

Table des matières

I.Fonctionnement.....	2
1.Mode de transport.....	2
2.Les composants d'IPSec.....	3
3.L'échange des clés.....	4
4.Informations complémentaires : accès à distances par les utilisateurs.....	5
a)Authentification IPSec de la phase 1.....	5
b)Xauth.....	5
c)Hybrid auth.....	5
d)Mode de configuration ISAKMP.....	5
II.Implémentation Linux.....	6
1.IPSec-tools (KAME-tools).....	6
a)Installation.....	6
b)Configuration de base.....	6
i.Association de sécurité pour l'authentification.....	7
ii.Association de sécurité pour l'encryptage.....	7
iii.Politique de sécurité.....	8
(1)Paquets entrants.....	8
(2)Paquets sortants.....	8
iv.Un exemple complètement statique en mode Tunnel (VPN).....	8
v.Un exemple complètement statique en mode Transport.....	10
vi.D'autres exemples de configuration IPSec.....	11
c)Configuration avec clés automatique par IKE et certificats pour utilisateur itinérant (serveur et client Linux).....	12
i.Installation de ipsec-tools.....	12
(1)Compilation sur le client.....	12
(2)Compilation sur le serveur.....	12
ii.Générer son autorité de certification.....	12
iii.Génération des certificats du serveur et des clients.....	13
iv.Mise en place des certificats et autres clés.....	13
v.Configuration de racoon.....	14
(1)Serveur.....	14
(2)Client.....	15
(3)Mémorisation du mot de passe sur le client.....	16
(4)Notes.....	17
vi.Démarrage.....	17
d)Connexion et déconnexion au VPN.....	17
2.Problème de NAT, de fragmentation et de délai de connexion.....	17
a)NAT.....	17
b)Fragmentation.....	18
c)Délai de connexion.....	18
3.Et iptables dans tout ça.....	18
a)La passerelle IPSec machine 1.....	18
b)La passerelle IPSec machine 2.....	19
c)En plus sur les passerelles IPSec.....	19

d)En plus sur le client.....	20
III.IPsec seul sous Windows.....	20
1.Configuration par clés prépartagées.....	20
a)Le fichier de configuration de racoon.....	20
b)Le fichier de clé prépartagées : psk.txt.....	21
c)Préparation de MMC.....	21
d)La politique de sécurité.....	21
2.Configuration par certificats.....	23
a)Le fichier de configuration de racoon.....	23
b)Génération des certificats.....	24
c)L'exportation des certificats.....	24
d)Préparation de MMC.....	25
e)L'importation des certificats.....	25
f)La politique de sécurité.....	25
3.Débogage.....	26
IV.Bibliographie.....	26

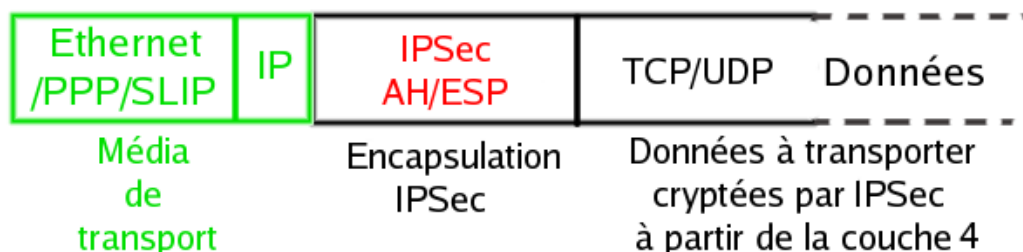
I. Fonctionnement

1. Mode de transport

IPsec est un protocole défini par l'IETF permettant de sécuriser les échanges au niveau de la couche réseau. Il s'agit en fait d'un protocole apportant des améliorations au niveau de la sécurité au protocole IP afin de garantir la confidentialité, l'intégrité et l'authentification des échanges.

Il existe deux modes pour IPsec :

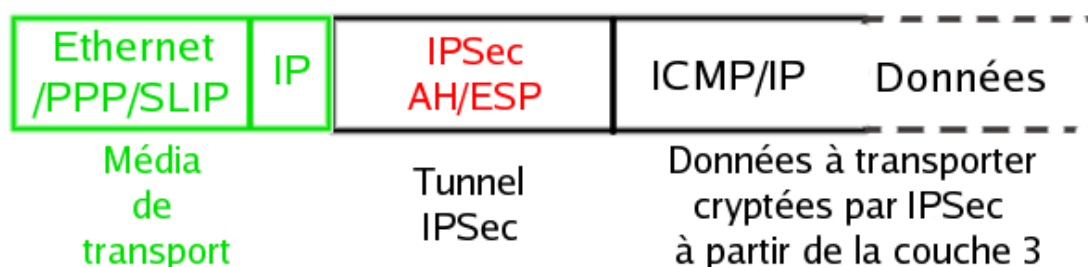
- le mode transport permet de protéger principalement les protocoles de niveaux supérieurs :
 - IPsec récupère les données venant de la couche 4 (TCP/transport), les signe et les crypte puis les envoie à la couche 3 (IP/réseau). Cela permet d'être transparent entre la couche TCP et la couche IP et du coup d'être relativement facile à mettre en place.
 - Il y a cependant plusieurs inconvénients :
 - l'entête IP est produite par la couche IP et donc IPsec ne peut pas la contrôler dans ce cas.
 - Il ne peut donc pas masquer les adresses pour faire croire à un réseau LAN virtuel entre les deux LAN reliés
 - cela ne garantit donc pas non plus de ne pas utiliser des options Ips non voulues



- le mode tunnel permet d'encapsuler des datagrammes IP dans des datagrammes IP
 - les paquets descendent dans la pile jusqu'à la couche IP et c'est la couche IP qui passe ses données à la couche IPsec. Il y a donc une entête IP encapsulée dans les données IPsec et une entête IP réelle pour le transport sur Internet (on pourrait imaginer que ce transport se

fasse sur de l'IPX ou NetBIOS puisqu'il n'y a pas de contrainte dans ce mode)

- Cela a beaucoup d'avantages :
 - l'entête IP réelle est produite par la couche IPSec. Cela permet d'encapsuler une entête IP avec des adresses relative au réseau virtuel et en plus de les crypter de façon à être sûr qu'elles ne sont pas modifiées.
 - On a donc des adresses IP virtuelles donc tirant partie au mieux du concept de VPN
 - On a le contrôle total sur l'entête IP produite par IPSec pour encapsuler ses données et son entête IPSec.



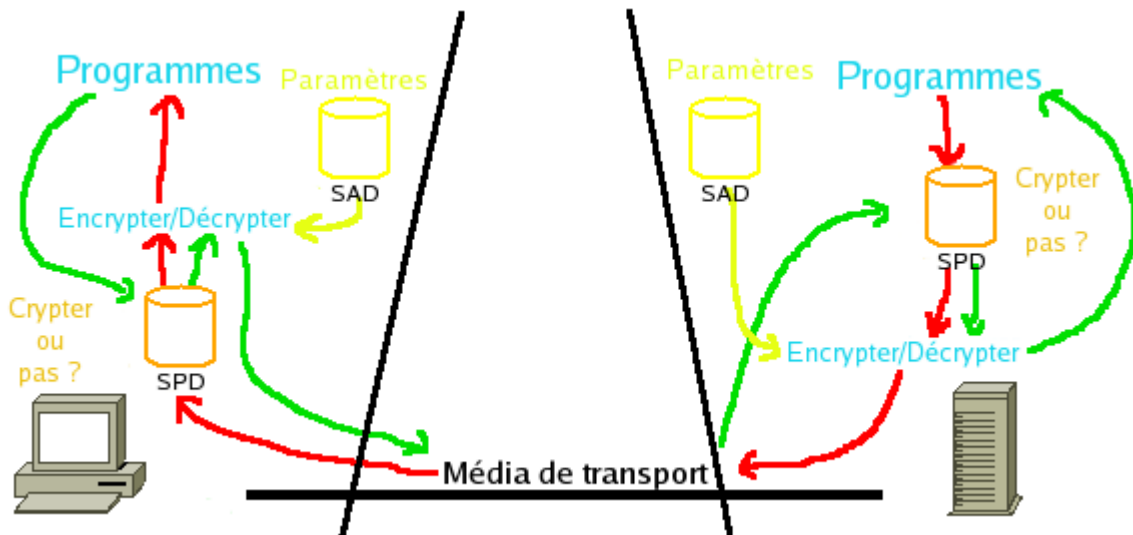
2. Les composantes d'IPSec

Le protocole IPSec est basé sur quatre modules :

