Cela a deux conséquences sur les traitements opérés par IPsec (cf. [BIKS02]) :

1 Le SPD doit pouvoir retrouver un SA à partir d'un nouveau type de triplet : ({groupe d'adresses destination}, SPI, AH/ESP). Deux solutions techniques sont alors possibles : soit autant de SA que d'adresses destination sont construits, soit les entrées des SPD sont généralisées sous la forme de groupes d'adresse. C'est ce dernier comportement qui est recommandé, mais le draft n'en adresse pas les conséquences : comment s'effectue le traitement pour un paquet entrant en regard d'un groupe d'adresses ? Comment identifier le bon groupe d'adresses ? L'utilisation du SPI comme élément déterminant devrait constituer la méthode la plus raisonnable, ainsi que conseillé dans [Ken02].

2 Les protocoles d'échange de clefs / de constitution d'associations de sécurité doivent assumer la complexité de SCTP. Pour cela, [BIKS02] recommande la construction d'un nouveau type `ID` pour ISAKMP : `ID_LIST`, qui représente un ensemble d'identités. L'alternative la plus évidente est en effet particulièrement coûteuse en temps et mémoire (établir un SA pour chaque couple de {source}X{destination}). L'usage de listes d'identités a cependant ses propres inconvénients : pour IKEv1, une signature doit être liée à une identité unique tout au long d'une même phase. Or, dans le contexte de SCTP, le signataire n'est pas nécessairement identique pour chaque message. En conséquence, dans l'état actuel de IKE, et sans doute aussi pour IKEv2, les groupes de signataires doivent partager des clefs identiques, avec les faiblesses de sécurité inhérentes à ces pratiques à grande échelle. Donc, soit des clefs symétriques sont partagées entre tous les acteurs, soit des listes de certificats sont envoyées avec des clefs publiques identiques pour chaque membre du groupe, soit on profite d'une caractéristique des certificats qui est l'encodage d'identités multiples en ASN.1 à l'intérieur d'un même certificat (pour une seule et même clef publique). C'est ce que recommande le draft, mais le support de cette fonctionnalité dans les implémentations des systèmes de certification est équivoque (certains préfèrent une association non-ambigüe entre une seule identité et une seule clef publique).

Avec le développement des réseaux IP, de nombreux protocoles sont destinées à naître, avec des caractéristiques au moins aussi complexes que celles de SCTP. Malheureusement, il est difficile d'anticiper sur ces développements. L'évolution de la technologie IP a déjà longtemps été considérée comme surprenante.

Session Initiation Protocol (SIP)

SIP est un protocole de signalisation pour les conférences, la téléphonie, la messagerie instantanée, la signalisation de présence ou d'événements ([Ros02]). Les interactions entre SIP et IPsec ne sont pas spécifiques à SIP, i.e. des interactions similaires se produisent avec d'autres protocoles, comme DNS, NFS... Cependant, SIP est particulièrement touché, pour des raisons de performances. Le groupe SIP étant très actif et surtout très productif, il est probable que tout un ensemble de documents viendront préciser les comportements de sécurité liant SIP à IPsec (cf. thread du 26 Septembre 2002 sur la liste IPsec : ``Protocol and port fields in selectors'', [Kry02]).

Le point à problème concerne les sélecteurs de trafic dont il a été question plus haut. SIP emploie le même numéro de port pour son trafic, que celui-ci soit un port UDP, TCP ou SCTP; et l'usage de ces trois protocoles se fera de façon non-déterministe (au contraire de l'utilisation de UDP ou TCP pour le DNS) et intensive. Créer un sélecteur de trafic pour chaque protocole de transport pose alors des problèmes importants de performances (Les proxy SIP doivent pouvoir supporter des charges très importantes de trafic). De plus, l'arrivée de nouveaux protocoles de transport tels que DCP (Datagram Control Protocol), etc., rendent nécessaire la recherche d'une réponse systématique à la question des SA mono-port/multi-protocoles.

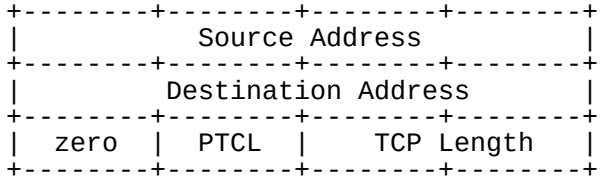
Comme il a été précisé plus haut (cf. « Les Sélecteurs de Trafic »), seules certaines implémentations permettent de construire des sélections de ce type.

Transmission Control Protocol (TCP)

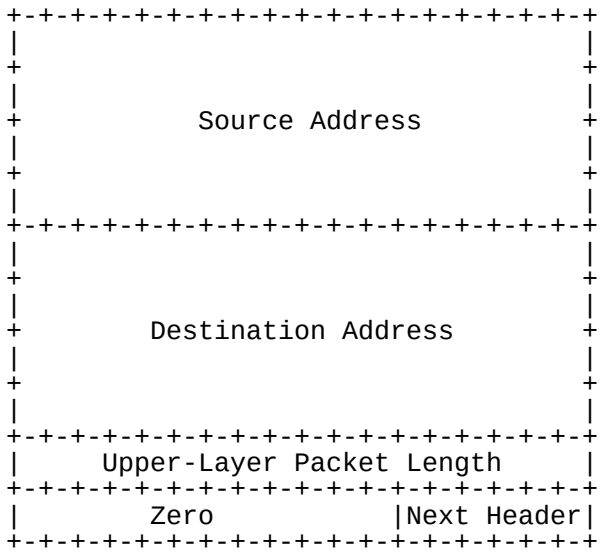
TCP est une source réelle de challenges d'interopérabilité pour de nombreux systèmes fonctionnant avec IP. Le reproche principal qui est fait à TCP est la méthode de calcul de son checksum :

Extrait de [Pos81-2] :
`The checksum also covers a 96 bit pseudo header conceptually prefixed to the TCP header. This pseudo header contains the Source Address, the Destination Address, the Protocol, and TCP length. This gives the TCP protection against misrouted segments. This information is carried in the Internet Protocol and is transferred across the TCP/Network interface in the arguments or results of calls by the TCP on the IP.`

En d'autres termes, le checksum de TCP incorpore des données caractéristiques de la couche IP de l'émetteur et du récepteur. Voici le détail des pseudo-entêtes en IPv4 et en IPv6 ([DH98]).



Pseudo-entête TCP/IPv4



Pseudo-entête TCP/IPv6

Plusieurs processus de niveau IP ont besoin de modifier les adresses IP pendant le transit, et notamment les systèmes de translation d'adresse. Dès lors, il suffit de modifier le checksum lorsqu'un tel traitement est effectué. Lorsque IPsec est utilisé de bout en bout, cela aboutit inévitablement à une erreur : si le paquet est protégé par chiffrement, il est impossible de recalculer le checksum pour le remettre à jour; si le paquet est protégé par authentification/intégrité, l'altération du checksum provoquera une erreur de vérification par IPsec. Ce sujet est particulièrement complexe, puisque de nombreux processus interviennent : NAT, AH ou ESP, TCP, mais aussi IKE (comment négocier un AS à travers un NAT ?), etc.

TCP est un protocole connecté. L'établissement des connexions et leur libération se fait sous la forme de paquets IP dont le contenu peut être déterminé très facilement `SYN/ACK,RST...`). Cette situation permet une attaque par séquences connues pour casser les clés de session utilisées par IPsec (cf. §2.2 p15 de [ADP99]).

Les traitements opérés au sein du réseau peuvent provoquer des pertes de paquets ou des ralentissements. Dans certains cas, les timers qui assurent les déblocages des états de l'automate de TCP peuvent arriver à expiration (ce qui déclenche généralement un call-back de traitement d'erreur pour l'application ou provoque un `SIG_SEGFAULT` suivi d'un *coredump*). IPsec s'en tire en général relativement bien à ce niveau : les précautions prisent dans la conception des systèmes afin de lutter contre le déni de service font que les temps de traitement sont limités (limitation de la fragmentation, file d'attente d'exposants pré-calculés, etc.). Cependant, en cas de forte charge, des latences importantes restent probables. Prenons pour exemple le cas d'un passerelle de sécurité qui redémarre. Les connexions TCP avec les ``peers'' (autres passerelles ou télétravailleurs) ont été rompues brusquement et doivent donc être re-établies. Il en résulte tout un ensemble de `SYN` TCP, lesquels déclenchent des négociations d'associations de sécurité, lesquelles sont consommatrices en échanges, calculs et mémoires. La passerelle étant chargée, les associations de sécurité peuvent ne pas être établies avant la fin de l'attente par TCP du `SYN/ACK`. Il s'agit plus d'une gêne occasionnée à l'utilisateur que d'un inconvénient majeur dans la plupart des cas. Cependant, l'utilisateur doit comprendre ce qui se passe et savoir qu'il peut relancer la connexion sans souci une fois l'AS établie. Une solution simple pour éviter ces désagréments est d'augmenter les temps d'attente.

Enfin, TCP requiert pour un fonctionnement optimal un acheminement correct des messages ICMP. Par exemple, une application peut gérer de deux façons différentes la reprise sur erreur dans le cas de la réception d'un message `DESTINATION_UNREACHABLE` d'une part, ou dans le cas de l'expiration d'un timer d'autre part. La distinction entre `PORT_UNREACHABLE` et `HOST_UNREACHABLE` constitue aussi une information riche pour l'application. Par ailleurs, une implémentation de TCP peut gérer les informations issues de la notification explicite de congestion (ECN; voir plus haut). Lorsque, en cas d'erreur, TCP attend un message ICMP qu'il ne recevra pas (parce qu'il est filtré par IPsec ou par un firewall), il peut rester bloqué en attente jusqu'à l'expiration du timer. Une solution simple pour éviter ces désagréments est de réduire les temps d'attente.

On notera que les deux solutions simples sus-mentionnées sont antagonistes (augmentation `vs` réduction des timers TCP).

User Datagram Protocol (UDP)

L'analyse précédente effectuée sur TCP nous amène à faire une brève digression sur UDP. En effet, l'entête UDP contient aussi un checksum, construit de la même façon que pour TCP : cf. §2,§3 de [Pos80].

