Document de référence

sur l'installation d'un serveur VPN redondant avec OpenBSD 3.8

Version 1.2
07/02/2006 00:17

Guillaume Coqueblin (guillaume@coqueblin.com)

Table des matières

But du projet

L'objectif du projet est d'étudier la mise en place et le fonctionnement d'un VPN redondant sous OpenBSD 3.8 ainsi que l'utilisation des nouveaux outils implémentés dans cette version (Sortie officielle 3.8 stable, 1er novembre 2005), comme isakmpd, pfsync, sasync, carp, pf, ipsec.

But de cette documentation

Ce projet est avant tout un projet d'école, et jusqu'à présent il a été réalisé en tant que tel, ce qui explique sans doute à certains peut-être un manque de professionalisme ou de rigueur dans la rédaction de cette documentation, ou plus techniquement des méthodes employées pour la réalisation du projet et la rédaction des fichiers de configurations.

Les contraintes de recette de ce projet étaient assez libres, c'est pourquoi les formats des sites (flash) ou des documentations (word et pdf) ne sont pas pour le moment à même de satisfaire toutes les personnes interessées.

Ce projet, je le rapelle à nouveau, scolaire, présente une configuration parmis tant d'autres. C'est celle que nous avons mis en place au sein du laboratoire de projets.

Les valeurs des données n'ont pas été modifiées pour garder une certaine cohérence avec les schémas et explications, et aussi mais surtout car au cours de cette étude nous nous sommes heurtés de nombreuses fois à des exemples de configuration trop généralistes, qui ne répondaient pas aux questions que l'on a pu se poser.

De mon point de vue, il me semble plus simple pour une personne voulant adapter cette configuration à son environnement à reprendre les schémas, modifier les adresses IP, et à s'y référer par la suite en ayant qu'à remplacer nos valeurs par les siennes.

Cette documentation évolue tous les jours, et est ouverte à toute critique, remarque, modification, ou question comme c'en est actuellement le cas.

Objectifs de la solution proposée

- Réalisation de la plate-forme
- Installation d'OpenBSD sur les passerelles
- Une solution VPN doit permettre aux utilisateurs distants de se connecter aux ressources du réseau local.
- Configuration réseau (routage, filtrage avec pf)
- Configuration IPsec
- La solution doit assurer la confidentialité et l'intégrité des données à travers Internet.
- Mise en place de la redondance (CARP, pfsync, sasync)
- Pour l'utilisateur aucune différence ne doit apparaître dans son utilisation des ressources du système d'information.
- Mise en place de l'authentification par certificats
- La solution doit générer et mettre à jour les clefs pour le chiffrement des données pour le client et le serveur.
- Analyse du trafic et des logs.

Prérequis

- Deux PC (Type i386) avec une distribution OpenBSD 3.8.
- 6 cartes réseau 10/100 Mbits + câbles paires torsadées RJ45
- 1 switch 10/100/1000 Mbit
- Deux réseaux distincts au minimum (Dans notre cas nous utiliserons comme réseau utilisateur : itinet et comme réseau externe internet)

Lexique

IPsec : IPsec est un protocole permettant le transport de données chiffrées sur un réseau IP pour établir un réseau privé virtuel.

Redondant : La redondance, d'une façon générale, se rapporte à la qualité ou à l'état d'être en surnombre, par rapport à la normale ou à la logique. Ce qui peut avoir la connotation négative de superflu, mais aussi un sens positif quand cette redondance est voulue afin de prévenir un dysfonctionnement.

VPN : Virtual Private Network (Réseau Privé Virtuel) liaison entre deux réseaux distants.

Sasync : Service de synchronisation des règles de chiffrement d'IPsec lors d'une indisponibilité d'une passerelle

Carp: Common Address Redundancy Protocol (Protocole de redondance d'adresse commune), CARP permet a plusieurs hôtes dans un même réseau local de partager une plage d'adresse IP. Son but premier est de s'assurer que les adresses sont toujours disponibles, mais dans certaines configurations, carp, permet aussi de fournir de la répartition de charge.

Pf: Packet Filter, est le pare-feu d'OpenBSD.

Routage: Dans le cas présent, le routage est le transfert de paquets d'un réseau à un autre

Filtrage: Règles établies par le firewall qui autorisent ou refusent le routage des paquets

NAT: Network Adress Translation, permet de rendre des adresses du réseau local routable sur internet.

Haute Disponibilité: Le cluster de haute disponibilité est un ensemble de serveurs physiques, au nombre minimum de deux, afin d'obtenir une activité de services par tous temps, en toutes conditions

