Étude des Interactions IPsec / DNS

Logiciels-
Réseaux

Jean-Jacques Puig
Maryline Laurent-Maknavicius

04002-LOR
2004

Mars 2004

LES INTERACTIONS IPSEC/DNS

---oOo---

Abstract : TCP/IP connections involve a number of related protocols which are usually misunderstood. This report studies the interactions between IPsec and DNS, and shows that using one of them may compromise the other.

Key words: IPsec, DNS

Résumé : La « connectivité TCP/IP » (les accès réseaux sont parfois qualifiés ainsi) implique l'utilisation de nombreux protocoles connexes, souvent mal compris. Dans ce rapport, nous nous intéressons aux interactions entre IPsec et DNS. Toute utilisation non concertée de l'un peut compromettre l'autre, comme nous allons le voir.

Mots-clés : IPsec, DNS

---oOo—

Jean-Jacques Puig
Doctorant
Institut National des Télécommunications
Département LOR
9 rue Charles Fourier 91011 Evry cedex
E-mail: Jean-Jacques.Puig@int-evry.fr

Maryline Laurent-Maknavicius
Maître de conférernce
Institut National des Télécommunications
Département LOR
9 rue Charles Fourier 91011 Evry cedex
E-mail: Maryline.Maknavicius@int-evry.fr

Mars 2004

Les interactions IPsec/DNS

Jean-Jacques Puig

Jean-Jacques.Puig@int-evry.fr

INT

Maryline Laurent-Maknavicius

Maryline.Maknavicius@int-evry.fr

INT

Mars 2004

SOMMAIRE

1. Typologie des Faiblesses DNS

Une consultation approfondie de la littérature traitant du DNS nous a permis de construire la typologie suivante des faiblesses du DNS. Ces faiblesses permettent d'utiliser le DNS pour mener des attaques de plus grande envergure contre un réseau.

1.1. Cartographie du Réseau

Cette cartographie est notamment permise grâce aux opérations de transferts de zone (requêtes DNS de type AXFR ou IXFR). Ces opérations entrent en jeu dans le processus de réplication des zones sur les DNS secondaires. On rencontre principalement deux défauts liés à leur mise en oeuvre.
Le premier est l'absence relativement fréquente de filtrage sur les adresses autorisées à déclencher un transfert:

```
dig int-evry.fr. @pompei.int-evry.fr. AXFR
```

La commande précédente retourne le détail de la zone int-evry.fr; pompei est un des serveurs de noms de cette zone. Le transfert est autorisé pour tout le domaine int-evry.fr.
Le second défaut, persistant même lorsque le filtrage est correctement configuré, est que ces opérations de transferts ne sont pas protégées par cryptographie. Il est alors relativement facile d'usurper l'identité d'un DNS secondaire légitime et de déclencher le transfert ou encore d'écouter sur le réseau les données transférées lors des mises à jour et qui ne sont pas chiffrées.
Les transferts de zone vers des systèmes n'appartenant pas au même domaine semblent désormais proscrits dans les configurations par défaut des serveurs DNS. Cependant, localement, du fait de cette attaque, l'intrus acquiert une connaissance plus précise des systèmes importants du réseau attaqué. Si de plus le réseau effectue de la translation d'adresse, cette méthode peut réussir là où une « échographie » externe aurait échoué.
Concernant le transfert de zone, nous avons donc un défaut fréquent de configuration au niveau du contrôle d'accès sur ce type particulier de requête, et une faiblesse globale de la sécurité des communications DNS (non authentifiées, non chiffrées).
TSIG, qui fait partie de DNSsec, permet d'authentifier les acteurs d'un transfert de zone, et prévient donc l'usurpation, en supposant qu'un contrôle d'accès sur les adresses soit établi. En revanche, la communication n'est pas protégée contre une écoute passive.

1.2. Les Dénis de Service

Ces dénis de service sont de deux types, suivant leur cible initiale.
Si la cible initiale est un système particulier, l'attaquant peut se servir du déséquilibre de taille entre les requêtes DNS et leurs réponses, et interroger de nombreux serveurs DNS en usurpant l'identité du système ciblé. Le rapport de tailles, appelé « facteur d'amplification », permet littéralement à un attaquant disposant d'une bande passante relativement faible de saturer un lien de plus grande capacité. Cette attaque rentre dans la catégorie des attaques par réflexion (bien qu'il ne s'agisse pas d'une réflexion au sens cryptographique du terme) et peut être menée avec tout protocole basé sur UDP (ou non connecté) et présentant des différences fortes entre le volume des requêtes et celui des réponses (ex: les protocoles de jeux en réseaux). En utilisant TSIG, il est possible d'empêcher l'utilisation des requêtes de type AXFR ou IXFR pour provoquer des dénis de service. En effet, les transferts de zone intéressent particulièrement les attaquants, de par leur facteur d'amplification extrêmement élevé.
Si la cible initiale du déni de service est un serveur DNS, la résolution de noms peut être compromise. Si le serveur neutralisé est un serveur secondaire, cela aura une incidence réduite (un autre serveur de nom sera interrogé après l'expiration du temps d'attente de la réponse). Si le serveur neutralisé est un serveur

primaire, les zones dont il se charge peuvent commencer à disparaître dans les limbes (i.e: l'information est toujours accessible dans les secondaires et les caches, mais disparaîtra après un certain temps). Enfin, si le serveur DNS est un « achemineur », les systèmes du réseau dont il a la charge risquent de ne plus avoir accès à la résolution de noms si ils ne connaissent que ce serveur DNS; cela peut engendrer un déni *d'utilisation* du service (i.e: ce sont les clients qui sont bloqués, et non les serveurs).

1.3. L'Usurpation

L'usurpation DNS peut prendre deux formes, selon le procédé utilisé pour l'attaque. Dans tous les cas, il convient de retenir qu'il est possible d'affecter un système sans l'attaquer directement, en provoquant la corruption d'un cache DNS intermédiaire auquel il serait amené à s'adresser, directement ou indirectement. Cela rend simultanément les attaques plus faciles à mener (plus il y a de cibles, plus il y a de chances d'en trouver une qui soit faible), et réduit leur traçabilité.

- L'usurpation d'identité DNS est la construction d'une réponse forgée qui prend le pas sur la réponse légitime. Pour ce faire, il est nécessaire de déterminer la valeur de l'identifiant inclus dans la requête et bien sûr d'usurper l'adresse réseau du serveur. Si de plus la requête requiert l'utilisation de TCP, il convient aussi de déterminer le numéro de séquence (mais l'utilisation de TCP est sensée être anecdotique). Dans tous les cas, cette attaque est difficile à mener en aveugle, mais s'avère en revanche triviale si l'attaquant peut observer les paquets et répondre avant le serveur. Les logiciels dnsspoof et ADMsniffID offrent des facilités pour la mise en oeuvre de cette technique.
- L'empoisonnement de cache DNS requiert, quant à lui, la main mise de l'attaquant sur un serveur malin. Les logiciels poisonivy et dnsa fournissent les fonctionnalités nécessaires pour un tel serveur. Il faut alors trouver une méthode (courriel, forum de discussion, téléphone, publicité, page web, etc.) pour forcer un client à consulter un nom de la zone maligne. Des champs d'enregistrements additionnels ou spéciaux sont alors ajoutés aux réponses légitimes et permettent à l'attaquant d'informer le cache DNS sur *d'autres zones*. Les serveurs DNS actuels refusent désormais pour la plupart de se laisser berner par de telles manipulations.

DNSsec apporte une protection contre ces attaques: il devient possible de vérifier que l'enregistrement a été signé par une autorité adéquate. L'intégrité du système intermédiaire acheminant l'information est alors une question secondaire.

1.4. Les Failles Logicielles

Comme tout programme, les serveurs et les « résolveurs » DNS peuvent comporter des failles, lesquelles constituent un tremplin pour l'exécution locale de code malin passé en paramètre via les données émises dans le protocole; le « ver SMTP » marque encore les esprits, et le DNS constitue une cible idéale pour de nombreuses raisons:

- Les systèmes DNS sont extrêmement nombreux: ils offrent une population très large de vecteurs pour l'infection.
- Un système DNS neutralisé ou corrompu est source de nombreux dommages (cf. parties précédentes).
- Les logiciels utilisés pour les DNS ont encore une diversité réduite. Cela simplifie le travail des délinquants et augmente les risques de mort ou de mutilation de la population de programmes DNS. Le logiciel bind couvre une part majeure du terrain, le reste étant le fief de windows. Cette absence de diversité est souvent soulignée comme un risque majeur.
- Les échanges courts du DNS s'effectuent avec UDP et réduisent donc la traçabilité des attaquants et des systèmes « de rebonds » infectés.

- Enfin, sur certains systèmes, la somme de contrôle d'UDP n'est pas contrôlée (c'est un comportement assez répandu). Un enregistrement DNS, transféré via UDP sur un réseau non fiable et sans vérification de somme de contrôle, peut « muter » du fait des erreurs de transmission. Cet enregistrement peut alors infecter des caches avec des informations erronées: il s'agit de l'apparition d'un virus « ex-machina ». Bien que la probabilité d'occurence d'un tel évènement sur une requête DNS soit très faible, il convient de considérer la quantité importante de transactions DNS qui sont menées à chaque instant. Ce type d'évènement a sans doutes déjà eu lieu, et le concept mérite d'être mentionné.

