Informe Técnico - Technical Report
DPTOIA-IT-2006/004
septiembre, 2006

Redes Privadas Virtuales

Jesús Fernández Hernández
José Luis Alonso Berrocal
Carlos G.-Figuerola Paniagua
Ángel F. Zazo Rodríguez

Departamento de Informática y Automática
Universidad de Salamanca

Revisado por:

Dr. Ángel Luis Sánchez Lázaro
Dra. Vivian Félix López Batista

Aprobado en el Consejo de Departamento de 28 de septiembre de 2006

Información de los Autores:

D. Jesús Fernández Hernández
Estudiante del Programa de Doctorado en Fuentes de información en seguridad en internet
Profesor de Secundaria de Sistemas Electrónicos
Profesor Asociado en el Departamento de Física Aplicada - Universidad de Salamanca
j.f.h@usal.es

Dr. José Luis Alonso Berrocal
Área de Lenguajes y Sistema Informáticos. Departamento de Informática y Automática
Facultad de Traducción y Documentación - Universidad de Salamanca
C/ Francisco Vitoria, 6-16. 37008 - Salamanca
berrocal@usal.es

Dr. Carlos G.-Figuerola Paniagua
Área de Lenguajes y Sistema Informáticos. Departamento de Informática y Automática
Facultad de Traducción y Documentación - Universidad de Salamanca
C/ Francisco Vitoria, 6-16. 37008 - Salamanca
figue@usal.es

Dr. Ángel F. Zazo Rodríguez,
Área de Lenguajes y Sistema Informáticos. Departamento de Informática y Automática
Facultad de Traducción y Documentación - Universidad de Salamanca
C/ Francisco Vitoria, 6-16. 37008 - Salamanca
afzazo@usal.es

Este documento puede ser libremente distribuido.
(c) 2006 Departamento de Informática y Automática - Universidad de Salamanca.

Resumen

En este documento se recoge los motivos y las razones por las cuales las Redes Privadas Virtuales están implantándose cada día con más fuerza en el ámbito de la comunicación de datos y la seguridad informática. También se razona y motiva la elección de un tipo concreto de Red Privada Virtual (OpenVPN), para su implementación y prueba. Se analiza a fondo este tipo de Red Virtual, con los diferentes escenarios en los cuales se puede aplicar, su configuración e instalación. Finalmente se hace un análisis de los resultados obtenidos y de los posibles problemas de seguridad que puede presentar.

Abstract

In this text we compile the grounds and reasons why the Virtual Private Networks are being strongly introduced in the context of data communication and computer security.It is also reasoned out the election of a specific Virtual Private Network (OpenVPN), in order to be implemented and tested. We deeply analyze this kind of Virtual Network, the different scenarios in which it can be applied, as well as its configuration and installation. Finally we analyze the achieved results and the possible security problems that may arise.

Índice

Índice de figuras

1. Introducción

En la actualidad como en el pasado la informática intenta dar soluciones a los problemas que la sociedad, la industria y las compañías le plantean.

En un pasado no muy lejano la prioridad era la necesidad de procesar y almacenar información, con lo que nacieron los equipos informáticos individuales, posteriormente surgió la necesidad no sólo de procesar y almacenar la información sino que además era preciso compartir la información en tiempo real entre distintos equipos informáticos, por ello surgieron las redes locales (LAN).

A medida que las empresas crecían las redes fueron creciendo y se fueron extendiendo en distintos edificios, localidades, regiones y países, nuevamente surgió la necesidad de compartir la información entre las distintas ubicaciones que se encontraban mucho más espaciadas.

Para cubrir esta demanda aparecen las redes de área extensa (WAN) implementándose las interconexiones de muy distinta forma (Red telefónica conmutada, conexión punto a punto, X25, Frame Relay, etc).

Finalmente aparece la red de redes, Internet, y rápidamente surgen aplicaciones que la utilizan como soporte para transmitir la información entre distintos puntos que pueden estar próximos o en extremos opuestos del planeta.

En muchas ocasiones la información a transmitir, es información sensible, y consecuentemente no debe ser accesible a terceros. ¿Cómo podemos transmitir información y que sólo sea accesible al transmisor y al receptor?. Parece claro que los sistemas criptográficos son la solución a este problema.

En la actualidad los sistemas criptográficos están ampliamente estudiados y desarrollados, aunque queda mucho por avanzar en este campo, no impide que se utilicen en la transmisión segura de información. Hay que notar, que aunque se utilicen canales privados de comunicación, la tecnología actual, la información y los conocimientos generales están a disposición de cualquiera, por lo que comunicaciones que aparentemente pueden ser seguras como una conexión punto a punto, pueda ser interceptada por un hacker para utilizarla en su propio provecho o para realizar daños a la empresa atacada.

Si una empresa decide utilizar un sistema criptográfico para enviar información, ¿necesita implementar desde cero dicho sistema? o ¿necesitaría aplicaciones específicas para la transmisión de estos datos? La respuesta es no ya que en la actualidad existen soluciones que permiten la transmisión ciertamente segura de la información. Estas soluciones se conocen como Redes Privadas Virtuales (**VPN** - *Virtual Private Network*).

2. Redes Privadas Virtuales

¿Qué son las Redes Privadas Virtuales?. Como su nombre indica, podemos deducir que son redes de comunicación entre dos puntos (próximos o todo lo alejados que podamos suponer), que queremos que sean privadas, o lo que es lo mismo, que la información que transmite sea privada entre emisor y receptor. El término de virtual se aplica a la *privacidad* y no al término de *red*, ya que lo que se pretende en la mayoría de los casos, es la transmisión a través de un canal público y abierto, con lo cual si tenemos una red privada a través de este canal público, ésta ha de ser virtual.

En definitiva, las definiciones que podemos encontrar del significado e implicación de una red privada virtual, son múltiples [10], que concretaremos en:

"*Una red privada virtual es una implementación o sistema que habilita una comunicación segura a través de un medio inseguro, siendo transparente para el usuario u aplicación que realiza y recibe la comunicación*" [1][15][5][6].

Esta definición que es perfectamente válida, se puede ampliar y podremos decir que siempre que queramos asegurar una comunicación entre dos puntos, podremos utilizar una red privada virtual, independientemente del potencial de privacidad que tengamos en la comunicación, si la información ha de ser preservada de terceros por su importancia.

También en algunas ocasiones se denomina a las redes privadas virtuales como túneles, ya que estas transportan la información por un canal público, pero aislando la información del resto y consecuentemente creando unas "paredes" virtuales que separan nuestra información de la del resto. Estas "paredes" virtuales forman un túnel virtual que impide atravesar la información en cualquiera de los dos sentidos. De ahí el nombre de túnel.

La separación de la información se logra mediante la encriptación de la misma, y el sistema será más seguro, cuanto mayor seguridad nos suministre el sistema criptográfico, siendo deseable que cualquier avance significativo en el campo de la criptografía, pueda ser implementado y transferido a nuestra Red Privada Virtual.

3. Aplicaciones de las Redes Privadas Virtuales

Los escenarios de aplicación de las Redes Privadas Virtuales son varios y van parejos con la evolución histórica de la informática.

Inicialmente la informática nace como consecuencia de la necesidad de cálculo y de procesamiento de datos en un tiempo muy corto (como ejemplo tenemos el cálculo de tiro parabólico de un cañón), donde la información a procesar se alimenta directamente al ordenador, éste la procesa y seguidamente devuelve los resultados. En este caso las necesidades de seguridad son mínimas, ya que la información sólo es accesible a los elementos que la manejan y en ningún instante está expuesta a

elementos externos.

Cuando la informática evoluciona, se comprueba que no es suficiente con procesar la información, sino que además, es preciso compartir esta información entre distintos equipos, así como los recursos que pueden ser muy costosos también se pueden compartir. Nace la Red de Área Local (LAN - Local Area Network). ¿Qué ocurre en este caso cuando tenemos que transmitir información confidencial de un equipo a otro?, la solución pasa por interconectar entre si aquellos equipos que comparten la información confidencial, separando físicamente la red confidencial de la red general. Esto presenta varios problemas, el primero es la duplicidad de recursos de red que se necesitan, ya que tendremos que montar y mantener dos redes. La segunda es que aún así la confidencialidad no se puede considerar segura, puesto que la seguridad viene impuesta por la separación física de las dos redes, pero nada impide que en una determinada parte de la instalación alguien se pueda conectar a dicha red y leer los datos que circulan. La solución pasa por usar la misma red general para transmitir la información confidencial junto a la información abierta. Por consiguiente, aquí nace el primer escenario de aplicación de las Redes Privadas Virtuales, como separación en una Intranet de aquellos departamentos, personas o equipos, que no deban tener acceso a la información confidencial de los que sí la puedan tener.

El siguiente paso de la Informática, se da de la mano de holding empresariales, que tienden a extender sus negocios en varios edificios y que necesitan interconectar entre sí todas sus oficinas. En este caso las soluciones de interconexión pasaban por utilizar líneas punto a punto, X25, Frame Relay, etc conocidas con el nombre de Redes de Área Extensa (WAN - Wide Area Network). En este caso la información se puede considerar hasta cierto punto segura, ya que los conocimientos necesarios para poder acceder a las líneas, no se encuentra al alcance de cualquier persona, además de los equipos necesario para ello. Hasta hace relativamente poco tiempo era el sistema preferido de bancos, oficinas estatales, ejército, etc. Realmente en este tipo de conexiones no es muy necesario la utilización de las VPN.

No obstante el crecimiento empresarial y la globalización, nos lleva a un nuevo escenario, donde las empresas tienen oficinas y sucursales en otras ciudades y en otros países. Al mismo tiempo los agentes móviles de las empresas (trabajadores que se desplazan por la geografía, como servicios técnicos, asesores, inspectores, etc), necesitan también acceder a la información de la oficina central o matriz, pero en este caso aparece un nuevo componente que es Internet, conocido como la Red de Redes, donde todos los ordenadores están conectados entre sí a nivel mundial. Por ser Internet de carácter público, parece lógico pensar que es el sistema más inseguro que podemos encontrar, y en efecto así es, la información que circula por la misma está al alcance de cualquiera, y en este caso podemos encontrarnos además con personal altamente cualificado en la interceptación de información. Pero pese a esta alto riesgo, las empresas están decididas a utilizar este medio, ¿Por qué?, bueno las razones son varias [16]:

- Los costes de las comunicaciones dentro de la empresa suponen un incremento importante en la cuenta de resultados de la misma. Como Internet es un servicio que suele estar implantado en la mayoría de ellas, si se pudiera utilizar para

transmitir la información, se abaratarían los mismos de forma considerable, ya que las conexiones de Internet tiene unos precios muy bajos.

- Dada la penetración de Internet en los distintos países del planeta es muy posible encontrar en la mayoría de la geografía un Proveedor de Servicios (IPS - Internet Service Provider), que permitirá la conexión de los agentes móviles de la empresa con la oficina principal.

- Internet asegura, un encaminamiento o ruta segura desde una localización a otra. Si por alguna circunstancia, la comunicación se corta entre los dos puntos de la comunicación, los nodos de la red de Internet son capaces de buscar caminos alternativos para que la información llegue a su lugar, claro está, siempre y cuando la zona afectada no sea uno de nuestros nodos de acceso.

Esto define el segundo escenario donde pueden aparecer las VPN, que es en las comunicaciones a larga distancia entre oficinas, agentes móviles y la oficina central usando Internet, ya que la encriptación de la información por la misma impide su lectura y/o modificación.

El último escenario donde podemos encontrar la utilización de las VPNs es relativamente reciente y aparece en las conexiones inalámbricas (wireless), dado que estas conexiones están expuestas a la interceptación de cualquiera que se encuentre dentro del alcance de la misma. Y aunque los routers con conexión wireless suelen utilizar algún algoritmo de cifrado WPA (Wi-Fi Protected Access - Acceso Protegido Wi-Fi) y WEP (Wired Equivalent Privacy - Privacidad Equivalente a Cableado); siendo este último sistema muchos menos seguro que el primero (ataque estadístico que permite recuperar la clave WEP). Aun así, si la información es realmente sensible, estos dos sistemas de cifrado pueden ser insuficientes y necesitaríamos recurrir a las VPNs.

Resumiendo los cuatro escenarios posibles son [7]:

1. Redes separadas y seguras en Intranets (VPN Interna).

2. Interconexión por Internet:

   a) Conexión permanente entre oficinas (VPN sitio-a-sitio).

   b) Conexiones aleatorias de los agentes móviles con su oficina (VPN de acceso remoto).

3. Conexiones entre equipos mediante Wireless (VPN Interna).

¿Qué podemos esperar de una VPN? pues básicamente debe de garantizar los siguientes puntos [4] [10]:

- **Confidencialidad o privacidad** Los datos transferidos deben estar disponibles sólo para las personas autorizadas.

- **Confiabilidad o integridad** Los datos transferidos no se deben poder cambiar entre el remitente y el receptor.
- **Disponibilidad** Los datos transferidos deben estar disponibles cuando son necesarios.
- **No Repudio** para impedir que una vez firmado un documento el signatario se retracte o niegue haberlo redactado.

Para cumplir las condiciones anteriores, los paquetes IP que se desean transmitir [1]:

- Se cifran para garantizar la confidencialidad.
- Se firman para garantizar la autenticidad, integridad y no repudio.

El paquete resultante se encapsula en un nuevo paquete IP y se envía a través de la red insegura al otro extremo de la VPN:

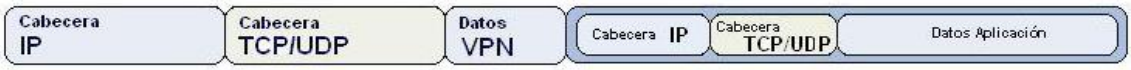

Figura 1: Encapsulación

4. Implementación de las Redes Privadas Virtuales

Como hemos comentado, las redes privadas virtuales, deben de ser transparentes a los usuarios o aplicaciones que las utilizan. ¿Cómo podemos implementar una VPN? [10].

Para responder a esta pregunta, analicemos el sistema que en la actualidad predomina en las comunicaciones, el modelo OSI (Open Systems Interconnection). Dentro de este modelo de interconexión, como podemos observar en la figura 2, sigue un flujo continuo de datos.

El proceso de encriptación y desencriptación de la información se puede realizar en cualquier punto del flujo de la información, sólo con la restricción de realizar los procesos referidos en las mismas capas equivalentes [17].

Por consiguiente, atendiendo al modelo OSI, podemos apreciar en el mismo dos grandes zonas: *Hardware* y *Software*. El término *Hardware* se refiere al sistema de interconexión física de los dos equipos (capa física), mientras que el término *Software* se aplica al resto de capas del modelo OSI. Dado que la encriptación y desencriptación se puede realizar en los puntos que queramos, siempre y cuando sean capas equivalentes, podremos seleccionar estos dos puntos definidos, VPN por *Hardware* y VPN por *Software*.

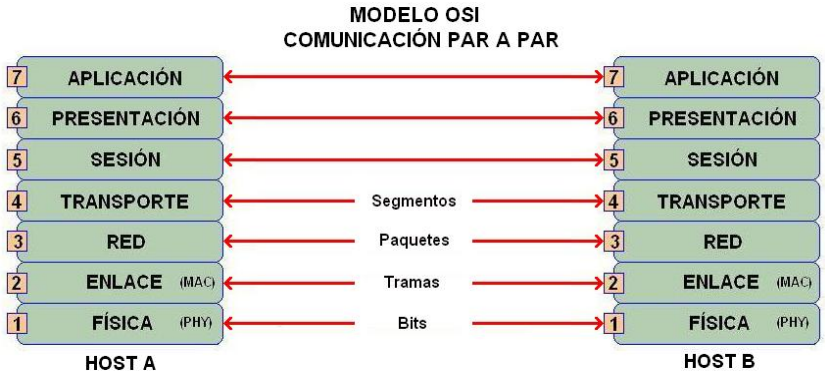

Figura 2: Modelo OSI

4.1. Implementación de VPN por Hardware

El proceso de encriptación y desencriptación se realiza a nivel físico en los puntos inmediatamente anterior e inmediatamente posterior al comienzo de la línea de comunicación. Por realizarse a nivel físico, necesitamos unos equipos que permitan realizar esta tarea de forma transparente. Por lo general los elementos utilizados son los routers con VPN incorporada. Estos dispositivos llevan incorporado un procesador y algoritmos de encriptación y desencriptación. Tienen la ventaja de que el fabricante nos da realizada la implementación y su instalación y uso es extremadamente sencillo, ya que solo tenemos que intercalar los routers en los puntos de salida y entrada de la línea de comunicación y activar en los routers la encriptación-desencriptación, así como configurar la contraseña, certificación o medio que servirá para la encriptación y desencriptación de la información.

Las VPN implementadas por hardware, presentan el inconveniente, de que el sistema de encriptación viene impuesto por el fabricante, y depende del mismo para las actualizaciones.

Dentro de esta categoría se puede incluir los routers wireless con encriptación, bien mediante WPA o/y WEP, ya que crean un túnel entre el router y la tarjeta wireless que impiden en cierta forma la lectura y modificación de la información. En este caso el medio de transporte son las ondas electromagnéticas y por tener el router y la tarjeta inalámbrica la antena, la encriptación se realiza a nivel de capa física.

En [13] podemos encontrar un manual de configuración de uno de estos productos.

Las ventajas e inconvenientes que presenta este tipo de configuración son:

1. Ventajas

   - La instalación y la configuración son relativamente sencillas.
   - No necesita personal especializado y su mantenimiento es mínimo.

- Un único elemento puede habilitar varias VPNs ubicadas en distintos sitios.
- El sistema es independiente de las máquinas conectadas a la red.
- No necesitamos máquinas dedicadas para realizar la VPN.

2. Inconvenientes

- Depende de una tecnología externa y cerrada.
- El firmware de los sistemas es cerrado y dependemos del fabricante para poder cambiarlo.
- Los sistemas de encriptación suelen ser cerrados y el fabricante suele utilizar un único tipo.
- En la mayoría de las ocasiones los elementos hardware de los extremos que componen la red privada virtual, que deben ser iguales o por lo menos del mismo fabricante. No siendo corriente que sean intercambiables por los de otros fabricantes.
- Sólo sirven para realizar conexiones VPN dentro de la misma red (intranet) o sólo fuera de la red, pero no pueden realizar simultáneamente las dos opciones, aunque esto es algo que pudiera cambiar en el futuro.
- La seguridad sólo se implementa desde los dos extremos de la VPN, siendo inseguro el camino que recorre la información desde el ordenador hasta el dispositivo VPN.

4.2. Implementación de VPN por Software

Cada día se está imponiendo más la utilización de Redes Privadas Virtuales por software. La explicación radica en la necesidad que cada vez más tienen los medianos y pequeños usuarios, de implementar sistemas de seguridad en el acceso a sus máquinas. Como además son sistemas que tienden a crecer de forma rápida, es mucho mas barato la utilización de Redes Privadas Virtuales por software que por hardware.

Las ventajas y desventajas que pueden presentar este tipo de redes son:

1. Ventajas:

- Existe una gran variedad de Redes Privadas Virtuales desarrolladas por software, donde elegir y que están continuamente mejorando sus prestaciones.
- El número de usuarios de este tipo de red es mucho mayor que el número de usuarios de VPNs realizadas por hardware, con lo que la posibilidad de encontrar documentación y ayuda para estos elementos es mayor.
- Pueden dar cobertura tanto a redes internas (intranet) como redes externas.

- La seguridad puede cubrir de máquina a máquina, donde se encuentren colocados los extremos de la VPN.

2. Inconvenientes:

   - Es necesario instalar el software en una máquina, pudiendo ser necesario, si la carga de información es muy grande, tener que dedicar una máquina para este menester.
   - El sistema de claves y certificados están en máquinas potencialmente inseguras, que pueden ser atacadas.
   - Si el software es de libre distribución, éste puede estar modificado y contener puertas traseras.

En el presente trabajo estudiaremos las Redes Privadas Virtuales por software, dadas las ventajas que presentan frente a sus homónimas realizadas por hardware.

5. Redes Privadas Virtuales por Software

Como se ha comentado con anterioridad, existe un amplio abanico de implementaciones de Redes Privadas Virtuales por Software. Pasaremos a describir las más utilizadas con sus ventajas e inconvenientes, para finalmente elegir aquella que mas seguridad, fiabilidad y ventajas presente frente a las otras.

5.1. Redes Privadas Virtuales por Software más habituales

De todos los tipos disponibles, podemos citar por ser las más utilizadas:

- IPSec
- PPTP
- L2TP
- VPNs SSL/TLS
- OpenVPN

5.1.1. IPSec

Es la abreviatura de **Internet Protocol Security**. [18][19] Inicialmente se desarrolló para usarse con el estandar IPv6 y posteriormente se adaptó a IPv4. Es una extensión al protocolo IP. Añade los servicios de autenticación y cifrado. IPSec actúa dentro del modelo OSI en la capa 3 (capa de red). No está ligado a ningún

algoritmo de encriptación o autenticación, tecnología de claves o algoritmos de seguridad específico. De hecho es un estandar que permite que cualquier algoritmo nuevo se pueda introducir. Por sus características es considerado como el protocolo estándar para la construcción de redes privadas virtuales.

[15] La especificación del protocolo se encuentra en la RFC 2401 [12]. IPSec cuenta con dos protocolos diferentes, de forma que se empleará uno u otro en función de lo que nos interese proteger y el modo en que realicemos las comunicaciones.

- **Cabecera de Autenticación (Authentication Header, AH)**. Se trata de una nueva cabecera que obtenemos de la básica IP y que se añade a los resúmenes criptográficos ("hash") de los datos e información de identificación.
- **Encapsulado de Seguridad (Encapsulating Security Payload, ESP)**. Permite reescribir los datos en modo cifrado. No considera los campos de la cabecera IP por lo que sólo garantiza la integridad de los datos.

Ambos protocolos controlan el acceso y distribuyen las claves criptográficas. No pueden ser aplicados los dos a la vez. Lo que sí se permite es aplicarlos uno después de otro, es decir, a un datagrama IP aplicarle un protocolo y al paquete resultante aplicarle otro. Si se hace esto el orden de aplicación es: ESP-AH Cada uno de estos protocolos pueden funcionar en dos modos distintos:

- modo transporte.
- modo túnel.

El **modo transporte** [14] es el que usa un anfitrión que genera los paquetes. En modo transporte, las cabeceras de seguridad se añaden antes que las cabeceras de la capa de transporte (TCP, UDP), antes de que la cabecera IP sea añadida al paquete.

El **modo Túnel** se usa cuando la cabecera IP extremo-a-extremo ya ha sido adjuntada al paquete, y uno de los extremos de la conexión segura es solamente una pasarela.

5.1.2. PPTP

PPTP **(Point to Point Tunneling Protocol)** [18] es un protocolo desarrollado por Microsoft, U.S. Robotics, Ascend Communications, 3Com/Primary Access, ECI Telematics conocidas colectivamente como PPTP Forum, para implementar redes privadas virtuales o VPN.

La seguridad de PPTP ha sido completamente rota y las instalaciones con PPTP deberían ser retiradas o actualizadas a otra tecnología de VPN. La utilidad ASLEAP puede obtener claves de sesiones PPTP y descifrar el tráfico de la VPN. Los ataques

a PPTP no pueden ser detectados por el cliente o el servidor porque el exploit es pasivo.

El fallo de PPTP es causado por errores de diseño en la criptografía en los protocolos handshake o apretón de manos LEAP de Cisco y MSCHAP-v2 de Microsoft y por las limitaciones de la longitud de la clave en MPPE.

La actualización de PPTP para las plataformas Microsoft viene por parte de L2TP o IPsec. Su adopción es lenta porque PPTP es fácil de configurar, mientras L2TP requiere certificados de clave pública, e IPsec es complejo y poco soportado por plataformas antiguas como Windows 98 y Windows Me.

5.1.3. L2TP

(Layer 2 Tunneling Protocol) [18] L2TP fue diseñado por un grupo de trabajo de IETF como el heredero aparente de los protocolos PPTP y L2F, creado para corregir las deficiencias de estos protocolos y establecerse como un estándar.

El transporte de L2TP está definido para una gran variedad de tipos de paquete, incluyendo X.25, Frame Relay y ATM.

A pesar de que L2TP ofrece un acceso económico, con soporte multiprotocolo y acceso a redes de área local remotas, no presenta unas características criptográficas especialmente robustas. Por ejemplo:

- Sólo se realiza la operación de autenticación entre los puntos finales del túnel, pero no para cada uno de los paquetes que viajan por él. Esto puede dar lugar a suplantaciones de identidad en algún punto interior al túnel.
- Sin comprobación de la integridad de cada paquete, sería posible realizar un ataque de denegación del servicio por medio de mensajes falsos de control que den por acabado el túnel L2TP o la conexión PPP subyacente.
- L2TP no cifra en principio el tráfico de datos de usuario, lo cual puede dar problemas cuando sea importante mantener la confidencialidad de los datos.
- A pesar de que la información contenida en los paquetes PPP puede ser cifrada, este protocolo no dispone de mecanismos para generación automática de claves, o refresco automático de claves. Esto puede hacer que alguien que escuche en la red y descubra una única clave tenga acceso a todos los datos transmitidos.