Présentation de l'architecture générale du projet

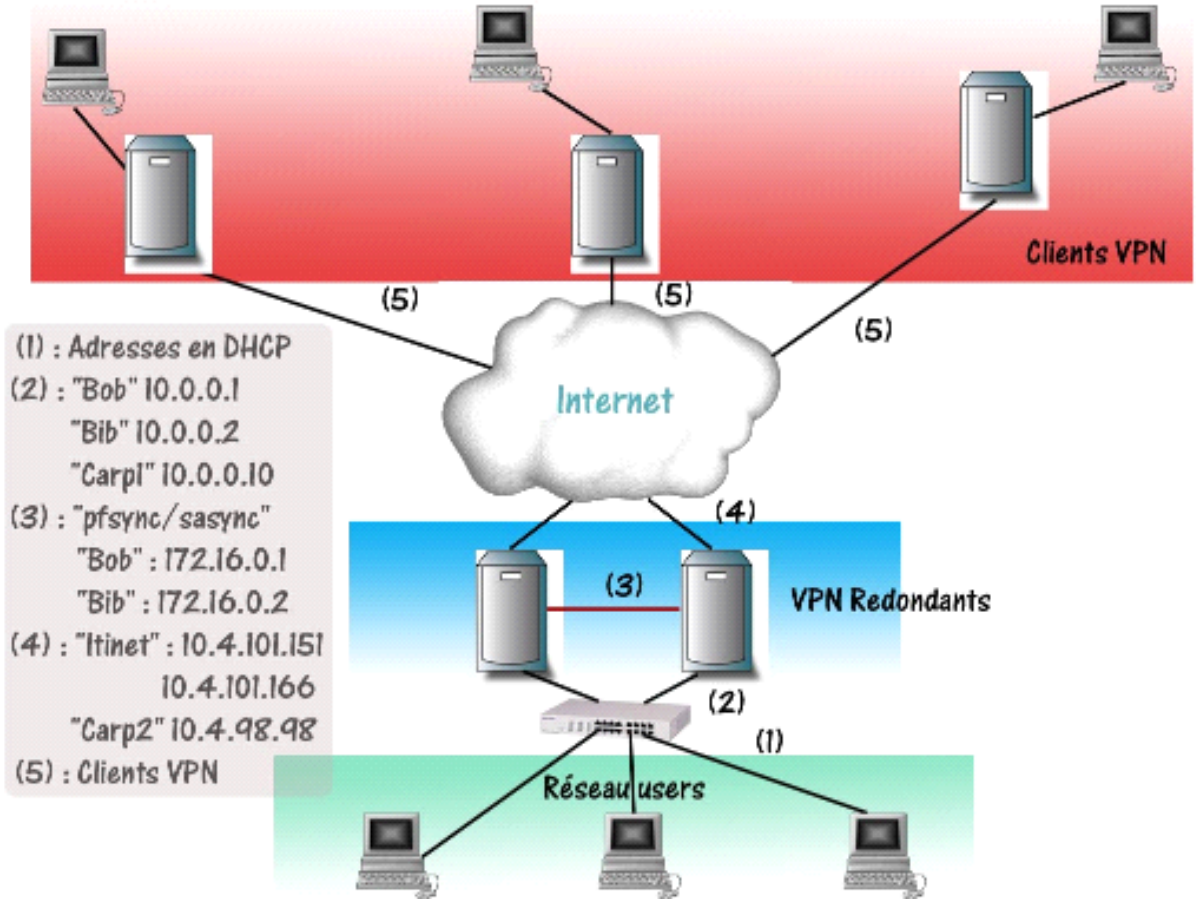
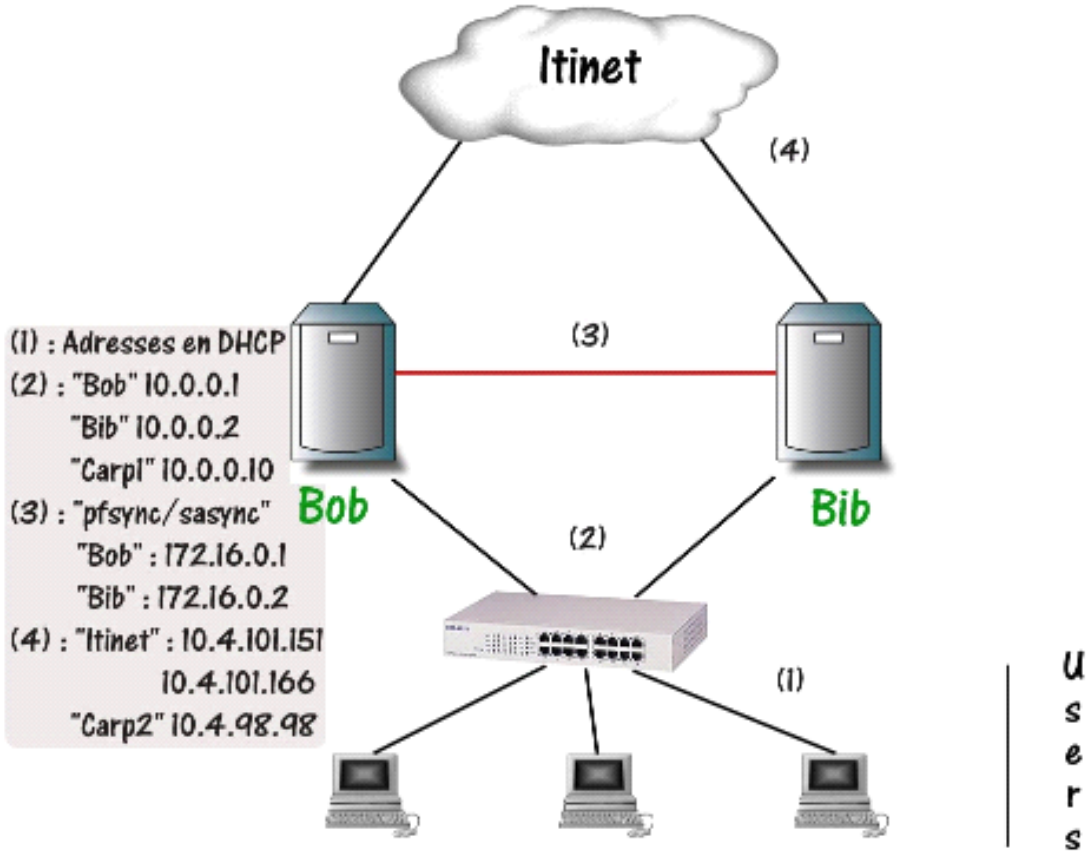


Schéma détaillant la partie « vpn redondant »



Fonctionnement de PF

Packet Filter se place dans le kernel.

Un périphérique virtuel, **/dev/pf**, permet, à travers une interface « **ioctl** », aux processus locaux de contrôler le comportement de **Packet Filter**.

Il y a des commandes pour activer et désactivers le filtre, charger des listes de règles, ajouter ou retirer des règles individuelles ou des entrées dans la table d'états, et récuperer des statistiques. Les fonctions les plus courantes sont disponibles avec « **pfctl** ».

Dans notre cas nous avons défini des règles de filtrage qui limitent le trafic au minimum requis, c'est-à-dire autoriser tout le trafic en sortie (du réseau utilisateur, vers internet), et limiter le trafic en entrée, soit aux connexions déjà établies, soit aux connexions entrantes venant des autres serveurs vpn qui vont venir se connecter à notre serveur.

Fonctionnement de CARP

L'interface **carp** est un périphérique virtuel qui implémente et contrôle le protocole **CARP**.
Carp permet à de multiples hôtes de partager un lot d'adresses IP et MAC sur le même réseau local.

Son but primaire est de s'assurer que ces adresses sont toujours disponibles, mais dans certains cas **carp** permet de fournir de la répartition de charge.

Pour cela il faut créer un hôte virtuel commun par réseau.

(dans notre cas 10.0.0.10 – pour le reseau « users » et 10.4.98.98 – pour le réseau « itinet »)

Fonctionnement de PFSYNC

L'interface **pfsync** est un périphérique virtuel qui permet de visualiser les modifications de la table d'états utilisée par **pf**. (Que l'ont peut visualiser en faisant « **tcpdump –i pfsync0** »).

Lorsque que **pfsync** est configuré pour être utilisé avec une interface physique de synchronisation, **pfsync** enverra aussi les changements d'états sur cette interface même en utilisant du multicast IP, et insérer les changements des états venant d'autres systèmes qui ont été reçus sur cette interface.

Par défaut, tous les changements de la table d'états faits en local sont traités par **pfsync**.

Par ailleurs les changements venant de paquets reçus par **pfsync** par le réseau ne sont pas « rebroadcastés ».

