TRABAJO FIN DE CARRERA.

TÍTULO DEL TFC: Implementación de una red privada virtual para el control remoto de equipos de laboratorio.

TITULACIÓN: Ingeniería Técnica de Telecomunicaciones, especialidad en Telemática.

AUTORES: Rafael Pinilla Vico
Óscar Sánchez Sánchez

DIRECTOR: Pere Bruna

FECHA: 21 de diciembre de 2009

TÍTULO: Implementación de una red privada virtual para el control remoto de equipos de laboratorio.

AUTORES: Rafael Pinilla Vico
Óscar Sánchez Sánchez

DIRECTOR: Pere Bruna

FECHA: 21 de diciembre de 2009

Resumen

Las redes privadas virtuales (VPN en sus siglas en inglés) consisten en redes en las que algunos o todos los equipos de los que forman parte están unidos por conexiones o circuitos virtuales, en lugar de estarlo físicamente. Esto, a parte de comportar una reducción de costes, permite aumentar la seguridad de las comunicaciones y facilitar la conexión entre máquinas de distintos rangos con IP's fijas o dinámicas.

El objetivo de este trabajo es la puesta en marcha de una VPN del grupo de investigación de Materiales Metaestables y Nanoestructurados del Departamento de Física Aplicada de la UPC. Este grupo está formado por profesores de distintos departamentos, y ubicados en distintas escuelas, con lo que los rangos IP de los ordenadores son diferentes. Además se dispone de laboratorios con equipos controlados informáticamente. Por tanto el trabajo se estructurará de la siguiente forma:

1. Analizar la seguridad actual de la red frente a ataques externos. Se estudiará los puntos débiles, y la forma de mejorarlos.
2. Instalación de la VPN en una red multiplataforma, en la que se tendrán que analizar los pros y contras de las distintas opciones disponibles. Se deberá decidir el programa a usar (libre o comercial) y sobre la conveniencia de tener un servidor propio o uno externo.
3. Establecer un protocolo detallado sobre el alta de nuevos ordenadores en la red, así como el de la baja de algún equipo obsoleto.
4. Implementar un sistema que permita el control remoto a través de la VPN de los equipos de laboratorio conectados a la VPN.
5. Potenciar y optimizar los recursos de cada ordenador con la VPN.

Al finalizar el trabajo, se deberá establecer una red privada virtual segura que permita maximizar los recursos de cada ordenador.

TITTLE: Implementing a virtual private network to laboratory computer remote.

AUTHOR: Rafael Pinilla Vico
Óscar Sánchez Sánchez

DIRECTOR: Pere Bruna

DATE: 21 de diciembre de 2009

Overview

The virtual private network (VPN) is a network in which some or all of its computers are connected by virtual connections instead of by physical cables. This fact, aside of entailing a reduction of costs in the establishment of a network, allows to increase the security of the communications and to facilitate the connection among machines of different ranks with static or dynamic IP's.

The goal of the present work is the implementation of the VPN of the Metastable and Nanostructured Materials research group of the Department of Applied Physics of the UPC. This group of research it is constituted by teachers of several departments and placed in different schools so the ranks of the several computers are different. Besides it has several laboratories with different equipment controlled by a computer, so the work will be structured as follows:

1. To analyze the security of the current network to external attacks, updating the operative systems installed if needed. Detect the weak points and to study how to minimize them.
2. Installation of the VPN in a multi platform network, so that the pros and cons of the several available options will have to be analyzed. Basically it will have to be decided among which software to use (free or commercial) and about the convenience to have an own server or to use an external one.
3. To establish a detailed protocol about the registration of new computers in the network, as well as of how to unregister some equipment that has come off obsolete.
4. Implementing a system that allows computers of the VPN to take the remote control through the VPN of the several equipments of the laboratory.
5. Exploring how to promote and optimize the resources of each computer with the VPN.

At the end of the work, a private and safe network will be created and it will allow the maximization of the resources of each computer, from the point of view of execution of software as well as of control of external hardware.

DEDICATORIA.

Rafael Pinilla Vico

Dedico el proyecto final de carrera:

A mi novia, que ha tenido mucha paciencia durante todos estos años, y ha sido una ayuda fundamental para conseguir esta gran meta.

A mis padres, y hermana, que siempre han confiado en mí, y me han animado a seguir adelante con mis estudios, y con todo lo que me he propuesto.

Gracias a todos.

Óscar Sánchez Sánchez

Dedico este TFC a mi hija, a mi madre, y especialmente a mi mujer, que con su apoyo y paciencia me ha ayudado día a día a conseguir una de mis mayores metas personales. Gracias a las tres.

1. INTRODUCCIÓN.

1.1. Necesidad de crear una VPN.

Hoy en día, para las empresas resulta fundamental poder disponer de comunicaciones a través de las redes de información. Con frecuencia, muchas de estas empresas pueden crear redes (físicas o virtuales) dentro de su propio ámbito local, lo que permite poder implantar y gestionar políticas de seguridad (privacidad e integridad de los datos). Sin embargo, esto no siempre es posible dado que existe la casuística en la cual las sedes de estas empresas se encuentran separadas físicamente, lo que dificulta poder implantar la solución anterior por restricciones tecnológicas. Por sus costes económicos y su escalabilidad se opta por la construcción de enlaces privados. El uso de redes privadas supone la implantación de un nuevo enlace cada vez que se quiera unir un nuevo miembro a la red de una organización, con los consiguientes elevados costes. La solución en estos casos es hacer llegar este tráfico de red atravesando redes públicas (como Internet) con la consecuente vulnerabilidad a los ataques de usuarios mal intencionados. Para evitar esto, es imprescindible proteger el tráfico que circulará por las redes públicas.

La solución más óptima son las Redes Privadas Virtuales (VPN en su acrónimo en inglés de “Virtual Private Network”), que permiten crear conexiones seguras a través de la infraestructura de redes públicas mediante una herramienta de comunicación segura, bajo coste y alta privacidad. Una VPN permite crear enlaces punto a punto, conectar distintas redes locales entre sí, o permitir por ejemplo a un empleado remoto conectarse a la sede de su empresa desde cualquier acceso a Internet (caso conocido como roadwarrior).

La implementación de una VPN esta basada en “túneles”, donde la información que atraviesa las redes públicas viaja encapsulada en paquetes de forma que el contenido resulta invisible hasta llegar a su destino. Una vez el paquete llega a su destino, este se desencapsula y la información vuelve a ser visible.

Desde el grupo de investigación de Materiales Metaestables y Nanoestructurados del Departamento de Física Aplicada de la UPC, se planteó la necesidad de poder disponer de una red privada segura que permitiera conectar todos sus ordenadores entre sí, para poder intercambiar todo el material del grupo, así como poder controlar remotamente los ordenadores.

Se planteó la situación estudiando el escenario del grupo de investigación, y aunque se trata de un caso en el que se podría haber aplicado una configuración de red a nivel local, las restricciones aplicadas por el departamento Informático de la UPC han desembocado en la implantación de una VPN.

Se ha tomado cada laboratorio o despacho, como una sede independiente que se conecta a un servidor de VPN.

1.2. Ventajas de una VPN.

Implantar una VPN tiene varios puntos beneficiosos:

- Integridad, confidencialidad y encriptación de datos. La integridad de los datos hace referencia a que un mensaje enviado no pueda ser alterado. La confidencialidad se refiere a que sólo los usuarios permitidos tienen acceso a la información de la VPN. La encriptación de datos está basada en cifrar estos para que no puedan ser leídos por personas a las que no van dirigidos.
- Reducen los costes y son sencillas de usar.
- Facilita la comunicación entre dos usuarios/sedes situados en lugares distantes.

El escenario implantado permitirá conectar a cualquier usuario roadwarrior desde Internet a la VPN del grupo de investigación, y poder utilizar los recursos disponibles en esta (documentos, impresoras, etc.).

1.3. Repercusión sobre el estado actual.

Actualmente el escenario esta compuesto de varios ordenadores ubicados en los distintos laboratorios del grupo de investigación (en diferentes edificios del campus de Castelldefels de la UPC). Estos ordenadores están conectados a la red de la UPC utilizando su electrónica de red. Sin embargo, la configuración actual no permite que la seguridad aplicada sea totalmente óptima. En el capítulo 3 se explicará el esquema de red y su seguridad actual.

La implantación de este proyecto comporta la instalación de un servidor VPN (al cual se conectarán los clientes), que validará comprobando una serie de parámetros la confidencialidad de la conexión. Tanto el servidor como los clientes seguirán perteneciendo a la red de la EPSC (con la misma configuración de red tanto a nivel físico como lógico). Además formarán parte de una nueva red privada “paralela” a ella, con nuevas funcionalidades.

2. VPN (Virtual Private Network).

2.1. Descripción.

Como ya se ha dicho anteriormente, una Red Privada Virtual, es una red que permite una comunicación entre usuarios remotos a una red, o la comunicación entre diversas redes distantes entre si (*Fig.2.1).* Esta comunicación se establece de forma segura entre los dos extremos y es totalmente transparente para todos los usuarios, redes, etc, que existan entre los dos extremos de la VPN.

Para el usuario roadwarrior, la comunicación que realiza con el servidor de la VPN, es como si tuviera un enlace dedicado y totalmente privado (*Fig.2.2).*
Lo mismo ocurre cuando una empresa se quiere comunicar con otra red o empresa ubicada en otro lugar, que tanto para origen como destino, es como si todo fuese una única red privada.

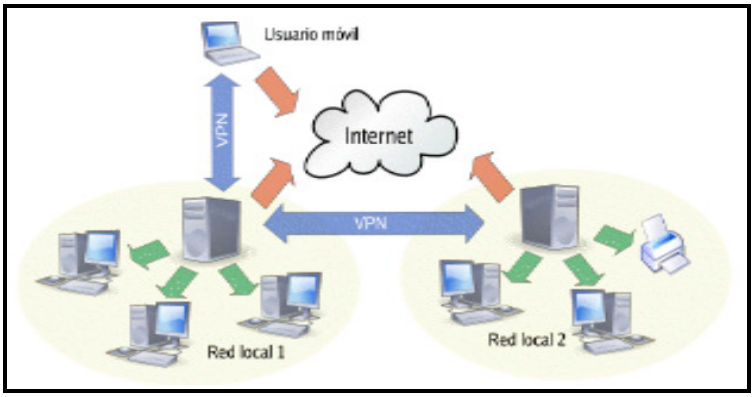

Fig.2.1 Comunicaciones en una VPN.

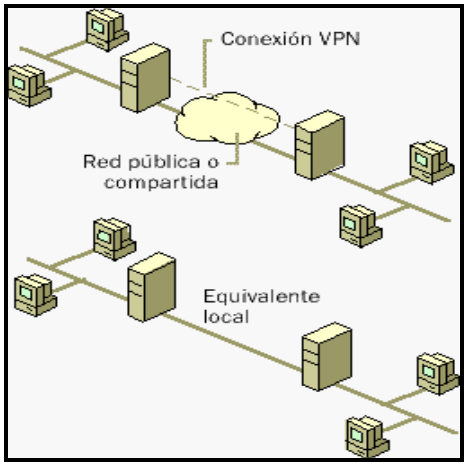

Fig.2.2 Diagrama lógico de una VPN.

2.2. Seguridad.

Esta es la característica más importante de una VPN. Para garantizar la seguridad son necesarios los siguientes puntos:

- La integridad de los datos, es decir que los datos que llegan hasta el receptor no hayan sido alterados. Para ello se utilizan funciones de Hash (anexo F). Las más usadas son *Message Digest* (MD2 y MD5) y el Secure Hash Algorithm (SHA), siendo SHA más seguro, debido a que la longitud de la función de hash es mayor.
- La encriptación se consigue mediante el cifrado que permite que los datos puedan ser descifrados por el receptor y entenderlos correctamente. Esto es necesario ya que como los datos viajan por Internet, cualquier usuario malintencionado podría interceptarlos. Se utilizan algoritmos de cifrado como Data Encryption Standard (DES) que es un algoritmo que toma un texto en claro de una longitud fija de bits y lo transforma mediante una serie de complicadas operaciones en otro texto cifrado de la misma longitud, Triple DES (3DES) y Advanced Encryption Standard (AES), que consisten en variaciones de DES. AES es el más seguro y utilizado hoy en día. No se entrará en detalle fondo este tipo de cifrados porque es muy complejo y se extendería demasiado.
- La autenticación, de forma que solo el destinatario sea el que recibe los datos. Se ha implantado mediante usuarios de los distintos S.O (sistemas operativos) y limitando el acceso de los programas utilizados (que se explicará durante la memoria) sólo para nuestra red privada.

Los puntos anteriores son necesarios para la creación de una VPN, pero aún es posible aumentar la seguridad. Para ello:

- Utilizar una firma digital, de forma que el emisor no puede negar que él haya sido el emisor de la información (no repudio).
- Usar certificados de autenticación (claves de seguridad) para realizar la conexión.
- Revocar las claves de seguridad. Las claves deben ser generadas y renovadas cada cierto tiempo, de forma que al cambiar las claves se aumenta la seguridad de la red. Ya que en caso de que algún usuario malintencionado se apropie de una clave, al revocarla nuevamente, la clave anteriormente conseguida por ese usuario ya no le será válida. El administrador de la red será el encargado de definir cada cuanto tiempo se deberán revocar las claves.

Los datos viajan encapsulados por el túnel, con una cabecera que contiene información sobre su destino y la ruta que debe tomar hasta llegar a él.

En una VPN cada cliente tiene una dirección IP de forma que esta dirección sólo podrá ser vista por los componentes de la VPN, y no podrá ser accesible desde fuera.

2.3. Arquitecturas.

Como se ha dicho anteriormente, para evitar el problema de que una comunicación pueda ser interceptada por una tercera persona con la técnica de hombre en el medio (man in the middle) utilizando un sniffer (programa de captura de las tramas de red), se utilizan túneles, mediante la técnica de tunneling. El Tunneling lo que hace es abrir una conexión entre dos puntos (emisor – receptor) utilizando un protocolo seguro como SSL (Secure Socket Layer) o SSH (Secure Shell), anexo G. La información que anteriormente estaba en claro, ahora será enviada por el túnel de forma segura. Para ello, la información se cifra, después se encapsula, y seguido se le cambia la cabecera con información acerca del emisor y el receptor, creando un túnel lógico entre los dos extremos de la comunicación.

Existen dos tipos de arquitecturas para una VPN:

- **VPN extremo a extremo:** también se le conoce con el nombre de LAN to LAN. En este modelo de VPN una oficina de una empresa o una sede se conecta a la oficina central, en la cual está situado el servidor VPN, que se encarga de crear los túneles para las conexiones, y de dirigir el tráfico *(Fig.2.3)*.

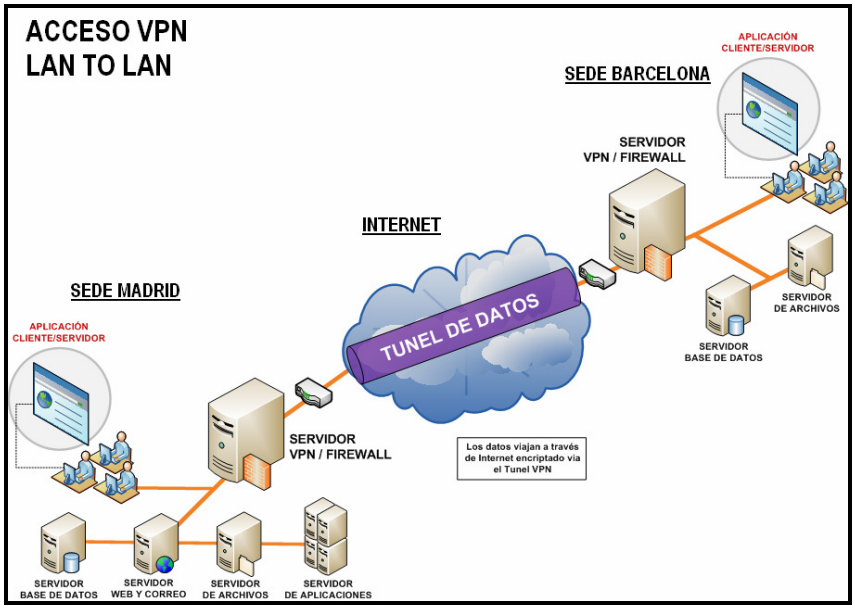

Fig.2.3 VPN LAN to LAN

- **VPN de acceso remoto:** un usuario se conecta a una empresa desde algún lugar remoto (roadwarrior), como puede ser su hogar, un aeropuerto, hotel, etc. La conexión la realiza mediante Internet, con autenticación entre cliente y servidor (*Fig.2.4).*

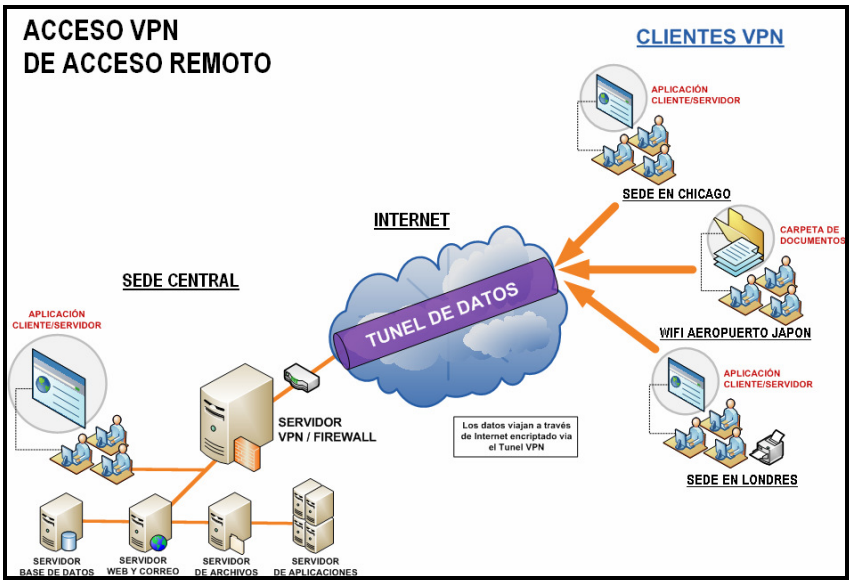

Fig.2.4 VPN de acceso remoto

- **VPN interna VLAN:** Poco usado pero muy útil para empresas. Para realizar las conexiones, se utiliza la LAN, y no Internet como en los dos anteriores casos. Sirve para aislar zonas o recursos de la red. Ayuda a mejorar las prestaciones de seguridad de las redes inalámbricas (WIFI).

2.4. Servidor VPN.

Existen 2 posibles configuraciones a la hora de instalar el servidor de la VPN (encargado de gestionar las conexiones de los clientes):

- **Servidor interno**. En esta configuración, el servidor se encuentra accesible físicamente por el personal de mantenimiento informático de la propia empresa. De esta manera se consigue un ahorro económico (ya que no se ha de pagar a una empresa externa que administre el servidor), y una mayor facilidad de gestión (dado que tendremos un acceso más inmediato para realizar configuraciones, etc.).
- **Servidor externo**. En este caso, se encuentra externalizado en una empresa dedicada a albergar y administrar, entre otros dispositivos, servidores.

La opción escogida ha sido la primera por las ventajas ya explicadas anteriormente.

3. SITUACION ACTUAL DE LA EPSC.

Describiremos brevemente como está montada la red del campus de la EPSC. Actualmente existen los siguientes edificios:

- **ESAB**: Escuela Superior de Agricultura de Barcelona.
- **EPSC**: Escuela Politécnica Superior de Castelldefels.
- **CIMNE:** Centro Internacional de Métodos Numéricos en Ingeniería.
- **UTG**: Unidad Troncal de Gestión, más conocido como edificio de servicios.

En cada uno de los edificios, existen en sus sótanos habitaciones destinadas a albergar todo el material de hardware relacionado con la electrónica de red. En estos emplazamientos hay instalados racks de comunicaciones, en los que se encuentran los patch panels que contienen los puntos de red de todos los ordenadores del edificio en cuestión. Estos patch panel van conectados a los switches (dispositivos electrónicos de interconexión de red). El número de racks, así como de estos dispositivos electrónicos, es variable en cada edificio dependiendo del número de tomas de red a cubrir en cada caso.

No es necesario poner en ninguna planta del edificio un nuevo armario con switches para evitar que la señal de red se atenúe, ya que los edificios no son muy altos y las distancias hasta el sótano no superan los teóricos 100 metros permitidos en este tipo de cableado de red UTP.

A su vez los switches están configurados en cluster, es decir, conectados entre ellos y administrados como si fueran uno único, lo que ofrece mayor facilidad gestión y administración. De esta manera si uno de ellos cae, todas las redes seguirían funcionando por otro.

Los edificios se conectan, mediante cableado de fibra óptica (soportando mayor ancho de banda en la transmisión que el cable UTP) que se extiende por túneles de comunicaciones, al edificio de la EPSC, concretamente a la habitación llamada “troncal”. En esta, es donde se encuentran 2 routers que conectan por un lado a los 4 edificios entre sí, y por otro con Campus Nord (Barcelona), por donde se realiza la salida hacia Internet (*Fig.3.1*). Ambos routers están configurados en modo balanceo de carga, técnica usada para compartir el trabajo a realizar. Se mantiene gracias a un algoritmo que divide de la manera más equitativa posible el trabajo, para evitar los así denominados *cuellos de botella*. Suele suceder cuando se deben servir las solicitudes de un gran número de usuarios. Se trata de un problema de escalabilidad que surge con el continuo crecimiento del número de usuarios.

En cuanto al nivel lógico de los routers, a nivel de capa 3 de OSI (anexo K), hay creadas diferentes VLANs. Una VLAN (acrónimo de ‘red de área local virtual’) es un método de crear redes lógicamente independientes dentro de una misma red física. Esto permite que ordenadores de distintas redes se comporten como si estuviesen conectados al mismo conmutador, aunque pueden estar en realidad conectados físicamente a diferentes segmentos de una red de área local. Son configurables mediante software en lugar de hardware, lo que las

hace extremadamente flexibles. Una de las mayores ventajas de las VLAN surge cuando se traslada físicamente algún ordenador a otra ubicación, ya que puede permanecer en la misma VLAN sin necesidad de cambiar la configuración IP de la máquina. A continuación se explican los rangos de IP's de cada edificio:

- **Edificio EPSC**. Dispone de 4 rangos (de los cuales solo hemos recibido información acerca del 147.83.114.x y del 147.83.155.x).
- **Edificio UTG**. Se dispone de 1 rango (pero se desconoce el número de este).
- **ESAB.** Dispone de 2 rangos (de los cuales solo hemos recibido información acerca del 147.83.16.x).
- **CIMNE.** Dispone de 1 rango (147.83.12.x).

En cuanto a seguridad, no hay implantado ningún firewall a nivel de hardware. Otro punto que facilita un ataque externo es el hecho de que los ordenadores disponen de IP's públicas, accesibles desde Internet. Esto es así para tener mayor flexibilidad en la configuración de escenarios de red en las aulas destinadas a docencia.

Por otro lado, los laboratorios de investigación de este TFC se encuentran en esta misma situación en cuanto a seguridad, porque en el momento en que se realizó el estudio y la implantación de la red no se especificaron políticas de seguridad para estos lugares.

Sin embargo, existe un sistema de seguridad a nivel de servicio que es realizado mediante un sistema de monitorización por parte de UPCNET. De tal manera que si detectan algún tipo de anomalía (acceso no autorizado, etc.) pasan notificación a UTG, quienes proceden a revisar el ordenador realizando un mantenimiento, pudiendo llegar a descargar una imagen en el ordenador (mediante el sistema implantado en el campus REMBO). Este, restaura a su estado original el disco duro del PC. De esta manera se deshacen todos los cambios realizados en el ordenador desde la fecha de la última restauración de esta imagen.

La IP de los routers es la puerta de enlace que usan los ordenadores de cada una de las VLAN para poder conectar a Internet y al resto de equipos.

Viendo esta infraestructura de red y siguiendo una evidente política de seguridad de la EPSC, como estudiantes de telemática no nos ha sido posible acceder a las herramientas de configuración de los switches y routers. Esto hubiera permitido poder buscar una solución algo más rápida y estándar, que a priori podría haber sido la creación de una VLAN en la que incluir todos los ordenadores del escenario del TFC, o haber podido cambiar las IP's públicas por privadas para crear otro tipo de configuración. Por este motivo se ha tenido que crear una VPN utilizando un software libre, aprovechando las IP's públicas de estos ordenadores, y limitando los accesos a estos mediante los firewall de sus S.O.

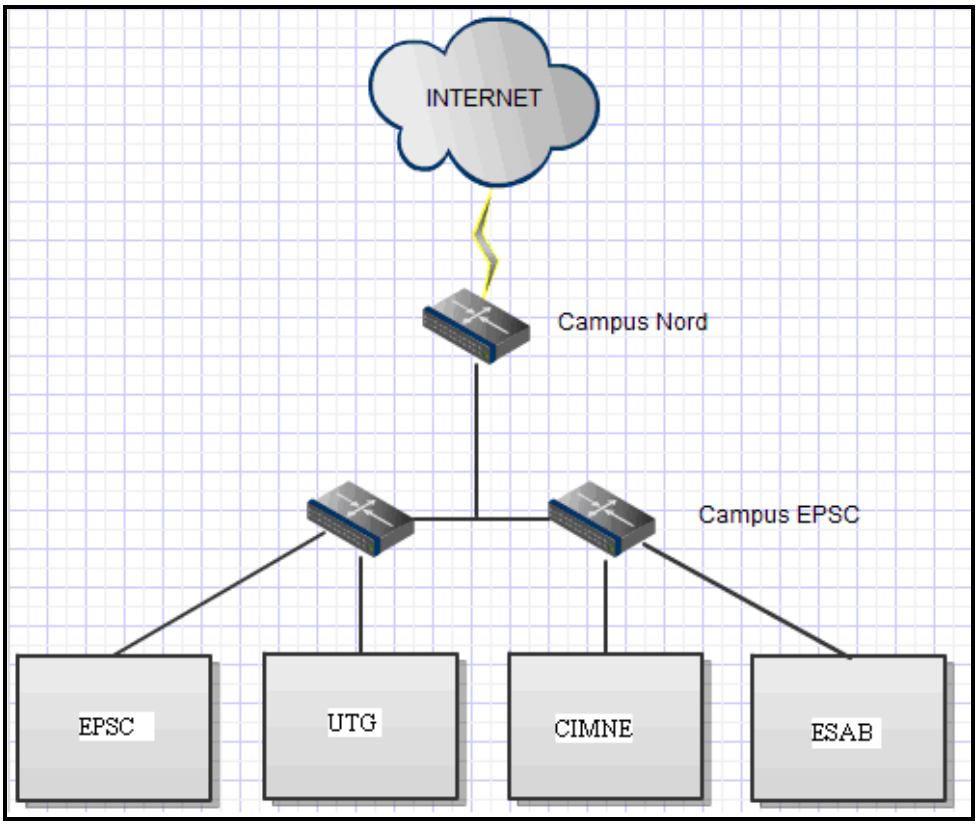

Fig.3.1 Plano de red de la EPSC

4. SOLUCIÓN IMPLANTADA.

4.1. Tipos de VPN.

Una VPN se puede crear mediante hardware o software.