En réalité, si ce checksum contient une information intéressante, sa prise en compte demeure de la responsabilité de l'application s'appuyant sur UDP. Le RFC précise à ce niveau qu'un checksum de `0x00` est associé à une signification particulière : ``debug'' ou « l'application supérieure ne s'intéresse pas au checksum ».

Afin d'en savoir plus, j'ai procédé à une observation de divers paquets UDP sur le réseau. Le checksum inclus dans les paquets est systématiquement non nul. Cependant, les valeurs

des checksums ne se sont pas toujours révélées correctes dans certaines requêtes DNS (`Standard_Query_PTR`). Cela n'a pas empêché le serveur de noms de répondre correctement à ces requêtes. On peut donc conclure directement que certaines implémentations ne prennent pas en compte la vérification du checksum.

Il est à craindre que de très rares protocoles ou de rares implémentations de protocoles connus fassent cette vérification, auquel cas une incompatibilité NAT/IPsec/checksum_UDP est à craindre. Je n'ai pas connaissance de l'existence de tels protocoles ou implémentations.

IPSEC ET LES PROTOCOLES DE RÉSOLUTIONS D'ADRESSES

De nombreux protocoles ont pour objectif l'association d'identifiants représentatifs de couches protocolaires différentes; Ces associations peuvent se produire pour des couples quelconques de couches, mais elles se produisent généralement entre :

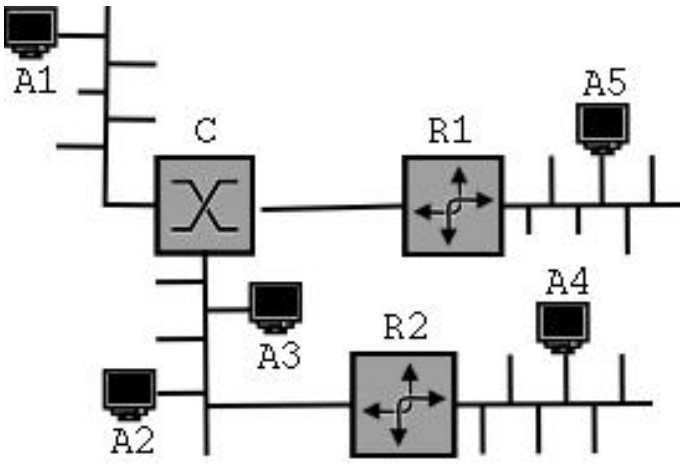
- La couche Liaison de Données et la couche Réseau
En effet, la couche réseau offre une vision de bout en bout, mais ne sait pas adresser les informations sur le lien de données. Il est donc nécessaire de traduire les adresses de niveau 3 en adresses de niveau 2 lors de l'émission sur le médium. Pour la réception, l'écoute du médium est suffisante.

- La couche Réseau et les couches supérieures
Les adresses réseaux permettent de déterminer une machine, mais cela intéresse peu l'utilisateur. Ce dernier accède à des « services », auxquels on attribue généralement un nom, une url, etc. Un mécanisme est nécessaire pour déterminer l'adresse réseau de la machine fournissant le service.

La couche réseau est souvent qualifiée, dans le monde IP, de « couche de convergence ». Elle réalise l'abstraction entre un support et des services. Il est donc normal que des mécanismes de résolution de noms ou d'adresses soient mis en oeuvre à ce niveau.

Address Resolution Protocol (ARP)

ARP permet d'associer des adresses IP et des interfaces réseau. Considérons un exemple simple :



Le schéma ci-dessus représente un ensemble de réseaux ethernet interconnectés. A1, A2 et A3 appartiennent au même sous-réseau IP (pas de VLAN ethernet).

Dans le cas où A1 veut discuter avec A2, A1 doit connaître l'adresse ethernet de A2, sinon le commutateur C ne sera pas en mesure d'acheminer les trames ethernet de A1 vers A2.

A2 et A3 sont sur un ethernet partagé. En conséquence de quoi, A3 reçoit systématiquement tout le trafic issu de A2, même celui qui ne lui est pas destiné. Il pourrait alors différencier les paquets qui lui sont destinés par l'adresse IP destination. Il n'est donc pas nécessaire, en théorie, que A2 connaisse l'adresse MAC de A3 pour lui envoyer des trames; en réalité, pour des raisons de simplicité, et parce que A2 ne sait pas a priori si A3 est sur le même ethernet partagé ou si il est accessible via un commutateur, A2 cherchera tout d'abord l'adresse MAC de A3, et A3 n'acceptera (dans le cadre d'une utilisation normale) que les trames destinées à sa propre adresse MAC). Il est intéressant de noter un effet de bord de ce scénario : dans certains systèmes d'exploitation, lorsqu'une carte ethernet est passée en mode d'écoute (*promiscuous* - réception de toutes les trames), la couche IP traite à son habitude les paquets pour lesquels l'adresse IP destination correspond à la machine locale. On peut donc repérer les espions utilisant de tels systèmes : si par exemple A2 a des doutes sur l'honnêteté de A3, il peut envoyer un message `ICMP_ECHO` (ping) avec pour destination ethernet l'adresse MAC de A1, et pour destination IP celle de A3. Si A3 répond (`ICMP_ECHO_REPLY`), cela prouve bien qu'il espionnait le trafic.

Si A2 désire envoyer un paquet IP à A4, la couche IP de A2 l'informe qu'il doit envoyer les trames ethernet au routeur R2. On est donc ramené au cas précédent : A2 contacte un système sur l'ethernet partagé, et pourrait donc le faire sans connaître l'adresse MAC de ce système. R2, en recevant un paquet qui n'est pas adressé à une machine du sous-réseau IP auquel appartiennent A1, A2 et A3, peut prendre la décision de l'acheminer. Cependant, cette situation devient complexe si deux routeurs sont sur le même ethernet partagé. Il est alors préférable que les stations s'adressent explicitement à un routeur, via son adresse MAC.

Lorsque A1 cherche à contacter A5, sa configuration IP l'informe que la passerelle est R1. A1 et R1 sont séparés par un commutateur, par conséquent, comme lorsque A1 contacte A3, la connaissance de l'adresse MAC destination est nécessaire.

ARP permet de retrouver l'adresse de couche liaison de données d'une machine à partir de son adresse réseau. Dans l'exemple précédent, il s'agit d'obtenir l'adresse MAC à partir de l'adresse IP. Sur ethernet, un broadcast ethernet est effectué, et la machine qui reconnaît son adresse IP répond en envoyant son adresse MAC à l'émetteur (le broadcast ethernet est relayé par les commutateurs). Dans d'autres types de réseau (les réseaux NBMA), un serveur ARP reçoit les requêtes et y répond.

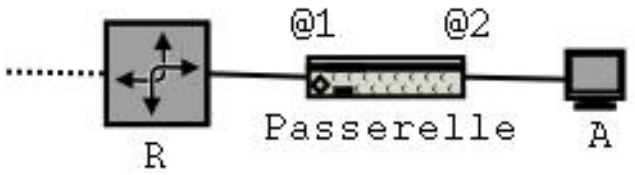
ARP ne s'appuie pas sur IP et n'a pas d'interaction directe avec IPsec. Par ailleurs, ARP ne peut pas être sécurisé avec IPsec non plus.

Dès lors, on peut en tirer les enseignements suivants :

- Un routeur qui sépare des réseaux IP différent n'a pas à faire transiter les trames ARP.
- Une passerelle de sécurité qui se comporte à la manière d'un routeur n'a pas à acheminer ARP (dès lors que toutes ses interfaces sont sur des réseaux différents).
- Un routeur (ou une passerelle) possédant deux interfaces sur

un même réseau IP n'est pas utilisé(e) convenablement. En effet, entre ces interfaces, son rôle n'est pas le routage, mais la commutation ou le bridging ; un tel dispositif ne peut donc être considéré comme une passerelle de sécurité entre ces interfaces.

- Dans la configuration suivante :



Quelles que soient les adresses IP @1 et @2 (et même si elles sont identiques), la passerelle n'a pas à acheminer ARP : tout trafic à destination de A passe par R, qui envoie ses trames avec pour destination l'adresse MAC de la passerelle. La passerelle effectue alors (si nécessaire) les traitements IPsec et achemine le paquet IP résultant dans une trame à destination de l'adresse MAC de A. Par exemple, si un paquet a pour destination l'adresse IP de A, lorsque R fait un broadcast ARP pour apprendre l'adresse MAC de A, la passerelle se fait passer pour A, en renvoyant sa propre adresse MAC. Elle récupère ensuite le paquet pour A et l'expédie avec pour destination l'adresse MAC de A; il s'agit de la méthode utilisée classiquement par les proxy-ARP pour créer les proxy transparents pour le web ou des pseudo-bridges; une passerelle de ce type n'effectue pas réellement un routage. Si un paquet a pour destination la passerelle, R l'envoie par la méthode classique. Un paquet protégé ayant A pour destination finale a @1 comme adresse IP destination du tunnel, ce qui est le fonctionnement classique du mode tunnel. Malheureusement, il n'est pas évident que toutes les implémentations fonctionnent en faisant du proxy-ARP. Dès lors, il faut se renseigner au cas par cas. Pour éviter tout problème, il est préférable d'utiliser des passerelles qui font du routage et qui nécessitent d'appartenir à des réseaux IP différents. Les modules IPsec de type ``Bump In The Wire'' (BITW), qui travaillent en mode transport ou en mode tunnel sont soumis aux mêmes considérations.

- Les hôtes, i.e. les machines utilisant IPsec en mode transport ou en mode tunnel avec une interface virtuelle, ont de fortes chances de recevoir des requêtes ARP et d'y répondre. Certaines adresses ARP (routeur, serveur, etc.) peuvent être définies de façon statique dans le cache ARP pour éviter toute usurpation (voir plus bas).

ARP autorise deux types d'attaques :

- Usurpation de l'identité du destinataire
- Usurpation de l'identité d'un routeur

Usurper une identité n'est possible que sur un même réseau IP. Dès lors, pour se faire passer pour quelqu'un d'autre, l'attaquant forge des réponses ARP associant sa propre adresse MAC à l'adresse IP du destinataire usurpé.