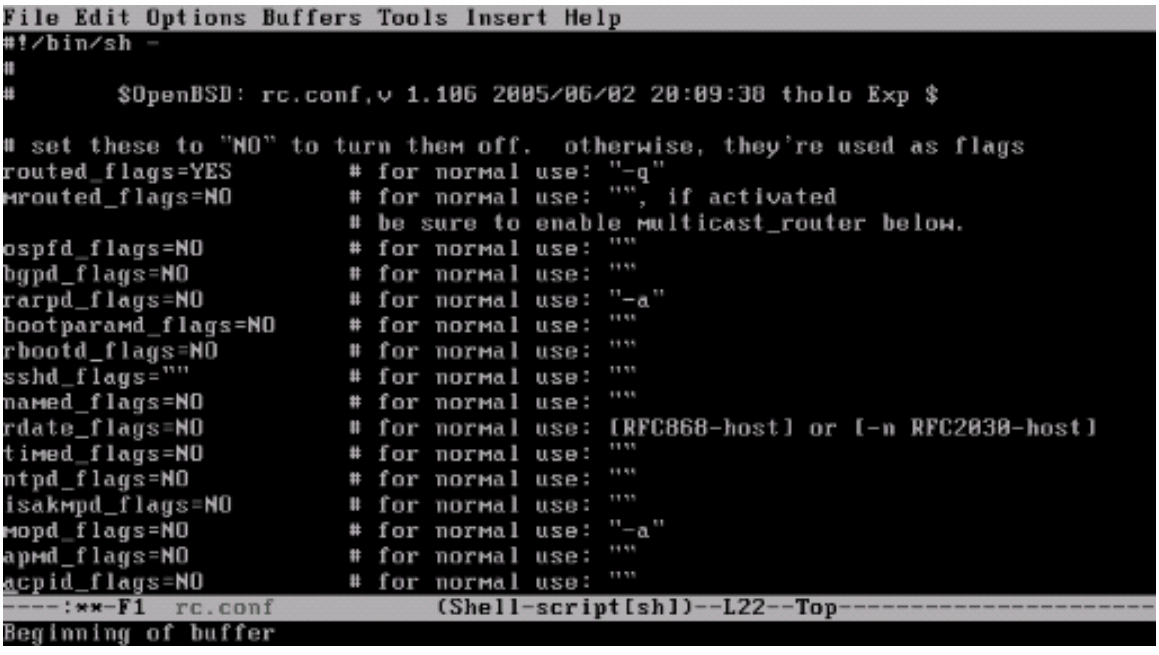
Les états crées par une règle marquée de la suite de mots « **no-sync** » ne sont pas transmis à l'interface **pfsync**

L'interface pfsync essayera au maximum de mettre un maximum de mises à jour du même état dans un seul message lorsque c'est possible.

Détails de la configuration de la première machine « BOB »

A Partir d'une installation d'OpenBSD 3.8, suivez les étapes de configuration.
Première vérification du fichier

/etc/rc.conf



En toute théorie ce fichier ne doit pas être modifié, on peut donc y trouver certaines variables définies à des valeurs qui ne correspondent pas à ce dont on a besoin.

Pour exemple, dans ce fichier, la variable « routed_flags » peut être, soit à « NO », soit à « YES ».

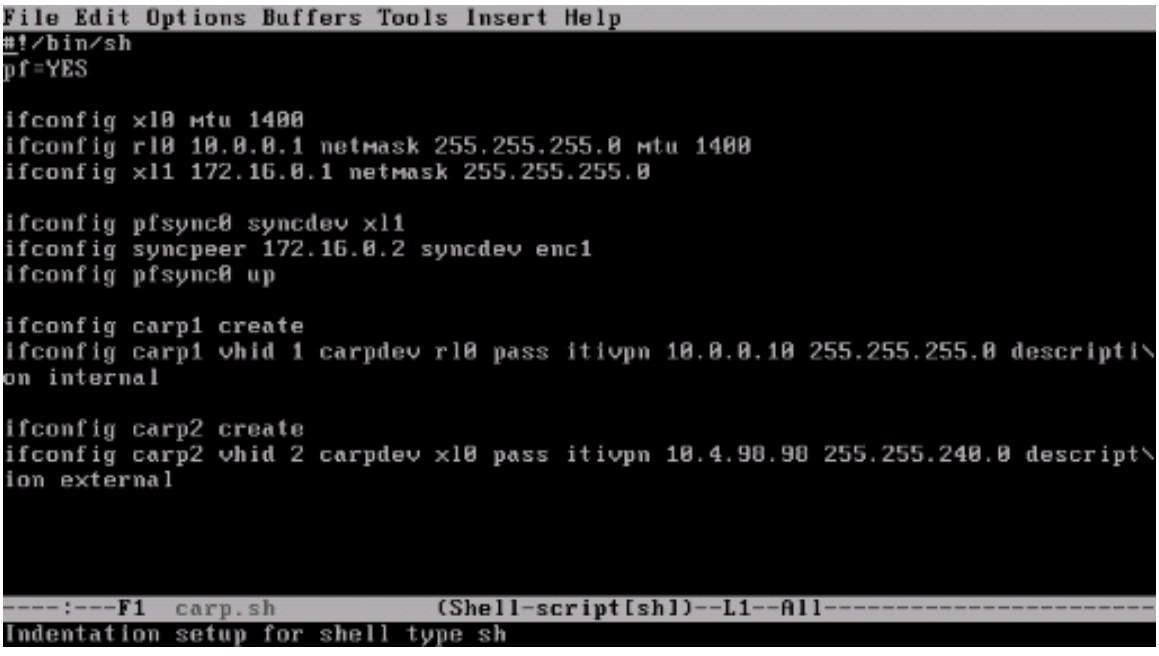
Quelque soit le cas, on a pas à s'en préoccuper, car par sécurité, dans le fichier rc.conf.local, on va lui définir la valeur « NO ».

/etc/rc.conf.local

Voilà le fichier dans lequel nous allons définir tous les paramètres de la configuration de notre machine.

Il faut ensuite créer un petit « shell-script » qui nous permettra de créer nos interfaces carp sur cette première machine ainsi que ses options de configuration que nous allons détailler.

/etc/carp.sh (ne pas oublier de le 'chmod +x')

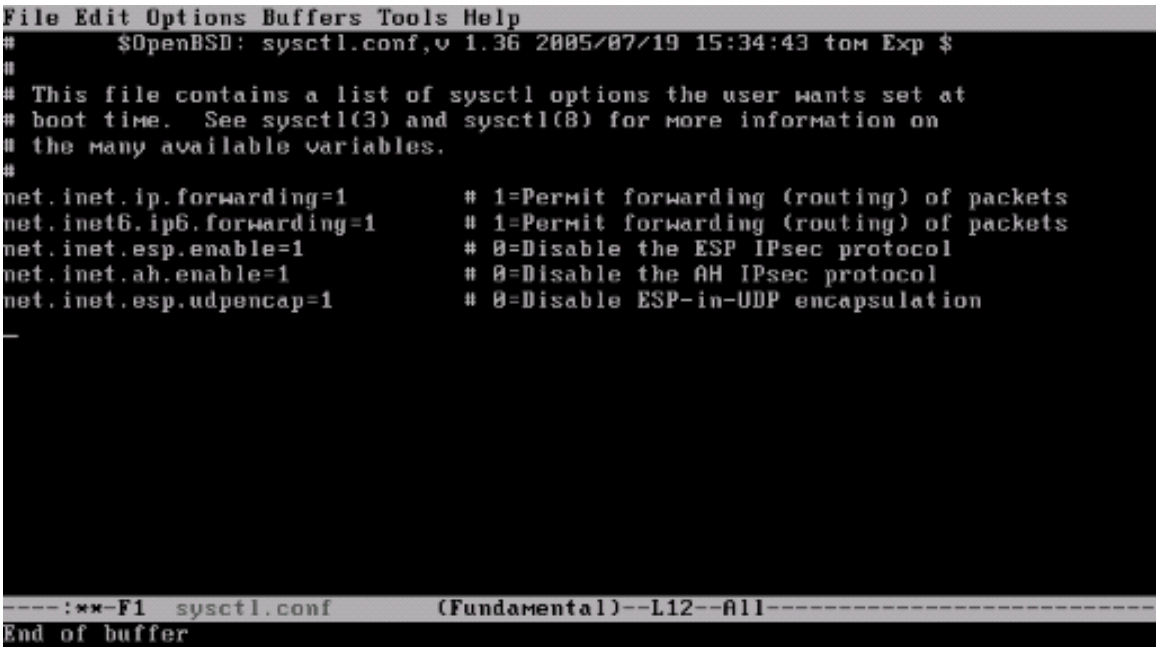


- On fixe la MTU à 1400 sur les cartes réseau entrantes et sortantes, c'est-à-dire celles connectées aux deux réseaux de chaque coté (utilisateurs et internet), pour éviter tout problèmes d'encapsulation (double NAT+IPsec).