2. DNS affecte IPsec

Dans cette section, nous étudions de quelles manières une action hostile sur le DNS ou via le DNS peut compromettre l'utilisation d'IPsec. Reprenons donc les failles présentées dans notre typologie:

2.1. Cartographie du Réseau

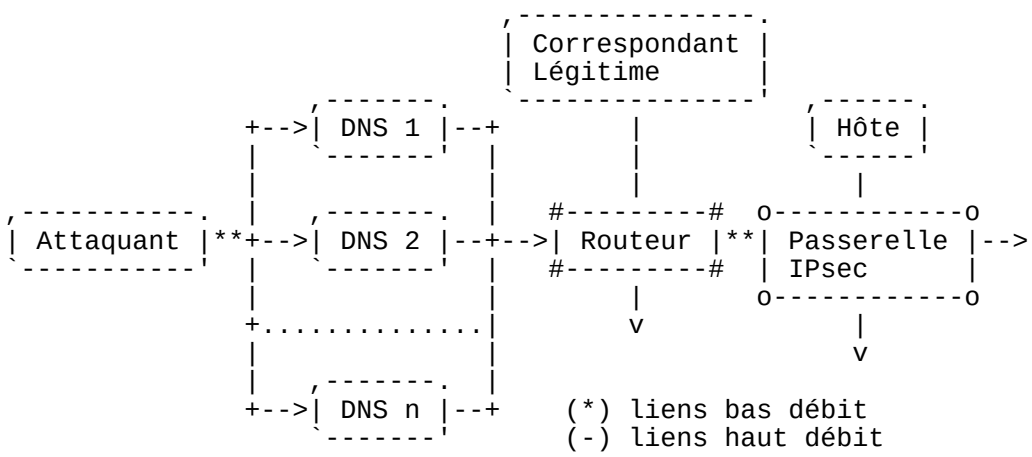
Il ne semble pas évident, de prime abord, de déterminer dans quelles mesures une cartographie de réseau obtenue via DNS peut affecter IPsec. En effet, par définition, une carte est un objet passif; un contexte de topographie (structure du réseau) et d'intention (cible de l'attaquant) peut seul permettre d'en comprendre l'usage. De plus, l'interprétation (les noms DNS) a un poids non négligeable dans la lecture d'une carte.
Tout au plus pouvons-nous supposer que l'attaquant apprend par cartographie l'adresse des systèmes utilisant IPsec. Peut-être peut-il aussi trouver ainsi des routes alternatives ? La latitude sur laquelle IPsec peut se trouver affecté ne peut être déterminée a priori dans le cas général. En tout cas, l'examen de la zone int-evry.fr. n'a rien révélé d'intéressant.
Bien sûr, l'utilisation d'IPSECKEY changerait radicalement les règles du jeu en révélant les identités des passerelles, leurs clefs publiques, leurs priorités. Mais ces informations ne sont elles pas sensées être publiques ? La visibilité des données est un problème général d'Internet; en ce qui concerne IPSECKEY, l'objectif officieux étant le support du chiffrement opportuniste, ces données ne peuvent être que publiques. Tout au plus pouvons-nous présenter l'utilisation des adresses « anycast » comme une prémisse d'alternative à l'utilisation d'IPSECKEY ? Ces adresses apportent une solution qui semble plus élégante, bien que moins flexible, pour déterminer les adresses des passerelles sans trop en révéler.
Cependant, cela ne signifie pas qu'aucune stratégie de sécurité ne puisse être établie au niveau de la cartographie DNS dans la réalité. Au delà de l'usage de DNSsec, et d'une configuration appropriée du contrôle d'accès, une analyse par l'administrateur du site de la topologie du réseau et des informations contenues dans le DNS peut lui permettre d'établir une stratégie correcte d'utilisation d'IPsec dans son cas particulier et de construire des scénarii de déception des attaquants (avec par exemple la mise en place de « pots de miel » bien situés, avec des noms appropriés dans le DNS et des adresses dans la même zone que les serveurs).

2.2. Les Dénis de Service

2.2.1. En utilisant DNS comme « zombi »

Évidemment, utiliser le système de noms comme intermédiaire pour saturer un lien sur lequel IPsec pourrait être utilisé ne semble pas relever d'une interaction directe. Cependant, il est intéressant d'observer ce type de situation à travers le scénario qui sera présenté dans cette section.
Ainsi, sur le schéma suivant, IPsec subit l'inondation sans possibilité d'action:

Figure 1. Déni de service par inondation de la liaison entre routeur et passerelle IPsec



L'attaquant dispose d'un lien bas débit, et sollicite de nombreux serveurs DNS en usurpant, par exemple, l'identité de l'hôte. Le facteur d'amplification permet d'engorger la liaison entre le routeur et la passerelle IPsec.

Notons cependant que si la liaison entre le routeur et la passerelle de sécurité est à double sens (i.e: le médium de transport des paquets n'est pas partagé entre les émetteurs), l'hôte parviendra toujours a envoyer des paquets vers son correspondant. Seules les réponses de ce dernier risquent d'être perdues. De manière générale, l'architecture du réseau, la capacité des liens ainsi que la position des acteurs jouent pour beaucoup sur l'efficacité et les limites de l'attaque. L'attaquant procède donc au préalable à une phase d'évaluation des routes, des liens, des « zombis » possibles...

Pour procéder d'une manière simple à l'évaluation de la faisabilité d'un engorgement, nous nous sommes reportés sur la simulation d'une inondation via des requêtes ICMP de type Echo. En effet, la mise en place de systèmes DNS intermédiaires serait coûteuse en temps, et par ailleurs, des outils tels que dig ne semblent pas en mesure de procéder aux requêtes de transferts de zone en UDP ([1]). La commande lancée par notre « attaquant virtuel » est donc la suivante:

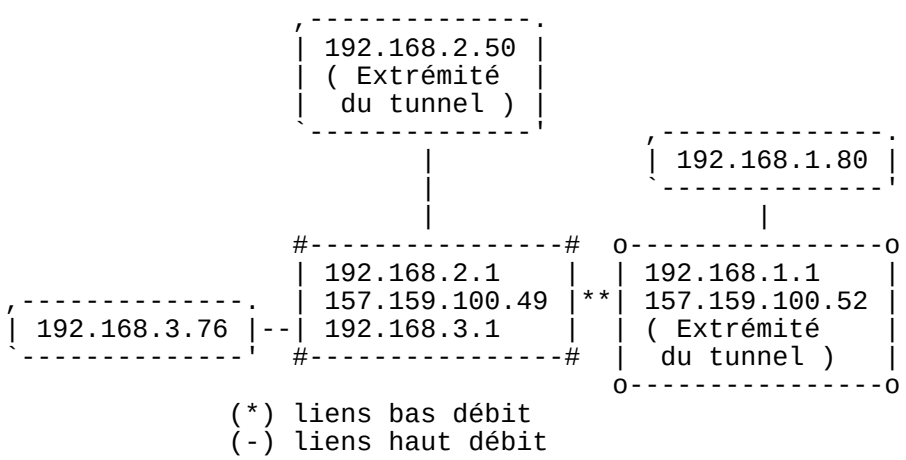
`ping -n -q -f -Q 0x08 -M dont -l nombre -s taille destination`([2]).

Dans le cadre de notre manipulation:

- nombre = 16000000
- taille = 1472 (le maximum possible compte tenu de l'unité maximale de transmission sur les réseaux traversés).
- destination = 192.168.1.80

Avec la maquette suivante (tous les systèmes sont dotés du noyau Linux 2.4.23 et de FreeSwan 2.04):

Figure 2. Congestion d'un lien



L'attaquant, ou plutôt le système « de rebond », est simulé par 192.168.3.76, qui dispose d'une bande passante supérieure à celle disponible entre 157.159.100.49 et 157.159.100.52. Afin de simuler le lien bas débit, nous utilisons sur l'interface appropriée du routeur la mécanique de contrôle de trafic de noyau:

```
tc qdisc add dev eth2 root handle 1: cbq avpkt 1000 bandwidth 1mbit
tc class add dev eth2 parent 1: classid 1:1 cbq rate 64kbit \
  allot 1500 prio 5 bounded isolated
tc filter add dev eth2 parent 1: protocol ip prio 16 u32 \
  match ip dst 157.159.100.52 flowid 1:1 ([3])
```

Les résultats sont sans appel: une fois l'attaque lancée, la session de transfert de fichiers (un simple scp) entre 192.168.2.50 et 192.168.1.80 se fige. Toute nouvelle ouverture de session est impossible, et IKE échoue à la reconstruction des associations de sécurité. Une ironie de cette attaque, lorsqu'elle fait appel au DNS comme « zombi », est qu'elle compromet aussi les résolutions DNS des systèmes impliqués, et aboutit donc à des effets similaires à ceux que nous allons présenter dans la partie suivante.