Las creadas por hardware pueden conseguir un mayor rendimiento y a su vez son fáciles de configurar, pero sin embargo tienen más limitaciones que las creadas por software, ya que disponen de menos parámetros configurables. Existe una gran variedad de productos que permiten la creación de VPN por hardware, como pueden ser los productos de Nortel, Cisco, Linksys, Symantec, Nokia, U.S. Robotics, D-link, etc.

Las VPN por software tienen un menor rendimiento, pero permiten una mayor flexibilidad a la hora de configurarlas. Además, en caso de que surjan problemas, se pueden solucionar en versiones posteriores y actualizar su software. Otro inconveniente, es que son algo más complejas que las de hardware (debido a que permiten más opciones de configuración), pero simultáneamente esto se convierte en un punto a favor, ya que se podrá crear una VPN que se adapte mejor a las necesidades específicas.

El protocolo más usado para crear comunicaciones seguras (como puede ser una VPN) es IPSEC (capítulo 4.6), pero también existen el PPTP, L2F, L2TP, SSL/TLS, SSH, etc. Todos poseen diferentes niveles de seguridad, facilidad, mantenimiento, y clientes soportados.

Existen dos formas de crear una VPN por software. La primera es mediante las opciones que vienen en los propios S.O. como pueden ser Windows XP, Windows 2003 server, Windows Vista, GNU/Linux y Unix, aunque estas redes suelen ser algo limitadas. Y la otra opción, y más recomendable debido a las características que se explicarán en el capítulo 4.3, es mediante aplicaciones de código abierto, como por ejemplo OpenVPN, OpenSSH, Vtun, Hamachi, FreeS/Wan, etc.

Tanto en la solución por hardware como en la de software, se pueden usar cortafuegos (firewall), que serán explicados con más detalle en el capítulo 7 de la memoria y el anexo D, y permiten conseguir un mayor nivel de seguridad.

4.2. Elección del software OpenVPN.

En primer lugar se ha decidido montar la VPN mediante software, ya que sobre el escenario en el que se va a trabajar no podemos acceder a los routers de la EPSC. El mantenimiento y configuración del hardware corresponde a una empresa llamada UPCNet, y por tanto era más viable buscar una solución en la que pudiésemos trabajar directamente nosotros sin tener que depender de nadie. Además en caso de hacerla por hardware, quizás hubiésemos tenido que modificar partes de la LAN de la EPSC, y esto no nos estaba permitido.

Otro de los motivos es que de esta forma se podría adaptar más la VPN a nuestras necesidades, ya que nos permitía una mayor flexibilidad de configuración.

Para decidir que aplicación usar, se consultó Internet (páginas dedicadas a VPN, foros, etc.), y se consultó a profesores de la EPSC expertos en el tema para valorar su opinión.

Se decidió usar OpenVPN que es una solución de conectividad basada en software mediante SSL/TLS. En el siguiente apartado se explica detalladamente las características que han propiciado la elección de esta solución.

4.3. Características de OpenVPN.

OpenVPN es una solución para la creación de VPN mediante software. Fue creado por James Yonan en el año 2001. Es de software libre, es decir, puede ser usado, copiado, cambiado/mejorado y redistribuido libremente. No hay que confundirlo con software gratuito ya que no es obligatorio que todo software libre sea gratuito. Sin embargo OpenVPN también es gratuito. Todo esto hace que esté en continua mejora, ya que millones de personas lo utilizan diariamente en el mundo, por lo que se podrá disponer de futuras actualizaciones de una herramienta a la que se tendrá acceso de manera fácil y gratuita. Es de fácil uso y configuración para usuarios inexpertos, respecto a otras opciones, como puede ser el uso del protocolo IPSEC, que aunque es uno de los más conocidos (el estándar para crear una VPN), es más complejo y requiere mayor experiencia para su uso.

OpenVPN es un software basado en SSL/TLS que permite crear VPN LAN to LAN, VPN de acceso remoto y VPN interna (para redes inalámbricas). TLS está desarrollado a partir de SSL, que en un principio solo se usaba para aplicaciones Web. Esto hizo que se estudiaran profundamente los problemas que tenían, y hoy en día su uso ya no está limitado exclusivamente a aplicaciones Web. Además esto ha permitido que se utilice para la autenticación e intercambio de claves.

Es un software para crear redes seguras en la capa 2 o 3 del modelo OSI (según se use: Tunnel o Bridge). Por tanto un cliente no podrá realizar una conexión al servidor VPN mediante un navegador Web, ya que OpenVPN no opera en la capa 7 (Aplicación) de la pila OSI.

Algunas de las características y ventajas que ofrece OpenVPN son:

- Facilidad de instalación y uso respecto a otros softwares más complejos.
- Es multiplataforma (Linux, Solaris, Mac OS X y Windows 2000/XP/Vista, Symbian OS, etc).
- Soporta IP's dinámicas y NAT (Network Address Translation - Traducción de Dirección de Red, que es un mecanismo utilizado por

routers IP para intercambiar paquetes entre dos redes que se asignan mutuamente direcciones incompatibles).

- Es estable y escalable para los clientes.
- Flexibilidad (es personalizable).
- Permite autenticación mediante plugins y autenticación basada con el paquete PAM (Pluggable Authentication Method).
- Uso de certificados para autenticación.
- Utiliza algoritmos HMAC para la autenticación de los datos del túnel.
- Seguridad basada en el uso de SSL/TLS para la autenticación.
- Se ha construido de forma modular. Lo relacionado con la encriptación está soportado por la librería OpenSSL (capítulo 4.7.4), y lo relacionado con la funcionalidad de los túneles IP está proporcionado por el adaptador virtual TUN/TAP (capítulo 4.7.3). Esto permite que OpenVPN pueda adaptarse a nuevas versiones de OpenSSL.
- Es un software rápido y ágil en su ejecución ya que no necesita grandes recursos de procesamiento.
- Ofrece un alto control de seguridad mediante parámetros (permisos, etc.).
- Soporta autenticación del cliente utilizando políticas de acceso a usuarios y grupos específicos, y reglas de firewall aplicadas a las interfaces virtuales usadas por OpenVPN para el filtrado de paquetes IP.
- Es compatible con infraestructuras de clave pública (PKI) mediante el uso de certificados y el intercambio de claves RSA (capítulo 4.4).
- OpenVPN trabaja en el espacio de usuario, de forma que en caso de fallo de un componente, este contiene dicho fallo, y no permite que el sistema esté en peligro.
- Permite tunelizar una subred IP a través de un puerto TCP o UDP.
- Un servidor VPN puede tener miles de clientes VPN.
- OpenVPN puede trabajar con cifrados simétricos (compartiendo claves) o cifrados asimétricos con certificados y el intercambio de claves mediante TLS (capítulo 4.4).
- Compresión en tiempo real y gestión del tráfico para manejar el uso del BW (ancho de banda).
- Se puede ejecutar como servidor (esperando conexiones entrantes) o como cliente (iniciando conexiones).
- Crear puentes ethernet seguros utilizando para ello adaptadores ethernet virtuales TUN/TAP.
- Controlar y monitorizar conexiones OpenVPN mediante interfaces graficas de usuario (GUI) para diferentes plataformas (Windows, Linux, Unix, MAC OS X,…).
- Al permitir múltiples conexiones (desde la versión 2.0), únicamente es necesario abrir un puerto (tanto en TCP como en UDP).
- Tiene reservado oficialmente el puerto 1194 por la IANA (Agencia de asignación de números de Internet. En 1998 fue sustituido por la ICANN (Internet Corporation for Assigned Names and Numbers), que es una organización internacional, encargada de asignar las direcciones IP, y la asignación de los nombres de dominio).
- Las conexiones OpenVPN pueden ser realizadas a través de casi cualquier firewall. Si se posee acceso a Internet y se puede acceder a

sitios HTTPS, entonces un túnel OpenVPN debe funcionar sin problemas.

Al no llevar muchos años en desarrollo OpenVPN, tiene algunas desventajas. Pero esta desventajas con el tiempo seguramente se irán solventando, ya que OpenVPN cada día está más en auge, y al ser código libre hay muchas empresas que trabajan en ese software para mejorarlo en futuras versiones.

- No tiene compatibilidad con IPSEC que justamente es el estándar actual para soluciones VPN.
- Poca gente conoce como usar OpenVPN. Y por tanto es difícil encontrar expertos en este software.
- No ofrece compatibilidad con estándares como IKE, PPTP o L2TP.

Tal y como se ha comentado anteriormente en este mismo capítulo, OpenVPN permite crear redes seguras en la capa 2 o 3 del modelo OSI. A continuación se explica la implementación en ambas capas.

4.3.1. Implementación en capa 2 – Enlace.

El encapsulamiento a este nivel permite tunelizar cualquier tipo de paquete (incluso protocolos no-IP, como IPX4 de Netware Systems). Se crea un dispositivo virtual con el que se establece la conexión con el otro lado del túnel. La implementación en la capa de enlace es usada por:

- **PPTP**: Point to Point Tunneling Protocol. Desarrollado por Microsoft, es una extensión de PPP. Únicamente puede establecer un túnel entre pares.
- **L2F**: Layer 2 Forwarding. Desarrollado por Cisco principalmente, puede establecer conexiones simultáneas.
- **L2TP**: Layer 2 Tunneling Protocol. Usado por Cisco y otros fabricantes, es el estándar ya que combina las ventajas de PPTP y L2F. Su principal problema es que no ofrece seguridad.
- **L2Sec**: Layer 2 Security Protocol. Mejora la seguridad utilizando SSL/TLS. Su problema es que sobrecarga bastante la comunicación.

4.3.2. Implementación de capa 3 – Red.

IPSEC es la tecnología más usada, y un estándar de seguridad de Internet en capa 3. IPSEC puede encapsular cualquier tráfico de capa 3 pero no el de capas inferiores, por tanto no se podrá utilizar para protocolos no-IP como IPX. Su gran ventaja es que es multiplataforma, y sea mediante software o hardware, tiene múltiples soluciones.

Existen dos métodos principales usados por IPSEC:

- **Modo Tunnel**. Todos los paquetes IP son encapsulados en un nuevo paquete y enviados a través del túnel siendo desempaquetados en el otro extremo, y posteriormente dirigidos a su destinatario final. En este modo se protegen las direcciones IP de emisor y receptor así como el resto de los datos de los paquetes.
- **Modo Transporte**. Solo la carga útil (payload) de la sección de datos es cifrada y encapsulada. La sobrecarga entonces, es sensiblemente menor que en el caso anterior, pero se exponen los datos a posibles atacantes que podrán ver quien se está comunicando con quien.

4.4. Seguridad en OpenVPN.

OpenVPN tiene dos métodos para cifrar datos. El primer método consiste en utilizar claves estáticas pre-compartidas y el segundo método es usar certificados con protocolos SSL/TLS y claves RSA (es un algoritmo asimétrico que cifra por bloques, mediante claves públicas y privadas), compatible con NAT (anexo I), y DHCP (protocolo que permite a un equipo obtener una configuración de red de forma automática, facilitando la administración de la red).

Sin duda, el segundo método es el más seguro de los dos, pero a su vez más complejo. A continuación se explican ambos métodos.

4.4.1. Cifrado simétrico y claves pre-compartidas.

En este cifrado todos los usuarios usan la misma clave, y esta es instalada en todas las máquinas. Se usa la misma clave para cifrar y descifrar (Fig.4.2*).*

El problema es que si alguien consigue esta clave, sólo tiene que capturar los datos, y con la clave instalada en su PC se podría comportar como un usuario más de la red, y por tanto descifrar esa información.

Por tanto su principal problema o inconveniente es buscar un canal seguro para realizar el intercambio de dicha clave. Dado que toda la seguridad está en la clave, es importante que sea muy difícil adivinarla. Esto quiere decir que el abanico de claves posibles, debe ser amplio. Actualmente, los ordenadores pueden descifrar claves con extrema rapidez, y ésta es la razón por la cual el tamaño de la clave es importante en los criptosistemas modernos. El algoritmo de cifrado DES, por ejemplo, usa una clave de 56 bits, lo que significa que hay 2 elevado a 56 claves posibles (72.057.594.037.927.936 claves ya que el sistema binario trabaja con dos posibles valores, 1 y 0 →). Esto representa un número muy alto de claves, pero un ordenador genérico puede comprobar el conjunto posible de claves en cuestión de días. Una máquina especializada puede hacerlo en horas. Algoritmos de cifrado de diseño más reciente como 3DES usa claves de 128 bits, lo que significa que existen 2 elevado a 128 claves posibles. Esto equivale a muchísimas más claves, y aun en el caso de que todas las máquinas del planeta estuvieran cooperando, tardarían más tiempo en encontrar la clave que la edad del universo.

Otra opción para que este tipo de cifrados sean más seguros es modificar el "lifetime" (tiempo de vida) de las claves. Esto consiste en cambiar cada cierto tiempo las claves, así cuando un usuario consiga la clave, quizás haya pasado el tiempo de validez de esta y ya no le resulte válida.

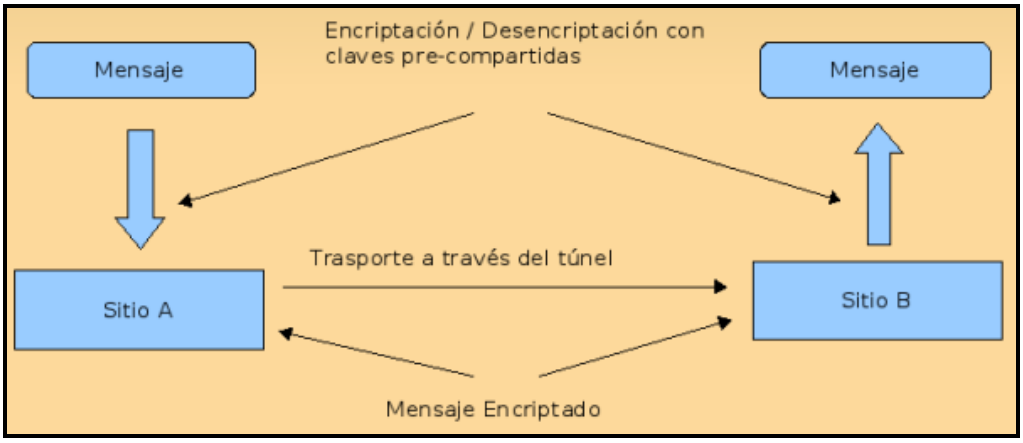

Fig.4.2 Cifrado simétrico

Por defecto OpenVPN utiliza, como algoritmo de cifrado simétrico, el algoritmo Blowfish, con un tamaño de clave por defecto de 128 bits. Pero OpenVPN soporta cualquier algoritmo de cifrado que proporcione la librería OpenSSL, por ejemplo, el algoritmo AES, muy utilizado en la actualidad, con un tamaño de clave de 256 bits, añadiendo a los archivos de configuración la siguiente línea:

cipher AES-256-CBC

El método de clave pre-compartida es más sencillo, ya que no usa certificados, y por tanto no necesita una unidad certificadora (CA). Al ser un sistema menos seguro, las claves han de estar bien guardadas, ya que se usarán durante todo el tiempo de vida de la VPN.

4.4.2. Cifrado asimétrico con SSL/TLS.

La criptografía asimétrica es el método criptográfico que usa un par de claves para el envío de mensajes. Las dos claves pertenecen a la misma persona a la que se ha enviado el mensaje. Una clave es *pública* y se puede entregar a cualquier persona, la otra clave es *privada* y el propietario debe guardarla de modo que nadie tenga acceso a ella. Además, los métodos criptográficos garantizan que esa pareja de claves sólo se puede generar una vez, de modo que se puede asumir que no es posible que dos personas hayan obtenido casualmente la misma pareja de claves.

Si el remitente usa la clave pública del destinatario para cifrar el mensaje, una vez cifrado, sólo la clave privada del destinatario podrá descifrar este mensaje,

ya que es el único que la conoce. Por tanto se logra la *confidencialidad* del envío del mensaje, nadie salvo el destinatario puede descifrarlo.

Si el propietario del par de claves usa su clave privada para cifrar el mensaje, cualquiera puede descifrarlo utilizando su clave pública. En este caso se consigue por tanto la identificación y autentificación del remitente, ya que se sabe que sólo pudo haber sido él quien empleó su clave privada (salvo que alguien se la hubiese robado). Este es el fundamento de la firma electrónica.

Los sistemas de cifrado asimétricos o de clave pública se inventaron para evitar el problema del intercambio de claves de los sistemas de cifrado simétricos. Con las claves públicas no es necesario que el remitente y el destinatario se pongan de acuerdo en la clave a emplear. Todo lo que se requiere es que, antes de iniciar la comunicación secreta, el remitente consiga una copia de la clave pública del destinatario. Es más, esa misma clave pública puede ser usada por cualquiera que desee comunicarse con su propietario. Por tanto, se necesitarán sólo n pares de claves por cada n personas que deseen comunicarse entre sí.

La mayor ventaja de la criptografía asimétrica es que se puede cifrar con una clave y descifrar con la otra, pero este sistema tiene bastantes desventajas:

- Para una misma longitud de clave y mensaje se necesita mayor tiempo de proceso.
- Las claves deben ser de mayor tamaño que las simétricas.
- El mensaje cifrado ocupa más espacio que el original.

Algunos algoritmos de técnicas de clave asimétrica son:

- Diffie-Hellman
- RSA
- ElGamal

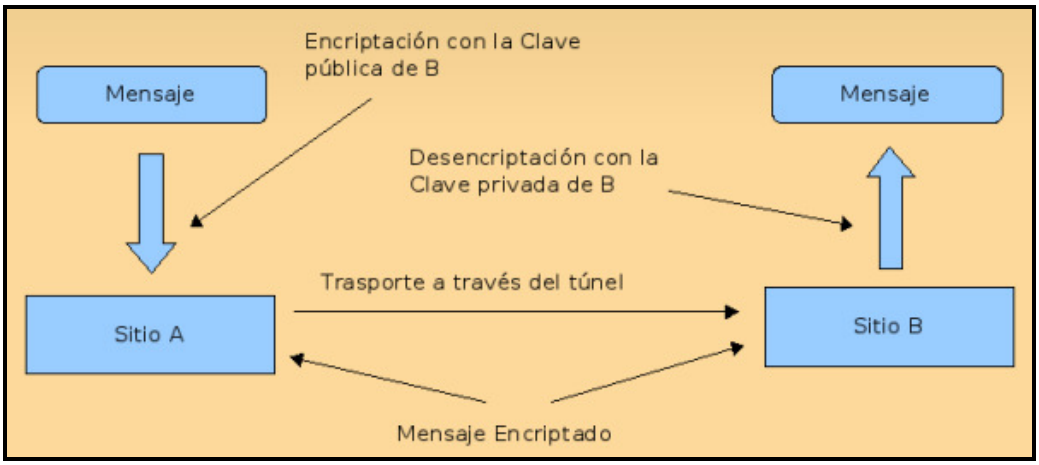

Fig.4.3 Cifrado asimétrico

Las bibliotecas SSL/TLS son parte del software de las librerías de OpenSSL que vienen instaladas en cualquier sistema moderno, e implementan mecanismos de cifrado y autenticación basados en certificados. Los certificados generalmente son emitidos por entidades de reconocida confiabilidad, aunque también es posible emitirlos nosotros mismos y usarlos en nuestra VPN. Si un certificado además lleva una firma digital, asegura que esa persona es la propietaria.

4.5. Encapsulado de OpenVPN.

OpenVPN permite usar como protocolo de transporte tanto UDP como TCP para realizar las conexiones punto a punto. Para ello utiliza el comando *–proto p*, en el que esta "p" puede ser "udp", "tcp-client" (en el cliente), o "tcp-server" (en el servidor). Si no se utiliza ese comando, OpenVPN utiliza por defecto udp.

Si al realizar el cliente la petición de conexión, no recibe respuesta por parte del servidor, este esperará 5 segundos para hacer otro intento de conexión. Este valor por defecto está a 5 segundos, pero se puede ajustar con la directiva –connect-retry.

TCP también utiliza temporizadores que pueden variar incrementándose en caso de expirar su tiempo, para no congestionar la red. Esto puede ocasionar problemas cuando existen dos capas que utilizan este método. Es muy posible que ocurra en las VPN, ya que se puede tener un paquete TCP, que se encapsule mediante otro TCP a través de la VPN. Esto es en caso de usar este protocolo en OpenVPN. En este caso, el problema es que si el protocolo TCP de la capa inferior no recibe los paquetes que está esperando (ya sean ACK u otro tipo de paquetes), aumentará su temporizador al igual que aumentará su cola de retransmisiones. Por el contrario el protocolo TCP de la capa superior se quedará esperando el ACK correspondiente, y al no recibirlo también aumentará su temporizador y su cola de retransmisiones. La diferencia entre las dos capas es que el temporizador de la capa superior será menor que el de la capa inferior, y por tanto la cola de retransmisiones de la capa superior aumentará más rápido de lo que la capa inferior puede procesar. Esto podría llegar a provocar congestión en la red.

Otro gran problema que existe si se da este caso, es que TCP trabaja con el tamaño máximo de un paquete, y por tanto, al encapsularse de nuevo por la capa superior, ese paquete excede de tamaño y ha de ser fragmentado. Esta fragmentación puede provocar que el router no sepa encaminar paquetes tan grandes, y que el receptor no pueda volver a juntar toda la información al completo, ya que puede haber pérdida de paquetes o que algunos se reciban con información repetida.

Para solventar los problemas mencionados anteriormente, OpenVPN recomienda UDP, y usar sólo TCP en los casos que UDP no puede ser usado.

UDP ofrece una mayor seguridad frente a posibles ataques, además de permitir a OpenVPN trabajar de manera más eficiente.
En la siguiente imagen se observa como OpenVPN realiza el encapsulado de las tramas (Fig.4.4*).*

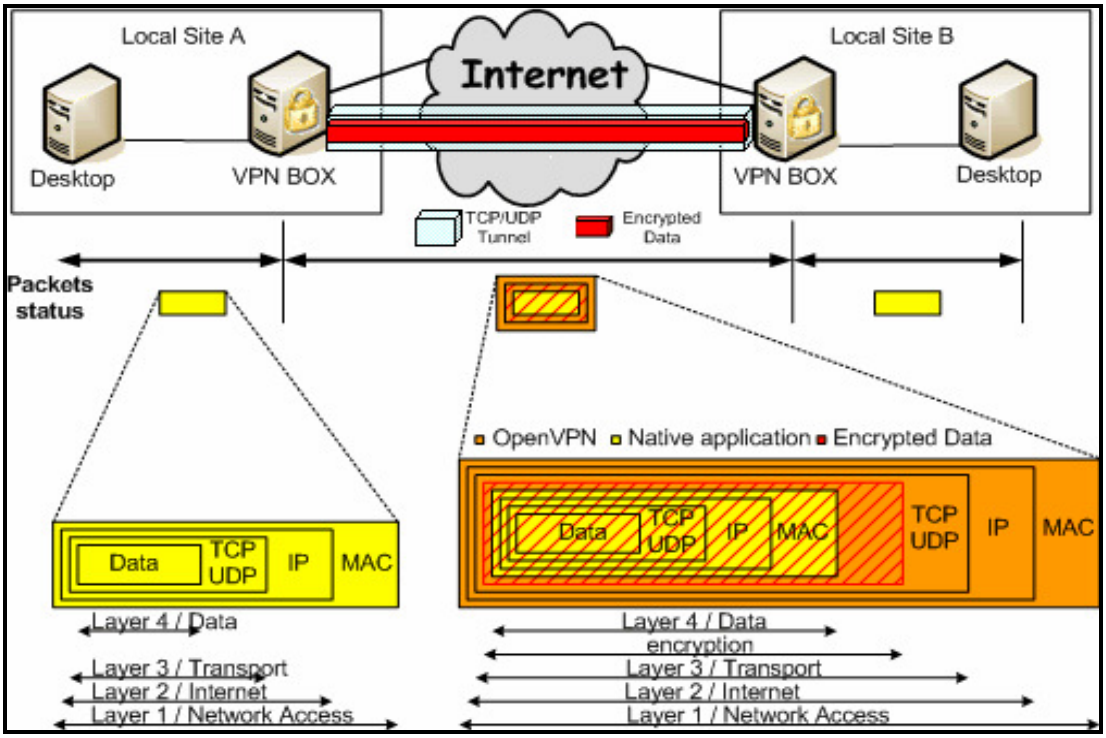

Fig.4.4 Encapsulado de OpenVPN.

La parte izquierda de la imagen (amarillo) es el mensaje original, formado por los datos más la cabecera. Posteriormente, en el túnel de la VPN, el mensaje es cifrado (en rojo), y se le añade la cabecera correspondiente de la VPN, con el protocolo utilizado (TCP o UDP), las direcciones IP origen y destino, y la dirección MAC de origen.

4.6. Comparativa entre OpenVPN e IPSEC.

El protocolo IPSEC actúa en la capa de red (capa 3 de OSI). Esto le permite ser más flexible, ya que puede ser uszado para proteger protocolos de la capa 4 (como TCP y UDP). Permite la comunicación segura entre equipos, y es el protocolo estándar para crear VPNs. Para implementar una VPN con IPSEC, existen softwares por ejemplo “Panda GateDefender Integra”, que usa equipos físicos que harán de gateway y su correspondiente software. A continuación se muestra una comparativa entre OpenVPN e IPSEC (Tabla 4.1).

Tabla 4.1 Comparativa entre IPSEC e OpenVPN.