Si les hôtes trompés utilisent directement IPsec localement (sans passer par une passerelle), les conséquences de l'attaque peuvent être réduites. Pour cela, il est nécessaire que les hôtes disposent - avant toute négociation - de moyens cryptographiques pour s'authentifier mutuellement, par exemple:

- Un secret partagé correctement identifié.
- Les valeurs des clefs publiques respectives, ou un hash de ces clefs.
- La valeur de la clef publique d'une autorité de certification permettant de vérifier l'authenticité des certificats des partenaires.

L'attaquant n'est pas en mesure de connaître le secret partagé ou les valeurs des clefs secrètes qui lui permettraient d'usurper totalement les identités des partenaires. En revanche, à défaut de pouvoir modifier le trafic, son attaque via ARP lui permettra:

- d'observer le contenu des échanges si aucun chiffrement n'est appliqué.
- dans le cas contraire, d'enregistrer les données chiffrées pour procéder à leur cryptanalyse.

Si l'hôte trompé utilise IPsec via une passerelle de sécurité, alors les attaques par ARP peuvent fonctionner. L'attaquant peut par exemple se faire passer pour la passerelle auprès de l'hôte et pour l'hôte auprès de la passerelle. La communication hôte - passerelle n'étant pas sécurisée, l'attaquant a les pleins pouvoirs sur le trafic issu ou à destination de l'hôte. Pour éviter cela, il faut garantir les associations (adresse IP, adresse MAC), ou sécuriser les communications entre les hôtes et la passerelle, via IPsec en mode tunnel par exemple; l'intérêt de la passerelle est alors de séparer les associations de sécurité locales de celles avec les systèmes étrangers et de forcer l'application des politiques de sécurité.

Les attaques par ARP se faisant sur le réseau local, elles sont souvent sous-estimées par les administrateurs, qui considèrent que le réseau local est « de confiance ». D'après le CLUSIF (cf. p25 de [Clu01]), les incidents locaux constituent cependant une part non négligeable des sinistres informatiques. Par ailleurs, une machine corrompue par un attaquant extérieur peut lui permettre d'enchaîner sur des attaques locales via ARP.

Dans un contexte IPv6, ARP n'existe plus. On peut alors imaginer de nombreuses solutions. Par exemple, si les adresses IPv6 sont construites à partir de l'adresse MAC, la connaissance de l'adresse physique est un corollaire de celle de l'adresse réseau. En revanche, connaître l'adresse réseau de son correspondant devient plus difficile; cela augmente le travail du DNS. Si les adresses IPv6 sont construites de façon arbitraire, les hôtes peuvent connaître l'adresse physique des routeurs via les `Router_Advertisement`, et peuvent découvrir leurs adresses respectives par le mécanisme de `Neighbor_Discovery`. Cependant, ces techniques ont leurs propres inconvénients en matière de sécurité (nécessité d'avoir des groupes multicasts sécurisés, d'authentifier les avertissements des routeurs, etc.). Ces sujets constituent des pôles de recherche en cours de défrichage, et aucun standard n'émergera avant au moins un an ou deux.

Domain Name System (DNS)

Pour saisir comment DNS peut compromettre IPsec, il faut dans un premier temps décrire les vulnérabilités du DNS. Ces descriptions sont issues d'un rapport de l'ISP BGP & DNS Working Group ([Isp02]), rédigé dans le cadre du ``National Security Telecommunications Advisory Committee'' (créé par R. Reagan en 1982). Pour des descriptions plus précises au niveau des réalisations pratiques, se reporter à [AA02].

Vulnérabilités du DNS :

- Les échanges DNS engendrent des volumes de données asymétriques. Plus précisément, une réponse contient en général bien plus d'informations qu'une requête. Dès lors,

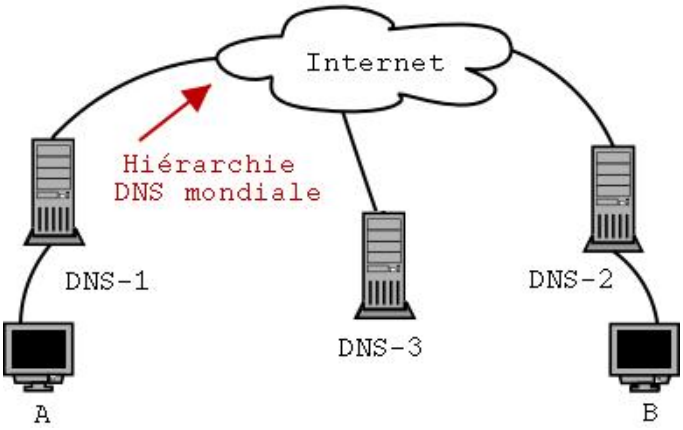
les DNS sont utilisables en tant que tierces parties pour provoquer des dénis de service. Pour ce faire, l'attaquant envoie de multiples requêtes DNS avec pour source l'adresse de sa victime. Les serveurs DNS vont alors inonder cette dernière avec leurs réponses (cf. [Hou00]).

- Les serveurs DNS acceptant des enregistrements de ressources (``Ressource Records'') en provenance de sources ne faisant pas autorité (non-authoritative sources) peuvent être trompés par un serveur DNS piraté. Cela peut notamment affecter les liaisons descendantes de l'arbre DNS (glue records), et plus précisément les références vers les serveurs de domaines de premier niveau (Top Level Domains), tels que fr, com, mil, org, etc. cf. [DD01].
- L'absence d'authentification permet à quiconque de répondre à des requêtes DNS. Une attaque en aveugle est possible, l'attaquant devant alors déterminer la date de la requête, le `DNS_Transaction_ID` et le `DNS_Query_Port`. De la même manière, il est possible d'effectuer un détournement de connexion TCP lors d'un transfert de zones (description du contenu de la zone, avec notamment les serveurs DNS, les serveurs mails, etc.). Les ISPs sont les premières victimes potentielles de telles attaques.
- Lorsque les informations concernant une zone ne sont pas hébergées par un serveur DNS appartenant à la zone elle-même, mais par un serveur appartenant à une autre zone, la chaîne des serveurs DNS pouvant fournir une fausse information sur la zone s'allonge. Par exemple, pour contacter www.vilya.org, nous pouvons nous adresser à un serveur root, à un serveur de .org ou à un serveur de vilya.org. En réalité, les serveurs à qui nous nous adressons feront exactement ces démarches (à moins qu'ils n'aient déjà la réponse dans leurs caches). Un serveur de nom de vilya.org est ns0.blackcatnetworks.co.uk . Cela implique que n'importe quel serveur des zones root, org, vilya.org, uk, co.uk, blackcatnetworks.co.uk peut envoyer de fausses informations sur www.vilya.org. D'autre part, les informations de la zone blackcatnetworks.co.uk auraient elles-aussi pu être contenues dans d'autres zones, etc. De plus, par le jeu des serveurs caches DNS, plusieurs serveurs appartenant à la zone d'où sont émises les requêtes peuvent aussi fournir de fausses informations.
- Le DNS dynamique ([VTRB97]) permet l'ajout, la suppression et la modification d'enregistrements du DNS, par exemple par des clients classiques ou des serveurs DHCP. Sans précautions, des personnes non autorisées peuvent effectuer des mises à jour.
- Les serveurs DNS sont des programmes connus pour leur quantité notable de bugs et de failles de sécurité, et pour le retour de ces bugs et failles lors de changements de versions. Seules les mauvaises réputations des serveurs mails et des serveurs web parviennent à faire oublier celle des serveurs DNS. Il n'est cependant pas irréaliste d'imaginer l'émergence d'un ``worm'' qui se propagerait par des DNS vulnérables.

La mise en oeuvre d'IPsec peut provoquer un excès de confiance que les faiblesses du DNS permettent à un attaquant d'exploiter.

Tout d'abord, le DNS est indispensable au fonctionnement du réseau. On pourrait imaginer des systèmes de très haute sécurité dans lesquels les utilisateurs seraient forcés de mémoriser des adresses réseau, mais dans un monde de cryptographie omniprésente, cela n'a pas lieu d'être, les smes d'authentification étant là pour répondre aux problèmes d'usurpation de nom(s). Avec les perspectives d'évolution d'IP, et notamment la mobilité et les systèmes d'adressage d'IPv6, la correspondance nom - adresse IP perd de sa rigidité. Par conséquent, le nom DNS demeure la solution la plus viable pour assurer la disponibilité des services. Bien que doté d'une structure arborescente et d'une autorité de gestion (le NIC), le DNS n'est absolument pas sûr, et le pouvoir n'y est pas centralisé. Aux extrêmes, il est possible de trouver certains DNS qui rejettent totalement le NIC et donnent leur propre vision de l'arborescence mondiale, avec les risques que cela implique dans les communications entre machines utilisant deux arbres DNS différents et s'échangeant des noms (par exemple des mails ou des URLs). Des solutions de sécurité pour le DNS existent (DNSsec, [Eas99]), mais leurs besoins en termes de performances sont très importants, de par l'usage de la cryptographie pour authentifier des requêtes (pour éviter le rejeu, confirmer aussi l'absence d'une entrée dans le DNS, etc.). En conclusion, DNS est indispensable et ne sera pas sûr avant longtemps.