2.2.2. En bloquant DNS

Pour déterminer de quelles façons un blocage DNS pourrait affecter IPsec, il faudrait avant tout test mener une analyse en boîte blanche des implémentations, et tracer les appels aux différentes bibliothèques de résolution de noms. Un tel travail nécessite des modifications *majeures* du système, mais se trouve être le seul pouvant justifier de la validité des tests.
Une source d'information alternative et partielle est apportée par l'analyse des options de configuration du logiciel. Cette démarche est cependant valable: elle a le mérite de souligner les possibilités d'action de l'administrateur qui ne désire pas modifier trop en profondeur son système.
Initialement, notre administrateur (sous Linux 2.4.23 avec FreeSwan 2.04), peut configurer IPsec via:

- La configuration du noyau (fichier .config).
- Le fichier ipsec.secrets
- Le fichier ipsec.conf

Par ailleurs, il peut agir sur la configuration de la résolution de noms de différentes manières, les plus classiques passant par la modification des fichiers nsswitch.conf, resolv.conf ou hosts. On ne retiendra cependant ici que les aspects liés à la configuration d'IPsec, et on supposera une configuration « classique » de la résolution de noms.

2.2.2.1. .config

Les options de configuration du noyau relevant d'IPsec sont les suivantes:

```
CONFIG_IP_NF_MATCH_AH_ESP
CONFIG_IPSEC_IPIP
CONFIG_IPSEC_AH
CONFIG_IPSEC_AUTH_HMAC_MD5
CONFIG_IPSEC_AUTH_HMAC_SHA1
CONFIG_IPSEC_ESP
CONFIG_IPSEC_ENC_3DES
CONFIG_IPCOMP
CONFIG_IPSEC_DEBUG
CONFIG_IPSEC_REGRESS
```

Aucune de ces options n'est directement liée au DNS.

2.2.2.2. ipsec.secrets

Ce fichier héberge les clefs privées (RSA) et les clefs partagées (PSK) utilisées par IKE. Le fichier est structuré en une succession d'enregistrements associant d'une part un index et de l'autre la clef et le type de la clef. Les types pris par l'index sont tout à fait intéressants pour notre travail, mais précisons tout d'abord que l'index peut être omis. La clef liée à un enregistrement sans index est celle qui s'applique par défaut lorsqu'aucune correspondance plus précise n'a pu être établie. La configuration par défaut ne contient donc souvent que cet enregistrement, avec une clef RSA, et il serait sage que la plupart des administrateurs s'en contentent.
Par ailleurs, un index, lorsqu'il est mentionné, est en réalité une liste d'identités, ces identités pouvant être des adresses IP, des noms de domaine totalement qualifiés résolus au chargement du fichier, des noms de domaine totalement qualifiés non résolus, ou les valeurs spéciales %any, %any6. Ces deux dernières valeurs constituent en fait des masques qui correspondent à n'importe quelle identité de type ID_IPV4_ADDR ou ID_IPV6_ADDR.
Pendant la négociation des associations de sécurité avec IKE, les partenaires sont amenés à décliner leur identité. Ce couple d'identités est recherché par un parcours itératif dans la liste des index afin de déterminer la correspondance la plus spécifique, qui fournit la clef appropriée (secrète ou partagée). Deux observations s'imposent alors à l'esprit:

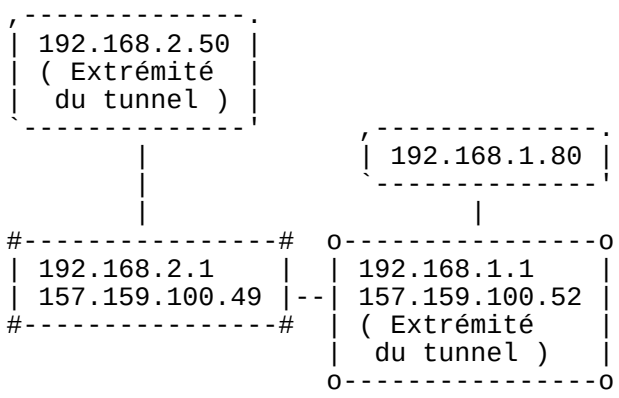
1. Lorsque RSA est utilisé, la clef secrète sélectionnée peut dépendre de l'identité du partenaire.
2. Lorsque PSK est utilisé, l'identité utilisée pour sélectionner la clef est nécessairement l'adresse IP; en effet, l'implémentation de IKE (pluto) ne propose que le mode incluant la protection de l'identité. Pour une clef partagée, il est impossible de déchiffrer l'identité protégée sans pouvoir déterminer la clef appropriée ! Cependant, d'un point de vue strictement politique, l'identité présentée, une fois déchiffrée est aussi prise en compte (cf. partie ipsec.conf). Par conséquent, les index pour les clefs partagées doivent toujours être résolus en adresses IP dès que le fichier est chargé par le répondeur.

Nous disserterons plus avant des conséquences d'une résolution erronée sur les index de ce fichier dans la partie traitant des risques liés à l'usurpation DNS et ne traiterons ici que des conséquences d'un déni de service sur la résolution.

2.2.2.2.1. Déni de service sur une résolution portant sur le nom du propriétaire du secret

Afin de tester les risques potentiels de la résolution de noms sur le choix des secrets, nous avons repris une partie de l'architecture précédente:

Figure 3. Déni de service



Nous avons choisi de nous attaquer plus précisément à 192.168.2.50 puisque, par nature, un télé-travailleur est dans un environnement plus vulnérable (le serveur DNS a pu lui être imposé). Pour simuler le déni de service DNS, nous avons tout simplement joué sur la présence ou l'absence du nom à résoudre dans le fichier /etc/hosts. La première version du fichier secret modifié est de la forme suivante:

```
roadwarrior.maquette.int-evry.fr: RSA {
  ... clef RSA 1 ...
  }

: RSA {
  ... clef RSA 2 ...
  }
```

Dans l'exemple, la passerelle 192.168.1.1 dispose de la clef publique associée à la clef RSA 1, pour une connexion pour laquelle l'identité du partenaire est 192.168.2.50.
Dans le fichier, la première clef est donc plus spécifique que la seconde, pour peu que la résolution de noms réussisse. Effectivement, quand la résolution réussit, cette clef est sélectionnée et le tunnel est établi avec la passerelle. Cependant, lorsque la résolution échoue, nous avons constaté que le tunnel était tout de même construit ! Ce fait surprenant est le fruit d'une logique un peu particulière: lorsque la résolution échoue, le premier index devient tout aussi « imprécis » que le deuxième, et le choix s'effectue alors de façon séquentielle, ce qui fait que le premier est quand même sélectionné ! Par conséquent, avec le fichier suivant:

```
: RSA {
  ... clef RSA 2 ...
  }

roadwarrior.maquette.int-evry.fr: RSA {
  ... clef RSA 1 ...
  }
```

Nous avons pu constater sur cette autre version du fichier que lorsque la résolution réussit, la clef RSA 1 est sélectionnée (l'index est plus spécifique) et le tunnel est établi. Lorsque la résolution échoue, les index sont équivalents, le choix est déterminé par la séquentialité, la clef RSA 2 est sélectionnée et la construction du tunnel échoue. D'où la conclusion générale sur le processus (cf. figure suivante):
Initialement, le fichier est constitué d'une suite d'enregistrements, observés séquentiellement (Liste 1).
L'identité de l'hôte est ensuite le critère le plus significatif aboutissant au classement préférentiel des clefs, vient ensuite l'identité du correspondant (Liste 2). Il s'agit donc d'un tri multi-champs.

Les index pour lesquels l'identité de l'hôte n'a pu être résolue perdent leur préférence (Liste 2'). Tout le talent de l'attaquant est de constituer la liste qui l'arrange en compromettant certaines résolutions.

Figure 4.

```
[1, IP, *, RSA]   [3, Nb, C, RSA]   [4, Nr, C, PSK]
[2, Nr, *, RSA]   [4, Nr, C, PSK]   [1, IP, *, RSA]
[3, Nb, C, RSA]   [1, IP, *, RSA]   [2, Nr, *, RSA]
[4, Nr, C, PSK]   [2, Nr, *, RSA]   [3, Nb, C, RSA]
[5, * , C, PSK]   [5, * , C, PSK]   [5, * , C, PSK]
[6, * , C, RSA]   [6, * , C, RSA]   [6, * , C, RSA]

    Liste 1           Liste 2           Liste 2'

Légende: [ordre, hôte, partenaire, clef]
         Nb : Nom bloqué (résolution contrée)
         Nr : Nom résolu
         IP : Adresse IP
         *  : Masque (Sélecteur par défaut)
```

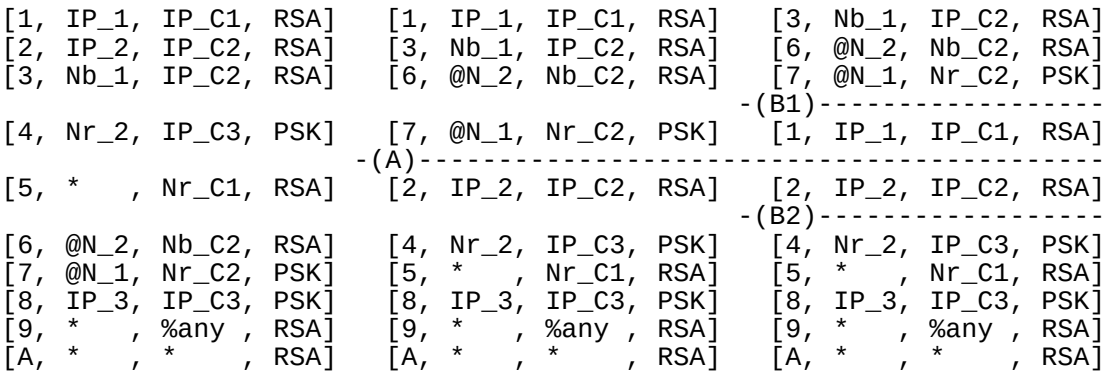
Ainsi, sur cet exemple, en bloquant la résolution pour le nom mentionné dans le troisième index ([3, Nb, C, RSA]), un attaquant favorise l'ordre de la Liste 2'. D'après cette liste, si l'hôte initie vers son partenaire C, il le fera en utilisant une clef partagée. En revanche, si C initie, l'hôte utilisera évidemment le type de clef approprié (vraisemblablement [1, IP, *, RSA] car C utilisera sans doutes RSA).
De là, trois conséquences sont possibles; elles peuvent advenir séparément ou conjointement:

1. L'attaquant a réussi à forcer la sélection d'une clef dont il sait qu'elle est plus faible; il obtient ainsi plus d'échantillons pour casser la clef.
2. L'attaquant a réussi à forcer l'utilisation d'une clef qu'il possède, par exemple une clef partagée qu'il possède de façon légitime dans le cadre d'une politique de groupe. Il peut soit décoder les messages issus de C (cas d'une clef RSA), soit s'intercaller dans la communication ou se faire directement passer pour C (cas d'une clef PSK).
3. Si C initie, la clef publique ou le secret partagé qu'il sélectionne ne correspond sans doutes pas à la contrepartie sélectionnée par l'hôte. Ce risque existe aussi si l'hôte initie. Les négociations échouent alors, et le tunnel ne peut être construit.

2.2.2.2.2. Déni de service sur une résolution portant sur le nom du correspondant

Dans le même ordre d'idée, la clef secrète est aussi sélectionnée en fonction du correspondant, bien que ce choix s'effectue bien sûr de façon moins prioritaire que par rapport à l'identité de l'hôte. Considérons donc un exemple de configuration, dans lequel l'hôte, ayant obtenu l'adresse IP_1 associée au nom Nb_1 désire contacter le correspondant C2; dans un premier temps, on suppose l'attaquant inactif:

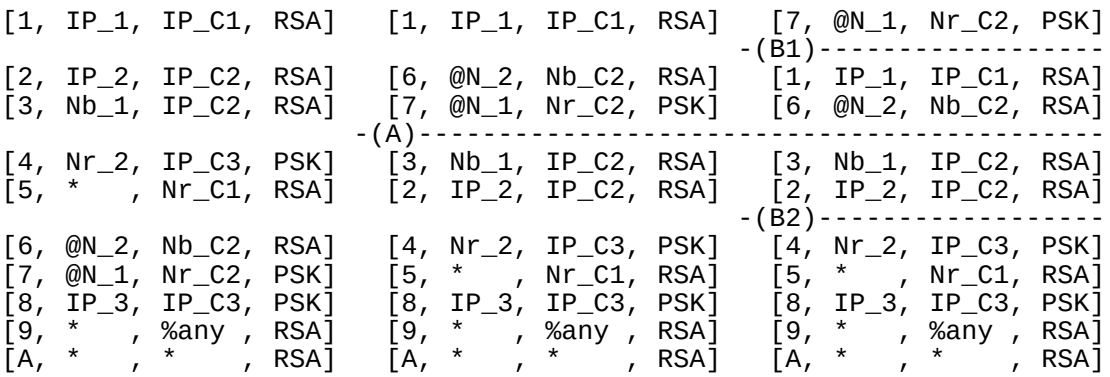
Figure 5.



Légende: [ordre, hôte, partenaire, clef]
Nb : Nom bloqué (résolution contrée)
Nr : Nom résolu
@N : Nom symbolique (non-résolu)
IP : Adresse IP
* : Masque (Sélecteur par défaut)
%any: n'importe quel correspondant

La Liste 1 constitue la succession des enregistrements telle que saisie dans le fichier. La Liste 2 présente le résultat du classement selon l'identité de l'hôte; la ligne (A) sépare les deux niveaux de précision. On remarque que @N_2, ne dépendant pas réellement de l'adresse ou du nom, est considéré comme précis. La Liste 3 présente enfin la dernière étape, dans laquelle pour chacun des niveaux de précision séparés par (A), on classe par degré de précision portant sur le destinataire, ce qui crée les sous-partitions représentées par les lignes (B1) et (B2). Si l'attaquant agit sur les résolutions Nb, on obtient finalement les résultats suivants:

Figure 6.



Légende: [ordre, hôte, partenaire, clef]
Nb : Nom bloqué (résolution contrée)
Nr : Nom résolu @N : Nom symbolique (non-résolu)
IP : Adresse IP
* : Masque (Sélecteur par défaut)
%any: n'importe quel correspondant

En bloquant les résolutions portant successivement sur Nb_1 et Nb_C2, l'attaquant arrive finalement, dans le cas présent, à forcer la sélection d'une clef partagée; les conséquences sont les mêmes que celles présentées précédemment.
Il convient cependant de relativiser tout ce qui vient d'être présenté car, dans la réalité:

- il y a peu de chances pour que la succession des enregistrements se prête aussi bien à une attaque,
- l'attaquant n'est pas sensé disposer d'informations sur cette liste,
- les fichiers ipsec.secrets sont en général bien moins complexes et se limitent dans la majorité des cas à une clef RSA.

Pour conclure au sujet de ce fichier ipsec.secrets, précisons tout de même qu'à défaut de présenter la nature des risques liés au DNS, la page de manuel mentionne tout de même leur existence.

```
In many cases it is a bad idea to use domain names because
the name server may not be running or may be insecure.
```

Par conséquent, les utilisateurs de FreeSwan sont tout de même prévenus qu'ils utilisent les noms à leurs risques et périls.

2.2.2.3. ipsec.conf

Le fichier ipsec.conf est le principal élément de configuration. Il accepte deux types de sections de configuration: configuration globale et connexions.

2.2.2.3.1. config

À l'heure actuelle, il n'existe qu'une section de ce type, et elle s'applique donc par défaut. Le seul paramètre lié à la résolution de noms dans cette section est myid. Ce paramètre définit l'identité par défaut annoncée dans IKE pour les connexions par défaut. Il peut être de l'un des types suivants:

1. Adresse IP (identité de type ID_IPV4_ADDR ou ID_IPV6_ADDR).
2. Nom DNS résolu au chargement (identité de type ID_IPV4_ADDR ou ID_IPV6_ADDR).
3. Nom DNS non résolu (identité de type ID_FQDN ou ID_USER_FQDN).
4. « Indéterminé »

Si la valeur est explicitement donnée par l'utilisateur, le type n°2 peut poser problème. Lors d'un déni de service sur la résolution de ce nom, les connexions utilisant cette identité pour IKE échouent. Plus précisément, la commande nous a renvoyé:

```
024 need --listen before --initiate
```

Ce message est issu du démon IKE, et recommande de donner la priorité à l'écoute (réponse aux sollicitations IKE des partenaires) plutôt qu'à l'initiation d'une connexion particulière. C'est une recommandation singulière, mais dans certains cas, il peut être nécessaire d'accepter la construction de diverses associations de sécurité avant qu'une connexion initiée depuis l'hôte puisse réussir (par exemple, une des connexions intermédiaires pourrait mener au DNS et permettre de résoudre l'identité...).
Par ailleurs, si le paramètre myid n'a pas de valeur explicitement précisée (Le type « Indéterminé » doit aussi être renseigné explicitement), le système tente de lui affecter une valeur automatique, dans l'ordre suivant:

1. Adresse IP de l'interface à laquelle est affectée la route par défaut, si un enregistrement IPSECKEY ([4]) a pu être trouvé dans la zone retour du DNS à cette adresse.

2. Nom de l'hôte (hostname), si un enregistrement IPSECKEY a pu être trouvé dans la zone directe du DNS pour ce nom.
3. « Indéterminé »

S'il parvient à bloquer les réponses DNS lors de ce processus, l'attaquant pourra provoquer un déni de service sur les connexions pour lesquelles une identité explicite n'a pas été donnée (avec la même erreur que ci-dessus). S'il ne bloque que la première requête (portant sur l'adresse), l'attaquant force peut-être ainsi l'utilisation d'une identité qui lui convient plus, soit parce qu'il a des facilités pour agir sur des requêtes portant sur la zone directe, soit parce que la clef publique de la zone directe a une contrepartie privée qui semble plus facile à casser. Il est généralement difficile de déterminer la motivation de l'attaquant dans un tel cas.
Enfin, le paramètre myid est l'identité qui s'applique par défaut pour les connexions implicites. Ces connexions étant basées sur de l'opportunisme, l'attaquant peut compromettre l'établissement d'un tunnel et forcer ainsi la transmission en clair, s'il s'agit de la politique par défaut élue localement.