IPSEC	OpenVPN
Estándar de la tecnología VPN	Aun desconocida y no compatible con IPSEC
Plataformas de hardware + software	Plataforma de software en todos los sistemas operativos disponibles
Tecnología conocida y probada	Tecnología nueva y aun en crecimiento
Muchas interfaces gráficas disponibles	Sin interfaces gráficas profesionales, aunque ya existen algunos proyectos prometedores
Necesidad de modificaciones críticas al kernel	Interfaces de red y paquetes estandarizados
Necesidad de permisos de administrador	Se ejecuta en el espacio del usuario
Diferentes implementaciones de distintos proveedores pueden ser incompatibles entre si	Tecnologías de cifrado estandarizadas
Configuración compleja y tecnología compleja	Facilidad, buena estructuración, tecnología modular y facilidad de configuración
Curva de aprendizaje muy pronunciada	Fácil de aprender y éxito rápido para principiantes
Necesidad de uso de muchos puertos y protocolos en el firewall	Utiliza solo un puerto del firewall
Problemas con direcciones dinámicas en ambos extremos	Trabaja con servidores de nombres dinámicos como DynDNS con reconexiones rápidas y transparentes
Problemas de seguridad de las tecnologías IPSEC	SSL/TLS como estándar de criptografía
No permite control de tráfico	Control de tráfico (Traffic shaping)
Bajas velocidades	Velocidad (más de 20 Mbps en máquinas de 1Ghz)
Algunos problemas con firewall	Compatibilidad con firewall y proxy
Algunos problemas con NAT	Ningún problema con NAT (ambos lados puede ser redes NATeadas)
No permite roadwarriors	Posibilidades para roadwarriors

A parte de todas las diferencias descritas anteriormente, una de las más importantes es el nivel de seguridad que hay entre ambas.

OpenVPN opera en el espacio de usuario, y no en el kernel como IPSEC. Al operar en el kernel (núcleo de Linux que se puede definir como el corazón de este S.O. Es el encargado de que el software y el hardware de un ordenador puedan trabajar juntos), le añade una gran complejidad, y esto provoca una falta de seguridad en todo el sistema (que puede ser provocado por un simple

desbordamiento del buffer). A diferencia de esto, OpenVPN contiene su complejidad dentro del espacio de usuario, así puede contener cualquier error en este espacio, sin comprometer la seguridad de todo el sistema, y sin la necesidad de intervenir en las funciones del kernel.

IPSEC está lejos de la arquitectura de S.O. seguro basado en anillos, que es el principio de no interferencia con el espacio de usuario. Este principio divide el S.O. en anillos numerados con diferentes grados de privilegios. El anillo 0 está reservado para el núcleo en sí y procesos esenciales. El anillo 1 es para otros procesos de sistema que necesitan un acceso a bajo nivel de hardware. Al incrementar el número del anillo los privilegios disminuyen. En el anillo 3 están la mayoría de los procesos. Los procesos de un anillo, no pueden afectar a los procesos de un anillo inferior, de esta forma se garantiza la estabilidad y seguridad del sistema. OpenVPN opera en el anillo número 3, el cual es el nivel que realmente se busca.

En ocasiones, para proveer de encriptación al enlace, las aplicaciones necesitan intervenir con el kernel del S.O. para ganar acceso a bajo nivel en la interfaz de red del enlace. En estos casos, OpenVPN emplea “interfaces virtuales” del espacio de usuario para controlar y acceder sin la necesidad de depender del kernel. Estas interfaces virtuales ofrecen un punto extra de seguridad.

4.7. Funcionamiento de OpenVPN.

4.7.1. Librerías previas de instalación.

Como ya se ha comentado anteriormente, OpenVPN es un software fácil de instalar.

Dependiendo del nivel de seguridad que se quiera instalar en nuestra VPN, existen unas librerías previas que se deben instalar.

- En caso de no querer ningún tipo de cifrado, no es necesario instalar ningún tipo de librería OpenSSL.
- Si se usa un cifrado simétrico, es decir, con una clave estática pre-compartida, será necesario instalar la librería “crypto” de OpenSSL.
- En caso de usar el cifrado asimétrico en el que se usan claves públicas y privadas, y el uso de certificados, las librerías que son necesarias instalar son “crypto” y “SSL”

Existe otra librería llamada “LZO”, para el caso en que se quiera la compresión de datos en tiempo real, que permitirá conseguir que el canal sea más rápido y fluido. Realizará la compresión sólo cuando el flujo de datos del túnel pueda ser comprimido.

4.7.2. Como ejecutar comandos y archivos en OpenVPN.

OpenVPN tiene dos formas de poder ejecutarse por consola mediante comandos. La primera forma debe llevar dos guiones, seguido del comando y los parámetros (ej: "--*comando parámetros*"), y la segunda es mediante un fichero de configuración con la directiva "--*config ruta_del_fichero*". Los dos guiones sólo son necesarios en caso de escribir los comandos por consola, pero en el contenido de los archivos no deben de aparecer, es decir, estará el mismo comando que se hubiera puesto por consola, pero sin los guiones.

Una de las diferencias que existen entre usar OpenVPN en Linux o Windows, es la extensión de los archivos de configuración. En Windows la extensión es ".ovpn", mientras que en Linux la extensión es ".conf". Para ejecutar estos archivos el comando también es distinto, ya que una vez situados en el directorio, en Windows se debe escribir en consola:

openvpn nombre_del_fichero

y en Linux:

openvpn --config nombre_del_fichero

Como se puede ver, la diferencia es que en Linux se ha de añadir "*--config*". También existe la posibilidad de ejecutar los archivos mediante entorno gráfico, más fácil de usar para un usuario inexperto.

Una forma de implementar un túnel en ambos sentidos sin cifrado ni autenticación, mediante comandos en consola sería la siguiente:

openvpn --remote dominio1.com --dev tun0 --ifconfig 10.8.0.1 10.8.0.2 –-verb6

openvpn --remote dominio2.com --dev tun0 --ifconfig 10.8.0.2 10.8.0.1 –-verb6

Las direcciones IP 10.8.0.1 y 10.8.0.2 pertenecen a cada uno de los extremos del túnel, y las interfaces virtuales TUN/TAP trabajan en el modo túnel con el nombre "tun0" en las máquinas de ambos extremos.

También se puede hacer mediante ficheros de configuración. Lo primero es crear los ficheros, y después editarlos escribiendo (*Tabla 4.2)*:

Tabla 4.2 Túnel mediante archivos de configuración de OpenVPN.

Extremo1.conf (Linux) **Extremo1.ovpn (Windows)**	**Extremo2.conf (Linux)** **Extremo2.ovpn (Windows)**
remote IPextremo2 dev tun0 ifconfig 10.8.0.1 10.8.0.2 verb6	remote IPextremo1 dev tun0 ifconfig 10.8.0.2 10.8.0.1 verb6

“remote” indica la IP del otro extremo del túnel.
“dev tun” que se va a implementar un túnel y no un puente.
“ifconfig” las IP virtuales de los dos extremos.
“verb6” el nivel de información que nos va a devolver en caso de error.

Una vez creados los archivos de configuración hay que ejecutarlos en un terminal de consola en ambos extremos del túnel:

En Linux:

openvpn --config Extremo1.conf
openvpn --config Extremo2.conf

En Windows:

openvpn Extremo1.ovpn
openvpn Extremo2.ovpn

Se puede comprobar el éxito de la conexión haciendo un ping desde un Terminal al otro extremo del túnel.

4.7.3. Los controladores virtuales TUN/TAP y VTUN.

En primer lugar es importante conocer los conceptos de controlador de red e interfaz de red (ambos explicados en el capítulo 8 de los anexos).

OpenVPN utiliza los controladores virtuales TUN/TAP para establecer el túnel entre los dos extremos. Con una interfaz de red virtual se pueden obtener varias direcciones IP’s con una sola tarjeta de red física.

Los túneles virtuales TUN/TAP se han incorporado en el kernel de Linux a partir de la versión 2.4.x. Para versiones anteriores de kernel de Linux, u otros S.O. (excepto Windows) es necesario instalar la herramienta VTUN. Las primeras versiones de OpenVPN no soportaban Windows, por tanto no se podía instalar esta herramienta. Posteriormente se desarrolló una nueva herramienta llamada TAP-Win32 que permitiera crear interfaces virtuales TUN/TAP para Windows.

OpenVPN para implementar una VPN utiliza el espacio de usuario y enlaza una interfaz de red virtual punto a punto llamada “tun” (normalmente tunX, en el que X es un número), con otra interfaz de red virtual punto a punto (“tun”) remota, como si fuera una línea dedicada entre los dos puntos. A una interfaz virtual se le puede aplicar reglas, firewall, rutas, etc., como si fuera una tarjeta de red física ethernet instalada en la máquina. El funcionamiento del interfaz virtual “tun” es que si desde una máquina se quieren enviar datos a otra, primero se copian los datos desde el interfaz virtual hasta el socket de red del S.O., se envían los datos, se reciben en el socket de red del S.O. de la otra máquina, y se pasan los datos a la interfaz virtual.

Se le llama “tap” cuando trabaja en modo bridge o puente, emulando a un adaptador de red Ethernet en lugar de una interfaz punto a punto, como si hubiera una red ethernet entre los dos extremos.

Lo normal es usar el modo tun para enviar tráfico IP, pero si se quiere enviar otro tipo de tráfico (broadcast, tráfico no IP como Netbios o IPX, etc.) se deberá utilizar el modo tap.

Algunas de las características de la herramienta VTUN son:

- Es fácil de usar para crear túneles virtuales en redes TCP/IP, pudiendo implementar varios tipos de estos túneles con diferentes características de cifrado y compresión.
- Permite gran variedad de configuraciones y es muy utilizado para crear VPN.
- Soporta compresión de datos mediante las librerías “Zlib” (solo para TCP) y “LZO” (para TCP y UDP).
- Permite implementar diferentes tipos de autenticación, cifrado mediante Blowfish con clave de 128 bits y funciones Hash MD5.
- Permite crear túneles utilizando como protocolos de transporte tanto TCP como UDP.
- Está disponible para varios S.O., por ejemplo: Linux, FreeBSD, OpenBSD, MAC, Solaris,…

4.7.4. Seguridad en OpenSSL.

OpenSSL es un software libre desarrollado por los miembros de la comunidad Open Source para libre descarga, y está basado en SSLeay y desarrollado por Eric Young y Tim Hudson.

Es un robusto paquete de herramientas de administración y librerías relacionadas con la criptografía, que suministran funciones criptográficas a otros paquetes como OpenVPN, OpenSSH y navegadores Web (para acceso seguro a sitios HTTPS). Estas herramientas ayudan al sistema a implementar el protocolo SSL (versiones 2 o 3), así como otros protocolos relacionados con la seguridad, como el protocolo TLS (versión 1).

OpenVPN lo utiliza para implementar la seguridad, utilizando el protocolo SSL/TLS. Este paquete de software es importante para cualquiera que esté planeando usar cierto nivel de seguridad en su máquina con un S.O. libre basado en GNU/Linux. También nos permite crear certificados digitales que se pueden aplicar a nuestro servidor.

Soporta un gran número de algoritmos criptográficos diferentes según la finalidad:

- Algoritmos de cifrado: Blowfish, AES, DES, RC2, RC4, RC5, IDEA, Camellia.
- Algoritmos para funciones hash: MD5, SHA, MD2, MDC-2.
- Algoritmos de intercambio de clave pública: RSA, Diffie-Hellman, DSA.

Además OpenSSL proporciona las herramientas y funciones:

- Generar y gestionar claves asimétricas y simétricas para los distintos algoritmos de cifrado.
- Generar números aleatorios y pseudos aleatorios.
- Utilizar los algoritmos para firmar, certificar y revocar claves.
- Manejar y gestionar formatos de certificados existentes en el mundo (X.509, PEM, PKCS7, PKCS8, PKCS12).
- Cálculo de resúmenes de mensajes.
- Cifrado y descifrado mediante algoritmos de cifrado.
- Manejo de correo S/MIME firmado o cifrado.

Está compuesto de la herramienta de línea de comandos, “OpenSSL”, y de las librerías “SSL” (archivo libssl.a) y “Crypto” (archivo libcrypto.a).

Dependiendo del nivel de seguridad necesario (ya se mencionó en el capítulo 4.4), será necesario o no instalar las librerías “ssl” y “crypto”.

OpenSSL es un software multiplataforma. Su instalación dependerá del S.O. utilizado. Por ejemplo: en Windows al instalar OpenVPN, OpenSSL se instala simultáneamente, sin embargo en Linux se necesita la instalación previa de OpenSSL, y de las librerías necesarias. Para ello es posible hacerlo mediante interfaz gráfica, o bien por comandos:

./config
make
make test
make install

4.7.5. Compresión LZO.

LZO es una librería de compresión de datos diseñada para comprimir y descomprimir en tiempo real. Esto significa que favorece la velocidad frente al ratio de compresión. Esta librería es necesaria para instalar OpenVPN.

Tanto el código como los datos comprimidos están escritos en ANSI C.

La librería “lzo” tiene las siguientes características:

- La descompresión de los datos es simple y rápida.
- No requiere memoria para la descompresión.
- Compresión rápida de los datos.
- Requiere de algún tipo de buffer de 64 kB de memoria para la compresión.
- No necesita ningún tipo de buffer o memoria para la descompresión además de los buffers fuente y destino.
- Permite al usuario realizar un ajuste entre calidad y velocidad de compresión.
- Proporciona niveles de compresión para la realización de una pre-compresión de los datos con el que se logra un ratio de compresión totalmente competitivo.

- El algoritmo es sin perdidas.
- LZO soporta superposición en las compresiones, es decir, realizar varias compresiones simultáneas.

4.7.6. Autenticación de OpenVPN

OpenVPN permite diversos métodos de autenticación, obteniendo con ello una mayor seguridad en la VPN.

Se puede usar el protocolo LDAP (Lightweight Directory Access Protocol) mediante login y password. Para usar este método se utilizan las librerías incluidas en OpenLDAP.

Otra posible opción es la de implantar login mediante usuario y password mediante el paquete PAM (Pluggable Authentication Module), que no ha sido implementado en este TFC pero que ha sido añadido como un punto de mejora para futuras actuaciones. Este paquete no está disponible para servidores con Windows.

PAM puede ser usado para un gran número de aplicaciones, a la vez que permite una gran flexibilidad para el administrador de la red o el desarrollador de la aplicación.

En el directorio "/etc/pam.d/nombre_archivo" se encuentran los archivos que serán usados con PAM.

Hay 4 módulos PAM que son usados en el proceso de autenticación:

- **auth:** autentifican a los usuarios pidiendo un password.
- **Account:** se encargan de controlar que el acceso a ese usuario se permite.
- **Password:** verifican que este es correcto.
- **Session:** configuran y administran las sesiones de usuario

4.7.7. Asignación de las direcciones IP.

Es importante tener en cuenta que este punto se desglosa en 2 dependiendo de si se trabaja con un S.O. Windows o no Este cambio es debido a que Windows tiene una limitación en sus controladores TUN/TAP (TAP Win32) para emular túneles. En este caso se hará subnetting creando subredes con una mascara de red /30. Este es el caso encontrado en nuestro TFC, ya que hay PC's de sobremesa y portátiles con S.O. Windows XP o Windows Vista.

4.7.7.1. Asignación de IP's sin clientes Windows.

En primer lugar se va a ver como está reservado el rango de direcciones IP para redes privadas según la IANA -RFC 1918- (*Tabla 4.3).*

Tabla 4.3 Distribución de IP's por la IANA.

Grupo A	10.0.0.0	10.255.255.255	10/8
Grupo B	172.16.0.0	172.31.255.255	172.16/16
Grupo C	192.168.2.0	192.168.2.255	192.168.2/24

Como este rango de direcciones se puede usar tanto para redes privadas (es decir, para una LAN de una empresa por ejemplo), como para VPN, existe la posibilidad de que se tenga un conflicto de direcciones IP. Ya que si un usuario que está en una empresa se quiere conectar con una VPN, y tanto la LAN de la empresa como la VPN utilizan el mismo rango de direcciones de red, los routers no sabrían hacia donde deberían encaminar los paquetes.

Para evitar este problema no hay una solución 100% efectiva. El único consejo es utilizar un rango intermedio de uno de los grupos. En este TFC se ha mantenido el rango que proporciona OpenVPN, que es el 10.8.0.0, ya que los desarrolladores del software han realizado un estudio del mercado, y han determinado que este es un rango frecuentemente libre para poder ser usado en nuestra VPN.

En la versión 1.6 de OpenVPN, cada vez que se realiza una conexión entre cliente y servidor, se crea una nueva interfaz "tun" en el servidor. Cada cliente si que tendrá su propia interfaz tun. A la vez que habrá un túnel para cada enlace punto a punto.

Sin embargo en la versión 2.0 de OpenVPN se pueden soportar múltiples clientes con una sola interfaz "tun" y un único puerto (1194) en el servidor. Cada cliente sÍ que tendrá su propia interfaz tun. Continuará existiendo un túnel para cada conexión punto a punto, pero ahora, se puede usar una misma dirección IP en el servidor para todos los túneles con los clientes.

En este TFC, se solicitó que serían alrededor de 4 máquinas las que formarían parte de la VPN (inicialmente), así se decidió limitar el rango de IP's, con una máscara de red /24 (255.255.255.0), fijando los 3 primeros bytes para red, y el último para hosts. Quedando 254 hosts disponibles (1 servidor y 253 clientes). Por tanto se ha usado la dirección de red 10.8.0.0/24. De esta forma también queda sobredimensionada para las posibles máquinas que se incorporen en un futuro.

Aunque la red 10.8.0.0 es del grupo A y tiene una mascara de red /16, se ha asignado una máscara de red /24 para limitar el número de hosts que puedan conectarse a nuestra red. Si en un futuro se necesitase ampliar este número de hosts, simplemente habría que modificar la máscara de red de los archivos de configuración de OpenVPN.

En este caso OpenVPN asignará las direcciones IP de la siguiente forma (*Tabla 4.4)*:

Tabla 4.4 Asignación de IP's por OpenVPN.

10.8.0.0	/24	Dirección de red de la VPN
10.8.0.1	/24	Dirección IP del servidor
10.8.0.2	/24	Dirección IP del primer cliente en conectar
10.8.0.3 – 10.8.0.254	/24	Direcciones IP de los siguientes clientes en conectar
10.8.0.255	/24	Dirección de broadcast de la VPN

A continuación se detalla el esquema de la asignación de la mascara de red y el número de hosts disponibles, mostrando que bytes pertenecen al identificador de red, y cuales a los hosts de dicha red *(Fig.4.5)*.

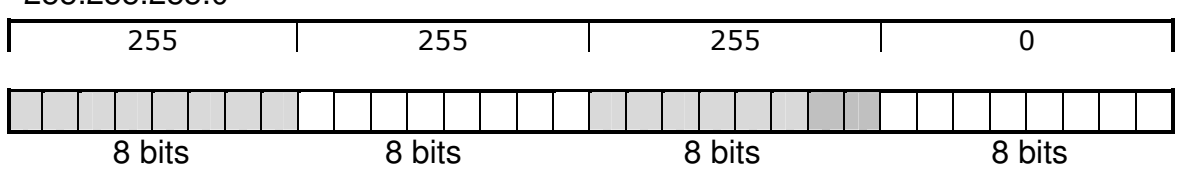

Fig.4.5 Bytes de la máscara de red.

Si se coge el último byte (compuesto de 8 bits).

2^8 = 256 -2 (se elimina la primera "dirección de red" y la última dirección IP "dirección de broadcast") = 254 hosts disponibles

Los tres primeros bytes son para el identificador de la red y el último es para hosts. Para que un byte pertenezca al identificador de red todos sus bits deben estar a 1. Por ejemplo:

11111111 (son los 8 bits de un byte) = 2^8=256 (va desde 0 a 255)

4.7.7.2. Asignación de IP's con clientes Windows (subnetting).

A continuación se explica como OpenVPN hace el subnetting /30. A su vez, el número de redes y hosts disponibles:

Se tiene la dirección IP 10.8.0.0

Al usar una mascara de red de tipo "C" le corresponde 255.255.255.0

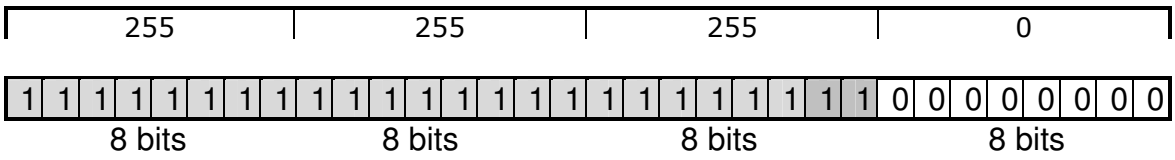

Fig.4.6 Máscara de rango C.

Se coge el último byte que está a 0 y se pone a 1 los 6 bits de la izquierda para llegar a los 30 bits de la máscara /30, *(Fig. 4.7)*.

8 bits + 8 bits + 8 bits + 6 bits = 30 bits

Fig.4.7 bits del último byte de la máscara.

Si se cogen los 6 bits cambiados a 1 se puede calcular el número de subredes:

2^6 = 64-2 = 62 **subredes** (se resta 2 ya que se eliminan la primera y última subred porque contienen el identificador de red y la dirección de broadcast de red respectivamente).

Con los dos bits que están a 0 se calcula el número de hosts disponibles para cada subred:

2^2 = 4-3 = 1**host** (se resta 3 porque se eliminan la primera, segunda, y cuarta, se explica porqué se eliminan en las *Tablas 4.7, 4.8, 4.9, 4.10*).

Si se pone a 1 los 6 bits que se han definido para hacer el subnetting, *(Fig.4.8):*

128	64	32	16	8	4	2	1
1	1	1	1	1	1	0	0

Fig.4.8 bits para el cálculo de subredes y hosts.

128+64+32+16+8+4 = **252**

Entonces la mascara de red queda de la siguiente manera, *(Fig.4.9):*

255	255	255	252

Fig.4.9 Máscara de red.

Para calcular el subnet Id se pasa la IP y la nueva mascara de red de decimal a binario y se hace una operación AND entre ambas.

Las operaciones AND funcionan de la siguiente manera (*Tabla 4.5):*

Tabla 4.5 Tabla de la verdad de operaciones AND.

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

Tabla 4.6 Operación AND entre la IP y la máscara de red.

10.8.0.0	00001010.00001000.00000000.00000000
255.255.255.252	11111111.11111111.11111111.111111100
10.8.0.0	00001010.00001000.00000000.00000000

subnet Id: 10.8.0.0

Ahora se calcula el rango de direcciones IP que tendrá cada subred, elevando al cuadrado el número de bits reservados para hosts, (Fig.4.10):

	128	64	32	16	8	4	2	1
Host	0	0	0	0	0	0	1	1

2^2=4

Fig.4.10 Bits reservados para hosts.

Cada subred estará formada por 4 direcciones IP. En las siguientes tablas se puede ver a que está asignada cada dirección IP en cada una de las subredes, *Tablas 4.7, 4.8, 4.9, 4.10:*

Tabla 4.7 Asignación de IP's de la 1ª subred por OpenVPN.

10.8.0.0	/30	dirección IP de esta subred y dirección IP de red de la VPN
10.8.0.1	/30	Dirección IP del servidor OpenVPN
10.8.0.2	/30	Dirección IP asignada al otro extremos del enlace punto a punto del servidor
10.8.0.3	/30	Dirección IP de broadcast de esta subred

Tabla 4.8 Asignación de IP's de la 2ª subred por OpenVPN.

10.8.0.4	/30	Dirección IP de esta subred
10.8.0.5	/30	Dirección IP virtual del servidor OpenVPN
10.8.0.6	/30	Dirección IP asignada al primer cliente en conectar a la VPN
10.8.0.7	/30	Dirección de broadcast de esta subred

Tabla 4.9 Asignación de IP's desde la 3ª hasta la 63ª subred. Seguiría la mima estructura que en la 2ª.

10.8.0.X	/30	Dirección IP de esta subred
10.8.0.X+1	/30	Dirección IP virtual del servidor OpenVPN
10.8.0.X+2	/30	Dirección IP asignada al cliente que se conecta a la VPN
10.8.0.X+3	/30	Dirección de broadcast de esta subred

Tabla 4.10 Asignación de IP's de la 64ª subred por OpenVPN.

10.8.0.252	/30	Dirección IP de esta subred
10.8.0.253	/30	Dirección IP virtual del servidor OpenVPN
10.8.0.254	/30	Dirección IP asignada al otro extremos del enlace punto a punto del servidor
10.8.0.255	/30	Dirección IP de broadcast de red de la VPN

Todo esto hace prescindir de un gran número de direcciones IP, pero es la única forma de que OpenVPN pueda trabajar con clientes de cualquier S.O. (en especial Windows).

Si se está seguro que todos los clientes que se van a conectar a la VPN no utilizan Windows, se puede utilizar el comando *--ifconfig-pool-linear*, y de esta forma el subnetting quedará anulado, usando la máscara de red que se haya asignado inicialmente a nuestra red. Es importante recalcar que si se usa esta función, cualquier máquina con Windows no podrá conectarse al servidor VPN.

4.8. Instalación de OpenVPN.

4.8.1. Introducción.

Se hará una explicación de las instalaciones realizadas tanto en el servidor Suse, como en los clientes con Linux (Ubuntu) y Windows (Windows XP), junto a imágenes de apoyo. Se revisarán o instalarán en caso de que sean necesarios los paquetes previos necesarios a la instalación de OpenVPN. Los archivos de configuración que vienen por defecto, serán modificados a nuestras necesidades. Por tanto se explicará cada una de las líneas de estos archivos, y las posibilidades que ofrecen.

Como ya se ha comentado anteriormente, OpenVPN es un software libre, y por tanto se puede descargar directamente desde la sección de download de su página Web: http://openvpn.net. En ella se encuentran diversas versiones de la aplicación para los distintos S.O. disponibles. En la sección Books, se puede encontrar un completo manual de OpenVPN y también hay disponible una sección de FAQ (Frequently Asked Question – preguntas más frecuentes), y un foro para resolver dudas, etc.

Para Windows existe otra web, http://openvpn.se, de la que se puede descargar el software con su aplicación gráfica (GUI). Para este TFC se ha utilizado la primera web tanto para Linux como para Windows

Linux también permite la instalación de la aplicación OpenVPN desde los repositorios del S.O. Suse además tiene una aplicación llamada Yast (acrónimo de Yet another Setup Tool → otra herramienta de configuración más. Es una aplicación para la distribución de OpenSuse incluida en este, que facilita la administración del sistema y la instalación de software del ordenador), desde la que permite instalar aplicaciones o actualizaciones.

