

Sécurité GNU/Linux

By ShareVB

Layer 2 Tunneling Protocol

Table des matières

I.Introduction.....	2
II.Fonctionnement et bases.....	2
III.Implémentation Linux : l2tpns (serveur) et rp-l2tp (client).....	3
a)Introduction.....	3
b)Prérequis noyau Linux.....	4
c)Configuration d'IPSec.....	4
1.Installation.....	4
2.Une configuration avec clé prépartagée.....	4
i.Configuration du serveur.....	4
3.Autres configurations.....	5
d)Configuration de MySQL.....	5
e)Configuration du serveur FreeRADIUS.....	6
f)Remplissage de la table MySQL.....	7
1.Les tables de la base radius.....	7
2.Ajout d'un utilisateur.....	7
3.Ajout d'un utilisateur à un groupe.....	8
4.Les attributs des utilisateurs et groupes.....	8
5.Ajouter des attributs à un utilisateur ou groupe.....	9
6.Test de l'authentification.....	9
g)Serveur VPN l2tpns.....	10
1.Installation.....	10
2.Configuration.....	10
h)Passerelle.....	12
IV.Configuration du client Windows.....	12
a)Prérequis IPSec.....	12
b)Configuration.....	12
V.Configuration du client Linux (itinérant).....	13
a)Configuration d'IPSec sur le client.....	13
b)Configuration d'IPSec sur le serveur.....	15
c)Configuration de rp-l2tp.....	16
VI.Et iptables dans tout ça.....	19
a)La base.....	19
b)Pour IPSec.....	19
c)Pour L2TP.....	20

d)Pour FreeRADIUS.....	21
e)Pour MySQL.....	21
VII.Bibliographie.....	22

I. Introduction

Voici le second protocole VPN que Windows sait utiliser nativement. Il se base sur le protocole L2F de Cisco pour faire un tunnel non crypté et nécessite donc IPSec pour sécuriser le tout. A noter que c'est un protocole relativement lourd.

II. Fonctionnement et bases

C'est un protocole très proche des protocoles PPTP et L2F et est normalisé dans un RFC. Cette fois les trames PPP sont encapsulées dans le protocole L2TP lui-même et les trames PPP peuvent encapsuler des paquets IP, IPX, NetBIOS ou autre. Il se base aussi souvent sur IPSec.

Il y faut deux types de serveurs pour utiliser L2TP :

- LAC (L2TP Access Concentrator) : concentrateur d'accès L2TP. Il sert à fournir un moyen physique pour se connecter à un ou plusieurs LNS par le protocole L2TP. Il est responsable de l'identification et construit le tunnel vers les LNS. Il se trouve obligatoirement dans l'infrastructure du FAI de chaque utilisateur du VPN. Cela est donc très lourd (et cher) à mettre en place dans la mesure où il faut louer une place dans un serveur de connexion du FAI.
- LNS (L2TP Network Server) : serveur réseau L2TP, il assure la communication entre le réseau auquel il est connecté et les LAC vers lesquels il a un tunnel. Il se trouve généralement dans l'entreprise ou le service auquel appartient l'utilisateur distant.

Plus techniquement, voici l'encapsulation qu'engendre L2TP (de bas en haut, dans le cas d'un HTTP) :

- couche 2 -> IP -> UDP -> L2TP -> PPP -> IP -> TCP -> HTTP

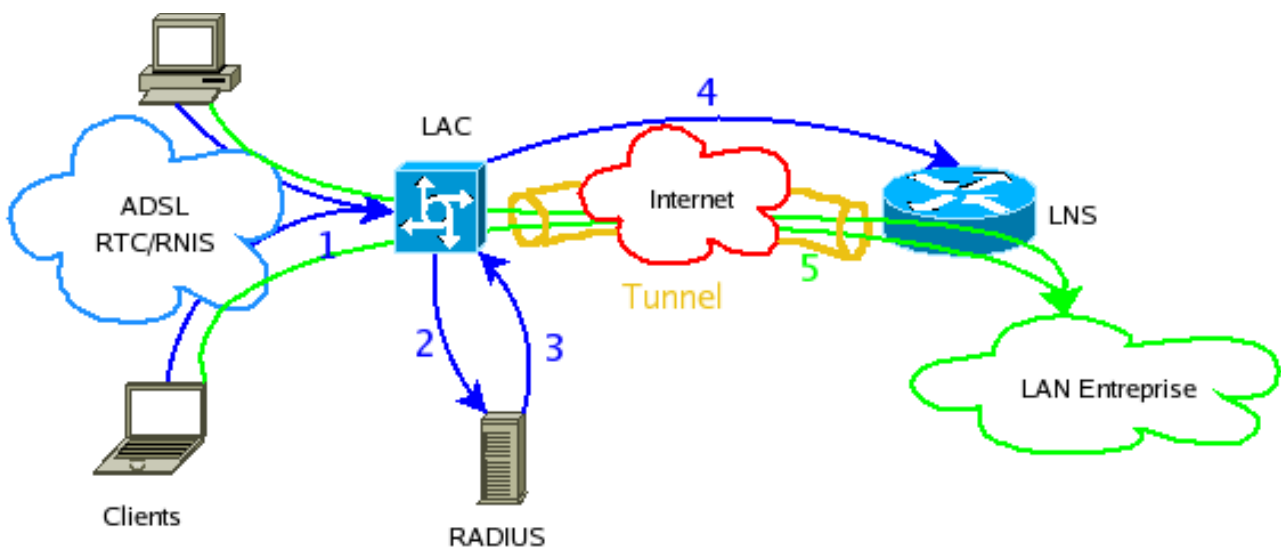
Il est donc relativement lourd, d'autant que les MTU (taille max des paquets) des lignes traversées peuvent générer de la fragmentation. Son seul avantage est de pouvoir terminer une session PPP n'importe quand ce qui permet à un utilisateur mobile de pouvoir se connecter facilement en VPN.