A causa de estos inconvenientes se tomó la decisión de utilizar los propios protocolos IPSec para proteger los datos que viajan por un túnel L2TP.

L2TP es en realidad una variación de un protocolo de encapsulamiento IP. Un túnel L2TP se crea encapsulando una trama L2TP en un paquete UDP, el cual es encapsulado a su vez en un paquete IP, cuyas direcciones de origen y destino definen los extremos del túnel. Siendo el protocolo de encapsulamiento más externo IP, los protocolos IPSec pueden ser utilizados sobre este paquete, protegiendo así la información que se transporta por el túnel.

5.1.4. VPNs SSL/TLS

SSL/TLS **Secure Sockets Layer/Transport Layer Security** [18] existen pequeñas diferencias entre SSL 3.0 y TLS 1.0, pero el protocolo permanece sustancialmente igual. El término "SSL"según se usa aquí, se aplica a ambos protocolos a menos que el contexto indique lo contrario. SSL proporciona autentificación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía. Habitualmente, sólo el servidor es autentificado (es decir, se garantiza su identidad) mientras que el cliente se mantiene sin autentificar; la autentificación mutua requiere un despliegue de infraestructura de claves públicas (o PKI) para los clientes. Los protocolos permiten a las aplicaciones cliente-servidor comunicarse de una forma diseñada para prevenir escuchas (*eavesdropping*), la falsificación de la identidad del remitente y mantener la integridad del mensaje.

SSL implica una serie de fases básicas:

- Negociar entre las partes el algoritmo que se usará en la comunicación.
- Intercambio de claves públicas y autenticación basada en certificados digitales.
- Encriptación del tráfico basado en cifrado simétrico.

Durante la primera fase, el cliente y el servidor negocian qué algoritmos criptográficos se van a usar. Las implementaciones actuales proporcionan las siguientes opciones:

- Para criptografía de clave pública: RSA, Diffie-Hellman, DSA (Digital Signature Algorithm) o Fortezza.
- Para cifrado simétrico: RC2, RC4, IDEA (International Data Encryption Algorithm), DES (Data Encryption Standard), Triple DES o AES (Advanced Encryption Standard).
- Con funciones hash: MD5 o de la familia SHA.

TLS/SSL poseen una variedad de medidas de seguridad:

- Numerando todos los registros y usando el número de secuencia en el MAC.
- Usando un resumen de mensaje mejorado con una clave (de forma que solo con dicha clave se pueda comprobar el MAC).
- Protección contra varios ataques conocidos (incluidos ataques *man in the middle attack*), como los que implican un degradado del protocolo a versiones previas (por tanto, menos seguras), o conjuntos de cifrados más débiles.
- El mensaje que finaliza el protocolo handshake (Finished) envía un hash de todos los datos intercambiados y vistos por ambas partes.

- La función seudo aleatoria divide los datos de entrada en 2 mitades y las procesa con algoritmos hash diferentes (MD5 y SHA), después realiza sobre ellos una operación XOR. De esta forma se protege a sí mismo de la eventualidad de que alguno de estos algoritmos se revelen vulnerables en el futuro.

SSL se ejecuta en una capa entre los protocolos de aplicación como HTTP, SMTP, NNTP y sobre el protocolo de transporte TCP, que forma parte de la familia de protocolos TCP/IP. Aunque pueda proporcionar seguridad a cualquier protocolo que use conexiones de confianza (tal como TCP), se usa en la mayoría de los casos junto a HTTP para formar HTTPS.

SSL también puede ser usado para tunelar una red completa y crear una red privada virtual (VPN), como en el caso de OpenVPN.

5.1.5. OpenVPN

OpenVPN [18] es una solución de conectividad basada en software: SSL (Secure Sockets Layer) VPN Virtual Private Network (red virtual privada), OpenVPN ofrece conectividad punto-a-punto con validación, jerárquica de usuarios y host conectados remotamente, resulta una muy buena opción en tecnologías Wi-Fi (redes inalámbricas EEI 802.11) y soporta una amplia configuración, entre ellas balanceo de cargas entre otras. Está publicado bajo licencia de código-libre (*Open Source*).

Algunas de sus ventajas son [4]:

- **Implementación de la VPN en la capa 2 y la capa 3 del modelo OSI**: OpenVPN ofrece dos modos básicos, que funcionamiento como la capa 2 o capa 3. Así los túneles de OpenVPN pueden también transportar tramas de Ethernet, los paquetes del IPX, y los paquetes del navegador de la red de Windows (NETBIOS), que son un problema en la mayoría de las otras soluciones de VPN.
- **Protección de sesión con el cortafuego interno**: Una sesión conectada con la oficina central de su compañía con un túnel VPN puede cambiar el setup de su red en su ordenador portátil, para enviar todo su tráfico de la red a través del túnel. Una vez que OpenVPN haya establecido un túnel, el cortafuego central en la oficina central de la compañía puede proteger el ordenador portátil, aún cuando él no sea una máquina local. Solamente un puerto de la red se debe abrir en local para trabajar la sesión. El cortafuego central protege al empleado siempre que él o ella esté conectado a través del VPN.
- **Las conexiones de OpenVPN pueden ser establecidas a través de casi cualquier cortafuego**: Si tienes acceso a Internet y si puedes tener acceso a la Web, los túneles de OpenVPN deben de trabajar.
- **Soporte de Proxy y configuraciones**: OpenVPN tiene soporte de Proxy y se puede configurar para funcionar como un servicio de TCP o de UDP, y como

servidor o cliente. Como servidor, OpenVPN espera simplemente hasta que un cliente solicita una conexión, mientras que como cliente, intenta establecer una conexión según su configuración.

- **Apertura de un solo puerto en el cortafuego para permitir conexiones entrantes**: Desde OpenVPN 2.0, el modo especial del servidor permite conexiones entrantes múltiples en el mismo puerto del TCP o del UDP, mientras que todavía usa diversas configuraciones para cada conexión.
- **Los interfaces virtuales permiten reglas muy específicas del establecimiento de una red y del cortafuego**: Todas las reglas, restricciones, mecanismos de la expedición, y conceptos como NAT se pueden utilizar con los túneles de OpenVPN.
- **Alta flexibilidad con posibilidades extensas de lenguaje interpretado (scripting)**: OpenVPN ofrece numerosos puntos durante la conexión para la ejecución de los scripts individuales. Estos script se pueden utilizar para una gran variedad de propósitos de la autentificación, recuperación en caso de fallos (*failover*) y más.
- **Soporte transparente y alto rendimiento para IPs dinámicas**: Si se usa OpenVPN, no hay necesidad de utilizar más IPs estáticas de cualquier lado del túnel. Ambas puntos finales del túnel pueden tener acceso barato de ADSL con el IPs dinámicas y los usuarios no notarán un cambio del IP de cualquier lado. Las sesiones del Terminal Server de Windows y las sesiones seguras de Shell (SSH) parecerán congeladas solamente por algunos segundos, pero no terminarán y continuarán con la acción solicitada después de una corta pausa.
- **Ningún problema con NAT**: El servidor y los clientes de OpenVPN pueden estar dentro de una red usando solamente direcciones privadas del IP. Cada cortafuego se puede utilizar para enviar el tráfico del túnel al otro punto final del túnel.
- **Instalación simple en cualquier plataforma**: La instalación y el uso son increíblemente simples. Especialmente, si ha intentado instalar IPsec con diversas configuraciones, se apreciará la facilidad de instalación de OpenVPN.
- **Diseño modular**: El diseño modular con un alto grado de simplicidad en seguridad y establecimiento de una red es excepcional. Ninguna otra solución de VPN puede ofrecer la misma gama de posibilidades a este nivel de seguridad.

Además con la versión 2.0 se incorporó las siguientes mejoras:

- **Soporte Multi-cliente**: OpenVPN ofrece un modo de conexión especial, donde proporcionan a los clientes TLS-autenticados al estilo DHCP de IPs en el establecimiento de la red (túnel). De esta manera, varios túneles (hasta 128) pueden comunicarse sobre el mismo puerto del TCP o del UDP. Obviamente, es necesario activar un switch para activar el modo servidor.

- **Opciones de Envío/Recepción**: La configuración de la red de clientes puede ser controlada por el servidor. Después de la configuración correcta del túnel, el servidor puede decir al cliente (Windows y Linux) que utilice una configuración diferente de red instantáneamente.
- **Interfaz de control (Telnet)**: Se ha añadido una interfaz de control via Telnet.
- **El driver y el software de Windows se han mejorado extensamente.**

5.2. Comparación entre OpenVPN y VPN IPSec

Aun cuando IPsec es el estándar de facto, existen muchos argumentos para usar OpenVPN [21]. La tabla siguiente puede darnos argumentos para seleccionar OpenVPN (los puntos precedidos por "+" son ventajas y puntos precedidos por "-" son las desventajas)[4]:

IPsec VPN	**OpenVPN**
+ Es la tecnología estándar de VPN	- Todavía es algo desconocido, no es compatible con IPsec
+ Plataformas de hardware (dispositivos, aplicaciones)	- Solamente se puede instalar en los ordenadores, pero en todos los sistemas operativos. La excepción al párrafo anterior, es cuando tenemos dispositivos, donde está ejecutándose OpenWrt[1] bajo UNIXs y similares
+ Tecnología bien conocida	- Nueva tecnología; todavía creciendo y aumentando
+ Existen muchos GUIs para su administración	- No hay ningún GUI profesional; sin embargo, hay algunos proyectos interesantes y prometedores
- Modificación compleja de la pila del IP	+ Tecnología simple
- Es necesaria una modificación crítica del núcleo	+ Interfaces y paquetes estandarizados de red
- Son necesario privilegios de administrador	+ El software de OpenVPN puede funcionar en el espacio de usuario, y puede ser *chroot-ed*
- Diversas implementaciones de IPsec de diversos fabricantes pueden ser incompatibles	+ Usa tecnologías estandarizadas de cifrado
- Configuración compleja, tecnología compleja	+ Tecnología fácil, bien estructurada, modular, configuración fácil
- Curva grande de aprendizaje para los novatos	+ fácil de aprender, éxito rápido para los novatos

- Son necesarios varios puertos y protocolos en el cortafuego	+ Solamente es necesario un puerto en el cortafuego
- Problemas con direcciones dinámicas en ambos lados	+ DynDNS trabaja enteramente, vuelve a conectar más rápidamente
- Problemas de la seguridad con las tecnologías de IPsec	+SSL/TLS como capa criptográfica estándar industrial
	+ Traffic shaping (*Control de tráfico*)
	+ velocidad (hasta 20 Mbps en una máquina 1Ghz)
	+ Compatibilidad con los cortafuegos y los proxies
	+ Ningunos problemas al realizar NAT (ambos lados pueden estar en las redes de NATed)
	+ Posibilidades de la configuración del viajante (*road warriors*)

Tabla 1: Open VPN Versus IPSec

6. ¿Qué Red Privada Virtual elegir?

De los diversos tipos de redes privadas virtuales analizadas vamos a comentar las dos posibles candidatas. La primera de ellas es la *IPSec*, ¿por qué hemos seleccionado esta red?, la razón es bien simple: se considera un estándar dentro de la industria y está ampliamente difundida e instalada. La segunda candidata es la OpenVPN, en este caso se ha seleccionado por las críticas tan favorables que se encuentran en la literatura consultada [9]. De las dos seleccionadas, nos decantamos por la segunda opción, siendo las razones desfavorables de más peso de la primera las siguientes:

- Proceso de instalación y configuración muy complicados.
- Es necesario recompilación del núcleo en el caso de Linux.
- Una mala configuración puede dar problemas de seguridad (una excepción en la ejecución nos abre una shell con privilegios de root).

En el caso de la OpenVPN las razones que han hecho nos decidamos por ella, son:

[1] Apoyándose en Linux, desarrolla todo un sistema operativo embebido para routers inalámbricos y puntos de acceso como Cisco Linksys WRT54G. OpenWRT puede ejecutarse en un Cisco Linksys con apenas 16MB de memoria y 8MB de flash, y tiene soporte de VLAN, Firewalling, Snort, Asterisk, VoIP, DNS, DHCP, TFTP, NFS, SAMBA, BGP, QoS, NTP, etc, etc.

- Fácil instalación y configuración.
- Es una aplicación de libre distribución (licencia Open Source).
- Es Código abierto.
- El sistema de criptografía se basa en OpenSSL.
- Es multiplataforma corriendo en los sistemas: Linux, Windows 2000/XP and higher, OpenBSD, FreeBSD, NetBSD, Mac OS X, and Solaris y OpenVPN PocketPC port esta en desarrollo [20].

Por todo ello, la opción elegida es **OpenVPN**.

7. Escenarios de aplicación de OpenVPN

¿En qué situaciones podemos aplicar VPN?, ¿Qué escenarios son los más adecuados para su implementación?. Seguidamente daremos cuenta de las situaciones estudiadas.

7.1. Unión de dos redes separadas mediante OpenVPN en un medio público (Puente)

Un caso frecuente es que tengamos dos redes que pertenecen a la misma corporación y que se necesita unir para compartir recursos (acceso a servidores, trabajo de herramientas corporativas, etc.) Las dos redes se encuentran separadas físicamente y necesitamos unirlas a través de un medio público como puede ser una línea ADLS que da acceso a Internet.

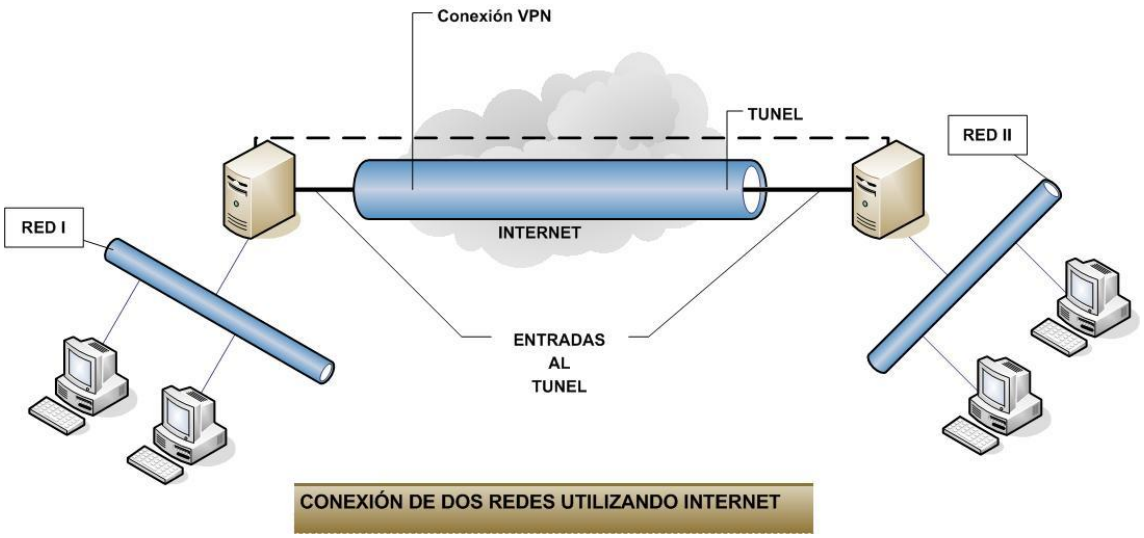

Figura 3: Interconexión de redes

En este caso usaremos dos ordenadores donde levantaremos los extremos del túnel. Toda la información que pongamos en un extremo del túnel pasará a través de éste hasta el otro extremo, de forma cifrada y comprimida. El modo de operación de la VPN es en modo **puente** (*bridge*). En este caso estamos transmitiendo la información a nivel de enlace (nivel 2 del Modelo Osi) y consecuentemente, de modo independiente al protocolo usado. Este quiere decir que todas las tramas (tcp, udp, ipx, NetBEUI, etc) que pongamos en el extremo del túnel aparecerá en el otro extremo, quedando unidas de esta forma las dos redes.

Sin embargo la implementación de prueba que se iba a realizar, no se pudo llevar a cabo debido a problemas con la máquina virtual. La implementación que se intentó montar fue la de la figura 4.

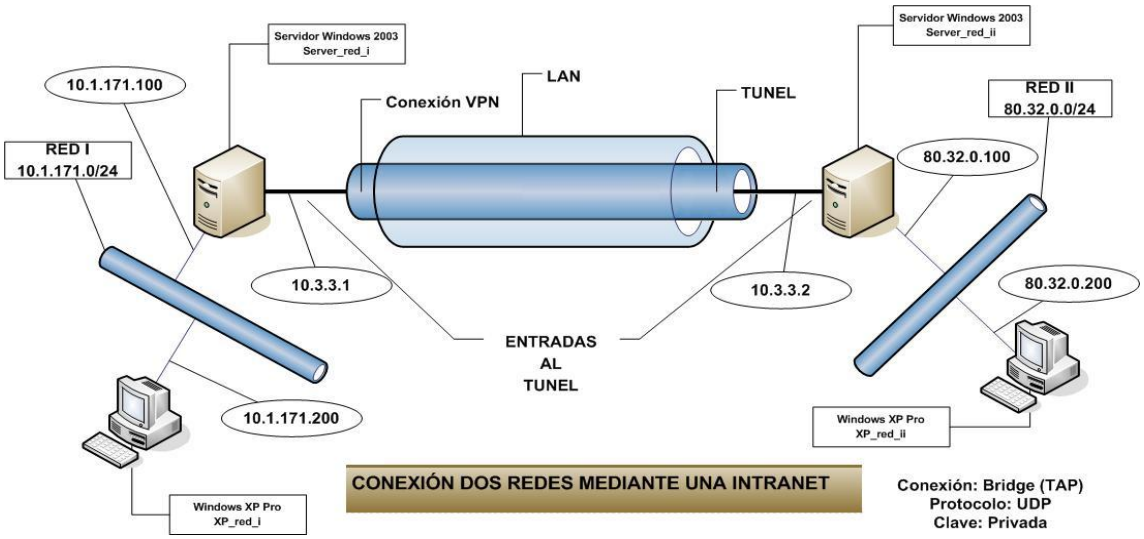

Figura 4: Implementación de la interconexión de redes

Lo que se pretendía, era compartir las carpetas ubicadas en cada una de las redes que estaban unidas mediante la VPN, para comprobar que las tramas de NetBEUI podían atravesar de forma transparente el túnel. Sin embargo no fue posible poner en marcha esta implementación por problemas de las máquinas virtuales, que lo impidió. Ante estos problemas se decidió buscar una alternativa que permitiera comprobar estos extremos. La solución buscada fue la siguiente:

En la figura 5 están las dos máquinas virtuales montadas sobre la misma red local, pero los firewall de los dos equipos se han configurado de tal forma que las únicas tramas que se permiten son las del túnel VPN. Si efectivamente el túnel permite el paso de cualquier trama, al estar levantado se podrán compartir y ver las carpetas entre ambos equipos, pero al deshabilitar el túnel las carpetas no serán visibles desde el otro equipo[1].

Desde una de las máquinas virtuales podemos hacer ***ping*** a los extremos del túnel para comprobar su funcionamiento, tal y como vemos en la figura 6.

[1]Se puede consultar el Anexo E para ver cómo están configurados los firewalls.

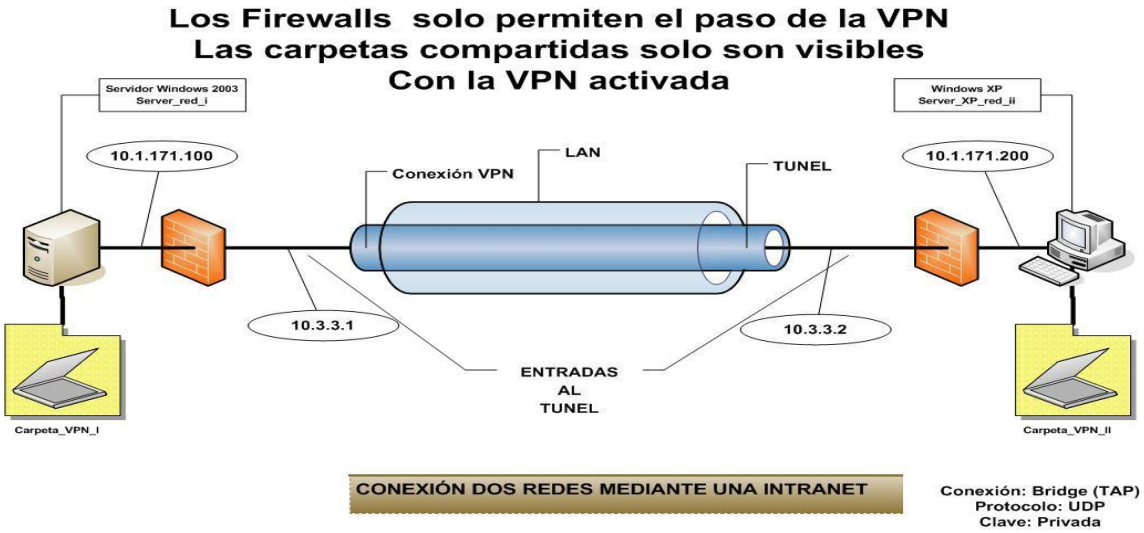

Figura 5: Implementación final

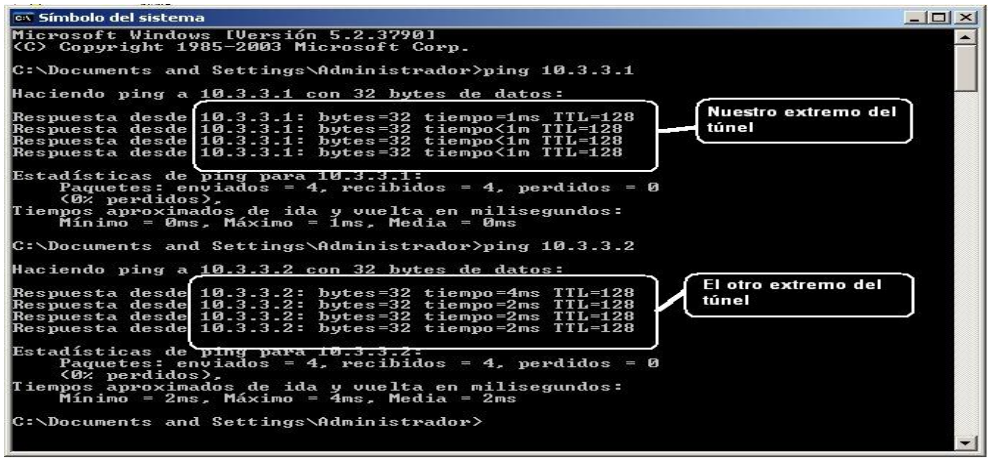

Figura 6: Ping a los extremos del túnel

Aquí podemos ver las dos máquinas virtuales activas con el puente funcionando y podemos ver las carpetas compartidas.

Si por el contrario deshabilitamos el túnel en uno de los extremos para que deje de funcionar, obtenemos el siguiente resultado de la figura 8 cuando intentamos abrir las carpetas compartidas.

De esto deducimos que tanto las tramas *NetBEUI* como las tramas *tcp* (como se puede comprobar con el *ping*), están atravesando el túnel. Por consiguiente hemos conseguido comprobar nuestro primer objetivo, y que no era otro que verificar que la red OpenVPN en modo ***bridge*** actúa a nivel de enlace, permitiendo el paso de cualquier trama a través del túnel, lo cual deja unidas de forma segura las dos redes.

