

Conexiones VPN con PPTP bajo Linux

Índice

Redes privadas virtuales y pptp
Requisitos
Instalación y configuración de pptp
Depurando errores
Configuración de clientes pptp
Bibliografía y referencias

Por Jordi Ivars Oller
Mayo del 2004

1.- Redes privadas virtuales y pptp.

Una red privada virtual consiste en dos máquinas (una en cada "extremo" de la conexión) y una ruta o "túnel" que se crea dinámicamente en una red pública o privada. Para asegurar la privacidad de esta conexión los datos transmitidos entre ambos ordenadores son encriptados por el Point-to-Point protocol (PPP), un protocolo de acceso remoto, y posteriormente enrutados o encaminados sobre una conexión previa (también remota, LAN o WAN) por un dispositivo PPTP. En el entorno Windows 95 o NT 4.0, este dispositivo se conoce como adaptador de red privada virtual.

El PPTP es un protocolo de red que permite el tráfico seguro de datos desde un cliente remoto a un servidor corporativo privado, estableciéndose así una Red Privada Virtual (VPN) basada en TCP/IP. PPTP soporta múltiples protocolos de red (IP, IPX y NetBEUI) y puede ser utilizado para establecer dichas redes virtuales a través de otras redes públicas o privadas como líneas telefónicas, redes de área local o extensa (LAN's y WAN's) e Internet u otras redes públicas basadas en TCP/IP.

El escenario más común para el funcionamiento de una vpn sobre protocolo pptp es el siguiente: Una máquina cliente (Windows 9x, XP o Windows 2000) se conecta a un Proveedor de Servicios de Internet (ISP) utilizando una conexión de acceso telefónico a redes o bien usando una típica conexión ADSL o cablemodem.

En otro punto de Internet existe una máquina (por ejemplo, un servidor Linux) ofreciendo servicio de conexión pptp. El cliente puede en ese momento lanzar una conexión de acceso remoto usando su adaptador de red privada virtual a dicho servidor, creándose un túnel privado sobre Internet que conecta ambas máquinas como si estuvieran en la misma red local, pudiendo así tener acceso a recursos compartidos tales como carpetas o impresoras.

Este escenario puede variar según el tipo de conexión que tengan tanto el cliente como el servidor. Organizaciones con acceso permanente a Internet, pueden configurar servidores de acceso remoto para que soporten PPTP. Esto permite que colaboradores en cualquier parte del mundo puedan conectarse a ellos, usando sus accesos a Internet habituales. Así, será posible participar de los recursos de la red corporativa y con seguridad garantizada gracias a los sistemas de encriptación de 128 bits.

2.-Requisitos.

Para configurar una vpn bajo protocolo pptp necesitamos un servidor, un cliente (tantos como deseemos) y una conexión a Internet para cada uno de ellos.

El servidor puede ser una distribución cualquiera de Linux configurada con el servicio Poptop (www.poptop.org), que es el encargado de ofrecer conexiones PPTP. No necesita grandes requisitos de hardware y con un simple Pentium 300 con 128 megas de RAM podremos ofrecer conexiones a la vpn a un máximo de 15 a 20 clientes.

El cliente puede ser una máquina con cualquier Windows, desde Windows 98 hasta Windows 2003, pasando por XP o Windows 2000. En todo caso, se recomienda usar Windows 2000 o XP ya que su soporte de compresión de tráfico y su mayor nivel de encriptación ofrecerán un mejor rendimiento al conectarse a la vpn.

En cuanto a los protocolos de red necesarios, PPTP permite el uso de redes privadas virtuales sobre redes TCP/IP ya existentes, como Internet, pero preservando el uso de los protocolos de red, direcciones de nodo y nombres de máquinas ya existentes en la red privada. Por tanto, no se requiere el uso de nuevos protocolos ni cambios en las aplicaciones de red.

Mediante el "túnel privado" que se establece a través de la red pública TCP/IP, pueden usarse, por ejemplo, los protocolos IPX y NetBEUI usados en la red privada para la ejecución de las aplicaciones de red.