L2TP est encapsulé dans des paquets UDP entre le LAC et le LNS et utilise le port 1701.

La connexion d'un utilisateur se passe donc comme suit :

- 1) Un utilisateur se connecte à un LAC par le biais de sa connexion Internet ou d'un Modem bas débit. Ce LAC fait partie de l'infrastructure du FAI de l'utilisateur.

- 2) Il s'authentifie auprès de ce LAC. Ce dernier transmet les informations de login/mot de passe fournies au serveur RADIUS d'authentification. Ce dernier contient une liste associative login/nom_de_domaine/mot_passe <--> LNS.
- 3) Si le login/mot de passe est valide, cela permet au LAC de connaître le LNS auquel l'utilisateur peut se connecter pour être sur le VPN de son entreprise.
- 4) Si aucun tunnel n'existe entre le LAC et le LNS, un tunnel est créé à l'initiative du LAC
- 5) Une session PPP est créée à l'intérieur de ce tunnel
- L'utilisateur obtient donc une connexion PPP virtuelle entre lui et le réseau de son entreprise.



III. Implémentation Linux : l2tpns (serveur) et rp-l2tp (client)

a) Introduction

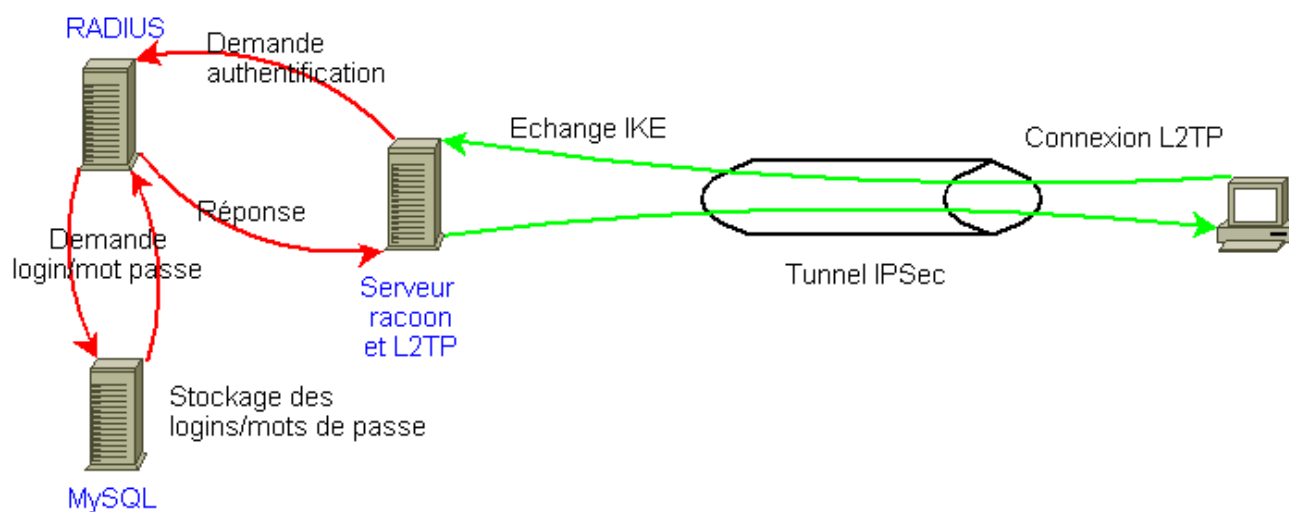
Comme vous l'avez constaté, dans la théorie, il faut deux serveurs (LNS et LAC) et au moins un client pour se connecter au LAC et un serveur dans le réseau interne que l'on souhaite atteindre (pas nécessaire, mais un VPN sans services dans le réseau interne, ce n'est pas très utile).

Cependant dans la pratique, on peut résumer le schéma typique client->LAC->LNS->réseau interne à client->LNS->réseau interne :

Il est enfin nécessaire de noter que dans l'implémentation serveur L2TP linux nécessite les composants suivants afin d'être compatibles avec les clients Windows :

- les fonctionnalités IPsec activées dans le noyau
- un serveur IKE, pour l'échange des clés IPsec : racoon
- un serveur L2TP, pour la tunnelisation VPN : l2tpns

- un serveur RADIUS, pour l'authentification des utilisateurs du VPN : freeradius
- un serveur MySQL pour stocker les logins/mots de passe des utilisateurs VPN



b) Prérequis noyau Linux

Le système d'exploitation utilisé est Debian avec un noyau 2.6 gérant IPsec et le pack cryptographique (par défaut).

