Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

poptop: haciendo túneles VPN de Windows a Linux (o de Linux a Linux)

Per **Carles Pina i Estany**, ***cpina*** (http://pinux.info)

Creado el 29/04/2003 14:28 modificado el 04/05/2003 11:52

pptp[1] es un protocolo para hacer VPN, el cual viene en casi cualquier Hasefroch[2] (98, 2000,...). Nos permitirá, desde un Hasefroch conectarnos de una manera muy fácil a un servidor Linux con pptp de forma encriptada, y también acceder a la red local donde se encuentre el servidor. Veremos como configurar el servidor y un poco por encima el Hasefroch para hacer de cliente.

Actualizado: configuración cliente Linux[3], aclarado porqué poner IP's de la misma red (gracias a proxyarp), añadida sección Enlaces[4] y retoques menores.

Introducción

Aclaremos algunas ideas que tendremos que usar en el artículo:

- **VPN** (Virtual Private Network-Red Privada Virtual): nos permite extender nuestra red "más lejos". Es decir, tenemos nuestra red privada (con IP's privadas, etc.) y desde fuera de la red podremos acceder a los recursos de esta (normalmente de forma más o menos segura)
- **poptop**[5]: es la implementación para Linux del protocolo de Hasefroch
- **MSCHAPv2**: protocolo de autentificación cifrado
- **MPPE**: protocolo de cifrado de los datos

Las **ventajas** de usar pptp (no exclusivas de él) para hacer una VPN son:

- Conectaremos de Windows a Linux de forma segura a través de una red insegura
- También nos permitirá acceder a la red privada donde está conectado el servidor Linux
- La parte cliente viene con cualquier Windows. De esa forma para una empresa no hace falta instalar software adicional (como podria ser para IPSec)
- Permite usar también Linux como cliente

Desventaja: es un protocolo que no es totalmente seguro.[6] De todas formas seguramente nos servirá en muchos casos, sobretodo por las ventajas vistas antes. En otros casos mejor usar de IPSec (p. ej. freeswan[7])

Configuración del servidor sin cifrado

Para instalarlo en Debian necesitamos el paquete `pptpd` (en otras distribuciones puede variar)

Para el ejemplo imaginaremos un servidor Linux con dos interfaces: una pública donde recibe las peticiones y otra privada (p. ej. 192.168.1.1).
El demonio pptpd se queda escuchando al puerto 1723 para peticiones de clientes. Una vez recibida una petición pasará el control de autentifiación al demonio pppd.

El fichero de configuración `/etc/pptpd.config`:

```
speed 115200
#en verdad conectará a la velocidad máxima de la red
```

```
option /etc/ppp/pptpd-options
localip 192.168.2.234-238,192.168.2.245
remoteip 192.168.1.234-238,192.168.1.245
#yo pongo directamente una IP privada de la red del servidor de Linux
#como IP remota, será transparente para los equipos de la red.
#Podemos hacerlo así debido a que tenemos la opción proxyarp en
#/etc/ppp/pptp-options y en el cliente. Así el servidor pptp
#hará de servidor proxy, y aún sin estar en la misma red local
#será transparente al sistema operativo como si lo estuviera.
```

En el fichero `/etc/ppp/pptp-options` pondremos:

```
name servername
domain domainname
auth
ms-dns 62.37.228.20
netmask 255.255.255.0
nodefaultroute
proxyarp
lock
```

Y en el fichero `/etc/ppp/pap-secrets` algo como:

```
hola    *       adeu    *
```

(el segundo y cuarto campo desde donde pueden conectarnos a nosotros.
Ahora iremos al Hasefroch y configuramos la conexión **sin cifrado**. Desde ahora deberiamos poder conectarnos con el Hasefroch al Linux, haciendo desde Hasefroch un ping a la IP privada de Linux.
Para poder hacer pings a las otras IP's privadas de la red de Linux tendriamos que tener una configuración de iptables al menos así:

```
iptables -F FORWARD
iptables -P FORWARD ACCEPT
echo 1 >  /proc/sys/net/ipv4/ip_forward
#¡ojo! eso es muy inseguro, limitad más el FORWARD entre interfaces,
#sólo apto para hacer pruebas
```

Configuración del servidor con cifrado

Nos queda la parte más "divertida" del asunto: necesitamos parchear el Kernel y el pppd para poder usar contraseña cifrada y también datos cifrados.
Lo podemos hacer mediante el fichero ppp-2.4.2.tar.gz[8] que encontraremos en la sección Downloads[9] de la web de poptop[10].