- *IP Authentication Header (AH)* gère
 - l'intégrité : on s'assure que les champs invariants pendant la transmission, dans l'entête IP qui précède l'entête AH et les données
 - l'authentification pour s'assurer que l'émetteur est bien celui qu'il dit être
 - la protection contre le rejeu : un paquet intercepté par un pirate ne peut pas être renvoyé
 - il ne gère pas la confidentialité : les données sont signées mais pas cryptées
- *Encapsulating Security Payload (ESP)*
 - en mode transport, il assure
 - confidentialité : les données du datagramme IP encapsulé sont cryptées
 - authentification : on s'assure que les paquets viennent bien de l'hôte avec lequel on communique (qui doit connaître la clé associée à la communication ESP pour s'authentifier)
 - l'unicité optionnelle contre le rejeu des paquets
 - l'intégrité des données transmises
 - en mode tunnel, c'est l'ensemble du datagramme IP encapsulé dans ESP qui est crypté et subit les vérifications suivantes. On peut donc se passer de AH.
- *Security Association (SA)* définit l'échange des clés et des paramètres de sécurité. Il existe une SA par sens de communication. Les paramètres de sécurité sont les suivants :
 - protocole AH et/ou ESP
 - mode tunnel ou transport
 - les algo de sécurité utiliser pour encrypter, vérifier l'intégrité
 - les clés utilisées
- La *SAD (Security Association Database)* stocke les SA afin de savoir comment traiter les paquets arrivant ou partant. Elles sont identifiées par des triplets :
 - adresse de destination des paquets
 - identifiant du protocole AH ou ESP utilisé

- un index des paramètres de sécurité (Security Parameter Index) qui est un champ de 32bits envoyer en clair dans les paquets
- La SPD (Security Policy Database) est la base de configuration de IPSec. Elle permet de dire au noyau quels paquets il doit traiter. C'est à sa charge de savoir avec quel SA il fait le traitement.

En résumé, le SPD indique quels paquets il faut traiter et le SAD indique comment il faut traiter un paquet sélectionné.

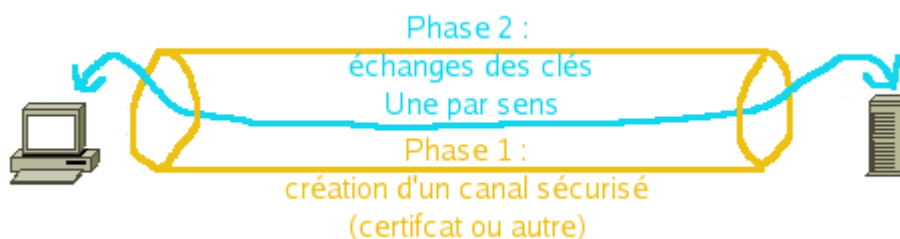


3. L'échange des clés

L'échanges des clés nécessaires au cryptage des données dans IPSec peut se faire de trois façons différentes :

- à la main : pas très pratique
- IKE (Internet Key Exchange) : c'est un protocole développé pour IPSec. ISAKMP (Internet Security Association and Key Management Protocol) en est la base et a pour rôle la création (négociation et mise en place), la modification et la suppression des SA. Elle se compose de deux phases :
 - la première permet de créer un canal sécurisé (par Diffie-Hellman) et authentifié à travers duquel on échange un secret pour dériver les clés utilisées dans la phase 2.
 - la seconde permet de mettre en place IPSec avec ses paramètres et une SA par sens de communication. Les données échangées sont protégées par le canal mis en place dans la phase 1.

A l'issue de ces deux phases, le canal IPSec est mis en place.



4. Informations complémentaires : accès à distances par les utilisateurs

a) Authentification IPsec de la phase 1

La phase 1 d'IPsec fait parti du protocole IKE (IPsec Key Exchange) géré par le démon racoon. Son but est d'authentifier les machines en présence (client et serveur) afin de définir une clé maître pour sécuriser la phase 2 d'IPsec. Le but de la phase 2 est, quant à elle, de dériver les clés utilisées pour les échanges de trafic par IPsec.

La phase 1 d'IPsec offre deux méthodes d'authentification :

- les clés prépartagés. Plus difficile à maintenir mais rapide à mettre en place. Moins sûr.
- les certificats. Plus facile à maintenir et moins rapide à mettre en place. Plus sûr.

b) Xauth

Xauth est une extension IKE qui se passe après la phase 1 et ajoute une authentification login/mot de passe sécurisé par la phase 1 (mot de passe pas en clair).

c) Hybrid auth

L'authentification hybride est une autre extension IKE qui rend la phase 1 asymétrique. Durant la phase 1, la passerelle VPN peut utiliser un certificat alors que l'utilisateur distant n'a pas besoin de s'authentifier. Après la phase 1, on est donc dans la situation suivante :

- l'utilisateur distant sait qu'il parle bien à la bonne passerelle VPN
- la communication entre l'utilisateur distant et la passerelle VPN est sécurisée
- la passerelle VPN ne sait pas réellement si elle parle au bon client ou à un hacker

Après la phase 1, un échange Xauth peut intervenir pour authentifier de façon sûre l'utilisateur distant. Ensuite la phase 2 peut commencer.

Le niveau de sécurité de IPsec + Xauth + Hybrid est à peu près équivalent à une authentification par mot de passe en SSH.

d) Mode de configuration ISAKMP

Cela permet de donner automatiquement une configuration à un utilisateur authentifié de façon à ce que la configuration du VPN soit automatique. C'est encore une extension IKE qui autorise le serveur VPN à fournir une configuration réseau à la machine de l'utilisateur distant : adresse IP interne, adresse DNS, nom de domaine...