Cela a des implications directe sur l'utilisation d'IPsec, puisque les utilisateurs auront besoin du DNS pour faire l'association entre un service et une machine. Dès lors, cela implique que IPsec doit disposer de règles de filtrage appropriées pour le DNS. Il est évidemment hors de question de bloquer le DNS, il reste donc la possibilité de protéger DNS par IPsec ou de le laisser passer « en clair ». Protéger DNS par IPsec n'a d'efficacité que si les correspondants sont tous les deux décrits de façon correcte dans le même DNS avec lequel s'établit la résolution de nom protégée, ou si leurs descriptions sont accessibles dans des DNS qui appartiennent à un même réseau de confiance (PKI ou PGP). Par souci de clarté, considérons le schéma suivant :



DNS-1 contient la description de A et DNS-2 contient celle de B, les liaisons A <-> DNS-1 et B <-> DNS-2 sont protégées par IPsec. Si un utilisateur de A cherche à contacter B par son nom, A demandera à DNS-1 l'adresse de B. A partir de là, plusieurs possibilités se présentent :

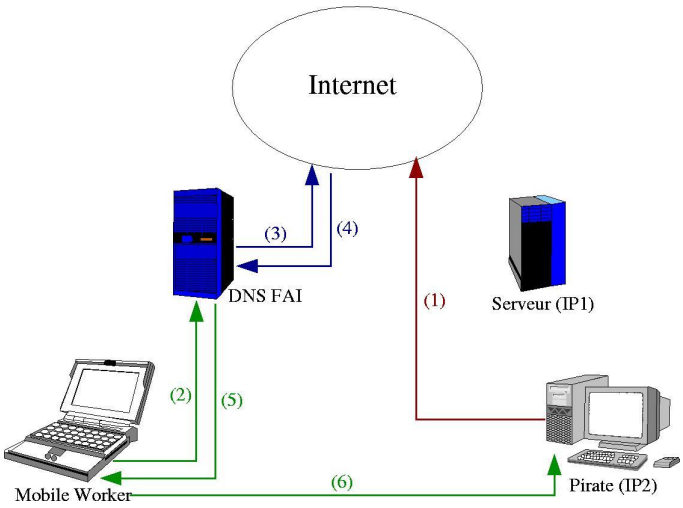
- Si DNS-1 == DNS-2 , et en l'absence de mécanismes « douteux » de mise à jour des entrées du DNS (DynDNS), alors la résolution s'effectue en toute confiance.
- Si DNS-1 et DNS-2 se connaissent, savent pour quelles zones ils ont autorité, et peuvent établir des associations de sécurité entre eux, la résolution se fait en toute confiance. Ce scénario peut cependant nécessiter l'appartenance de DNS-1 et DNS-2 à un réseau de confiance (Web of Trust ou PKI).
- Si DNS-1 et DNS-2 ne se connaissent pas, le jeu des

résolutions récursives va amener DNS-1 à contacter d'autres serveurs DNS. Si il contacte un DNS-3 en lequel il a confiance et que DNS-3 a confiance en DNS-2, alors des heuristiques de transfert de la confiance peuvent être mises en oeuvre pour déterminer une mesure de confiance sur la résolution du nom.

- Si, par résolutions récursives, DNS-1 n'obtient jamais une preuve directe depuis DNS-2 de la relation liant le nom de B avec son adresse, la résolution est incertaine, et l'utilisateur n'est en général pas mis au courant : les interfaces utilisateurs n'ont pas prévu cette situation, et de plus c'est à l'utilisateur de préciser s'il veut une résolution sûre ou non, et encore une fois, les interfaces ne permettent pas de spécifier ce point de façon souple. Notons que lors d'une résolution, un DNS-3 a peut être attesté par signature de confiance que DNS-2 est autorité pour la zone à laquelle appartient B, mais cela ne prouve pas pour autant que la réponse de DNS-2 sera valide (n'importe qui peut l'intercepter et la modifier).

Considérons maintenant quelques scénarios d'attaque :

Scénario 1

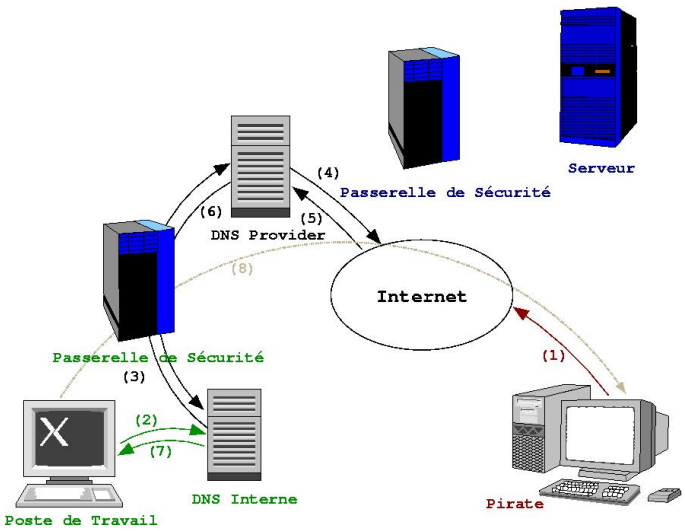


Dans ce scénario, un attaquant usurpe l'identité DNS d'un serveur. Pour ce faire, il corrompt les données d'un serveur DNS sur Internet (1), en profitant d'un faux transfert de zone, en effectuant une fausse réponse, une mise à jour dynamique, etc. Il associe le nom du serveur à son adresse (IP2). Par la suite, un télétravailleur désire se connecter au serveur. Son application est un simple browser, et il saisit l'URL du serveur, par son nom. En temps normal, une fois l'adresse IP (IP1) du serveur récupérée, la pile IPsec locale observe qu'une entrée existe dans la base des politiques de sécurité pour l'adresse IP destination (IP1) et déclenche donc IKE pour établir une association de sécurité avec le serveur. Sauf qu'ici, la résolution DNS renvoie une fausse réponse : le client demande l'adresse IP du serveur au DNS du FAI (2), qui fait remonter la requête (3) jusqu'à un DNS qui lui fournit la réponse forgée par l'attaquant. L'information est alors redescendue (4) jusqu'au client (5). L'adresse IP2 n'est pas listée dans le SPD, donc IKE n'établit pas d'association de sécurité, et la politique par défaut s'applique (acheminement en clair ou rejet du paquet). Il y a donc soit détournement de connexion, soit déni de service. Pour éviter ces soucis, il est possible :

- d'utiliser un DNS plus sûr que celui du FAI.
- de se connecter au serveur en spécifiant directement l'adresse IP.
- de décrire les politiques avec des noms de machines et de mettre à jour le SPD lorsque des résolutions de ces noms sont déclenchées (cela requiert quelques modifications dans le processus d'envoi et de réception de paquets).

On remarque encore qu'il est indispensable de définir des politiques de sécurité appropriées pour gérer les accès Internet des télétravailleurs.

Scénario 2



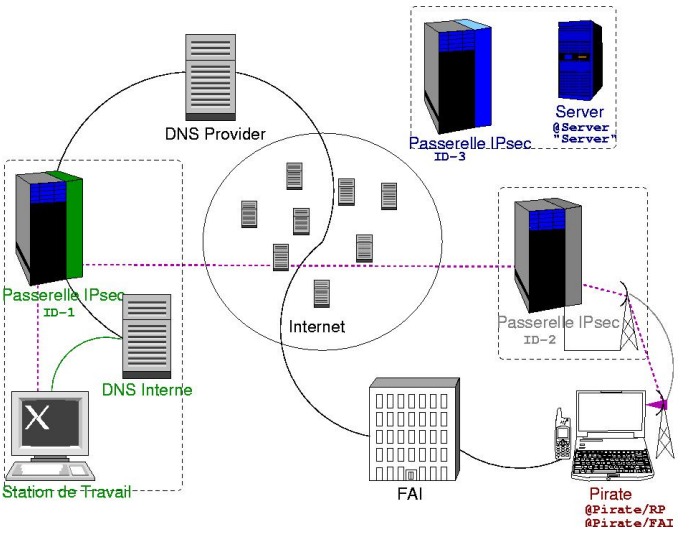
Dans ce scénario, un utilisateur cherche à contacter un serveur depuis son réseau d'entreprise. Le serveur peut appartenir à un réseau protégé par une passerelle de sécurité ou disposer lui-même d'IPsec.

De la même manière que précédemment, l'utilisateur rentre l'URL du serveur dans un browser. Normalement, la passerelle de sécurité, en voyant partir la requête http à destination du serveur doit la faire transiter dans un tunnel IPsec. Cependant, le pirate a empoisonné un cache DNS ou envoyé une mauvaise réponse à un `DNS_Query`. Lorsque l'utilisateur déclenche la résolution de nom (2), le DNS interne interroge le DNS du provider en passant sans dommage par la passerelle de sécurité (3). Le DNS du provider récupère depuis Internet l'information corrompue par le pirate (4)(5), et la fait parvenir au DNS interne (6), qui renvoie au client (7). Le client envoie alors ses paquets http à la mauvaise adresse, et comme la passerelle de sécurité n'a pas de politique de sécurité particulière pour cette adresse, le traitement par défaut est effectué : rejet ou acheminement sans protection. Bien souvent, il s'agira d'acheminement sans protection (8), et le pirate mettra en place un faux serveur web lui permettant de duper l'utilisateur. Ce dernier peut ne se rendre compte de rien, sauf s'il remarque que la machine avec laquelle il communique n'a pas la bonne adresse IP ou que la connexion s'est établie trop vite (i.e. il n'y a pas eu le ralentissement caractéristique dû aux constructions des associations de sécurité, mais ce ralentissement peut aussi être simulé par le pirate !).

Les solutions à mettre en oeuvre sont similaires à celles présentées dans le scénario précédent.

Dans les scénarios 1 et 2, l'étape de protection par IPsec est évincée par l'attaquant par une attaque sur le DNS. Dans le scénario suivant, IPsec est effectivement mis en oeuvre, mais détourné au profit du pirate.

Scénario 3



Dans ce scénario, trois réseaux interviennent. La passerelle de sécurité du client (ID-1) est capable d'authentifier les passerelles de sécurité des deux autres réseaux (ID-2 et ID-3), parce qu'elle connaît leurs clefs publiques ou parce qu'elle partage un secret avec elles. Respectivement, les passerelles des autres réseaux sont capables d'authentifier ID-1. En revanche, les passerelles ID-2 et ID-3 peuvent ne pas se connaître, appartenir à des entreprises différentes, **voire concurrentes**.

Le pirate a réussi à accéder au réseau dans lequel se trouve la passerelle ID-2 grâce à la portée trop importante du réseau wireless (cf. ``Conducting the Site Survey'', p39, [AS02]). Il dispose d'une adresse, `@pirate/RP`, dans ce réseau privé, et d'une autre adresse, `@pirate/FAI`, sur le réseau d'un fournisseur d'accès internet auquel il accède par GSM (il existe des moyens plus discrets pour obtenir le même résultat, dont il ne sera pas question ici pour des raisons de simplicité). Eventuellement, le pirate peut agir de concert avec un autre pirate.