- On crée et configure notre interface « pfsync0 », en lui attribuant comme adresse physique « xl1 », et en chiffrant les données vers 172.16.0.2 grâce à « enc1 »

- On crée et configure nos interfaces « carp1 » et « carp2 » en leur désignant respectivement « rl0 » et « xl0 » comme cartes réseau physiques, dont les options comportent le type d'interface (carpdev), la carte réseau utilisée, le mot de passe pour l'authentification entre les deux hôtes, l'adresse ip commune qu'ils vont utiliser, le masque, et pour finir une courte description pour plus de facilité de lecture de « ifconfig ».

/etc/sysctl.conf



Le fichier sysctl.conf utilisé par sysctl permet de fournir au kernel des options particulières relatives dans notre cas au forwarding de paquets (ipv4 et ipv6), à l'activation de l'esp et son encapsulation, et à l'activation de ah.

Les options présentes dans ce fichier sont indispensables au bon fonctionnement du serveur vpn.

/etc/pf.conf

Les règles présentes dans le fichier de configuration pf.conf (en fin de document), sont celles utilisées dans notre cas.

/etc/sasyncd.conf

```
File Edit Options Buffers Tools Help
carp interface carp1 interval 1
sharedkey fb76feb53a48d8dbce6a48ad836e748e
flushmode startup
listen on xl1 inet
mode master
peer 10.0.0.2

carp interface carp2 interval 1
sharedkey fb76feb53a48d8dbce6a48ad836e748e
flushmode startup
listen on rl0 inet
mode master
peer 10.4.101.166
-----:---F1  sasyncd.conf      (Fundamental)--L1--All----------------------------
For information about the GNU Project and its goals, type C-h C-p.
```

Le fichier sasyncd.conf nous permet de configurer l'interface carp qui va être utilisée, la clef partagée pour chiffrer les données (cette clef doit être la même sur les deux hôtes redondants), l'interface sur laquelle le service va écouter le trafic, le mode dans lequel il doit démarrer, et son autre hôte.

Nous sommes pour le moment à la configuration du premier serveur, c'est-à-dire celui qui a été défini comme le maître sur les deux machines.

C'est pourquoi nous retrouvons pour les deux adresses ip partagées le « mode master ». C'est-à-dire que c'est lui qui répondra aux requêtes diverses venant de n'importe quel réseau.

/etc/isakmpd/isakmpd.conf

Il se compose de plusieurs parties dont nous allons voir les parties cruciales, qui souvent posent problème (Attention, ceci n'est pas la configuration complète !):

- [General]

Listen-on= 10.4.98.98

 # Ici l'ip de notre interface qui est sur le réseau "Itinet"

#Authentication=

 # Cette option ne doit pas être utilisée lors de la combinaison d'isakmpd et des certificats.

- [Phase 1]

Default=

82.230.55.32= spawn-p1

 # Adresse ip d'un des clients qui va se connecter au vpn

81.57.45.105= eski-p1

 # Adresse ip du deuxieme. Le « p1 » dans les noms ici sont pratiques pour mieux visualiser à quelle zone du fichier de configuration cette phase correspond. Il est très peu conseillé d'utiliser des noms proches, qui pourraient porter a confusion.

82.123.13.236= spyke-p1

- [Phase2]

Passive-connections= spawn-p2, eski-p2, spyke-p2

Sur le serveur, les clients sont définis en connexions passives. Isakmpd va donc attendre une connexion venant de ces identifiants.

- [my-fqdn]

 # Toute cette zone de configuration correspond au serveur

ID-type= FQDN

 # Dans notre cas nous utilisons le nom de domaine. Pour utiliser une adresse ip fixe, il faut définir « ID-type » en IPV4_ADDR, et remplacer « name » par « adress ».

- [my-ipv4-net]

ID-type= IPV4_ADDR_SUBNET

 # Permet de definir que le reseau sera en ipv4

Network= 10.0.0.0

- [spawn-p1]

ID= my-fqdn

 # Le serveur utilisera son fqdn comme id

Remote-ID= spawn-fqdn

 # et le fqdn du client comme remote-id

- [spawn-ipv4-net]

ID-type= IPV4_ADDR_SUBNET

 # Comme pour le serveur juste au dessus on défini le type de réseau

Network= 192.168.2.0

 # Définition du réseau du client. Pour éviter tout conflit de réseaux, il est préférable de choisir des réseaux differents.

- [Conf-main-mode]

DOI= IPSEC

EXCHANGE_TYPE= ID_PROT

Transforms= 3DES-SHA-RSA_SIG

- [Conf-quick-mode]

DOI= IPSEC

EXCHANGE_TYPE= QUICK_MODE

Suites= QM-ESP-3DES-SHA-PFS-SUITE

- [X509-certificates]

Accept-self-signed= no

 # Très important que isakmpd n'accepte pas les certificats auto-signés

/etc/isakmpd/isakmpd.policy

```
File Edit Options Buffers Tools Help
Authorizer:      "POLICY"
Licencees:       "CA"
#Licensees:      "DN:/C=FR/L=Paris/O=IPsec Labs/OU=Certif/CN=itinet.fr"
#Conditions:     app_domain == "IPsec policy"
#                && doi == "ipsec" && pfs == "yes"
#                && esp_present == "yes" && ah_present == "no"
#                &&(esp_enc_alg == "3des" || esp_enc_alg == "aes")
#                && esp_auth_alg == "hmac-sha"
#                && (remote_id == "bloodylan.hd.free.fr"  ||
#                remote_id == "spykoland.dyndns.org"  ||
#                remote_id == "gw.whatsite.se")
                 -> "true" ;
-----:---F1  isakmpd.policy      (Fundamental)--L1--All-------------------------
For information about the GNU Project and its goals, type C-h C-p.
```

Ce fichier est utilisé par isakmpd pour définir des règles plus précises quant à l'utilisation des certificats.