II. Implémentation Linux

Il existe plusieurs implémentations de IPsec dans Linux suivant la version du noyau :

- 2.4 et inférieur : FreeS/WAN (<http://www.freeswan.org>) et OpenSWAN (<http://www.openswan.org>)
- 2.6 et supérieur : support Natif (KAME-tools at <http://ipsec-tools.sourceforge.net/>), OpenSWAN ou isakmpd (de OpenBSD)

1. IPSec-tools (KAME-tools)

a) Installation

Pour utiliser IPSec en natif dans Linux, il faut avoir compilé son noyau avec les options (à y) (le critère de recherche dans *config-noyau* est indiqué entre parenthèses) :

- Network support
 - PF_KEY sockets (NET_KEY)
 - AH transformation (INET_AH)
 - ESP transformation (INET_ESP)
 - IPSec user configuration interface (XFRM_USER)
- Cryptographic API, en vrac : HMAC, Null, MD5, SHA1, DES et Triple DES, AES (CRYPTO_*)

Pour vérifier qu'une option *critère* est incluse dans le noyau ou dans un module de celui-ci :

```
cat /boot/config-`uname-r` |grep critère
```

Doit renvoyer au moins une ligne telle que : CONFIG_*critère*=y ou CONFIG_*critère*=m

Il faut aussi installer les IPSec-tools (<http://ipsec-tools.sourceforge.net/>) :

```
./configure --with-kernel-headers=/lib/modules/2.6.X/build/include  
make  
make install
```

où *X* est le troisième chiffre de la version de votre noyau.

b) Configuration de base

La configuration est assez compliquée mais peut se résumer de la façon suivante :

- on crée un script pour définir les paramètres IPSec à mettre dans la SAD (Security Association Database) et dans la SPD (Security Policy Database) :
 - SAD
 - les IP source et destination
 - le mode tunnel ou transport
 - les algos de hachage, cryptage et autres
 - des clés pour AH et ESP
 - SPD
 - les IP source et destination
 - le sens du trafic
 - si AH et/ou ESP sont requis
- on exécute le script avec la commande `setkey -f`.

i. Association de sécurité pour l'authentification

Une association de sécurité d'authentification a la forme suivante :

```
add IP_pub_machine1 IP_pub_machine2 ah SPI1 -A algo_hash
"clé_authentification";
```

Ceci indique que le trafic de *IP_pub_machine1* à *IP_pub_machine2* doit pouvoir être authentifié avec une entête AH signée avec l'algo de hachage *algo_hash* et la clé d'authentification *clé_authentification*. Cette association est repérée par l'index *SPI1* (un nombre). Les associations sont symétriques, c'est à dire qu'il faut la mettre identique sur les deux machines communicantes. De plus, elle n'agit que sur un sens de communication donc si l'on veut authentifier ou crypter dans les deux sens il faut une association par sens.

IP_pub_machine1 et *IP_pub_machine2* sont les IP de ces machines. La clé d'authentification est un nombre (décimal ou hexa) dont la taille en bits dépend de l'algo de hachage. *algo_hash* peut être choisi parmi : hmac-md5, hmac-sha1, keyed-md5, keyed-sha1, hmac-sha256, hmac-sha384, hmac-sha512, hmac-ripemd160, aes-xcbc-mac ou tcp-md5. Suivant la valeur de ce dernier, la taille de la clé peut être différente (md5 : 128bits, sha1 : 160bits...)

On peut aussi ajouter `-m tunnel` pour passer l'association en mode tunnel.

ii. Association de sécurité pour l'encryptage

Une association de sécurité d'encryptage a la forme suivante :

```
add IP_pub_machine1 IP_pub_machine2 esp SPI2 -E algo_crypt
"clé_encryptage";
```

Ceci indique que le trafic de *IP_pub_machine1* à *IP_pub_machine2* doit pouvoir être crypté (avec donc une entête ESP) avec l'algo de cryptage *algo_crypt* et la clé de cryptage *clé_encryptage*. Cette association est repérée par l'index *SPI2* (un nombre). Les associations sont symétriques, c'est à dire qu'il faut la mettre identique sur les deux machines communicantes. De plus, elle n'agit que sur un sens de communication donc si l'on veut authentifier ou crypter dans les deux sens il faut une association par sens.

IP_pub_machine1 et *IP_pub_machine2* sont les IP de ces machines. La clé d'encryptage est un nombre (décimal ou hexa) dont la longueur en bits dépend de l'algo d'encryptage. *algo_encryptage* peut être choisi parmi : des-cbc, 3des-cbc, blowfish-cbc, cast128-cbc,... Suivant la valeur de ce dernier, la taille de la clé peut être différente (des : 64bits, 3des : 192bits...)

On peut aussi ajouter `-m tunnel` pour passer l'association en mode tunnel.

iii. Politique de sécurité

(1) Paquets entrants

Une politique de sécurité a la forme suivante pour les paquets entrants (appliquée sur la machine 1) :

```
spdadd IP_pub_machine2 IP_pub_machine1 any -P IN ipsec
    esp/mode/src-dst/require
    ah/mode/src-dst/require;
```

Cela indique que le trafic venant de *IP_pub_machine2* et à destination de notre interface *IP_pub_machine1* doit être correctement authentifié et crypté et les paquets doivent être dans la mode *mode* (transport ou tunnel). Les paquets en clair (donc pas IPSec) venant de *IP_pub_machine2* seront détruits.

A noter que sur la machine 2, on doit inverser *IP_pub_machine1* et *IP_pub_machine2* et mettre OUT à la place de IN.

IP_pub_machine1 et *IP_pub_machine2* sont les IP de ces machines ou de leurs sous réseaux d'appartenance suivant la portée que l'on veut donner au VPN. La partie *src-dst* peut être omise si le mode est transport. Si le mode est tunnel, il faut préciser entre quelles machines *src-dst* (IP) se fait le tunnel.

(2) Paquets sortants

Une politique de sécurité a la forme suivante pour les paquets sortants (appliquée sur la machine 1) :

```
spdadd IP_pub_machine1 IP_pub_machine2 any -P OUT ipsec
      esp/mode/src-dst/require
      ah/mode/src-dst/require;
```

Cela indique que le trafic sortant de notre interface *IP_pub_machine1* et à destination de *IP_pub_machine2* doit être authentifié et crypté et les paquets doivent être émis dans la mode *mode* (transport ou tunnel).

A noter que sur la machine 2, il est préférable de mettre une règles dans laquelle *IP_pub_machine1* et *IP_pub_machine2* sont inversés et où IN est mis à la place de OUT.

IP_pub_machine1 et *IP_pub_machine2* sont les IP de ces machines ou de leurs sous réseaux d'appartenance suivant la portée que l'on veut donner au VPN . La partie *src-dst* peut être omise si le mode est transport. Si le mode est tunnel, il faut préciser entre quelles machines *src-dst* (IP) se fait le tunnel.

iv. Un exemple complètement statique en mode Tunnel (VPN)

Note : vous pouvez remplacer les clés par des clés générées par vous même avec `/dev/urandom` par exemple.

Voici un exemple de fichier de configuration statique pour le mode tunnel entre une machine 1 et une machine 2 qui permettent de relier les réseaux 1 et 2 (les différences sont en **rouge**). On remarquera que l'on utilise que ESP car en mode tunnel, ce protocole assure à la fois l'encryptage et l'authentification. Attention, seuls les paquets entre une machine du réseau 1 et du réseau 2 seront encryptés. Par contre, si on envoie un ping de *IP_pub_machine1* à *IP_pub_machine2*, il sera en clair car aucunes des IP *IP_pub_machine1* et *IP_pub_machine2* ne font parties des IP des réseau 1 et 2.