Por otro lado, los métodos de resolución de nombres de la red privada (WINS, DNS, SAP) no necesitan modificarse. Además, en el caso de las direcciones IP, pueden usarse para la red privada direcciones no válidas en Internet. Por ejemplo, una organización puede usar internamente un conjunto de direcciones IP de clase B sin preocuparse de que dichas direcciones ya estén siendo utilizadas por otra en Internet. Sin embargo, tanto el servidor PPTP como el cliente tienen que tener asignadas direcciones IP válidas en Internet antes de poder establecer el "túnel". En ese momento cada uno tendrá dos direcciones IP, una válida en Internet y otra válida en la red privada. PPTP se encargará de "encapsular" los paquetes IP con destino a la red privada dentro de otro paquete que viajará por la red pública (Internet) hasta el servidor PPTP. Éste extraerá el paquete IP con destino a la red privada y se lo transmitirá.

En cuanto a las garantías de seguridad que ofrece PPTP, la autenticación de usuarios se realiza a través de los protocolos PAP y CHAP). PPTP hace uso de la seguridad que da el protocolo PPP. La autenticación MS-CHAP se utiliza para validar las credenciales del usuario remoto contra dominios NT a través del servidor Linux con PPTP, capaz de emular cualquier tipo de conexión y protocolo desde y hacia sistemas Windows.

Sólo los usuarios con permiso para ello pueden realizar la conexión. Una vez que se comprueba que el usuario tiene permiso (es decir, que su nombre de usuario y contraseñas existen) para iniciar una sesión, se genera una "llave" de 40 bits en Windows 98 y NT y de 128 bits en XP y Windows 2000 a partir de la clave del usuario (llave que SIEMPRE viaja ya encriptada por la red) que es utilizada para encriptar a su vez los datos del usuario (encriptación RC-4).

3.-Instalación y configuración de pptp en Linux.

En Linux tenemos soporte para vpns bajo protocolo pptp gracias al proyecto Poptop, encargado de la creación del programa pptpd que es el que usaremos para crear nuestra vpn. Básicamente y con el uso de este programa, conseguiremos que los clientes Windows accedan a nuestra red local asignándoles un interfaz de red nuevo en el servidor pptp (un interfaz que realmente es virtual, no va asociado a ningún hardware en especial) con una dirección IP capaz de llegar a nuestra redes locales. El protocolo pptp, al estar basado en conexiones punto a punto con protocolo ppp, creará en nuestro servidor sencillos interfaces ppp. A cada conexión pptp nueva tendremos un nuevo interfaz ppp, empezando desde el **ppp0** y subiendo a 1, 2, 3, etc a cada nueva conexión simultánea que obtengamos.

Para la instalación del soporte pptp para Linux tenemos dos opciones, o bien usar el soporte pptp simple sin encriptación, con el cual nos ahorramos tener que modificar nuestro kernel (el núcleo del sistema) para añadirle soporte de encriptación MPPE o bien optar por acabar modificando nuestro kernel para añadirle dicho soporte, con lo que obtendremos un sistema mucho más seguro al estar protegido por una potente encriptación de hasta 128 bits.

En el primero de los casos, y para empezar con el segundo, necesitaremos el paquete que nos proporciona el sistema de vpn por pptp que podremos obtener desde la web del proyecto Poptop (www.poptop.org) o en otros casos obtener el paquete desde nuestra propia distribución, que normalmente recibirá el nombre de pptpd (como por ejemplo en Debian o RedHat, ya sea en formato .deb o en formato .rpm, dependiendo de nuestra distribución).

Su instalación es sencilla y de configuración muy fácil, ya sea instalando el paquete de nuestra distribución o compilándolo nosotros mismos.

Tendremos tres archivos básicos que tendremos que controlar para poder poner en marcha nuestra vpn con pptp. Estos tres archivos son:

/etc/pptpd.conf
/etc/ppp/pptpd-options

/etc/ppp/chap-secrets

Por otro lado, tenemos el archivo **/etc/init.d/pptpd** con el cual, pasándole la orden **stop** o **start** podremos arrancar o parar nuestra vpn.