Para la seguridad en OpenVPN, necesitaremos una Unidad Certificadora (CA) que se encargará de crear los certificados y las claves de seguridad. Además la CA necesita un certificado y clave maestra para generar los del servidor y clientes. Se ha instalado la CA en el servidor, pero se puede hacer en distintas máquinas. Para crear las claves privadas se usa el protocolo Diffie Hellman, el cual genera claves simétricas, y permite el intercambio de las claves privadas de forma segura, mediante encriptación.

4.8.2. Prerrequisitos de la instalación de OpenVPN.

Para instalar OpenVPN en Linux no hay problema, ya que soporta prácticamente cualquier tipo de Linux (Suse, Ubuntu, RedHat, Fedora, etc). Sin embargo en Windows está algo más limitado, ya que únicamente soporta las versiones 2000, XP y Vista.

Algunos requisitos previos del S.O. en el que se vaya a instalar OpenVPN son:

- Soportar los drivers TUN/TAP. Casi todas las versiones de Linux con un kernel superior o igual a 2.4, soporta estos drivers. El problema es si se tiene alguna versión de Linux con Kernel inferior a 2.4, o un kernel que se haya modificado y no se le haya dado soporte para estos drivers. Para solucionar esto, se puede instalar la aplicación VTUN disponible en http://vtun.sourceforge.net/tun.
- Será necesario tener instaladas las librerías OpenSSL, en caso de querer usar algún tipo de cifrado, ya sea simétrico o asimétrico. Es muy recomendable tener instalada esta librería y usar algún tipo de cifrado (mucho más seguro el asimétrico). En caso de no tener instaladas dichas librerías, se pueden encontrar en: http://www.openssl.org.

- En caso de querer usar la compresión de datos en tiempo real, se necesita tener instaladas las librerías LZO. En caso de necesitarla se puede encontrar en la Web: http://www.oberhumer.com/opensource/lzo.

4.8.3. Instalación de OpenVPN en OpenSuse 11.0.

El servidor de este TFC se encuentra sobre el S.O. OpenSuse 11.0. Para realizar la instalación de OpenVpn se debe hacer desde el Yast de Suse. Para ello ir al Yast, instalar/desinstalar software, y buscar el repositorio de OpenVpn. Marcarlo para realizar su instalación. Esto se hará tanto para el caso del servidor, como para un cliente. El aspecto una vez instalado es el que aparece en la *Fig.4.11.*

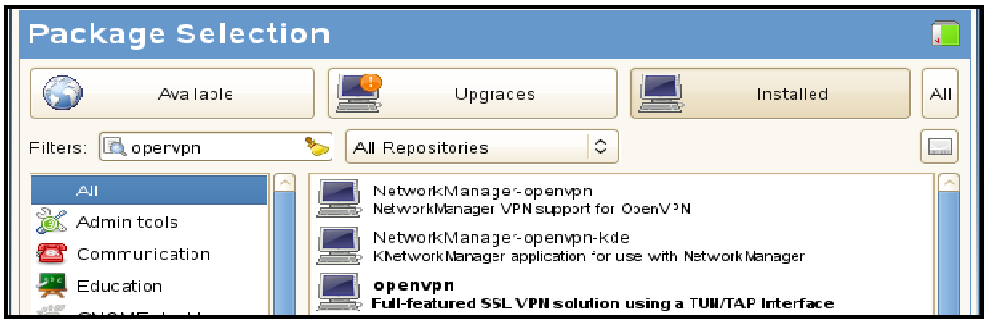

Fig.4.11 Instalación de OPENVPN en SUSE.

4.8.3.1. Necesidad de usar claves y certificados.

Para construir una VPN con OpenVPN 2.0 es necesario crear una PKI (Infraestructura de Clave Publica - Public Key Infrastructure). Esta PKI está formada por:

- Un certificado (conocido como clave pública) y una clave privada para el servidor y para cada cliente.
- Un Certificado para la CA y su clave, que se usará para firmar los certificados del servidor y los clientes.

A partir de ahora todo el proceso de generación de claves, certificados, y firma digital, se realizará desde el servidor u otra máquina designada para ello, que no sea un cliente. Esto es por seguridad, ya que no es conveniente que un cliente contenga todas las claves y certificados. En este TFC, se ha hecho desde la máquina que hace de servidor de la VPN. Por tanto será servidor de la VPN y Unidad Certificadora.

4.8.3.2. Generación de clave y certificado para la CA.

Una vez instalado OpenVpn, la administración de la PKI para la creación de la clave y el certificado de la CA, se hará mediante unos scripts que vienen con OpenVPN. Estos scripts se encuentran en la carpeta /usr/share/openvpn/easy-rsa/2.0.

Se debe crear como usuario root la carpeta /etc/openvpn/easy-rsa-V2.0/ en consola mediante el comando:

#mkdir /etc/openvpn/easy-rsa-V2.0/

Y después se copian todos los scripts que vienen por defecto a este directorio:

#cp /usr/share/openvpn/easy-rsa/2.0/ /etc/openvpn/easy-rsa-V2.0*

Mediante el siguiente comando se crearÁ un nuevo directorio donde se almacenarán las claves privadas, los archivos de requerimiento de certificado (.csr), los certificados (.crt), y otros archivos como el serial y el index.txt.

#mkdir -p /etc/openvpn/easy-rsa-V2.0/keys

Se debe entrar al directorio en el que están los scripts:

#cd /etc/openvpn/easy-rsa-V2.0

Y editar el archivo vars, en el que se encuentra la ruta del fichero donde se crearán las claves y certificados, el tamaño de las claves privadas (del servidor, cliente y CA), y los valores por defecto de algunos campos que se debe modificar. Para ello:

#kwrite vars

Se modifican los siguientes parámetros con los valores de nuestra VPN:

export KEY_COUNTRY="ES"
export KEY_PROVINCE="BCN"
export KEY_CITY="CTF"
export KEY_ORG="EPSC"
export KEY_EMAIL=pbruna@fa.upc.edu

También se puede cambiar el tamaño de la clave. El valor predeterminado es 1024, que ya es suficiente, pero si se quiere aumentar la seguridad, es posible hacerlo aumentando a 2048 o 4096. Esto no afectará negativamente al funcionamiento de la VPN, y sólo afectará en la generación de los parámetros Diffie Hellman (proceso que será explicados a continuación), y una validación algo más lenta en la negociación a través del protocolo Handshake de SSL/TLS.

export KEY_SIZE=2048

Una vez definidos nuestros valores, se pasa a inicializar la PKI de la siguiente manera:

#source ./vars

Y aparece la siguiente nota:

NOTE: If you run ./clean-all, I will be doing a rm -rf on /etc/openvpn/easy-rsa-V2.0/keys

Y se configura un entorno nuevo:

 # ./clean-all

Ahora se generan los parámetros Diffie Hellman (explicados en los anexos) que se utiliza para la encriptación:

#./build-dh

Se crea el certificado y la clave privada para la CA con el comando:

#./pkitool --initca

Con este procedimiento se han generado 3 ficheros:

- **ca.crt:** es el certificado público de la CA.
- **ca.key:** es la clave privada de la CA, la cual debe mantenerse protegida porque es la clave más importante de toda la PKI.
- **ph1024.pem:** generado a partir de los parámetros Diffie Hellman que se utiliza para poder intercambiar una clave entre dos participantes de manera segura.

4.8.3.3. Generación de clave y certificado para el servidor

Para generar la clave privada y el certificado del servidor se debe escribir el siguiente comando en la consola de comandos:

#./pkitool --server servidor

“Servidor” corresponde al nombre que se necesite dar.

Los valores son tomados del archivo vars, que en este caso, al usar la misma máquina para la CA y el servidor, no es necesario modificar previamente, ya que se ha echo para la CA.

Se han generado 3 ficheros nuevos:

- **servidor.crt:** es el certificado público del servidor.

- **servidor.key:** es la clave privada del servidor, que debe permanecer protegida.
- **servidor.csr:** este fichero sirve para poder crear el certificado del servidor en otra máquina que pueda crearlo y firmarlo, ya que este fichero tiene toda la información que le hace falta.

4.8.3.4. Generación de clave y certificado para el cliente

Para generar la clave y el certificado del servidor se debe escribir el siguiente comando en la consola de comandos*:*

#./pkitool cliente1

Los valores son tomados del archivo vars, con el nombre del argumento que se ha escrito como parámetro (en este caso se ha puesto cliente1)**.** Cada vez que se necesite añadir un nuevo cliente a la VPN, se debe crear un nuevo certificado y una nueva clave para ese cliente. Para ello se debe ejecutar los siguientes comandos:

#source ./vars
#./pkitool clienteN

Se han generado 3 ficheros nuevos:

- **cliente.crt:** es el certificado público del cliente.
- **cliente.key:** es la clave privada del cliente, que debe permanecer protegida.
- **cliente.csr:** este fichero sirve para poder crear el certificado del cliente en otra máquina que pueda crearlo y firmarlo, ya que este fichero tiene toda la información que le hace falta.

4.8.3.5. Generación de la firma digital HMAC.

Con la firma digital HMAC se garantiza un nivel de seguridad extra al SSL/TLS, para protegernos frente a posibles ataques que puedan hacer vulnerable nuestro sistema.

Todos los paquetes deberán tener la firma digital para garantizar la autenticación del emisor, y si llega un paquete sin firma, será rechazado. Este nivel de seguridad es conveniente, ya que así nos se puede proteger de posibles ataques.

El uso de esta firma conlleva crear una nueva clave de seguridad privada que deberán tener tanto los clientes como el servidor. Para ello con el usuario root, desde el servidor se crea la nueva clave:

#openvpn --genkey --secret ta.key

La clave generada es “ta.key”. Esta clave deberá estar en el servidor, y en todos los clientes. Para ello deberá ser enviada a los clientes por un canal seguro. En el servidor se debe añadir en el archivo server.conf la línea:

tls-auth ta.key 0

Y en el cliente se debe añadir en el archivo client.conf (para Linux) o en el client.ovpn (para Windows) la línea:

tls-auth ta.key 1

En ambos casos se tendrá que poner la ruta del archivo en caso de que el ta.key no se encuentre en el mismo directorio que los archivos de configuración. Por ejemplo: /etc/openvpn/keys/ta.key.

Añadiendo esta opción en ambos extremos de la VPN se añade un HMAC extra a todos los paquetes del protocolo Handshake de SSL/TLS para dar un nivel más de autenticación. De esta manera se podría decir que se está añadiendo un firewall HMAC, ya que cualquier paquete que no tenga la misma HMAC será tirado sin realizar ningún tipo de procesamiento ni malgastar recursos extra.

4.8.3.6. Archivos de los certificados y claves.

En la Tabla *4.11* se ve una tabla de todos los archivos que se han generado y quien deberá tenerlos. Los archivos que se han de enviar a los clientes, debe ser mediante un canal seguro como podría ser SSH. Los archivos cliente1.crt y cliente1.key se envían al cliente 1, los que se hayan creado con el nombre cliente2.crt y cliente2.key al cliente 2, y así sucesivamente para todos los clientes de la VPN.

Tabla 4.11 Archivos de certificados y claves.

Archivo	Poseedor	Función	Secreto
ca.crt	Servidor y todos los clientes	Certificado para root	No
ca.key	Unidad certificadora	Clave para root CA	Si
dh2048.pem	Servidor	Parámetros Diffie Hellamn	No
servidor.crt	Servidor	Certificado para servidor	No
servidor.key	Servidor	Clave privada para servidor	Si
clienteX.crt	ClienteX	Certificado para clienteX	No
clienteX.key	ClienteX	Clave privada para clienteX	Si
ta.key	Servidor y todos los clientes	Clave con firma HMAC para nivel extra de seguridad	Si

4.8.3.7. Lista de revocación de certificados (CRL).

Revocar un certificado consiste en anular ese certificado para que no se pueda usar, y por tanto no se permita la conexión a la VPN. Existen varios motivos por los que se puede querer revocar un certificado, algunos de ellos son:

- La clave privada asociada al certificado esta comprometida o ha sido robada.
- El usuario de una clave privada que además está cifrada, olvidó la contraseña de la clave.
- Se quiere cancelar el acceso de un usuario a la VPN.

A continuación se explican los pasos para revocar un certificado. Como usuario root se abre una consola de comandos y se entra a la carpeta easy-rsa-V2.0:

#cd /etc/openvpn/easy-rsa-V2.0

Y se reinician las variables de OpenVPN:

#source ./vars

Posteriormente se debe revocar el certificado que se desee anular (en este caso es cliente 1). Para ello se escribe el siguiente comando (*Fig.4.18)*:

#./revoke-full cliente1

```
sisif:/etc/openvpn/easy-rsa-V2.0 # ./revoke-full cliente1
Using configuration from /etc/openvpn/easy-rsa-V2.0/openssl.cnf
Revoking Certificate 07.
Data Base Updated
Using configuration from /etc/openvpn/easy-rsa-V2.0/openssl.cnf
cliente1.crt: /C=ES/ST=BCN/L=CTF/O=EPSC/CN=cliente1/emailAddress=pbruna@fa.upc.edu
error 23 at 0 depth lookup:certificate revoked
```

Fig.4.18 Revocado de un certificado.

En caso de aparecer la *Fig.*4.18, ese certificado ya estará revocado. Pero a veces puede pasar que haya algún problema y nos muestre la *Fig.*4.19*:*

```
sisif:/etc/openvpn/easy-rsa-V2.0 # ./revoke-full cliente1
Using configuration from /etc/openvpn/easy-rsa-V2.0/openssl.cnf
error on line 282 of config file '/etc/openvpn/easy-rsa-V2.0/openssl.cnf'
24074:error:0E065068:configuration file routines:STR_COPY:variable has no value:co
nf_def.c:629:line 282
Using configuration from /etc/openvpn/easy-rsa-V2.0/openssl.cnf
error on line 282 of config file '/etc/openvpn/easy-rsa-V2.0/openssl.cnf'
24075:error:0E065068:configuration file routines:STR_COPY:variable has no value:co
nf_def.c:629:line 282
cliente1.crt: /C=ES/ST=BCN/L=CTF/O=EPSC/CN=cliente1/emailAddress=pbruna@fa.upc.edu
error 8 at 0 depth lookup:CRL signature failure
24077:error:04077077:rsa routines:RSA_verify:wrong signature length:rsa_sign.c:154
:
24077:error:0D0C5006:asn1 encoding routines:ASN1_item_verify:EVP lib:a_verify.c:16
8:
```

Fig.4.19 Problemas al revocar certificados.

En caso de que suceda esto, se debe abrir el archivo */etc/openvpn/easy-rsa-V2.0/openssl.cnf* y comentar la última sección *(Fig.*4.20*):*

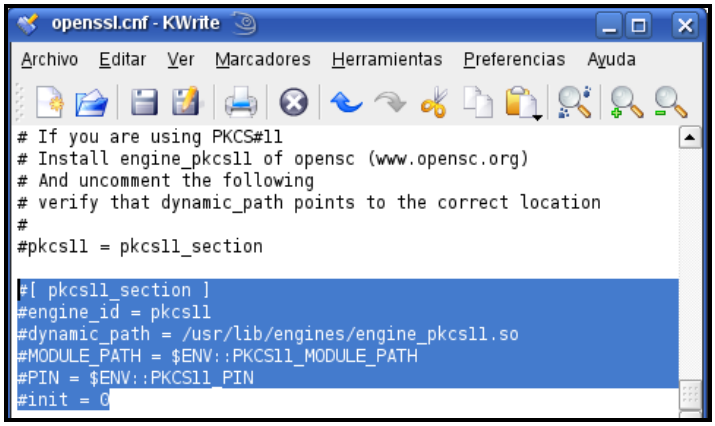

Fig.4.20 Fichero openssl.cnf.

Posteriormente, en la consola de comandos volver a revocar el certificado como se hizo anteriormente:

#./revoke-full cliente1

Al ejecutar el comando revoke-full se crea un archivo llamado crl.pem en el directorio keys de OpenVPN que contiene el CRL que se ha creado. Este fichero debe estar en el servidor. En el server.conf ubicado en el servidor, se debe hacer referencia a este archivo para que al conectarse un cliente, se lea dicho archivo y se sepa si el certificado con el que se intenta realizar la conexión por parte del cliente, es válido o está revocado, y en caso de estar revocado no se puedan conectar a la VPN. Para ello, editar el server.conf, y escribir la siguiente línea:

crl-verify /etc/openvpn/keys/crl.pem

En caso de revocar un certificado que haya sido cifrado, se puede generar un nuevo certificado con el mismo nombre del certificado revocado, ya que al generarlo nuevamente, será distinto al revocado.

4.8.4. Configuración de OpenVPN en el servidor OpenSuse.

En el servidor existe un archivo llamado server.conf que es el encargado de la configuración de la VPN en el servidor. Se puede encontrar un archivo de ejemplo en */usr/share/doc/packages/openvpn/sample-config-files*, que se ha de copiar a /etc/openvpn. Se puede copiar este archivo mediante la consola de comandos:

#cd /usr/share/doc/packages/openvpn/sample-config-files
#cp server.conf /etc/openvpn

Desde ahora se trabaja con la copia del server.conf que se ha echo. Editar el archivo con un editor de texto:

#cd /etc/openvpn
#kwrite server.conf

Se modifica el archivo añadiendo la ruta de los certificados y claves:

ca /etc/openvpn/easy-rsa-V2.0/keys/ca.crt
cert /etc/openvpn/easy-rsa-V2.0/keys/servidor.crt
key /etc/openvpn/easy-rsa-V2.0/keys/servidor.key
dh /etc/openvpn/easy-rsa-V2.0/keys/dh2048.pem
tls-auth /etc/openvpn/easy-rsa-V2.0/keys/ta.key 0

Una vez establecidas las rutas, se guarda y se cierra el archivo.

4.8.4.1. Archivo de configuración del servidor.

En este apartado se ve el fichero de configuración server.conf situado en el servidor.

port 1194
dev tun
proto udp
tls-server
persist-key
persist-tun
dh /etc/openvpn/easy-rsa-V2.0/keys/dh2048.pem
ca /etc/openvpn/easy-rsa-V2.0/keys/ca.crt
cert /etc/openvpn/easy-rsa-V2.0/keys/servidor.crt
key /etc/openvpn/easy-rsa-V2.0/keys/servidor.key
tls-auth /etc/openvpn/easy-rsa-V2.0/keys/ta.key 0
ifconfig-pool-persist ipp.txt 60
client-to-client

mode server
ifconfig-pool 10.8.0.4 10.8.0.120
route 10.8.0.0 255.255.255.0
push " route 10.8.0.0 255.255.255.0"
comp-lzo
keepalive 10 120
status openvpn-status.log
verb 6

A continuación se explicará cada línea del fichero.

- **port 1194**: se usa el puerto 1194 para OpenVPN.
- **dev tun:** se va a usar un túnel mediante una interfaz virtual "tun" de los drivers TUN/TAP.
- **proto udp:** se usa udp como protocolo de transporte.
- **tls-server:** será el servidor en el establecimiento del protocolo TLS.
- **persist-key:** hace que si el servidor OpenVPN es reiniciado, no tenga que volver a leer las claves.
- **persist-tun**: si se reinicia el servidor, el túnel no tiene que ser cerrado y reabierto.
- **dh /etc/openvpn/easy-rsa-V2.0/keys/dh2048.pem**: carga los parámetros Diffie Hellman.
- **ca /etc/openvpn/easy-rsa-V2.0/keys/ca.crt:** carga el certificado público de la CA
- **cert /etc/openvpn/easy-rsa-V2.0/keys/servidor.crt**: carga el certificado del servidor.
- **key /etc/openvpn/easy-rsa-V2.0/keys/servidor.key:** carga la clave privada del servidor.
- **tls-auth /etc/openvpn/easy-rsa-V2.0/keys/ta.key 0:** carga la firma digital. El 0 indica que somos el servidor.
- **ifconfig-pool-persist ipp.txt 60:** guarda siempre la misma IP para un cliente debido al nombre (ejemplo: cliente1).
- **client-to-client:** permite que los clientes puedan verse entre si.
- **mode server:** trabajará en modo servidor para aceptar múltiples clientes.
- **ifconfig-pool 10.8.0.4 10.8.0.120:** se limita el rango de IP para los clientes desde la 10.8.0.4 hasta la 10.8.0.120, ya que no se tienen tantos hosts por ahora que se vayan a conectar a la VPN.
- **route 10.8.0.0 255.255.255.0:** Agrega la ruta de red 10.8.0.0 a la interfaz virtual TUN/TAP.
- ***push "route 10.8.0.0 255.255.255.0":*** este comando va relacionado obligatoriamente con el comando client-to-client del client.conf (cliente). Permite que el cliente agregue la ruta de red indicada a su interfaz virutal TUN/TAP.
- **comp-lzo**: uso de la librería de compresión LZO.
- **keepalive 10 120:** el servidor enviará un ping cada 10 segundos, y en caso de no recibir respuesta en 120 segundos, deducirá que ese cliente se ha caído.

- **status openvpn-status.log**: indica el fichero que contendrá la información de estado del túnel.
- **verb 6:** indica el grado de detalle de información de estado del túnel.

4.8.5. Conexión de OpenVPN en el servidor OpenSuse.

Para iniciar el servidor hay dos opciones. La primera es mediante comandos:

#service openvpn start

Y la segunda es con el Yast de OpenSuse de manera que OpenVPN arrancará automáticametne al iniciar el servidor. Para ello se debe acceder mediante YAST a System Services y activar el servicio OpenVPN (*Fig.*4.21).

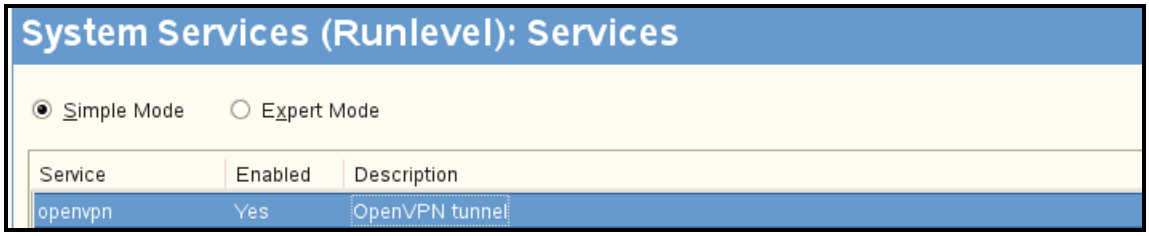

Fig.4.21 Servicio OpenVPN en Yast.

Falta abrir el puerto 1194 para que los clientes se puedan conectar, pero este punto se explicará más detalladamente en el capítulo 4 del anexo.

4.8.6. Instalación de OpenVPN en Ubuntu.

La instalación en Debian o en distribuciones basadas en Debian, como Ubuntu, se realiza mediante el comando “apt”. Es muy simple, ya que sólo hay que poner en consola el siguiente comando:

apt-get install openvpn

Una vez instalado, los archivos copiados de OpenVPN son los mostrados en la *Fig.*4.23.

Full Path and File Installed by OpenVPN	Function
/etc/openvpn	Directory containing configuration files
/etc/network/if-up.d/openvpn /etc/network/if-down.d /etc/network/if-down.d/openvpn	Start/stop openvpn when the network goes up/down
/etc/init.d/openvpn	Start/stop script for services
/sbin/openvpn	The binary
/usr/share/doc/openvpn	Documentation files
/usr/share/man/man8/openvpn.8.gz	Manual page
/usr/share/doc/openvpn/examples/sample-config-files	Example configuration files
/usr/share/doc/openvpn/examples/sample-keys	Example keys
/usr/share/doc/openvpn/examples/easy-rsa	easy-rsa—a collection of scripts useful for creating tunnels
/usr/share/doc/openvpn/changelog.Debian.gz /usr/share/doc/openvpn/changelog.gz	Version history
/usr/share/openvpn/verify-cn	verify-cn function (revoke command)
/usr/lib/openvpn/openvpn-auth-pam.so /usr/lib/openvpn/openvpn-down-root.so	Libraries for PAM-Authentication and chroot mode

Fig.4.23 Archivos instalados de OpenVPN y su función en sistemas Debian.

4.8.7. Configuración de OpenVPN en el cliente OpenSuse o Ubuntu.

Existen dos formas de usar un cliente OpenVPN en Linux:

- Sin control de usuario: en el que se configura todo a mano y el programa corre en el background del sistema. No solicita ningún password al usuario.
- Con control de usuario: en el que se instala alguna aplicación que proporciona un control del usuario que accede a la VPN. Solicita un password al usuario.

Sin control de usuario ha sido la opción implantada en el TFC porque así se especificó. Sin embargo se explica adicionalmente la opción con control de usuario para que si en el futuro se decide mejorar la seguridad en este sentido sea posible.

4.8.7.1. Sin control de usuario.

Como se indicó en el capítulo 4.8, una vez que se ha recibido en el cliente por parte del servidor los certificados, claves, y firma que le corresponden, se debe editar el archivo client.conf que es el encargado de la configuración de la VPN en el cliente. Se puede encontrar un archivo de ejemplo en */usr/share/doc/ packages/openvpn/sample-config-files*, que se ha de copiar a /etc/openvpn. Es posible copiar este archivo mediante la consola de comandos:

#cd /usr/share/doc/packages/openvpn/sample-config-files
#cp server.conf /etc/openvpn

Desde ahora se trabaja con la copia del server.conf. Editar el archivo:

#cd /etc/openvpn
#kwrite server.conf

Ahora se modifica la ruta de la ubicación donde se han almacenado los archivos anteriormente nombrados, recibidos desde el servidor. Por ejemplo:

ca /etc/openvpn/ca.crt
cert /etc/openvpn/cliente1.crt
key /etc/openvpn/cliente1.key
tls-auth /etc/openvpn/easy-rsa-V2.0/keys/ta.key 1

Además se les debe dar permisos a los archivos:

#chmod 644 ca.crt
#chmod 644 cliente1.crt
#chmod 600 cliente1.key

Y modificar la línea *remote my-server-1 1194,* en el que my-server es la dirección IP del servidor VPN. Paralelamente se ha de abrir el puerto 1194 en el firewall del S.O.

Para conectar un cliente al servidor, únicamente se ha de abrir una consola de comandos y escribir:

#service openvpn start

En ese momento se lee el archivo client.conf y realiza la conexión. Nos asigna una IP privada de la VPN, y crea una interfaz de red virtual TUN. Si se ejecuta un ifconfig en consola, se pueden ver estos datos.

4.8.7.2. Con control de usuario.

Para dar control de usuario en la conexión a la VPN se puede utilizar una herramienta gráfica como knetworkmanager.

En primer lugar se marcan los paquetes necesarios para instalar, *Fig.*4.24.