/etc/ssl/x509v3.conf (*et génération des certificats*)

```
File Edit Options Buffers Tools Help
# default settings
CERTPATHLEN             = 1
CERTUSAGE               = digitalSignature,keyCertSign
CERTIP                  = 82.230.89.213
CERTFQDN                = itinet.fr_

# This section should be referenced when building an x509v3 CA
# Certificate.
# The default path length and the key usage can be overriden
# modified by setting the CERTPATHLEN and CERTUSAGE environment
# variables.
[x509v3_CA]
basicConstraints=critical,CA:true,pathlen:$ENV::CERTPATHLEN
keyUsage=$ENV::CERTUSAGE

# This section should be referenced to add an IP Address
# as an alternate subject name, needed by isakmpd
# The address must be provided in the CERTIP environment variable
[x509v3_IPAddr]
subjectAltName=IP:$ENV::CERTIP

# This section should be referenced to add a FQDN hostname
----:**-F1  x509v3.cnf         (Fundamental)--L5--Top----------------------------
Auto-saving...done
```

Avant de générer les certificates sur le serveur ou sur un client, pour que les certificats soient générés avec les bonnes options et qu'ils soient valides, il faut configurer les variables CERTIP et/ou CERTFQDN du fichier x509v3.conf.

Dans notre cas nous utiliserons l'ip fixe de la connexion, et le nom de domaine.

Une fois ceci fait, la démarche à suivre pour générer les certificats est la suivante :

```
# openssl req -x509 -days 365 -newkey rsa:1024 \

                -keyout /etc/ssl/private/ca.key \

                -out /etc/ssl/ca.crt
```

```
Organizational Unit Name (eg, section) []:
Common Name (eg, fully qualified host name) []:^C
sl/private/ca.key -out /etc/ssl/ca.crt                                    <
Generating a 1024 bit RSA private key
...++++++
.......++++++
writing new private key to '/etc/ssl/private/ca.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:FR
State or Province Name (full name) []:
Locality Name (eg, city) []:Paris
Organization Name (eg, company) []:IPsec Labs
Organizational Unit Name (eg, section) []:Certif
Common Name (eg, fully qualified host name) []:itinet.fr
Email Address []:
# _
```

openssl req va demander des informations qui seront incorporées dans la demande de certificate.

Dans les informations entrées il faudra définir un « Nom Distinct » (DN).

```
# openssl req -new -key /etc/isakmpd/private/local.key \

        -out /etc/isakmpd/private/itinet.fr.csr
```

(dans notre cas, sinon 10.0.0.1.csr)

```
te/local.key -out /etc/isakmpd/private/10.0.0.1.csr                    <
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:FR
State or Province Name (full name) []:
Locality Name (eg, city) []:Paris
Organization Name (eg, company) []:IPsec Labs
Organizational Unit Name (eg, section) []:Certif
Common Name (eg, fully qualified host name) []:itinet.fr
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:password
An optional company name []:
# _
```

Remplacer Itinet.fr.csr ou 10.0.0.1.csr par l'adresse qui sera utilisée par isakmpd comme idendité de certificat.

```
# openssl x509 -req -days 365 -in /etc/isakmpd/private/10.0.0.1.csr \

        -CA /etc/ssl/ca.crt -CAkey /etc/ssl/private/ca.key \

        -CAcreateserial -extfile /etc/ssl/x509v3.cnf \

        -extensions x509v3_IPAddr \

                -out /etc/isakmpd/certs/10.0.0.1.crt
```

For a FQDN certificate, do:

```
# openssl x509 -req -days 365 -in /etc/isakmpd/private/itinet.fr.csr \

        -CA /etc/ssl/ca.crt -CAkey /etc/ssl/private/ca.key \

        -CAcreateserial -extfile /etc/ssl/x509v3.cnf \
```

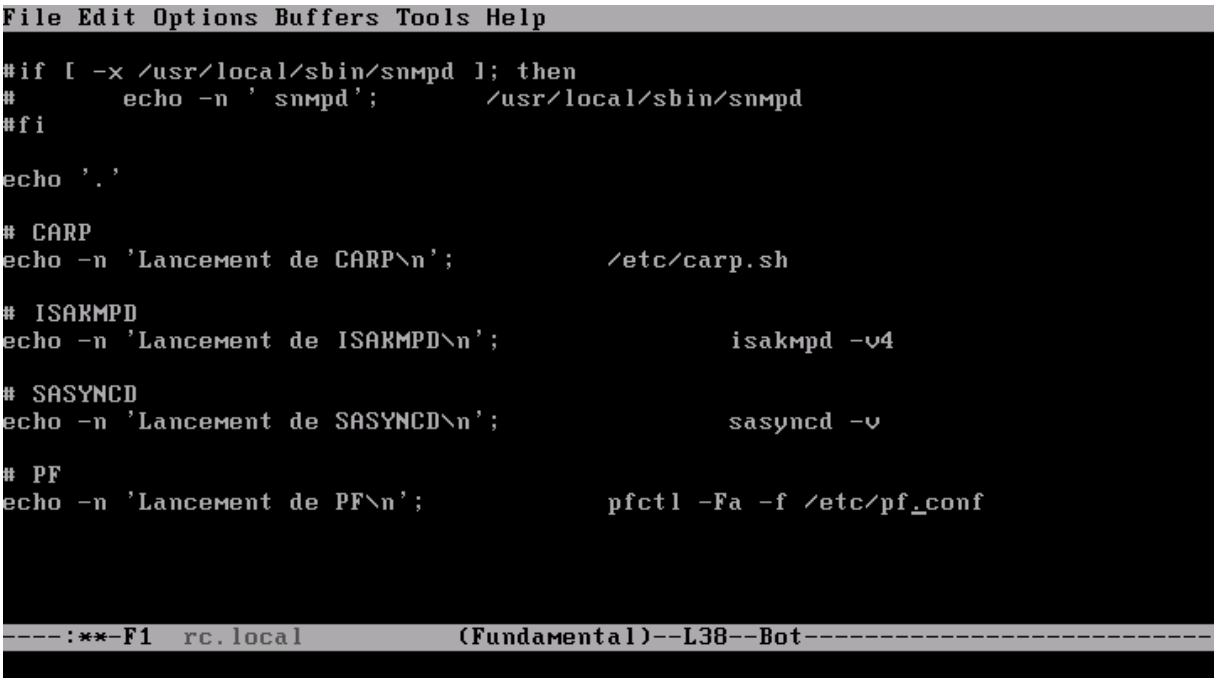
```
        -extensions x509v3_FQDN \

        -out /etc/isakmpd/certs/itinet.fr.crt
```

Ensuite, il ne reste plus qu'a copier les certificats (les .crt) dans /etc/isakmpd/certs/ (chose qui logiquement a été faite dans notre commande), et copier le CA présent de /etc/ssl/ca.crt vers /etc/isakmpd/ca/

Pour révoquer un certificat, il faut créer un fichier Liste de Révocation de Certificats (CRL) et de l'installer dans /etc/isakmpd/crls/

/etc/rc.local


File Edit Options Buffers Tools Help
#if [-x /usr/local/sbin/snmpd]; then
#	echo -n ' snmpd';	/usr/local/sbin/snmpd
#fi
echo '.'
CARP
echo -n 'Lancement de CARP\n';	/etc/carp.sh
ISAKMPD
echo -n 'Lancement de ISAKMPD\n';	isakmpd -v4
SASYNCD
echo -n 'Lancement de SASYNCD\n';	sasyncd -v
PF
echo -n 'Lancement de PF\n';	pfctl -Fa -f /etc/pf.conf
----:**-F1 rc.local	(Fundamental)--L38--Bot