Veamos el primero de los archivos de configuración, el **/etc/pptpd.conf**:

```
#####  
#  
# Sample PoPToP configuration file  
#  
# for PoPToP version 0.9.12  
#  
#####  
  
# TAG: speed  
#  
# Specifies the speed for the PPP daemon to talk at.  
#  
speed 115200  
  
# TAG: option  
#  
# Specifies the location of the PPP options file.  
# By default PPP looks in '/etc/ppp/options'  
#  
  
option /etc/ppp/pptpd-options  
  
# TAG: debug  
#  
# Turns on (more) debugging to syslog  
#  
  
#debug  
  
# TAG: localip  
# TAG: remoteip  
#  
# Specifies the local and remote IP address ranges.  
#  
# You can specify single IP addresses seperated by commas or you can  
# specify ranges, or both. For example:  
#  
# 192.168.0.234,192.168.0.245-249,192.168.0.254  
#  
# IMPORTANT RESTRICTIONS:  
#  
# 1. No spaces are permitted between commas or within addresses.  
#  
# 2. If you give more IP addresses than MAX_CONNECTIONS, it will  
# start at the beginning of the list and go until it gets  
# MAX_CONNECTIONS IPs. Others will be ignored.  
#  
# 3. No shortcuts in ranges! ie. 234-8 does not mean 234 to 238,
```

```
# you must type 234-238 if you mean this.
#
# 4. If you give a single localIP, that's ok - all local IPs will
# be set to the given one. You MUST still give at least one remote
# IP for each simultaneous client.
#
```

```
localip 192.168.2.1
remoteip 192.168.5.1-100
```

Este archivo tiene muy pocas opciones de configuración que necesitemos considerar. Las opción **speed** no tiene ninguna utilidad si nuestro servidor va a funcionar sobre redes ethernet. Pongamos lo que pongamos, siempre irá a la máxima velocidad que nuestra red permita.

La opción **debug** nos permitirá ver en los logs todos y cada uno de los pasos y procesos que realiza nuestro servidor pptp para establecer las conexiones de los clientes. Tenerla activada puede ser una buena ayuda cuando nuestro servidor no funciona y no sabemos porque, ya que obtendremos mucha más información de los logs del sistema y podremos buscar entre ellos donde estamos fallando.

La opción **option** especifica el archivo de opciones de conexión (donde especificamos encriptación, servidores dns, etc), que por defecto es /etc/ppp/pptpd-options y que en principio no necesitamos cambiar para nada.

Las opciones **localip** y **remoteip** si que tienen bastante más importancia ya con ellas especificaremos los rangos ip de nuestra vpn. En **localip** bastará con especificar una de las IPs locales que pueda tener nuestro servidor pptp (si solo tiene una, pues esa misma servirá).

En **remoteip** tendremos que especificar el rango de direcciones IP que asignaremos a aquellos clientes que se conecten a nuestra red via pptp. Podemos usar direcciones de un rango de red ya creado (obviamente, deben ser direcciones IP que nadie esté utilizando) o bien crear un rango de red propio que solamente utilizarán las máquinas que se conecten usando la vpn y que se comunicará con el resto de nuestros rangos locales gracias al mismo servidor pptp, que automáticamente enrutará las máquinas que entren vpn hacia todos los rangos de red que el mismo servidor pptp sea capaz de encontrar en su misma red local.

Podemos especificar grupos de IPs, para no agotar todo el rango. Por ejemplo, **remoteip 192.168.5.1-100** solo asignará ips entre 192.168.5.1 hasta la 192.168.5.100.