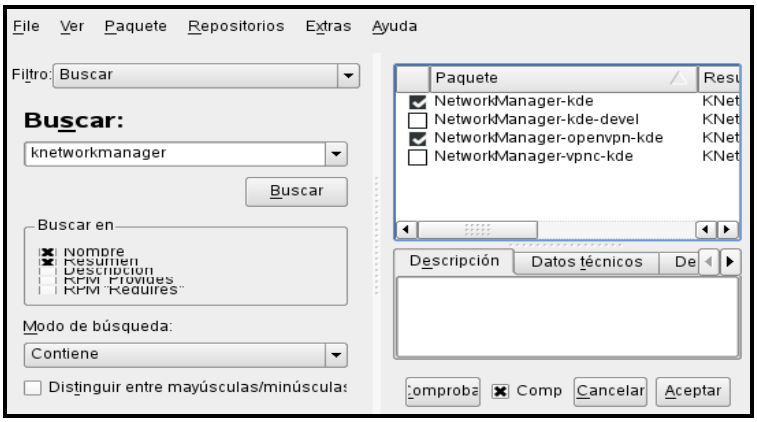

Fig.4.24 Instalación paquetes para OpenVPN con control de usuario.

Una vez instalado knetworkmanager, se ha de configurar la conexión de red. Se debe hacer mediante YAST → Dispositivos de red → Ajustes de la red, pestaña Opciones globales, y cambiar el método que viene por defecto (ifup) por Network Manager, y pulsar terminar *(Fig.*4.25*).*

Fig.4.25 Configuración conexión de red.

Se ejecuta KnetworkManager y aparece un icono en la bandeja del sistema. Pulsar click derecho del ratón→ Editar conexiones → Nueva conexión → VPN.

- En gateway se pone la dirección IP del servidor OpenVPN, que en nuestro caso es 10.8.0.1
- En port se introduce el puerto que se ha utilizado en nuestra VPN. En nuestro caso el 1194, que es el que viene por defecto.
- En CA file se busca nuestro archivo ca.crt.
- En certificado se busca el certificado del cliente, en el caso del cliente 1 es el cliente1.crt
- Y en key la clave del cliente, para el cliente 1 es el cliente1.key

- En la pestaña de información opcional se ha de marcar las opciones que se hayan usado o modificado en la VPN (compresión LZO, TAP, TCP, etc.).

Una vez configurado todo, presionar siguiente y dar un nombre a la conexión. Para conectar, pulsar el botón derecho del mouse sobre el icono de KnetworkManager → Iniciar conexión VPN y seleccionar la conexión creada anteriormente.

La primera vez que se selecciona una conexión aparecerá una ventana que solicitará un password, que será guardado para esa conexión en caso de marcar la opción correspondiente. Por tanto cada vez que se realice esta conexión, se debe introducir el mismo password.

4.8.7.3. Archivo de configuración del cliente.

En este apartado se visualiza el fichero de configuración client.conf (en un cliente con Linux), o el client.ovpn (en clientes con Windows) situados en el cliente.

dev tun
proto udp
remote 147.83.12.65 1194
tls-client
nobind
persist-key
persist-tun
ca /etc/openvpn/ca.crt
cert /etc/openvpn/cliente2.crt
key /etc/openvpn/cliente2.key
tls-auth /etc/openvpn/ta.key 1
comp-lzo
resolv-retry infinite
keepalive 10 120
status openvpn-status.log
verb 6

A continuación se explicará cada línea del fichero.

- **dev tun:** se va a usar un túnel mediante una interfaz virtual “tun” de los drivers TUN/TAP.
- **proto udp:** que se usa udp como protocolo de transporte.
- **remote 147.83.12.65 1194:** indica la dirección IP física de la máquina a la que se ha de conectar para establecer el túnel (al servidor), y que utilizará el puerto 1194.
- **tls-client:** indica que esta máquina actuará como cliente durante el establecimiento del protocolo TLS.

- **nobind:** permite que OpenVPN no se vincule a la IP local y puerto de esta máquina. Esta directiva se utiliza cuando se utiliza la directiva "remote".
- **persist-key:** hace que si el cliente OpenVPN es reiniciado, no tenga que volver a leer las claves.
- **persist-tun**: si se reinicia el cliente, el túnel no tiene que ser cerrado y reabierto.
- **ca /etc/openvpn/ca.crt:** carga el certificado público de la CA
- **cert /etc/openvpn/cliente2.crt**: carga el certificado del servidor.
- **key /etc/openvpn/cliente2.key:** carga la clave privada del servidor.
- **tls-auth /etc/openvpn/ta.key 1:** carga la firma digital. El 1 indica que somos un cliente.
- **comp-lzo**: uso de la librería de compresión LZO.
- **resolv-retry infinite:** el cliente intentará de manera indefinida resolver la dirección o nombre de host dado por la directiva "remote".
- **keepalive 10 120:** el cliente enviará un ping cada 10 segundos, y en caso de no recibir respuesta en 120 segundos por parte del servidor, deducirá que este ha caído.
- **status openvpn-status.log**: indica el fichero que contendrá la información de estado del túnel.
- **verb 6:** indica el grado de detalle de información de estado del túnel.

4.8.8. Instalación de OpenVPN en Windows.

En primer lugar hay que descargar el archivo de la Web de OpenVPN. Una vez descargado se procede a su instalación ejecutándolo. Actualmente ya viene en el mismo paquete tanto el OpenVpn como el OpenVpn-gui (es su interfaz gráfica), ambos son necesarios para su instalación. Dejar todas las opciones que vienen por defecto durante todo el proceso.

Una vez finalizada la instalación, se crea un icono de la GUI de OpenVPN para su acceso de forma gráfica.

4.8.9. Configuración de OpenVPN en Windows.

Al terminar la instalación, lo que se debe hacer es configurar OpenVPN. Acceder a la carpeta que se ha creado a partir de la instalación hecha en *C:\Program Files\OpenVPN\sample-config* y coger el archivo de configuración para clientes client.ovpn, y copiarlo a la carpeta *C:\ProgramFiles\OpenVPN* config, el cual se modificará para realizar la conexión al servidor de la VPN.

Como se indicó en el capítulo 4.8, una vez se ha recibido en el cliente por parte del servidor los certificados, claves, y firma que le corresponden, se deben copiar a la carpeta C:\Programs Files\OpenVPN\easy-rsa. Los archivos son:

ca.crt: certificado de la unidad certificadora
cliente1.crt: certificado del cliente 1
cliente1.key: clave del cliente 1

ta.key: firma digital

Seguidamente se debe editar el archivo client.ovpn ubicado en C:\Programs Files\OpenVPN\config y modificar la ruta de la ubicación de esos archivos. Para el caso de Windows es importante indicar que al poner la ruta, las barras utilizadas deben ser dobles tal y como se muestra a continuación:

ca c:\\Programs Files\\OpenVPN\\easy-rsa \\ca.crt
cert c: \\Programs Files\\OpenVPN\\easy-rsa \\cliente1.crt
key c: \\Programs Files\\OpenVPN\\easy-rsa \\cliente1.key
tls-auth c:\\Programs Files\\OpenVPN\\easy-rsa\\ta.key 1

Y se debe modificar la línea: remote my-server-1 1194, en el que my-server es la dirección IP del servidor VPN.

Se ha de abrir el puerto 1194 en el firewall.

4.8.10. Conexión de OpenVPN en Windows.

Para conectarnos como cliente, previamente se ha instalado la aplicación, se ha configurado el cliente correctamente y se sabe que el servidor está levantado esperando las peticiones de conexión por parte de los clientes.

Hacer clic con el botón derecho del mouse sobre el icono que se ha creado en la barra de herramientas de Windows (son 2 PC's de color rojo con una bola del mundo), que representa a la GUI de OpenVPN, y seleccionar la opción de "Connect".

Un vez que se ha realizado la conexión, el icono que estaba con los dos PC's de color rojo, cambia de color a verde, y si se sitúa el cursor encima, nos muestra que la conexión esta realizada como cliente, la fecha y hora en que se a realizado la conexión, y la IP que ha sido asignada *(*Fig.4.26*).*

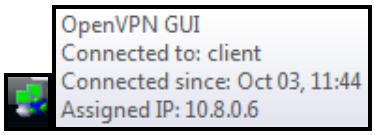

Fig.4.26 Conexión establecida como cliente.

Si se pierde la conexión en algún momento, el icono pasa a ser de color amarillo, y luego a color rojo.

Una vez se realiza la conexión desde el cliente, se puede ver como se ha añadido una interfaz de red virtual para TUN/TAP (Fig.4.27), *"Conexión de área local 4".* Esta interfaz tendrá una dirección MAC (Fig.4.28), como si de una interfaz física de red se tratase.

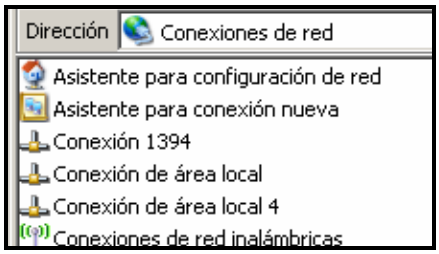

Fig.4.27 Interfaz de red virtual de OpenVPN en Windows XP

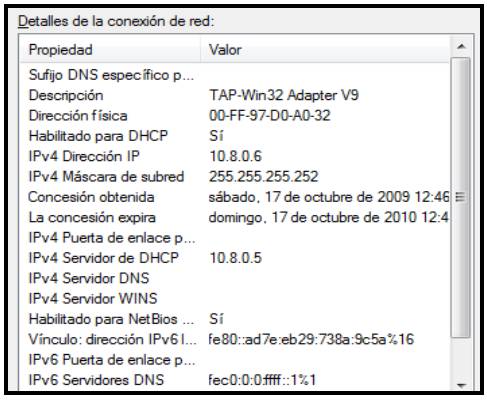

Fig.4.28 Dirección MAC de la interfaz de red virtual.

Es recomendable configurar el S.O Windows para que al iniciarse cargue automáticamente OpenVPN. Para ello hay que copiar el acceso directo de la aplicación en la carpeta Inicio → Programas → Inicio. De esta manera el icono quedará cargado en la barra de tareas de Windows a la espera de que el usuario realice la conexión manualmente hacia la VPN.

5. SAMBA

5.1. Introducción.

Una vez instalada la solución para la VPN y dado el escenario del TFC en el que existen varios ordenadores con diferentes S.O., es necesario implantar un software multiplataforma que permita compartir recursos en red, siendo estos visibles solo para los ordenadores conectados a la VPN. La solución escogida es Samba.

Samba es una implementación libre del protocolo de archivos compartidos de Microsoft Windows, llamado SMB (Server Message Block). Esto permite que ordenadores con Linux o Mac actúen como clientes de datos en redes de Windows o servidores de datos con otros sistemas Linux. Una condición indispensable es que todos estos ordenadores se encuentren dentro del mismo Workgroup o grupo de trabajo (fácilmente configurable tanto para la plataforma Linux como para la de Windows) que es simplemente una etiqueta de nombre de grupo que identifica a una determinada colección de ordenadores, y sus recursos sobre una red SBM.

Un ordenador Unix con Samba puede enmascararse como servidor en una red Microsoft y ofrecer los siguientes servicios:

- Compartir uno o más sistemas de archivos.
- Compartir impresoras, instaladas tanto en el servidor como en los clientes.
- Autentificar clientes logeándose contra un dominio Windows.
- Proporcionar o asistir con un servidor de resolución de nombres WINS, también conocido como el servidor de nombres NetBIOS sobre TCP/IP.
- Ahorro económico de licencia de servidor Windows NT para obtener las funcionalidades propias que Microsoft proporciona.
- Autenticar mediante el logon de entrada de Active Directory.
- Autentificar con su propio archivo de usuarios.

De las características anteriores, se implantará en este TFC (debido al escenario disponible) solo la funcionalidad de compartición de archivos e impresoras. Las opciones utilizadas y explicadas en esta parte, serán únicamente las implantadas en este TFC.

Una vez se tenga Samba instalado y con varios recursos compartidos, para los usuarios de Microsoft Windows, estos recursos aparecerán como carpetas normales de red. Los usuarios de Linux pueden montar en sus sistemas de archivos estas unidades de red como si fueran dispositivos locales, o utilizar la orden smbclient para conectarse a ellas. Cada directorio puede tener diferentes permisos de acceso sobrepuestos a las protecciones del sistema de archivos que se esté usando en Linux.

5.2. Instalación de Samba en el servidor y cliente Linux

El servidor de Samba se ha implantado en el mismo ordenador escogido para el servidor de OpenVPN (S.O. Linux Suse 11.0), de nombre sisif.upc y cuya IP pública es 147.83.12.65. Normalmente suele estar instalado en el S.O. por defecto, por lo que antes de instalarlo es interesante comprobarlo.

La instalación se realiza mediante la herramienta Yast. Una vez instalado, Samba permite ser configurado editando su fichero de configuración (*smb.conf*), o por interfaz gráfica (herramienta SWAT explicada en el apartado 2.2 del anexo) más fácil e intuitiva.

Una vez instalado Samba server y samba client, se comprueba que se han cargado un par de demonios (Daemon en inglés → Disk And Execution MONitor, que es un tipo especial de proceso informático que se ejecuta en segundo plano en vez de ser controlado directamente por el usuario), que proporcionan recursos compartidos a clientes SMB sobre la red:

- **Smbd**. Es el encargado de manejar los recursos compartidos entre el servidor Samba y sus clientes. Proporciona servicios de archivos, impresión y visualización a los clientes SMB a través de una o más redes. Controla todas las notificaciones entre el servidor Samba y los clientes de red. Asimismo, es responsable de la autentificación de usuarios, bloqueo de recursos, y la compartición de datos a través del protocolo SMB.
- **Nmbd.** Es un sencillo servidor de nombres que imita la funcionalidad de los servidores WINS y de resolución de nombres NetBIOS. En otras palabras permite conectarse a otros ordenadores de la red a través de su nombre y no de su IP (que es más difícil de recordar). Este demonio está a la escucha de peticiones para el servidor de nombres y proporciona la información apropiada cuando se le llama. La información de correspondencia entre nombre de ordenador y su IP se almacenan en el fichero lmhost (ubicado en la carpeta \etc\samba).

Se puede comprobar si se están ejecutando estos demonios mediante los comandos de terminal *netstat* y *ps* de Linux.

Nos centraremos en la configuración de Samba mediante la edición del archivo *smb.conf*.

El fichero smb.conf puede ser muy simple o extremadamente complejo debido a las múltiples opciones que presenta. Se comentará brevemente las distintas partes del fichero ya que se ha añadido una copia en el anexo B de la configuración del mismo, explicando cada una de las opciones. Se ha configurado Samba para que solo puedan conectarse aquellos ordenadores que estén dentro de un rango de IP's determinados (los asignados a la red de OPENVPN), aumentando así la seguridad.

5.3. Secciones.

5.3.1. Sección [globals]

Cualquier opción de esta sección se aplicará al resto de recursos, como si los contenidos de la sección se copiasen a todas las demás. Serán considerados como valores por defecto para las secciones donde no se especifique nada sobre ellos. A tener en cuenta que otras secciones pueden contener la misma opción pero con distinto valor. Lo último prevalece siempre sobre lo antiguo, así que ese último valor prevalecerá sobre el establecido en esta sección.

5.3.2. Sección [homes]

Permite a los usuarios remotos acceder a sus respectivos directorios principales en la máquina Linux local (cada uno al suyo nada más). Esto es, si un usuario de Windows intenta conectar a este recurso desde su máquina Windows, será conectado a su directorio personal. A tener en cuenta que para hacer esto, tiene que disponer de una cuenta en la máquina Linux.

5.3.3. Sección [printers]

La sección [printers] trabaja como la sección [homes] pero para el caso de las impresoras. Si existe una sección [printers] en el archivo de configuración, los usuarios podrán conectarse a cualquier impresora especificada en el archivo printcap del servidor.

A continuación se muestran algunos comandos útiles para controlar el servicio Samba:

- */etc/init.d/smb start.* Para iniciar el servicio samba.
- */etc/init.d/smb stop.* Para detener el servicio samba.
- */etc/init.d/smb restart.* Para reiniciar el servicio.
- */etc/init.d/smb status.* Para dar a conocer el estado en el que se encuentra el servicio.

Para estos y otros comandos es importante tener en cuenta que se deben ejecutar como usuario root (máximo usuario administrador de una máquina Linux).

- *smbclient //IPdelServidorSamba*

  Comando para conectar al servidor samba desde línea de comandos terminal. Sin embargo la forma más cómoda es hacerlo a través de un navegador de archivos escribiendo *smb://nombre_equipo o IP*.

Para conectar Samba mediante el nombre de máquinas, Tras tener el servicio samba configurado, es necesario editar el fichero lmhost de los equipos Linux, ubicado en */etc/samba/lmhost.* Este fichero permite resolver nombres a direcciones IP en el sistema, para poder conectarnos a los equipos con samba mediante su nombre sin tener que recordar su IP. En el anexo A se puede consultar este fichero.

5.4. Configuración de smb.conf

A continuación se explica algunas de las opciones más significativas que se han utilizado en el servidor samba. El contenido del fichero smb.conf, como ya se mencionó, se encuentra en el del anexo B. Para abrir este fichero se puede utilizar un editor de texto:

[globals]

- ***workgroup*** *= EPSC.* Esta línea asigna el grupo de trabajo. Los clientes que quieran acceder a sus recursos deberán pertenecer al mismo grupo de trabajo. Es importante indicar que debe ser el mismo nombre de grupo de trabajo que el de Windows. De lo contrario los equipos Windows no tendrán acceso a los recursos samba de los ordenadores Linux.
- ***server string*** *= Samba Server versión %v.* En esta línea se puede poner un mensaje de bienvenida para el Servidor Samba. Si se añade %v, aparece el número de versión de Samba.
- ***netbios name*** *= sisif.upc.* Nombre con el que aparece el servidor en la red.
- ***log file*** *= /var/log/samba/log.%m.* Ubicación donde serán guardados los logs.
- ***max log size*** *= 50*. Se define el tamaño máximo del log.
- ***printcap name*** *= cups*
  printing *= cups*
  disable spoolss *= yes*
  show and printer wizard *= no*

Estas líneas permiten compartir todas las impresoras del ordenador de forma remota junto con la sección printers que se explicará brevemente en este mismo apartado.

- ***security*** *= share*. Se configura la seguridad a nivel de directorio compartido.
- ***hosts allow*** *= 10.8.0.*
- ***interfaces*** *= 10.8.0.0/24*

Con las dos anteriores líneas es posible permitir el acceso solo de ciertas redes al servidor (el punto final es importante ya que indica que se trata de todo la red del rango 10.8.0).

A continuación se explican las líneas para compartir una carpeta en el ordenador sisif para intercambiar ficheros dentro de la VPN:

- ***[Dades].*** Este es el nombre con el que aparece la carpeta compartida.

- ***comment*** *= Directorio de Acceso general*. Comentario visible en la red de este recurso compartido.
- ***path*** *= /home/pbruna/dades*. Ruta de acceso al recurso. Esta carpeta debe existir por lo que es necesario crearla previamente.
- ***read only*** *= No*. Definir que el recurso no sea solo de lectura.
- ***guest ok*** *= yes.*
- ***browseable*** *= no.* Permitir que el recurso sea visible por todos.
- ***writable*** *= yes.* Permiso de escritura sobre el recurso.
- ***directory mask*** *= 0777.* Definir los permisos al crear subdirectorios.
- ***create mask*** *= 0777.* Permisos al crear archivos.

[printers]

- ***comment*** *= Printers*
- ***path*** *= /var/spool/samba*
- ***printable*** *= yes*
- ***create mode*** *= 0700*
- ***browseable*** *= Yes*
- ***guest ok*** *= yes*
- ***print ok*** *= yes*
- ***use client driver*** *= yes*
- ***public*** *= yes*

Comentar que es importante poner printable = yes en todos los recursos de impresora (y/o en [printers]), de forma que Samba pueda saber que se trata de impresoras. Si se olvida, estos recursos no podrán ser usados para imprimir y aparecerá como recursos de disco. El directorio /var/spool/samba se debe crear si no existe, con permisos 0700.

Por último se debe descomentar en el archivo /etc/cups/mime.convs la línea: application/octeto-string application/vnd.cups-raw 0 -. De este modo cups (Common Unix Printing System- Sistema de impresión común de Unix, es un sistema de impresión para S.O. de tipo Unix que permite que un computador actúe como servidor de impresión) permitirá compartir la impresora remotamente. Una vez realizados estos cambios se debe reiniciar el servicio cups y samba: service cups restart y service samba restart (para OpenSUSE).

5.5. Instalación de Samba en clientes Windows.

Hasta ahora se ha explicado SAMBA en el caso de Linux, pero también se explicará el caso para Windows. En este S.O. la compartición de recursos se realiza de una manera algo más manual, ya que no se dispone de ningún tipo de fichero de configuración que haga referencia a los recursos compartidos en el PC, y en sus opciones. En el anexo B se explica como compartir una carpeta.

6. VNC

Uno de los requisitos de este TFC es, una vez establecida la VPN, poder controlar remotamente los ordenadores conectados a esta. La elección ha sido VNC (Virtual Network Computing), basado en el protocolo RFB (remote framebuffer), protocolo de acceso remoto de interfaz grafica de usuario. VNC es un software libre basado en una arquitectura cliente-servidor que permite controlar de forma remota el ordenador que actúa como servidor desde el ordenador cliente.

En muchos casos se hace necesario poder controlar remotamente un PC, bien desde casa hacia la oficina (el administrador de la red que puede gestionar muchos ordenadores, evitándose muchos desplazamientos) o bien dentro de una misma corporación en la que los centros de trabajo no se encuentran próximos entre sí. De esta forma se consigue un aumento considerable de ahorro de tiempo y dinero.

Dentro del software VNC existen diferentes programas como pueden ser TightVNC, RealVNC, los cuales poseen diferentes opciones de cara al usuario. VNC consta de 2 partes fundamentalmente:

- La parte servidor que debe instalarse en el ordenador al que se desea conectar remotamente. Esta parte, consta de varias configuraciones como la contraseña de conexión, el puerto de conexión, etc.
- La parte cliente (viewer) que debe instalarse en el ordenador que se desea usar para controlar el escritorio remoto del servidor.

El software elegido ha sido:

- Para Linux (OPEN SUSE y Ubuntu) se ha escogido el vino (para servidor), Vinagre (para cliente versión gráfica) y TightVNC (parte cliente mediante comandos). Vienen integrados en el propio S.O. y son fáciles de utilizar.
- Para Windows se ha escogido tanto para servidor como para cliente Tight VNC, porque ofrece gran número de opciones a la hora de conectarse a un escritorio remoto (más que por ejemplo Real VNC).

Los motivos por los que se ha escogido el grupo de software anterior son los siguientes:

- Sencillez de instalación y uso ya que su configuración es simple.
- Se trata de software gratuito.
- Son multiplataforma, es decir, puede ser instalado en distintos S.O.
- Son softwares libres.
- Versatilidad entre los distintos softwares de VNC. Desde cualquier ordenador conectado a la VPN con un cliente VNC (ya sea RealVNC, TightVNC, etc.) puede conectarse a otro ordenador con un VNC instalado.

- TightVNC ha sido escogido para mantener una uniformidad tanto en Linux como en Windows. Debido a que Real VNC está muy extendido para Windows también se ha añadido la explicación del uso de este.
- TightVNC presenta las siguientes características:

    - **Transferencia de ficheros en Windows.** Es posible subir ficheros del ordenador local hacia el servidor VNC y viceversa.
    - **Escalado del escritorio remoto**. Es posible visualizar el ordenador remoto en una pantalla pequeña o realizar un zoom de esta para ver con más detalle la pantalla.
    - **Codificación eficiente de la compresión opcional JPEG.** Se ha mejorado el software para optimizar el tráfico de red en conexiones de bajo/medio ancho de banda. Esto es configurable pudiendo escoger el nivel de compresión más adecuado para cada conexión.
    - **Mejoras en el acceso Web.** TightVNC incluye una mejorada versión de Java viewer con pleno soporte para codec de modos de color 24 bits. Es posible acceder al applet de Java viewer vía http.
    - **Soporte para 2 contraseñas, control total y solo lectura.** El servidor permite configurar el acceso al ratón y al teclado remoto en función de la contraseña de autenticación.
    - **Tunelación SSH automática en Unix.** La versión de TightVNC Viewer para Unix puede crear túneles vía SSH, utilizando la instalación del cliente SSH/OpenSSH de manera local. Esta opción no ha sido implantada dado que ya se esta en una red segura creada con OPENVPN.

Para la instalación y configuración de VNC se puede consultar el anexo C.

7. SEGURIDAD EN ORDENADORES

Para aumentar la seguridad de la VPN es necesario limitar los accesos a los ordenadores que la componen con el firewall del S.O. Tal y como se muestra en la *Fig.*7.1, para poder usar los servicios de VNC y SAMBA de los ordenadores de la VPN se debe hacer desde un ordenador que esté conectado a esta. En caso contrario no será posible acceder desde un ordenador de la EPSC, que esté ajeno a la VPN.

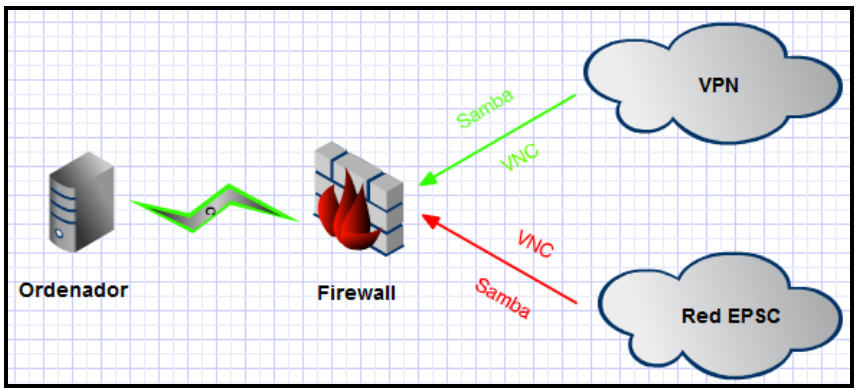

Fig.7.1 Conexión de la VPN con el firewall.

No obstante, es importante introducir una breve descripción de que es un firewall y sus principales funciones y características.