Los problemas encontrados a la hora de realizar esta implementación, fueron los siguientes:

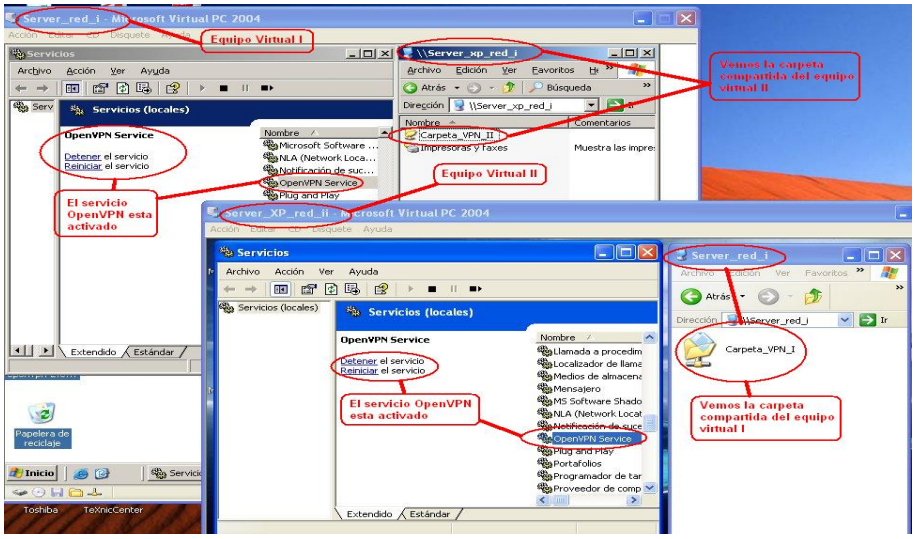

Figura 7: Las dos máquinas virtuales con el túnel levantado

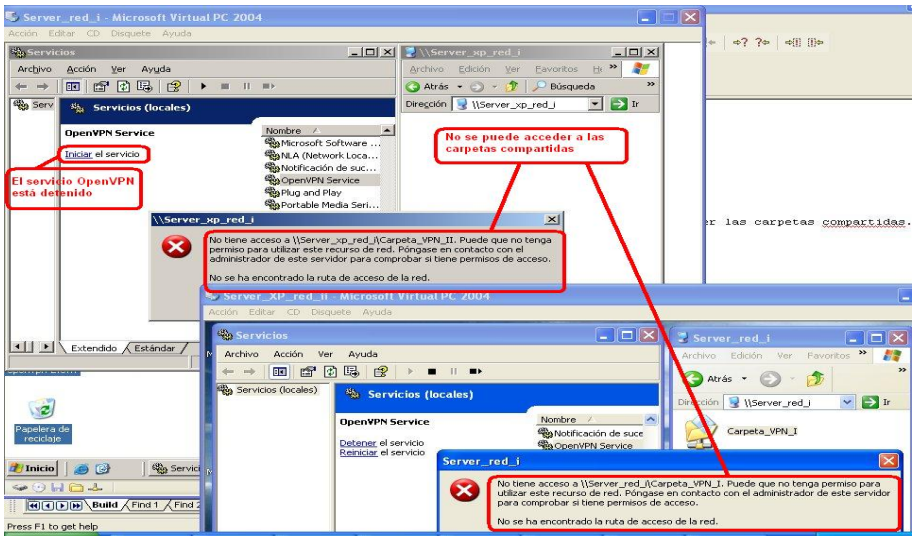

Figura 8: Las dos máquinas virtuales con el túnel deshabilitado

- No se puede levantar un túnel entre 2 Windows 2003 Server (si se cambia un Windows 2003 Server por un XP funciona correctamente).

- Para unir dos redes locales independientes necesitamos dos tarjetas de red que se conectarán mediante un puente de Windows y esto no se ha podido realizar porque las tarjetas de red de la máquina virtual han de ser distintas.

La configuración de los dos equipos la podemos ver en el Anexo C.

7.2. Conexión de Clientes a un Servidor (Túnel) - Road Warrior

Esta es una situación muy habitual dentro de las empresas corporativas. Nos encontramos con una serie de agentes móviles que necesitan desplazarse por distintas ubicaciones relacionadas con la empresa (visita a clientes, visita a sucursales, visita a proveedores, etc). En esta situación también es típico que el *viajero* necesite acceder a la red corporativa, para realizar pedidos, informes, consultas, acceder a su correo, etc. Un gráfico de esta situación lo tenemos en la figura 9.

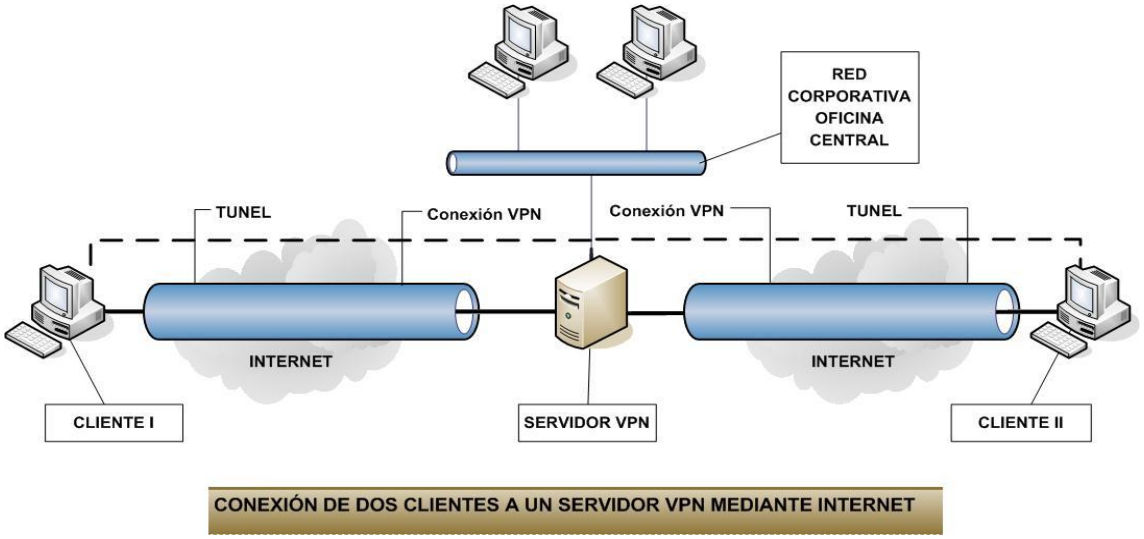

Figura 9: Conexión de clientes a un servidor

La OpenVPN permite un modo de conexión que soluciona este problema. Es el modo *dev_ tun*. En este modo la comunicación se realiza a nivel de red (nivel 3 del *modelo OSI*). También permite la conexión de más de un cliente con el servidor de forma segura.

En este modo el servidor actúa de DHCP para dar direcciones a los clientes. Estas direcciones tienen una máscara de subred, que no permite al cliente establecer ninguna conexión más.

En esta opción comprobaremos que es posible la comunicación a través del *túnel* de la OpenVPN entre distintas plataformas (Linux, Windows 2003 server y Windows XP), así como la utilización de certificados para asegurar la seguridad de la comunicación. La simulación que se realizará la podemos ver en la figura 10.

Como se puede observar en la figura utilizaremos 4 *máquinas virtuales* con un Linux Red Hat, un Windows 2003 server y 2 Windows XP, siendo el servidor la máquina Linux. Probaremos a hacer un *telnet* desde los clientes al servidor y comprobaremos como se encuentran montados los túneles entre los clientes y el servidor, así como el funcionamiento adecuado de los certificados digitales.

Una vez configurados las 4 OpenVPN con los ficheros de configuración y con los certificados procedemos a levantar el servidor, ejecutando el script que tenemos a

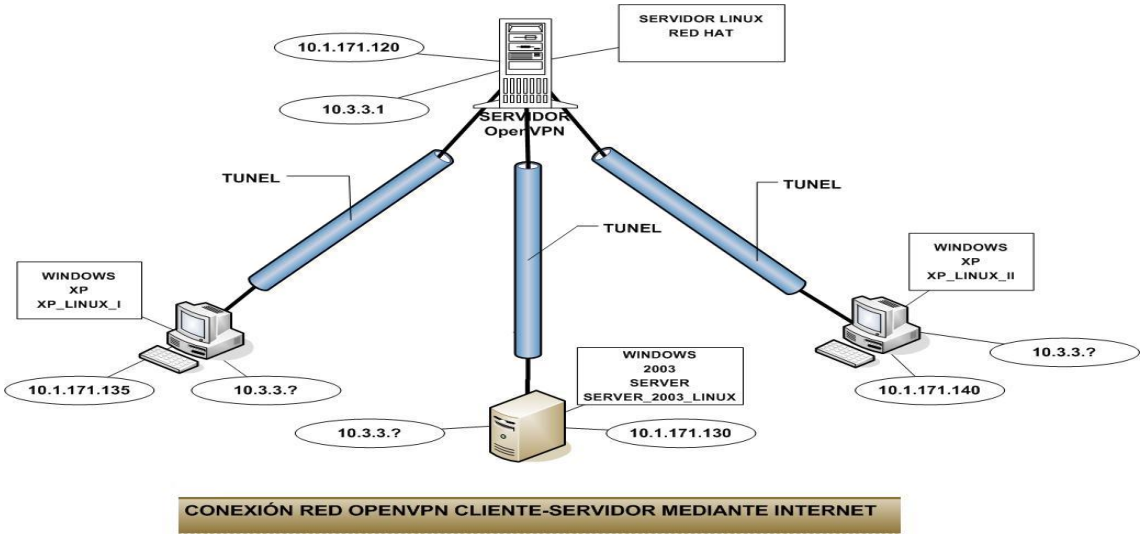

Figura 10: Conexión de 3 clientes a 1 servidor con distintas plataformas

tal efecto (consultar el anexo G):

```
. ./openvpn-startup.sh
```

Y comprobamos que el extremo del túnel de la parte del servidor ha funcionado correctamente, comprobando el archivo de log *server-openvpn.log* como podemos ver seguidamente:

```
Wed Jul  5 06:21:26 2006 OpenVPN 2.0.7 i686-pc-linux [SSL] [LZO] built on Jun  6 2006
Wed Jul  5 06:21:26 2006 Diffie-Hellman initialized with 1024 bit key
Wed Jul  5 06:21:26 2006 TLS-Auth MTU parms [ L:1544 D:140 EF:40 EB:0 ET:0 EL:0 ]
Wed Jul  5 06:21:26 2006 TUN/TAP device tun0 opened
Wed Jul  5 06:21:26 2006 /sbin/ifconfig tun0 10.3.3.1 pointopoint 10.3.3.2 mtu 1500
Wed Jul  5 06:21:26 2006 /sbin/route add -net 10.3.3.0 netmask 255.255.255.0 gw 10.3.3.2
Wed Jul  5 06:21:26 2006 Data Channel MTU parms [ L:1544 D:1450 EF:44 EB:135 ET:0 EL:0 AF:3/1 ]
Wed Jul  5 06:21:26 2006 Listening for incoming TCP connection on [undef]:1194
Wed Jul  5 06:21:26 2006 TCPv4_SERVER link local (bound): [undef]:1194
Wed Jul  5 06:21:26 2006 TCPv4_SERVER link remote: [undef]
Wed Jul  5 06:21:26 2006 MULTI: multi_init called, r=256 v=256
Wed Jul  5 06:21:26 2006 IFCONFIG POOL: base=10.3.3.4 size=62
Wed Jul  5 06:21:26 2006 IFCONFIG POOL LIST
Wed Jul  5 06:21:26 2006 LIN-CLI-1,10.3.3.4
Wed Jul  5 06:21:26 2006 LIN-CLI-2,10.3.3.8
Wed Jul  5 06:21:26 2006 LIN-CLI-3,10.3.3.12
Wed Jul  5 06:21:26 2006 MULTI: TCP INIT maxclients=1024 maxevents=1028
Wed Jul  5 06:21:26 2006 Initialization Sequence Completed
```

El extremo del túnel ha arrancado perfectamente. Seguidamente arrancamos la segunda *máquina virtual* con *Windows 2003 server* y estableceremos el primer puente completo. Para ello pondremos en marcha el servicio de la OpenVPN en el *Windows 2003 server*. El fichero de log de esta máquina nos confirmará si se ha establecido el túnel:

```
Mon Aug 14 22:06:15 2006 NOTE: --user option is not implemented on Windows
Mon Aug 14 22:06:15 2006 NOTE: --group option is not implemented on Windows
```

```
Mon Aug 14 22:06:15 2006 OpenVPN 2.0.7 Win32-MinGW [SSL] [LZO] built on Apr 12 2006
Mon Aug 14 22:06:15 2006 IMPORTANT: OpenVPN's default port number is now 1194, based on an official port number
assignment by IANA.  OpenVPN 2.0-beta16 and earlier used 5000 as the default port.
Mon Aug 14 22:06:15 2006 LZO compression initialized
Mon Aug 14 22:06:15 2006 Control Channel MTU parms [ L:1544 D:140 EF:40 EB:0 ET:0 EL:0 ]
Mon Aug 14 22:06:15 2006 Data Channel MTU parms [ L:1544 D:1450 EF:44 EB:135 ET:0 EL:0 AF:3/1 ]
Mon Aug 14 22:06:15 2006 Local Options hash (VER=V4): '69109d17'
Mon Aug 14 22:06:15 2006 Expected Remote Options hash (VER=V4): 'c0103fa8'
Mon Aug 14 22:06:15 2006 Attempting to establish TCP connection with 10.1.171.120:1194
Mon Aug 14 22:06:15 2006 TCP connection established with 10.1.171.120:1194
Mon Aug 14 22:06:15 2006 TCPv4_CLIENT link local: [undef]
Mon Aug 14 22:06:15 2006 TCPv4_CLIENT link remote: 10.1.171.120:1194
Mon Aug 14 22:06:16 2006 TLS: Initial packet from 10.1.171.120:1194, sid=016b51be 77a9837d
Mon Aug 14 22:06:16 2006 VERIFY OK: depth=1, /C=SP/ST=AV/L=AVILA/O=USAL/CN=LINUX/emailAddress=j.f.h@usal.es
Mon Aug 14 22:06:16 2006 VERIFY OK: depth=0, /C=SP/ST=AV/O=USAL/CN=SERVER/emailAddress=j.f.h@usal.es
Mon Aug 14 22:06:16 2006 Data Channel Encrypt: Cipher 'BF-CBC' initialized with 128 bit key
Mon Aug 14 22:06:16 2006 Data Channel Encrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Mon Aug 14 22:06:16 2006 Data Channel Decrypt: Cipher 'BF-CBC' initialized with 128 bit key
Mon Aug 14 22:06:16 2006 Data Channel Decrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Mon Aug 14 22:06:16 2006 Control Channel: TLSv1, cipher TLSv1/SSLv3 DHE-RSA-AES256-SHA, 1024 bit RSA
Mon Aug 14 22:06:16 2006 [SERVER] Peer Connection Initiated with 10.1.171.120:1194
Mon Aug 14 22:06:17 2006 SENT CONTROL [SERVER]: 'PUSH_REQUEST' (status=1)
Mon Aug 14 22:06:17 2006 PUSH: Received control message: 'PUSH_REPLY,route 10.3.3.1,ping 10,
ping-restart 120,ifconfig 10.3.3.6 10.3.3.5'
Mon Aug 14 22:06:17 2006 OPTIONS IMPORT: timers and/or timeouts modified
Mon Aug 14 22:06:17 2006 OPTIONS IMPORT: --ifconfig/up options modified
Mon Aug 14 22:06:17 2006 OPTIONS IMPORT: route options modified
Mon Aug 14 22:06:18 2006 TAP-WIN32 device [OpenVPN] opened: \\.\Global\{0D1F1079-1514-4DA8-9C8F-00A5FDD9F37F}.tap
Mon Aug 14 22:06:18 2006 TAP-Win32 Driver Version 8.1
Mon Aug 14 22:06:18 2006 TAP-Win32 MTU=1500
Mon Aug 14 22:06:18 2006 Notified TAP-Win32 driver to set a DHCP IP/netmask of
10.3.3.6/255.255.255.252 on interface {0D1F1079-1514-4DA8-9C8F-00A5FDD9F37F}
[DHCP-serv: 10.3.3.5, lease-time: 31536000]
Mon Aug 14 22:06:18 2006 Successful ARP Flush on interface [2] {0D1F1079-1514-4DA8-9C8F-00A5FDD9F37F}
Mon Aug 14 22:06:18 2006 TEST ROUTES: 0/0 succeeded len=1 ret=0 a=0 u/d=down
Mon Aug 14 22:06:18 2006 Route: Waiting for TUN/TAP interface to come up...
Mon Aug 14 22:06:19 2006 TEST ROUTES: 0/0 succeeded len=1 ret=0 a=0 u/d=down
Mon Aug 14 22:06:19 2006 Route: Waiting for TUN/TAP interface to come up...
Mon Aug 14 22:06:20 2006 TEST ROUTES: 0/0 succeeded len=1 ret=0 a=0 u/d=down
Mon Aug 14 22:06:20 2006 Route: Waiting for TUN/TAP interface to come up...
Mon Aug 14 22:06:21 2006 TEST ROUTES: 0/0 succeeded len=1 ret=0 a=0 u/d=down
Mon Aug 14 22:06:21 2006 Route: Waiting for TUN/TAP interface to come up...
Mon Aug 14 22:06:22 2006 TEST ROUTES: 0/0 succeeded len=1 ret=0 a=0 u/d=down
Mon Aug 14 22:06:22 2006 Route: Waiting for TUN/TAP interface to come up...
Mon Aug 14 22:06:23 2006 TEST ROUTES: 1/1 succeeded len=1 ret=1 a=0 u/d=up
Mon Aug 14 22:06:23 2006 route ADD 10.3.3.1 MASK 255.255.255.255 10.3.3.5
Mon Aug 14 22:06:23 2006 Route addition via IPAPI succeeded
Mon Aug 14 22:06:23 2006 Initialization Sequence Completed
```

Podemos apreciar como la secuencia de inicialización en el Windows 2003 Server ha finalizado correctamente. Para verificar que el puente funciona correctamente veamos como ha quedado la red de esta máquina según la figura 11.

Observar la máscara de subred que el servidor ha dado al cliente (255.255.255.252).

Si realizamos un ping sobre el extremo del túnel en el servidor el resultado es la figura 12.

El resultado sobre el servidor lo podemos ver en el siguiente listado:

```
Wed Jul  5 06:21:26 2006 OpenVPN 2.0.7 i686-pc-linux [SSL] [LZO] built on Jun  6 2006
Wed Jul  5 06:21:26 2006 Diffie-Hellman initialized with 1024 bit key
Wed Jul  5 06:21:26 2006 TLS-Auth MTU parms [ L:1544 D:140 EF:40 EB:0 ET:0 EL:0 ]
Wed Jul  5 06:21:26 2006 TUN/TAP device tun0 opened
Wed Jul  5 06:21:26 2006 /sbin/ifconfig tun0 10.3.3.1 pointopoint 10.3.3.2 mtu 1500
Wed Jul  5 06:21:26 2006 /sbin/route add -net 10.3.3.0 netmask 255.255.255.0 gw 10.3.3.2
```

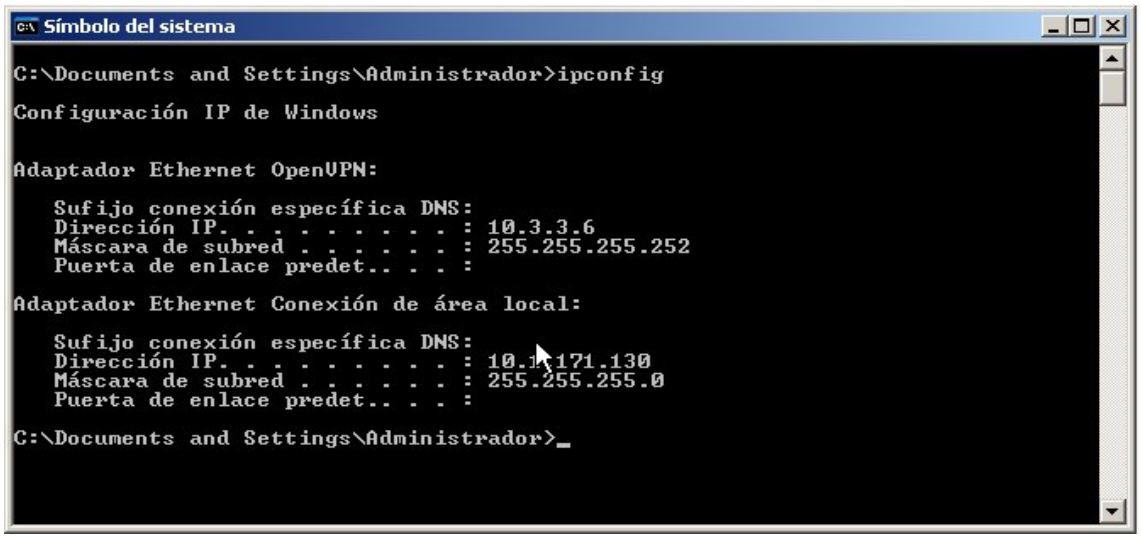

Figura 11: IPCONFIG del cliente

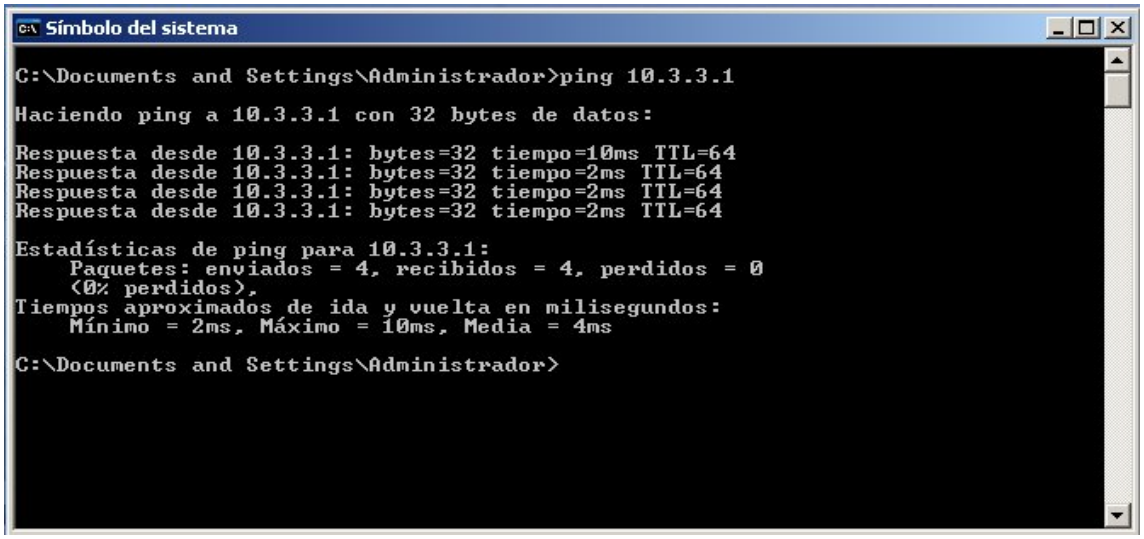

Figura 12: Ping al extremo del túnel opuesto

```
Wed Jul  5 06:21:26 2006 Data Channel MTU parms [ L:1544 D:1450 EF:44 EB:135 ET:0 EL:0 AF:3/1 ]
Wed Jul  5 06:21:26 2006 Listening for incoming TCP connection on [undef]:1194
Wed Jul  5 06:21:26 2006 TCPv4_SERVER link local (bound): [undef]:1194
Wed Jul  5 06:21:26 2006 TCPv4_SERVER link remote: [undef]
Wed Jul  5 06:21:26 2006 MULTI: multi_init called, r=256 v=256
Wed Jul  5 06:21:26 2006 IFCONFIG POOL: base=10.3.3.4 size=62
Wed Jul  5 06:21:26 2006 IFCONFIG POOL LIST
Wed Jul  5 06:21:26 2006 LIN-CLI-1,10.3.3.4
Wed Jul  5 06:21:26 2006 LIN-CLI-2,10.3.3.8
Wed Jul  5 06:21:26 2006 LIN-CLI-3,10.3.3.12
Wed Jul  5 06:21:26 2006 MULTI: TCP INIT maxclients=1024 maxevents=1028
Wed Jul  5 06:21:26 2006 Initialization Sequence Completed
Wed Jul  5 06:33:02 2006 MULTI: multi_create_instance called
Wed Jul  5 06:33:02 2006 Re-using SSL/TLS context
Wed Jul  5 06:33:02 2006 LZO compression initialized
Wed Jul  5 06:33:02 2006 Control Channel MTU parms [ L:1544 D:140 EF:40 EB:0 ET:0 EL:0 ]
Wed Jul  5 06:33:02 2006 Data Channel MTU parms [ L:1544 D:1450 EF:44 EB:135 ET:0 EL:0 AF:3/1 ]
Wed Jul  5 06:33:02 2006 Local Options hash (VER=V4): 'c0103fa8'
Wed Jul  5 06:33:02 2006 Expected Remote Options hash (VER=V4): '69109d17'
```

```
Wed Jul  5 06:33:02 2006 TCP connection established with 10.1.171.130:1120
Wed Jul  5 06:33:02 2006 TCPv4_SERVER link local: [undef]
Wed Jul  5 06:33:02 2006 TCPv4_SERVER link remote: 10.1.171.130:1120
Wed Jul  5 06:33:02 2006 10.1.171.130:1120 TLS: Initial packet from 10.1.171.130:1120, sid=ef731f91 e1b57f6a
Wed Jul  5 06:33:02 2006 10.1.171.130:1120 VERIFY OK: depth=1,
 /C=SP/ST=AV/L=AVILA/O=USAL/CN=LINUX/emailAddress=j.f.h@usal.es
Wed Jul  5 06:33:02 2006 10.1.171.130:1120 VERIFY OK: depth=0,
 /C=SP/ST=AV/O=USAL/CN=LIN-CLI-1/emailAddress=j.f.h@usal.es
Wed Jul  5 06:33:02 2006 10.1.171.130:1120 Data Channel Encrypt:
Cipher 'BF-CBC' initialized with 128 bit key
Wed Jul  5 06:33:02 2006 10.1.171.130:1120 Data Channel Encrypt:
Using 160 bit message hash 'SHA1' for HMAC authentication
Wed Jul  5 06:33:02 2006 10.1.171.130:1120 Data Channel Decrypt:
Cipher 'BF-CBC' initialized with 128 bit key
Wed Jul  5 06:33:02 2006 10.1.171.130:1120 Data Channel Decrypt:
Using 160 bit message hash 'SHA1' for HMAC authentication
Wed Jul  5 06:33:03 2006 10.1.171.130:1120 Control Channel:
TLSv1, cipher TLSv1/SSLv3 DHE-RSA-AES256-SHA, 1024 bit RSA
Wed Jul  5 06:33:03 2006 10.1.171.130:1120 [LIN-CLI-1] Peer Connection Initiated with 10.1.171.130:1120
Wed Jul  5 06:33:03 2006 LIN-CLI-1/10.1.171.130:1120 MULTI: Learn: 10.3.3.6 -> LIN-CLI-1/10.1.171.130:1120
Wed Jul  5 06:33:03 2006 LIN-CLI-1/10.1.171.130:1120 MULTI: primary virtual IP for
LIN-CLI-1/10.1.171.130:1120: 10.3.3.6
Wed Jul  5 06:33:04 2006 LIN-CLI-1/10.1.171.130:1120 PUSH: Received control message: 'PUSH_REQUEST'
Wed Jul  5 06:33:04 2006 LIN-CLI-1/10.1.171.130:1120 SENT CONTROL [LIN-CLI-1]:
'PUSH_REPLY,route 10.3.3.1,ping 10,ping-restart 120,ifconfig 10.3.3.6 10.3.3.5' (status=1)
```

El siguiente paso es levantar los túneles de las dos máquinas *Windows XP* y comprobar que existe comunicación desde las tres máquinas al servidor. Para ello haremos uso del comando *netstat -n*, que nos indicara las conexiones activas que tenemos, la figura 13 nos indica el resultado obtenido.

