

# Sécurité GNU/Linux

By ShareVB

## Le protocole PPTP

### Table des matières

I.Fonctionnement.....	1
II.Implémentation Linux.....	3
1Client.....	3
2Serveur.....	4
III.Et iptables dans tout ça.....	5
1Pare-feu serveur.....	5
a)Établissement de la connexion et tunnel PPP/GRE.....	5
b)Filtrage du trafic à destination du serveur par le tunnel.....	6
c)Filtrage du trafic à travers le tunnel vers le réseau interne du serveur.....	6
2Pare-feu client.....	7
IV.Configuration du client Windows.....	7
V.Bibliographie.....	9

### I. Fonctionnement

Le principe du protocole PPTP (RFC2637) (*Point To Point Tunneling Protocol*) est de créer des trames avec le protocole PPP et de les crypter puis de les encapsuler dans un paquet IP.

Cela permet de relier les deux réseaux par une connexion point-à-point *virtuelle* acheminée par une connexion IP sur Internet. Cela fait croire aux deux réseaux qu'ils sont reliés par une ligne directe.

On garde, ainsi les adresses des réseaux physiques dans la trame PPP cryptées et cette trame est acheminée normalement sur Internet vers l'autre réseau.

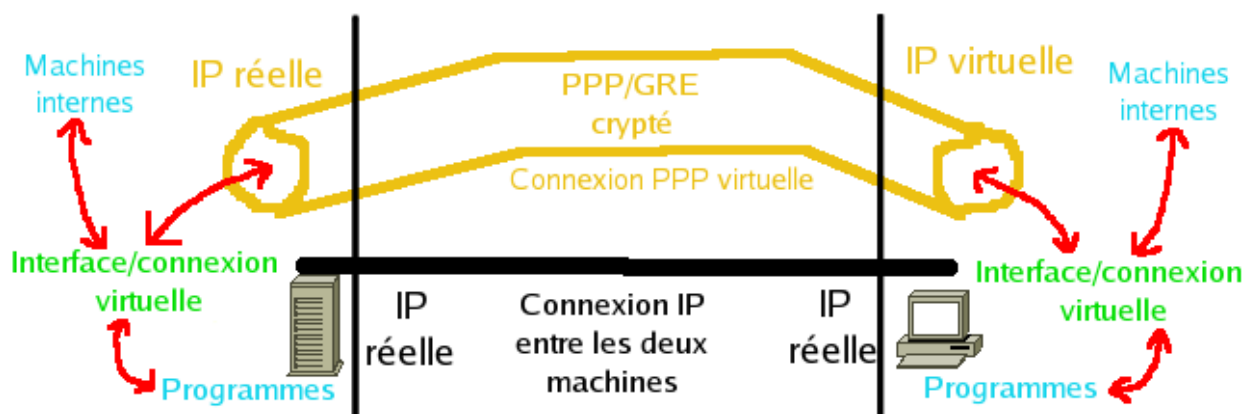
Il permet les opérations suivantes :

- L'authentification se fait par le protocole MS-CHAP (Challenge Handshake Authentication Protocol) version 2 ou avec le protocole PAP (Password Authentication Protocol)
- L'encryption se fait par le protocole MPPE (Microsoft Point-to-Point Encryption). Cela crée un tunnel de niveau 3 (Réseau) géré par le protocole GRE (Generic Routing Encapsulation).
- La compression peut se faire avec le protocole MPPC (Microsoft Point to Point Compression)

- On peut ajouter autant de protocole que l'on veut dans le protocole PPTP pour l'encryption et la compression des données

La connexion se passe donc ainsi :

- Le client se connecte à Internet par son modem par le protocole PPP (classiquement)
- Le client se connecte alors au serveur VPN par une connexion IP encapsulant les paquets GRE/PPP cryptés. Ainsi cela forme deux connexions l'une sur l'autre
  - la connexion normale à Internet : elle achemine le trafic vers/depus Internet
  - la connexion virtuelle au dessus de la connexion Internet : elle achemine le trafic vers/depus le réseaux VPN
- A la fin de la connexion c'est le serveur qui ferme le tunnel



On obtient donc une connexion PPP au dessus de la connexion Internet ou Ethernet qui nous donne accès au serveur VPN pptpd. Cette connexion PPP obtient une IP de la plage définie dans la configuration de pptpd. Sur le serveur, on a une connexion de son IP publique vers l'IP virtuelle du client et sur le client c'est l'inverse.