2.2.2.3.2. conn

Les sections du type conn décrivent des connexions. Les paramètres concernés par le DNS sont les suivants:

- left (/right)
- leftsubnet (/rightsubnet)
- leftnexthop (/rightnexthop)
- leftid (/rightid)
- leftrsasigkey (/rightrsasigkey, leftrsasigkey2, rightrsasigkey2)

Chacune des variables ci-dessus se décline en une version « gauche » et une version « droite », qui correspondent aux deux côtés d'une connexion. Dans certains cas, la description est portable, c'est à dire que les deux extrémités sont soit équivalentes, soit en mesure d'identifier si elles constituent la partie gauche ou la partie droite de la connexion. Dans tous les cas asymétriques (ex: passerelle <=> télétravailleurs), la partie gauche correspond au système local, et la droite au système distant.
Le paramètre left décrit la « localisation » du système. Il peut être initialisé avec la valeur de l'adresse IP liée à l'interface de la route par défaut ou une adresse peut lui être explicitement affecté, soit directement, soit via un nom. Dans ce dernier cas, un déni sur le nom empêche la validation de la connexion (et donc la construction des associations de sécurité). Le démon IKE échoue alors en renvoyant la même erreur que précédemment. Enfin, ce paramètre peut aussi prendre différentes valeurs « joker »:

- %any est un masque associé par défaut à l'adresse IP de tout partenaire. Il est évidemment impossible d'être initiateur vers un tel partenaire.
- %group est un masque associé aux différents blocs CIDR mentionnés dans un fichier de groupes. Il s'agit d'une fonctionnalité tout à fait à même d'intéresser un opérateur. Elle est disponible depuis FreeSwan 2.00.
- %opportunistic signifie que le DNS sera interrogé pour obtenir les informations nécessaires sur le système. Plus précisément, si cette valeur est sensée correspondre à l'hôte local, il y a erreur de configuration, et IKE échoue avec le message:

```
022 "rwsg": we have no ipsecN interface for either end of this
            connection
```

En revanche, si cette valeur correspond à l'hôte distant, lorsque le partenaire initie, l'identité présentée par ce dernier provoque une recherche dans le DNS pour déterminer sa clef publique et s'il s'agit bien d'une passerelle. Si l'identité est de type ID_IPV4_ADDR ou ID_IPV6_ADDR, la zone inverse est consultée. Si le nom est de type ID_FQDN ou ID_USER_FQDN, la consultation a lieu dans la zone directe. Si un attaquant contre les résolutions, l'implémentation tente d'interroger à nouveau le DNS plusieurs fois. Pendant ce temps, l'offre IKE de l'initiateur reste sans réponse, ce qui le force lui-même à re-proposer son offre plusieurs fois, jusqu'à expiration du nombre maximum d'essais:

```
031 "rwsg" #1: max number of retransmissions (2) reached
               STATE_MAIN_I3. Possible authentication failure:
               no acceptable response to our first encrypted
               message
```

- %opportunisticgroup propose le même comportement que opportunistic, mais pour les blocs d'adresses CIDR configurés par l'utilisateur. Les conséquences en cas de dénis de service sur les résolutions DNS sont les mêmes.

Le paramètre leftsubnet permet de définir pour quel sous-réseau le système est passerelle de sécurité. Ainsi, sur la Figure 2, « Congestion d'un lien », 157.159.100.52 avait pour sous-réseau 192.168.1.0/24. Une autre description valide de notre sous-réseau est toto-le-sous-reseau/24, où toto-le-sous-reseau est un nom pour lequel le DNS renvoie une adresse dans le sous-réseau 192.168.1.0/24. Cette description particulière des sous-réseaux est permise par les fonctions du « résolveur ». Elle permet notamment d'avoir une certaine dynamique pour des réseaux dont le préfixe change (par exemple, des réseaux mobiles). Évidemment, si un attaquant contre la résolution de toto-le-sous-reseau, le démon IKE refuse le descriptif de la connexion:

```
021 no connection named "rwsg"
```

Plus précisément, c'est l'interface utilisateur au démon IKE (via un « socket » UNIX) qui se plaint:

```
ipsec__plutorun: whack error: "rwsg" does not look numeric and name
                 lookup failed "toto-le-sous-reseau/24"
ipsec__plutorun: ...could not add conn "rwsg"
```

La description de la connexion n'étant pas validée, il n'y a même pas de phase 1, et l'attaquant a ainsi empêché la construction du tunnel.
Le paramètre leftnexthop indique le prochain routeur vers lequel acheminer les paquets. Sa valeur est le plus souvent assignée par défaut et, d'expérience, c'est ainsi que les connexions fonctionnent le mieux. Si un nom est cependant utilisé pour décrire ce champ, un déni de service sur la résolution de ce nom empêche l'acheminement des paquets et donc la construction du tunnel. Dans notre configuration de test, nous avions leftnexthop=toto-le-routeur; un déni sur ce nom a provoqué l'erreur désormais bien connue:

```
024 need --listen before --initiate
```

ainsi que:

```
ipsec__plutorun: whack error: "rwsg" does not look numeric and name
                 lookup failed "toto-le-routeur/24"
ipsec__plutorun: ...could not add conn "rwsg"
```

Le paramètre leftid constitue l'identité du système pour IKE. Nous avons présenté plus haut le paramètre myid, issu de la section config. Le comportement de ces deux paramètres est rigoureusement identique vis-à-vis des interactions avec le DNS, par conséquent nous ne développerons pas plus sur ce point ici.
Le paramètre leftrsasigkey permet de définir la clef publique du système de « gauche » (et respectivement de droite pour rightrsasigkey). Les deux valeurs particulières %dnsonload et %dnsondemand permettent

de spécifier la recherche de la clef publique dans le DNS, respectivement au chargement du descriptif de la connexion ou à l'établissement de la connexion. Un déni sur ces requêtes aboutit aux erreurs suivantes:

```
021 no connection named "rwsg"
```

ainsi que:

```
ipsec__plutorun: 028 failure to fetch key for @toto-la-passerelle
                 from DNS: failure querying DNS for TXT of
                 toto-la-passerelle: Hostname lookup failure
ipsec__plutorun: 028 failure to fetch key for @toto-la-passerelle
                 from DNS: failure querying DNS for KEY of
                 toto-la-passerelle: Hostname lookup failure
ipsec__plutorun: ...could not add conn "rwsg"
```

Donc, la construction du tunnel est compromise.

2.3. L'Usurpation

De part les mécanismes mis en oeuvre, cette partie nous amène à approfondir la précédente. En effet, il ne s'agit plus de bloquer les résolutions, mais d'y apporter une réponse maligne. Nous nous plaçons donc dans le contexte où la validité des enregistrements ne peut être vérifiée par DNSsec, et nous développons plus avant sur les fichiers ipsec.secrets et ipsec.conf.

2.3.1. Usurpation portant sur les noms associés aux clefs dans ipsec.secrets

Nous avons présenté dans la partie précédente le fonctionnement de ce fichier. Par conséquent, on devine que l'usurpation peut être de deux types:
Si l'usurpation porte sur un des noms de l'hôte -soit N ce nom-, et que la réponse IP(N) n'est pas une des adresses IP de cet hôte, le nom sera interprété comme étant celui d'un peer accessible à cet adresse IP(N). Ensuite,

- Si aucun peer n'était initialement associé à cette clef, cela aboutit à une « sur-spécialisation » de la clé, c'est à dire qu'elle ne pourra plus être utilisée que pour communiquer avec IP(N). Pour tout autre système, s'il y a une autre clef disponible, le tunnel sera tout de même construit; sinon, on aura donc un déni de service, avec l'erreur suivante:

```
003 "rwsg" #1: unable to locate my private key for RSA Signature
```

Si une autre clef est disponible, mais inappropriée (déni de service), on obtient la même erreur.
Si IP(N) est un peer légitime, rien ne garantit qu'il pourra trouver la clef publique correspondante.
Si IP(N) n'est pas un peer légitime, alors cette clef devient inutile, puisque aucune description de connexion n'y est associée dans ipsec.conf
Cette usurpation a donc plusieurs buts possibles: provoquer un déni de service ou encore empêcher la sélection de cette clef (ce qui aboutira dans la majorité des cas aussi à un déni de service).
- Si d'autres peers étaient initialement associés à cette clef, leurs connexions ne seront pas compromises. L'attaque n'a donc d'effet que si IP(N) aurait dû être associé à une autre clef (même erreur que plus haut, code '003').

Si l'usurpation porte sur un des noms des peers, et que la réponse est une des adresses IP de l'hôte, le nom sera interprété comme un des noms du système local. La clef perdra donc sa spécificité vis à vis des peers, et deviendra plus précise côté hôte. Suivant l'ordre de classement des clefs, les peers habituellement

concernés par cette clef pourront ou non établir la connexion. De même, certains peers non habituellement concernés par la clef pourront se trouver concernés. Dans la majorité des cas, cela provoquera aussi un déni de service avec erreur '003' (voir plus haut).
Enfin, si l'usurpation porte sur un des noms des peers, et que la réponse est l'adresse IP d'un *autre* peer, cela peut affecter le choix de la clef pour ce dernier peer, suivant son ordre d'apparition (cf. partie 'dénis de service' pour un exemple de manipulation basé sur la séquentialité).