En dehors des requêtes DNS depuis les serveurs DNS internes, **tout trafic interne à destination d'un site inconnu est rejeté au niveau de la passerelle**. Les échanges sont donc authentifiés, chiffrés, etc., systématiquement, et aucune discussion n'est engagée avec des systèmes inconnus autres que les DNS des providers.

Le pirate (ou un de ses complices), empoisonne un cache DNS judicieusement choisi (DNS du FAI, ou d'un domaine intervenant dans la résolution), et provoque l'association du nom `serveur` à l'adresse `@pirate/RP`.

Lorsque la station de travail tente de se connecter au serveur, une résolution DNS est faite (Station de travail <-> DNS interne <-> DNS provider <-> DNS d'Internet <-> DNS empoisonné), qui retourne la mauvaise adresse IP.

Le client envoie alors ces paquets à destination de `@pirate/RP`. La passerelle de sécurité ID-1 reconnaît l'adresse comme appartenant au réseau dans lequel se trouve ID-2, et établit une association de sécurité avec ID-2. Les paquets à destination du serveur, qui auraient dû passer par ID-3 sont ainsi amenés à passer par ID-2. ID-2 les décapsule et les envoie au pirate. IPsec a donc été détourné de son objectif initial et a protégé une communication qui était illicite !

On pourrait simplifier le scénario en supposant que le pirate est un utilisateur régulier du réseau protégé par ID-2 et procède à cette attaque pour obtenir des informations sur son concurrent (protégé par ID-3) en trompant son fournisseur ou son acheteur.

Dans tous ces scénarios, ce n'est pas tant les faiblesses d'IPsec qui sont en cause que celles du DNS ou l'ignorance des utilisateurs qui ont tendance à associer de façon rigide adresse IP, nom DNS, et identité pour IKE. Comme souvent, c'est le maillon le plus faible qui est exploité. Etablir ses propres DNS et limiter la confiance dans les DNS non authentifiables constitue une pratique raisonnable pour la sécurité, bien que peu dynamique pour le service rendu.

IPsec et la Translation d'Adresse (NAT)

Le §1 de [SE01] définit le translateur d'adresse réseau comme un système permettant à un réseau d'isoler son modèle d'adressage interne du modèle externe, pour des raisons d'intimité ou tout simplement parce que l'espace d'adressage interne ne pourrait pas être utilisé tel quel avec le reste du réseau (cas des adresses privées non routables).

NAT / NA(P)T

Afin de remplir les objectifs énoncés ci-dessus, les implémentations classiques de translateurs d'adresses peuvent mettre en oeuvre deux types de solution :

1 Les ``Basic NAT'' permettent le partage de N adresses publiques entre M machines, avec N < M (voir §4.1.1 de [SH99]). Les associations entre une adresse publique et une adresse privée changent régulièrement ou après un certain temps d'inactivité dont la valeur par défaut est modifiable et peut dépendre des implémentations. Lors des périodes de pointe, si les N adresses publiques sont déjà allouées à N machines, les M-N machines restantes ne peuvent accéder à l'extérieur.

2 Les ``NAPT'' (translateurs de ports) poussent plus loin le principe en travaillant sur les identificateurs de la couche transport, i.e. les numéros de ports pour TCP et UDP ou les `Query_Identifiers` pour ICMP (cf. §4.1.2 de [SH99]).

En réalité, comme suggéré dans le §5.4 de [SE01], de très nombreux dispositifs NAT sont capables d'utiliser les deux techniques précédentes, i.e. d'opérer dans un premier temps en ``Basic NAT'', puis de passer en ``NAPT'' quand les adresses publiques viennent à manquer. Il existe par ailleurs d'autres pratiques pour le NAT, mais ces dernières sont plus rares (le lecteur en trouvera une liste non exhaustive avec description au §4 de [SH99]).

De part l'isolation qu'il instaure entre réseau interne et réseau externe, le NAT est considéré par certains comme un dispositif de sécurité. En effet, en l'absence de définition de liaison statique entre des coordonnées (adresse pour le ``Basic NAT'' ou couple (adresse, port) pour ``NAPT'') externes et internes, un système extérieur n'a pas le droit d'établir une connexion avec un système interne (les connexions issues de l'extérieur sont interdites). Seules les sessions établies depuis l'intérieur sont alors autorisées. Cependant, le premier usage des translateurs est de répondre au manque d'adresse IPv4, les avantages en terme de sécurité constituent plutôt un effet secondaire dont il faut connaître les limitations. Malheureusement, on a pu constater dans les fiches techniques

de certains constructeurs (U.S. Robotics et surtout BeWAN Systems) que le NAT est listé dans les dispositifs de sécurité (avec le pare-feux, la zone démilitarisée, IPsec et le verrouillage horaire) et non dans les fonctionnalités de connectivé (où l'on trouve DynDNS et DHCP).

Quelques problématiques de sécurité liées à l'utilisation du NAT sont les suivantes :

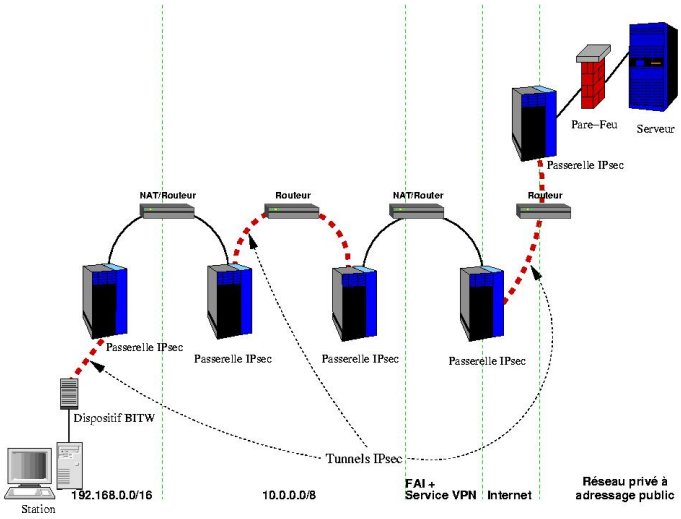
- Le fait que les ouvertures de sessions se fassent à l'initiative de membres du réseau interne n'empêche absolument pas les envois de paquets depuis l'extérieur, l'altération de paquets ou le détournement de connexions une fois les sessions démarrées.
- Il est difficile de traquer le responsable d'une attaque perpétrée depuis le réseau interne à destination d'un site extérieur (cf. §6.1 de [SE01]).
- Suivant les implémentations de translateurs, (cf. §4.6 de [SE01]), l'usage des options IP `Record_Route`, `Strict_Source_Route` et `Loose_Source_Route` peut permettre de passer outre la translation d'adresse des paquets sortant, ce qui permet ainsi de faire du ``spoofing'' (usurpation d'adresse) d'une adresse publique ou privée depuis le réseau interne. Cette technique peut notamment être utilisée pour provoquer des dénis de service utilisant un DNS comme intermédiaire (voir plus haut) ou pour effectuer un scan furtif.

NAT encapsulé par IPsec

Le §2.1 de [SE01] souligne que l'utilisation du NAT n'affecte en rien le fonctionnement des deux réseaux (privé et public), à condition bien sûr que NAT puisse effectuer ses opérations. Notamment, NAT n'affecte en rien le routage.

Par conséquent, il est possible d'utiliser IPsec dans n'importe lequel des deux réseaux, mais pas entre les deux (voir partie suivante).

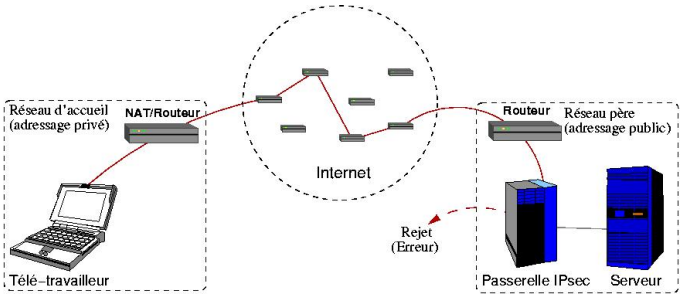
Cela signifie que le scénario suivant fonctionne :



On trouve ainsi sur le marché des dispositifs NAT & VPN, et ces dispositifs constituent de bonnes solutions de connectivité SOHO (Satellite Office - Home Office). En réception, les paquets sont traités tout d'abord par IPsec, puis par NAT, et sont enfin acheminés vers l'hôte interne. En émission, le NAT effectue la translation, puis fournit le paquet à IPsec avant émission sur le réseau public.

IPsec translaté

L'utilisation du NAT à l'issu de l'application d'une protection par IPsec est fatale. Notamment, un scénario aussi simple que le suivant ne pourra fonctionner :



Ce scénario n'a rien d'original, bien au contraire. Le manque d'adresses IPv4 force nombre de PME à utiliser des dispositifs NAT, ce qui rend difficile l'établissement d'associations de sécurité de bout en bout.
De même, les utilisateurs mobiles se connectant depuis un aéroport ou un hôtel ont de fortes chances d'être derrière un translateur (et dans ces exemples, la sécurité n'est absolument pas facultative). Enfin, même les Fournisseurs d'Accès Internet commencent à manquer d'adresses publiques et utilisent d'ores et déjà des translateurs d'adresse sur le terrain (Wanadoo et Oléane).

Des causes diverses participent à cet échec.

Si on considère l'utilisation de AH (en mode transport ou en mode tunnel), les opérations effectuées par le NAT - modification des adresses source et destination, et éventuellement des identifiants de niveau transport - compromettent l'intégrité du paquet, en conséquence de quoi IPsec le rejettera.

Si on considère l'utilisation de ESP, les données protégées sont chiffrées et/ou authentifiées. Cela implique qu'une modification de ces données fera échouer IPsec. Or, les protocoles de transport les plus utilisés incorporent une somme de contrôle qui dépend des adresses source et destination : UDP [Pos80], TCP [Pos81-2] et SCTP [SCTP00] font parti de ces protocoles. Les opérations effectuées par le NAT requièrent donc la modification des sommes de contrôle de ces protocoles. Cette contrainte est allégée pour UDP, pour lequel une somme de contrôle nulle est suffisante (cf. [Bra89] et voir plus haut :``IPsec et la Couche Transport'') : les différentes implémentations d'UDP recevant cette valeur doivent ignorer la vérification de la somme de contrôle.