- Sur la machine 1 :

```
#!/usr/sbin/setkey -f

# Effacer les associations et les polices existantes
flush;
spdflush;

# ESP SAs doing encryption using 192 bit long keys (168 + 24 parity)
# and authentication using 128 bit long keys
add IP_pub_machine1 IP_pub_machine2 esp 0x201 -m tunnel -E 3des-cbc
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831
```

```
-A hmac-md5 0xc0291ff014dccdd03874d9e8e4cdf3e6;

add IP_pub_machine2 IP_pub_machine1 esp 0x301 -m tunnel -E 3des-cbc
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df
-A hmac-md5 0x96358c90783bbfa3d7b196ceabe0536b;

# Security policies
spdadd adresse_reseau1 adresse_reseau2 any -P out ipsec
    esp/tunnel/IP_pub_machine1-IP_pub_machine2/require;

spdadd adresse_reseau2 adresse_reseau1 any -P in ipsec
    esp/tunnel/IP_pub_machine2-IP_pub_machine1/require;

#il peut être nécessaire pour kernel >= 2.6.10 et ipsec-tools < 0.5
# d'autoriser les paquets à traverser IP_pub_machine1 et IP_pub_machine2
spdadd adresse_reseau2 adresse_reseau1 any -P fwd ipsec
    esp/tunnel/IP_pub_machine2-IP_pub_machine1/require;
```

- Sur la machine 2 (même chose mais dans le sens inverse) :

```
#!/usr/sbin/setkey -f

# Effacer les associations et les polices existantes
flush;
spdf flush;

# ESP SAs doing encryption using 192 bit long keys (168 + 24 parity)
# and authentication using 128 bit long keys
add IP_pub_machine1 IP_pub_machine2 esp 0x201 -m tunnel -E 3des-cbc
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831
-A hmac-md5 0xc0291ff014dccdd03874d9e8e4cdf3e6;

add IP_pub_machine2 IP_pub_machine1 esp 0x301 -m tunnel -E 3des-cbc
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df
-A hmac-md5 0x96358c90783bbfa3d7b196ceabe0536b;

# Security policies
spdadd adresse_reseau1 adresse_reseau2 any -P in ipsec
    esp/tunnel/IP_pub_machine1-IP_pub_machine2/require;

spdadd adresse_reseau2 adresse_reseau1 any -P out ipsec
    esp/tunnel/IP_pub_machine2-IP_pub_machine1/require;

#il peut être nécessaire pour kernel >= 2.6.10 et ipsec-tools < 0.5
# d'autoriser les paquets à traverser IP_pub_machine1 et IP_pub_machine2
spdadd adresse_reseau1 adresse_reseau2 any -P fwd ipsec
    esp/tunnel/IP_pub_machine1-IP_pub_machine2/require;
```

On peut vérifier que les paquets partent et reviennent bien encryptés sur une *IP_interne1* avec `tcpdump host IP_pub_machine2` sur la machine 2:

- le ping : on voit bien que l'on envoie pas le paquet echo-request en clair. On remarquera que l'adresse de destination n'est pas celle de la machine *IP_interne1*. En effet, cette adresse se trouve dans le paquet IP crypté dans les données ESP. Il y a bien un tunnel crypté entre *IP_pub_machine2* et *IP_pub_machine2*

```
11:42:57.676757 IP IP_pub_machine2 > IP_pub_machine1:
ESP (spi=0x00000201, seq=0x52), length 116
```

- le pong : on voit un paquet crypté arriver et on voit bien un paquet ICMP echo-reply remonter. On rappelle que le mode tunnel encapsule l'entête IP réelle dans les données de ESP, ESP étant lui-même encapsulé dans un paquet IP entre les deux bout du tunnel ce qui fait que l'on

croit recevoir deux paquets alors ce n'en est qu'un : on reçoit les données ESP cryptées qui sont décryptées et repassées à la couche IP :

```
11:42:57.677604 IP IP_pub_machine1 > IP_pub_machine2:
ESP(spi=0x00000301,seq=0x6f), length 116
```

```
11:42:57.677604 IP IP_interne1 > IP_pub_machine2: ICMP echo reply, id 6417, seq
5, length 64
```

Cela peut se schématiser ainsi :

v. Un exemple complètement statique en mode Transport

Note : vous pouvez remplacer les clés par des clés générées par vous même avec /dev/urandom par exemple.

Voici un exemple de fichier de configuration statique pour le mode transport entre une machine 1 et une machine 2. **Attention, le mode transport assure l'encryptage ESP et l'authentification AH entre les machines 1 et 2 mais absolument pas entre les réseaux 1 et 2.**

- Sur la machine 1 :

```
#!/sbin/setkey -f

# Configuration for machine1

# Flush the SAD and SPD
flush;
spdflush;

# Attention: Use this keys only for testing purposes!
# Generate your own keys!

# AH SAs using 128 bit long keys
add machine1 machine2 ah 0x200 -A hmac-md5
0xc0291ff014dccdd03874d9e8e4cdf3e6;
add machine2 machine1 ah 0x300 -A hmac-md5
0x96358c90783bbfa3d7b196ceabe0536b;

# ESP SAs using 192 bit long keys (168 + 24 parity)
add machine1 machine2 esp 0x201 -E 3des-cbc
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;
add machine2 machine1 esp 0x301 -E 3des-cbc
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df;

# Security policies
spdadd machine1 machine2 any -P out ipsec
    esp/transport//require
    ah/transport//require;

spdadd machine2 machine1 any -P in ipsec
    esp/transport//require
    ah/transport//require;
```

- Sur la machine 2 (même chose mais dans le sens inverse) :

```
#!/usr/sbin/setkey -f

# Configuration for machine2

# Flush the SAD and SPD
```

```

flush;
spdflush;

# Attention: Use this keys only for testing purposes!
# Generate your own keys!

# AH SAs using 128 bit long keys
add machine2 machine1 ah 0x200 -A hmac-md5
0xc0291ff014dccdd03874d9e8e4cdf3e6;
add machine1 machine2 ah 0x300 -A hmac-md5
0x96358c90783bbfa3d7b196ceabe0536b;

# ESP SAs using 192 bit long keys (168 + 24 parity)
add machine2 machine1 esp 0x201 -E 3des-cbc
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;
add machine1 machine2 esp 0x301 -E 3des-cbc
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df;

# Security policies
spdadd machine2 machine1 any -P in ipsec
    esp/transport//require
    ah/transport//require;

spdadd machine1 machine2 any -P out ipsec
    esp/transport//require
    ah/transport//require;

```

On peut vérifier que les paquets partent et reviennent bien encryptés sur *machine1* avec `tcpdump host machine2`:

- le ping : on voit bien que l'on envoie pas le paquet echo-request en clair

```

11:07:19.411163 IP machine1 > machine2: AH(spi=0x00000200, seq=0x8) :
ESP(spi=0x00000201, seq=0x8), length 88

```

- le pong : on voit un paquet crypté arriver et on voit bien que ce n'est pas le paquet ICMP echo-reply en clair :

```

11:07:19.411830 IP machine2 > machine1: AH(spi=0x00000300, seq=0x8) :
ESP(spi=0x00000301, seq=0x8), length 88

```

vi. D'autres exemples de configuration IPSec

Pour plus d'informations sur les configurations avec IPSec-tools, voir <http://www.ipsec-howto.org/x299.html>.

c) Configuration avec clés automatique par IKE et certificats pour utilisateur itinérant (serveur et client Linux)

Si le protocole IKE et son serveur racoon génèrent les clés à la volée et les SA pour la SAD, il ne gère pas les règles de police de sécurité à mettre dans la SPD. Et cela est évident, comment pourrait-t-il savoir ce que veut l'admin et entre quoi et quoi se fait l'encryptage.

i. Installation de ipsec-tools

Si vous souhaitez utiliser l'authentification hybride entre deux machines Linux (comme le montre la configuration suivante), il vous sera nécessaire de disposer d'un noyau 2.6 et de compiler les « ipsec-tools » dans une version supérieure à 0.5.2 (version packagée dans Debian Sarge, par exemple, mais

qui ne contient pas ce type d'authentification hybride). Pour cela, il vous sera nécessaire d'installer les paquets suivants :

```
[root]# apt-get install libssl0.9.6 libssl-dev flex bison  
libreadline5 libreadline5-dev libc6-dev bzip2 libpam0g-dev
```

On pourra télécharger le .tar.bz2 à l'adresse <http://ipsec-tools.sourceforge.net/>