BOB & BIB

/etc/rc.conf.local

```
pf=yes

routed_flags=NO
```

/etc/rc.local

```
# CARP

echo -n 'carp.sh';          /etc/carp.sh

# ISAKMPD

echo -n 'isakmpd';          isakmpd -v4

# SASYNCD

echo -n 'sasyncd';          sasyncd -v

# PF

echo -n 'pf.conf loaded';   pfctl -Fa -f /etc/pf.conf
```

/etc/sysctl.conf

```
net.inet.ip.forwarding=1    # 1=Permit forwarding (routing) of
packets

net.inet6.ip6.forwarding=1 # 1=Permit forwarding (routing) of
packets

net.inet.esp.enable=1       # 0=Disable the ESP IPsec protocol
```

```
net.inet.ah.enable=1        # 0=Disable the AH IPsec protocol

net.inet.esp.udpencap=1     # 0=Disable ESP-in-UDP encapsulation

net.inet.carp.preempt=1
```

/etc/carp.sh

```
#!/bin/sh

pf=YES



ifconfig xl0 mtu 1400

ifconfig rl0 10.0.0.1 netmask 255.255.255.0 mtu 1400

ifconfig xl1 172.16.0.1 netmask 255.255.255.0



ifconfig pfsync0 syncdev xl1

ifconfig syncpeer 172.16.0.2 syncdev enc1

ifconfig pfsync0 up



ifconfig carp1 create

ifconfig carp1 vhid 1 carpdev rl0 pass itivpn 10.0.0.10 255.255.255.0
description internal



ifconfig carp2 create

ifconfig carp2 vhid 2 carpdev xl0 pass itivpn 10.4.98.98
255.255.240.0 description external
```

/etc/pf.conf

```
      ## vpn rules
GATEWAY_INTECH = "82.230.89.213"
GATEWAY_SPYKE = "82.123.154.114"
GATEWAY_ESKI = "81.57.45.105"
GATEWAY_SPAWN = "82.230.55.32"


NETWORK_INTECH = "10.0.0.0/24"
NETWORK_SPYKE = "192.168.1.0/24"
NETWORK_ESKI = "192.168.0.0/24"
NETWORK_SPAWN = "192.168.2.0/24"
ext_if="xl0"
mid_if="xl1"
int_if="rl0"
carp1_if="carp1"
carp2_if="carp2"


      ## SCRUB
scrub fragment reassemble no-df
scrub on enc0 max-mss 1200


      ## nat du local vers le net
nat on $ext_if from $int_if:network to any -> ($carp2_if)
```

```
      ## Default deny
block log on { $ext_if $int_if } all
pass in tag OK keep state
pass out tagged OK

      ## ICMP
pass quick on { $ext_if $int_if } proto icmp

      ## pfsync
pass quick on { lo pfsync0 } all
pass quick on { $mid_if } proto pfsync

      ## carp
pass quick on { $ext_if $int_if } proto carp keep state

      ## antispoof
antispoof quick for { $ext_if, $int_if } inet

      ## Default lan accept
pass in on $int_if from $NETWORK_INTECH to any
pass out on $int_if from any to $NETWORK_INTECH

      ## pass encrypted traffic from security gateways
pass in on $ext_if proto esp from $GATEWAY_SPYKE to $GATEWAY_INTECH
```

```
pass in on $ext_if proto esp from $GATEWAY_ESKI to $GATEWAY_INTECH

pass in on $ext_if proto esp from $GATEWAY_SPAWN to $GATEWAY_INTECH



     ## ipencap

pass in on enc0 proto ipencap from $GATEWAY_SPAWN to $GATEWAY_INTECH

pass in on enc0 proto ipencap from $GATEWAY_SPYKE to $GATEWAY_INTECH

pass in on enc0 proto ipencap from $GATEWAY_ESKI to $GATEWAY_INTECH



     ## isakmpd

pass in on $ext_if proto udp from $GATEWAY_SPYKE to $GATEWAY_INTECH
port = 500 keep state

pass in on $ext_if proto udp from $GATEWAY_SPYKE to $GATEWAY_INTECH
port = 4500 keep state

pass in on $ext_if proto udp from $GATEWAY_SPAWN to $GATEWAY_INTECH
port = 500 keep state

pass in on $ext_if proto udp from $GATEWAY_SPAWN to $GATEWAY_INTECH
port = 4500 keep state

pass in on $ext_if proto udp from $GATEWAY_ESKI to $GATEWAY_INTECH
port = 500 keep state

pass in on $ext_if proto udp from $GATEWAY_ESKI to $GATEWAY_INTECH
port = 4500 keep state
```

/etc/sasyncd.conf

```
carp interface carp1 interval 1

sharedkey fb76feb53a48d8dbce6a48ad836e748e

flushmode startup

listen on xl1 inet

mode master

peer 10.0.0.2


carp interface carp2 interval 1

sharedkey fb76feb53a48d8dbce6a48ad836e748e

flushmode startup

listen on rl0 inet

mode master

peer 10.4.101.166
```