Les parties suivantes viennent compléter avec force de détails les remarques précédentes. Elles s'appuient sur le document [AD02] du 18 août 2002 dont l'objectif est justement le recensement des incompatibilités entre IPsec et le NAT.

Incompatibilités intrinsèques à NA(P)T

Ces incompatibilités sont inhérentes au principe de fonctionnement de NA(P)T.

- NA(P)T met en échec AH [KA98-2] en corrompant les adresses source et destination du paquet : la preuve d'intégrité (HMAC) devient fausse. Voir ci-dessus.
- Les protocoles de transport TCP, UDP et SCTP incorporent une somme de contrôle dont le calcul fait intervenir les adresses IP source et destination. Cette somme de contrôle n'est facultative que pour UDP en IPv4. Le NAT doit donc modifier cette valeur. Si ESP en mode transport chiffre le paquet, le NAT ne peut effectuer le calcul. Si ESP en mode transport protège le paquet en intégrité, les modifications

opérées par le NAT feront échouer la vérification d'intégrité. NAT et ESP en mode transport sont donc incompatibles, **sauf pour transporter UDP sur IPv4** (voir plus haut) ou tout autre protocole ne faisant pas intervenir les adresses IP dans la somme de contrôle (et notamment ICMP [Pos81-1]). En revanche, ESP en mode tunnel est compatible avec le ``Basic NAT'', mais pas avec NAPT.

- Quand les identificateurs `ID_IPv4_ADDR`, `ID_IPv6_ADDR`, `ID_IPv4_ADDR_SUBNET`, `ID_IPv6_ADDR_SUBNET`, `ID_IPv4_ADDR_RANGE`, `ID_IPv6_ADDR_RANGE` sont utilisées dans IKE, ceux-ci doivent être cohérents avec les adresses du paquet, sinon celui-ci est rejeté (cf. [HC98]). Toute autre identité pourra être utilisée à la place (cf. [Pip98] pour la liste des types d'identité du DOI IPsec), mais ces dernières ne seront pas compatibles avec les entrées décrivant des sous-réseaux dans la base des politiques de sécurité.
  Rq : Pour les adresses IPv6, cela a peut d'incidence sur le NAT, puisque les translateurs perdent une majeure partie de leur intérêt avec IPv6 et seront a priori peu utilisés dans ce contexte.
- Si plusieurs hôtes derrière le dispositif NAPT désirent établir des associations de sécurité avec des partenaires extérieurs via IKE, une translation du port source de IKE est requise pour chaque hôte. Cela n'a pas d'incidence immédiate (sauf pour certaines implémentations : voir plus bas), mais après un certain temps, le rafraîchissement des clés peut se faire à l'initiative de n'importe quel partenaire, et il est nécessaire que les partenaires extérieurs utilisent le port `flottant' alloué par NAPT précédemment, et non le port UDP 500. Par ailleurs, même en respectant cette règle, il n'est pas évident que le dispositif NAPT conserve le contexte associé à ce port `flottant' (auquel cas, le rafraîchissement échoue).
- Lorsque plusieurs hôtes derrière un NAPT établissent des associations de sécurité avec un même partenaire, ce dernier ne voit qu'une seule adresse (celle du dispositif NAT) et ne peut donc discriminer les associations de sécurité en fonction de l'adresse, lors du passage par la base des politiques de sécurité.
- Les `Index de Paramètres de Sécurité' (SPI) utilisés dans AH et ESP sont à sens unique. Par conséquent, pour un échange donné, les paquets émis depuis un hôte du réseau privé à destination d'un serveur sur le réseau public ont un index différent de ceux émis dans le sens contraire. Lorsque plusieurs associations de sécurité ont été établies simultanément, le dispositif NAT peut être tout simplement incapable de déterminer une destination correcte à partir du SPI d'un paquet issu du réseau public, et c'est pourtant la seule information dont il dispose si le paquet est chiffré par ESP.
- Certains protocoles s'échangent des informations mettant en jeux des adresses IP qui sont souvent celles des hôtes. Dans les implémentations classiques de NAT, ces protocoles nécessitent des traitements spécifiques appelés ``Application Layer Gateways'' (ALG) (cf. [linux]). Parmi les protocoles concernés, on trouve FTP [PR85], IRC [Kal00-1] & [Kal00-2], SNMP [CFSD90], LDAP [YHK95], H.323 [H.323.00], SIP [SIP02]... Lorsque les paquets sont protégés par IPsec, toute modification au niveau applicatif est exclue, soit parce que le contenu est chiffré, soit parce que cela compromet l'intégrité du paquet.

Faiblesses des implémentations

Ces incompatibilités sont le fait d'implémentations maladroites mais malheureusement très répandues. On ne devrait plus les rencontrer dans les implémentations futures.

- Certains NAPT sont soit incapables d'acheminer du trafic TCP, soit incapables d'acheminer autre chose que UDP. Ces dispositifs ne permettent pas la traversée de AH et ESP.
- Comme cela a été mentionné plus haut, la relation entre (adresse locale, port local) et (adresse publique, port public) expire après un certain temps. Dans certaines implémentations de NAPT, ce temps est trop court pour permettre à IKE d'achever la construction des associations de sécurité pour le DOI en phase 2. Un ordre d'idée de ces temps est disponible sur le site de [cisco.com] : 5 minutes pour du trafic UDP classique, 1 minute pour du trafic DNS, et jusqu'à 24 heures pour une session TCP qui n'a pas été interrompue explicitement. Dans la majorité des cas, IKE doit pouvoir s'accommoder d'un temps maximum de 5 minutes entre les messages. Cela dépend cependant de la taille des clefs utilisées (surtout si de la cryptographie à clef publique est utilisée), et de la nécessité ou non de valider une chaîne de certification pour la clef du peer.
- Très peu de dispositifs NAPT sont aptes à gérer la fragmentation réalisée sur le réseau privé, pour les deux raisons suivantes :
  1. Lorsque plusieurs hôtes du réseau privé communiquent avec un même serveur, il existe une probabilité (cf. §6.3 de [SE01]) pour que des identificateurs de fragments identiques soient utilisés pour plusieurs flux différents. La destination ne sera pas capable de faire la différence, ce qui aboutira à un échec de tous ces flux. Certains dispositifs corrigent ce problème en faisant de la `translation d'identificateur de fragments'.
  2. Les fragments peuvent avoir une taille minimale de 68 octets, ce qui est insuffisant pour transporter l'entête TCP dans son intégralité, et notamment le champ hébergeant la somme de contrôle. Les dispositifs NAT sont ainsi obligés de modifier un des fragments suivants, lequel a pu arriver avant, car l'ordre d'arrivé des fragments n'est pas garanti ! De plus, certaines informations, telles que des adresses réseau, peuvent nécessiter un traitement particulier par une ALG (voir plus haut) et ont pu être divisées entre plusieurs fragments. Cela implique qu'un dispositif NAT doit ré-assembler les fragments.

  En revanche, la plupart des dispositifs sont capables de fragmenter les paquets si nécessaire avant acheminement.
- Lorsqu'un paquet arrive par fragments depuis le réseau public, le dispositif NAPT peut ne pas avoir toutes les informations nécessaires afin de l'acheminer. Pour cela, il lui faut les informations contenues dans l'entête du protocole de transport (UDP, TCP, ICMP ou SCTP), lesquelles peuvent arriver en retard par rapport aux autres fragments. Deux solutions peuvent être mises en oeuvre :

  Réassemblage total du paquet, puis acheminement du paquet vers le destinataire final.

  Attente des informations de l'entête de transport, définition d'une translation d'identificateur de fragments, et acheminement des paquets vers le destinataire final.

Très peu de dispositifs NAPT répondent correctement aux problèmes de fragmentation présentés ci-avant.

Incompatibilités des `mécanismes d'aide'

Certaines implémentations mettent en place des mécanismes sensés aider le passage de certains protocoles. Ces mécanismes

ont cependant des inconvénients assez lourds.

- Quelques implémentations de NAT tentent d'utiliser les ``cookies'' de IKE pour trier les différentes sessions de IKE liées à des paires d'hôtes différentes. Comme les ``cookies'' changent lors d'un rafraîchissement de clefs, et que ces rafraîchissements se font sous la protection de la phase 1 de IKE, cela cause des échecs.
- Certaines implémentations de IKE n'acceptent pas les paquets dont le port source est différent de 500. Afin d'éviter cette situation, certains NAPT ne translatent pas le port 500. Par conséquent, ces NAPT ne supportent simultanément qu'un client IKE, à moins qu'ils n'inspectent les champs de IKE, ce qui aboutit aux incompatibilités listées ci-dessus et ci-dessous.
- ISAKMP est un protocole riche et complexe. Les implémentations qui analysent les champs d'ISAKMP ne sont pas toutes en mesure de supporter les différentes combinaisons de champs, les différentes manières de les ordonner ou les mutations propriétaires de ces champs (`vendor_id`, valeurs réservées, etc. cf. [MSST98]).

Perspectives

Les drafts [HSSVD02] et [KHSV02] apportent depuis peu une solution aux problèmes causés par les NATs. Comme mentionné plus haut, le trafic UDP sur IPv4 a des affinités avec NAT, notamment parce que le calcul de la somme de contrôle est facultatif. Il s'agit d'ailleurs du protocole de transport qui cause le moins d'incompatibilités avec NAT. Les contraintes énoncées par le draft des ``requirements'' ([AD02]) précisent qu'une solution ne doit pas impliquer des modifications dans le réseau, i.e. sur les routeurs ou les dispositifs NAT. En conséquence de quoi, la construction d'un tunnel UDP semble toute indiquée.

D'emblée, AH est exclu, de part la portée de son contrôle d'intégrité, qui est trop grande. ESP Tunnel et ESP Transport sont supportés explicitement par le draft. Dans le cas de ESP Transport, c'est le support de L2TP/IPsec ([PADZB01]) qui est visé, puisque cette technologie va connaître une diffusion importante (au sein des systèmes d'exploitation et des routeurs), et parce que Microsoft et Cisco sont des acteurs important dans les travaux concernant l'interopérabilité de NAT et de IPsec.