2.3.2. Usurpation portant sur les noms spécifiés dans ipsec.conf

Section « config »

Si le paramètre myid (cf. section sur le déni de service) est renseigné avec un nom DNS et qu'un attaquant usurpe ce nom, l'adresse IP retournée par la résolution ne correspond pas à une adresse locale, et les connexions utilisant cette identité (ce qui inclut par défaut les connexions opportunistes) échouent:

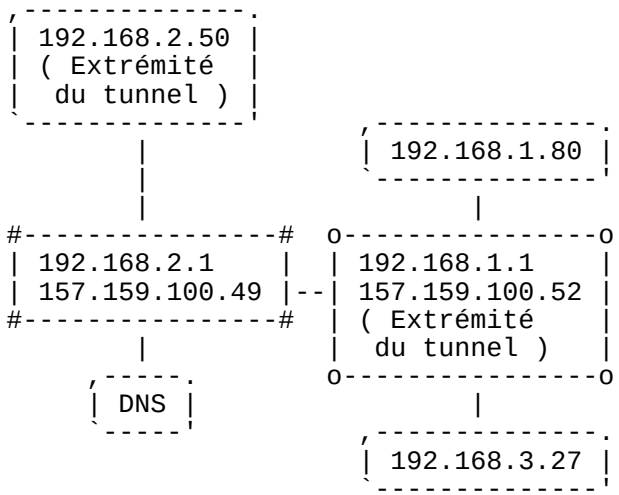
```
ipsec__plutorun: 027 bad --keyid "%myid": illegal (non-DNS-name) character in name
```

Ce message aurait tendance à faire croire qu'il s'agit d'une erreur de configuration, ce qui n'est pas le cas; simplement, l'attaque a empêché la bonne définition de myid, ce qui fait que %myid n'a pas de sens quand cette valeur est invoquée dans les descriptions des connexions.
Par ailleurs, l'attaquant peut aussi agir contre les tentatives effectuées par le système pour attribuer à cette variable une valeur automatique. Pour ce faire, il lui suffit de renseigner le système avec un champ IPSECKEY dans le DNS. Cependant, si la clef publique mentionnée dans ce champ ne correspond pas à une de celles disponibles sur le système, les connexions échoueront comme dans le cas précédent. Enfin, si l'attaquant tente de forcer ainsi la sélection d'une clef privée qui n'est pas celle qui devrait être utilisée pour la connexion, cette dernière échoue encore.

Section « conn »

Nous invitons le lecteur à consulter la partie sur les dénis de services ou la page de manuel de ipsec.conf pour obtenir les descriptions des paramètres de connexions, que nous ne représenterons pas ici.
Comme nous l'avons vu précédemment, le paramètre leftsubnet permet de préciser pour quel sous-réseau le système agit en tant que passerelle. Nous avons aussi vu que ce paramètre peut être renseigné sous la forme nom/masque, par exemple toto-le-sous-reseau/24. Cette fonctionnalité pourrait être très intéressante pour des sous-réseaux de taille fixe dont le préfixe évolue. Mais simultanément, elle offre une possibilité intéressante à un attaquant:

Figure 7.



Dans cette situation, supposons que 192.168.1.0/24 soit le réseau à protéger et que 192.168.3.0/24 soit un réseau démilitarisé. Les connexions entre 192.168.2.50 (le télétravailleur) et 157.159.100.52 (la passerelle) sont décrites de façon identique sur ces deux systèmes (c'est très souvent le cas, parce que l'interface de configuration facilite cela), et il y est mentionné que le sous-réseau privé est sous-reseau-prive/24, avec sous-reseau-prive pointant par exemple vers 192.168.1.1 dans le DNS.
Dans cette situation, si l'attaquant arrive à affecter la résolution de sous-reseau-prive des deux côtés de la connexion et à la faire pointer sur 192.168.3.1 (par exemple), un tunnel sera établi pour les communications entre le télétravailleur et la zone démilitarisée, tandis que les communications entre le télétravailleur et le réseau privé ne sont plus protégées. Du point de vue des implémentations, il n'y a cependant pas d'erreur, ce qui fait que l'attaque risque de passer inaperçue.
Il se peut que d'autres implémentations d'IPsec soient vulnérables à cette attaque. Cela mérite une investigation plus poussée. Par ailleurs, il convient aussi de relativiser les risques: quasiment personne ne décrirait un sous-réseau par un nom.
L'attaquant peut affecter le choix de la route par défaut en jouant sur la résolution de leftnexthop. Une autre route sera alors sélectionnée, pour peu qu'elle soit directement accessible sur une des interfaces de la passerelle. Cela n'affectera pas la confidentialité ou l'intégrité des données. Tout au plus, l'intérêt d'une telle manoeuvre est d'intercepter les paquets pour effectuer une analyse de trafic ou encore de provoquer un déni de service en envoyant les paquets vers un « trou noir ».
Dans la partie traitant des dénis de service, nous avions présenté le paramètre leftid, et nous avions renvoyé le lecteur vers la partie traitant du paramètre myid, dont le comportement est similaire. Nous avions aussi présenté le paramètre left. Il n'était pas alors utile de présenter la différence entre les versions « gauche » et les versions « droite » de ces paramètres. Ici encore, les comportements sont similaires pour l'hôte, et une usurpation sur les résolutions n'apporte rien de nouveau (risques de déni de service).
Il convient cependant de parler ici plus précisément de la localisation et de l'identité du peer, renseignées respectivement par right et rightid. Dans le cas général, si l'attaquant affecte les résolutions de ces paramètres, lorsqu'il s'agit de noms, il provoquera un déni de service ou le tunnel s'établiera tout de même (nous avons pu constater que la vérification de l'identité -rightid- est parfois plus « souple » qu'on ne pourrait s'y attendre, ce qui n'est pas nécessairement un mal). Une nouvelle opportunité se présente cependant si ces paramètres sont initialisés en fonction de l'existence d'un enregistrement IPSECKEY dans le DNS. Dans un tel cas, si l'attaquant arrive à usurper la réponse du DNS et à fournir un enregistrement à son avantage, il peut détourner le tunnel et se faire passer pour le peer. En effet, il aura pris soin de fournir une clef publique appropriée dans l'enregistrement. On a donc une faille de sécurité majeure.
Cette faille n'est pas nouvelle (du moins pour les personnes qui travaillent sur IPSECKEY), et concerne aussi le paramètre rightrsasigkey, qui renseigne sur la clef publique du peer. Ce paramètre peut être initialisé au chargement d'IPsec ou à la demande à partir d'un enregistrement IPSECKEY obtenu depuis le DNS. Par conséquent, la même vulnérabilité existe ici.
Ce qui a été présenté précédemment semble plaider pour l'interdiction de l'usage d'IPSECKEY. Il convient de clarifier les choses. Tout d'abord, il est tout à fait possible de construire un « résolveur » DNS qui refuserait un enregistrement IPSECKEY s'il n'a pas été correctement authentifié (par exemple via DNSsec). Ensuite, les failles mentionnées sont issues d'une mauvaise utilisation de la fonctionnalité de chiffrement opportuniste. L'objet de cette méthode n'est pas de fournir un cadre de description différent pour sécuriser des connexions usuelles, mais plutôt d'offrir une alternative au texte clair. Pour éviter toute erreur d'utilisation de ces fonctionnalités, il convient de considérer que, d'un point de vue politique, une connexion opportuniste est l'équivalente d'une connexion en clair.

2.4. Les Failles Logicielles

Il est difficile de déterminer dans quelles latitudes une faille logicielle d'une implémentation DNS pourrait compromettre IPsec. Faute de mieux, nous pouvons décliner les hypothèses en quelques catégories:

- Si l'implémentation DNS compromise est exécutée par le système mettant en oeuvre IPsec, cette dernière peut être aussi très largement compromise (les protections actuelles contre les situations de ce type, par exemple le chroot, ne sont pas totalement satisfaisantes).
- Si l'implémentation DNS compromise est liée à l'exécution d'IPsec sur un autre système via des échanges DNS, l'issue la plus probable est le déni de service d'IPsec. Cependant, on peut aussi imaginer que l'attaquant possède (de façon légitime ou pas) les clefs nécessaires pour communiquer avec l'hôte utilisant IPsec. Le DNS peut alors être utilisé comme présenté plus haut pour forcer l'écrasement d'une identité par une autre. Enfin, si IPSECKEY est utilisé dans ce DNS, alors l'attaquant dispose désormais du pouvoir de re-router les communications (en imposant un de ses systèmes comme une passerelle de sécurité), et de s'interposer en homme du milieu (en fournissant les clefs publiques appropriées).
- Dans certains cas, le système DNS compromis permet, par des réponses appropriées aux résolutions de noms, de contourner les passerelles (par exemple, en prétendant que l'adresse associée à un nom externe est une adresse interne; en l'occurence, le DNS corrompu est aussi un bon système pour une usurpation en interne).

3. IPsec affecte DNS

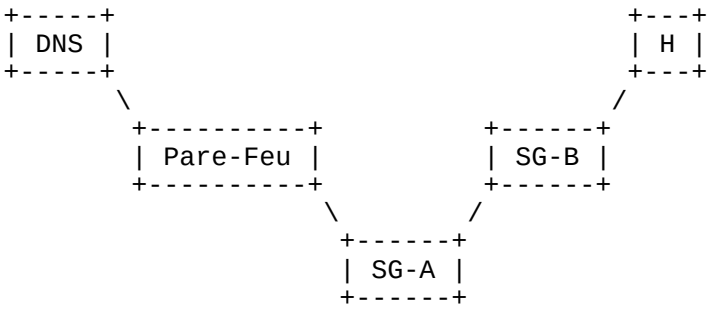
IPsec est la réunion de plusieurs composantes orientées « réseau », qui pourraient potentiellement interagir avec le DNS. Reprenons-en la liste:

- Des protocoles de protection des données (AH et ESP)
- Un filtre à paquets (la SPD)
- Des protocoles d'échanges de clefs (IKE ou IKEv2)

3.1. Interactions via AH ou ESP

Du point de vue du protocole DNS, AH et ESP appartiennent à des couches inférieures et sont donc transparents.
Cependant, entre les messages DNS courts, s'accommodant parfaitement d'UDP, et les messages DNS longs, nécessitant TCP, il existe un risque pour que des messages longs soient acheminables via UDP. Considérons alors le scénario suivant:

Figure 8.