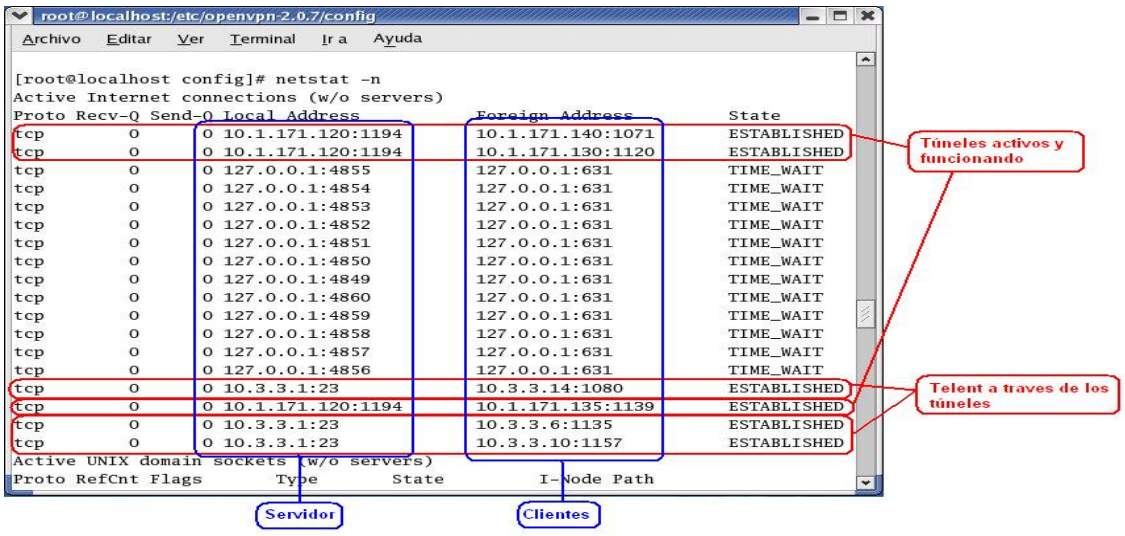

Figura 13: Conexiones activas con el servidor

Podemos observar que el extremo del túnel del servidor tiene la dirección IP 10.3.3.1, y las otras tres máquinas tienen las direcciones IPs: 10.3.3.6 10.3.3.10 10.3.3.14. También se puede observar el establecimiento de los túneles entre el servidor 10.1.171.120 y los clientes 10.1.171.130 10.1.171.135 10.1.171.140.

Todo ello indica claramente que los tres clientes están conectados vía *telnet* al servidor.

La configuración la podemos ver en el Anexo D y la creación de los certificados y de las claves en el Anexo F [11].

Finalmente en la figura 14 podemos ver las cuatro máquinas virtuales.

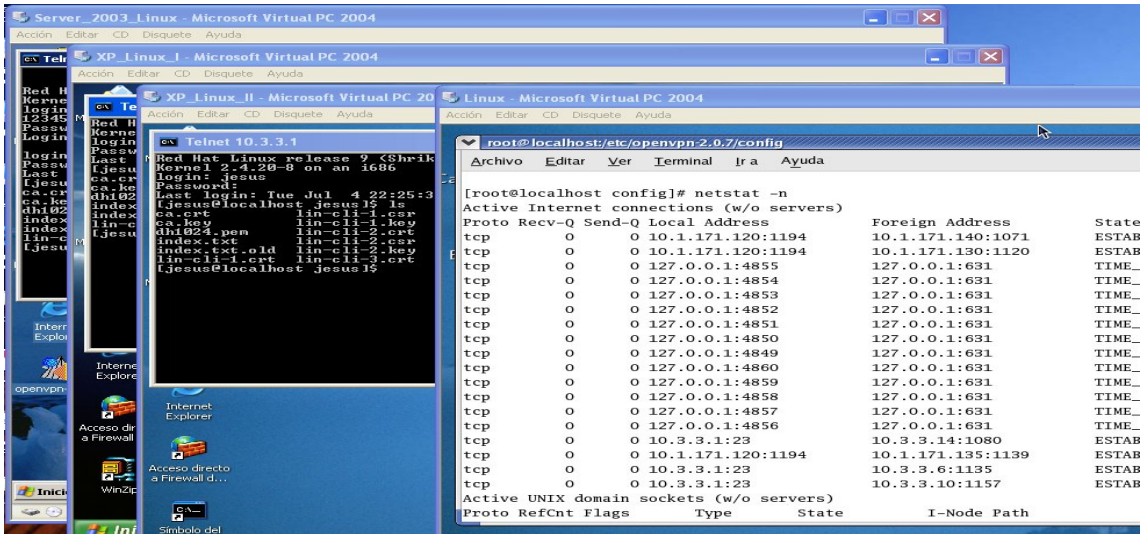

Figura 14: Las cuatro *máquinas virtuales* funcionando

7.3. Conexión real vía Internet a un servidor mediante un túnel

Como última implementación de un escenario, vamos a montar una OpenVPN de forma real entre dos máquinas separadas físicamente y conectadas mediante Internet.

El servidor es un Windows 2003 Server que ha generado los certificados para él y para el cliente. El cliente se encuentra en una *máquina virtual* instalada en un *ordenador portátil*.

Se establecerá la conexión entre el servidor y el portátil y se harán pruebas de rendimiento en la transmisión de los datos, para ello se enviará un fichero en distintas situaciones mediante *ftp*. La configuración que vamos a implementar la tenemos en la figura 15.

En el lado del servidor tenemos un router que ha de ser configurado (Ver el anexo H) para que admita el puerto de OpenVPN y además tenemos que realizar NAT para enviar los paquetes de la VPN al servidor correspondiente. En la parte del cliente no hace falta, ya que como son salida de paquetes no hay que abrir ningún puerto. También dejaremos abierto unos puertos para poder conectarnos al servidor con el programa de control VNC.

Comenzaremos activando el servicio de OpenVPN en el servidor, figura 16.

Comprobamos que se ha levantado el extremo del túnel correctamente, figura 17.

Y realizamos la misma operación en el cliente, figura 18.

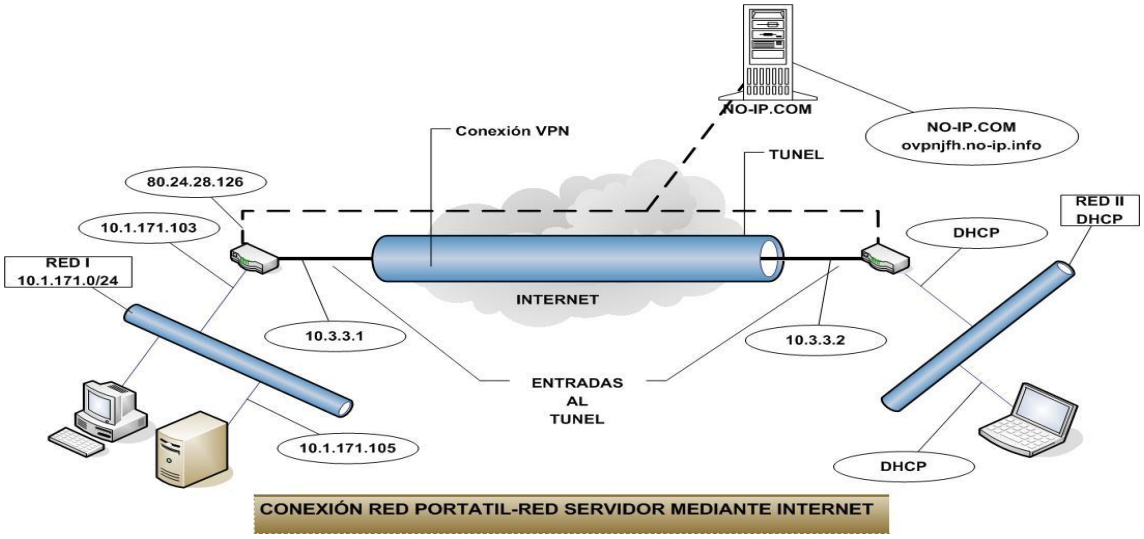

Figura 15: Implementación real de una OpenVPN

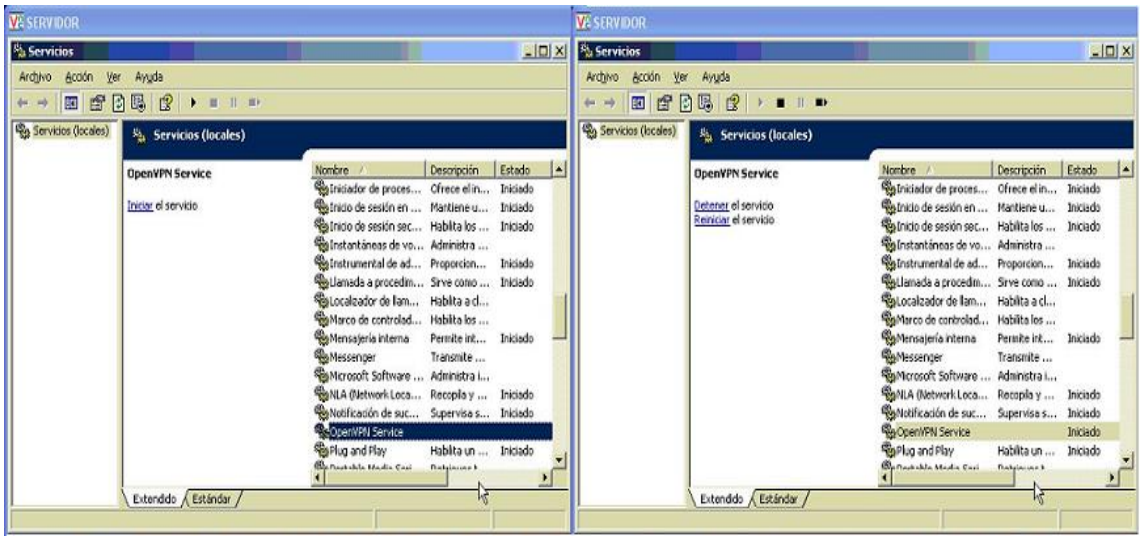

Figura 16: Activación del Servicio en el Servidor

En el servidor, una vez se ha conectado el cliente, podemos observar esta acción en la figura 19.

Comprobamos que el DHCP del servidor le ha dado la IP del extremo del túnel y verificamos en el fichero de *log* que se ha realizado la conexión correctamente, esto lo podemos observar en la figura 20.

7.3.1. Transferencia de información por el canal sin VPN

Vamos a verificar las velocidades de transferencia entre el cliente y el servidor. Para ello se ha dispuesto un servidor *ftp*, que nos permitirá transferir un fichero entre el cliente y el servidor en distintas situaciones y medir el tiempo de transferencia [3].

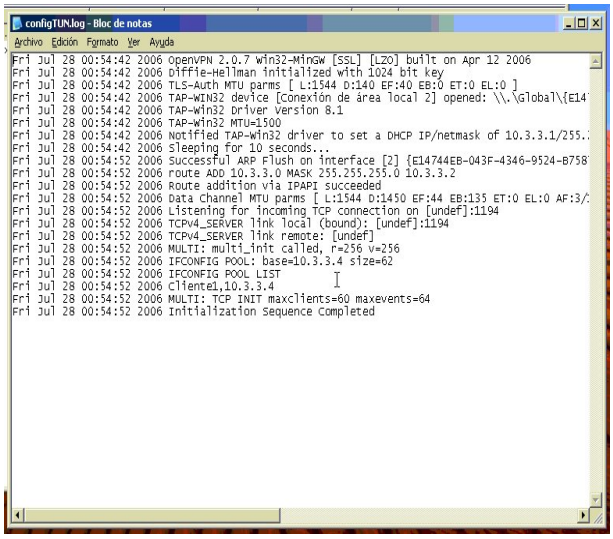

Figura 17: Verificación del funcionamiento de la OpenVPN

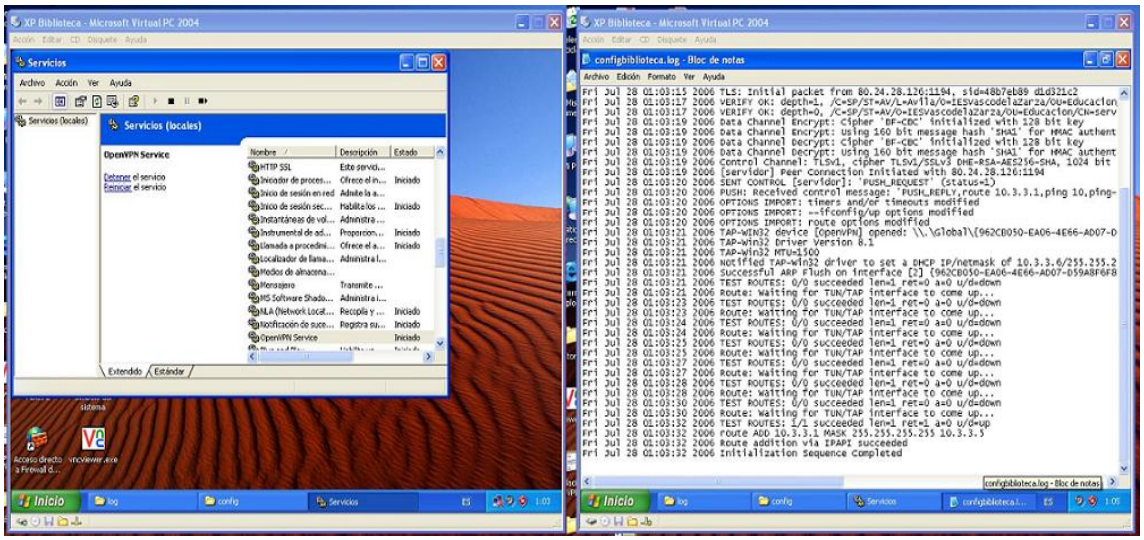

Figura 18: Activación del Servicio en el Cliente

La primera prueba de velocidad se realizará transfiriendo un fichero de aproximadamente 10 Mb entre las dos máquinas, sin pasar por la VPN. En la figura 21 podemos ver el cliente y el servidor *ftp* transfiriendo la información.

El resultado de la transferencia es: **se envió 10.566.617 bytes en 493,543 segundos, que corresponde a un tiempo de 8' 13"**, como se pude apreciar en la figura 22.

7.3.2. Transferencia de información por el canal con VPN y compresión

Seguidamente se conecta el cliente y el servidor a través de la VPN, verificando en la figura 23 el establecimiento de la conexión.

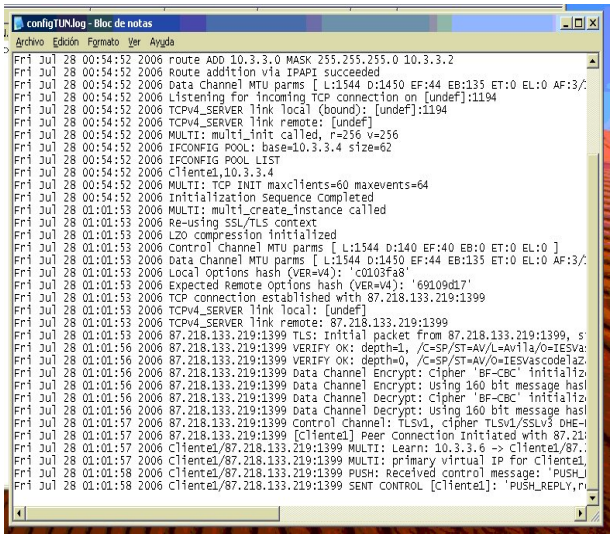

Figura 19: Verificación de la conexión del cliente

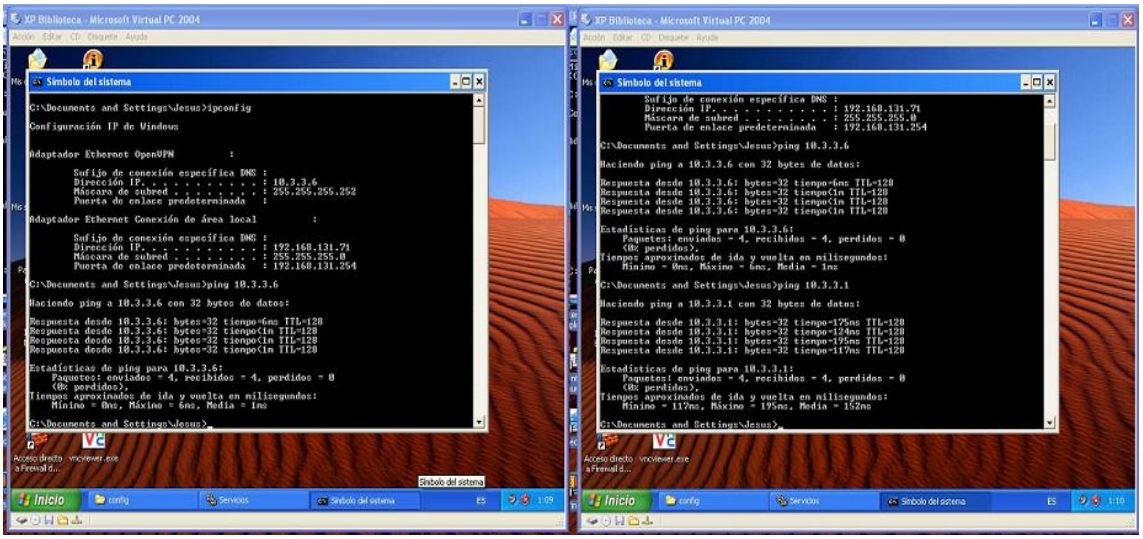

Figura 20: Verificación de la Conexión

El resultado de la transferencia del mismo fichero a través de la VPN con compresión de datos es el siguiente: **10.566.617 bytes en 482,734 segundos, que corresponde a un tiempo de 8' 2"**, como se pude apreciar en la figura 24. Este tiempo es ligeramente inferior al obtenido con la transferencia normal.

7.3.3. Transferencia de información por el canal con VPN y sin compresión

Para finalizar esta prueba de transferencia de información a través de la VPN, volveremos a transmitir el fichero, pero en este caso quitaremos la compresión, para verificar que este parámetro puede tener relevancia a la hora de enviar una gran cantidad de información.

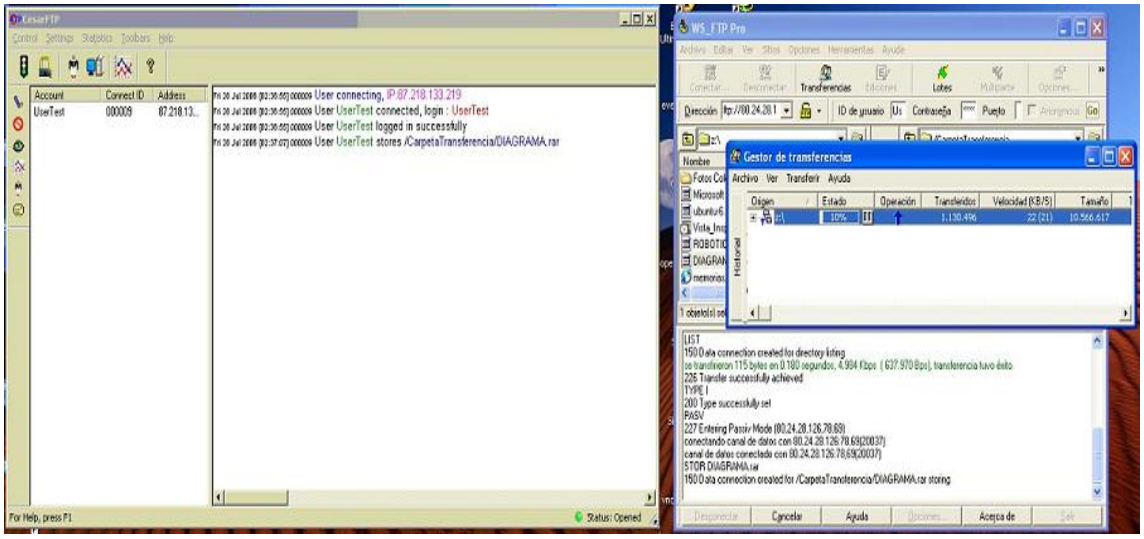

Figura 21: Cliente y servidor transfiriéndose la información

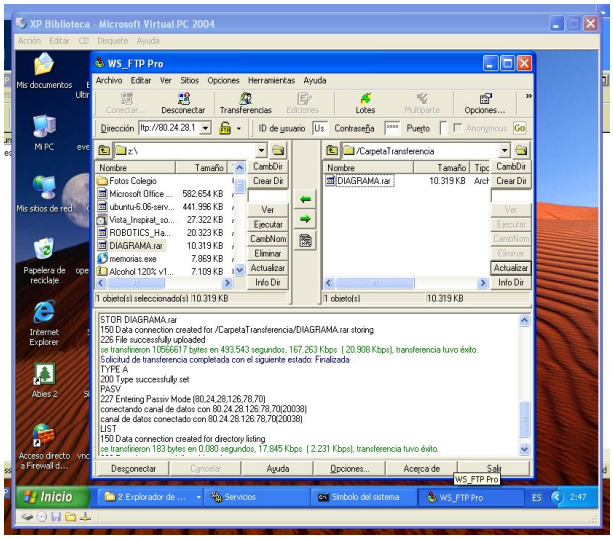

Figura 22: Resultado de la transferencia normal

Se modifican los dos ficheros de configuración para suprimir la compresión como se puede apreciar en la figura 25.

Se vuelve a transferir el fichero, una vez eliminada la compresión, obteniendo los siguientes resultados: **10.566.617 bytes en 464,682 segundos, que corresponde a un tiempo de 7' 44"**. Tiempo inferior a los dos anteriores. Estos no son significativos, dado que solo se ha realizado una única transmisión para cada caso y pueden haber influido otros factores que no se han teniendo en cuenta.

En la figura 26 podemos ver la pantalla del *cliente ftp* con los resultados comentados.