Básicamente, un firewall o cortafuegos en castellano, es un elemento que permite o deniega las transmisiones de una red a otra, actuando como filtro según las políticas de red que haya definido la organización responsable de esta. Por ejemplo, podría examinar si la comunicación es hacia dentro (trafico entrante) o hacia fuera (tráfico saliente), y dependiendo de su dirección podría permitirla o no. Normalmente suele situarse entre una red local (privada) y la red pública de Internet para evitar que los intrusos (usuarios malintencionados) puedan acceder a información confidencial de cualquier ordenador/servidor de la red privada. Aunque el usuario medio pueda creer que no hay necesidad de protegerse de cara a la conexión con Internet, el firewall se convierte en un elemento imprescindible si se está conectado permanentemente mediante ADSL. El firewall evitará la entrada de los programas que rastrean direcciones IP's.

Un firewall, puede ser un dispositivo software o hardware, o sea, un aparato electrónico que se conecta entre la red privada y la red pública (Internet), o bien un programa instalado en el ordenador que tiene el MODEM o router que conecta con Internet. En ambos casos, el funcionamiento se basa en el filtrado de paquetes (información).

7.1. Ventajas de un Firewall

A continuación se detallan:

- Permite al administrador de la red definir un "choke point" (embudo), manteniendo al margen los usuarios no-autorizados (usuarios malintencionados como hackers) fuera de la red. Permiten administrar los accesos posibles de Internet a la red privada y viceversa.
- El firewall ofrece un punto donde la seguridad puede ser monitorizada.
- Protección de información privada. Permite definir distintos niveles de acceso a la información, de manera que en una organización cada grupo de usuarios definido tenga acceso sólo a los servicios e información definidos.

7.2. Limitaciones de un Firewall

A continuación se detallan:

- No protege contra aquellos ataques cuyo tráfico no pase a través de él, ya que no puede proteger de las amenazas a las que está sometido por ataques internos a la red, o usuarios negligentes/malintencionados como espías corporativos que copien datos sensibles en cd's o memorias USB, y los extraigan del edificio.
- No protege de los fallos de seguridad de los servicios y protocolos de los cuales se permita el tráfico. Hay que configurar correctamente y cuidar la seguridad de los servicios que se publiquen en Internet.
- No protege de virus, es decir, no puede actuar como antivirus.
- No puede proteger contra los ataques de ingeniería social (arte de manipular personas para eludir los sistemas de seguridad). Esta técnica consiste en obtener información de los usuarios por teléfono, correo electrónico, correo tradicional, o contacto directo. Los atacantes de la ingeniería social usan la fuerza persuasiva y se aprovechan de la inocencia del usuario haciéndose pasar por un compañero de trabajo, un técnico o un administrador, etc.

7.3. Políticas de un Firewall

Es posible optar entre dos configuraciones opuestas en la política de seguridad de un firewall:

- ***Política restrictiva***. Se deniega todo el tráfico excepto el que está explícitamente permitido. El firewall obstruye todo el tráfico y hay que habilitar expresamente el tráfico de los servicios que se necesiten. Esta opción es la que se ha utilizado para el TFC.
- ***Política permisiva***. Se permite todo el tráfico excepto el que esté explícitamente denegado. Cada servicio potencialmente peligroso necesitará ser aislado uno a uno, mientras que el resto del tráfico no será filtrado.

La opción de política restrictiva, es la más segura, ya que es más difícil permitir por error algún tráfico potencialmente peligroso, mientras que en la política permisiva es posible que no se haya contemplado algún caso de tráfico peligroso y sea permitido por defecto.

7.4. Tipos de firewall

7.4.1. Nivel de aplicación de pasarela.

Aplica mecanismos de seguridad para aplicaciones específicas, tales como servidores FTP y Telnet.

7.4.2. Circuito a nivel de pasarela.

Aplica mecanismos de seguridad cuando una conexión TCP o UDP es establecida. Una vez que la conexión se ha hecho, los paquetes pueden fluir entre los anfitriones sin más control. Permite el establecimiento de una sesión que se origine desde una zona de mayor seguridad, hacia una zona de menor seguridad.

7.4.3. Firewall de capa de red o de filtrado de paquetes

Funciona a nivel de red (nivel 3) de la pila de protocolos (TCP/IP) como filtro de paquetes IP. A este nivel se pueden realizar filtros según los distintos campos de los paquetes IP: dirección IP origen, dirección IP destino, etc. A menudo en este tipo de firewall se permiten filtrados según campos de nivel de transporte (nivel 4), como el puerto orígen y destino, o a nivel de enlace de datos (nivel 2), como la dirección MAC. Este es uno de los principales tipos de firewall. Se considera bastante eficaz y transparente pero difícil de configurar.

7.4.4. Firewall de capa de aplicación

Trabaja en el nivel de aplicación (nivel 7), de manera que los filtrados se pueden adaptar a características propias de los protocolos de este nivel. Por ejemplo, si se trata de tráfico HTTP, se pueden realizar filtrados según la URL a la que se está intentando acceder.

Un firewall a nivel 7 de tráfico HTTP, suele denominarse proxy, y permite que los computadores de una organización entren a Internet de una forma controlada. Un proxy oculta de manera eficaz las verdaderas direcciones de red.

En este TFC se ha utilizado una combinación de firewall a nivel de aplicación de pasarela (ya que se ha permitido el acceso de los programas utilizados en el TFC) y de capa de red o filtrado de paquete porque se ha limitado el rango de IP’s para el acceso a dichos programas.

La configuración de los firewall de los ordenadores (para Windows y Linux) se encuentra explicada en el capítulo 4 del anexo.

7.5. Firewall en LINUX

Para la configuración del firewall de Linux se ha tenido que utilizar IPTables. Como viene siendo habitual, es posible configurarlo tanto a nivel de fichero como por interfaz gráfica. Sin embargo, antes de comenzar a explicar su configuración es importante explicar brevemente este concepto. Cabe recalcar que la configuración de IPTables, para conseguir una seguridad óptima, se necesitaría profundizar en él. Por este motivo y dado que en el TFC se ha utilizado únicamente para dar un grado de seguridad determinado (dando acceso a las aplicaciones anteriormente explicadas), la explicación de IPTables será breve y centrada en aquellos aspectos relacionados con los implantados en el escenario.

IPtables es un sistema de firewall vinculado al kernel de Linux que se ha extendido enormemente.

7.5.1. Características de IPTables

Es una herramienta de firewall que permite no solamente filtrar paquetes, sino también realizar traducción de direcciones de red (NAT) para IPv4 o mantener registros de log. Las funciones más importantes del mismo, aunque no las únicas, son:

- Administración de la memoria para todos los programas y procesos en ejecución.
- Administración del tiempo de procesador que los programas y procesos en ejecución utilizan.
- Es el encargado de que sea posible acceder a los periféricos/elementos del ordenador de una manera cómoda.

Un firewall de iptables no es como un servidor, que se inicia o se detiene, o que se pueda caer por un error de programación, ya que iptables está integrado con el kernel y es parte del S.O. Para ello se ejecuta el comando iptables, con el que se añaden, borran, o crean reglas. Por ello no es sino un simple script de shell (intérprete de comandos de Linux) en el que se van ejecutando las reglas de firewall. Es la herramienta de espacio de usuario mediante la cual el administrador puede definir políticas de filtrado del tráfico que circula por la red. Permite al administrador del sistema definir reglas acerca de qué hacer con los paquetes de red. Es un software disponible en prácticamente todas las distribuciones de Linux actuales.

Las reglas se agrupan en cadenas: cada cadena es una lista ordenada de reglas. Las cadenas se agrupan en tablas: cada tabla está asociada con un tipo diferente de procesamiento de paquetes.

El kernel, dependiendo de si el paquete es para la propia máquina o para otra, consulta las reglas de firewall y decide que hacer con el paquete según mande el firewall. Por tanto hay tres tipos de reglas en iptables que se describen brevemente a continuación:

- **mangle table** (Tabla de destrozo). Esta tabla es la responsable de ajustar las opciones de los paquetes como por ejemplo la calidad de servicio (QoS - Quality of Service). Son las tecnologías que garantizan la transmisión de cierta cantidad de datos en un tiempo dado (importante sobre todo para transmisión de video o voz).
- **nat table** (Tabla de traducción de direcciones de red). Esta tabla es la responsable de configurar las reglas de reescritura de direcciones de origen y destino, o de puertos de los paquetes.
- **filter table** (Tabla de filtros). Esta tabla es la responsable del filtrado (es decir, de bloquear o permitir que un paquete continúe su camino). Todos los paquetes pasan a través de la tabla de filtros. El administrador puede crear tantas como desee. Las cadenas, no son más que un listado de reglas, con las cuales se controla cada uno de los paquetes que pasan. Una regla consta de 2 partes, y no es más que una condición y una acción. Si se cumple la condición se ejecuta la acción. El kernel de Linux posee, predefinidas, 3 cadenas y cualquier paquete pasará por una de ellas:

    - **INPUT chain** (Cadena de ENTRADA). Todos los paquetes destinados a este sistema atraviesan esta cadena (y por esto se la llama algunas veces LOCAL_INPUT o ENTRADA_LOCAL).
    - **OUTPUT chain** (Cadena de SALIDA). Todos los paquetes creados por este sistema atraviesan esta cadena (a la que también se la conoce como LOCAL_OUTPUT o SALIDA_LOCAL).
    - **FORWARD chain** (Cadena de REDIRECCIÓN). Todos los paquetes que meramente pasan por este sistema para ser encaminados a su destino, recorren esta cadena.

8. Configuración del escenario real.

8.1. Ordenadores utilizados.

En este punto se explicará detalladamente cual es el escenario real sobre el que se ha montado la VPN y se ha desarrollado el TFC.

El escenario que se ha dejado totalmente configurado formando la VPN, y funcionando correctamente, consta de cuatro ordenadores:

- Uno con S.O. OpenSuse 11.0 (i586), y kernel 2.6.25.11-0.1. Tiene un procesador Intel(R) Pentium(R) 4 CPU 3.20GHz y 1GB de memoria RAM. Este PC es el más importante de todos ya que es el servidor de la VPN. Está situado en el despacho 106 del edificio CIMNE del campus.

- Uno con S.O. Ubuntu. Es un cliente de la VPN situado en el despacho 105 del edificio CIMNE del campus. Este ordenador se utiliza habitualmente como ordenador de despacho.

- Uno con S.O. Windows XP. Es un cliente de la VPN situado en el laboratorio 110 del edificio ESAB del campus. Este ordenador se ha utilizado como modelo para el control remoto de un equipo de laboratorio desde cualquier ordenador conectado a la VPN.

- Un portátil de nuestra propiedad, con S.O. Windows Vista. Es un cliente de la VPN que se ha usado desde casa para cubrir el caso roadwarrior.

Se han configurado estas cuatro máquinas para montar la VPN. El resto de la red no se ha modificado dado que uno de los requisitos para la realización del TFC era que se mantuviera toda la red como estaba, ya que existen VLAN's que no se pueden modificar. Tampoco se nos permitía el acceso a los switches y routers. El mantenimiento de la red como ya se comentó anteriormente, es llevado a cabo por la empresa UPCNet.

8.2. Configuración de red de los PC's.

La configuración de red de los PC's con los que se ha trabajado se muestra en la Tabla 8.1:

Tabla 8.1 Configuración de red del PC OpenSuse.

Configuración de Red	PC del despacho 106 del C3 – OpenSuse
IP	147.83.12.65
DNS	147.83.2.10 147.83.2.3
Default Gateway	147.83.12.1
Subnet mask	255.255.255.0

Tabla 8.2 Configuración de red del PC Ubuntu.

Configuración de Red	PC del despacho 105 del C3 – Ubuntu
IP	147.83.12.83
DNS	147.83.2.10 147.83.2.3
Default Gateway	147.83.12.1
Subnet mask	255.255.255.0

Tabla 8.3 Configuración de red del PC Windows XP.

Configuración de Red	PC lab.110 del ESAB–Windows XP
IP	147.83.2.23
DNS	147.83.2.10 147.83.2.3
Default Gateway	147.83.12.1
Subnet mask	255.255.255.0

Tabla 8.4 Configuración de red del PC Windows Vista.

Configuración de Red	PC portátil nuestro–Windows Vista
IP	Dinámica
DNS	87.216.1.65 87.216.1.66
Default Gateway	192.168.1.1
Subnet mask	255.255.255.0

La configuración de red de los tres primeros PC's no la hemos realizado nosotros, sino que se ha respetado las configuraciones que tenía cada máquina, sin modificarlas. La única máquina que se ha configurado por nosotros ha sido el PC portátil con Windows Vista.

8.3. Esquema de los PC's en la VPN

A continuación se muestra un esquema de cómo ha quedado el escenario una vez configurado *(Fig.8.1).*

Como se explicó en el capítulo 4.7.7.2, en nuestro escenario tenemos clientes que utilizan Windows, por tanto OpenVpn hará subnetting /30. Esto provoca que las IP's de los clientes no sean consecutivas.

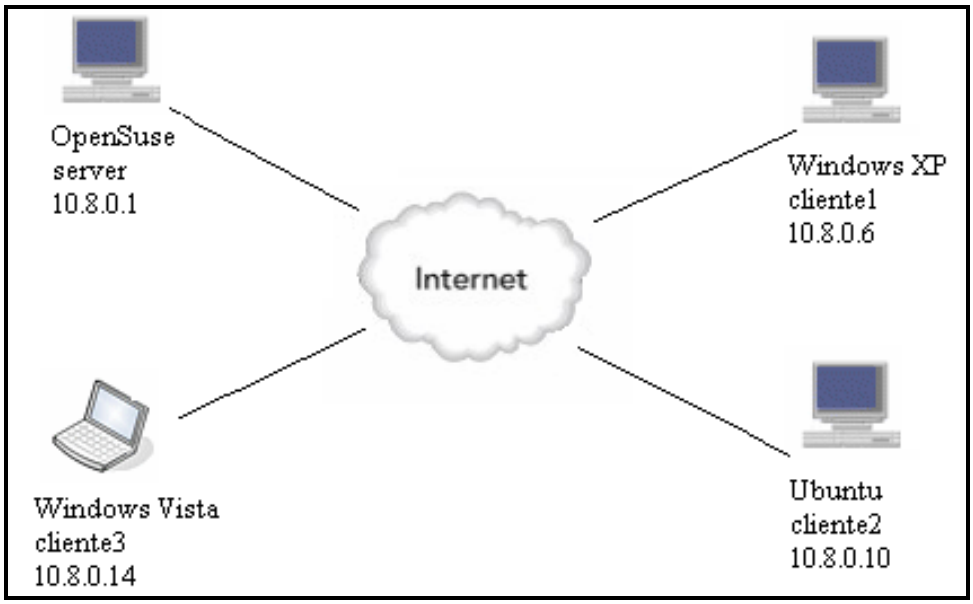

Fig.8.1 Esquema real de la VPN.

Las direcciones IP mostradas en la imagen, son de la interfaz virtual “tun0”.

8.4. Conexiones a la VPN desde los clientes

Con el escenario que se ha configurado, se han cubierto las posibles formas de conexión que pueda realizarse desde un cliente a la VPN.

Por un lado el cliente Ubuntu, del cual tenemos una conexión desde la misma LAN a la que pertenece el servidor, con una IP estática del mismo rango que este. Está situado en el mismo edificio del servidor y la conexión es de Linux a Linux.

Por otro lado el cliente Windows XP, que está conectado desde la misma LAN a la que pertenece el servidor, con una IP estática de diferente rango a la del servidor. Está situado en distinto edificio al servidor, y la conexión es entre Linux y Windows.

Finalmente el cliente de Windows Vista, con el que se ha realizado una conexión desde casa (fuera de la LAN a la que pertenece el servidor) con IP dinámica. Aprovechando que es un portátil, se han realizado distintas pruebas de conexión wifi, sin afectar ello a su configuración y funcionamiento. Se ha probado la conexión vía wifi desde la universidad, y una conexión wifi en casa. Para esta última prueba se ha habilitado en el router de casa la opción de conexión vía wireless.

Se ha comprobado el acceso entre clientes. Desde cualquier cliente, situado en la LAN o fuera de ella, podemos conectarnos a cualquier otro cliente conectado a la VPN.

8.5. Cambio de servidor

En caso de que exista la necesidad, o se requiera sustituir el servidor, hay que tener en cuenta que ha de ser una máquina con S.O OpenSuse. Se podría montar el servidor de la VPN con otro S.O., pero eso afectaría a la instalación y configuración de OpenVpn, y aunque no variaría demasiado su instalación y configuración respecto a este manual, sí que habría algunas partes que requerirían alguna modificación.

Hay que seguir los pasos de los puntos de instalación y configuración de OpenVpn, Samba y VNC, pero únicamente las partes referentes al servidor.

Dado que en nuestro servidor VPN hemos instalado nuestra CA, habría que volver a crear la CA en el nuevo servidor, o bien crearla en otra máquina. Esto depende de nuestra elección.

Es muy importante recuperar del servidor que se quiere sustituir los siguientes archivos:

ca.crt
ca.key
dh2048.pem
servidor.crt
servidor.key
ta.key
server.conf

Si se copian estos archivos desde el antiguo servidor al nuevo, el cambio de servidor no afectará a los clientes. En caso de no poder copiarlos, debido a que haya algún problema con el antiguo servidor, habría que volver a crear todos estos archivos, y los correspondientes a los clientes, que se deberían de sustituir, ya que sino habría una incoherencia entre los certificados y claves de clientes y servidor. Hay que recordar que el envío de estos archivos a los clientes, se ha de hacer por un canal seguro como puede ser SSH. Los archivos de los clientes que se deberían modificar en este caso son los siguientes:

ca.crt
clienteX.crt
clienteX.key
ta.key

Si los archivos anteriormente mencionados (los del servidor) no se instalan en la misma ruta, se debe modificar la ruta de los archivos en el server.conf.

Se asume que el nuevo servidor tiene una configuración de red igual al anterior. En caso contrario habría que entrar en el client.conf de los clientes y modificar la línea “remote IP”, para adaptarlo a la nueva dirección IP.

Finalmente se deberá configurar el Firewall, para permitir el acceso a la VPN y garantizar la seguridad. Hay que habilitar los puertos de OpenVPN, Samba, y VNC.

Si queremos que funcione la traducción de nombres de máquinas con su dirección IP, habrá que modificar los archivos lmhost, necesario para poder usar Samba mediante nombres, y el hosts, para el funcionamiento de la red en otros programas o conexiones (por ejemplo: VNC).

8.6. Añadir nuevos clientes.

Para añadir un nuevo cliente a la VPN, en primer lugar se ha de hacer la instalación y configuración de OpenVpn.

En caso de ser un cliente Linux, se instalará Samba.

Posteriormente se instalará TightVNC, RealVNC, o cualquier otra herramienta de VNC para el control remoto.

Al añadir un nuevo cliente a la VPN, la CA deberá crear los nuevos archivos de certificados y claves para el nuevo cliente (clienteX.crt y clienteX.key), que deberán ser enviados por un canal seguro al cliente, junto a los archivos ca.crt y ta.key.

Asimismo el cliente deberá configurar el client.conf o client.ovpn.

Una vez instalado todo el software necesario, deberemos configurar el firewall para poder acceder a la VPN, con un mayor nivel de seguridad. Hay que habilitar los puertos de OpenVpn, Samba, y VNC.

Si queremos que funcione la traducción de nombres de máquinas con su dirección IP, habrá que modificar los archivos lmhost (en caso de ser un cliente Linux), necesario para poder usar Samba mediante nombres, y el hosts, para el funcionamiento de la red en otros programas o conexiones (por ejemplo: VNC).

9. Futuras mejoras

9.1. Eliminar IP's públicas

Un punto débil que tiene la red de la universidad, aunque tengan ciertos niveles de seguridad, como pueden ser los firewall de cada PC, es el uso de IP's públicas. Es cierto que el firewall protege al PC, pero una IP pública es accesible desde Internet, y por tanto un hacker con experiencia, podría saltarse nuestro firewall y acceder a la máquina. Es algo que la EPSC debería de plantearse si algún día quiere aumentar su seguridad, sustituyéndolas por IP's privadas, y así evitar este posible problema.

9.2. Autenticación en OpenVpn mediante login y password

Otro punto a mejorar dentro de OpenVpn es la autenticación de clientes mediante login y password. Para ello, previamente se tendría que crear un usuario en el servidor OpenVPN, por ejemplo, si se crea el usuario "VPN" y con password "epscvpn", cuando el cliente OpenVPN intente conectarse a la VPN enviará el login "VPN" y el password "epscvpn".

Para este tipo de autenticación se tendrá que añadir al fichero de configuración del cliente la directiva "--auth-user-pass". Esto indica que para realizar la conexión será necesario su login y password. El login y password del cliente se pasa al servidor sobre el canal seguro proporcionado por TLS. Posteriormente se configurará el servidor para usar autenticación con login y password, que llamará a un plugin (como un por ejemplo un script), un fichero objeto compartido o un plugin DLL, que le pasará el login y el password enviados por el cliente.

10. Resultados y conclusiones

El motivo del TFC tuvo su origen para cubrir las necesidades planteadas por el grupo de investigación del departamento de Física Aplicada de la UPC. Se planteó realizar una VPN que garantizara un mayor nivel de seguridad al ofrecido por la EPSC, y permitiera la transferencia de archivos, la comunicación entre máquinas (de la EPSC y PC's personales de los profesores), y el control remoto de equipos.

Al iniciar el proyecto, se ha explicado en que consiste una VPN, así como un estudio de la situación actual de la EPSC, para así valorar las posibles soluciones al crear la VPN.

Debido a los problemas de acceso a los componentes hardware de la red, se optó por utilizar una solución software. Se decidió OpenVPN basado en el protocolo SSL/TLS, por las grandes ventajas que ofrece, y por ser de software libre y gratuito, con lo cual no hemos tenido que afrontar ningún tipo de gasto económico.

Posteriormente se usaron programas para cumplir con los objetivos del TFC, como han sido Samba para la transferencia de ficheros, y TightVNC (entre otras opciones explicadas en la memoria) para el control remoto de PC's.

Una vez montado el escenario real, con el servidor y los clientes, se comprueba que existe conexión entre los dos extremos de los túneles (entre el servidor y cualquier cliente de la VPN, y entre clientes).

Realizar este TFC nos ha permitido por un lado poner en práctica conocimientos adquiridos durante la carrera, y por otro un amplio aprendizaje autodidáctico en varios temas, que a continuación se explican en detalle:

- Conceptos teóricos de la carrera aplicados:
    - **Criptografía**. En la carrera se impartió, en la asignatura Sistemas y Aplicaciones, los conceptos de firma digital, encriptación Diffie Hellman, certificados, y funciones de hash. Esto se ha implantado para aumentar el nivel de seguridad en la VPN.
    - **Subnetting**. Aunque en la asignatura Laboratorio de Telemática ya se vió este concepto, aplicarlo a un escenario real, nos ha ayudado a reforzarlo y ver su utilidad en el trabajo cotidiano.
    - **Firewall**. En la asignatura de Sistemas y Aplicaciones se nos explicó las características de este elemento. En el caso práctico hemos comprobado que existe una gran variedad de tipos de firewall, así como opciones en su configuración.
    - **VPN**. Hasta el momento solo conocíamos el concepto de este término adquirido en la asignatura de Arquitecturas Telemáticas. Sin embargo no lo habíamos llegado a conocer de una manera práctica. Sin embargo ahora tenemos la capacidad de implantar una VPN y, por consiguiente, hemos podido comprobar la utilidad de esta en un caso real como es el de una Universidad pública.

- **Configuración de redes**. En la asignatura de Laboratorio de Redes pudimos realizar varios escenarios prácticos de configuración de redes con rangos de IP's diferentes, etc. Esto nos ha permitido comprender a fondo el escenario sobre el que debíamos trabajar y el escenario a desarrollar con la VPN.
- **DNS.** Este concepto lo conocíamos anteriormente a la carrera, aunque estudiarlo de nuevo nos ha ayudado a comprenderlo mejor. Nos los enseñaron en la asignatura Fundamentos de sistemas distribuidos, y lo hemos aplicado para la traducción de IP's de los ordenadores a sus nombres correspondientes.

Todos estos conceptos han sido implantados en el escenario del TFC. De esta manera hemos conseguido implementar de forma práctica los conceptos teóricos, y esto nos ha ayudado a comprenderlos mejor.

- Nuevos conocimientos adquiridos:

  - **OpenVPN, VNC y Samba**. Estos programas eran totalmente desconocidos por nosotros antes de realizar este TFC. Actualmente el estudio y la implantación de cada uno de ellos nos ha permitido conocerlos en profundidad.
  - **Linux**. Anteriormente al TFC, teníamos nociones muy básicas de este tipo de S.O. Por motivos ajenos a este TFC, tanto a nivel personal como profesional no estábamos habituados a trabajar con Linux. Esto, nos provocó tener que invertir un gran número de horas en el proceso de aprendizaje ya que las diferencias respecto a Windows, son considerables. Actualmente podemos desenvolvernos con soltura en dos de sus distribuciones (OpenSuse y Ubuntu). Con el resto de distribuciones, aunque no hemos trabajado con ellas, nos resultaría más fácil aprender a manejarlas que antes.
  - **Red del Campus**. Hasta comenzar con el TFC, desconocíamos la arquitectura de red de la EPSC. Actualmente tenemos conocimientos generales de cómo está configurada tanto a nivel físico como lógico.
  - **Búsqueda de información**. El hecho de desconocer las herramientas software utilizadas y no contar con apoyo de ningún experto en el tema, nos ha obligado a realizar búsquedas por Internet (foros, etc.) que nos permitiera conseguir la información necesaria para nuestro escenario. Es importante para nosotros recalcar el especial esfuerzo que hemos realizado para conseguirla, ya que no siempre es fácil encontrar información técnica tan específica. Adicionalmente, en la parte práctica siempre surgen problemas que dificultan el correcto funcionamiento, y provocan tener que invertir tiempo adicional en su solución.

Como comentario final indicar que para nosotros ha sido muy motivador y satisfactorio la totalidad del desarrollo de este TFC, tanto la fase previa de estudio como su posterior implantación. Es alentador para cualquier ingeniero comprobar que su proyecto tiene un fin práctico y útil.

Escola Politècnica Superior
de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

ANEXOS

TÍTULO DEL TFC: Implementación de una red privada virtual para el control remoto de equipos de laboratorio.

TITULACIÓN: Ingeniería Técnica de Telecomunicaciones, especialidad en Telemática.

AUTORES: Rafael Pinilla Vico
Óscar Sánchez Sánchez

DIRECTOR: Pere Bruna

FECHA: 22 de diciembre de 2009

A. Archivos de traducción de nombres.

A.1. Lmhost.