(1) Compilation sur le client

```
[root]# tar xjvf ipsec-tools-*.tar.bz2  
[root]# cd ipsec-tools-  
[root]# ./configure --enable-natt --enable-frag --enable-hybrid \  
--enable-dpd --enable-adminport \  
--sysconfdir=/etc/racoon --localstatedir=/var \  
--prefix=/usr  
[root]# make  
[root]# make install
```

(2) Compilation sur le serveur

```
[root]# tar xjvf ipsec-tools-*.tar.bz2  
[root]# cd ipsec-tools-  
[root]# ./configure --enable-natt --enable-frag --enable-hybrid \  
--enable-dpd --sysconfdir=/etc/racoon --with-libpam \  
--prefix=/usr  
[root]# make  
[root]# make install
```

ii. Générer son autorité de certification

Il peut être nécessaire de générer une autorité de certification maison pour signer ses propres certificats. Un certificat CA n'est rien de plus qu'un certificat autosigné :

```
[root]# cd /dossier/de/stockage  
[root]# mkdir -p demoCA/newcerts  
[root]# touch demoCA/index.txt  
[root]# echo "00" > demoCA/serial  
[root]# mkdir certs  
[root]# openssl genrsa -des3 -out certs/ca.key 2048
```

Cette commande vous demande une passphrase pour protéger la clé privée créée.

```
[root]# chmod 600 certs/*.key  
[root]# openssl req -days 3650 -x509 -key certs/ca.key -new -out certs/ca.crt
```

Cette dernière commande vous demandera des informations sur le certificat de l'autorité CA créée et la passphrase pour décrypter la clé privée de l'autorité.

Ensuite, on va conserver ces deux clés afin de signer tous les certificats que l'on va générer par la suite.

iii. Génération des certificats du serveur et des clients

Note : le Common Name est celui du fichier, par exemple CN = client_ipsec, certificat client_ipsec.crt. De plus, Organization Name doit être le même pour le certificat CA et pour les certificats des clients et du serveur.

Il faut d'abord modifier la ligne contenant `dir` dans le fichier `/etc/openssl/openssl.cnf` (ou `/etc/pki/tls/openssl.cnf`) pour la faire pointer vers `/dossier/de/stockage/demoCA`.

Il faut générer un certificat pour toutes les machines qui vont communiquer (par exemple avec une clé de 1024 bits) :

```
[root]# /usr/bin/openssl genrsa -out certs/fichier.key nb_bits_clé
[root]# chmod 600 certs/*.key
```

Il ne faut pas donner de passphrase car racoon ne sait pas les décrypter.

```
[root]# /usr/bin/openssl req -new -key certs/fichier.key -out
certs/fichier.csr
[root]# openssl ca -cert /dossier/de/stockage/ca.crt -keyfile
/dossier/de/stockage/ca.key -in certs/fichier.csr -out
certs/fichier.crt -days nb_jour_validité
```

Le certificat généré se trouve alors dans le fichier `fichier.crt` et la clé privée dans `fichier.key`.

iv. Mise en place des certificats et autres clés

Chaque machine doit avoir sa propre clé privée, son certificat et la certificat de la CA dans son répertoire `/etc/certs/` (root:root 660) . Il doit aussi avoir le certificat des hôtes distants auxquels il se connecte.

Pour le serveur IPsec :

- le certificat de la CA : `ca.crt`
- sa clé privée et son certificat : `serv_ipsec.key` (root:root 600) et `serv_ipsec.crt`
- le certificat de tous ces clients : `client_ipsec.crt`

Pour les clients :

- le certificat de la CA : `ca.crt`
- sa clé privée et son certificat : `client_ipsec.key` (root:root 600) et `client_ipsec.crt`
- le certificat du serveur IPsec : `serv_ipsec.crt`

v. Configuration de racoon

Le démon racoon sert à gérer le protocole IPsec pour le mettre en place automatiquement par IKE. Il peut se lancer par `racoon -f /etc/racoon/racoon.conf` ou `/etc/init.d/racoon start`.

(1) Serveur

Il faudra copier le script `phase1-down.sh` dans le dossier `/etc/racoon` :

```
[root]# cp /usr/share/doc/racoon/examples/samples/roadwarrior/server/phase1-
down.sh /etc/racoon/
```

```
[root]# chmod 755 /etc/racoon/*.sh
```

Dans un fichier /etc/racoon/motd, on pourra mettre un message de bienvenue aux utilisateurs se connectant.

Pour le serveur le fichier /etc/racoon/racoon.conf contient :

```
#config donnée depuis
# /usr/share/doc/racoon/examples/samples/roadwarrior/server/racoon.conf

#chemin où sont stockés les certificats et clés privées
path certificate "/etc/certs";

#indique les clients distants autorisés
remote anonymous {
    #mode nécessaire à un utilisateur itinérant
    exchange_mode aggressive;
    #certificat du serveur et clé privée
    certificate_type x509 "serv_ipsec.crt" "serv_ipsec.key";
    #certificat de l'autorité de certification
    ca_type x509 "ca.crt";
    #indique d'utiliser le CN des certificats pour s'identifier
    my_identifier asn1dn;
    #obéir à la proposition de jeu de cryptage des clients
    proposal_check obey;
    #générer les règles IPSec à la connexion des clients
    generate_policy on;
    #permet de traverser une passerelle NAT entre le serveur et le client
    nat_traversal on;
    #indique le délai de régénération des clés
    dpd_delay 20;
    #indique que l'on peut fragmenter les paquets IKE et autres
    ike_frag on;
    #indique un script pour nettoyer la connexion à la déconnexion
    script "/etc/racoon/phase1-down.sh" phase1_down;
    #proposition de jeu de cryptage aux clients pour la phase 1
    proposal {
        encryption_algorithm 3des;
        hash_algorithm sha1;
        authentication_method hybrid_rsa_server;
        dh_group 2;
    }
}

#indique la configuration réseau à donner aux clients
mode_cfg {
    #première adresse à attribuer au client
    network4 première_IP_réseau_VPN;
    #nombre d'IP donc de clients possibles
    pool_size nb_clients;
    #masque du réseau des clients VPN
    netmask4 masque_réseau_VPN;
    #vérifier les login/mots de passes à partir de PAM sur le serveur
    auth_source pam;
    #adresse IP du DNS pour les clients
    dns4 IP_serveur_DNS;
    #adresse IP du WINS pour les clients
    wins4 IP_serveur_WINS;
    #chemin du fichier contenant le message de bienvenu pour les clients
    banner "/etc/racoon/motd";
}

#jeu de cryptage pour la phase 2 d'IPSec (dont trafic)
sainfo anonymous {
```

```

    pfs_group 2;
    lifetime time 12 hour;
    encryption_algorithm 3des;
    authentication_algorithm hmac_shal;
    compression_algorithm deflate;
}

```

(2) Client

Il faudra copier le script `phase1-down.sh` et `phase1-up.sh` dans le dossier `/etc/racoon` :

```
[root]# cp /usr/share/doc/racoon/examples/samples/roadwarrior/server/*.sh
/etc/racoon/
```

```
[root]# chmod 755 /etc/racoon/*.sh
```