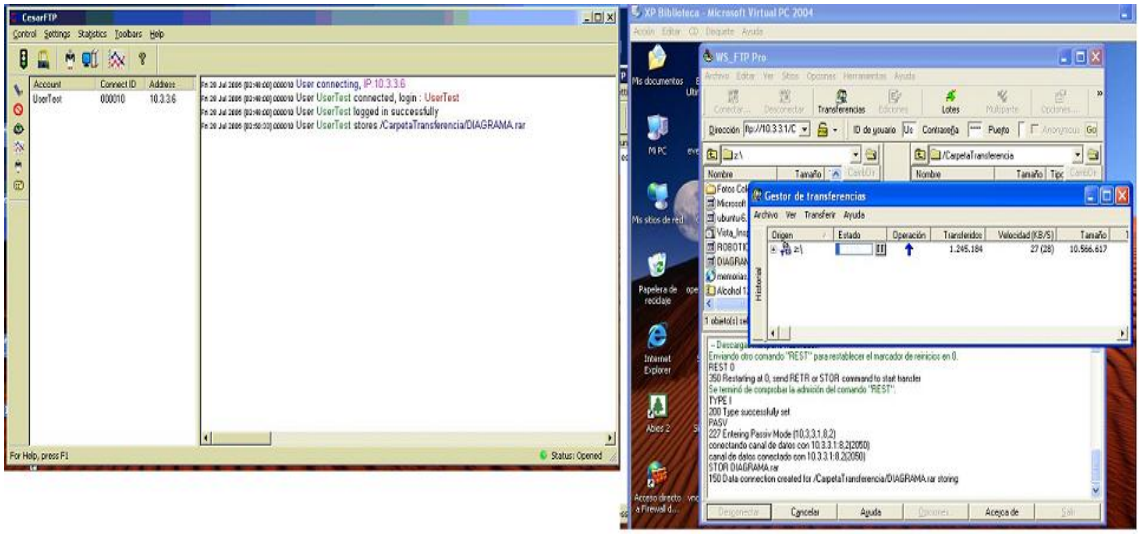

Figura 23: Cliente y servidor transfiriéndose la información a través de la VPN

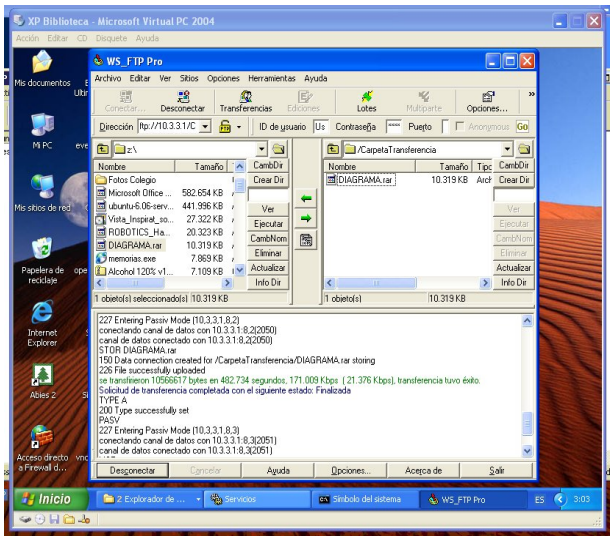

Figura 24: Resultado de la transferencia a través de la VPN

7.3.4. Ejecución de una aplicación a través de la VPN

Para finalizar las pruebas de conexión de la VPN a través de Internet, vamos a ejecutar una aplicación que va a transferir los datos a través del túnel de la VPN.

En concreto la aplicación consiste en un programa de control de bibliotecas *denominado Abies 2*, que utiliza como base de datos Microsoft Access. Esta base de datos se encuentra en el servidor y se abrirá desde el cliente a través del túnel para ver en la pantalla del mismo la aplicación en ejecución.

Empezamos configurando la aplicación para que los datos pasen a través del túnel de la VPN, tal y como se observa en la figura 27, apuntando al extremo del túnel de la VPN (10.3.3.1).

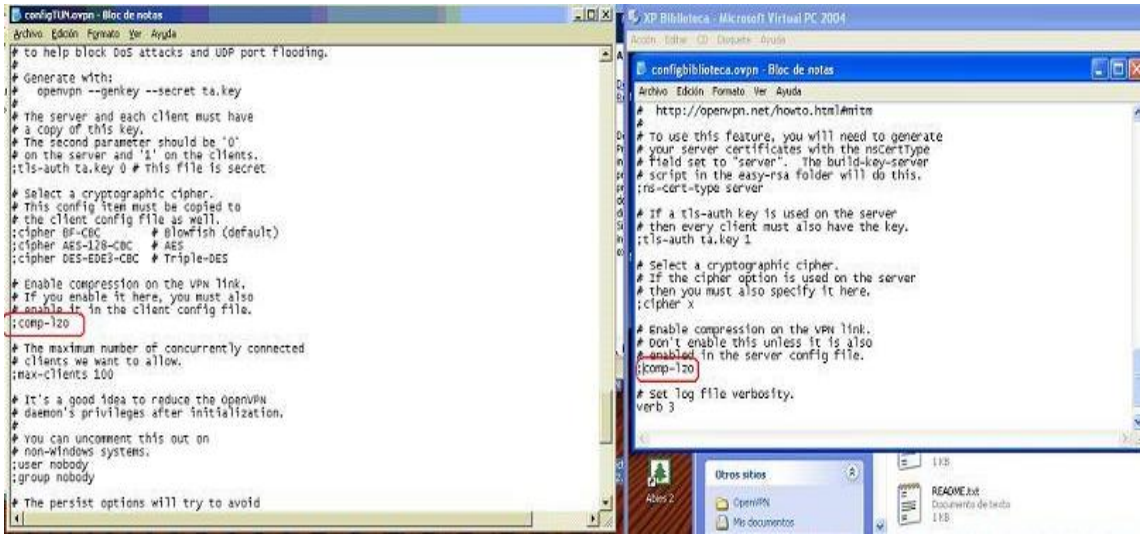

Figura 25: Eliminación de la compresión del cliente y del servidor

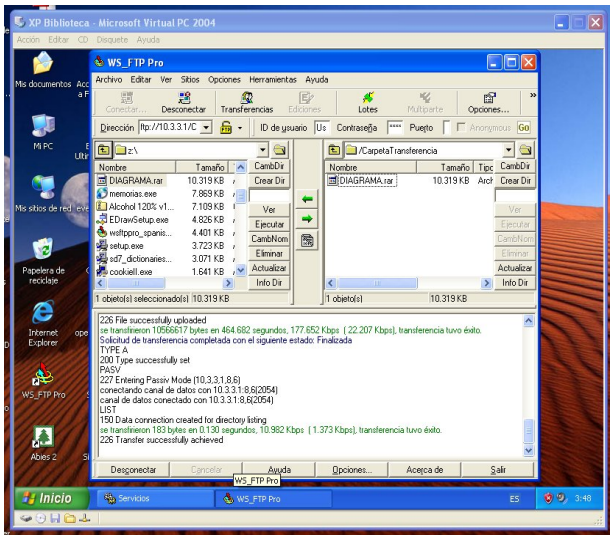

Figura 26: Resultado de la transferencia a través de la VPN sin compresión

En la Administración de Equipos podemos comprobar que la conexión se ha realizado correctamente a través del túnel de la VPN. Para ello veremos las sesiones que tenemos abiertas y las direcciones IP de los equipos conectados; también comprobaremos los archivos abiertos que han viajado a través de la VPN, como observamos en la figura 28.

La configuración utilizada tanto en el servidor como en el cliente, es muy similar a la empleada en el apartado anterior.

Se pueden consultar los Anexos D y F para configurar el servidor y el cliente, así como para generar los certificados y las claves.

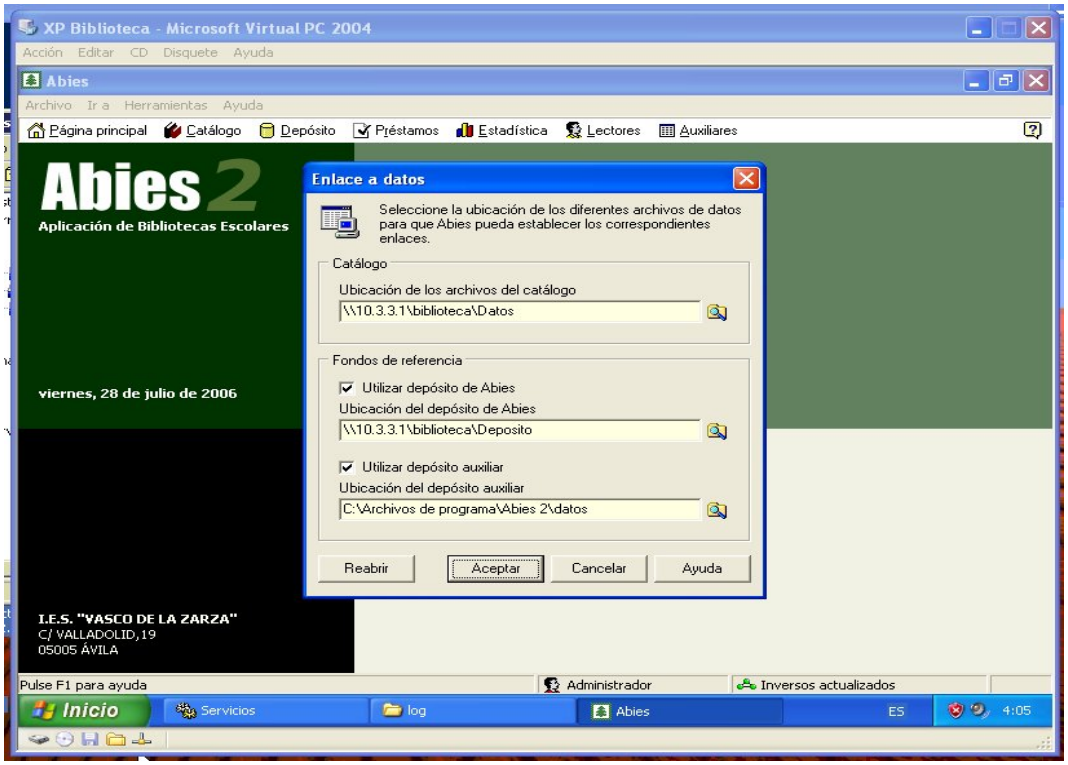

Figura 27: Configuración de la aplicación

8. Exploit

Una búsqueda de los exploit[2] conocidos hasta la fecha, que ha padecido OpenVPN, lo podemos localizar en la referencia [2] que nos da una relación de los últimos exploit de esta VPN, siendo:

.

- **CVE-2006-2229**
  - **Summary:** OpenVPN 2.0.7 and earlier, when configured to use the –management option with an IP that is not 127.0.0.1, uses a cleartext password for TCP sessions to the management interface, which might allow remote attackers to view sensitive information or cause a denial of service.
  - **Published:** 5/5/2006
  - **CVSS Severity:** 3.7 (Low)

[2]Exploit (del inglés to exploit, explotar, aprovechar) es el nombre con el que se identifica un programa informático malicioso, o parte del programa, que trata de forzar alguna deficiencia o vulnerabilidad de otro programa. El fin puede ser la destrucción o inhabilitación del sistema atacado, aunque normalmente se trata de violar las medidas de seguridad para poder acceder al mismo de forma no autorizada y emplearlo en beneficio propio o como origen de otros ataques a terceros [18]

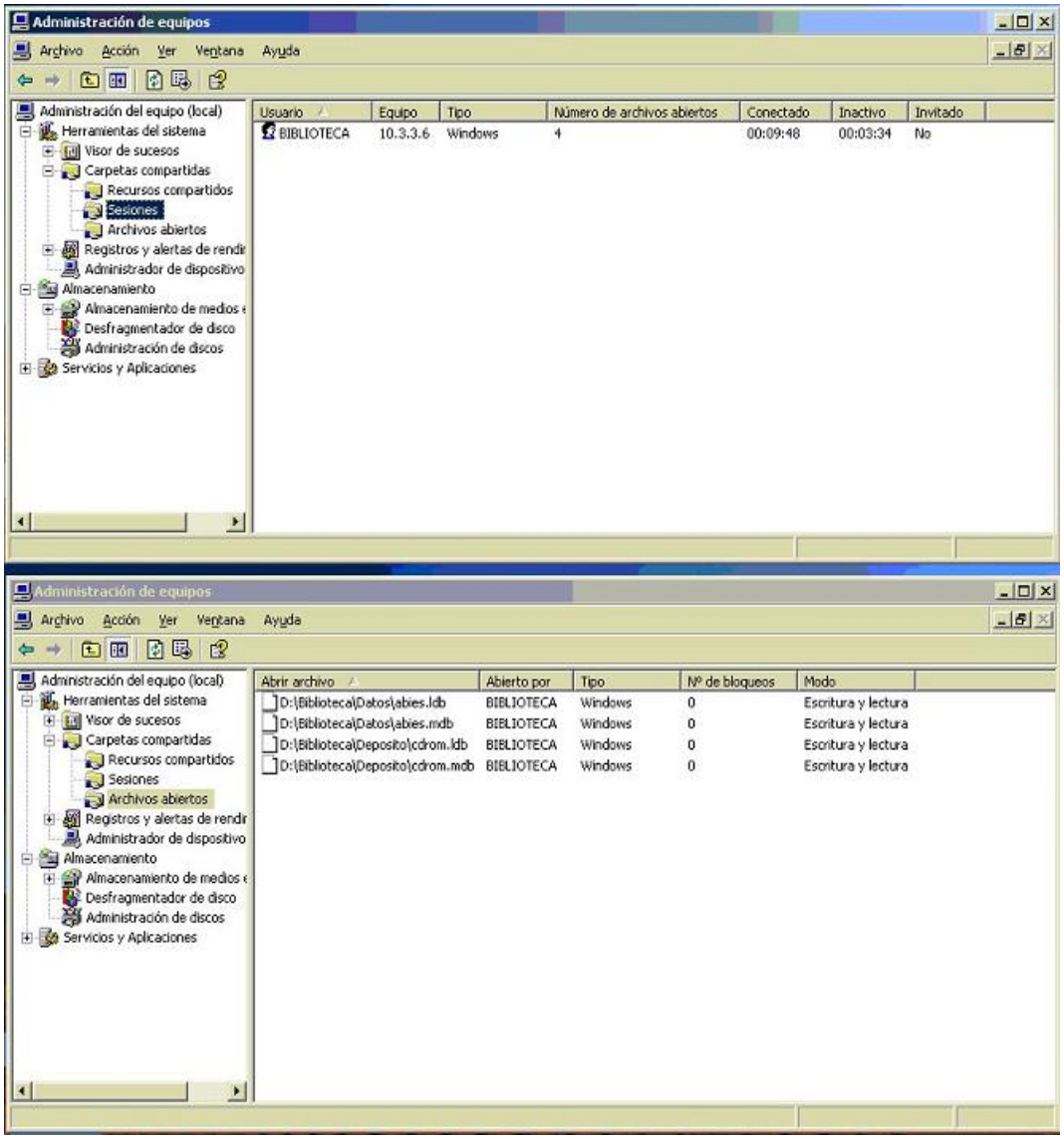

Figura 28: Comprobación de las conexiones en el servidor

- **CVE-2006-1629**
  - **Summary:** OpenVPN 2.0 through 2.0.5 allows remote malicious servers to execute arbitrary code on the client by using setenv with the LD_PRELOAD environment variable.
  - **Published:** 4/6/2006
  - **CVSS Severity:** 6.0 (Medium)
- **CVE-2005-3409**
  - **Summary:** OpenVPN 2.x before 2.0.4, when running in TCP mode, allows remote attackers to cause a denial of service (segmentation fault) by forcing the accept function call to return an error status, which leads to a null dereference in an exception handler.

- **Published:** 11/1/2005
- **CVSS Severity:** 3.3 (Low) Approximated

- **CVE-2005-3393**
  - **Summary:** Format string vulnerability in the foreign_option function in options.c for OpenVPN 2.0.x allows remote clients to execute arbitrary code via format string specifiers in a push of the dhcp-option command option.
  - **Published:** 11/1/2005
  - **CVSS Severity:** 8.0 (High) Approximated

- **CVE-2005-2534**
  - **Summary:** Race condition in OpenVPN before 2.0.1, when –duplicate-cn is not enabled, allows remote attackers to cause a denial of service (server crash) via simultaneous TCP connections from multiple clients that use the same client certificate.
  - **Published:** 8/24/2005
  - **CVSS Severity:** 3.3 (Low) Approximated

- **CVE-2005-2533**
  - **Summary:** OpenVPN before 2.0.1, when running in "dev tap"Ethernet bridging mode, allows remote authenticated clients to cause a denial of service (memory exhaustion) via a flood of packets with a large number of spoofed MAC addresses.
  - **Published:** 8/24/2005
  - **CVSS Severity:** 2.3 (Low) Approximated

- **CVE-2005-2532**
  - **Summary:** OpenVPN before 2.0.1 does not properly flush the OpenSSL error queue when a packet can not be decrypted by the server, which allows remote authenticated attackers to cause a denial of service (client disconnection) via a large number of packets that can not be decrypted.
  - **Published:** 8/24/2005
  - **CVSS Severity:** 3.3 (Low) Approximated

- **CVE-2005-2531**
  - **Summary:** OpenVPN before 2.0.1, when running with "verb 0.ªnd without TLS authentication, does not properly flush the OpenSSL error queue when a client fails certificate authentication to the server and causes the error to be processed by the wrong client, which allows remote attackers to cause a denial of service (client disconnection) via a large number of failed authentication attempts.

- **Published:** 8/24/2005
- **CVSS Severity:** 3.3 (Low) Approximated

Veremos en más detalle el último exploit conocido:

CVE-2006-2229

OpenVPN 2.0.7 y versiones anteriores, cuando está configurado para utilizar la opción –*management* con una IP que no sea 127.0.0.1, utilizando una contraseña en blanco para las sesiones de TCP contra la interfaz de mantenimiento, puede permitir que los atacantes remotos vean información sensible o causen una denegación de servicios.

En definitiva se puede abrir una puerta para entrar al sistema como administrador y sin necesidad de poner una password. Vamos a simular esta situación entre dos *máquinas virtuales*. Empezaremos estudiando el comando *management*:

8.1. Interfaz de administración de OpenVPN

La interfaz *OpenVPN Management* [18] permite que OpenVPN sea administrativamente controlado por un programa externo vía conexión TCP.

El interfaz se ha diseñado específicamente para desarrolladores de GUI que quisieran controlar mediante un programa o remotamente un demonio OpenVPN.

Esta interfaz de administración se implementa usando una conexión TCP cliente/servidor, donde OpenVPN escuchará una dirección IP y un puerto procedentes de conexiones de clientes administradores.

El protocolo de administración es actualmente *texto en claro* sin una capa explícita de la seguridad. Por esta razón, se recomienda que el interfaz de administración escuche en el localhost (127.0.0.1) o en la dirección local de la VPN. Es posible conectar remotamente con el interfaz de administración sobre la propia VPN, aunque algunas capacidades serán limitadas en este modo, tal como la capacidad de proporcionar contraseñas y claves privadas.

Futuras versiones de la interface de administración.

Las versiones futuras de la interfaz de administración podrán permitir conexiones out-of-band (es decir no sobre la VPN) y aseguradas con SSL/TLS.

El interfaz de administración permite en el archivo de configuración de la OpenVPN las siguientes directivas:

```
--management
--management-query-passwords
--management-log-cache
```

Ver la página del manual para documentarse sobre estas directivas.

Una vez que OpenVPN se esté ejecutando con la capa de administración habilitada, se puede realizar telnet al puerto de administración (cerciorarse de poder utilizar un cliente de telnet que entienda el modo ”raw“).

Una vez conectado con el puerto de administración, puedes utilizar el comando ”help“ para listar todos los comandos.

8.2. Comando: –management IP port [pw-file]

Habilitar un servidor TCP en IP:port para utilizar funciones del demonio de administración. pw-file, si está especificado, es un archivo de la contraseña (contraseña en la primera línea) o ”stdin” al prompt desde la entrada estándar. La contraseña proporcionada fijará la contraseña que los clientes del TCP necesitarán proporcionar para tener acceso a las funciones de administración.

La interfaz de administración proporciona un modo especial donde el enlace TCP de administración puede funcionar sobre el mismo túnel. Para permitir este modo, fijar IP = "túnel". El modo túnel hará que la interfaz de administración espere a escuchar una conexión TCP en la dirección local de la interfaz TUN/TAP de la VPN.

Mientras que el puerto de administración se ha diseñado para el control programático de OpenVPN por otras aplicaciones, es posible hacer telnet al puerto, usando un cliente del telnet en modo “raw”. Una vez que esté conectado, teclear ”help” para una lista de comandos.

Se recomienda encarecidamente que la IP esté fijada a 127.0.0.1 (localhost) para restringir la accesibilidad del servidor de administración a los clientes locales.

8.3. Implementación del exploit

Vamos a reproducir en qué circunstancias podemos encontrarnos con este exploit. Para ello modificaremos el script de arranque de la máquina Linux para permitir la conexión a la interfaz administrativa sin contraseña, quedando el script como:

```
#!/bin/bash

# directorio de openvpn para los ficheros de configuración
dir=/etc/openvpn-2.0.7/config
filelog=/etc/openvpn-2.0.7/config/server-openvpn.log

# cargar el modulo del kernerl TUN/TAP
modprobe tun

# habilitar IP forwardig
# echo 1 > /proc/sys/net/ipv4/ipforward

# invocar openvpn
/etc/openvpn-2.0.7/openvpn --cd $dir --log $filelog
  --daemon --config server.conf --management 10.1.171.120 7505
```

Donde se ha añadido el comando: **–management 10.1.171.120 7505**

Seguidamente, una vez levantada la VPN en Linux hacemos desde otra máquina un telnet con la siguiente orden:

telnet 10.1.171.120 7505

Y seguidamente ejecutamos el comando "help" cuyo resultado se presenta en la figura 29.

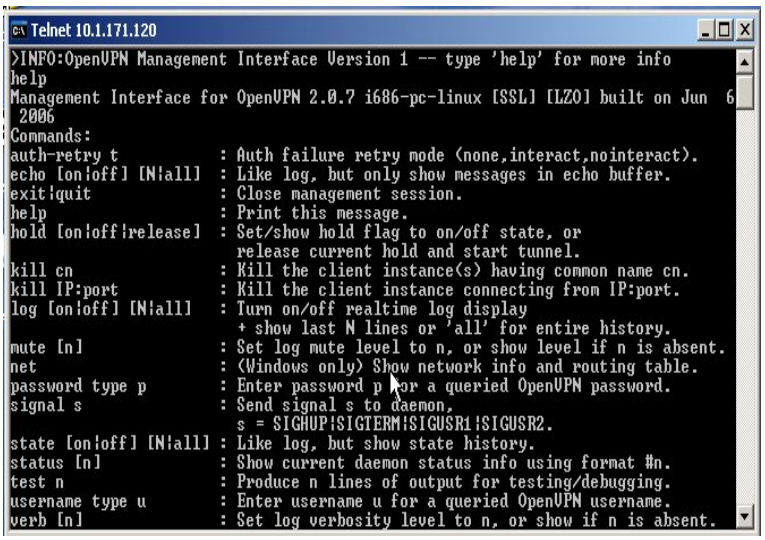

Figura 29: Sesión Telnet interfaz administrador sin clave

Añadimos al script el nombre de un fichero, cuya primera línea contiene la clave (*OpenVPN*) a utilizar, quedando el script:

```
#!/bin/bash

# directorio de openvpn para los ficheros de configuración
dir=/etc/openvpn-2.0.7/config
filelog=/etc/openvpn-2.0.7/config/server-openvpn.log

# cargar el modulo del kernerl TUN/TAP
modprobe tun

# habilitar IP forwardig
# echo 1 > /proc/sys/net/ipv4/ipforward

# invocar openvpn
/etc/openvpn-2.0.7/openvpn --cd $dir --log $filelog
--daemon --config server.conf --management 10.1.171.120 7505 paswwd.txt
```

Al ejecutar el Telnet, nos pide la password, como se puede ver en la figura 30.

8.4. Conclusiones sobre el exploit

Una vez implementada la simulación mediante dos máquinas virtuales, llegamos a la siguiente conclusión sobre el exploit:

1. Se han simulado o verificado los siguientes puntos:

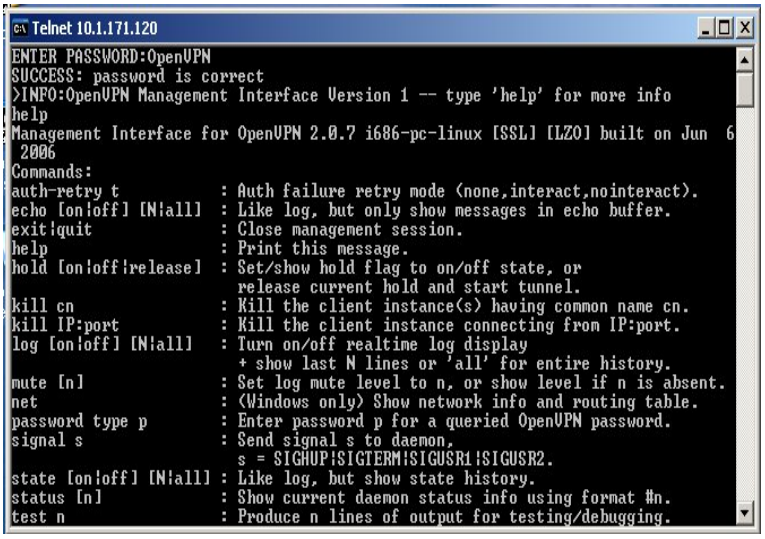

Figura 30: Sesión Telnet interfaz administrador con clave

- La administración remota no se permite por defecto, hay que habilitarlo explícitamente.
- Se permite la habilitación de la administración de forma remota y sin contraseña.
- Se permite la habilitación de la administración de forma remota con contraseña, pero esta contraseña se envía como texto en claro, con lo cual, la seguridad se ve comprometida.
- El intento de realizar la administración remota, a través de la VPN, no ha sido posible realizarla, dado que cuando asignamos la IP de administración al extremo del túnel, obtenemos un error, de que dicha IP no es válida (el túnel no se encuentra levantado cuando realiza la verificación). Esto implica que no se puede realizar la administración remota a través de la VPN.

2. A nuestro entender no se puede considerar un exploit, como tal, aunque hay diversidad de opiniones sobre este punto. Para nosotros es suficiente con no habilitar la *administración remota*, habida cuenta que esta perfectamente documentado y advertido en los manuales de la VPN. Pero un usuario poco avanzado o que no lea todos los puntos sobre la documentación de uso de la VPN, puede pasar por alto este problema y dejar abierta una puerta al sistema. Sobre este punto la opinión mas conservadora es que "**la seguridad debe de ser por defecto**".