c) Configuration d'IPsec

1. Installation

```
[root]# apt-get install racoon ipsec-tools
```

2. Une configuration avec clé prépartagée

Ce n'est pas vraiment la meilleure solution dans la mesure, où les IP des clients doivent être connues ce qui peut ne pas être le cas pour une entreprise avec des clients itinérants se connectant par Internet. Il est préférable d'utiliser les certificats. Voir le tuto consacré à IPsec et la section « IPsec seul avec Windows » (sauf les stratégies de sécurité Windows).

i. Configuration du serveur

Dans le fichier `/etc/racoon/racoon.conf`, on inscrira la configuration de racoon IKE suivante :

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

# indique les conditions de la première phase d'IPsec
remote anonymous {
```

```

exchange_mode main;
#obéir aux exigences du client
proposal_check obey;
#faire une proposition quand même
proposal {
    encryption_algorithm 3des;
    hash_algorithm sha1;
    authentication_method pre_shared_key;
    dh_group modp1024;
}
#générer les règles de décision IPSec
generate_policy on;
#attendre les requêtes du client
passive on;
#autoriser IPSec par NAT
nat_traversal on;
}
# deuxième phase : échange des valeurs de cryptage du tunnel
sainfo anonymous
{
    pfs_group 2;
    lifetime time 12 hour ;
    encryption_algorithm 3des, blowfish 448, twofish, rijndael
;
    authentication_algorithm hmac_md5, hmac_sha1 ;
    compression_algorithm deflate;
}

```

Dans le fichier `/etc/racoon/psk.txt`, on inscrira la clé prépartagé associée à chaque machine (par son IP), par exemple :

```
192.168.0.1 cle_secrete
```

3. Autres configurations

Voir le tuto sur IPSec, section « IPSec seul avec Windows » (sauf les stratégies de sécurité Windows).

d) Configuration de MySQL

Pour plus d'informations sur MySQL voir le tuto sur LAMP (Linux Apache MySQL PHP).

Installation du serveur SQL :

```
[root]#apt-get install mysql-server
```

Définir un mot de passe (différent du root système) pour le root de MySQL :

```
[root]# /usr/bin/mysqladmin -u root password 'mysql_password'
```

Il peut être nécessaire de fixer le mot de passe root MySQL dans le dossier de root système pour laisser fonctionner les scripts cron :

```
[root]# touch /root/.my.cnf
[root]# chmod 600 /root/.my.cnf
```

Et dans `/root/.my.cnf`, ajouter :

```
[mysqladmin]
user      = root
password = mysql_password
```

Redémarrer MySQL:

```
[root]# /etc/init.d/mysql restart
```

e) Configuration du serveur FreeRADIUS

Installer freeradius et son plugin MySQL :

```
apt-get install freeradius freeradius-mysql
```

Création de la base radius :

```
[root]# gunzip /usr/share/doc/freeradius/examples/db_mysql.sql.gz
[root]# cp /usr/share/doc/freeradius/examples/db_mysql.sql
/tmp/db_mysql.sql
```

A cause d'un bug, il est nécessaire d'éditer le fichier `/tmp/db_mysql.sql` et de modifier à la fin du script la création de la table « *nas* » :

```
Supprimer:
    DEFAULT '0'
de:
    id int(10) DEFAULT '0' NOT NULL auto_increment
de sorte que cela ressemble à ceci:
    id int(10) NOT NULL auto_increment
```

Création de la table radius et remplissage :

```
[root]# mysqladmin create radius
[root]# cat /tmp/db_mysql.sql | mysql -p radius
```

Il faut ensuite créer un utilisateur radius et lui accorder tous les droits sur la base radius :

```
[root]# mysql -p
GRANT ALL ON radius.* TO 'radius'@'localhost' IDENTIFIED BY
'radius_password';
GRANT ALL ON radius.* TO 'radius'@'%' IDENTIFIED BY 'radius_password';
exit
```

Modifier le fichier `/etc/freeradius/sql.conf` et définir le serveur, le login et

le mot de passe MySQL :

```
server = "localhost"  
login = "radius"  
password = "radius_password"
```

Modifier le fichier `/etc/freeradius/clients.conf` pour définir un nouveau secret (authentification des requêtes radius) pour le serveur radius. La section client 127.0.0.1 devrait ressembler à ceci :

```
client 127.0.0.1 {  
    secret          = radius_secret  
    shortname       = localhost  
    nasstype        = other  
}
```

Dans le fichier `/etc/freeradius/radiusd.conf` :

- Décommenter « sql » dans les sections (vers la fin du fichier) :
 - `authorize{}`
 - `accounting{}`
 - `session{}`
- Commenter « radutmp » dans les sections (module inutile si l'on utilise MySQL) :
 - `accounting{}`
 - `session{}`
- Commenter absolument « files » dans la section `authorize{}` (sinon PAP ne marche pas)
- Commenter « unix » et « pam » dans `authenticate{}` (modules inutiles dans notre cas)

« sql » ne peut pas se trouver dans `authenticate{}` car ce n'est pas un serveur d'authentification mais un source d'autorisation.

Redémarrer freeradius :