Sur la figure précédente, supposons que H désire consulter DNS et obtenir des informations étendues. DNS émet donc un paquet UDP d'une taille conséquente. Ce paquet, une fois arrivé à SG-A devrait être encapsulé dans le tunnel menant à SG-B. Malheureusement, le paquet UDP est trop grand pour la MTU du tunnel et par conséquent sa fragmentation est requise. Quasi systématiquement, cette fragmentation ne sera pas effectuée parce que le bit Don't Fragment est activé par défaut dans les paquets IP émis par une majorité de systèmes d'exploitation ou de routeurs. Un message ICMP de type Destination Unreachable / Fragmentation Needed and Don't Fragment was set est donc renvoyé à DNS. L'administrateur de Pare-Feu, particulièrement zélé, a bloqué aveuglément les messages ICMP, et l'erreur ne parvient pas à DNS. H ne peut donc jamais obtenir sa réponse.
Ce scénario requiert un certain concours de circonstances, cependant il est courant de rencontrer dans la réalité des applications recherchant quantité d'informations dans le DNS et des pare-feux bloquant le « path-MTU ». Certaines implémentations d'IPsec (dont FreeS/WAN) proposent même de passer outre le bit Don't Fragment, et effectuent simultanément un envoi d'erreur ICMP à la source du paquet et une fragmentation du paquet protégé avant son émission sur le lien.

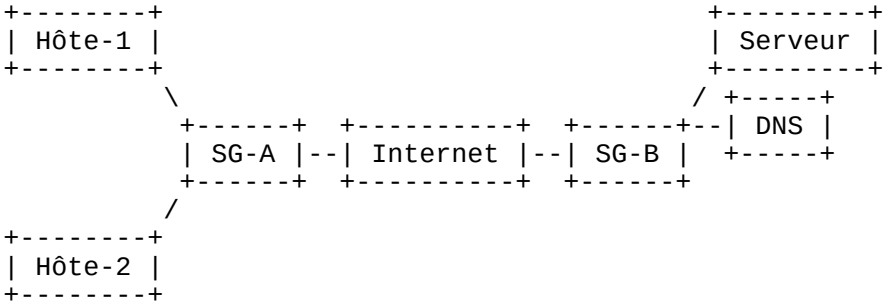
3.2. Interactions d'ordre politique

Ces interactions aboutissent généralement à des dénis de service des résolutions de nom. Pour cette raison, elles sont relativement simples à détecter, et constituent souvent plutôt les effets secondaires d'une configuration qui en est à ses balbutiements.

Un cas relativement difficile à détecter rejoint le scénario présenté dans la partie précédente: la SPD peut bloquer des messages ICMP d'erreur requérant une fragmentation de la part de l'émetteur d'une requête DNS; la SPD joue alors le rôle que tenait Pare-Feu dans la figure précédente.
Nous appelons les autres cas « dumb locks ». Il s'agit d'erreurs de configuration courantes:

1. Dans le cas où on refuse d'acheminer tout trafic non explicitement autorisé, penser à définir une régle pour autoriser la transition des requêtes DNS en clair entre certains systèmes. Par exemple, entre le serveur DNS de la DMZ et l'extérieur, ou encore entre l'acheminer DNS du réseau interne et l'extérieur. Cet oubli est trivial à détecter puisque les effets s'en ressentent vite au niveau applicatif.
2. Le cas suivant est un exemple classique de connexion SOHO; DNS est le serveur DNS global de l'entreprise:

Figure 9.



Deux risques existent alors: si le tunnel est rompu, les résolutions sont compromises, et Hôte-1 et Hôte-2 ne peuvent plus se joindre par leurs noms. Par ailleurs, nous sommes ramenés au cas précédent si le tunnel est établi à la demande, pour des raisons économiques (soit pour minimiser le temps de connexion, soit pour réduire le trafic total échangé). En effet, les paquets émis par Hôte-1 et à destination de Serveur arrivent alors sur SG-A et déclenchent la construction du tunnel. Malheureusement, dans le cas présent, le tunnel est nécessaire pour accéder à DNS, ce qui fait que Hôte-1 peut être dans l'impossibilité de déterminer l'adresse associée au nom de Serveur si le tunnel n'a pas encore été établi. Cela implique, une nouvelle fois, que la SPD doit établir automatiquement le tunnel pour les requêtes DNS. D'un point de vue « déploiement DNS », il peut être utile de considérer que les deux sous-réseaux appartiennent à des sous-domaines différents (il n'est pas nécessaire que la description de ces sous-domaines soit publique).

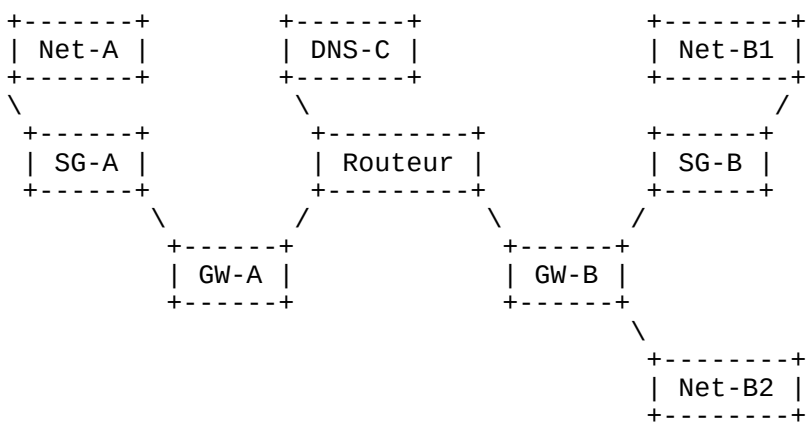
3. Un dernier cas un peu plus complexe se présente avec l'utilisation d'IPSECKEY. Il est en effet légitime de vouloir utiliser DNS pour publier / récupérer des clefs publiques et simultanément de vouloir établir un tunnel avec le serveur DNS. L'erreur classique est alors de requérir l'obtention de la clef publique du serveur DNS via un champ IPSECKEY pour serveur ad-hoc. Cette obtention est impossible car le tunnel n'existe pas encore et ne peut donc acheminer la requête, et respectivement, le tunnel ne pourra être construit tant que la requête restera sans réponse. Par ailleurs, le « résolveur » et IPsec opèrent chacun de leur côté, et il n'est pas évident qu'un message précisant la nature exacte de l'erreur soit remonté à l'utilisateur (le plus souvent, on en conclura que le serveur DNS n'est pas joignable). L'approche appropriée consiste donc à obtenir la clef publique du serveur DNS autrement que via une entrée dans la zone DNS.

3.3. Interactions depuis IKE

IKE est un protocole de niveau applicatif basé sur UDP. Il ne devrait pas y avoir, a-priori, d'interactions entre IKE et DNS. Cependant, si IKE échoue à l'établissement d'une association de sécurité destinée à acheminer les requêtes DNS, nous sommes ramenés au cas précédent.

Par ailleurs, étant basé sur UDP, le flux du trafic de IKE n'est pas contrôlé. Cela n'affectera pas les applications basées sur TCP (qui réduiront leur débit), mais les applications basées sur UDP, donc par exemple DNS, peuvent en subir les conséquences.
Le scénario suivant est ainsi tout à fait plausible:

Figure 10.



Sur la figure précédente, SG-A et SG-B interconnectent les réseaux Net-A et Net-B1, en utilisant par exemple un tunnel différent par couple d'hôtes; si SG-B est contrainte à redémarrer brutalement (coupure de courant, « watchdog », etc.), de nombreuses requêtes IKE tenteront de reconstruire les associations de sécurité perdues. Une congestion peut alors advenir entre GW-B et Routeur. Les messages DNS entre Net-B2 et DNS-C peuvent donc être perdus dans les files saturées de Routeur.
Il convient cependant de relativiser cette observation; en effet, plusieurs implémentations de IKE sont conçues pour une reprise aisée et limitent donc le nombre de requêtes émises par intervalle de temps, ce qui réduit les risques de congestion ainsi que la charge de calculs crytographiques à chaque instant.

4. IPsec et DNS travaillant de concert

4.1. IPsec utilisant DNS

DNS est l'un des deux systèmes de distribution de l'information sur IP ayant réussi à atteindre une dimension mondiale (l'autre est le routage). Il est particulièrement tentant de s'appuyer dessus pour diffuser de l'information sur les services disponibles; après tout, la plupart des DNS mettent à disposition des informations sur les services DNS eux-mêmes, mais aussi sur SMTP (enregistrement MX). De nombreux groupes de travail autour d'un protocole donné ont un jour où l'autre considéré cette question, et IPsec n'y fait pas exception. Dans tous les cas, évidemment, les parties engagées estiment que le déploiement de LEUR protocole est le plus à même de justifier l'alourdissement des zones DNS. Dans les faits, même pour IPsec, rien n'est moins évident. Rien n'empêche cependant de tenter l'expérience à faible échelle.
Il convient alors de déterminer quelles informations doivent être insérées dans le DNS afin de faciliter le déploiement d'IPsec. Classiquement, pour une destination à contacter, les deux informations retenues sont:

- La localisation (adresse ou nom) de la passerelle associée à la destination.
- La clef publique (et l'algorithme correspondant) utilisée par la passerelle.