Le principe de ce tunnel UDP [HSSVD02] est relativement simple : Les partenaires de l'association de sécurité communiquent via le port UDP 4500, et des ports dynamiques sont alloués sur le NAT. Les paquets UDP encapsulent tout le trafic IPsec :

- ESP Tunnel ou ESP Transport, en utilisant un SPI différent de zéro.
- IKE, en le préfixant d'un marqueur de 32 bits (aligné sur le SPI de ESP), ayant pour valeur zéro.
- ``NAT-keepalive'', des paquets sans effets de bout en bout, qui ne servent qu'à maintenir l'état des ports alloués par le NAT (ce datagramme est aisément reconnaissable car il ne contient qu'un octet de données, avec la valeurs `0XFF`).

Cette solution implique cependant de nombreuses précautions. Ainsi, en mode tunnel, une politique doit être mise en place afin de ne pas avoir d'incohérence d'adresses, par exemple entre deux réseaux privés utilisant tous les deux le sous-réseau 10.0.0.0/8. Dans ce cas, on peut envisager d'assigner au partenaire une adresse, ou on peut effectuer du ``Basic NAT''. En mode transport, c'est le calcul des sommes de contrôle (pour TCP, etc.) qui requiert le même type de mesure.

Enfin, d'incontournables brèches de sécurité se révèlent :

- L'utilisation de UDP facilite les attaques de type `déni de service'.
- En mode tunnel, si deux hôtes derrières deux NAPTs différents ont la même adresse et communiquent avec la même passerelle, cette dernière peut rencontrer des difficultés pour choisir vers quel NAT envoyer ses paquets.
- En mode transport, si plusieurs clients placés derrière un NAT communiquent avec un même serveur, ce dernier ne voit que l'adresse du dispositif NAT, et dispose donc de plusieurs associations de sécurité identifiées par ce dispositif. Si les trafics avec les clients sont de nature différente (i.e. UDP pour l'un, TCP pour l'autre, ICMP pour un autre, etc.), il n'y a aucune conséquence. En revanche, dans le cas plus probable où des trafics de même nature devraient être acheminés vers les clients, les sélecteurs de trafic actuels de la base d'association de sécurité seraient incapables de déterminer la bonne association de sécurité à appliquer.

Pour repérer la présence d'un ou plusieurs NAT et pour déterminer si le partenaire supporte la traversée du NAT pour IPsec, [KHSV02] propose plusieurs ajouts à IKE :

- Une implémentation supportant la traversée du NAT doit utiliser pour port destination des paquets IKE le port source depuis lequel elle a reçu des paquets IKE en provenance du partenaire.
- Un champ `vendor_id` spécifique a été réservé afin que deux implémentations supportant la traversée du NAT se reconnaissent.
- IKE est enrichi d'un champ `NAT_D (NAT Discovery)`. Ce champ permet de convoyer des hash des adresses IP et des ports des partenaires. Si la vérification d'un hash à partir d'une adresse d'un paquet reçu échoue, alors au moins un NAT sépare les partenaires. Suivant le hash mis en défaut, on peut déterminer lequel des deux est derrière le NAT (ou s'ils le sont tous les deux). Dès lors qu'un NAT est détecté, il faut envoyer des paquets ``NAT-keepalive'' (voir ci-dessus), mais avant tout basculer depuis le port classique de IKE (500) vers un port flottant (4500) afin d'évincer les mécanismes d'aide du NAT (voir plus haut).
- En phase 2, les partenaires peuvent s'envoyer leurs adresses IP réelles, grâce à un champ `NAT_OA (Original Address)`, afin de pouvoir apporter les corrections nécessaires aux sommes de contrôles des protocoles de transport encapsulés par IPsec en mode transport.

Ces ajustements sur le comportement de IKE introduisent aussi quelques nouvelles failles de sécurité :

- Un partenaire A1 camouflé par un NAT peut ne pas révéler son adresse IP à son partenaire B. L'utilisation d'un secret partagé dans cette situation est particulièrement problématique, car tous les hôtes dissimulés par le NAT et susceptibles de communiquer avec B doivent alors posséder ce même secret partagé. Cette situation est très dangereuse (les secrets partagés ne sont pas adaptés pour un passage à l'échelle) et doit donc être évitée.
- Le hash sur les coordonnées internes (adresse IP et port) ne permet pas une dissimulation efficace de l'adresse IP interne. En effet, avec un espace d'adressage sur 32 bits, une recherche exhaustive est aisée. Il n'est cependant pas évident que la confidentialité de l'adresse interne soit une question d'importance.

- Les champs `NAT_D` et `vendor_id` (voir plus haut) de ISAKMP ne sont pas protégés. Un attaquant peut donc enlever, modifier, ajouter de tels champs en vue de provoquer un déni de service ou une réduction de la bande passante.
- Un attaquant peut provoquer un déni de service en modifiant les adresses IP et les ports d'authentiques paquets issus du réseau privé. Le partenaire extérieur sera alors obligé de suivre ces changements et d'envoyer ses paquets vers une mauvaise destination. Cela nécessite cependant de l'attaquant une certaine continuité d'action : chaque paquet valide réussissant à sortir du réseau privé sans modification par l'attaquant tend à rétablir la situation dans le bon sens.

Les difficultés posées par le NAT semblent en passe d'être résolues. Bien que les besoins présentés dans [AD02] soient cohérents, il est regrettable que l'utilisation conjointe du NAT et d'IPsec soit présentée dans ce draft comme un mécanisme de transition avant que le réseau ne passe totalement à IPv6. Cette phase de transition risque d'avoir une durée non négligeable, et il est probable que le réseau IPv4 existera toujours en parallèle.

CONCLUSION

Les incompatibilités présentées dans ce rapport témoignent pour la mise en application de pratiques de sécurité plus raisonnées : tenter de sécuriser un protocole tel que IP après son déploiement à grande échelle rend impossible l'éradication de ses tumeurs. L'explosion des nouveaux protocoles et mécanismes en coeur de réseau n'allège pas non plus la pression sur IPsec. Peut-être qu'une opportunité de saisir la bride dès le départ se présente avec IPv6 ?

RÉFÉRENCES

[AA02] **THREAT ANALYSIS OF THE DOMAIN NAME SYSTEM**
Date: Février 2002
Status: Internet Draft
Auteur(s): D. Atkins, R. Austein
(Source: http://www.ietf.org/internet-drafts/draft-ietf-dnsext-dns-threats-01.txt)

[AD02] **IPSEC-NAT COMPATIBILY REQUIREMENTS**
Date: 18 Août 2002
Status: Internet Draft
Auteur(s): B. Aboba, W. Dixon
(Source: http://www.ietf.org/internet-drafts/draft-ietf-ipsec-nat-reqts-02.txt)

[ADP99] **L'ACCÈS SÉCURISÉ AUX DONNÉES**
Date: Novembre 1999
Status: Publication JRES99
Auteur(s): S.Aumont, R.Dirlewanger, O. Porte
(Source: http://www.cru.fr/securite/crypto-jres99.pdf)

[AS02] **PREPARING FOR WIRELESS LANS**
Date: 2ème trimestre 2002
Status: Article, Packet Vol 14, N°2
Auteur(s): B. Alexander, S. Snow
(Source: issn://1535-2439)

[BIKS02] **ON THE USE OF SCTP WITH IPSEC**
Date: 22 Octobre 2002
Status: Internet Draft
Auteur(s): S. M. Bellovin, J. Ioannidis, A. D. Keromytis, R. R. Stewart
(Source: http://www.ietf.org/internet-drafts/draft-ietf-ipsec-sctp-04.txt)

[BKRS02] **IP SECURITY POLICY REQUIREMENTS**
Date: Août 2002
Status: Internet Draft
Auteur(s): M. Blaze, A. Keromytis, M. Richardson, L. Sanchez
(Source: http://www.ietf.org/internet-drafts/draft-ietf-ipsp-requirements-02.txt)

[Bou02] **PATCH TCP-MSS**
Status: Linux Kernel Patch
Auteur(s): Marc Boucher
(Source: http://www.netfilter.org/documentation/)

[Bra89] **REQUIREMENTS FOR INTERNET HOSTS -- COMMUNICATION LAYERS**
Date: Octobre 1989
Status: RFC 1122 (Standard)
Auteur(s): R. Braden
(Source: http://www.ietf.org/rfc/rfc1122.txt)

[CFSD90] **A SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP)**
Date: 1 Mai 1990
Status: RFC 1157 (Standard)
Auteur(s): J. Case, M. Fedor, M. Schoffstall, J. Davin
(Source: http://www.ietf.org/rfc/rfc1157.txt)

[cisco.com] **WWW.CISCO.COM**
Status: Site Web
(Source: http://www.cisco.com)

[Clu01] **ETUDES ET STATISTIQUES SUR LA SINISTRALITÉ INFORMATIQUE EN FRANCE**
Date: 2001
Status: Rapport CLUSIF
Auteur(s): CLUSIF
(Source: http://www.clusif.asso.fr/fr/production/sinistralite/docs/etude2001.pdf)

[DD01] **CACHE CORRUPTION ON MICROSOFT DNS SERVERS**
Date: 31 août 2001
Status: CERT Incident Note
Auteur(s): C. Dougherty, R. Danyliw
(Source: http://www.cert.org/incident_notes/IN-2001-11.html)

[DH98] **INTERNET PROTOCOL, VERSION 6 (IPV6) SPECIFICATION**
Date: Décembre 1998
Status: RFC 2460 (Draft Standard)
Auteur(s): S. Deering, R. Hinden
(Source: http://www.ietf.org/rfc/rfc2460.txt)

[Eas99] **DOMAIN NAME SYSTEM SECURITY EXTENSIONS**
Date: Mars 1999
Status: RFC 2535 (Proposed Standard)
Auteur(s): D. Eastlake
(Source: http://www.ietf.org/rfc/rfc2535.txt)