El siguiente archivo que veremos es el **/etc/ppp/pptpd-options**:

```
#####
## turn pppd syslog debugging on
debug

## change 'servername' to whatever you specify as your server name in
chap-#secrets

# name servername

## change the domainname to your local domain

domain midominio

## these are reasonable defaults for WinXXXX clients
## for the security related settings

auth
#require-chap
```

```
#require-chapms
#require-chapms-v2
#+chap
## Fill in your addresses
```

```
ms-dns 192.168.2.2.252
ms-wins 292.168.2.250
```

```
## Fill in your netmask
```

```
netmask 255.255.255.0
```

```
## some defaults
```

```
nodefaultroute
#defaultroute
proxyarp
lock
```

Aquí las opciones que encontramos ya son bastante más extensas. Básicamente, en este archivo definimos cosas como el tipo de autenticación (**chap**, **require-chap**, **require-chapms**, etc) o simplemente si queremos autenticación (**auth**), si queremos ampliar el nivel de registro de la vpn en los logs al establecer el interfaz ppp (**debug**). Básicamente todas estas opciones, aquellas que necesitemos para un funcionamiento básicos, ya vienen activadas y los cambios que necesitaremos realizar en este archivo son mínimos.

Opciones a tener en cuenta son **ms-dns** y **ms-wins**, donde estableceremos, si los tenemos, los servidores dns y wins de nuestra red local para que las máquinas que entran a través de la vpn sean capaces de encontrar los nombres de los ordenadores de nuestra red local. La opción **domain** nos permite especificar el dominio Windows de nuestra red local, siempre y cuando lo tengamos.

Opciones como **nodefaultroute** o **proxyarp** nos permiten adoptar o no las rutas por defecto del servidor al que nos conectamos o la realización o no de un proxy arp, opción esta necesaria si estamos creando nuestra vpn en un segmento de red distinto al de nuestra propia red local.

En el caso de querer activar la encriptación, esta dependerá del cliente. Sin especificar opción alguna en el servidor, este aceptará el nivel de encriptación o no dependiendo de lo que quiera hacer el cliente, sin obligación alguna para que encripte sus datos o no. Si queremos obligar a que el cliente encripte sus conexiones, podríamos añadir las opciones:

mppe required, obligamos a usar cualquier tipo de encriptación al cliente
mppe no40, desactivamos el uso de llaves de 40 bits
mppe no56, desactivamos el uso de llaves de 56 bits
mppe no128, desactivamos el uso de llaves de 128 bits

Si tan solo queremos que encripten datos a 128, activaremos las tres primeras opciones, con lo que obligaremos a encriptar a 128. Hay que tener en cuenta, de todos modos, que si son Windows 98 o inferiores los que se conectarán a nuestra vpn, en principio estos encriptan tan solo con llaves de 40 bits.

Pasemos al siguiente archivo, el **/etc/ppp/chap-secrets**:

```
# Secrets for authentication using CHAP
# client      server secret          IP addresses
usuario1     *      pLy5r312                *
```

```
usuario2 * aPdA1e3Y *
usuariotest * test 192.168.5.7
```

Este archivo es básicamente para crear los nombres y contraseñas de los usuarios de la vpn, es decir, de aquellos usuarios que podrán acceder a nuestra red local a través de nuestro servidor pptpd.

Básicamente, pondremos el nombre del usuario en la primera columna de la izquierda y su contraseña en la tercera columna. Esta contraseña está en texto plano, por lo que los permisos de este archivo **chap-secrets** deben ser continuamente vigilados para que solamente root pueda ver su contenido. Finalmente, si deseamos que un usuario en concreto siempre que se conecte a la vpn lo haga con la misma ip, podemos añadir la ip que deseamos que use en la última de las columnas, teniendo en cuenta que esa ip ha de entrar dentro del rango que ya definimos en el archivo **pptpd.conf**.

Esta sería la configuración básica de nuestro servidor pptp y con esta ya podríamos ofrecer acceso a nuestro servidor para cualquier cliente que soportara dicho protocolo. Pero aun queda más, ya que si queremos que los clientes puedan encriptar sus comunicaciones con nosotros, deberemos modificar el kernel de nuestro Linux ya que, por defecto, este no ofrece soporte para el protocolo de encriptación de Microsoft (MPPE) ni para el protocolo de compresión de Microsoft (MPPC), con lo cual se lo tendremos que añadir nosotros a mano.