Pour le serveur le fichier `/etc/racoon/racoon.conf` contient :

```

#config donnée depuis
# /usr/share/doc/racoon/examples/samples/roadwarrior/client/racoon.conf

#chemin du fichier contenant les certificats et clés privées
path certificate "/etc/certs";

#indique de maintenir les correspondances NAT sur la passerelle
# en envoyant des paquets « vides » toutes les 5 secondes
timer {
    natt_keepalive 5sec;
}

#permet le contrôle de racoon par la commande racoonctl (en root)
listen {
    adminsock "/var/racoon/racoon.sock" "root" "operator" 0660;
}

#configuration pour la connexion au VPN
remote IP_serveur_VPN {
    #seul mode pour utilisateur itinérant
    exchange_mode aggressive;
    #certificat de l'autorité de certification
    ca_type x509 "ca.crt";
    #obéir au jeu de cryptage du serveur
    proposal_check obey;
    #autorisé à traverser une passerelle NAT entre le client et le serveur
    nat_traversal on;
    #autorise la fragmentation des paquets
    ike_frag on;
    #autorise le serveur à envoyer la configuration pour nous
    mode_cfg on;

    #ne pas vérifier les identifiants
    verify_identifier off;
    #vérifier les certificats
    verify_cert on;

    #certificat et clé privée du client
    certificate_type x509 "client_ipsec.crt" "client_ipsec.key";
    #utiliser le CN des certificats pour le serveur
    my_identifier asn1dn;
    #utiliser le CN des certificats pour les clients
    peers_identifier asn1dn;

    #script pour activer la configuration envoyée par le serveur
    script "/etc/racoon/phase1-up.sh" phase1_up;
}

```

```

#script pour désactiver la configuration à la déconnexion
script "/etc/racoon/phase1-down.sh" phase1_down;
#initier la connexion au serveur VPN
passive off;
#jeu de cryptage pour la phase 1
proposal {
    encryption_algorithm 3des;
    hash_algorithm sha1;
    authentication_method hybrid_rsa_client;
    dh_group 2;
}
}

#jeu de cryptage pour la phase 2 (dont trafic)
sainfo anonymous {
    pfs_group 2;
    lifetime time 12 hour ;
    encryption_algorithm 3des;
    authentication_algorithm hmac_sha1;
    compression_algorithm deflate ;
}
}

```

(3) Mémorisation du mot de passe sur le client

Si vous voulez ne pas avoir à taper votre mot de passe Xauth à chaque connexion, vous pouvez le mettre dans le fichier des clés prépartagées.

Pour cela, dans le fichier `/etc/racoon/racoon.conf`, il faut ajouter :

```

#chemin et nom du fichier contenant des clés prépartagées
path pre_shared_key "/etc/racoon/psk.txt";

```

et dans sa section `remote` :

```
xauth_login "nom_utilisateur";
```

Puis dans le fichier `/etc/racoon/psk.txt` :

```
nom_utilisateur mot_de_passe
```

Ceci vous permettra de vous connecter simplement en faisant :

```
[root]# racoonctl vc IP_serveur_VPN
```

(4) Notes

Note : les certificats qu'il soit au format PEM ou au format `.key/pub` doivent absolument être décrypté (et donc lisible uniquement par l'utilisateur sous lequel tourne racoon) car racoon n'a pas la possibilité de vous demander le mot de passe.

Note pour Windows : les certificats pour Windows sont au format PKCS#12 :

```
[root]# openssl pkcs12 -export -inkey fichier_key.pem -certfile
/etc/certs/cacert.pem -in fichier_cert.pem -out windows_client.p12
```

vi. Démarrage

Racoon va générer les SA et les règles de police pour vous.

Le démarrage se fait comme suit :

```
[root]# /etc/init.d/racoon start
```

d) Connexion et déconnexion au VPN

Il est nécessaire d'exécuter les commandes suivantes :

- pour se connecter

```
[root]# racoonctl vc -u utilisateur IP_ou_nom_VPN
```

- pour se déconnecter

```
[root]# racoonctl vd IP_ou_nom_VPN
```

Il vous sera alors nécessaire de donner votre mot de passe tel que définit dans `/etc/passwd` sur le serveur VPN.

La connexion (échange de clés) se fait ensuite à la seconde tentative d'émission de paquets au travers du VPN. Ne soyez donc pas surpris si la première tentative de ping échoue et que la seconde réussie.

2. Problème de NAT, de fragmentation et de délai de connexion

a) NAT

Si le client se trouve derrière une passerelle assurant le NAT tel que c'est souvent le cas lorsque l'on a un réseau local et une passerelle pour accéder au net, vous ne pourrez pas utiliser IPSec directement. En effet, les paquets sont encapsulé dans des paquets ESP (au dessus de IP) qui par définition ne possède pas de port est donc très difficile à NATer. Dans ce cas, on utilisera NATT afin d'encapsuler les paquets ESP dans des paquets UDP pour passer les passerelles.

Pour ce faire, dans le fichier `/etc/racoon/racoon.conf`, dans la section `remote`, on mettre `nat_traversal on`;

De plus, il peut arriver que l'échange IKE ne se fasse pas si le port UDP du NAT change du fait de la faible persistance des associations de NAT dans les passerelles. Il peut donc être utile de faire des envois de paquets « vide » pour faire persister l'association NAT sur la passerelle traversée.

Pour ce faire, on ajoutera ceci dans le fichier `/etc/racoon/racoon.conf` :

```
timer {
    natt_keepalive 5sec; #10 seconde maxi
}
```

b) Fragmentation

La fragmentation intervient lorsque l'on utilise, par exemple, une liaison DSL qui suppose par défaut que les paquets UDP ne sont pas de grande taille. La fragmentation va donc arriver plus fréquemment dans un IPSec NATé.

Il y a deux types de fragmentation possible pour IPSec :

- la fragmentation des paquets IKE lors de l'établissement des clés IPSec. On peut choisir la fragmentation IKE qui est une extension de ce protocole
- la fragmentation des paquets ESP lors des communications. Dans ce cas, il faut simplement fragmenter les paquets IP encapsuler dans ESP pour faire des paquets UDP/ESP de plus petite taille.

Dans le fichier `/etc/racoon/racoon.conf`, dans la section `remote`, on mettra `ike_fragmentation;`

c) Delai de connexion

Il peut arriver que la connexion entre le client et le serveur ne soit pas très fiable et donc que des déconnexions arrivent. Pour cela, on peut forcer le serveur IKE à renouveler les clés régulièrement.

Pour cela, dans le fichier `/etc/racoon/racoon.conf`, on ajoutera dans la section `remote`, `dpd_delay 20;`

3. Et iptables dans tout ça...

NOTE : la configuration suivante est valable uniquement pour le mode Tunnel. Le mode transport à un intérêt limité dans le VPN.

a) La passerelle IPsec machine 1

Si l'on veut limiter le forward aux paquets cryptés, il est nécessaire de marquer les paquets esp et/ou ah entrants :

```
iptables -t mangle -A PREROUTING -p 50 -j MARK --set-mark 1
iptables -t mangle -A PREROUTING -p 51 -j MARK --set-mark 1
```

Déjà, il faut autoriser ESP ou AH ou les deux en fonction de la configuration d'IPsec :

#ESP

```
iptables -A OUTPUT -p 50 -s IP_pub_machine1 -d IP_pub_machine2 -j
ACCEPT
```

```
iptables -A INPUT -p 50 -s IP_pub_machine2 -d IP_pub_machine1 -j
ACCEPT
```

#AH

```
iptables -A OUTPUT -p 51 -s IP_pub_machine1 -d IP_pub_machine2 -j
ACCEPT
```

```
iptables -A INPUT -p 51 -s IP_pub_machine2 -d IP_pub_machine1 -j
ACCEPT
```

Ensuite, il faut autoriser IKE :

```
iptables -A INPUT -p udp --sport 500 --dport 500 -s
IP_pub_machine2 -j ACCEPT
```

```
iptables -A OUTPUT -p udp --sport 500 --dport 500 -d
IP_pub_machine2 -j ACCEPT
```