Este fichero incluye el pppd con soporte para MPPE y los parches necesarios para el Kernel. Hay otros ficheros en Internet que son Kernels enteros, parches para el Kernel, parches para el pppd, etc. pero usaremos este sistema.
Procedemos:

pppd con MPPE

```
gunzip ppp-2.4.2.tar.gz
tar -xvf ppp-2.4.2.tar
cd ppp-2.4.2
./configure
make
make install
```

Eso es, los típicos pasos para compilar un programa desde código fuente. Este ppp ya viene parcheado para el soporte de `MSCHAPv2` y `MPPE`.
(En mi caso no compilaba por culpa del soporte de Radius que no necesitamos para lo que hacemos, en este caso editamos el fichero `pppd/plugins/Makefile` y en SUBDIRS quitamos la palabra "radius")

Nota: si bajamos la última versión del ftp de pppd[11] (actualmente *ppp-2.4.2b3.tar.gz* también tenemos soporte `MPPE` y `MSCHAPv2` de serie siempre que compilemos con `make CHAPMS=1 USE_CRYPT=1`.

Podemos comprobar que tenemos el pppd instalado con soporte de `mppe` haciendo eso:

```
strings $(which pppd) | grep -i mppe | wc -l
```

(si nos da diferente de 0 tenemos el pppd correcto)

kernel con mppe

Nos falta el parche para el kernel de `mppe`. Hay varios métodos de encontrar este parche (ya en Kernels compilados, en parche individual para algunos Kernels, etc.). Nosotros lo haremos entrando en el directorio `ppp-2.4.2/linux` y haciendo:

```
./kinstall.sh
cd mppe

chmod u+x mppeinstall.sh
./mppeinstall.sh /usr/src/linux-2.4.20
```

El `kinstall.sh` sólo nos actualiza un poco más si hace falta el driver ppp del kernel (normalmente no haria falta). El mppeinstall, parchea el kernel para que tenga soporte de mppe. Seguidamente configuraremos el Kernel como nos sea habitual pero miramos que en *Network device suppport* tengamos activado el *PPP* y el *PPP MPPE compression (encryption)* y después ya podemos compilar lo y reiniciamos de nuevo. Podremos cargar el módulo ppp-mppe:

```
modprobe ppp_mppe
```

Para ver que todo es correcto (es normal que nos diga que el kernel ha sido tainted).

Ahora activaremos la encriptación de contraseña y datos en el `/etc/ppp/pptp-options`:

```
require-mschap-v2
require-mppe-128
require-mppe-40
```

y ponemos la línea:

```
hola    *       adeu    *
```

En `/etc/ppp/chap-secrets`

Ahora podemos volver a la configuración del Hasefroch y activar el cifrado de la contraseña y el de datos para conectarnos a nuestro servidor pptp Linux de forma más segura.

Configuración cliente Hasefroch 2000

Explico paso a paso como hacerlo con Hasefroch. Casi no haria falta tan detallado, pero por si acaso:

1. Hacemos click con el botón derecho en *"Mis sitios de Red"*
2. Doble-click en *"Nueva Conexión"*
3. Marcamos *"Conectar a una red privada a través de Internet"* y hacemos *"Siguiente"*
4. Nos pide la IP o el nombre del servidor pptp. Lo ponemos y *"Siguiente"*
5. Ponemos la opción *"Sólo para mí"*
6. Nos pide un nombre para la conexión. Cualquier cosa y *"Siguiente"*
7. Ponemos el nombre de usuario y la contraseña que pusimos en el `pap-secrets/chap-secrets`.

SIN CIFRADO

8. Vamos a *"Propiedades"*, pestaña *"Seguridad"*
9. *"Avanzada (configuración personalizada)"* y hacemos click en *"Configuración"*
10. En la combobox de arriba seleccionamos *"Cifrado opcional (conectar incluso sin cifrado)"* y en abajo permitimos *"Contraseña no cifrada (PAP)"* y clickamos en *"Aceptar"*
11. Salimos aceptando todo, y finalmente presionamos en *"Conectar"*

CON CIFRADO

8. Presionamos en *"Conectar"* (por defecto viene el cifrado activado) (si habiamos desactivado el cifrado lo activamos de nuevo junto con la contraseña segura)

Configuración cliente Linux

Haremos directamente la versión *con* cifrado, ya que es lo que más usaremos y más problemas nos puede dar.