9. Conclusiones

La red privada virtual OpenVPN es una alternativa seria a las tradicionales y comerciales VPNs [8].

La facilidad de instalación y mantenimiento, su alto grado de seguridad, la posibilidad de instalación en la mayoría de las plataformas, ser código abierto (evitando puertas traseras), etc. la hacen idónea para ser instalada tanto en grandes corporaciones como en las instalaciones de pequeños usuarios para proteger sus redes inalámbricas.

El principal problema que tiene en la actualidad todas las VPNs es su desconocimiento por el público en general y la poca documentación que sobre el tema existe.

Sería interesante para futuros trabajos ahondar en el estudio de los diversos parámetros de configuración de OpenVPN, así como el estudio del código de la misma, intentando encontrar exploit y bugs que pudieran hacer esta magnífica VPN mas segura.

A. Instalación de OpenVPN en Windows

Para realizar la instalación de OpenVPN en Windows, comenzaremos por descargar el programa que encontraremos en la página del mismo cuya referencia tenemos en [20], descargando un programa autoinstalable.

El proceso de instalación es tan sencillo como seguir las instrucciones y seleccionar las opciones que nos presente el *asistente de instalación*.

Seguidamente procederemos a instalarlo apareciendo la ventana de bienvenida:

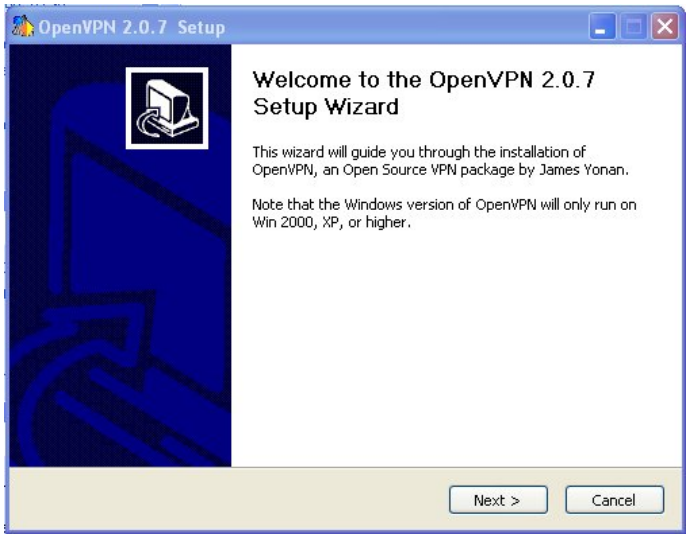

Figura 31: Ventana de bienvenida

Pulsamos en siguiente y se nos presenta la ventana con la licencia (***podemos observar que es Open Source***) que tendremos que aceptar pulsando en *I Agree*:

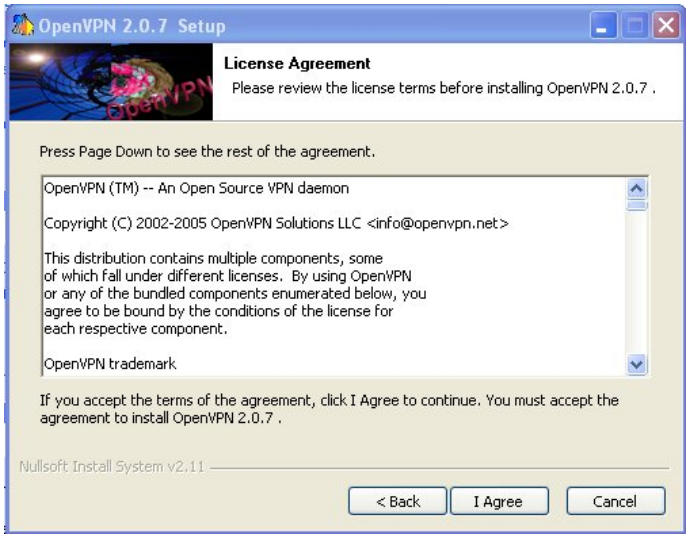

Figura 32: Licencia del producto

Seguidamente nos pide los componentes del producto a instalar, por defecto seleccionaremos todos, tal y como podemos ver en 33.

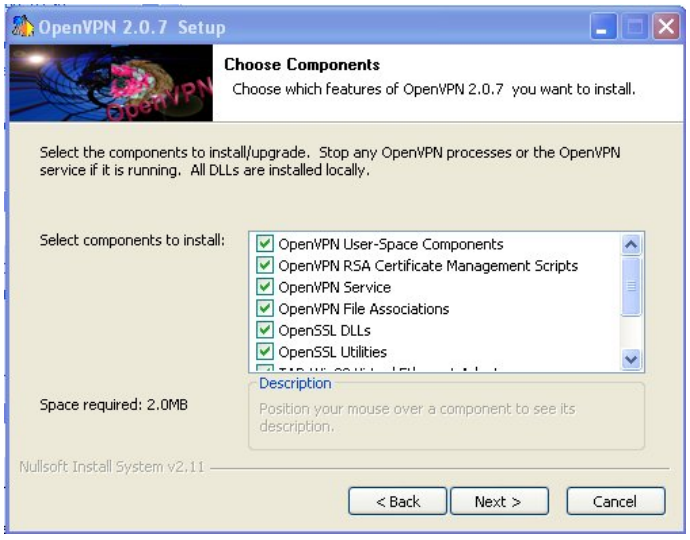

Figura 33: Componentes a instalar

El asistente nos indicará en qué directorio queremos colocar el software y el espacio requerido, así como el espacio disponible:

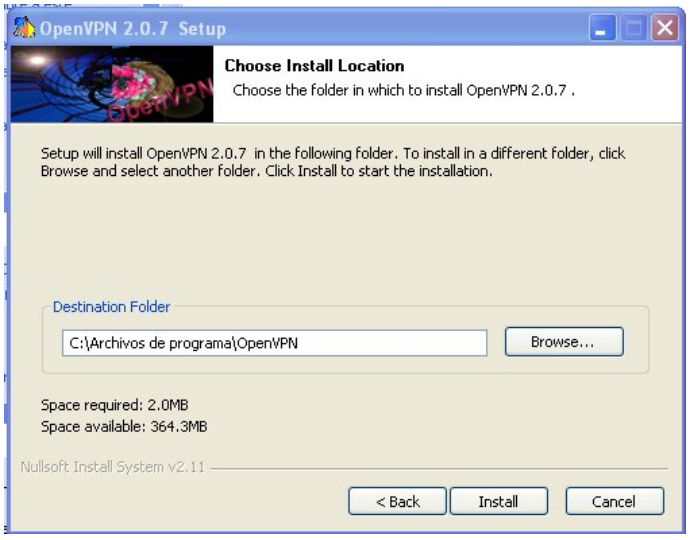

Figura 34: Ubicación de la instalación

Comienza la instalación.

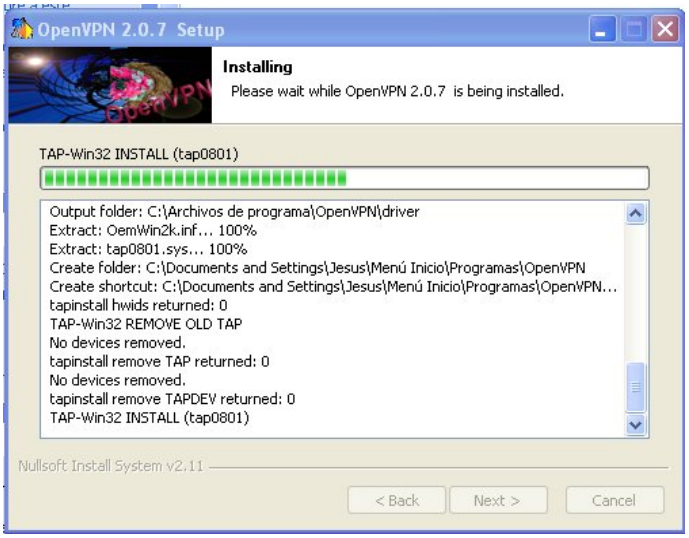

Figura 35: Proceso de instalación

Este proceso se para en el punto *TAP-Win32 INSTALL* donde Windows nos advierte que se va a proceder a instalar un componente potencialmente peligroso, en realidad estamos instalando una tarjeta de red virtual. Pulsamos *Continuar*.

Figura 36: Aviso de Windows

El proceso de instalación ha finalizado, al completar las dos siguientes ventanas, tal y como podemos observar:

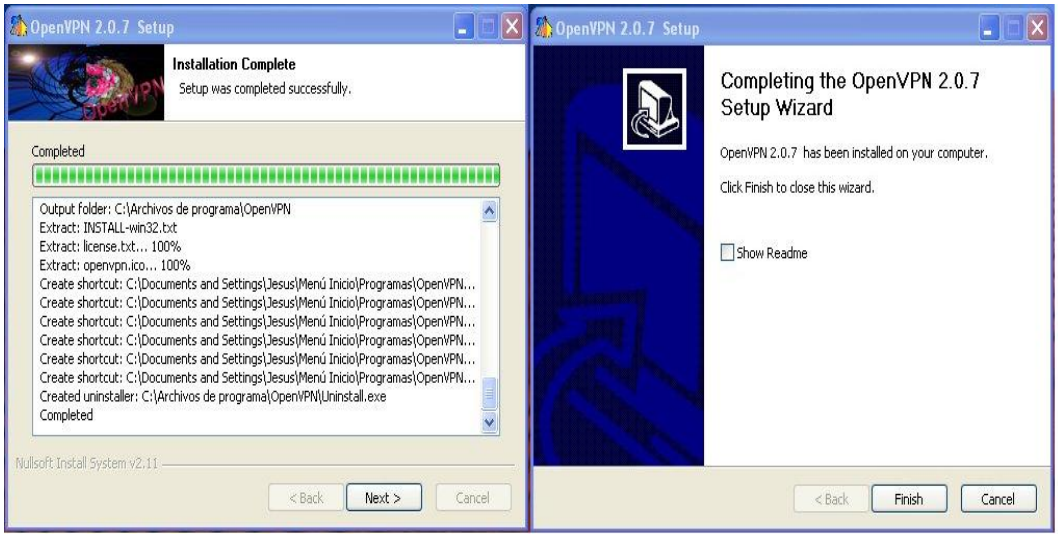

Figura 37: Finalización de la instalación

Para comprobar que la VPN se ha instalado como un servicio, realizaremos la siguiente secuencia de pulsaciones: *Inicio->Panel de Control->Herramientas Administrativas->Servicios*, y podremos comprobar que se ha instalado el *servicio de OpenVPN*, aunque se encuentra parado, como podemos observar en la figura 38.

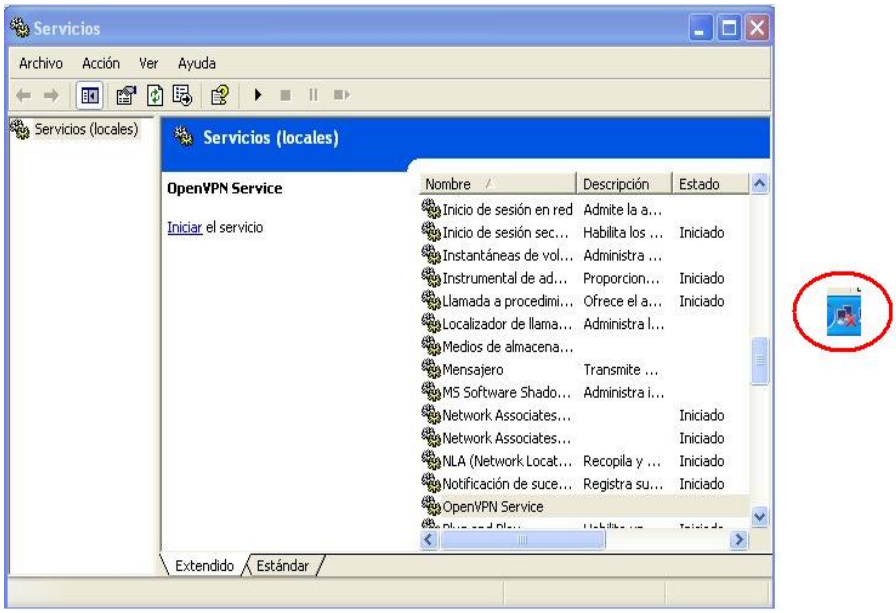

Figura 38: Servicio OpenVPN instalado e icono red

También podremos observar que tenemos un icono de una conexión de red que no está activada, dado que el túnel no se encuentra levantado.

B. Instalación de OpenVPN en Linux

De todas las posibles instalaciones de Linux, nos centraremos en la que vamos a realizar para una Red Hat 9, aunque la instalación en otras distribuciones es muy similar.

Empezaremos descargando el paquete de OpenVPN de su página [18], siendo la versión disponible en este momento la 2.0.7 y que corresponde con el archivo: *openvpn-2.0.7.tar.gz*, esta versión es *NO-RPM*, se ha seleccionado por ser un poco más compleja de instalar que las distribuciones bajo *RPM*.

Los pasos a seguir son:

Posicionarnos en el directorio donde queremos hacer la instalación (/usr) con el paquete a instalar y ejecutar:

tar xfz openvpn-2.0.7.tar.gz

Esto crea el directorio *openvpn-2.0.7* con toda su estructura interna, figura 39 :

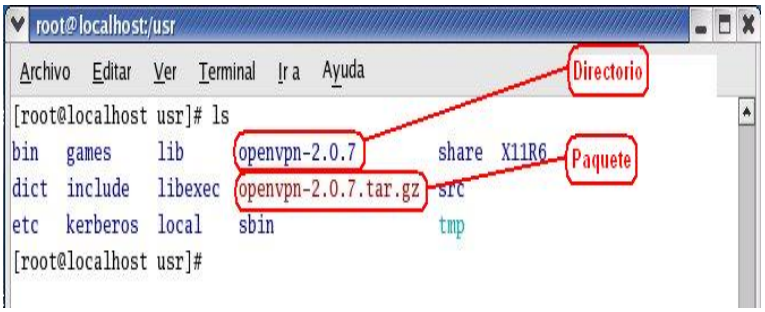

Figura 39: Descompresión del paquete y creación de la estructura de directorios

Entramos dentro del directorio creado *openvpn-2.0.7* con el comando:

cd openvpn-2.0.7

Seguidamente dentro del directorio ejecutamos:

./configure

Esto comprueba las dependencias del paquete OpenVPN con otros paquetes que deberá tener el sistema operativo. En caso de faltar algún elemento, descargarlo e instalarlo. En la figura 40 podemos ver el resultado de este comando.

Si no da errores o estos están subsanados, seguidamente teclearemos los comandos:

make make install

Estos comandos realizaran la compilación y ubicación de los programas de la OpenVPN, el resultado de la aplicación de los comandos se puede ver en la figura 41.

Si no ha habido errores la instalación está completada. Consultar los anexos correspondientes para configurar la OpenVPN.

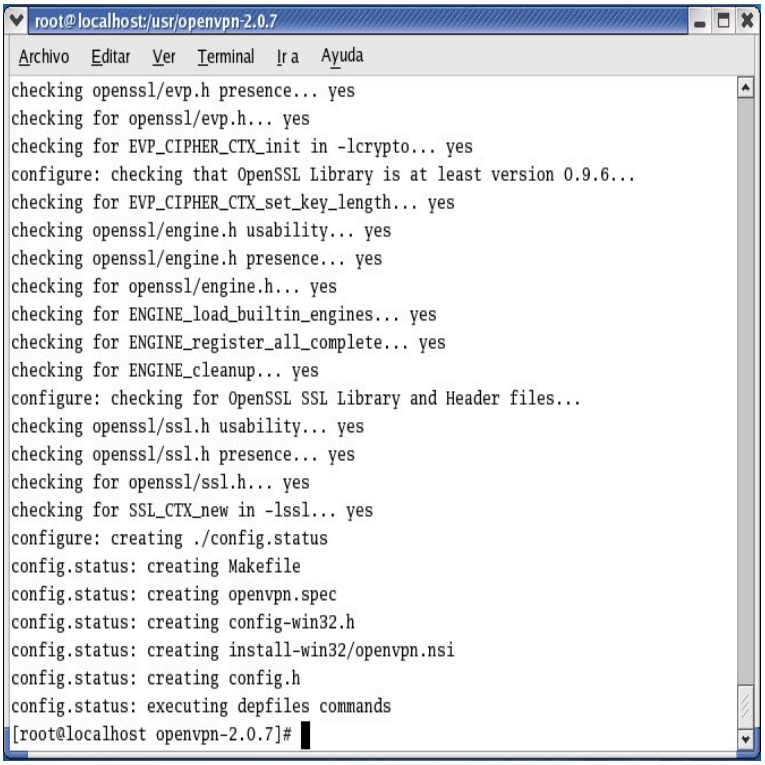

Figura 40: Resultado del *configure*

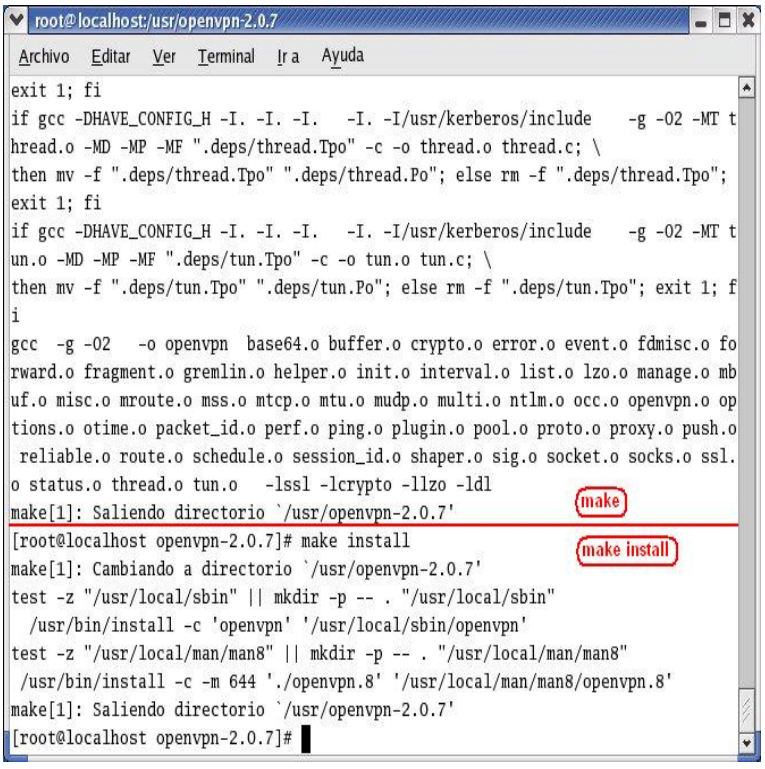

Figura 41: Resultado de *make y make install*

C. Configuración modo puente (bridge [dev-tap])

En la configuración en **Modo Puente** (*bridge*) utilizaremos como partida el fichero de plantilla que el proceso de instalación deja en la carpeta:

`C:\Archivos de programa\OpenVPN\sample-config\ en el caso de Windows`

Y que se llama:

sample.ovpn

Pondremos este archivo de forma comentada.

C.1. sample.ovpn

```
# Editar este archivo, y salvar con la extensión .ovpn
# para que OpenVPN lo active cuando funcione como un servicio.

# Cambiar 'myremote' a su host remoto,
# o comentarlo para poner el servidor en
# modo escucha
; remote myremote

# Descomentar esta línea para poner un
# número de puerto diferente al de por
# defecto 1194
; port 1194

# Elegir una de los tres protocolos soportados por
# OpenVPN. Si se comenta la línea por defecto se
# usa udp
; proto [tcp-server | tcp-client | udp]

# Se debe de especificar uno de los dos protocolos
# posibles de red, 'dev tap' o 'dev tun' para ser usado
# en ambos extremos de la conexión. 'tap crea una
# VPN usando el protocolo de ethernet mientras que
# 'tun' usa el protocolo IP. Use 'tap' si se quiere
# realizar un puente ethernet o hacer enrutamiento
# de broadcasts. 'tun' es algo mas eficiente pero
# requiere la configuración del cliente software
# que no depende de broadcasts. Algunas plataformas
# como Solaris, OpenBSD, y Mac OS X solo soportan
# interfaces 'tun', así que si desea conectar con
# alguna de estas plataformas debes también usar
# el interface 'tun' en el lado de Windows.

# Habilitar 'dev tap' o 'dev tun' pero no ambos!
```

```
dev tap

# Este es un 'dev tap' ifconfig que crea
# una subred ethernet virtual.
# 10.3.0.1 es la dirección IP local de la VPN
# y 255.255.255.0 es la subred VPN.
# Solamente definir esta opción para 'dev tap'.
ifconfig 10.3.0.1 255.255.255.0

# Este es un ifconfig 'dev tun' que crea
# una unión IP Punto a Punto.
# 10.3.0.1 es la dirección IP local de la VPN y
# 10.3.0.2 es la dirección IP remota de la VPN.
# Definir solamente esta opción para 'dev tun'.
# Se cerciore de incluir la opción "tun-mtu"
# en la máquina remota, pero intercambie las
# las direcciones en la orden ifconfig.
; cuba-MTU 1500
; ifconfig 10.3.0.1 10.3.0.2

# Si tiene fragmentaciones o mal configurados routers
# en el camino de los bloques MTU, bajo el TCP MSS e
# internamente fragmentos de protocolo no-TCP.
;fragment 1300
;mssfix

#si has instalado más de un adaptador TAP-Win32
# en tu sistema, debes referirle por su nombre.
;dev-node my-tap

# Puedes generar una llave estática para
# OpenVPN seleccionando la opción ``Generate Key'
# en el menú de inicio.
#
# También puede generar una key.txt manualmente
# con el siguiente comando:
#     openvpn --genkey --secret key.txt
#
# La llave ha de ser igual en ambos extremos de la
# conexión. Así que se debe de generar en una máquina
# y copiarla a la otra por un medio seguro.
# Colocar key.txt en el mismo directorio donde
# esta el fichero de configuración.
secret key.txt

# Descomentar esta sección para una detección
```

```
# más fiable que cuando un sistema pierde la conexión.
# Como por ejemplo llamadas telefónicas o portátiles
# que se desplacen a otras localizaciones.
#
# Si se habilita esta sección y ‘‘myremote’’
# es un nombre dinámico DNS (por ejemplo dyndns.org),
# OpenVPN ‘‘sigue’’ de forma dinámica la dirección IP
# si esta cambia.
; ping-restart 60
; ping-timer-rem
; persist-tun
; persist-key
; resolv-retry 86400

# ping conexión persistente
ping 10

# Habilitar la compresión LZO
comp-lzo

# respuesta moderada
verb 4
mute 10
```

Para el escenario especificado (ver figura 5) hemos utilizado los siguientes ficheros:

C.2. server_red_i

```
remote 10.1.171.200
port 1194
proto udp
dev tap
ifconfig 10.3.3.1 255.255.255.0
secret key.txt
ping 10
comp-lzo
verb 4
mute 10
```

C.3. Server_XP_Red_ii

```
remote 10.1.171.100
port 1194
proto udp
```

```
dev tap
ifconfig 10.3.3.2 255.255.255.0
secret key.txt
ping 10
comp-lzo
verb 4
mute 10
```

C.4. Clave key.txt

```
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
59880e66645bc643ed548873112fa3ee
4de557c4d262899d6e6e426078039b90
cea0c71c7ca6a131b3e5a5b238ad7fdb
439eeee749516b9c18cd0a9201eb9510
c7fec1c2ffd9e25ffefa5a715edc31cc
19e4a94d2f768d9f4cc7d53abc86c77c
4c52f48d9485d2207c825d3d15d1cdee
e577a9cb261e00dcbb92cea20dd262a6
633616e1a150ff2c241b81b6b951dc2f
b2268b50fd8128109e88307a1ccca296
2d92e2a971e71a6930a593be06b2fd91
8912b4baaf39e00c2227cc25686efc2e
8153e3773c84598dbb5be24ff4f9bcf6
fbbcb9cc9c30903978174516d95e896b
1cac378da9deefdee1433271da4bf6d3
2393d7338a4c01f284313f17413b01a0
-----END OpenVPN Static key V1-----
```

C.5. Verificación del funcionamiento

Si todo es correcto y la VPN funciona correctamente obtendremos un fichero con las siguientes líneas:

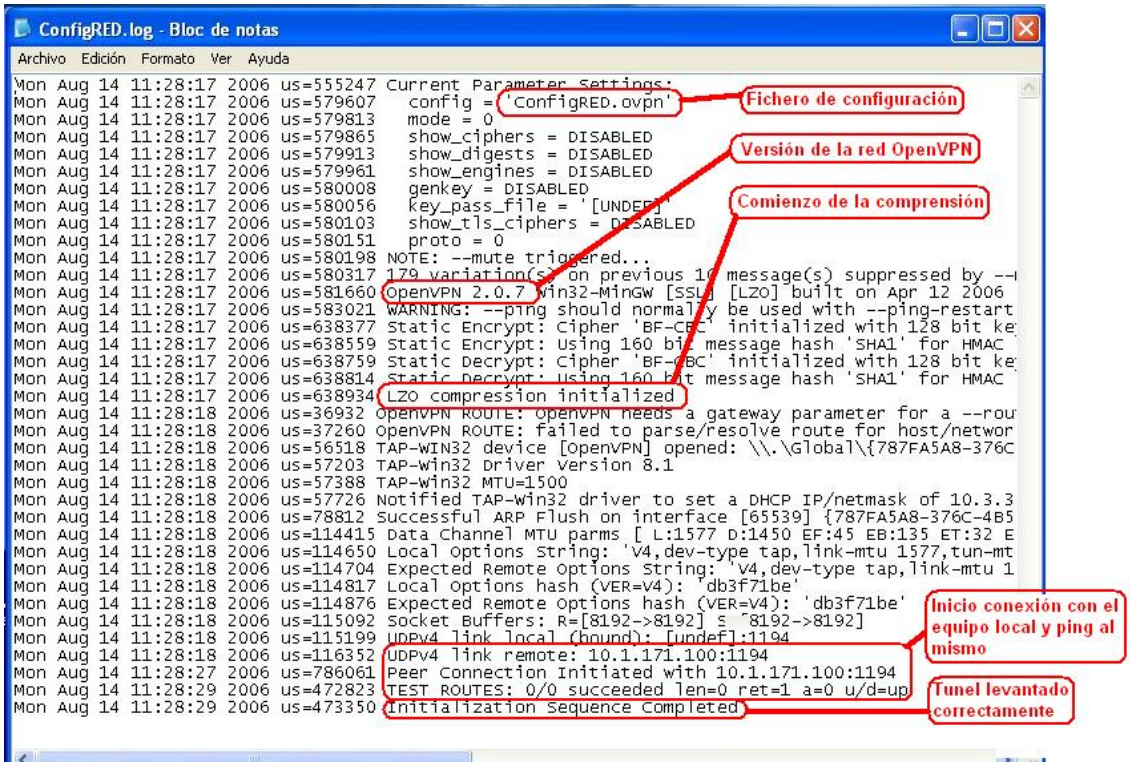

Figura 42: Resultado de la configuración y puesta en marcha de la VPN

D. Configuración modo túnel (tunnel [dev-tun])

Veamos los ficheros de ejemplo que trae OpenVPN y que son dos:

client.ovpn server.ovpn

Que permiten la configuración en modo cliente y modo servidor.