/etc/isakmpd/isakmpd.conf

```
[General]

Listen-on=                      10.4.98.98

Retransmits=                    4

Exchange-max-time=              3600

Check-interval=                 300

Policy-file=                    /etc/isakmpd/isakmpd.policy

Shared-SADB=                    Defined

Use-Keynote=                    No


[Phase 1]

Default=

82.230.55.32=                   spawn-p1

81.57.45.105=                   eski-p1

82.123.13.236=                  spyke-p1


[Phase 2]

Passive-connections=            spawn-p2, eski-p2, spyke-p2


[my-fqdn]

ID-type=                        FQDN

Name=                           itinet.fr
```

```
[my-ipv4-net]

ID-type=                        IPV4_ADDR_SUBNET

Network=                        10.0.0.0

Netmask=                        255.255.255.0


##############   SPAWN   ####################

################################################


[spawn-p1]

Phase=                          1

Transport=                      udp

Configuration=                  Conf-main-mode

ID=                             my-fqdn

Remote-ID=                      spawn-fqdn


[spawn-fqdn]

ID-type=                        FQDN

Name=                           bloodylan.hd.free.fr


[spawn-p2]

Phase=                          2

ISAKMP-peer=                    spawn-p1

Configuration=                  Conf-quick-mode

Local-ID=                       my-ipv4-net

Remote-ID=                      spawn-ipv4-net
```

```
[spawn-ipv4-net]
ID-type=                        IPV4_ADDR_SUBNET
Network=                        192.168.2.0
Netmask=                        255.255.255.0

##############   ESKI   ######################
################################################

[eski-p1]
Phase=                          1
Transport=                      udp
Configuration=                  Conf-main-mode
ID=                             my-fqdn
Remote-ID=                  eski-ip

[eski-ip]
ID-type=                        FQDN
#ID-type=                   IPV4_ADDR
Name=                           cabinet-crearte.com
#Address=                   81.57.45.105

[eski-p2]
Phase=                          2
```

```
ISAKMP-peer=                    eski-p1

Configuration=                  Conf-quick-mode

Local-ID=                       my-ipv4-net

Remote-ID=                      eski-ipv4-net


[eski-ipv4-net]

ID-type=                        IPV4_ADDR_SUBNET

Network=                        192.168.0.0

Netmask=                        255.255.255.0


##############   SPYKE   ######################

################################################


[spyke-p1]

Phase=                          1

Transport=                      udp

Configuration=                  Conf-main-mode

ID=                             my-fqdn

Remote-ID=                      spyke-fqdn

#Authentication=                 Va9evivu


[spyke-fqdn]

ID-type=                        FQDN

Name=                           spykoland.dyndns.org
```

```
[spyke-p2]

Phase=                          2

ISAKMP-peer=                    spyke-p1

Configuration=                  Conf-quick-mode

Local-ID=                       my-ipv4-net

Remote-ID=                      spyke-ipv4-net


[spyke-ipv4-net]

ID-type=                        IPV4_ADDR_SUBNET

Network=                        192.168.1.0

Netmask=                        255.255.255.0



############   TRANSFORM   ###################

##############################################


[Conf-main-mode]

DOI=                            IPSEC

EXCHANGE_TYPE=                  ID_PROT

Transforms=                     3DES-SHA-RSA_SIG


[Conf-quick-mode]

DOI=                            IPSEC

EXCHANGE_TYPE=                  QUICK_MODE

Suites=                         QM-ESP-3DES-SHA-PFS-SUITE
```

```
[X509-certificates]
CA-directory=                   /etc/isakmpd/ca/
Cert-directory=                 /etc/isakmpd/certs/
Private-key=                    /etc/isakmpd/private/local.key
Accept-self-signed=             no

[3DES-SHA]
ENCRIPTION_ALGORITHM=           3DES_CBC
HASH_ALGORITHM=                 SHA
AUTHENTICATION_METHOD=          MODP_1024
Life=                           LIFE_60_SECS,LIFE_1000_KB

[QM-ESP-3DES-SHA-PFS-SUITE]
Protocols=                      QM-ESP-3DES-SHA-PFS

[QM-ESP-3DES-SHA-PFS]
PROTOCOL_ID=                    IPSEC_ESP
Transforms=                     QM-ESP-3DES-SHA-PFS-XF
```

```
[QM-ESP-3DES-SHA-PFS-XF]
TRANSFORM_ID=                   3DES
ENCAPSULATION_MODE=             TUNNEL
AUTHENTICATION_ALGORITHM=       SHA
GROUP_DESCRIPTION=              MODP_1024
```

```
Life=                                   LIFE_60_SECS



[LIFE_60_SECS]

LIFE_TYPE=                              SECONDS

LIFE_DURATION=                          60,45:72



[LIFE_1000_KB]

LIFE_TYPE=                              KILOBYTES

LIFE_DURATION=                          1000,768:1536
```

/etc/isakmpd/isakmpd.policy

```
Authorizer:     "POLICY"

Licencees: "CA"

#Licensees:     "DN:/C=FR/L=Paris/O=IPsec
Labs/OU=Certif/CN=itinet.fr"

#Conditions:    app_domain == "IPsec policy"

#          &&doi == "ipsec" && pfs == "yes"

#          &&esp_present == "yes" && ah_present == "no"

#          &&(esp_enc_alg == "3des" || esp_enc_alg == "aes")

#          &&esp_auth_alg == "hmac-sha"

#          &&(remote_id == "bloodylan.hd.free.fr"  ||

#          remote_id == "spykoland.dyndns.org"  ||

#          remote_id == "gw.whatsite.se")

           ->"true" ;
```

/etc/ssl/x509v3.cnf

```
# default settings


CERTPATHLEN                = 1

CERTUSAGE             = digitalSignature,keyCertSign

CERTIP                     = 82.230.89.213

CERTFQDN              = itinet.fr


# This section should be referenced when building an x509v3 CA

# Certificate.

# The default path length and the key usage can be overriden

# modified by setting the CERTPATHLEN and CERTUSAGE environment

# variables.


[x509v3_CA]

basicConstraints=critical,CA:true,pathlen:$ENV::CERTPATHLEN

keyUsage=$ENV::CERTUSAGE


# This section should be referenced to add an IP Address

# as an alternate subject name, needed by isakmpd

# The address must be provided in the CERTIP environment
variable
```

```
[x509v3_IPAddr]

subjectAltName=IP:$ENV::CERTIP



# This section should be referenced to add a FQDN hostname

# as an alternate subject name, needed by isakmpd

# The address must be provided in the CERTFQDN environment
variable



[x509v3_FQDN]

subjectAltName=DNS:$ENV::CERTFQDN
```

/etc/pf.conf (fichier de configuration)

```
#	$OpenBSD: pf.conf,v 1.29 2005/08/23 02:52:58 henning Exp $
#
# See pf.conf(5) and /usr/share/pf for syntax and examples.
# Remember to set net.inet.ip.forwarding=1 and/or net.inet6.ip6.forwarding=1
# in /etc/sysctl.conf if packets are to be forwarded between interfaces.


## vpn rules
GATEWAY_INTECH = "82.230.89.213"
GATEWAY_SPYKE = "82.123.154.114"
GATEWAY_ESKI = "81.57.45.105"
GATEWAY_SPAWN = "82.230.55.32"

NETWORK_INTECH = "10.0.0.0/24"
NETWORK_SPYKE = "192.168.1.0/24"
NETWORK_ESKI = "192.168.0.0/24"
NETWORK_SPAWN = "192.168.2.0/24"


ext_if="xl0"
mid_if="xl1"
int_if="rl0"
carp1_if="carp1"
carp2_if="carp2"

## SCRUB
scrub in log all fragment reassemble no-df max-mss 1460

## nat du local vers le net
nat on $ext_if from $int_if:network to any -> $carp2_if

## Default deny
block log on { $ext_if $int_if } all
pass in tag OK keep state
pass out tagged OK

## ICMP
pass quick on { $ext_if $int_if } proto icmp

## pfsync
```

```
pass quick on { lo pfsync0 } all
pass quick on { $mid_if } proto pfsync

## carp
pass quick on { $ext_if $int_if } proto carp keep state

## antispoof
antispoof quick for $int_if inet

## Default lan  accept
pass in on $int_if from $NETWORK_INTECH to any
pass out on $int_if from any to $NETWORK_INTECH

## pass encrypted traffic from security gateways
pass in on $ext_if proto esp from $GATEWAY_SPYKE to $GATEWAY_INTECH
pass in on $ext_if proto esp from $GATEWAY_ESKI to $GATEWAY_INTECH
pass in on $ext_if proto esp from $GATEWAY_SPAWN to $GATEWAY_INTECH

## ipencap
pass in on enc0 proto ipencap from $GATEWAY_SPAWN to $GATEWAY_INTECH
pass in on enc0 proto ipencap from $GATEWAY_SPYKE to $GATEWAY_INTECH
pass in on enc0 proto ipencap from $GATEWAY_ESKI to $GATEWAY_INTECH

## isakmpd
pass in on $ext_if proto udp from $GATEWAY_SPYKE to $GATEWAY_INTECH port = 500 keep state
pass in on $ext_if proto udp from $GATEWAY_SPYKE to $GATEWAY_INTECH port = 4500 keep state
pass in on $ext_if proto udp from $GATEWAY_SPAWN to $GATEWAY_INTECH port = 500 keep state
pass in on $ext_if proto udp from $GATEWAY_SPAWN to $GATEWAY_INTECH port = 4500 keep state
pass in on $ext_if proto udp from $GATEWAY_ESKI to $GATEWAY_INTECH port = 500 keep state
pass in on $ext_if proto udp from $GATEWAY_ESKI to $GATEWAY_INTECH port = 4500 keep state
```