Voici ce que cela donne avec un ping vers une **machine derrière** le serveur:

- ping

```
11:00:17.003113 IP IP_client > IP_serveur: GREv1, call 128, seq 120, length 101: compressed PPP data
```

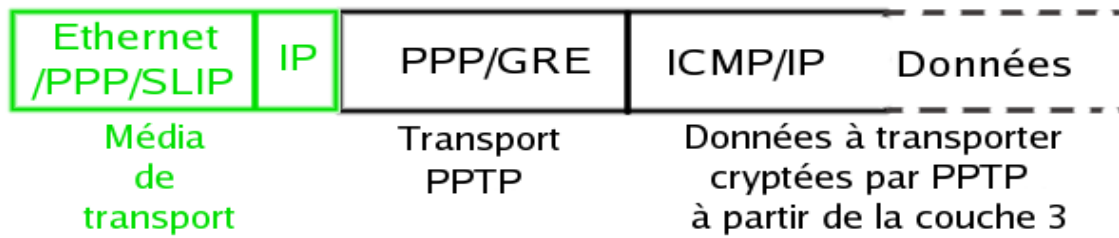
```
11:00:17.503088 IP IP_client > IP_serveur: GREv1, call 128, ack 116, no-payload, length 12
```

- pong

```
11:02:12.840243 IP IP_serveur > IP_client: GREv1, call 0, seq 135, ack 139, length 105: compressed PPP data
```

On voit donc uniquement les paquets cryptés.

Un paquet d'une connexion PPTP ressemble donc à ceci :



Il est encore beaucoup utilisé du fait qu'il est nativement intégré aux systèmes Windows. Mais les protocoles tels que IPSec ou OpenVPN sont bien meilleurs en sécurité et en performances.

## II. Implémentation Linux

### 1 Client

Il existe un client PPTP pour Linux acceptant ce protocole de Microsoft : `pptp-client` (<http://pptpclient.sourceforge.net/>). Sous Debian le paquetage s'appelle `pptp-linux` et sous Fedora `pptp`.

Les étapes d'installation générale sont les suivantes (pour plus de détails voir sur le site de `pptp-client`):

- intégrer MPPE (Microsoft Point-to-Point Encryption et MS-CHAPv2) à votre noyau :
  - Première solution recompilation noyau
    - il faut que vous téléchargez les sources de votre noyau soit sur [kernel.org](http://kernel.org), soit avec `yum` ou `apt-get` et le paquet `kernel-sources-votre-version-du-noyau` (voir `uname -r`). Les sources doivent se trouver dans `/usr/src/linux`.
    - récupérer le patch MPPE pour votre noyau
    - appliquer le et recompiler votre noyau (si vous ne savez pas le faire, opter plutôt pour la troisième solution)
  - Deuxième solution : compiler MPPE en module (par exemple, <http://pptpclient.sourceforge.net/howto-debian-dkms.phtml>)
  - Troisième solution : télécharger une version compilée pour votre version du noyau et l'installer (en `deb`, `rpm`...)
  - Pour avoir la démarche d'installation du module MPPE propre à votre distribution, reportez vous à <http://pptpclient.sourceforge.net/documentation.phtml>