```
[root]# /etc/init.d/freeradius restart
```

f) Remplissage de la table MySQL

1. Les tables de la base radius

La base radius contient les tables suivantes :

- La table « radcheck » contient les logins/mots de passe de tous les utilisateurs

autorisés.

- Une entrée doit être présente pour chaque utilisateur autorisé sur le VPN
- La table « radreply » table contient des attributs par utilisateur pour les réponses aux requêtes.
- La table « usergroup » contient les utilisateurs et leur appartenance à un groupe. Les groupes sont utilisés pour renvoyer des attributs pour un groupe d'utilisateurs.
- La table « radgroupreply » contient des attributs pour les groupes.

2. Ajout d'un utilisateur

Ensuite, il faut remplir la base SQL avec les utilisateurs autorisés. Le strict minimum est :

```
INSERT INTO radcheck (UserName, Attribute, op, Value) VALUES  
( 'nom_utilisateur', 'User-Password', '=', 'mot_de_passe');
```

Il n'est pas nécessaire de mettre Auth-Type := Local dans la table *radgroupcheck* comme beaucoup le dise car freeradius doit savoir se débrouiller pour trouver le type d'authentification avec la requête. De plus, cela ne marche pas avec mschap-v2.

3. Ajout d'un utilisateur à un groupe

Pour ajouter un utilisateur à un groupe :

```
INSERT INTO usergroup  
      (UserName, GroupName)  
  
VALUES ('nom_utilisateur', 'nom_groupe');
```

4. Les attributs des utilisateurs et groupes

Les attributs possibles sont les suivants (suivant la RFC) :

Attribut	Description
User-Name	Indique le nom de l'utilisateur qui vient d'être authentifié
NAS-IP Address	Adresse IP du NAS (Network Access Server) qui est à l'origine de l'authentification
Service-Type	Indique le type de service à fournir : •Login : l'utilisateur doit être connecté à une machine

	<ul style="list-style-type: none"> •Framed : l'utilisateur doit utiliser un protocole type PPP ou SLIP •Callback-Login :l'utilisateur doit se reconnecter à la machine •Callback-Framed : l'utilisateur doit se reconnecter à un protocole type PPP ou SLIP •Outbound : utilisateur autorisé sur les périphériques de sortie •Administrative : autorise l'utilisateur à se connecter en mode privilégié sur le NAS •NAS-Prompt : autorise l'utilisateur à se connecter en mode non privilégié sur le NAS •Authenticate Only : authentification sans attributs retournés •Callback-NAS-Prompt : autorise l'utilisateur à se reconnecter en mode non privilégié sur le NAS
Framed-Protocol	Indique le protocole à utiliser : PPP ou SLIP
Framed-IP-Address	Indique l'adresse IP à donner à l'utilisateur
Framed-IP-Netmask	Indique le masque de sous réseau à donner pour l'IP de l'utilisateur
Framed-Routing	<p>Indique le type de routage si l'utilisateur est un routeur :</p> <ul style="list-style-type: none"> •None •Send routing packets •Listen for routing packets •Send routing packets and listen for routing packets
Framed-MTU	Indique la taille maximum des paquets sur la connexion de l'utilisateur
Framed-Route	Indique une information de routage au format : « <i>adresse_reseau/nb_bits routeur</i> »

5. Ajouter des attributs à un utilisateur ou groupe

Pour ajouter des attributs à un utilisateur ou un groupe, il faut utiliser la syntaxe suivantes dans la base radius :

```
# pour un utilisateur
INSERT INTO radreply
    (UserName, Attribute, op, Value)
VALUES ('nom_utilisateur', 'nom_attribut', ':=', 'valeur');

# pour un groupe
INSERT INTO radgroupreply
    (GroupName, Attribute, op, Value)
VALUES ('nom_groupe', 'nom_attribut', ':=', 'valeur');
```

6. Test de l'authentification

Vous pouvez tester la configuration avec la commande radtest :

```
[root]# radtest nom_utilisateur mot_de_passe localhost 1812
radius_secret
```

g) Serveur VPN l2tpns

1. Installation

D'abord, il faut télécharger l2tpns en tar.gz à l'adresse <http://l2tpns.sourceforge.net/> et libcli 1.8.5 ou plus à l'adresse <http://sourceforge.net/projects/libcli/>.

```
[root]# tar xzvf libcli-*.tar.gz
[root]# cd libcli-*
[root]# make
[root]# make install PREFIX=/usr
[root]# cd ..
[root]#
[root]# tar xzvf l2tpns-*
[root]# cd l2tpns-*
[root]# make
[root]# make install
[root]# cd ..
```

2. Configuration

Dans le fichier /etc/l2tpns/startup-config, on inscrit la configuration de l2tpns :

```
# Niveau de débogage dans les logs
# 3 = informations, 4 = trace du programme, 5 = paquets
```

```
set debug 3

# Fichier de log
set log_file "/var/log/l2tpns"

# Fichier de pid pour le démon principal l2tpns
set pid_file "/var/run/l2tpns.pid"

# Eventuel secret entre le LAC et le serveur LNS l2tpns
set l2tp_secret "secret_l2tp"

# MTU pour l'interface L2TP
#set l2tp_mtu 1500

# IP des serveurs DNS primaire et secondaire
set primary_dns IP_dns_primaire
set secondary_dns IP_dns_secondaire

# IP du ou des serveurs RADIUS chargés de l'authentification
# des clients du VPN
# il est nécessaire de bien préciser le port
# sinon ca ne marche pas
# car l2tpns n'utilise pas par défaut le nouveau port freeradius