#si on fait du NAT-T

```
iptables -A INPUT -p udp --sport 4500 --dport 4500 -s
IP_pub_machine2 -j ACCEPT
```

```
iptables -A OUTPUT -p udp --sport 4500 --dport 4500 -d
IP_pub_machine2 -j ACCEPT
```

Ensuite, il faut autoriser le trafic entre les deux réseaux internes (si les paquets sont cryptés) :

```
# autoriser le transit par la passerelle
iptables -A FORWARD -m mark --mark 1 -j ACCEPT

# effectuer la NAT des paquets traversant
iptables -t nat -A POSTROUTING -s adresse_reseau1 -o
interface_public -j SNAT --to-source IP_public_machine1

#active le forwarding
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Enfin, il faut indiquer à la machine 1, la route vers le réseau interne 2. Note, si la machine 1 n'est pas la passerelle par défaut du réseau 1, il faut indiquer une route vers la machine 1 sur la passerelle par défaut :

```
route add adresse_reseau2 gateway IP_pub_machine2
```

b) La passerelle IPSec machine 2

La configuration est la même mais dans l'autre sens : on inverse adresse_reseau1 par adresse_reseau2, etc...

c) En plus sur les passerelles IPSec

Ensuite, on peut faire des règles de filtrages sur la table filter comme si on était une simple passerelle.

d) En plus sur le client

Le client est considéré avec la configuration précédente mais sans les règles de forward.

III. IPSec seul sous Windows

1. Configuration par clés prépartagées

a) Le fichier de configuration de racoon

```
#chemin et nom du fichier contenant des clés prépartagées
path pre_shared_key "/etc/racoon/psk.txt";

#désactive le socket local d'administration
listen {
    adminsock disabled;
}

#autorise tout clients à se connecter : roadwarrior

#phase1
```

```

remote anonymous {
    #mode principal
    exchange_mode main;
    #générer la SPD automatiquement
    #(vu que l'on ne connaît pas l'IP du client à l'avance)
    generate_policy on;

    #s'adapter au « jeu de cryptage » du client
    proposal_check obey;
    #identifier le client par son login @ son nom de domaine
    peers_identifier user_fqdn;
    #attendre les connexions des clients
    passive on;
    #autoriser IPSec par NAT
    nat_traversal on;

    #jeu de cryptage proposé au client
    proposal {
        encryption_algorithm 3des;
        hash_algorithm sha1;
        authentication_method pre_shared_key;
        dh_group modp1024;
    }
}

#phase2
#jeu de cryptage
sainfo anonymous {
    pfs_group 2;
    encryption_algorithm 3des;
    authentication_algorithm hmac_md5;
    compression_algorithm deflate;
}

```

b) Le fichier de clé prépartagées : psk.txt

Sur chaque ligne de ce fichier, on indiquera un couple [login@domaine](#) client <=> mot de passe textuel dans un des formats suivants :

```

login@domaine    mot_de_passe
IP               mot_de_passe

```

c) Préparation de MMC

- Cliquer sur Démarrer->Exécuter et taper 'mmc' et cliquer sur OK. La console de gestion devrait apparaître.
- Cliquer sur Fichier->Ajouter/Supprimer un composant logiciel enfichable...
- Cliquer sur Ajouter puis sélectionner « Gestion de la stratégie de sécurité du protocole IP » puis Ajouter
- Sélectionner « Ordinateur local » puis Terminer puis Fermer puis OK

d) La politique de sécurité

- Cliquer sur « Stratégies de sécurité IP sur Ordinateur local »
- Cliquer à droite dans le panneau de droite
- Cliquer à gauche sur « Créer une stratégie de sécurité IP »

- Cliquer sur Suivant, donner un nom et une description à votre politique de sécurité puis Suivant
- **Décocher « Activer la règle de réponse par défaut » puis Suivant.**
- **Cocher/Vérifier « Modifier les propriétés » puis Terminer**
- **Décocher « Utiliser l'Assistant Ajout »**
- *Ajout de la règle de filtrages Sortante (SPD)*
 - Cliquer sur « Ajouter » pour créer un nouvelle règle
 - Cliquer sur « Ajouter » pour créer une nouvelle *liste de filtres IP*
 - Appeler-la, par exemple, « Moi vers Serveur VPN » et donner lui éventuellement une description
 - **Décocher « Utiliser l'Assistant Ajout »**
 - *Filtre IP* : Cliquer sur « Ajouter » pour créer *un filtre IP*
 - Pour « Adresse Source », choisir « Mon adresse IP »
 - Pour « Adresse Destination »,
 - Si vous voulez **SEULEMENT** utiliser IPSec **ENTRE** le serveur VPN **ET** vous (note : si le serveur VPN est une passerelle et que vous cherchez à atteindre son réseau interne, alors le trafic sera en clair tout le temps)
 - « Un nom DNS spécifique » et entrer le nom DNS de votre serveur VPN racoon s'il en a un.
 - « Une adresse IP spécifique » et entrer l'IP de votre serveur
 - Si vous souhaitez faire un tunnel entre vous et le réseau se trouvant derrière le serveur VPN :
 - « Un sous-réseau IP spécifique » et entrner son adresse et son masque si votre serveur vous permet d'accéder à son sous-réseau interne.
 - Dans l'onglet « Protocole », vous pouvez choisir un autre protocole que « N'importe lequel », pazr exemple, ICMP, TCP ou UDP...si vous voulez que IPSec ne soit pas utiliser pour tout le trafic vers le serveur VPN.
 - **Décocher « Image miroir » puis cliquer sur OK et OK**
 - Cocher le filtre IP « Moi vers Serveur VPN » dans la liste des filtres IP
 - *Action de filtrage* : Dans l'onglet « Action de filtrage »
 - **Décocher « Utiliser l'Assistant Ajout »**
 - Cliquer sur « Ajouter » pour créer une nouvelle action de filtrage
 - Cocher/Vérifier « Négocier la sécurité »
 - Cliquer sur « Ajouter » pour définir une nouvelle méthode de sécurité
 - Cocher « Personnalisée » puis cliquer sur « Paramètres... »
 - Cocher « Cryptage et intégrité de données (ESP) »
 - Pour « Algorithme d'intégrité », sélectionner SHA1
 - Pour « Algorithme de cryptage », sélectionner AES (si présent) ou 3DES (sinon)
 - Cocher « Générer une nouvelle clé tous les » à gauche et « Générer une nouvelle clé toutes les » à droite
 - Cliquer sur OK et OK

- Cocher « Session de clé principale PFS (Perfect Forward Secrecy) »
 - Dans l'onglet Général, donner un nom à cette action de filtrage, par exemple, « Communication bilatérale sécurisée » et éventuellement une description, comme « Cette action de filtrage requière que les deux machines communiquent en utilisant IPSec »
 - Cliquer sur OK
- Cocher l'action de filtrage « Communication bilatérale sécurisée »
- *Méthode d'authentification* : Dans l'onglet « Méthode d'authentification »
 - Cliquer sur Editer
 - Cocher « Utiliser cette chaîne (clé partagée au préalable) »
 - Entrer la clé prépartagée puis OK
- Si vous avez choisi un sous réseau dans le filtre IP, alors :
 - Dans l'onglet « Paramètres du tunnel », cocher « Le point d'arrêt du tunnel est spécifié par cette adresse IP et entrer l'adresse IP du serveur VPN.
- Cliquer sur OK pour fermer la fenêtre de modification des Propriétés de la Nouvelle règle
- *Ajout de la règle de filtrages Entrante (SPD)*
 - Cliquer sur « Ajouter » pour créer un nouvelle règle
 - Cliquer sur « Ajouter » pour créer une nouvelle *liste de filtres IP*
 - Appeler-la, par exemple, « Serveur VPN vers Moi » et donner lui éventuellement une description
 - **Décocher « Utiliser l'Assistant Ajout »**
 - *Filtre IP* : Cliquer sur « Ajouter » pour créer un filtre IP
 - Pour « Adresse Source »,
 - Si vous voulez **SEULEMENT** utiliser IPSec **ENTRE** le serveur VPN **ET** vous (note : si le serveur VPN est une passerelle et que vous cherchez à atteindre son réseau interne, alors le trafic sera en clair tout le temps)
 - « Un nom DNS spécifique » et entrer le nom DNS de votre serveur VPN racoon s'il en a un.
 - « Une adresse IP spécifique » et entrer l'IP de votre serveur
 - Si vous souhaitez faire un tunnel entre vous et le réseau se trouvant derrière le serveur VPN :
 - « Un sous-réseau IP spécifique » et entrner son adresse et son masque si votre serveur vous permet d'accéder à son sous-réseau interne.
 - Pour « Adresse Destination », choisir « Mon adresse IP »
 - Dans l'onglet « Protocole », vous pouvez choisir un autre protocole que « N'importe lequel », pazr exemple, ICMP, TCP ou UDP...si vous voulez que IPSec ne soit pas utiliser pour tout le trafic vers le serveur VPN.
 - **Décocher « Image miroir » puis cliquer sur OK et OK**
 - Cocher « Serveur VPN vers Moi »
 - Dans l'onglet « Action de filtrage », sélectionner « Communication bilatérale sécurisée »
 - Dans l'onglet « Méthode d'authentification », cliquer sur Editer

- Cocher « Utiliser cette chaîne (clé partagée au préalable) »
- Entrer la clé prépartagée puis OK
- Si vous avez choisi un sous réseau dans le filtre IP, alors :
 - Dans l'onglet « Paramètres du tunnel », cocher « Le point d'arrêt du tunnel est spécifié par cette adresse IP et entrer votre adresse IP
- Cliquer sur OK pour fermer la fenêtre de modification des Propriétés de la Nouvelle règle
- Cliquer sur OK pour fermer les propriétés de la Nouvelle stratégie.
- Cliquer à droite sur la nouvelle stratégie que vous venez de créer puis cliquer à gauche sur « Assigner » et vérifier qu'un point vert est apparu sur l'icone de la nouvelle stratégie

2. Configuration par certificats

a) Le fichier de configuration de racoon