Podría decirse que este fichero es el equivalente al fichero /etc/hosts (estándar de Linux-Unix para la resolución de nombres e IP's), y su estructura es idéntica a la que se muestra a continuación:

 # This file provides the same function that the lmhosts file does for
 # Windows. It's another way to map netbios names to ip addresses.
#
 # See section 'name resolve order' in the manual page of smb.conf for
 # more information.

127.0.0.1 localhost
127.0.0.2 sisif.upc
10.8.0.14 Paula
10.8.0.10 massaguer
10.8.0.1 sisif
10.8.0.6 young

Tan solo hay que añadir en una línea la IP del ordenador y a continuación el nombre correspondiente.

De esta manera se puede pasar de escribir ej: *smb**10.8.0.14* a escribir *smb**Paula*.

Es interesante comentar que para el caso del S.O. OpenSuse, si se utiliza el explorador de archivos konqueror, este no permite copiar y pegar archivos. Por tanto se recomienda utilizar otro explorador conocido como Nautilus que, sin embargo, necesita realizar una operación para añadir una barra exploradora desde el menú superior de la ventana, seleccionando GO – location.

A.2. Hosts.

Por otro lado, la instalación de la parte cliente para Windows no es necesaria ya que este protocolo viene por defecto incluido en estos S.O. De igual manera que sucede en Linux, en Windows existe un fichero encargado de resolver IP's a nombres de equipo. Este fichero se encuentra en c:\Windows\system32\ drivers\etc y su nombre es hosts. El contenido del mismo se muestra a continuación:

 # Copyright (c) 1993-2006 Microsoft Corp.
#
 # This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
 # This file contains the mappings of IP addresses to host names. Each
 # entry should be kept on an individual line. The IP address should

```
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

127.0.0.1       localhost
::1             localhost
10.8.0.1        sisif
10.8.0.10       massaguer
10.8.0.14       paula
10.8.0.6        young
```

Para cualquier nuevo ordenador que se desee añadir en el futuro a la VPN de la EPSC, es necesario incluirlo en estos ficheros (tanto en el lmhost de Linux como el hosts de Windows), por lo que es necesario un cierto mantenimiento con el objetivo de tenerlo actualizado.

Para conectarnos a cualquier equipo de red, configurado para samba, contra Linux o Windows, bastará con escribir en el explorador de Windows *nombre_maquina* y se puede acceder a sus recursos compartidos.

B. SAMBA.

B.1. Configuración de smb.conf

```
[global]
        workgroup = EPSC
        netbios name = sisif.upc
        server string = Samba Server %v
        log file = /var/log/samba/log.%m
        printcap name = cups
        printing = cups
        printcap cache time = 750
        cups options = raw
        map to guest = Bad User
        include = /etc/samba/dhcp.conf
        logon path = \\%L\profiles\.msprofile
        logon home = \\%L\%U\.9xprofile
        logon drive = P:
        usershare allow guests = Yes
        security = share
        hosts allow = 10.8.0.
        interfaces = 10.8.0.0/24

[Dades]
        comment = Directorio de Acceso general
        path = /home/pbruna/dades
        read only = No
        guest ok = yes
        browseable = no
        writable = yes
        directory mask = 0777
        create mask = 0777
        public = yes

[printers]
        comment = All Printers
        path = /var/tmp
        printable = Yes
        create mask = 0600
        browseable = No

[print$]
        comment = Printer Drivers
        path = /var/lib/samba/drivers
        write list = @ntadmin root
        force group = ntadmin
        create mask = 0664
        directory mask = 0775
```

B.2. SWAP.

La configuración de Samba mediante la interfaz gráfica se puede realizar escribiendo en un navegador de archivos de Linux la siguiente dirección: *http://loscalhost:901.* Se debe introducir el usuario y la contraseña de root para acceder a esta consola de configuración. De esta manera se accede a la pantalla que se muestra a continuación, *Fig.B.1*:

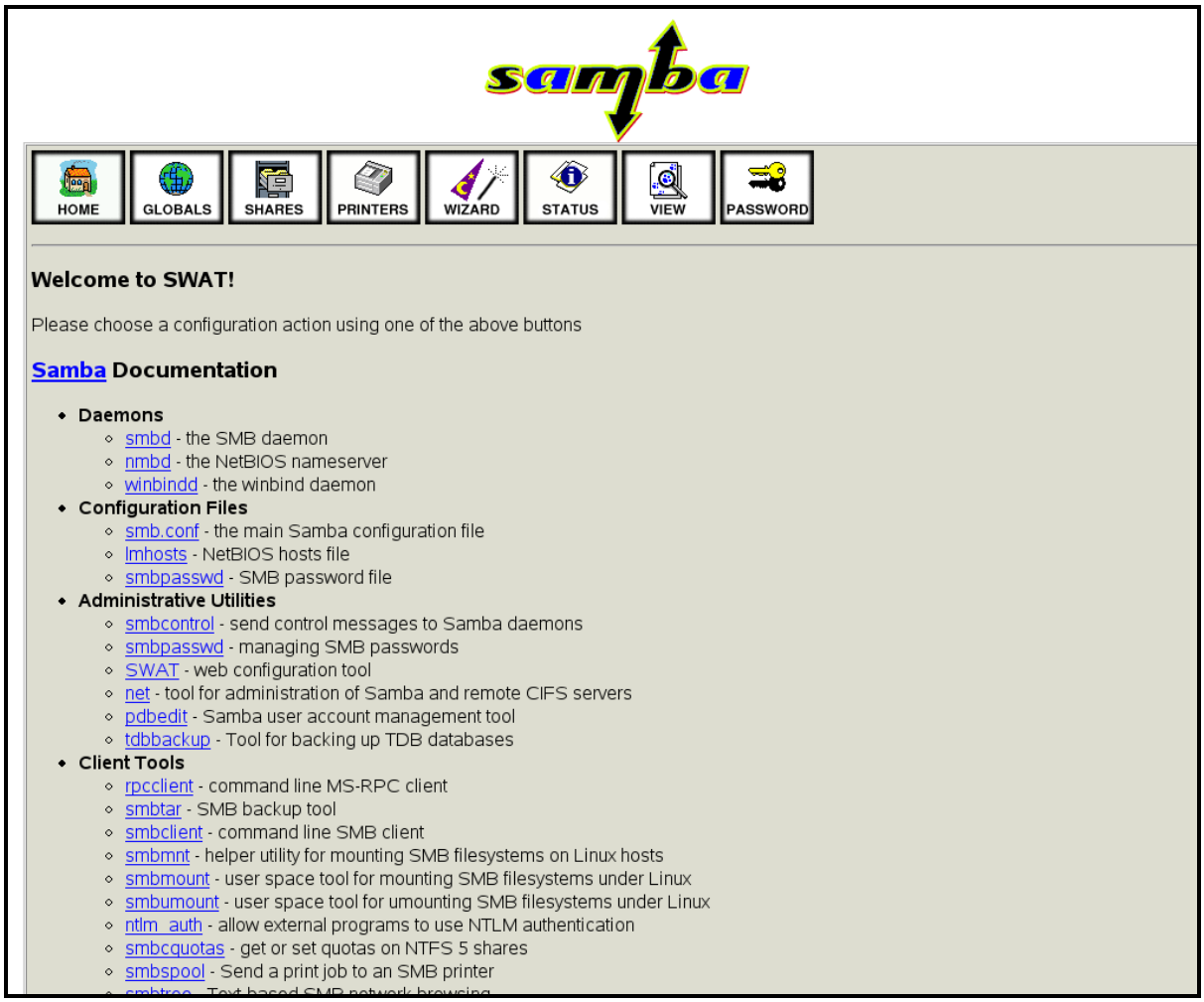

Fig.B.1 SWAP de Samba

Para configurar Samba desde esta interfaz se debe ir pulsando por los botones que aparecen en la cabecera: Home, globals, Shahres, Printers, etc. Cada una de estas partes accede a su sección correspondiente dentro del fichero smb.conf que anteriormente se ha explicado. No se detalla el uso a fondo de esta herramienta porque contiene las mismas opciones que hay en el fichero de configuración de samba, del capítulo 2.1 del anexo.

B.3. Compartir carpetas en Windows.

Pasos a seguir para definir permisos en las carpetas de Windows:

1. Crear la carpeta a compartir. Se pulsa el botón derecho sobre esta y se va a propiedades *(Fig.B.2).*

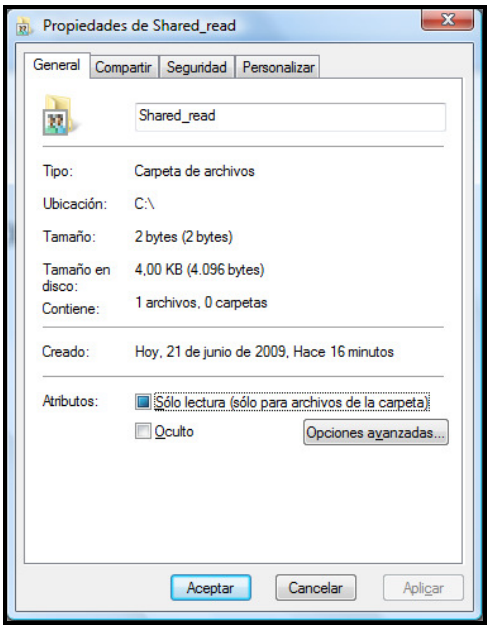

Fig.B.2 Propiedades de carpetas en Windows.

En esta pestaña es posible marcar la opción de “Sólo lectura” para no permitir que este fichero sea modificado por nadie.

2. En la pestaña Compartir *(Fig.B.3)* se puede definir los permisos a nivel de fichero (no confundir con los de usuario) de la carpeta. Para ello se debe pulsar en Permisos y nos aparece la *Fig.B.4*. Aquí se añade al usuario local de Windows con los permisos deseados.

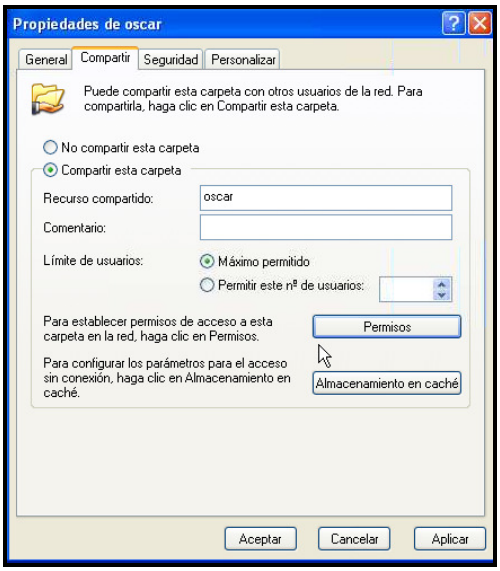

Fig.B.3 Compartir carpetas en Windows.

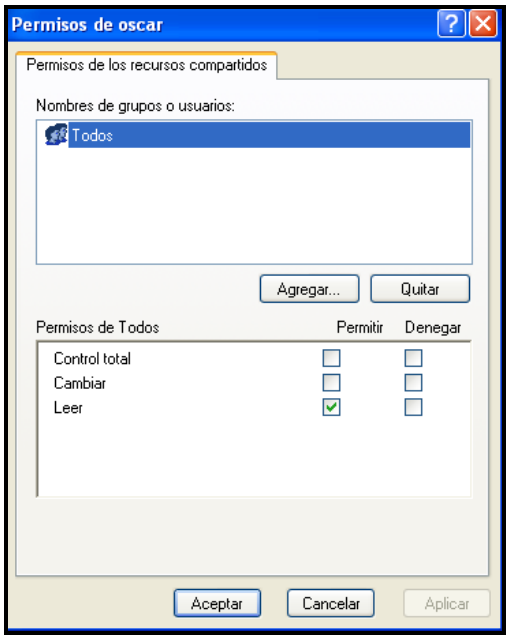

Fig.B.4 Permisos carpetas en Windows.

3. En la pestaña Seguridad se define los permisos de acceso a la carpeta a nivel de usuario *(Fig.B.5)*. Para editar se debe pulsar en agregar a un usuario o quitarlo de la lista. Los permisos a configurar se encuentran en la parte inferior de esta pestaña. Recordar que este usuario será el que se usa cuando se conecta desde un equipo con Linux mediante SAMBA.

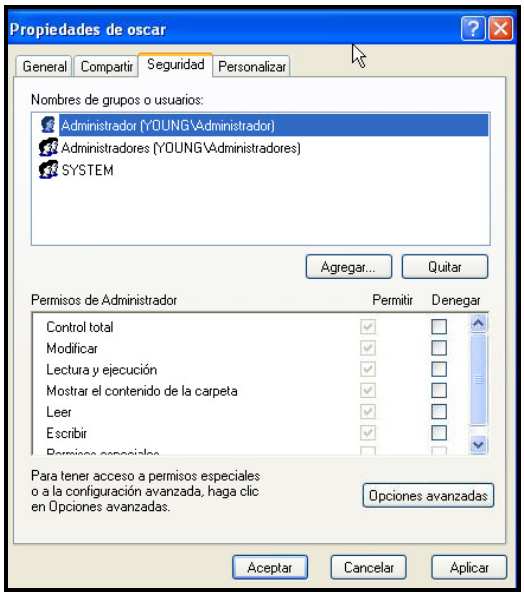

Fig.B.5 Permisos de usuario de carpeta en Windows.

Para el caso de Windows Vista el proceso sería muy similar.

B.4. Compartir carpetas en Samba.

En Linux, existen tres niveles de acceso a los archivos y carpetas para tres categorías diferentes de usuarios.

Categorías de usuarios o identidades:

1. Owner (propietario): La persona que el sistema reconoce como dueño de la carpeta o archivo.
2. Group (grupo): Grupo de usuarios con permisos similares.
3. Other (otros): Cualquier otra persona.

Niveles de acceso para archivos:

1. Read (lectura): Permiso para ver el archivo, sin hacer cambios.
2. Write (escritura): Permiso de escritura: puede escribir el archivo, y por tanto, cambiarlo.
3. Execute (ejecución): El archivo o directorio puede ser ejecutado.

Niveles de acceso para directorios (carpetas):

1. Read: Permiso para listar los archivos de un directorio.
2. Write: Permiso para añadir nuevos archivos al directorio.
3. Execute: Permiso para acceder a los archivos del directorio.

Como resultado de la combinación de los tres tipos de permisos (lectura, escritura y ejecución), con las tres clases de usuarios (dueño, grupo y otros), se

obtiene 23 = 8 permisos (en binario) que pueden ser asignados o denegados de forma independiente para cada identidad (el 1 da el permiso y el 0 lo quita), Tabla B.1.

Tabla B.1 Permisos de carpetas.

	Lectura	Escritura	Ejecucion	Total
Ningun permiso	0	0	0	0
Ejecución	0	0	1	1
Escritura	0	1	0	2
Escritura + Ejecución	0	1	1	3
Lectura	1	0	0	4
Lectura + Ejecución	1	0	1	5
Lectura + Escritura	1	1	0	6
Lectura + Escritura + Ejecución	1	1	1	7

De la tabla anterior se extrae que el grupo de 3 bits corresponden a los valores 4, 2 y 1. Así pues:

0 0 0 = 0 → (0+0+0)
0 0 1 = 1 → (0+0+1)
0 1 0 = 2 → (0+2+0)
0 1 1 = 3 → (0+2+1)
1 0 0 = 4 → (1+0+0)
1 0 1 = 5 → (4+0+1)
1 1 0 = 6 → (4+2+1)
1 1 1 = 7 → (4+2+1)

Por tanto, por cada identidad, podemos obtener un número comprendido entre 0 y 7, que delimitará sus privilegios. La asignación de permisos debe ser hecha como super usuario en una sesión de terminal, y desde la ruta de carpetas en la que se encuentre el archivo o la carpeta que se deseen cambiar. El comando para realizar cambios en los permisos es "chmod", y es posible obtener una ayuda desde consola escribiendo *chmod –help* para visualizar todas las opciones disponibles.

Por ejemplo, si desde el directorio que contiene el archivo prueba.txt se escribe chmod 777 prueba.txt, se le asignarían todos los permisos a este.

A continuación se tiene su equivalente en letras (que utilizaría en total nueve dígitos en lugar de tres):

0 = - - - = sin acceso
1 = - - x = ejecución
2 = - w - = escritura
3 = - w x = escritura y ejecución
4 = r - - = lectura
5 = r - x = lectura y ejecución

6 = r w - = lectura y escritura
7 = r w x = lectura, escritura y ejecución

Es importante resaltar que asignar permisos a un directorio (carpeta) no significa que los archivos o subcarpetas que pertenecen a esta hereden automáticamente los mismos permisos.

C. Instalación y configuración de VNC.

A continuación se explican los pasos para la instalación y configuración de VNC en los diferentes S.O.

C.1. En OPENSUSE y UBUNTU.

Se debe instalar (en cada S.O. mediante su herramienta correspondiente) Vinagre, Vino y TightVNC, escogiéndolo del listado de software disponible para descargar desde los repositorios de Linux *(Fig.C.1 y Fig.C.2)*. Hay que tener en cuenta que Linux descarga no solo el software, sino también una serie de librerías necesarias para la instalación del software.

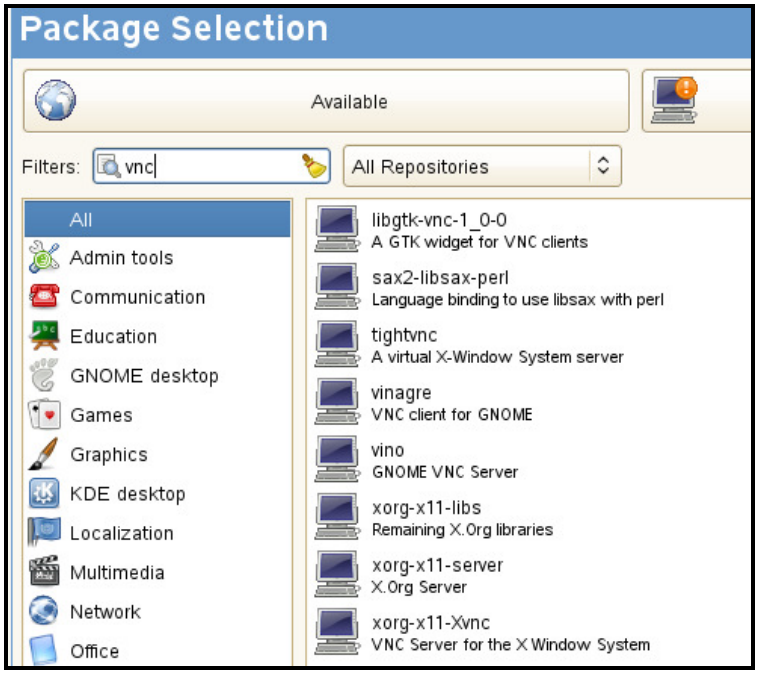

Fig.C.1 Repositorios de VNC en OpenSuse.

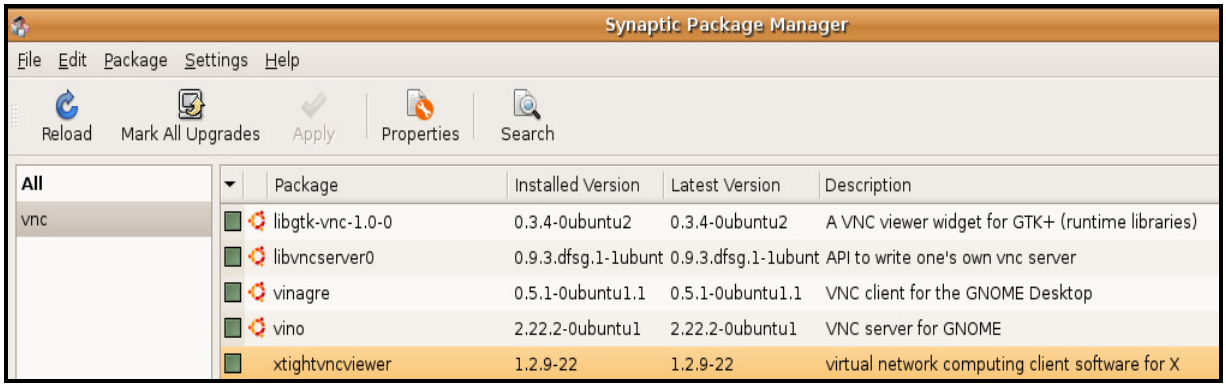

Fig.C.2 Repositorios de VNC en Ubuntu.

En ambos casos, para configurar la parte del servidor se debe acceder a través del icono Remote Desktop del S.O. Las opciones que aparecen son las que se muestran en las *Fig.C.3 y Fig.C.4*. De las opciones a configurar, se debe destacar únicamente la contraseña de conexión (que se solicitará cuando se conecte remotamente), y dejar el resto tal y como aparecen por defecto tras su instalación.

Fig.C.3 Configuración de VNC.

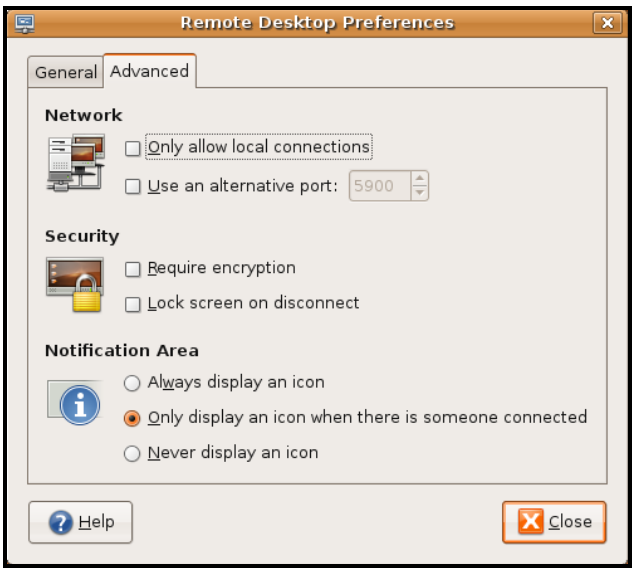

Fig.C.4 Configuración de VNC.

Para ejecutar la parte cliente (viewer), tal y como se ha comentado anteriormente, existen 2 maneras: vía gráfica o vía comando. Gráficamente se debe acceder a través de vinagre ejecutando el icono de Remote Desktop Viewer ubicado en Computer – More applications *(Fig.C.5)*.

Fig.C.5 Icono del vinagre.

En la *Fig.C.6,* se muestra la ventana que aparece una vez ejecutada la herramienta vinagre, donde hay que introducir el nombre del equipo con el que se desea realizar la conexión.

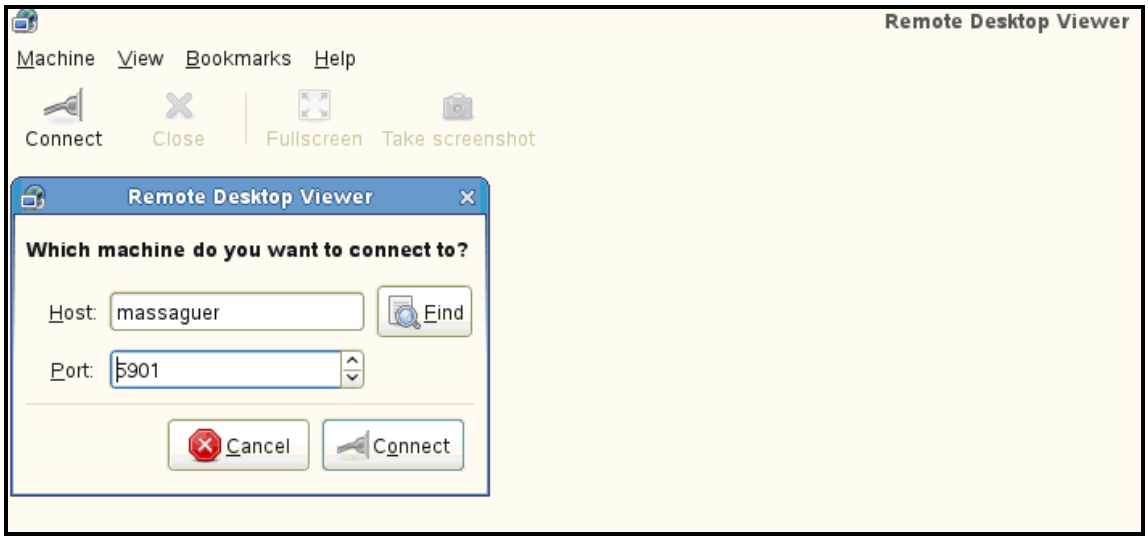

Fig.C.6 Vinagre

Para la ejecución por comando hay que abrir una sesión de terminal y escribir vncviewer. Aparecerá una ventana en la que se pedirá la IP/nombre del servidor a conectarse, y seguidamente la contraseña configurada en este *(Fig.C.7 y Fig.C.8).*

Fig.C.7 Solicitud de usuario en vncviewer.

Fig.C.8 Solicitud de password en vncviewer.

C.2. En WINDOWS.

C.2.1 Instalación de RealVNC.

Para la instalación de RealVNC, es necesario descargar RealVNC desde su Web oficial: *http://www.realvnc.com/.* La versión instalada es la 4.1.3. Una vez instalada siguiendo el asistente de instalación clásico de Windows se explica como debe configurarse para el escenario de este TFC.

A continuación se muestran algunas de las pantallas de instalación de RealVNC en Windows *(Fig.C.9 y Fig.C.10).*

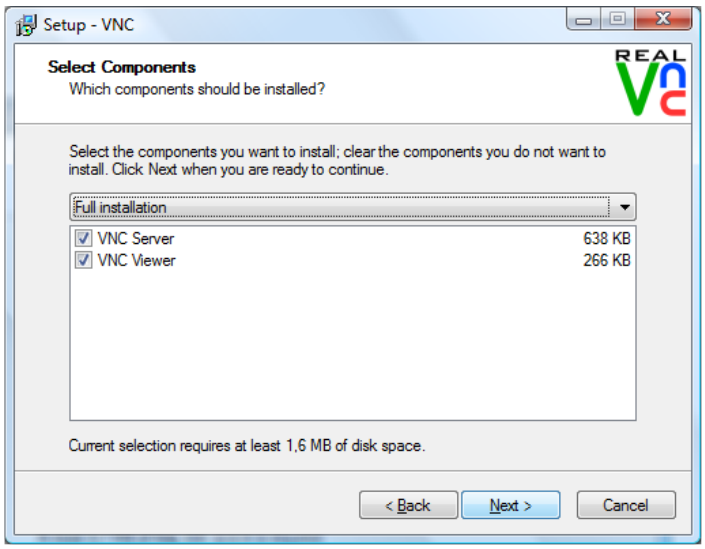

Fig.C.9 Instalación de RealVNC en Windows.