Une autre information qui pourrait être utile est l'identité de la passerelle. Cette identité pourrait s'avérer essentielle pour permettre à la passerelle de déterminer quel processus a amené l'initiateur à la contacter (scénario de type « Me Tarzan, You Jane »). Cependant, cette utilisation de l'identité semble trop récente pour pouvoir être incorporée dans la version actuelle d'IPSECKEY, même s'il est possible d'établir une convention sans modifier la description actuelle de l'enregistrement.
La publication des données dans le DNS peut s'opérer de plusieurs manières. Tout d'abord, plusieurs enregistrements standards pourraient convenir (en particuliers KX et CERT). Actuellement, les implémentations de méthodes de chiffrement opportuniste utilisent les champs KEY et TXT (mais cette utilisation du champ KEY est *très déconseillée* depuis peu). A terme, le champ IPSECKEY, ayant été construit spécifiquement pour IPsec, devrait s'imposer (mais il est difficile de considérer que le terme « s'imposer » est approprié pour désigner quelque chose dont le déploiement sera sans doute limité).
Il convient ensuite de répondre à la question: « Où placer l'enregistrement ? ». En effet, et bien que l'utilisation quotidienne du DNS ait tendance à faire oublier ce fait, le DNS supporte plusieurs types de zones, les plus importantes dans notre contexte étant la zone_forward et la zone_reverse.

4.1.1. Utilisation de la zone_forward

Cette zone permettra d'associer les informations pour IPsec à un nom DNS totalement qualifié; pour un hôte, cette zone fournit un avantage des plus intéressants, principalement pour deux raisons:

- Si l'hôte contacte un système qui lui est initialement inconnu, il y a de fortes chances pour que cela soit dû à la saisie d'un nom depuis une application utilisateur. Par conséquent, la zone_forward sera de toute façon interrogée pour obtenir l'adresse IP du peer. Découvrir par la même occasion l'adresse et la clef de la passerelle associée à ce peer constitue une opportunité intéressante (c'est effectivement un scénario typique de *chiffrement opportuniste*).
- Si l'hôte est un télétravailleur, il a besoin d'une méthode simple pour accréditer ses données (typiquement, sa propre clef publique) et les protéger, sans les lier à une adresse IP statique. La publication de ses informations de cryptographie dans la zone_forward, associée à la revendication d'une identité dans la phase 1 de IKE (« Je suis tarzan.jungle.tz ») répond à ces besoins. L'autorité de la zone DNS interrogée se porte ainsi garante de ces informations sur le télétravailleur.

4.1.2. Utilisation de la zone_reverse

La zone_forward est *inadaptée* pour le travail des passerelles de sécurité; en effet, lorsqu'elles décident de l'acheminement d'un paquet, les passerelles consultent les adresses source et destination. Elles n'ont aucun intérêt à effectuer une résolution de nom inverse pour ensuite interroger à nouveau la zone directe, laquelle peut leur apporter plusieurs réponses inconsistantes. La zone_reverse est donc plus particulièrement appropriée pour les passerelles.
Malheureusement, cette zone est très mal maintenue.

Évidemment, insérer des informations pour IPsec dans le DNS implique que l'on soit en mesure de les protéger. Cette protection passe nécessairement par DNSsec, dont la déployabilité à court terme est peu probable.
En revanche, avec un ensemble de serveurs DNS réunis sous une même autorité, il est possible de construire ainsi un système bien plus léger qu'une PKI et tout aussi sûr pour les éléments de ce réseau. Pour les éléments extérieurs, il reste les politiques de chiffrement opportuniste, ce qui constitue une victoire à la Pyrrhus.

4.2. DNS utilisant IPsec

DNS est doté de son propre système de protection quant à l'intégrité des données (DNSsec).
À court terme, IPsec peut pallier à la faiblesse du déploiement de DNSsec. Cependant, son utilisation protège les liens, et non la propagation de l'information DNS; s'il est, par exemple, possible de contrôler efficacement l'accès au service de transfert de zones avec IPsec, il est en revanche impossible de garantir la validité à l'origine des enregistrements transférés. Tous les systèmes relayant l'information, tel SMTP, sont soumis à ce type de problématique.
Cependant, à faible échelle et pour un usage strictement interne, il est possible de sécuriser DNS avec IPsec, puisque l'origine des données ne peut être qu'interne et renseignée par un administrateur. L'étendue de la protection est la suivante:

- Les opérations de cartographie du réseau via transfert de zones sont contrées par un contrôle d'accès réel et l'utilisation du chiffrement.
- L'utilisation du DNS pour déclencher des dénis de service est limité pour les mêmes raisons (contrôle d'accès).
- L'usurpation ainsi que l'empoisonnement de cache deviennent impossibles avec une utilisation de DNS strictement interne correctement administrée.
- Les utilisateurs peuvent enfin émettre des requêtes DNS chiffrées.

Malheureusement, l'intérêt d'utiliser le DNS strictement en interne est limité, et il n'existe pas de paradigme simple permettant de spécifier des politiques de confiance sur les serveurs DNS (cela permettrait de donner la priorité aux réponses du DNS interne pour un ensemble de zones, et de n'interroger le DNS public que pour les autres zones).

5. Conclusion

Il est légitime de vouloir simplifier le nommage, de vouloir remplacer des adresses par des noms. Simultanément, il convient de ne pas oublier que le nom DNS est avant tout un système d'indexation pour obtenir une localisation. Le trépied serait incomplet s'il n'existait la possibilité d'utiliser des identités.
Adresse IP, Nom DNS, Identité: Maîtriser le sens caché derrière chacun de ces concepts n'est pas évident, même pour les groupes de travail qui ont pour responsabilité de les spécifier ! Pour éviter tout problème de sécurité, il convient, comme nous l'avons vu au cours de cet article, de connaître le niveau de sécurité des systèmes qui permettent de passer de l'un à un autre. Il n'est pas impossible de tirer parti de la flexibilité des interfaces de configuration, qui offrent la possibilité de décrire des connexions IPsec par la méthode qui satisfait au désir immédiat de l'utilisateur, mais cela doit se faire en pleine connaissance de cause.

Références

[Be03] *Failles Intrinsèques du Protocole DNS*. Pierre Betouin. 20 Octobre 2003. http://securitech.homeunix.org/dnsa/ArticleDNS.pdf

[PBM03] « Étude du Chiffrement Opportuniste dans la Mobilité IP ». Jean-Jacques Puig, Julien Bournelle, et Maryline Laurent-Maknavicius. *DNAC*. Novembre 2003.

[DP02] « Exploitation Malicieuse du Protocole DNS ». Eric Detoisien et Daniel Polombo. *MISC*. 4. November 2002.

[Gr02] « Sécurité et Failles des Serveurs DNS ». Christophe Grenier. *Pirates'Mag*. 11.

[PM02] *Analyse de l'Impact de la mise en oeuvre d'IPsec dans les Architectures de Communications*. Jean-Jacques Puig et Maryline Maknavicius. Décembre 2002. RR-03002-LOR

[NSTAC02] *Enhancing the Security of Name Resolution and Inter-Domain Internet Routing*. Mai 2002. NSTAC

[FreeS/WAN] *Linux FreeS/WAN*. http://www.freeswan.org/

[poisonivy] *Poison Ivy DNS cache poisoner*. http://valgasu.rstack.org/

[dnsspoof] *dsniff*. http://packetstormsecurity.nl/sniffers/dsniff/

[admsniffid] « ADMSniffID ». *Phrack*. 52. Janvier 1998. http://www.phrack.org/

[rfc1034] *DOMAIN NAMES - CONCEPTS AND FACILITIES*. P. Mockapetris. Novembre 1987. RFC 1034. IETF.

[rfc1035] *DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION*. P. Mockapetris. Novembre 1987. RFC 1035. IETF.

[rfc1912] *Common DNS Operational and Configuration Errors*. D. Barr. Février 1996. RFC 1912. IETF.

[rfc2535] *Domain Name System Security Extensions*. D. Eastlake. Mars 1999. RFC 2535. IETF.

---

[1] Et ce, malgré les options effectivement proposées par dig, pour des raisons à déterminer par la lecture du code source.

[2] L'avertissement suivant est tout à fait normal:
```
WARNING: probably, rcvbuf is not enough to hold preload.
```

[3] Nous avons auparavant tenté cette manipulation via un lien série avec un câble croisé, puis avec le module simplifié de mise en forme du trafic. Dans les deux cas nous avons rencontré des comportements aberrants du système.

[4] En réalité, tant que l'enregistrement IPSECKEY n'est pas à l'état de standard, on utilise un enregistrement TXT pour le même objectif. Par soucis de clarté, nous considérons que nous utilisons IPSECKEY, même si la réalité diffère un peu.