#serveur RADIUS primaire obligatoire
set primary_radius 127.0.0.1
set primary_radius_port 1812

#serveur RADIUS secondaire facultatif
#set secondary_radius 0.0.0.0
#set secondary_radius_port 1812

#secret RADIUS nécessaire pour « authentifier » les requêtes
# auprès du serveur
set radius_secret "radius_secret"

# Type d'authentifications autorisés pour les clients
# par ordre de préférence
set radius_authtypes "chap pap"

# Active ou pas la notification de connexion
# auprès du serveur RADIUS (Accounting)
set radius_accounting no

# Autoriser ou pas des connexions multiples d'un nom d'utilisateur
set allow_duplicate_users no

# Chemin du dossier de stockage des connexions clientes
set accounting_dir "/var/run/l2tpns/acct"

# Adresse d'écoute de l2tpns (par défaut, toutes les interfaces)
#set bind_address 1.1.1.1
```

```

# IP de passerelle donnée aux clients
# (par défaut, IP du client et ça marche bien comme ça)
#set peer_address 0.0.0.0

# UID de l'utilisateur pour l2tpns
#set setuid 0

# Définit si on utilise le swap ou pas (peu augmenter la vitesse)
set lock_pages yes

# Retirer le domaine du nom d'utilisateur
load plugin "stripdomain"

# à laisser commenter
# active ou pas la jail des utilisateurs non authentifiés
# si actif, les utilisateurs qui échouent à l'authentification
# se retrouvent connectés sans pouvoir accéder
# au réseau du serveur l2tpns
# si inactif, les utilisateurs qui échouent à l'authentification
# se retrouvent déconnectés
#load plugin "garden"

```

Dans le fichier `/etc/l2tpns/ip_pool`, on inscrit une par ligne, les plages d'adresses IP disponibles pour les adresses virtuelles de clients VPN (au format `adresse_reseau/nb_bits`), le mieux étant de ne pas choisir un sous réseau inclu dans le sous réseau interne du serveur (sinon le routage ne marche pas) :

```
10.10.10.0/24
```

h) Passerelle

Il est important de définir le serveur L2TP comme une passerelle afin que les clients puissent accéder aux serveurs se trouvant sur le réseau interne du serveur. Pour cela, il faut activer le forwarding et faire un NAT sur les paquets provenant de l'interface `tunX` (où `X` est le numéro de l'interface utilisée par `l2tpns`, commence à 0) ou de la plage d'IP défini dans `/etc/l2tpns/ip_pool`. Enfin, on peut filtrer ce qui circule depuis et vers les clients sur la chaîne `FORWARD`.

```

[root]# echo 1 > /proc/sys/net/ipv4/ip_forward
[root]# iptables -t nat -A POSTROUTING -s plage_IP_reseau_VPN -j
SNAT --to-source IP_interne_serveur
ou
[root]# iptables -t nat -A POSTROUTING -o tunX -j SNAT --to-source
IP_interne_serveur

#tout autoriser pour les clients
[root]# iptables -A FORWARD -s plage_IP_reseau_VPN -j ACCEPT
[root]# iptables -A FORWARD -d plage_IP_reseau_VPN -j ACCEPT

```

IV. Configuration du client Windows

a) *Prérequis IPsec*

Si vous avez suivi le tutoriel IPsec, il est à noter que la partie « Stratégies de sécurité IP » ne doit pas être suivie.

Il est nécessaire pour les clients Windows qui doivent avoir un VPN L2TP actif de ne pas avoir la clé suivante ou définie à 0. Elle indique si on active ou pas IPsec. Pour L2TP, il est nécessaire qu'IPsec soit utilisé.

Soyez donc sûr de supprimer ou mettre à 0, la clé :

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\Parameters  
\ProhibitIPsec
```

b) *Configuration*

- Aller dans les « Connexions réseaux » du panneau de configuration
- Cliquer sur « Créer une nouvelle connexion » pour lancer l'assistant d'ajout puis « Suivant »
- Choisir « Connexion au réseau d'entreprise » puis « Suivant »
- Choisir « Connexion réseau privé virtuel » puis « Suivant »
- Donner le nom de cette nouvelle connexion VPN en tant que « Nom de la société » puis « Suivant »
- Choisir « Ne pas établir la connexion initiale » puis « Suivant »
- Entrer ensuite le nom ou l'IP du serveur L2TP auquel vous voulez vous connecter puis « Suivant » puis « Terminer »
- Une fenêtre apparaît alors pour vous permettre de vous connecter.
- Cliquer sur « Propriétés » puis dans l'onglet « Sécurité » :
 - choisir l'Option de sécurité « Avancées (paramètres personnalisés) »
 - cliquer sur « Paramètres... »
 - sélectionner « Aucun cryptage autorisé » dans la liste. (Cryptage inutile vu que l'on utilise IPsec)
 - cocher « Protocole CHAP » (et/ou éventuellement « Mot de passe non crypté (PAP) ») : l2tpns ne supporte que CHAP et PAP.
 - décocher « Protocole Microsoft CHAP » et « Protocole Microsoft CHAP Version 2 »
 - puis OK
- Si vous utilisez une clé prépartagée, cliquer sur « Paramètres IPsec... » puis cocher « Utiliser une clé pré-partagée pour l'authentification » et entrer votre clé puis OK.