- intégrer MPPE dans PPP (à faire si la suite ne marche pas ou que vous avez une version de ppp < 2.4.2 (pppd --version))
  - télécharger les sources de PPP, OpenSSL et le patch pour MMPE (PPP/OpenSSL)
  - décompresser toute cela et appliquer le patch
  - recompiler PPP (./configure et make)
  - copier le fichier pppd généré à la place de l'ancien (en faisant une copie avant)
- installer le client PPTP (package pptp) : `yum install pptp`
- installer pptpconfig, par exemple pour FC5 :
  - `rpm -Uvh`  
<http://pptpclient.sourceforge.net/yum/stable/fc5/pptp-release-current.noarch.rpm>
  - `yum --enablerepo=pptp-stable install pptpconfig`
- configurer le client PPTP et lancer le tunnel avec pptpconfig qui est une interface GUI
  - pour cela, il vous faut connaître l'IP ou le nom DNS du serveur, le nom du VPN, le nom d'utilisateur, le mot de passe, le type d'encryptage.
  - Lancer pptpconfig en tant que root puis entrer les informations précédente dans l'onglet Server. S'il y a encryptage, cocher la case MPPE dans l'onglet Encryption.
  - Cliquer sur Add et la connexion apparaît dans la liste. Sélectionnez la et cliquez sur Start. S'il n'y a pas d'erreur, cliquez sur le bouton Ping test.
  - Si pas d'erreurs, cliquez sur Stop, resélectionnez la connexion et dans l'onglet Routing, ajouter les routes vers les LAN que vous pouvez atteindre derrière le serveur.
  - Redémarrez ensuite la connexion par Start.

Le serveur PPTP peut être indistinctement sur un serveur Windows ou Linux.

## 2 Serveur

Le principal serveur PPTP pour Linux est PopTop. Le paquetage s'appelle pptpd sous Debian. Sous Fedora, vous devez le télécharger depuis le site [www.poptop.org](http://www.poptop.org).

Il requière pour son installation une partie des étapes d'installation du client PPTP (voir plus haut) :

- intégration de MPPE dans votre noyau
- intégration de MPPE dans PPP
- installation de PPTPD (paquetage du même nom ou à partir des sources)
- configuration du serveur
  - fichier `/etc/pptpd.conf`. Cela donne la plage des adresses virtuelles des clients,

par exemple :

```
#### /etc/pptp.conf ####
# Fichier d'options de PPTP
option /etc/ppp/pptpd-options
# Adresse IP publique du serveur PPTP

# que l'on est en train de configurer
localip IP_serveur_VPN
# Plage d'adresses IP à attribuer aux clients qui se connectent
remoteip plage_IP_VPN # par ex: 192.168.0.239-243,192.168.0.245
####
```

O fichier /etc/ppp/pptpd-options. Cela autorise le démon ppp à accepter PPTP, par exemple :

```
# Nom local pour la connexion PPTP
# (doit correspondre au second champ des lignes de /etc/ppp/chap-secrets)
name nomVPN

# Encryption
# Debian: on systems with a kernel built with the package
# kernel-patch-mppe >= 2.4.2 and using ppp >= 2.4.2, ...
# indique l'authentification pur PPTP
# {{{
refuse-pap
refuse-chap
refuse-mschap
# Utiliser obligatoirement MS-CHAPv2 [Microsoft
# Challenge Handshake Authentication Protocol, Version 2] pour authentification.
require-mschap-v2
# Utiliser un exncryptage MPPE 128-bit
# (MPPE nécessite l'utilisation de MSCHAP-V2 durant l'authentification)
require-mppe-128
# }}}

# Options réseaux et de routage

# permet d'indiquer des DNS primaire et secondaire
# pour les clients PPTP (Linux ou Windows)
#ms-dns IP_dns_primaire
#ms-dns IP_dns_secondaire

# permet d'indiquer des serveurs WINS primaire et secondaire
# pour les clients Windows
#ms-wins IP_wins_primaire
#ms-wins IP_wins_secondaire

# Ajouter une entrée dans la table ARP [Address Resolution Protocol] locale
# avec l'adresse IP et l'adresse MAC Ethernet des clients connectés
# Cela permet de faire comme si le client était sur le réseau interne du serveur
# uniquement nécessaire si le serveur ne fait pas passerelle
proxyarp

# Debian: do not replace the default route
nodefaultroute

# Miscellaneous

# Create a UUCP-style lock file for the pseudo-tty to ensure exclusive
# access.
lock
# Disable BSD-Compress compression
nobsdcomp
```

- fichier `/etc/ppp/chap-secrets`. Cela sert à contenir les logins/mots de passe des utilisateurs autorisés à se connecter, par exemple :

```
#### /etc/ppp/chap-secrets ####
# Il peut y avoir aussi les identifiants de connexion Internet ADSL du serveur

# utilisateur      nom VPN      mot de passe      @IP source
utilisateur        nomVPN      motdepasse        *
####
```