Documents de référence

Manpage de carp
(http://www.openbsd.org/cgi-bin/man.cgi?query=carp&apropos=0&sektion=0&manpath=OpenBSD+3.8&arch=i386&format=html)

Manpage de sasyncd
(http://www.openbsd.org/cgi-bin/man.cgi?query=sasyncd&apropos=0&sektion=0&manpath=OpenBSD+3.8&arch=i386&format=html)

Manpage de pfsync
(http://www.openbsd.org/cgi-bin/man.cgi?query=pfsync&apropos=0&sektion=0&manpath=OpenBSD+3.8&arch=i386&format=html)

Manpage de pf
(http://www.openbsd.org/cgi-bin/man.cgi?query=pf&apropos=0&sektion=0&manpath=OpenBSD+3.8&arch=i386&format=html)

Manpage de vpn
(http://www.openbsd.org/cgi-bin/man.cgi?query=vpn&apropos=0&sektion=8&manpath=OpenBSD+3.8&arch=i386&format=html)

Manpage de isakmpd.policy
(http://www.openbsd.org/cgi-bin/man.cgi?query=isakmpd.policy&apropos=0&sektion=5&manpath=OpenBSD+3.8&arch=i386&format=html)

Manpage de isakmpd.conf
(http://www.openbsd.org/cgi-bin/man.cgi?query=isakmpd.conf&apropos=0&sektion=5&manpath=OpenBSD+3.8&arch=i386&format=html)

Manpage de isakmpd
(http://www.openbsd.org/cgi-bin/man.cgi?query=isakmpd&apropos=0&sektion=8&manpath=OpenBSD+3.8&arch=i386&format=html)

Manpage de ipsec
(http://www.openbsd.org/cgi-bin/man.cgi?query=ipsec&apropos=0&sektion=0&manpath=OpenBSD+3.8&arch=i386&format=html)

http://hem.passagen.se/hojg/isakmpd/

http://pintday.org/hack/crypto/ca.shtml

http://openbsd.somedomain.net/isakmpd_with_certs/

Utiliser Le serveur OpenBSD avec un Linux 2.6
(http://www.ipsec-howto.org/x496.html)

Remerciements

Merci à tous ceux qui nous ont aidés pendant la durée du projet

Aux personnes qui ont répondu sur linuxfr et qui nous ont apporté rectifications, précisions ou informations.

Remarques à prendre en compte

Page 14, creation et paramétrage des interfaces: la méthode préconisée et maintenable sous OpenBSD est d'utiliser /etc/hostname.ifname (cf la page de man hostname.if), y compris pour les interfaces CARP et pfsync. C'est d'ailleur ainsi que le documentent les auteurs de pf et carp, sur http://www.countersiege.com/doc/pfsync-carp/ et dans la page de man de pfsync(4).
Par exemple dans ton cas, ca donnerai un fichier /etc/hostname.carp1 contenant quelque chose comme:
inet 10.0.0.10 netmask 0xffffff00 vhid 1 carpdev rl0 pass itivpn description iternal
etc. pour les autres interfaces (hostname.carp2, hostname.pfsync0, hostname.xl0, hostname.rl0, ...).

Page 26: la méthode recommandée (et moins bidouille) pour lancer isakmpd est de placer : isakmpd_flags=-v4 dans /etc/rc.conf.local (pas d'utiliser un shell script). La bonne préconisée (openbsdiste) de lancer d'initialiser les interfaces, (dont psync0 et carp), c'est de créer des fichiers hostname.if (comme dit plus haut). Et la bonne façon de lancer pf, c'est de mettre pf=YES dans /etc/rc.conf.local (dans ce cas il sera automatiquement initialisé, pas la peine d'en remettre une couche dans rc.local).*
Bref, l'ensemble des manips dans /etc/rc.local ne sont pas très openbsdistes, et on pourrait s'en passer.
Page 29: à quoi sert ce fichier /etc/pf.vpn.conf (il n'est jamais appellé depuis tes scripts) ? pourquoi ne pas placer toute ta conf dans /etc/pf.conf ? D'ailleur ton fichier pf.conf n'est pas indiqué dans le document (page 17: le fichier pf.conf n'est pas présent en fin de document). Au passage, on pourrai grandement simplifier ce ruleset, en factorisant les regles concernant GATEWAY et NETWORK*.*

Juste un petit complément d'infos: depuis une ou deux versions d'OpenBSD, la nouvelle façon en vogue (en fait, encouragée par les mainteneurs) de construire un vpn simple comme le tien, c'est d'utiliser ipsecctl(8) et ipsec.conf(5) et de le laisser configurer/piloter isakmpd.

Au résultat, on obtient un fichier de conf /etc/ipsec.conf d'une dizaine de lignes au lieu d'un isakmpd.conf + keynote + ... de centaines de lignes.
On peut aussi laisser le système générer les clefs RSA au premier démarage (à la manière d'OpenSSH, il le fait automatiquement, tout seul, si aucune clef n'est trouvée).

Ça pourrait être doublement interessant dans le cas de ta démonstration, car tu pourrai ainsi gagner en clarté, en réduisant la configuration à l'essentiel de ton sujet (sasyncd, la redondance, la haute disponibilité, ...), tout en réduisant la partie "paramétrage et mise en place des tunnels ESP et serveurs de clefs IKE", (moins interessante à mon avis: on trouve des centaines de docs sur ça, et c'est une partie que chacun devra adapter à son environement réseau), à quelques lignes de configuration très claires et simples. Bref, aller à l'essentiel de ton sujet: la haute disponibilité.

Et faire confiance au mainteneur d'ipsecctl (qui est aussi le mainteneur d'isakmpd) pour faire un bon paramétrage automatique d'isakmpd n'est pas forcément une mauvaise chose.

Autre détail: mieux vaut utiliser "current" (la version HEAD du cvs d'OpenBSD, ou mieux, un snapshot) car des problèmes très importants ont étés corrigés dans sasyncd(8) depuis la release 3.8. Actuellement "current" est très proche du feature freeze, et au grand public pour tests est en cours depuis presque un mois, c'est donc stable.

Dernier point: la partie sur les spécifications matérielles sont assez vagues (tu parle d'une machine d'"au moins 1Ghz"). Tu ne dit pas non plus quel volume de traffic tu doit gérer, mais attention, IPsec est gourmand. Si tu doit travailler à la vitesse du fast ethernet (ou pire, du gigabit), ça va être très difficile avec un pentium 1Ghz ! la commande "openssl speed" te montrera la vitesse de (de)chiffrement atteignable, selon l'algo, sur ta machine.

Le man ipsec.conf(5): http://www.openbsd.org/cgi-bin/man.cgi?query=ipsec.conf
Les snapshots de current: ftp://ftp.openbsd.org/pub/OpenBSD/snapshots