Para configurar el cliente Linux tendremos que instalar el pppd con soporte MPPE y CHAPMS2 tal como hiciemos en el servidor (aquí[12]).
También instalaremos los paquetes `pptp-linux` y `pptpd` (en Debian, en otras distribuciones puede cambiar el nombre). Una vez todo eso pondremos los ficheros de configuración. En el `/etc/ppp/options.pptp`:

```
debug noauth lock nobsdcomp nodeflate
```

Y en el `/etc/ppp/peers/tunnel`:

```
pty "pptp 192.168.1.3 --nolaunchpppd"
name hola
remotename PPTP
require-mschap-v2
require-mppe-128
file /etc/ppp/options.pptp
ipparam tunnel
```

Donde `name hola` es el login y `remotename PPTP` es el nombre que usaremos para referirnos a esa conexión (en otro fichero, etc.). La IP que vemos es la del servidor pptp.

Ahora sólo nos queda el `/etc/ppp/chap-secrets`:

```
hola    PPTP    adeu    *
```

Hecho eso ya nos podremos conectar mediante `pon tunnel`.

Aquí viene nuestra elección haber puesto en el `options.pptp` la opción `defaultroute` que hará que todo el tráfico que no sea a la red local pase a través del servidor pptp o bien hacerlo a mano.

Para desconectar hariamos `poff`.

Si añadimos en el fichero `/etc/ppp/options.pptp` `debug` podremos ver qué falla en los logs de `/var/log/daemon.log` (ver tanto del cliente como del servidor nos ayudará a entender qué pasa).

Si dudamos del `pppd` que tenemos podemos actualizar desde la página Web de ppp[13]. Tenemos la última beta aquí[11]. Pero importante, si queremos soporte para MPPE y MSCHAPv2 tenemos que compilarlo así (aparte de ejecutar los scripts para que parchee el Kernel):

```
make CHAPMS=1 USE_CRYPT=1
```

La versión `ppp-2.4.2b3.tar.gz` me ha funcionado correctamente sin tener que desactivar *radius* tal como hiciemos antes. Ya incluye los scripts para parchear el Kernel y añadirle soporte MPPE.

Enlaces

- Página del servidor pptp para Linux[1]
- Excelente página del cliente pptp para Linux.
- Dentro de la anterior, tenemos para resolver problemas[14] (tiene un listado de problemas/soluciones frecuentes **muy** bueno) y la configuración[15] que he usado para Debian Woody (también está para varias versiones de RedHat).
- Podemos encontrar el último ppp en su página Web[13] (CVS, que es el que ya lleva soporte para MSCHAP2 y MPPE).

Nota: aviso de seguridad[16] del paquete pptpd.

Lista de enlaces de este artículo:

1. http://www.poptop.org/
2. http://bulma.net/body.phtml?nIdNoticia=1613
3. http://bulma.net/body.phtml?nIdNoticia=1743&nIdPage=4#clientelinux
4. http://bulma.net/body.phtml?nIdNoticia=1743&nIdPage=4#enlaces
5. http://www.poptop.org
6. http://www.argo.es/~jcea/artic/hispasec17.htm
7. http://www.freeswan.org
8. http://prdownloads.sourceforge.net/poptop/ppp-2.4.2.tar.gz?download
9. http://sourceforge.net/project/showfiles.php?group_id=44827
10. http://poptop.sourceforge.net/
11. ftp://ftp.samba.org/pub/ppp/
12. http://bulma.net/body.phtml?nIdNoticia=1743nIdPage=3
13. http://www.samba.org/ppp/
14. http://pptpclient.sourceforge.net/
15. http://pptpclient.sourceforge.net/howto-debian.phtml
16. http://www.debian.org/security/2003/dsa-295

E-mail del autor: carles _ARROBA_ pinux.info
Podrás encontrar este artículo e información adicional en: http://bulma.net/body.phtml?nIdNoticia=1743