- Cliquer sur OK
- Entrer votre nom d'utilisateur et votre mot de passe (pour la connexion au serveur L2TP) et cocher « Enregistrer ce nom d'utilisateur... » (si vous voulez ne pas avoir à les retaper à chaque fois »
- Cliquer enfin sur « Se connecter »

V. Configuration du client Linux (itinérant)

a) Configuration d'IPSec sur le client

Il faut faire plusieurs choses :

- Générer les certificats et CA. Pour plus d'informations sur IPSec, voir le tuto consacré à IPSec.
- Définir les règles IPSec, dans le fichier `/etc/ipsec-tools.conf` :

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

## supprime tous les SA et SPD
#
flush;
spdflush;

## définit que le trafic (IP, ICMP...)
## doit être crypté entre le client
## (votre PC itinérant) et le serveur
## on utilise 0.0.0.0/0 du côté client
## parce que l'on ne connaît pas notre IP
#
spdadd 0.0.0.0/0[any] IP_serveur_VPN/32[any] any -P out ipsec
    esp/transport//require;

spdadd IP_serveur_VPN/32[any] 0.0.0.0/0[any] any -P in ipsec
    esp/transport//require;
```

- Définir la configuration de racoon, dans `/etc/racoon/racoon.conf` :

```
#définit le chemin du dossier contenant les certificats
#du client et des serveurs
path certificate "/etc/certs";

#définit un intervalle d'émission de paquets « vide »
#pour faire garder l'association NAT à la passerelle
timer {
    natt_keepalive 5sec;
}

#pas de socket d'administration
listen {
```

```

        adminsock disabled;
    }

#indique la configuration (phase 1)...vers le serveur VPN
remote IP_serveur_VPN {
    #mode des échanges
    exchange_mode main,aggressive;
    #définit le certificat de l'autorité de certification
    ca_type x509 "ca.crt";
    #indique d'obéir aux propositions du serveur (phase1)
    proposal_check obey;
    #indique que l'on peut traverser une passerelle NAT
    nat_traversal on;
    #indique que l'on peut fragmenter les paquets
    ike_frag on;
    #indique que l'on accepte pas un configuration serveur
    mode_cfg off;

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

    #indique le certificat et la clé privée cliente
    certificate_type x509 "client_ipsec.crt"
"client_ipsec.key";
    #indique d'utiliser le CN des certificats comme identifiant
    my_identifier asn1dn;
    #idem pour les identifiants clients
    peers_identifier asn1dn;

    #initier les connexions
    passive off;
    #jeu de cryptographie pour la phase 1
    proposal {
        encryption_algorithm 3des;
        hash_algorithm sha1;
        authentication_method rsasig;
        dh_group 2;
    }
}

#pour la phase 2
sainfo anonymous {
    pfs_group 2;
    encryption_algorithm 3des;
    authentication_algorithm hmac_sha1;
    compression_algorithm deflate ;
}

```

- redémarrer racoon et activer les règles IPsec

```
[root]# /etc/init.d/racoon stop && /etc/init.d/setkey restart &&
```

```
/etc/init.d/racoon start
```

b) Configuration d'IPSec sur le serveur

Pour plus d'informations sur IPSec, voir le tuto consacré à IPSec (configuration du serveur pour IPSec Windows).

Par exemple, dans `/etc/racoon/racoon.conf` :

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

#pas de socket d'administration
listen {
    adminsock disabled;
}

#pas de critère sur l'IP des clients (itinérants)
remote anonymous {
    #méthode de l'échange de clés
    exchange_mode main,aggressive;
    #générer les règles IPSec à la connexion des clients
    generate_policy on;

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

    #le certificat de l'autorité de certification
    ca_type x509 "ca.crt";
    #le certificat serveur et sa clé privée
    certificate_type x509 "serv_ipsec.crt" "serv_ipsec.key";
    #CN comme identifiant serveur
    my_identifier asn1dn;
    #obéir aux propositions du client
    proposal_check obey;
    #CN comme identifiant client
    peers_identifier asn1dn;
    #attendre les connexions serveur
    passive on;
    #autoriser les passages par une passerelle NAT
    nat_traversal on;
    #fragmentation autorisée
    ike_frag on;
    #délai d'attente pour régénération des clés
    dpd_delay 10;

    #jeu de cryptage pour la phase 1
    proposal {
        encryption_algorithm 3des;
        hash_algorithm sha1;
    }
}
```

```

        authentication_method rsasig;
        dh_group modp1024;
    }
}

#phase 2
sainfo anonymous {
    pfs_group 2;
    encryption_algorithm 3des;
    authentication_algorithm hmac_shal;
    compression_algorithm deflate;
}

```

c) Configuration de rp-l2tp

Là aussi, il faut faire beaucoup de choses :

- Télécharger rp-l2tp à l'adresse <http://sourceforge.net/projects/rp-l2tp/>
- Compiler et installer rp-l2tp

```
[root]# tar xzvf rp-l2tp-*.tar.gz
```

```
[root]# ./configure
```

```
[root]# make
```

```
[root]# make install
```