```
#chemin du dossier contenant les certificats
path certificate "/etc/certs/";

#désactive le socket local d'administration
listen {
    adminsock disabled;
}

#autorise tout clients à se connecter : roadwarrior

#phase1
remote anonymous {
    #mode principal
    exchange_mode main;
    #générer la SPD automatiquement
    #(vu que l'on ne connaît pas l'IP du client à l'avance)
    generate_policy on;

    #ne pas vérifier les identifiants ASN1DN des clients
    verify_identifier off;
    #vérifier les certificats
    verify_cert on;

    #le certificats de la CA
    ca_type x509 "ca.crt";
    #le certificat du serveur
    certificate_type x509 "serv_ipsec.crt" "serv_ipsec.key";
    #l'identifiant du serveur est le champ Subject de son certificat
    my_identifier asn1dn;
    #s'adapter au « jeu de cryptage » du client
    proposal_check obey;
    #l'identifiant du client est le champ Subject de son certificat
    peers_identifier asn1dn;
    #attendre les connexions clientes
    passive on;
    nat_traversal on;
    #jeu de cryptage proposée au client
    proposal {
        encryption_algorithm 3des;
        hash_algorithm sha1;
        authentication_method rsasig;
        dh_group modp1024;
    }
}
```

```
#phase2
#jeu de cryptage
sainfo anonymous {
    pfs_group 2;
    encryption_algorithm 3des;
    authentication_algorithm hmac_md5;
    compression_algorithm deflate;
}
```

b) Génération des certificats

Voir dans la section « Configuration avec clés automatique par IKE et certificats » pour ce qui concerne la génération de son autorité de certification et des certificats serveur et clients.

c) L'exportation des certificats

Les certificats Clients pour Windows sont au format PKCS#12.

Il faut donc convertir le certificat client généré :

```
[root]# openssl pkcs12 -export -inkey fichier_key_client.pem
-certfile /etc/certs/cacert.pem -in fichier_cert_client.pem -out
windows_client.p12
```

On prendra donc :

- ca.crt
- *windows_client.p12*
- serv.crt

d) Préparation de MMC

- Cliquer sur Démarrer->Exécuter et taper 'mmc' et cliquer sur OK. La console de gestion devrait apparaître.
- Cliquer sur Fichier->Ajouter/Supprimer un composant logiciel enfichable...
- Cliquer sur Ajouter puis sélectionner « Gestion de la stratégie de sécurité du protocole IP » puis Ajouter
- Sélectionner « Ordinateur local » puis Terminer
- Cliquer sur Ajouter puis sélectionner « Certificats » puis Ajouter
- Sélectionner « Le compte de l'ordinateur » puis Terminer puis Fermer puis OK

e) L'importation des certificats

- Ouvrir « Certificats (ordinateur local) » puis cliquer sur « Autorités de certification racines de confiance ».
- Cliquer sur Action->Toutes les tâches->Importer...
- Cliquer sur Suivant puis choisir le fichier ca.crt puis Suivant
- **Sélectionner « Sélectionner automatiquement le magasin de certificats selon le type de certificat » puis Suivant puis Terminer.** Il faut absolument laisser Windows XP choisir sinon IPsec ne marchera pas.
- Recommencer pour le certificat du serveur (s'il n'apparaît pas c'est qu'il est dans « Autres

personnes » du profil « Utilisateur courant » et pas « Ordinateur local »).

- Cliquer sur Action->Toutes les tâches->Importer...
- Cliquer sur Suivant puis choisir le fichier *windows_client.p12* puis Suivant
- Enter le mot de passe que vous avez donné à l'exportation puis Suivant
- **Sélectionner « Sélectionner automatiquement le magasin de certificats selon le type de certificat » puis Suivant puis Terminer.** Il faut absolument laisser Windows XP choisir sinon IPSec ne marchera pas.

f) La politique de sécurité

La méthode pour définir la politique de sécurité est la même que pour les clés prépartagées à l'exception de la méthode d'authentification :

- *Méthode d'authentification* : Dans l'onglet « Méthode d'authentification »
 - Cliquer sur Editer
 - Cocher « Utiliser un certificat émis par cette Autorité de Certification »
 - Cliquer sur « Parcourir » et choisissez le certificat CA que vous avez importé puis OK pour sélectionner le certificat

3. Débogage

If you are still with us at this point, you should now have an IPSec transport connection between your Windows XP SP2 computer and your Gentoo IPSec server. When you issue a ping to your Gentoo server, you should see a couple of messages saying "Negotiating IP Security". After 1-4 of these messages, your pings should start going through. If you continue to receive these messages, it is most likely that you either did not import your certificates correctly, you did not enter the correct IP address in one of the fields of the IP filters, or you did not select the correct options for the filter actions. It is also possible that you messed something up on your Gentoo side. To troubleshoot the connection, turn on Logon/Logoff auditing (Control Panel->Administrative Tools->Local Security Policy->Local Policies->Audit Policy) and use the Security Log in the Event Viewer.

IV. Bibliographie

[The official IPsec Howto for Linux](#)

[Chapitre 7. IPSEC: IP sécurisé à travers Internet](#)

[IPsec Tools Homepage](#)

[FreeS/WAN Project: Home Page](#)