- lancer le serveur `/etc/init.d/pptpd restart`

### III. Et iptables dans tout ça...

#### 1 Pare-feu serveur

##### a) Établissement de la connexion et tunnel PPP/GRE

```
#autorise les paquets PPTP (PPP/GRE) à entrer et sortir
iptables -A INPUT -p gre -j ACCEPT
iptables -A OUTPUT -p gre -j ACCEPT

#autorise la mise en place du tunnel PPTP (authentification)
iptables -A INPUT -p tcp --dport 1723 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 1723 -j ACCEPT
```

##### b) Filtrage du trafic à destination du serveur par le tunnel

Pour filtrer le trafic à destination du serveur passant par le tunnel PPTP, on doit définir :

- une règle pour INPUT sur l'interface `pppN` dédiée à PPTP

```
iptables -A INPUT -i pppN sélection_type_paquet -d
IP_res_int_serveur -j ACCEPT
```

- une règle pour OUTPUT sur l'interface `pppN` dédiée à PPTP

```
iptables -A OUTPUT -o pppN sélection_type_paquet -s
IP_res_int_serveur -j ACCEPT
```

Par exemple, pour des pings et des connexions HTTP sur le serveur :

```
#autorise les pings des clients vers le serveur
iptables -A INPUT -i pppN -p icmp --icmp-type echo-request -d IP_res_int_serveur
-j ACCEPT
iptables -A OUTPUT -o pppN -p icmp --icmp-type echo-reply -s IP_res_int_serveur
-j ACCEPT

#autorise les connexions HTTP des clients vers le serveur
```

```
iptables -A INPUT -i pppN -p tcp --dport 80 -d IP_res_int_serveur -j ACCEPT
iptables -A OUTPUT -o pppN -p tcp --sport 80 -s IP_res_int_serveur -j ACCEPT
```

### c) Filtrage du trafic à travers le tunnel vers le réseau interne du serveur

Le principe est le même que précédemment à l'exception que l'on utilise uniquement la chaîne FORWARD. Il faudra en plus activer l'IP forwarding/SNAT pour autoriser les paquets à traverser le serveur :

- une règle pour FORWARD (client <--> réseau interne) sur l'interface pppN dédiée à PPTP

```
iptables -A FORWARD -i pppN sélection_type_paquet -d
IP_res_int_serveur -j ACCEPT
```

- une règle pour FORWARD (réseau interne <--> client) sur l'interface pppN dédiée à PPTP

```
iptables -A FORWARD -o pppN sélection_type_paquet -s
IP_res_int_serveur -j ACCEPT
```

Activation de l'IP forwarding/SNAT :

```
# effectuer la NAT des paquets traversant le serveur PPTP
iptables -t nat -A POSTROUTING -s plage_IP_client_VPN -o interface_LAN -j SNAT
--to-source IP_serveur_LAN
```

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

Par exemple, pour des pings et des connexions HTTP sur le serveur :

```
#autorise les pings des clients vers le LAN du serveur
iptables -A FORWARD -i pppN -p icmp --icmp-type echo-request -d
Adresse_LAN_serveur -j ACCEPT
iptables -A FORWARD -o pppN -p icmp --icmp-type echo-reply -s
Adresse_LAN_serveur -j ACCEPT

#autorise les connexions HTTP des clients vers le LAN du serveur
iptables -A FORWARD -i pppN -p tcp --dport 80 -d Adresse_LAN_serveur -j ACCEPT
iptables -A FORWARD -o pppN -p tcp --sport 80 -s Adresse_LAN_serveur -j ACCEPT

#autorise les résolutions DNS des clients vers le LAN du serveur
iptables -A FORWARD -i pppN -p udp --dport 53 -d Adresse_LAN_serveur -j ACCEPT
iptables -A FORWARD -o pppN -p udp --sport 53 -s Adresse_LAN_serveur -j ACCEPT
```

## 2 Pare-feu client

Le pare-feu client peut se simplifier ainsi : « autorisation de tout trafic à travers le tunnel PPTP ».