Para ello necesitamos dos cosas, el código fuente de un núcleo Linux, que podremos obtener en www.kernel.org (cualquier versión 2.4.x nos valdrá, aunque cuanto más moderna sea mejor) y los parches para añadir soporte a los protocolos antes mencionados, que podremos obtener de www.polbox.com/h/hs001. De esta misma web podremos obtener también el código fuente y los parches necesarios para el ppp, que por defecto tampoco ofrece soporte de encriptación y que también tendremos que modificar para poder usar un pptp con soporte encriptado. Pero empezemos con el kernel. Si desconoces el proceso de configuración y compilación del kernel sería adecuado que antes de hacer nada y acabar con el sistema roto te documentaras sobre estos procesos, leyendo documentos como el Kernel-Como (<http://www.insflug.org/COMOs/Kernel-Como/Kernel-Como.html>), Como compilar el Kernel (http://aula.linux.org.ar/docs/principiantes/comp_kernel/comp_kernel.htm) o Configurar y compilar el Kernel (<http://www.frikis.org/staticpages/index.php?page=kernel>).

Una vez descargado nuestro kernel de la web citada anteriormente, lo descomprimiremos en `/usr/src`, que es donde solemos descomprimir las fuentes de nuestros kernels.

Una vez descomprimido el kernel descargaremos los parches para el mismo, guardándolos también en el directorio `/usr/src` y los aplicaremos al kernel de la siguiente manera:

```
gzip -cd linux-2.4.25-mppe-mppc-0.99.patch.gz ; patch -p0 (el nombre del parche variará dependiendo de la versión del kernel para el cual lo descarguemos)
```

Una vez aplicados los parches, configuraremos el kernel con las opciones referentes al protocolo ppp (en **Network Options**), donde además de las habituales tendremos las opciones referentes a MPPE/MPPC que también activaremos.

Con el kernel ya compilado, tendremos que configurar nuestro sistema para que cargue el módulo MPPE/MPPC cuando se necesite, tarea que podremos hacer modificando el archivo `/etc/modules.conf`, al que añadiremos lo siguiente:

```
alias ppp-compress-18 ppp_mppe_mppc
```

Dependiendo de nuestra distribución puede que el archivo `modules.conf` no sea el lugar adecuado para añadir lo anterior. Por ejemplo, el `modules.conf` de Debian es dinámico y se genera con los

datos encontrados en `/etc/modutils/`, con lo que si modificamos el `modules.conf` este perderá los datos que escribamos al reiniciar la máquina. En Debian, lo añadiríamos en `/etc/modutils/ppp`. Con el kernel compilado y ya en funcionamiento podremos pasar al siguiente paso que es adaptar el programa `ppp`, encargado de crear las interfaces punto a punto (`ppp0` y demás) y establecer las conexiones `vpn` (`pptp` sería el encargado de negociarlas, aunque es `ppp` quien realmente permite la conexión punto a punto), a los nuevos módulos de encriptación y compresión de datos que hemos añadido al kernel. Como ya antes habríamos descargado el `ppp` y el parche para el mismo de la misma web en la que encontramos los parches para el kernel, podremos pasar a compilarlo.

Para hacerlo lo descomprimiremos en cualquier directorio, descomprimiremos el parche y lo aplicaremos de la siguiente manera:

```
patch -p0 < ppp-2.4.2-mppe-mppc-0.82.patch
```

Luego tan solo tendremos que configurar, compilar e instalar el `ppp` de la forma acostumbrada y nuestro `ppp` será capaz de entenderse a la perfección con cualquier conexión `pptp` encriptada que podamos recibir.

Con esto quedaria finalizada la configuración de nuestro servidor `pptp` con Linux, ya sea sin o con soporte de encriptación y compresión de datos, aspectos estos últimos muy importantes para mejorar tanto la seguridad como el rendimiento de nuestras conexiones `vpn`.

4.-Depurando errores.

¿Que ocurre si algo funciona mal? ¿Donde podemos encontrar que y porqué está fallando nuestro servicio `pptpd`? La solución es mirar los logs, los registros en los que el `pptpd` guarda acción que realiza, ya sea esta satisfactoria o no. Para ello lo mas adecuado es activar las opciones **debug** tanto en el `pptpd.conf` como en `pptpd-options` y disponernos a observar todo lo que aparezca en los logs. Generalmente dichos logs los encontraremos en `/var/log/syslog` (en Debian) o, mas comunmente en otras distribuciones, en `/var/log/messages`.