D.1. client.ovpn

El fichero de ejemplo de configuración de un cliente en *modo túnel* es:

```
################################################
# Fichero de ejemplo de OpenVPN del lado del #
# cliente para conectar con un servidor      #
# multi-cliente.                             #
#                                            #
# Esta configuración se puede usar para      #
# múltiples clientes, no obstante cada       #
# cliente debe tener su propios ficheros     #
# de certificados y de claves.               #
#                                            #
# En Windows, puede ser que quiera renombrar #
# este fichero tiene que tener extensión     #
# .ovpn                                      #
################################################

# Específica que somos un cliente y que
# nosotros usaremos ciertas directivas
# de configuración del servidor.
client

# Utiliza el mismo ajuste que se usa en el
# servidor.
# En la mayoría de los sistemas la VPN no
# funcionará si no se deshabilita parcial
# o totalmente el firewall para el interface
# TUN/TAP
;dev tap
dev tun
# Windows necesita el nombre del adaptador
# TAP-Win32 del panel de Conexiones de Red
# si se tiene mas de uno. En XP SP2, puedes
# necesitar deshabilitar el firewall para
# el adaptador TAP
;dev-node MyTap

# Estamos conectados a un servidor TCP
```

```
# o UDP?  Usar la misma configuración
# que en el servidor.
;proto tcp
proto udp

# NombreHost/Ip y puerto del servidor.
# Puede tener múltiples entradas remotas
# para tener cargas balanceadas en los servidores
remote my-server-1 1194
;remote my-server-2 1194

# Elegir al azar un host de la lista
# remota para cargas balanceadas. En
# otro caso se usarán los host en el
# orden especificado.
;remote-random

# Mantener el intento indefinido de resolver
# el nombre del host de el servidor OpenVPN.
# Muy útil en las máquinas que no están
# permanentemente conectadas a Internet
# tales como ordenadores portátiles.
resolv-retry infinite

# La mayoría de los clientes no
# necesitan conectarse a un número
# específico de puerto local.
nobind

# Quita privilegios después de la inicialización
# (solo en sistemas No-Windows)
;user nobody
;group nobody

# Intenta preservar un cierto
# estado a través de los reinicios.
persist-key
persist-tun

# Si está conectado a través de un proxy HTTP
# para alcanzar el actual servidor OpenVPN,
# colocar el server/IP y número del puerto
# aquí. Ver la página del manual de su
# servidor proxy si se requiere autenticación.
;http-proxy-retry # Reintentar si la conexión falla
;http-proxy [proxy server] [proxy port #]
```

```
# Las redes Wireless producen muchos paquetes
# duplicados. Activar este indicador para silenciar
# el aviso de paquetes duplicados.
;mute-replay-warnings

# Parámetros SSL/TLS.

# Ver el fichero de configuración del servidor
# para una mayor descripción. Seria bueno usar
# un par de ficheros separados \emph{.crt/.key} para
# cada cliente. Un solo fichero \emph{ca} se puede usar
# para todos los clientes.
ca ca.crt
cert client.crt
key client.key

# Verifica el certificado del servidor
# comprobando que el certificado tenga
# el campo nsCertType puesto a "server".
# Esta es una importante precaución
# importante para proteger de un ataque
# potencial discutido aquí:
#  http://openvpn.net/howto.html#mitm
#
# Para utilizar esta característica,
# necesitaras generar tus certificados
# del servidor con el campo nsCertType
# fijado en "servidor". El script
# build-key-server situado en el
# directorio easy-rsa hará esto.
;ns-cert-type server

# Si una llave tls-auth se utiliza en el servidor
# entonces cada cliente debe también tener la llave.
;tls-auth ta.key 1

# Selecciona un cifrado criptográfico.
# Si la opción de cifrado se utiliza en el servidor
# entonces debe también especificarlo aquí.
;cipher x

# Habilitar compresión en el enlace VPN.
# No habilitar esta opción a no ser que
# esté habilitada también en el fichero
# de configuración del servidor.
```

```
comp-lzo

# Fijar el fichero de log a verbosity.
verb 3

# Silenciar cada repetición del mensaje
;mute 20
```

D.2. server.ovpn

El fichero de configuración que trae OpenVPN de ejemplo es:

```
################################################
# Ejemplo fichero de configuración OpenVPN 2.0  #
# para un servidor multi-cliente.               #
#                                               #
# Este fichero es para la parte servidor de     #
# una configuración OpenVPN                     #
# varios-clientes <-> un-servidor               #
#                                               #
# OpenVPN también soporta configuraciones       #
# una-máquina <-> una-máquina                   #
# (Ver la página de Ejemplos en su Web          #
# para mas información).                        #
#                                               #
# Esta configuración debe trabajar en sistemas  #
# Windows o Linux/BSD.  Recordar en             #
# Windows entrecomillar el pathnames y usar     #
# doble barra inversa, e.g.:                    #
# "C:\\Program Files\\OpenVPN\\config\\foo.key" #
#                                               #
# Los comentarios irán precedidos por '#' o ';' #
################################################

# ¿En cual dirección IP local debe OpenVPN
# escuchar? (opcional)
;local a.b.c.d

# ¿En qué puerto de TCP/UDP debe OpenVPN escuchar?
# Si desea ejecutar múltiple instancias de OpenVPN
# en la misma máquina, use diferentes número puerto
# para cada uno.  Necesitará abrir este puerto
# en su firewall.
port 1194
```

```
# ¿Servidor TCP o UDP?
;proto tcp
proto udp

# "dev tun" creará un túnel IP ruteador,
# "dev tap" creará un túnel de red.
# Use "dev tap0" si tienes un puente de red
# y tienes precreado un interface virtual tap0
# y el puente es su interface de red.
# Si deseas controlar políticas del acceso
# sobre el VPN, debes crear reglas en el firewall
# para el interfaz de TUN/TAP.
# En los sistemas de no-Windows, puedes dar
# un número explícito de la unidad, tal como tun0.
# En Windows, use "dev-node" para esto.
# En la mayoría de los sistemas, el VPN no funcionará
# a menos que inhabilite parcialmente o completamente
# el firewall para el interfaz de TUN/TAP.
;dev tap
dev tun

# Windows necesita el nombre del adaptador TAP-Win32
# del panel de las conexiones de red si
# tienes más de uno.  En XP SP2 o superiores,
# puede necesitar deshabilitar selectivamente el firewall de
# Windows para el adaptador TAP.  Los sistemas
# No-Windows no necesitan generalmente esto.
;dev-node MyTap

# SSL/TLS certificado raíz (ca), certificado
# (cert), y clave privada (key). Cada cliente
# y el servidor deben tener su propio certificado y fichero clave.
# El servidor y todos los clientes usarán el mismo archivo de \emph{ca}.
#
# Ver el directorio "easy-rsa" para una serie
# de scripts para la generación de los certificados
# RSA y las claves privadas. Recordar usar un único
# nombre común para el servidor y cada uno de los certificados
# del cliente.
#
# Cualquier sistema administración X509 puede ser utilizado.
# OpenVPN puede también utilizar un archivo clave ajustado al formato PKCS  #12
# (véase la directiva "pkcs12" en la página del manual).
ca ca.crt
cert server.crt
key server.key  # Este archivo se debe mantener secreto
```

```
# Parámetros Diffie Hellman.
# Generar uno propio con:
#   openssl dhparam -out dh1024.pem 1024
# Sustituya 1024 por 2048 si quiere usar
# claves de 2048 bit.
dh dh1024.pem

# Configura el modo servidor y suministra una subred
# de VPN para OpenVPN a las direcciones del cliente.
# El servidor tomará 10.8.0.1 para sí mismo,
# el resto será puesto a disposición los clientes.
# Cada cliente podrá alcanzar el servidor
# en 10.8.0.1. Comenta esta línea si se quiere tender
# un puente de red. Ver la página del manual para más Información.
server 10.8.0.0 255.255.255.0

# Mantiene un registro asociaciones de
# clientes <-> direcciones IP virtuales
# en este fichero. Si OpenVPN se detiene
# o reinicia, al reconectar los clientes
# se vuelve a asignar la misma dirección
# IP virtual desde la combinación
# previamente asignada.
ifconfig-pool-persist ipp.txt

# Configura el modo servidor para tender un puente red.
# Debe primero utilizar la capacidad que tiene un puente sobre
# su OS para tender un puente sobre el interfaz TAP
# con el interfaz del NIC de red.
# Entonces debes fijar manualmente IP/netmask
# en el interfaz del puente, aquí se
# asume 10.8.0.4/255.255.255.0.  Finalmente nosotros
# debe asignar un rango de IP a un lado en esta subred
# (comienzo=10.8.0.50 final=10.8.0.100) para asignar
# a los clientes que conectan.  Dejar esta línea comentada
# a menos que desee tender un puente de red.
;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100

# Impulsar rutas en el cliente para permitirle
# alcanzar otras subredes privadas detrás del servidor.
# Recordar que las subredes privadas también necesitan
# conocer como encaminar los paquetes del cliente
# OpenVPN (10.8.0.0/255.255.255.0) de nuevo al servidor
# OpenVPN
;push "route 192.168.10.0 255.255.255.0"
```

```
;push "route 192.168.20.0 255.255.255.0"

# Para asignar una dirección IP específica
# a un específico cliente o si un cliente
# que conecta tiene una subred privada
# detrás de él que deba también tener
# acceso a la VPN, usar el subdirectorio 'CCD'
# para ficheros de configuración cliente-específico
# (véase la página del manual para más información).

# EJEMPLO: Suponer el cliente
# que tiene el certificado con el nombre común
# "Thelonious", también tiene una subred pequeña
# detrás de su máquina que conecta como
# 192.168.40.128/255.255.266.248
# Primero, descomentar estas líneas:
;client-config-dir ccd
;route 192.168.40.128 255.255.255.248
# Crear el fichero ccd/Thelonious con la línea:
#   iroute 192.168.40.128 255.255.255.248
# Esto permitirá a la subred privada Thelonious
# el acceso a la VPN. Este ejemplo trabaja solamente
# si estás encaminando, no tendiendo un puente,
# es decir estas usando directivas "dev tun" y "server".

# EJEMPLO: Suponer que desea dar a Thelonious
# una dirección IP fija de VPN 10.9.0.1.
# primero decomentar las líneas:
;client-config-dir ccd
;route 10.9.0.0 255.255.255.252
# entonces agrega esta línea al ccd/Thelonious:
#   ifconfig-push 10.9.0.1 10.9.0.2

# Supone que desea permitir diferentes  políticas de
# acceso al firewall para diversos grupos de clientes.
# Hay dos métodos:
# (1) Ejecución de múltiples demonios OpenVPN, uno para
#     cada grupo, y firewall interfaz TUN/TAP para cada
#     grupo/demonio apropiadamente.
# (2) (Avanzado) Crear un script modificado dinámicamente
#     para que el firewall responda al acceso de diversos
#     clientes.  Ver las páginas del manual para más
#     información en aprender-trata los scripts.
;learn-address ./script

# Si está habilitado, esta directiva configurará
```

```
# todos los clientes para volver a redirigir su
# puerta de enlace por defecto a través de la VPN,
# causando que todo el tráfico IP tal como los navegadores y
# y operaciones de búsqueda del DNS a pasar a través del VPN
# (la máquina del servidor de OpenVPN puede necesitar
# NAT al interfaz de TUN/TAP al objeto de que Internet
# trabaje correctamente).
# ADVERTENCIA: Puede romperse la configuración de la
# red de clientes si el servidor DHCP local sirve paquetes
# que consiguen encaminarse a través del túnel.
# Solución: cerciórese de que el servidor local de DHCP
# del cliente sea accesible vía a una ruta más específica
# que el enrutado por defecto 0.0.0.0/0.0.0.0.
;push "redirect-gateway"

# Ciertos ajustes específicos de Windows
# de la red se pueden colocar en los clientes,
# tales como DNS o WINS servidor de direcciones.
# ADVERTENCIA:
# http://openvpn.net/faq.html#dhcpcaveats
;push "dhcp-option DNS 10.8.0.1"
;push "dhcp-option WINS 10.8.0.1"

# Descomentar esta directiva para permitir que diversos clientes
# puedan "ver" cada uno a otros.
# Por defecto, los clientes verán solamente el servidor.
# Para forzar a los clientes a ver solamente al servidor,
# también necesitarás configurar el servidor firewall
# con el interfaz TUN/TAP.
;client-to-client

# Descomentar esta directiva si múltiples
# conectan con el mismo fichero de certificado/clave
# o nombre común. Esto solo se recomienda con propósito
# de testeo. En producción use, cada cliente debe tener
# su propio par de certificado/clave.
#
# SI NO HAS GENERADO UNA PAREJA CERTIFICADO/CLAVE
# INDIVIDUAL PARA CADA CLIENTE,
# CADA UNO TIENE QUE TENER SU PROPIO "NOMBRE COMÚN ÚNICO",
# DESCOMENTAR ESTA LÍNEA.
;duplicate-cn

# La directiva keepalive envía mensajes ping-vivo hacia
# adelante y hacia atrás de modo que en el enlace cada
# lado sepa cuando el otra se ha desconectado.
```

```
# Ping cada 10 segundos, asume que el par remoto está
# desconectado si no recibe ningún ping durante
# un período de tiempo de 120 segundos.
keepalive 10 120

# Para seguridad adicional más allá de lo proporcionado
# por SSL/TLS, cree un "HMAC firewall"
# para ayudar a bloquear ataques del DOS y desbordar
# el puerto UDP.
#
# Generar con:
#   openvpn --genkey --secret ta.key
#
# El servidor y cada cliente deben tener
# una copia de esta llave.
# El segundo parámetro debe ser '0'
# en el servidor y  '1' en los clientes.
;tls-auth ta.key 0 # Este fichero es secreto

# Selecciona un cifrado criptográfico.
# Este opción del config se debe copiar a
# el archivo de config del cliente también.
;cipher BF-CBC        # Blowfish (default)
;cipher AES-128-CBC   # AES
;cipher DES-EDE3-CBC  # Triple-DES

# Permite la compresión en el enlace VPN.
# Si lo permite aquí, debe también
# permitirlo en el archivo de config de cliente.
comp-lzo

# El número máximo de clientes concurrentes conectados
# que deseamos permitir.
;max-clients 100


# Es una buena idea el reducir los privilegios
# de los demonios OpenVPN después de la inicialización
#
# Puede descomentar esta línea si el sistemas es
# distinto de Windows.
;user nobody
;group nobody

# La opción de persistencia quiere evitar
# el acceso a ciertos recursos en el reinicio, que
```

```
# no serán accesibles por que los privilegios
# disminuyen.
persist-key
persist-tun

# Hace salir el estado corto a un archivo
# que muestra las conexiones actuales,
# truncado y reescrito cada minuto.
status openvpn-status.log

# Por defecto, los mensajes del registro irán al syslog (o
# en Windows, si funcionan como servicio, van a al directorio
# "\Program Files\OpenVPN\log").
# Utiliza el registro o registro-añadido para eliminar este defecto.
# "log" truncará el fichero en el arranque diario de OpenVPN,
# mientras que "log-append" añadirá a él.  Utilizar uno
# o el otro (pero no ambos).

;log         openvpn.log
;log-append  openvpn.log

# Fija el nivel apropiado de
# verbosity del archivo de log.
#
# 0 es silencioso, a excepción de errores fatales
# 4 es razonable para el uso general
# 5 y 6 puede ayudar a eliminar errores de problemas de la conexión
# 9 es extremadamente prolijo
verb 3

# Silencia los mensajes repetidos.  A los 20
# mensajes secuenciales de la misma categoría
# de mensaje serán eliminados del registro.
;mute 20
```

D.3. Configuración del servidor (server.conf)

La configuración utilizada en la *máquina virtual* con *Linux Red Hat*, en formato reducido, es la siguiente:

```
port 1194
proto tcp
dev tun
ca /etc/openvpn-2.0.7/easy-rsa/keys/ca.crt
cert /etc/openvpn-2.0.7/easy-rsa/keys/server.crt
```

```
key /etc/openvpn-2.0.7/easy-rsa/keys/server.key
dh /etc/openvpn-2.0.7/easy-rsa/keys/dh1024.pem
server 10.3.3.0 255.255.255.0
ifconfig-pool-persist ipp.txt
keepalive 10 120
comp-lzo
persist-key
persist-tun
status openvpn-status.log
verb 3
```

D.4. Configuración de los clientes (ClientLinux.ovpn)

Los tres clientes se configuran igual, con la excepción del certificado y la clave que son únicos para cada uno de ellos. El fichero de configuración es:

```
client
dev tun
proto tcp
remote 10.1.171.120 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca   "C:\\Archivos de programa\\OpenVPN\\easy-rsa\\Keys\\ca.crt"
cert "C:\\Archivos de programa\\OpenVPN\\easy-rsa\\Keys\\lin-cli-1.crt"
key  "C:\\Archivos de programa\\OpenVPN\\easy-rsa\\Keys\\lin-cli-1.key"
comp-lzo
verb 3
```

E. Configuración de los firewalls

La configuración de los firewalls para verificar la conexión en puente (*bridge*), la vamos a realizar de tal forma que solo se permita el paso de la VPN, el resto de tráfico será rechazado.

Pulsar: *Inicio->Panel de Control->Firewall de Windows*. Obtendremos la pantalla de la figura 43.

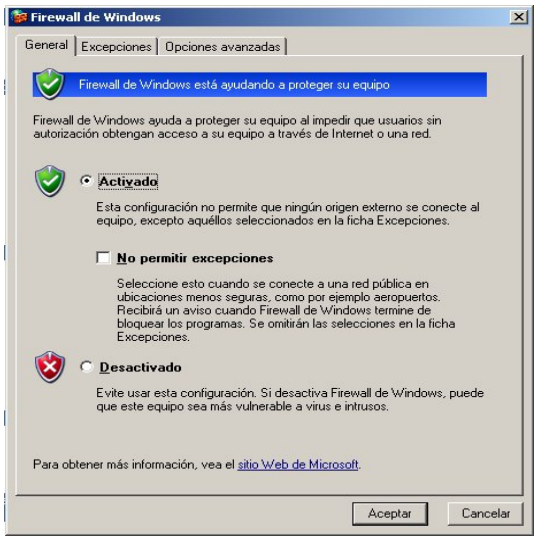

Figura 43: Activación del Firewall

Si el firewall está desactivado lo activaremos seleccionando la opción: *Activado*.

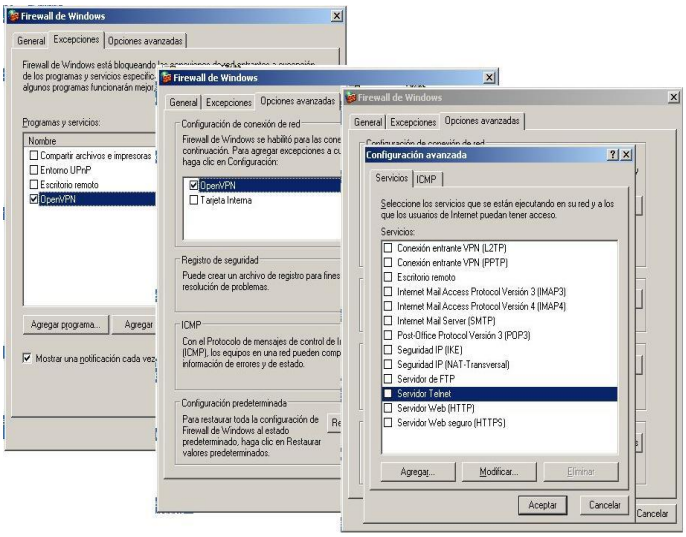

Figura 44: Configuración del Firewall

Como se puede apreciar en la figura 44, seleccionamos la pestaña *Excepciones* y deshabilitamos todos los programas y servicios a excepción de OpenVPN y en la pestaña *Opciones avanzadas* seleccionamos **OpenVpn** y pulsamos *configuración*, seguidamente desactivamos cualquier opción que estuviera activada.

F. Generación de las claves y de los certificados

F.1. Generación de claves estáticas

F.1.1. En Windows

En Windows podemos utilizar una utilidad que trae para generar de forma automática la clave, para ello accedemos al programa siguiendo la secuencia de pulsaciones indicadas:

inicio->Todos los programas->OpenVPN->Generate a static OpenVPN key tal y como se indica:

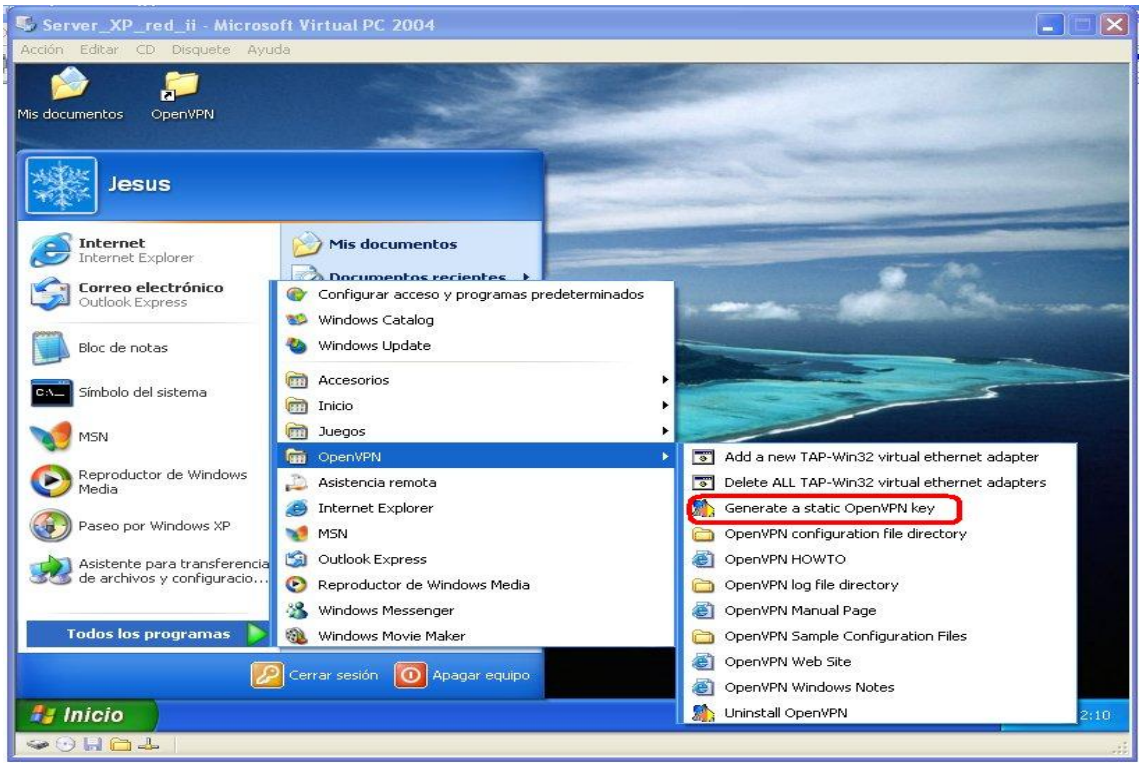

Figura 45: Acceso al programa de generación de claves estáticas en Windows

Una vez seleccionada la opción obtendremos una ventana indicándonos que se ha generado la clave y la ubicación de la misma que podemos observar en la figura 46.