Pour l'initialisation de la connexion PPTP :

```
#autorise les paquets PPTP (PPP/GRE) à entrer et sortir
iptables -A INPUT -p gre -j ACCEPT
iptables -A OUTPUT -p gre -j ACCEPT

#autorise la mise en place du tunnel PPTP (authentification)
iptables -A INPUT -p tcp --dport 1723 -j ACCEPT
```

```
iptables -A OUTPUT -p tcp --sport 1723 -j ACCEPT
```

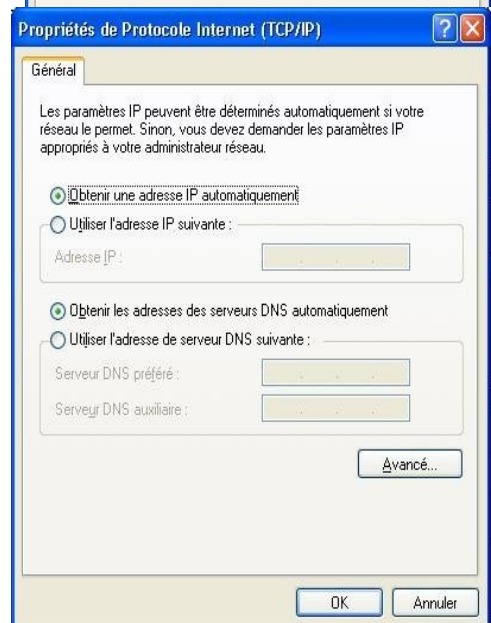
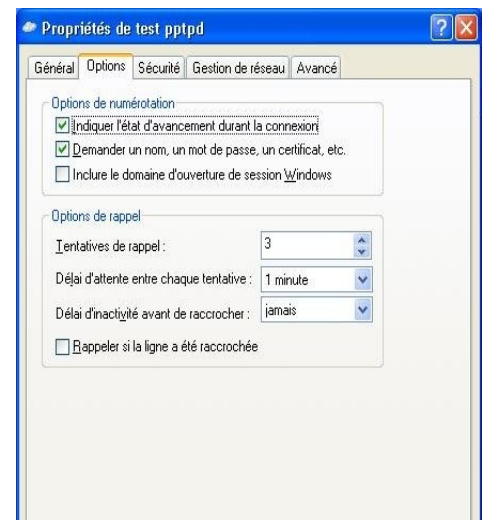
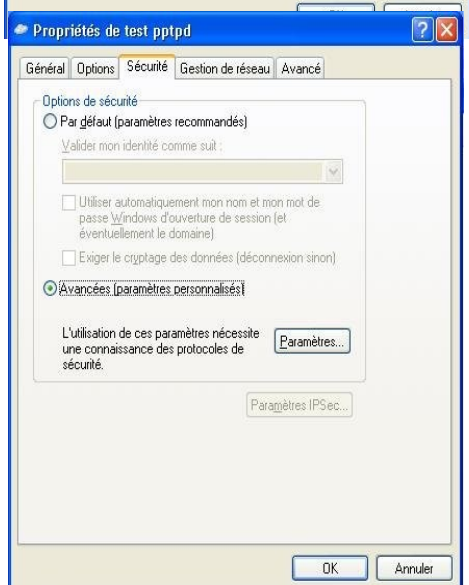
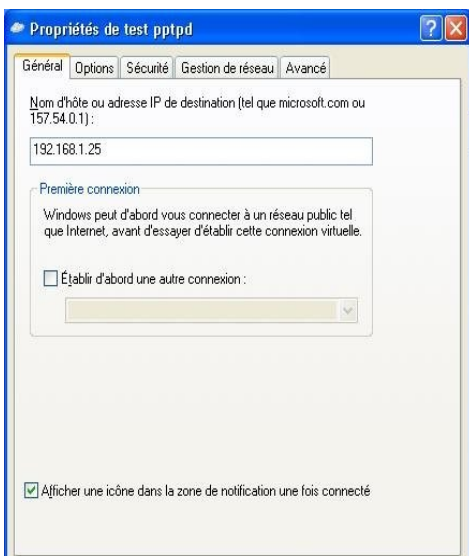
Pour le trafic passant par le tunnel :