- Configurer l2tpd, dans le fichier /etc/l2tp/l2tp.conf :

```

# section globale
global

# chargement des modules
# pour le tunnel L2TP/PPP
load-handler "sync-pppd.so"
# le shell de contrôle de L2TPD
load-handler "cmd.so"

# Port 1701 pour L2TP
listen-port 1701

# configuration du tunneleur PPP
section sync-pppd
#configuration en LAC
# -> authentication CHAP
# -> s'authentifier en tant que nom_utilisateur
# -> ne pas déduire l'adresse IP du nom d'hôte
# -> pas d'authentification par PPP
# -> définir le serveur VPN comme passerelle
# -> utiliser les DNS du serveur
# -> autoriser le serveur à donner l'IP locale
#         avec laquelle il nous voit
# -> autoriser le serveur à donner l'IP distante
#         avec laquelle il nous voit

```

```
# -> faire des ping LCP toutes les 30 secondes
# -> réessayer 6 fois avant d'abandonner
lac-pppd-opts "require-chap user nom_utilisateur noipdefault
noauth defaultroute usepeerdns ipcp-accept-local ipcp-accept-
remote lcp-echo-interval 30 lcp-echo-failure 6"
```

```
# Configuration relative à la connexion au serveur VPN
section peer
peer IP_serveur_VPN_L2TP
secret secret_l2tp
#port de connexion au serveur L2TP
port 1701
#indique de se configurer comme un LAC (en gros un client)
lac-handler sync-pppd
#ne rien cacher sur les pairs attribut/valeur
hide-avps no
```

```
# le shell de configuration
section cmd
```

- Ajouter l'utilisateur et mot de passe du VPN pour PAP (/etc/ppp/pap-secrets) :

```
nom_utilisateur * mot_de_passe
```

- Ajouter l'utilisateur et mot de passe du VPN pour CHAP (/etc/ppp/chap-secrets) :

```
nom_utilisateur * mot_de_passe *
```

- Créer le script de démarrage de rp-l2tp (/etc/init.d/rp-l2tp, chmod 755) :

```
#!/bin/sh
# Init file for L2TPD of rp-l2tp

PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
DAEMON=/usr/local/sbin/l2tpd
NAME=l2tpd
DESC="rp-l2tpd"
CONTROL=/usr/local/sbin/l2tp-control
DEFAULT=/etc/default/rp-l2tp
GATEWAY=""
ROUTES=""

test -x $DAEMON || exit 0

set -e

#is the default file exists ?
if [ -f "$DEFAULT" ] ; then
    . $DEFAULT
fi

#is the VPN gateway defined ?
if [ "$GATEWAY" == "" ]; then
    echo "No gateway defined in /etc/default/rp-l2tp"
    exit 1;
```

```

fi

#starting process
start () {

    echo -n "Starting $DESC: $NAME "
    touch /var/run/rp-l2tp
    start-stop-daemon --start --quiet --exec $DAEMON
    sleep 1

    #start a session
    $CONTROL "start-session $GATEWAY" \
        | sed -e 's/OK //' > /var/run/rp-l2tp
    echo "."
    sleep 5

    #setting some route
    echo -n "Setting up routes : "
    ${ROUTES}
    echo "done"
}

#stopping process
stop () {
    #kill session
    echo -n "Stopping $DESC: $NAME "
    $CONTROL "stop-session $(cat /var/run/rp-l2tp)"
    sleep 1
    #stop L2TPD
    rm /var/run/rp-l2tp
    $CONTROL exit
    killall pppd
    sleep 1
    echo "."
}

case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart|force-reload)
        stop
        sleep 1
        start
        ;;
    *)
        N=/etc/init.d/$NAME
        echo "Usage: $N {start|stop|restart|force-reload}" >&2
        exit 1
        ;;
)

```

```
esac
exit 0
```

- Inscrire la configuration du tunnel L2TP (/etc/default/rp-l2tp):

```
# IP du serveur VPN IPSec/L2TP
GATEWAY=192.168.0.25
```

```
#le serveur VPN est inscrit comme passerelle par défaut
#on peut spécifier en plus des routes
```

```
#liste de commandes route pour le VPN séparée par ;
ROUTES=""
```

- Inscrire le script pour lancement au démarrage (ordre > 20)

```
[root]# update-rc.d rp-l2tp defaults 99
```

VI. Et iptables dans tout ça

a) La base

Il est important de ne pas laisser trop accessible le serveur MySQL et FreeRadius.