F.1.2. Línea de Comandos

Si por el contrario queremos generar la clave desde la línea de comandos, tecleare-mos la siguiente instrucción, dentro del directorio

```
C:\Archivos de programa\OpenVPN\config
```

(también es válido para la generación de la clave en *Linux*):

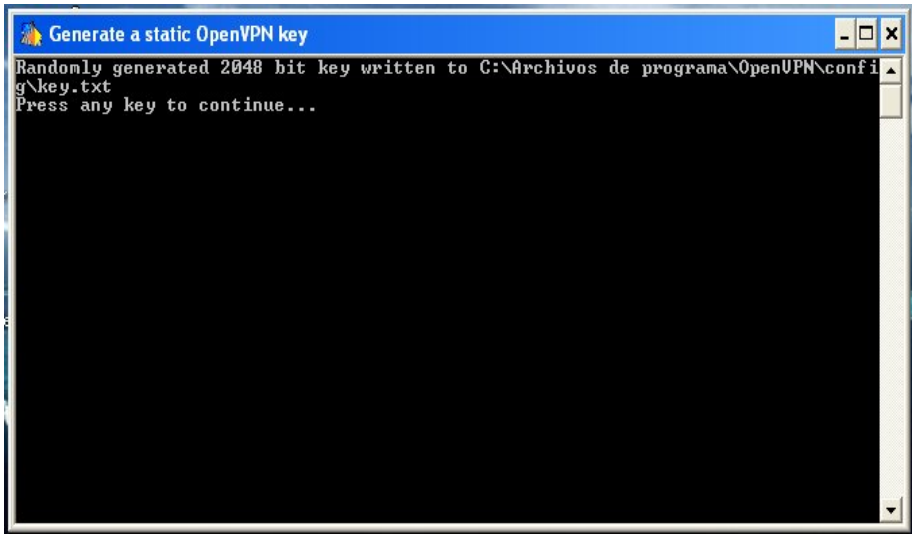

Figura 46: Resultado de la Generación

OpenVPN –genkey –secret key.txt

Obteniendo la imagen siguiente:

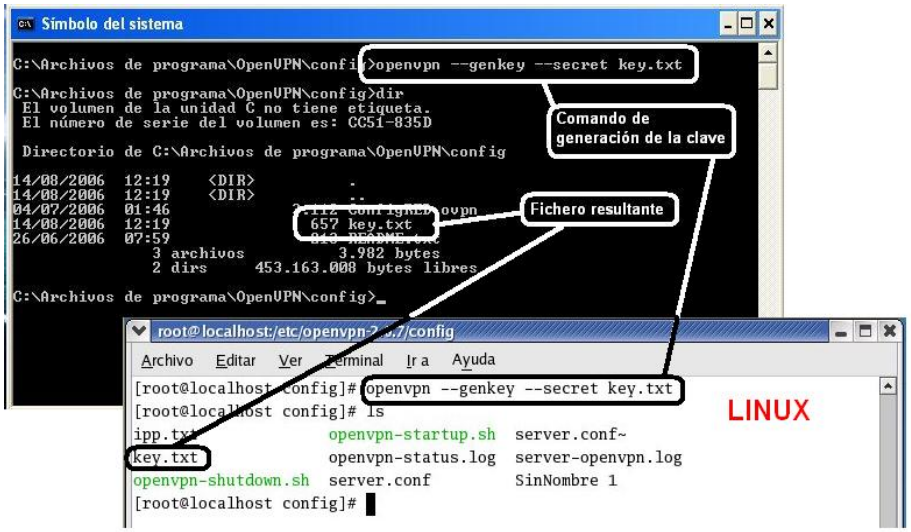

Figura 47: Generación de la clave desde la línea de comandos

F.2. Generación de los Certificados

Esta operación la realizaremos sobre la plataforma de Linux. El proceso en Windows es similar.

F.2.1. Generación del certificado maestro y clave de la Autoridad Certificadora (CA)

Nos situamos en el directorio:

```
/usr/openvpn-2.0.7/easy-rsa
```

Y una vez situados dentro de este directorio, ejecutaremos los siguientes comandos:

```
. ./vars
./clean-all
./build-ca
```

El último comando genera una pantalla interactiva que nos pide unos datos, tal y como se muestra:

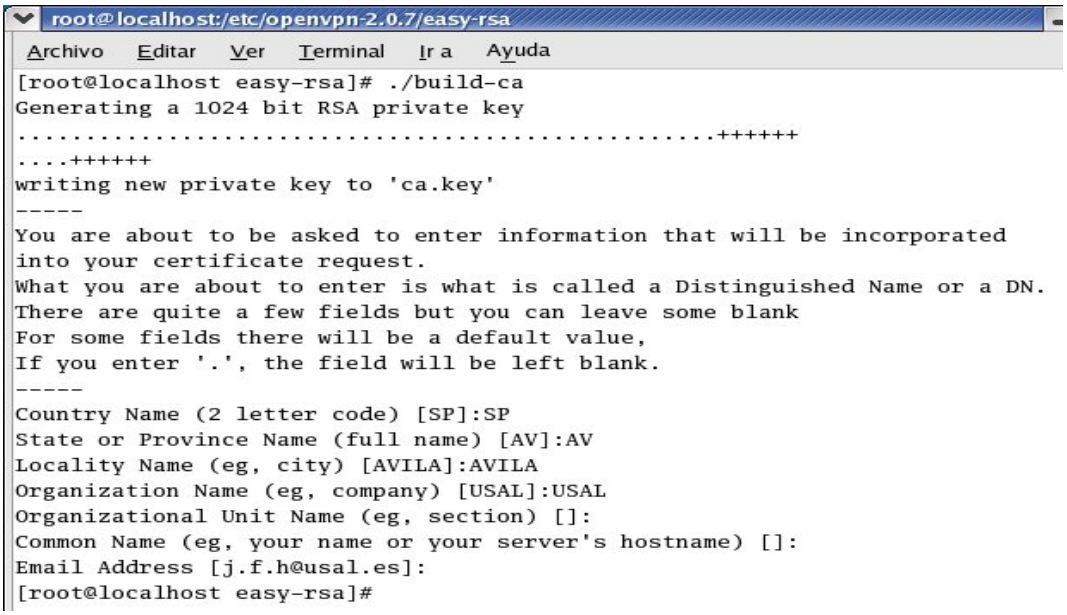

Figura 48: Generación certificado Autoridad Certificadora (CA)

Esto genera unos ficheros con nombres *ca.key* y *ca.crt* que moveremos a un directorio creado previamente *keys* y cuyo contenido lo podemos ver en la figura 49.

F.2.2. Generación del certificado y clave del Servidor

Ejecutar el siguiente comando:

```
./build-key-server server
```

Esto genera los ficheros *server.crt*, *server.csr* y *server.key* que también moveremos a la ubicación indicada con anterioridad. El contenido del certificado se expone en la figura 51.

Figura 49: Contenido del certificado CA

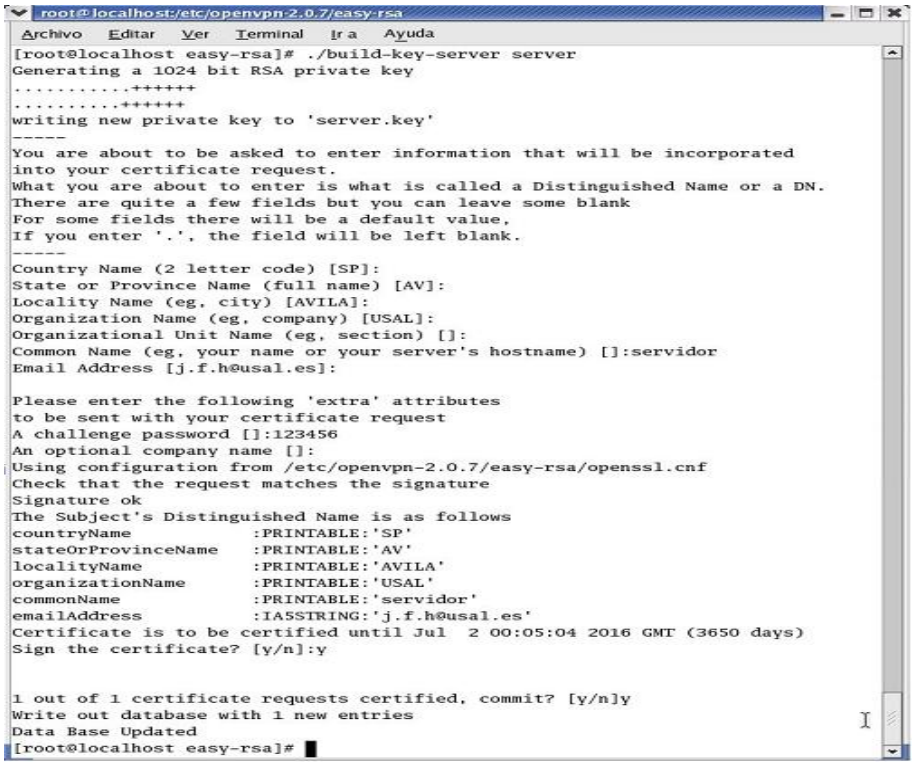

Figura 50: Generación certificado y clave servidor

F.2.3. Generación del certificado y clave para tres clientes

Ejecutar el siguiente comando:

```
./build-key cliente1
./build-key cliente2
./build-key cliente3
```

El resultado del script es muy similar al obtenido para el servidor. Se crean dos ficheros *cliente1.crt*, *cliente1.key*, *cliente2.crt*, *cliente2.key*, *cliente3.crt* y *cliente3.key* que pasaremos al directorio pertinente.

Nombre fichero	Necesario en	Propósito	Secreto
ca.crt	Servidor + todos los clientes	Certificado raíz CA	NO
ca.key	Máquina que firma el certificado	Clave raíz CA	SI
dh{n}.pem	Solo en el servidor	Parámetros Diffie Hellman	NO
server.crt	Solo en el servidor	Certificado del servidor	NO
server.key	Solo en el servidor	Clave del servidor	SI
cliente1.crt	Solo en el cliente 1	Certificado del cliente 1	NO
cliente1.key	Solo en el cliente 1	Clave del cliente 1	SI
cliente2.crt	Solo en el cliente 2	Certificado del cliente 2	NO
cliente2.key	Solo en el cliente 2	Clave del cliente 2	SI
cliente3.crt	Solo en el cliente 3	Certificado del cliente 3	NO
cliente3.key	Solo en el cliente 3	Clave del cliente 3	SI

Tabla 2: Ficheros de claves y certificados [13]

F.2.4. Generar parámetros de Diffie Hellman

Ejecutar el comando:

```
./build-dh
```

La pantalla de generación es la de la figura 52.

Esto crea el fichero *dh1024.pem*, que será movido como los anteriores al directorio creado a tal efecto.

Con esto se han generado todos los ficheros necesarios para trabajar con los certificados digitales. Seguidamente daremos una tabla con los ficheros, sus ubicaciones y si han de ser secretos o no (Tabla 2).

Tanto en el servidor como en los clientes se creará un directorio con nombre *keys*, donde su ubicarán todos los ficheros que correspondan según la tabla 2.

Figura 51: Contenido del certificado del servidor

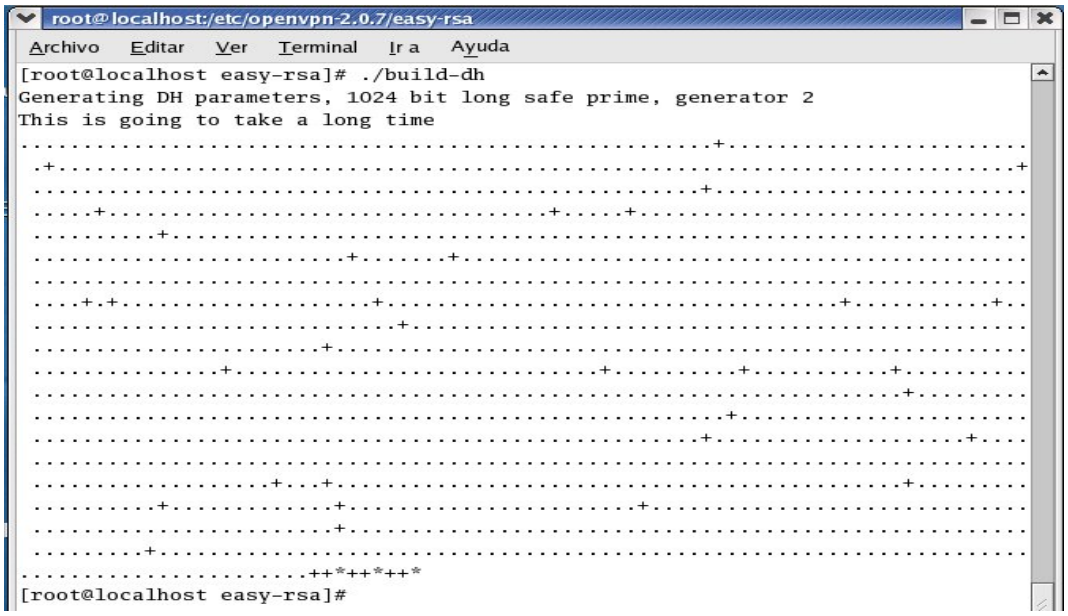

Figura 52: Generación del fichero de Diffie Hellman

G. Script de arranque y parada en Linux

G.1. Script de Arranque

```
#!/bin/bash

# directorio de openvpn para los ficheros de configuración
dir=/etc/openvpn-2.0.7/config
filelog=/etc/openvpn-2.0.7/config/server-openvpn.log

# cargar el modulo del kernerl TUN/TAP
modprobe tun

# habilitar IP forwardig
# echo 1 > /proc/sys/net/ipv4/ipforward

# invocar openvpn
/etc/openvpn-2.0.7/openvpn --cd $dir --log $filelog --daemon --config server.conf
```

G.2. Script de Parada

```
#!/bin/bash
# Parar todos los procesos openvpn

killall -TERM openvpn
```

H. Configuración del router

Para poder trabajar correctamente con OpenVPN en una red donde tenemos un router que filtra los paquetes que nos llegan del exterior, es preciso configurar el mismo. Esta configuración se compone de 3 partes:

- Enrutamiento de los paquetes de OpenVPN.
- Apertura del puerto de OpenVPN.
- NAT[3] al equipo servidor.

.

El router que vamos a configurar es un Prestige 643 de ZyXEL, aunque cualquier otro se configurará de forma similar.

H.1. Enrutamiento de los paquetes de OpenVPN

En este paso vamos a indicarle al router lo que tiene que hacer con los paquetes que le llegan del extremo del túnel y tiene que volver por él. Como la dirección IP del extremo del túnel no es una dirección de la red física, el router no sabe donde enviarlo y por defecto lo pasará al exterior que es donde envía los paquetes que no son de su red. Le tenemos que decir que todo aquello que vaya al extremo del túnel de la VPN, vaya al dispositivo de red del servidor. En nuestro caso enviaremos todo lo que sea de la red 10.3.3.0 al servidor cuya dirección es 10.1.171.105 y que es donde está el extremo del túnel de la VPN.

Una vez conectados al router, seguimos la siguiente secuencia de comandos: 12(Static Routing Setup)->1(IP Static Route)->1(OpenVPN), tal y como se muestra en la figura 53.

Las opciones a aplicar son:

```
Route #: 1 (n° de la ruta)
Route Name= OpenVPN (Nombre de la ruta)
Active= Yes (Si esta la ruta activa o no)
Destination IP Address= 10.3.3.0 (Dirección IP destino)
IP Subnet Mask= 255.255.255.0 (Mascara de subred)
Gateway IP Address=10.1.171.105 (punto de destino de los paquetes)
Metric= 1
Private= Yes
```

[3]**NAT (Network Address Translation - Traducción de Dirección de Red**) Se utiliza una o más direcciones IP para conectar varios ordenadores a otra red (normalmente a Internet), los cuales tiene una dirección IP completamente distinta. En este caso utilizaremos *NAT estático* que realiza un mapeo en la que una dirección IP privada se traduce a una correspondiente dirección IP pública de forma unívoca. Normalmente se utiliza cuando un dispositivo necesita ser accesible desde fuera de la red privada [18].

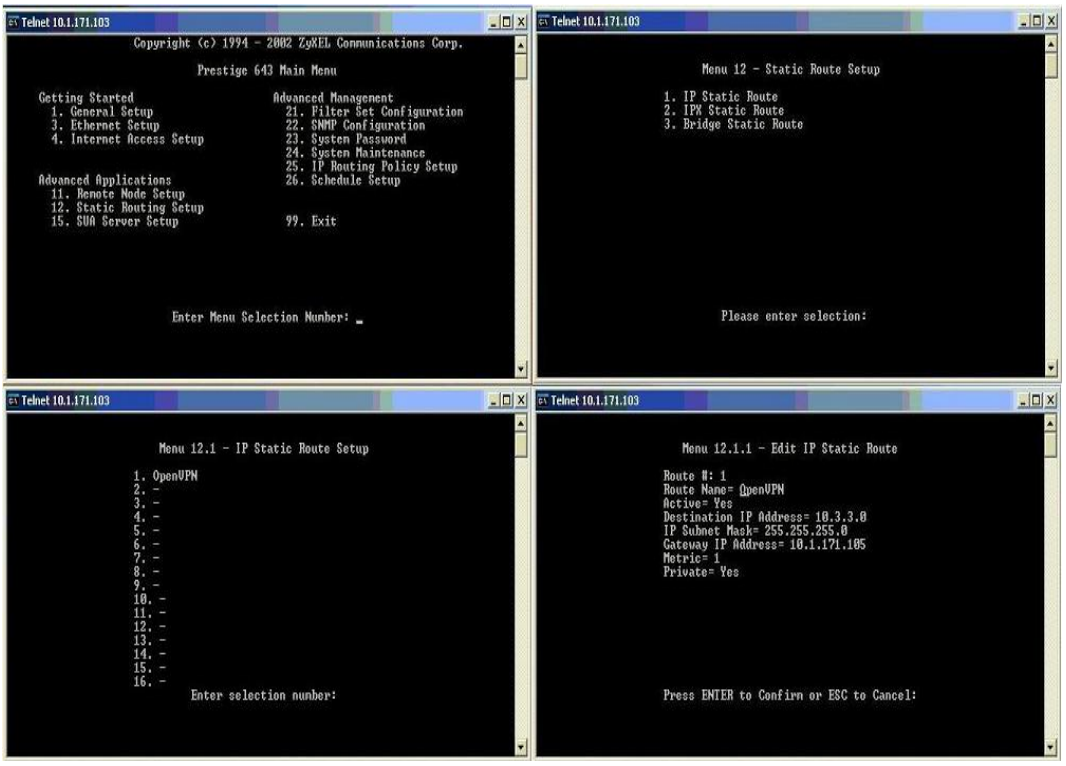

Figura 53: Configuración del router I

H.2. Apertura del puerto y NAT al servidor

En este router, estas dos operaciones se realizan de forma simultánea, al mismo tiempo que se abre el puerto, se indica a que dirección IP interna se envían los paquetes. Para realizar la configuración, desde la pantalla principal seleccionamos la opción 15 (SUA Server Setup) y especificamos el puerto a abrir (1194) y la dirección donde se enviarán los paquetes (10.1.171.105).

En este tipo de router al abrir el puerto, se abre para todas las tramas (TCP, UDP, etc) con lo que no hace falta especificar nada más. Observar que también se ha abierto el puerto 21 (ftp) y los puertos 5800 y 5900 para poder administrar de forma remota el servidor mediante la aplicación *VNC*. En la figura 54 tenemos esta opción del router configurada.

Figura 54: Configuración del router II

Referencias

[1] Isaac Clerencia. "redes privadas virtuales". In *Warp Networks S.L.*, 2005. [citado 19 septiembre 2006]. [*Citado en págs. 2 y 5.*]

[2] Computer Security Division. "national vulnerability database". http://nvd.nist.gov/, 2006. [citado 19 septiembre 2006]. [*Citado en pág. 32.*]

[3] Nick G. Duffield, Pawan Goyal, Albert G. Greenberg, Partho Pratim Mishra, K. K. Ramakrishnan, and Jacobus E. van der Merive. "a flexible model for resource management in virtual private networks". In *SIGCOMM*, pages 95–108, 1999. [citado 19 septiembre 2006]. [*Citado en pág. 26.*]

[4] Markus Feilner. *"OpenVPN - Building and Integrating Virtual Private Network"*. Packt Publishing Ltd., 2006. ISBN 1-904811-85-X [citado 19 septiembre 2006]. [*Citado en págs. 4, 12 y 14.*]

[5] Paul Ferguson and Geoff Huston. "what is a vpn: Part i". In *Protocol Journal*, pages vol. 1, no. 1, June 1998. [citado 19 septiembre 2006]. [*Citado en pág. 2.*]

[6] Paul Ferguson and Geoff Huston. "what is a vpn: Part ii". In *Protocol Journal*, pages vol. 1, no. 2, September 1998. [citado 19 septiembre 2006]. [*Citado en pág. 2.*]

[7] Damián Ferrer. "vpn: Una introducción a las redes privadas virtuales". In *VPN: Una introducción a las Redes Privadas Virtuales*, 2005. [citado 19 septiembre 2006]. [*Citado en pág. 4.*]

[8] Charlie Hosner. "openvpn and the ssl vpn revolution". http://www.sans.org/reading_room/whitepapers/vpns/1459.php. [citado 19 septiembre 2006]. [*Citado en pág. 38.*]

[9] Charlie Hosner. "ssl vpns and openvpn: A lot of lies and a shred of truth". Technical report, newsforge, http://software.newsforge.com/print.pl?sid=05/09/22/164231, 2005. [citado 19 septiembre 2006]. [*Citado en pág. 15.*]

[10] Ray Hunt and Chris Rodgers. "virtual private networks: Strong security at what cost¿'. http://citeseer.ist.psu.edu/hunt01virtual.html. [citado 19 septiembre 2006]. [*Citado en págs. 2, 4 y 5.*]

[11] Telindus High-Tech Institute. "openvpn 101: introduction to openvpn". http://openvpn.net/papers/openvpn-101.pdf, 2004. [citado 19 septiembre 2006]. [*Citado en pág. 25.*]

[12] S. Kent and R. Atkinson. "security architecture for the internet protocol", November 1998. RFC 2401 [citado 19 septiembre 2006]. [*Citado en pág. 9.*]

[13] Linksys. *"VPN usando routers Linksys por banda ancha con IP dinámica"*. [citado 19 septiembre 2006]. [*Citado en págs. 6 y 83.*]

[14] Marco Hernáiz Mayo. "ipv6 e ipsec - seguridad en redes telemáticas". http://asignaturas.diatel.upm.es/seguridad/trabajos/trabajos/ipv6c.pdf. [citado 19 septiembre 2006]. [*Citado en pág. 9.*]

[15] Coral Martínez Millas. "vpn redes virtuales privadas - seguridad en redes telemáticas". http://asignaturas.diatel.upm.es/seguridad/trabajos/trabajos/vpn.pdf. [citado 19 septiembre 2006]. [*Citado en págs. 2 y 9.*]

[16] Virtual Private Networks. "network working group a. nagarajan, ed. request for comments: 3809 juniper networks category: Informational june 2004 generic requirements for provider provisioned". http://citeseer.ist.psu.edu/660271.html. [citado 19 septiembre 2006]. [*Citado en pág. 3.*]

[17] Gustav Rosenbaum, William Lau, and Sanjay Jha. "recent directions in virtual private network solutions". http://citeseer.ist.psu.edu/617359.html. [citado 19 septiembre 2006]. [*Citado en pág. 5.*]

[18] Inc. Wikimedia Foundation. "wikipedia". http://es.wikipedia.org/wiki/Portada. [citado 19 septiembre 2006]. [*Citado en págs. 8, 9, 10, 11, 12, 32, 35, 49 y 93.*]

[19] Paul Wouters and Ken Bantoft. *"Building and Integrating Virtual Private Networks with Openswan"*. Packt Publishing Ltd., 2006. ISBN 1-904811-25-6 [citado 19 septiembre 2006]. [*Citado en pág. 8.*]

[20] James Yonan. "openvpn". http://openvpn.net/. [citado 19 septiembre 2006]. [*Citado en págs. 16 y 43.*]

[21] James Yonan. "the user-space vpn and openvpn". http://openvpn.net/papers/BLUG-talk/BLUG-talk.ppt, 2003. [citado 19 septiembre 2006]. [*Citado en pág. 14.*]