[FS00] **A CRYPTOGRAPHIC EVALUATION OF IPSEC**
Date: 2000
Status: Publication
Auteur(s): N. Ferguson, B. Schneier
(Source: http://www.counterpane.com/ipsec.pdf)

[H.323.00] **PACKET-BASED MULTIMEDIA COMMUNICATIONS SYSTEMS**
Date: Novembre 2000
Status: H.323 (Norme ITU)

[H3KP02-1] **DESIGN RATIONAL FOR IKEV2**
Date: Février 2002
Status: Internet Draft
Auteur(s): D. Harkins, C. Kaufman, T. Kivinen, S. Kent, R. Perlman
(Source: www.ietf.org/internet-drafts/draft-ietf-ipsec-ikev2-rationale-00.txt expired))

[H3KP02-2] **PROPOSAL FOR THE IKEV2 PROTOCOL**
Date: Avril 2002
Status: Internet Draft
Auteur(s): D. Harkins, C. Kaufman, S. Kent, T. Kivinen, R. Perlman
(Source: www.ietf.org/internet-drafts/draft-ietf-ipsec-ikev2-02.txt (expired))

[HC98] **THE INTERNET KEY EXCHANGE (IKE)**
Date: Novembre 1998
Status: RFC 2409 (Proposed Standard)
Auteur(s): D. Harkins, D. Carrel
(Source: http://www.ietf.org/rfc/rfc2409.txt)

[Hof02] **FEATURES OF PROPOSED SUCCESSORS TO IKE**
Date: 31 Mai 2002
Status: Internet Draft
Auteur(s): P. Hoffman
(Source: http://www.ietf.org/internet-drafts/draft-ietf-ipsec-soi-features-01.txt)

[Hou00] **DENIAL OF SERVICE ATTACKS USING NAMESERVERS**
Date: 28 Avril 2000
Status: CERT Incident Note
Auteur(s): Kevin Houle
(Source: http://www.cert.org/incident_notes/IN-2000-04.html)

[HSSVD02] **UDP ENCAPSULATION OF IPSEC PACKETS**
Date: Juin 2002

Status: Internet Draft
Auteur(s): A. Huttunen, B. Swander, M. Stenberg, V. Volpe, L. DiBurro
(Source: http://www.ietf.org/internet-drafts/draft-ietf-ipsec-udp-encaps-03.txt)

[Isp02] **RECOMMENDATIONS FOR INCREASING THE SECURITY OF DOMAIN NAME RESOLUTION AND THE RELIABILITY OF TRAFFIC ROUTING ON THE INTERNET**
Date: 1 Mai 2002
Status: Rapport Gouvernement US
Auteur(s): ISP Working Group
(Source: ups/rp-http://www1.ietf.org/mail-archive/working-grosec/current/msg00220.html)

[KA98-1] **SECURITY ARCHITECTURE FOR THE INTERNET PROTOCOL**
Date: Novembre 1998
Status: RFC 2401 (Proposed Standard)
Auteur(s): S. Kent, R.Atkinson
(Source: http://www.ietf.org/rfc/rfc2401.txt)

[KA98-2] **IP AUTHENTICATION HEADER**
Date: Novembre 1998
Status: RFC 2402 (Proposed Standard)
Auteur(s): S. Kent, R. Atkinson
(Source: http://www.ietf.org/rfc/rfc2402.txt)

[KA98-3] **IP ENCAPSULATING SECURITY PAYLOAD (ESP)**
Date: Novembre 1998
Status: RFC 2406
Auteur(s): S. Kent, R. Atkinson
(Source: http://www.ietf.org/rfc/rfc2406.txt)

[Kal00-1] **INTERNET RELAY CHAT : CLIENT PROTOCOL**
Date: Avril 2000
Status: RFC 2812 (Informational)
Auteur(s): C. Kalt
(Source: http://www.ietf.org/rfc/rfc2812.txt)

[Kal00-2] **INTERNET RELAY CHAT : SERVER PROTOCOL**
Date: Avril 2000
Status: RFC 2813 (Informational)
Auteur(s): C. Kalt
(Source: http://www.ietf.org/rfc/rfc2813.txt)

[Ken02] **ESPV2, AHV2, 2401BIS**
Date: 16 Avril 2002
Status: Présentation IETF
Auteur(s): S. Kent
(Source: http://www.ietf.org/proceedings/02mar/slides/ipsec-4/index.html)

[KHSV02] **NEGOTIATION OF NAT-TRAVERSAL IN THE IKE**
Date: 24 Juin 2002
Status: Internet Draft
Auteur(s): T.Kivinen, A. Huttunen, B. Swander, V. Volpe
(Source: http://www.ietf.org/internet-drafts/draft-ietf-ipsec-nat-t-ike-03.txt)

[KMS95] **THE ESP DES-CBC TRANSFORM**
Date: Août 1995
Status: RFC 1829 (Proposed Standard)
Auteur(s): P. Karn, P. Metzger, W. Simpson
(Source: http://www.ietf.org/rfc/rfc1829.txt)

[Kry02] **PROTOCOL AND PORT FIELDS IN SELECTORS**
Date: 9 Octobre 2002
Status: IPsec WG mail archive
Auteur(s): A. Krywaniuk
(Source: http://www.atm.tut.fi/list-archive/ipsec/msg00839.html)

[MD98] **THE ESP DES-CBC CIPHER ALGORITHM WITH EXPLICIT IV**
Date: Novembre 1998
Status: RFC 2405 (Proposed Standard)
Auteur(s): C. Madson, N. Doraswamy
(Source: http://www.ietf.org/rfc/rfc2405.txt)

[MSST98] **INTERNET SECURITY ASSOCIATION AND KEY MANAGEMENT PROTOCOL (ISAKMP)**
Date: Novembre 1998
Status: RFC 2408 (Proposed Standard)
Auteur(s): D. Maughan, M. Schertler, M. Schneider, J. Turner
(Source: http://www.ietf.org/rfc/rfc2408.txt)

[PADZB01] **SECURING L2TP USING IPSEC**
Date: Novembre 2001
Status: RFC 3193 (Proposed Standard)
Auteur(s): B. Patel, B. Aboba, W. Dixon, G. Zorn, S.Booth
(Source: http://www.ietf.org/rfc/rfc3193.txt)

[Pip98] **THE INTERNET IP SECURITY DOMAIN OF INTERPRETATION FOR ISAKMP**
Date: Novembre 1998
Status: RFC 2407 (Proposed Standard)
Auteur(s): D. Piper
(Source: http://www.ietf.org/rfc/rfc2407.txt)

[PM02] **ANALYSE CRITIQUE DES PROTOCOLES IPSEC**
Date: Mai 2002
Status: Rapport
Auteur(s): JJ. Puig, M. Laurent-Maknavicius
(Source: http://www-lor.int-evry.fr/~puig/travaux/)

[Pos80] **USER DATAGRAM PROTOCOL**
Date: 28 Août 1980
Status: RFC 768 (Standard)
Auteur(s): J. Postel
(Source: http://www.ietf.org/rfc/rfc768.txt)

[Pos81-1] **INTERNET CONTROL MESSAGE PROTOCOL**
Date: 1er Septembre 1981
Status: RFC 792 (Standard)
Auteur(s): J. Postel
(Source: http://www.ietf.org/rfc/rfc792.txt)

[Pos81-2] **TRANSMISSION CONTROL PROTOCOL**
Date: 1 Septembre 1981
Status: RFC 793 (Standard)
Auteur(s): J. Postel
(Source: http://www.ietf.org/rfc/rfc793.txt)

[PR85] **FILE TRANSFER PROTOCOL (FTP)**
Date: 1 Octobre 1985
Status: RFC 959 (Standard)
Auteur(s): J. Poster, J. Reynolds
(Source: http://www.ietf.org/rfc/rfc959.txt)

[RFB01] **THE ADDITION OF EXPLICIT CONGESTION NOTIFICATION (ECN) TO IP**
Date: Septembre 2001
Status: RFC 3168 (Proposed Standard)
Auteur(s): K. Ramakrishnan, S. Floyd, D. Black
(Source: http://www.ietf.org/rfc/rfc3168.txt)

[Ros02] **THE SESSION INITIATION PROTOCOL (SIP) UPDATE METHOD**

Date: Octobre 2002
Status: RFC 3311 (Proposed Standard)
Auteur(s): J. Rosenberg
(Source: http://www.ietf.org/rfc/rfc3311.txt)

[SCTP00] **STREAM CONTROL TRANSMISSION PROTOCOL**
Date: Octobre 2000
Status: RFC 2960 (Standards Track)
Auteur(s): R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, V. Paxson
(Source: http://www.ietf.org/rfc/rfc2960.txt)

[SE01] **TRADITIONAL IP NETWORK ADDRESS TRANSLATOR (TRADITIONAL NAT)**
Date: Janvier 2001
Status: RFC 3022 (Informational)
Auteur(s): P. Srisuresh, K. Egevang
(Source: http://www.ietf.org/rfc/rfc3022.txt)

[SH99] **IP NETWORK ADDRESS TRANSLATOR (NAT) TERMINOLGY AND CONSIDERATIONS**
Date: Août 1999
Status: RFC 2663 (Informational)
Auteur(s): P. Srisuresh, M. Holdrege
(Source: http://www.ietf.org/rfc/rfc2663.txt)

[SIP02] **SIP : SESSION INITIATION PROTOCOL**
Date: Juin 2002
Status: RFC 3261 (Proposed Standard)
Auteur(s): J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler
(Source: http://www.ietf.org/rfc/rfc3261.txt)

[VTRB97] **DYNAMIC UPDATES IN THE DOMAIN NAME SYSTEM (DNS UPDATE)**
Date: Avril 1997
Status: RFC 2136 (Proposed Standard)
Auteur(s): P. Vixie,S. Thomson, Y. Rekhter, J. Bound
(Source: http://www.ietf.org/rfc/rfc2136.txt)

[YHK95] **LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL**
Date: Mars 1995
Status: RFC 1777 (Draft Standard)
Auteur(s): W. Yeong, T. Howes, S. Kille
(Source: http://www.ietf.org/rfc/rfc1777.txt)

Jean-Jacques - Puig

Octobre 2002