Dans tous les cas, et pour le bon fonctionnement de Linux, il faut tout autorisé sur la boucle locale :

```
[root]# iptables -A INPUT -i lo -j ACCEPT
[root]# iptables -A OUTPUT -o lo -j ACCEPT
```

Et enfin, une police de suppression par défaut :

```
[root]# iptables -A INPUT -P DROP
[root]# iptables -A OUTPUT -P DROP
[root]# iptables -A FORWARD -P DROP
```

b) Pour IPSec

Déjà, il faut autoriser ESP (ou AH ou les deux mais surtout ESP) en fonction de la configuration d'IPSec :

```
#ESP
iptables -A OUTPUT -p 50 -s IP_publicue -j ACCEPT
iptables -A INPUT -p 50 IP_publicue -j ACCEPT
#AH
iptables -A OUTPUT -p 51 -s IP_publicue -j ACCEPT
iptables -A INPUT -p 51 -d IP_publicue -j ACCEPT
```

Ensuite, il faut autoriser IKE (racoon) :

```
iptables -A INPUT -p udp --sport 500 --dport 500 -d IP_publicue
```

```
-j ACCEPT
iptables -A OUTPUT -p udp --sport 500 --dport 500 -s IP_publicue
-j ACCEPT
#si on fait du NAT-T
iptables -A INPUT -p udp --sport 4500 --dport 4500 -d IP_publicue
-j ACCEPT
iptables -A OUTPUT -p udp --sport 4500 --dport 4500 -s IP_publicue
-j ACCEPT
```

c) Pour L2TP

L2TP utilise les ports UDP :

- 32792 : port utilisé pour le clustering
- 1701 : le port des paquets UDP L2TP
- 1702 : normalement pas utile
- 3799 : port RADIUS pour DAE (Changement d'autorisation, déconnexion)

De plus, il faut interdire l'accès à des paquets non cryptés pour L2TP car L2TP doit être encapsulé dans IPSec. Pour cela, il est nécessaire d'utiliser le marquage de paquets afin de savoir quels paquets ont été décryptés mais cela repose aussi sur les principes suivants :

- le marquage et le module « *mark* » d'iptables sont interne au noyau et ne modifie donc pas les paquets une fois sortie de la carte réseau
- le marquage des paquets est conservé pendant et après le décryptage du paquet

En conséquence, on devra faire :

```
#marquage des paquets cryptés
[root]# iptables -t mangle -A PREROUTING -i interface_publicue -p
esp -j MARK --set-mark 1
#accepter les paquets qui sont marqués donc qui ont été décryptés
[root]# iptables -A INPUT -i interface_publicue -m mark --mark 1
-j ACCEPT
[root]# iptables -A OUTPUT -p udp --sport 1701 - j ACCEPT
```

Et éventuellement, accepter le clustering de L2TP, uniquement sur le réseau local :

```
[root]# iptables -A INPUT -p udp --dport 32792 -s IP_réseau_local
- j ACCEPT
[root]# iptables -A OUTPUT -p udp --sport 32792 -d IP_réseau_local
- j ACCEPT
```

d) Pour FreeRADIUS

Le service RADIUS utilise historiquement le port UDP 1645 mais les nouvelles RFC lui accorde de

préférence le port UDP 1812.

Au final, RADIUS utilise les port suivant par défaut :

- 1812 : pour les requêtes d'authentification
- 1813 : pour la gestion des connexions des utilisateur authentifiés (accounting)
- 1814 : pour la gestion de proxy RADIUS

Dans notre cas, le serveur RADIUS se trouve sur la même machine que le serveur L2TP, ce qui peut se réaliser de deux façons :

- dans le fichier `/etc/freeradius/radiusd.conf`, on peut mettre `bind_address = 127.0.0.1` pour forcer l'écoute sur la seule boucle locale.
- fermer l'accès depuis tout le reste sauf localhost en utilisant la police DROP par défaut

Si l'on souhaite déplacer le serveur RADIUS sur une autre machine, on devra faire ceci :

```
#requêtes
[root]# iptables -A INPUT -p udp --dport 1812 -s IP_serveur_L2TP
- j ACCEPT
[root]# iptables -A OUTPUT -p udp --sport 1812 -d IP_serveur_L2TP
- j ACCEPT
#accounting
[root]# iptables -A INPUT -p udp --dport 1813 -s IP_serveur_L2TP -
j ACCEPT
[root]# iptables -A OUTPUT -p udp --sport 1813 -d IP_serveur_L2TP
- j ACCEPT
#proxy
[root]# iptables -A INPUT -p udp --dport 1814 -s IP_serveur_L2TP -
j ACCEPT
[root]# iptables -A OUTPUT -p udp --sport 1814 -d IP_serveur_L2TP
- j ACCEPT
```

e) Pour MySQL

MySQL utilise le port TCP 3306 pour les connexions au démon `mysqld`. Il est presque toujours impératif que seuls les machines autorisées aient accès à ce port.

Dans notre cas, on peut forcer MySQL à n'écouter que sur localhost, en mettant `bind-address = 127.0.0.1` dans `/etc/mysql/my.cnf`.

Si le serveur MySQL n'est pas sur le serveur RADIUS, on utilisera :

```
[root]# iptables -A INPUT -p tcp --dport 3306 -s IP_serveur_RADIUS
- j ACCEPT
[root]# iptables -A OUTPUT -p tcp --sport 3306 -d
IP_serveur_RADIUS - j ACCEPT
```

VII. Bibliographie

[Architecture L2TP - Layer 2 Tunneling Protocol](#)

[Protocole L2TP](#)

[Layer 2 Tunneling Protocol – Wikipédia](#)

[SourceForge.net: l2tpns](#)

[Debian Administration :: Creating a radius based VPN with support](#)

[SourceForge.net: RP-L2TP](#)

[Using a Linux L2TP/IPsec VPN server](#)