Un registro típico de establecimiento de conexión seria algo como lo que sigue:

```
May 5 10:21:28 firewall pptpd[19912]: CTRL: Client 213.96.74.18 control connection started
May 5 10:21:28 firewall pptpd[19912]: CTRL: Starting call (launching pppd, opening GRE)
May 5 10:21:28 firewall pppd[19913]: pppd 2.4.2b3 started by root, uid 0
May 5 10:21:28 firewall pppd[19913]: using channel 50
May 5 10:21:28 firewall pppd[19913]: Using interface ppp0
May 5 10:21:28 firewall pppd[19913]: Connect: ppp0 <--> /dev/pts/1
May 5 10:21:33 firewall pppd[19913]: MPPC/MPPE 128-bit stateless compression enabled
May 5 10:21:33 firewall pppd[19913]: found interface eth2 for proxy arp
May 5 10:21:33 firewall pppd[19913]: local IP address 200.1.2.57
May 5 10:21:33 firewall pppd[19913]: remote IP address 192.168.5.3
```

Este registro seria el típico, aquel que veriamos sin tener las opciones **debug** activadas. Con estas activadas, a parte de lo anterior, veriamos líneas similares a las siguientes:

```
May 5 10:21:28 firewall pppd[19913]: sent [LCP ConfReq id=0x1 <asyncmap 0x0> <auth eap>
<magic 0x70732ce4> <pcomp> <accomp>]
May 5 10:21:28 firewall pppd[19913]: rcvd [LCP ConfReq id=0x0 <mru 1400> <magic
0x6e652842> <pcomp> <accomp> <callback CBCP>]
May 5 10:21:28 firewall pppd[19913]: sent [LCP ConfRej id=0x0 <callback CBCP>]
May 5 10:21:28 firewall pppd[19913]: rcvd [LCP ConfReq id=0x1 <mru 1400> <magic
0x6e652842> <pcomp> <accomp>]
```

```

May 5 10:21:28 firewall pppd[19913]: sent [LCP ConfAck id=0x1 <mru 1400> <magic
0x6e652842> <pcomp> <accomp>]
May 5 10:21:31 firewall pppd[19913]: sent [LCP ConfReq id=0x1 <asynctest 0x0> <auth eap>
<magic 0x70732ce4> <pcomp> <accomp>]
May 5 10:21:31 firewall pppd[19913]: rcvd [LCP ConfNak id=0x1 <auth chap MS-v2>]
May 5 10:21:31 firewall pppd[19913]: sent [LCP ConfReq id=0x2 <asynctest 0x0> <auth chap
MS-v2> <magic 0x70732ce4> <pcomp> <accomp>]
May 5 10:21:31 firewall pppd[19913]: rcvd [LCP ConfAck id=0x2 <asynctest 0x0> <auth chap
MS-v2> <magic 0x70732ce4> <pcomp> <accomp>]
May 5 10:21:31 firewall pppd[19913]: sent [CHAP Challenge id=0xbe
<2188039659e5ac45baf40c3f5d0cf66a>, name = "pptp.servidor.com"]
May 5 10:21:31 firewall pptpd[19912]: CTRL: Ignored a SET LINK INFO packet with real
ACCMs!
May 5 10:21:31 firewall pppd[19913]: rcvd [LCP code=0xc id=0x2 6e 65 28 42 4d 53 52 41 53
56 35 2e 31 30]
May 5 10:21:31 firewall pppd[19913]: sent [LCP CodeRej id=0x3 0c 02 00 12 6e 65 28 42 4d
53 52 41 53 56 35 2e 31 30]
May 5 10:21:31 firewall pppd[19913]: rcvd [LCP code=0xc id=0x3 6e 65 28 42 4d 53 52 41 53
2d 31 2d 56 49 43 54 4f 52]
May 5 10:21:31 firewall pppd[19913]: sent [LCP CodeRej id=0x4 0c 03 00 16 6e 65 28 42 4d
53 52 41 53 2d 31 2d 56 49 43 54 4f 52]
May 5 10:21:32 firewall pppd[19913]: rcvd [CHAP Response id=0xbe
<46f117a744fab601b0c665accde72b6000000000000000049a057f846881bdf82b23282eb423
f00224119aea3360c9b00>, name = "prueba"]

```

En este caso ya no parece tan claro. En los logs sin debug se ve el intento de conexión, el interfaz ppp que se asigna, el nivel de compresión utilizado y finalmente la ip asignada a la conexión.