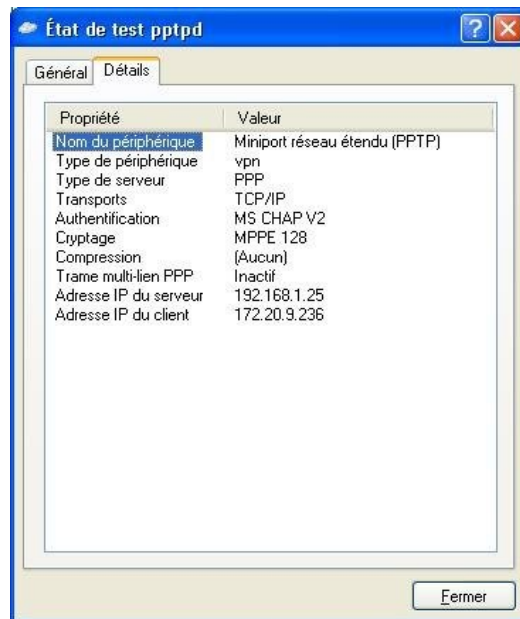
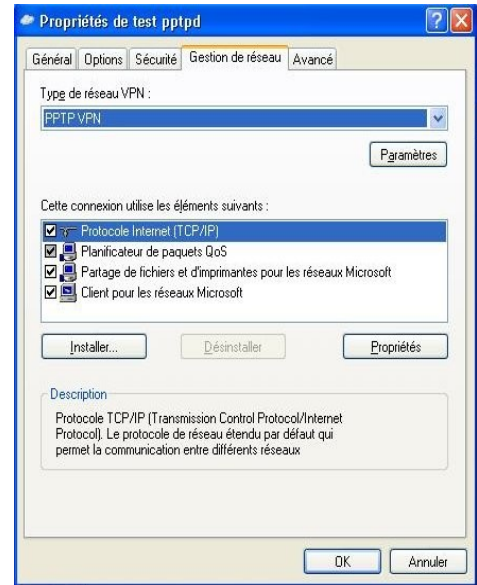
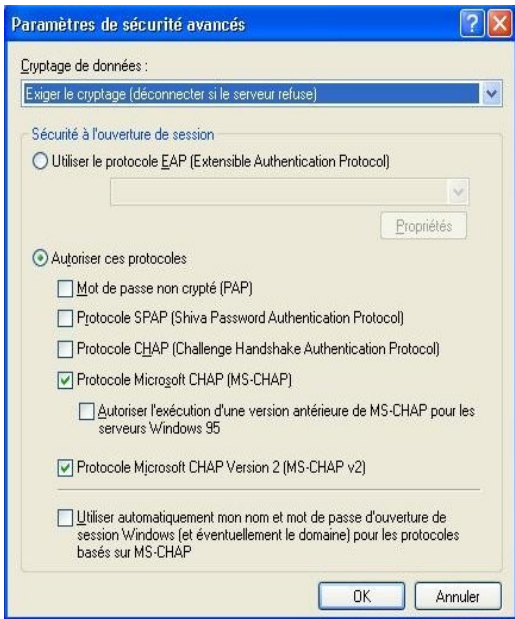
```
#autorise le trafic par le tunnel PPTP sur pppN
iptables -A INPUT -i pppN -j ACCEPT
iptables -A OUTPUT -o pppN -j ACCEPT
```

## IV. Configuration du client Windows

Pour configurer un client PPTP sous Windows (peut varier en fonction des versions) :

- Panneau de configuration
- Connexions réseaux et accès à distance
- Créez une nouvelle connexion à distance :
  - Réseau d'entreprise/Réseau Privé Virtuel
  - remplir avec l'IP du serveur PPTP et les logins/mots de passes que l'on vous a attribués
- Editer les propriétés de la connexion afin d'obtenir quelque chose comme ceci :





## V. Bibliographie

[PPTP Client](#)

[Protocole PPTP \(Point-to-Point Tunneling Protocol\)](#)

[Poptop - Open Source PPTP Server](#)