En el segundo caso, a parte de los mismos datos que en el primero, veriamos toda una serie de datos que en principio pueden sonarnos a chino, pero que muestran cosas interesantes, como por ejemplo el usuario que ha establecido la conexión (la última línea), el tipo de autenticación (chap MS-v2 en este caso), los servidores wins y dns que se asignan (no aparecen aquí, estos son unos logs "resumidos") y bastantes cosas mas. Lo interesante de estos logs ya no es tanto entender o no la información que dan sino que esta información nos servirá para buscar cualquier error producido. El mensaje de error que nos pueda salir será una fuente de búsqueda de información de inestimable valor. Ese mensaje y buscadores como Google (www.google.com) o Buscadoc (www.buscadoc.org) nos darán casi de forma segura una respuesta a que hemos hecho mal.

http://poptop.sourceforge.net/dox/pptp_win9x_me/

5.-Configuración de los clientes pptp

En principio cualquier sistema operativo con soporte de protocolo pptp será capaz de conectarse a nuestro pptpd con Linux pero, normalmente, los que se conecten serán clientes con Windows en cualquiera de sus distintas versiones.

La configuración de Windows para que se conecte a una vpn via pptp es muy sencilla y no requiere de ningún cliente adicional (como seria el caso si necesitásemos conectar un Windows a una vpn ipsec).

Solamente tendremos que crear una nueva conexión de acceso a redes y seleccionar una conexión a redes privadas virtuales.

Pero realmente dependiendo de cada Windows la configuración cambiará en pequeños detalles por lo que para cada modelo de Windows tendremos que tener en cuenta sus especificidades.

Hay disponibles diversos documentos para configurar clientes pptp en Windows que nos servirán de largo para tener diversos ejemplos de configuración para nuestros sistemas. Los podemos encontrar en las siguientes direcciones:

Para Windows 9x::

http://poptop.sourceforge.net/dox/pptp_win9x_me/

Para Windows NT4:

http://poptop.sourceforge.net/dox/pptp_winnt4/

Para Windows 2000:

http://poptop.sourceforge.net/dox/pptp_win2k/

Para Windows XP:

http://poptop.sourceforge.net/dox/pptp_winxp/VPN_Verbindungsaufbau_mittels_PPTP.pdf

Para Linux:

<http://pptpclient.sourceforge.net/>

Con esta serie de documentos, explicados paso a paso para cada caso, no tendremos problema alguno para configurar nuestros clientes. En todo caso, si queremos usar soporte de encriptación para llaves de 128 bits en sistemas Windows 95, Windows 98 y Windows 98SE podemos recurrir a los parches que Microsoft ha publicado para solucionar tales casos. En la página web de Microsoft podremos encontrar dichos parches.

6.-Bibliografía y referencias.

<http://www.poptop.org>

<http://www.polbox.com/h/hs001/>

<http://bulma.net/impresion.phtml?nIdNoticia=1743>

<http://www.argo.es/~jcea/artic/hispasec17.htm>

<http://www.telecentros.info/pdfs/pptp.pdf>

<http://www.disc.ua.es/es/asignaturas/rc/inginf/labvirtual/manualpptp.html>

<http://www.google.es/search?q=pptp>

<http://www.buscadoc.org/cgi-bin/s.cgi?q=pptp>

Este documento fue redactado entre el 3 y el 5 de Mayo del 2004 por Jordi Ivars Oller.
Este documento no está sujeto a ninguna licencia ni copyright. Está permitida su libre difusión y/o modificación haciendo referencia tansolo al nombre de su autor.