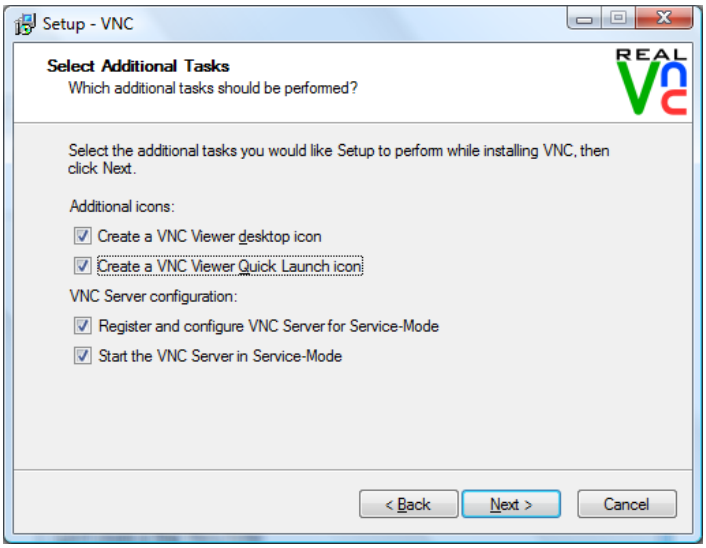

Fig.C.10 Instalación de RealVNC en Windows.

Para que el servicio de VNC se arranque de manera automática al inicio del S.O, se debe registrar a través de Inicio – Programas – Real VNC – VNC 4 (Service Mode) – Register VNC Service. Una vez registrado, hay que asegurarse que en el gestor de servicios, está configurado en modo automático, tal y como se muestra en la *Fig.C.11.*

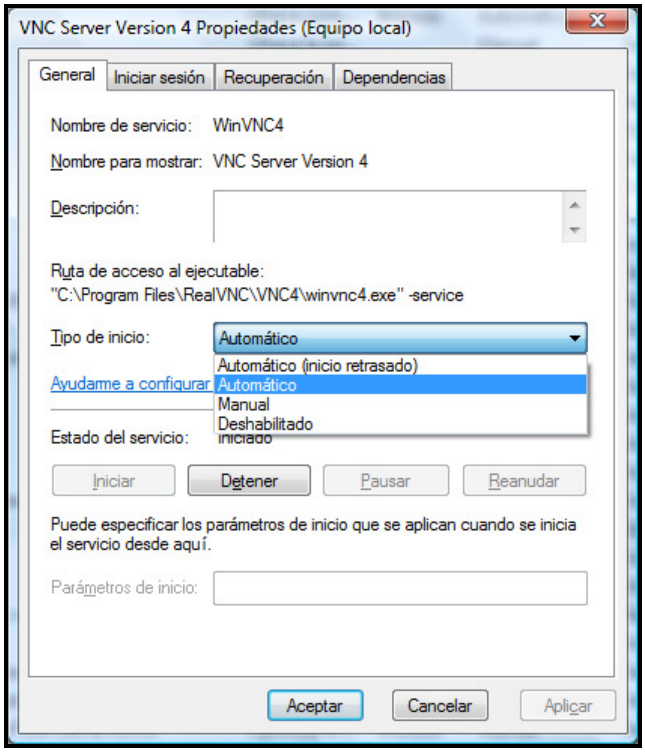

Fig.C.11 Arranque automático de VNC.

C.2.2 Configuración de RealVNC en el servidor.

Para configurar el servidor se ha de ir a Inicio - Programas - RealVNC - VNC Server 4 Service Mode, donde se encuentran las siguientes cinco opciones:

- *Register y Unregister*: permite añadir o eliminar RealVNC de la lista de servicios de Windows, como se explicó anteriormente.
- *Start y Stop***:** permiten arrancar o detener el servicio, aunque también se puede hacer desde Herramientas Administrativas del propio Windows.
- *Configure*: permite modificar las opciones de VNC.

Se comentaran pestaña a pestaña las diferentes opciones configuradas para el escenario. El resto de opciones han sido dejadas tal y como están por defecto, aunque se explican las más significativas:

C.2.2.1 Pestaña "Authentication"

Se procederá a configurar la contraseña de acceso. Para ello hay que pulsar el botón *Configure (Fig.C.12),* y escribir dos veces la contraseña con la que se accederá desde el exterior. Para aumentar la seguridad, se puede seguir las normas recomendadas para contraseñas como: un mínimo de ocho caracteres, que combine mayúsculas, minúsculas, caracteres y números etc., como por ejemplo:

Contraseña adecuada: Ab&78$!_BS30
Contraseña inadecuada: Barcelona

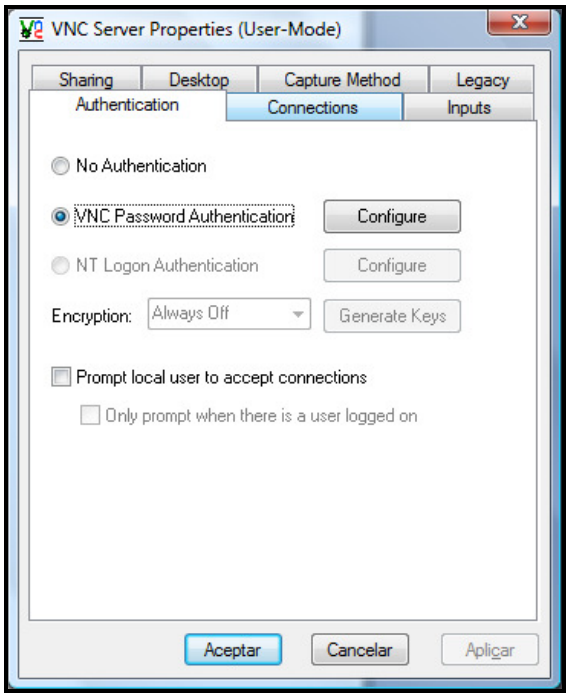

Fig.C.12 Password en VNC.

No obstante la parte de seguridad está explicada en el anexo D donde se aplicarán filtros de acceso en el Firewall para que solo puedan acceder ordenadores pertenecientes a la VPN. Es importante desmarcar la casilla "Prompt local user to accept connections", ya que sino cada conexión externa debe ser autorizada por alguien que esté sentado delante del ordenador. Esto permite poderse conectar de una forma "directa".

C.2.2.2 Pestaña "Connections"

En esta parte se definirán el puerto y el rango de IP's (correspondiente a la VPN) al que se dará acceso de conexión para este ordenador. De esta manera sólo los ordenadores conectados a la VPN podrán conectarse mediante este al ordenador. Aunque ya está configurado así en el firewall del S.O, el software lo permite para aquellos casos en los que se utilice un firewall.

Se borra la regla que se crea por defecto en la instalación, seleccionándola del listado y pulsando el botón Remove. Esta regla inicial autoriza el acceso a todo el mundo (que sepa la contraseña). Tal y como se muestra en las *Fig.C.13 y Fig.C.14* se ha de añadir el rango de IP's que se deseen, que en nuestro caso es el 10.8.0.0/24 (o máscara 255.255.255.0), tal y como muestra la figura *Fig.C.13*. El puerto debe dejarse por defecto el 5900 (TCP). Para volver a dejar paso a todo el mundo, como estaba al principio, se borrarán todas las reglas existentes y se añadirá la siguiente: 0.0.0.0 Allow.

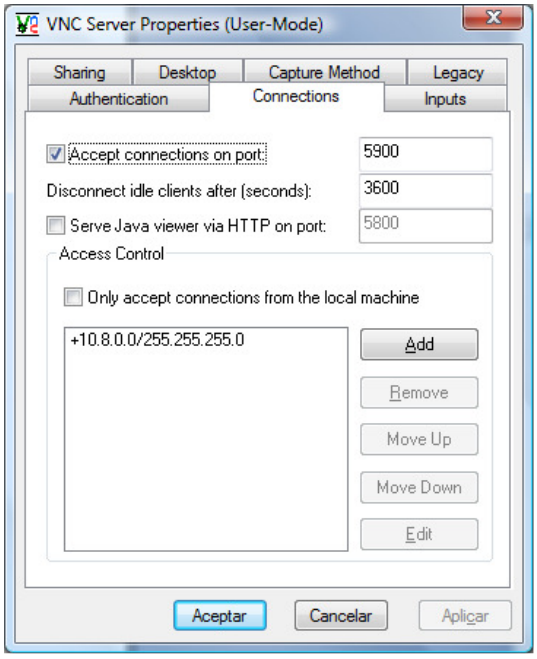

Fig.C.13 Rango de IP's permitido en VNC.

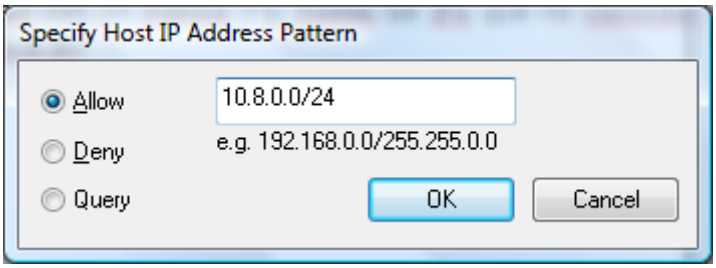

Fig.C.14 Rango de IP's permitido en VNC.

Las otras opciones son Deny (rechazar) y Query (preguntar primero). Esta última hace que cada vez que se produzca una conexión, se pida primero una autorización. Las IP's en modo Query se añaden con un símbolo de interrogación (?) al listado, mientras que las Deny lo hacen con un menos (-). Todas las IP's que no coincidan con alguna del listado se rechazarán.

La opción "Disconnect idle clientes after (seconds)", desconectará todos los clientes transcurrido el tiempo en segundos que se indique.

Por último, si no se va a usar el cliente en Java, se puede desconectar la casilla Server Java viewer vía HTTP on port, y cerrar así otra posible puerta de acceso desde fuera.

C.2.2.3 Pestaña "Desktop"

Se pueden desactivar algunas opciones del S.O. controlado con el fin de mejorar el rendimiento del control remoto, concretamente el del rendimiento de la conexión. A menor ancho de banda disponible, es preferible desactivar estas opciones para obtener una conexión que no sea muy ralentizada. Tal y como se puede ver en la *Fig.C.15*, con *Remove wallpaper y Remove background pattern,* se desactivan respectivamente la imagen y el fondo del escritorio de Windows, mientras que con *Disable user interface effects* se eliminan efectos como por ejemplo el de difuminado de los menús.

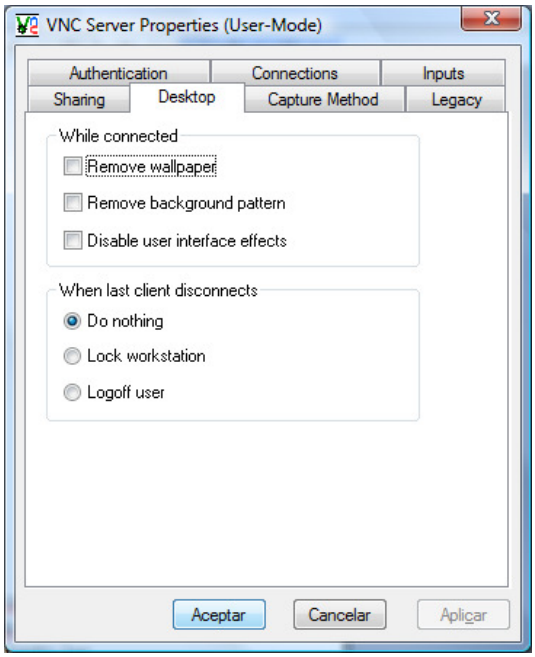

Fig.C.15 Opciones de escritorio en VNC.

Es posible acceder a estas opciones desde el icono que aparece en la barra de tareas de Windows, haciendo click en el botón derecho del ratón y yendo a "Options" (*Fig.C.16*).

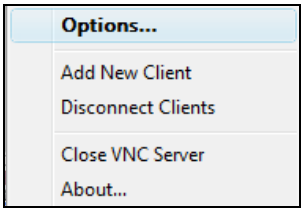

Fig.C.16 Acceder a las opciones de VNC.

C.2.3 Configuración de RealVNC en el cliente.

Una vez instalado, se ejecuta desde Inicio – Programas - RealVNC - VNC Viewer 4 - Run VNC Viewer (Fig.C.17).

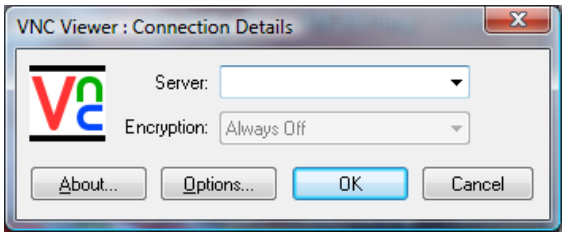

Fig.C.17 Pantalla de VNC Viewer.

En el campo Server hay que teclear la IP o nombre del ordenador. Después de introducir la contraseña, ya se tendrá control sobre el ordenador al que se esté conectando.

En el botón Options de la *Fig.C.17* se encuentran las siguientes opciones:

- **Colour & Encoding**: en Color level se puede ajustar el número de colores que se quiera visualizar en la pantalla. A mayor número de colores, más ancho de banda es necesario.

- **Inputs**: aquí se configura qué órdenes aceptará el servidor: las del ratón (send pointer); las del teclado (send keyboard); que el contenido de nuestro portapapeles pase al servidor (send clipboard); al contrario, que el portapapeles de la máquina controlada pase al nuestro (accept clipboard) y las teclas especiales como la de Windows o la de Aplicación (pass special keys). También se encuentra una tecla Menu Key (inicialmente F8) para poder sacar el menú cuando se encuentra en el modo "pantalla completa".

C.3. Uso de RealVNC en el cliente.

Durante la conexión remota a un ordenador, se tiene disponibles algunas opciones: es posible pulsar la barra de título con el botón derecho del ratón o dar a la tecla Menu Key (normalmente F8) para sacar el menú. Las opciones más significativas son:

- **Full screen**: para pasar a pantalla completa.
- **Ctrl**/**Alt**: escogiendo una de las dos, se envía la pulsación de esa tecla al PC remoto.
- **Send Ctrl-Alt-Del**: La pulsación de Control+Alt+Supr (*Del* en teclados ingleses) seguirá afectando al equipo cliente, no al remoto. Si se quiere enviar esta orden al servidor, hay que usar esta opción.
- **Refresh screen**: en conexiones lentas o saturadas, a veces se tiene que usar esta opción para actualizar la imagen.

C.4. Instalación, configuración, y uso de TightVNC.

TightVNC es una versión ligera, compatible y mejorada de RealVNC. Se puede descargar TightVNC desde su página Web oficial http://www.tightvnc.com. La instalación es muy sencilla, y sólo hay que pulsar en siguiente en las pantallas que nos va mostrando (*Fig.C.18*).

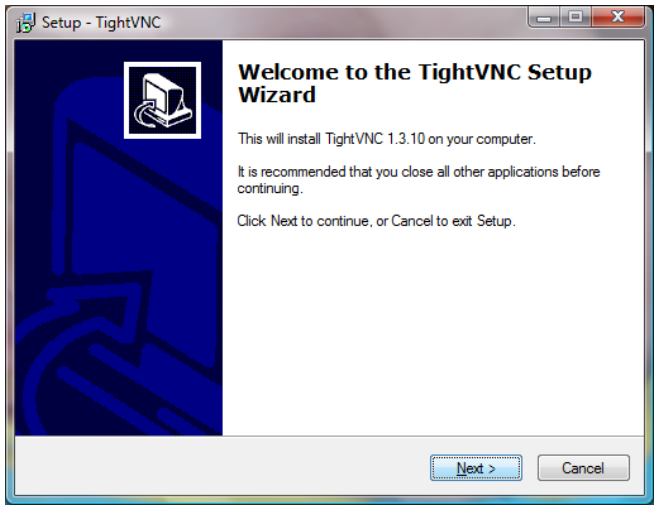

Fig.C.18 Instalación de TightVNC.

La configuración del servicio en modo servidor es similar a RealVNC. Por otro lado las opciones del software son algo diferentes aunque también se cuenta con la posibilidad de poner una contraseña de conexión tal y como se muestra en la *Fig.C.19*. No obstante TightVNC no permite limitar el rango de IP's al que se permite conectarse al ordenador. Esto no es un problema ya que se puede realizar esta limitación desde el Firewall de Windows, creando reglas para VNC, como se explicará en el anexo D. El resto de opciones se deben dejar tal y como vienen por defecto.

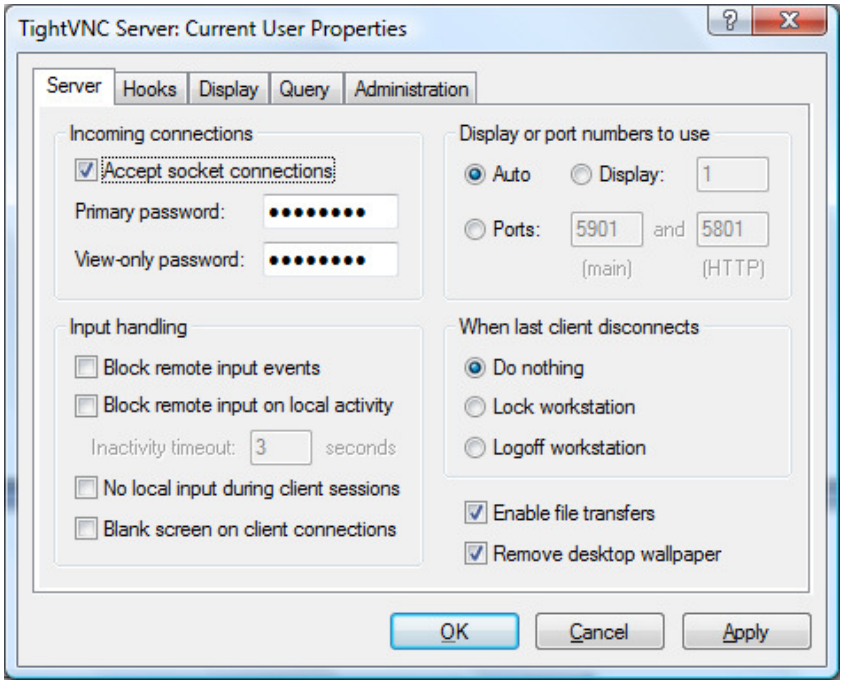

Fig.C.19 Opciones de TightVNC.

Se ha de tener en cuenta que hay que incluir dentro de la carpeta de inicio de Windows el icono de TightVNC para que cada vez que se inicie el SO lance el VNC Server.

Es importante recalcar que, en Windows, la configuración de este programa se limita al ámbito del usuario en cuestión con el que se esté validando en ese momento. Esto significa que, una vez hecha una configuración determinada con una contraseña concreta, si posteriormente se cambia de sesión no se mantendrá.

El uso de TightVNC es muy simple y similar al explicado anteriormente en RealVNC. Cualquier usuario inexperto será capaz de trabajar con este escritorio remoto sin problemas, ya que a la práctica es como si estuviera trabajando delante de la misma máquina a la que se conecta remotamente.

D. Configuración del Firewall.

A continuación se explica como se han configurado los firewall correspondientes a cada uno de los 3 S.O. utilizados. De este modo se ha permitido el acceso a los puertos utilizados para las aplicaciones siguientes:

- **SAMBA**. Utiliza los puertos:
  - TCP. Puertos 139 y 445.
  - UDP. Puertos 137 y 138.
- **VNC**. Utiliza el puerto 5900 de TCP.
- **OPENVPN**. Utiliza el puerto 1194 UDP.

A continuación se explica la configuración de los firewall de Linux y Windows tomando como ejemplo la aplicación VNC. A su vez, hay que crear las reglas propias de cada aplicación con su configuración correspondiente.

D.1. Configuración del Firewall en Windows XP.

Desde las propiedades del adaptador de red del ordenador se puede acceder a la configuración del firewall. Para ello se debe ir a la pestaña “Opciones avanzadas” y pulsar “Configuración” tal y como se muestra en la *Fig.D.1*.

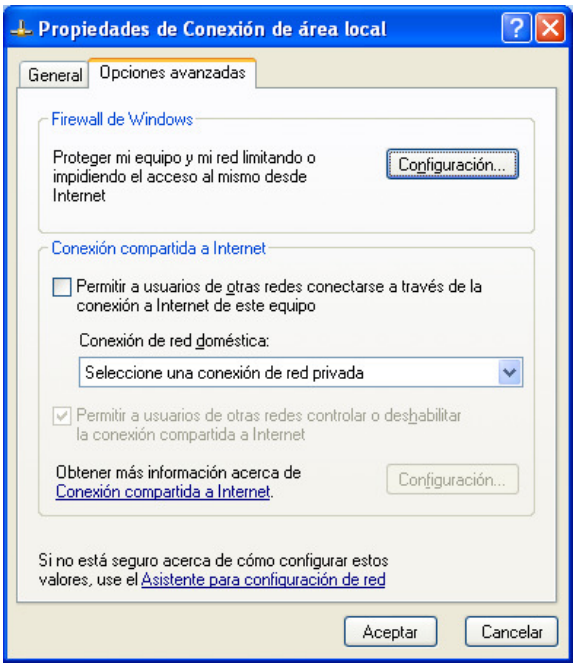

Fig.D.1 Acceso a la configuración del firewall.

En la siguiente ventana hay que acceder a la pestaña “Excepciones”, donde aparecen las entradas permitidas hacia el ordenador (Fig.D.2).

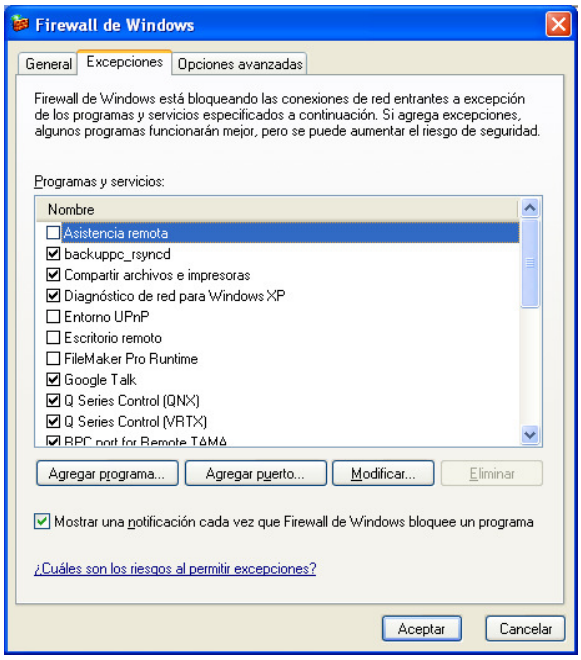

Fig.D.2 Entradas permitidas por el firewall de Windows.

Al pulsar en agregar puerto para añadir una nueva entrada para el firewall (*Fig.D.2*). aparece una nueva ventana (*Fig.D.3*).

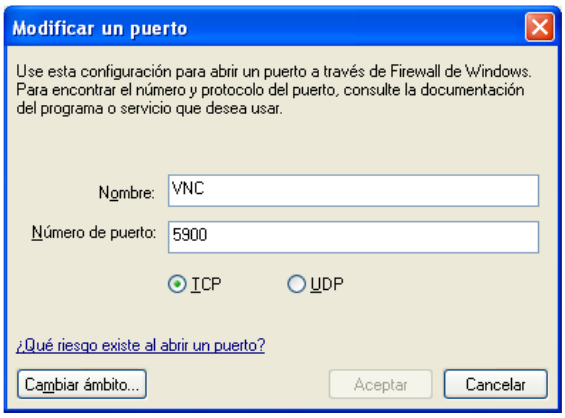

Fig.D.3 Agregar nueva entrada.

Pulsar en cambiar ámbito para definir el rango de IP's que tendrán permiso de conexión, en nuestro caso el de la VPN, (*Fig.D.4*).

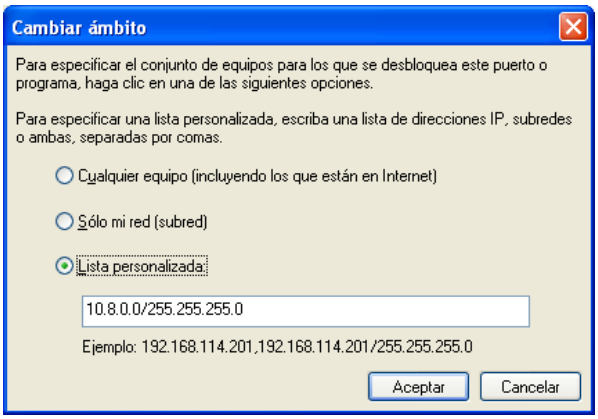

Fig.D.4 Rango de IP's para una nueva entrada.

Para asegurar que el firewall está activado se puede revisar el estado de activación en la pestaña General (*Fig.D.5*).

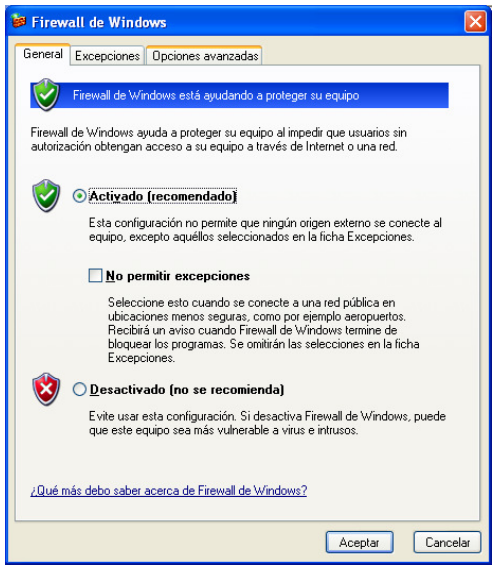

Fig.D.5 Activar firewall.

D.2. Configuración del Firewall en Windows Vista.

En nuestro caso se ha usado Windows Vista un portátil para el caso de roadwarrior. Windows Vista dispone también de Firewall configurable de una manera muy similar al anterior. Para acceder a la consola de configuración se debe acceder a través de Inicio-Ejecutar y escribir "mmc". En la pantalla emergente se debe añadir el complemento propio del Firewall y pulsar en agregar (*Fig.D.6*).

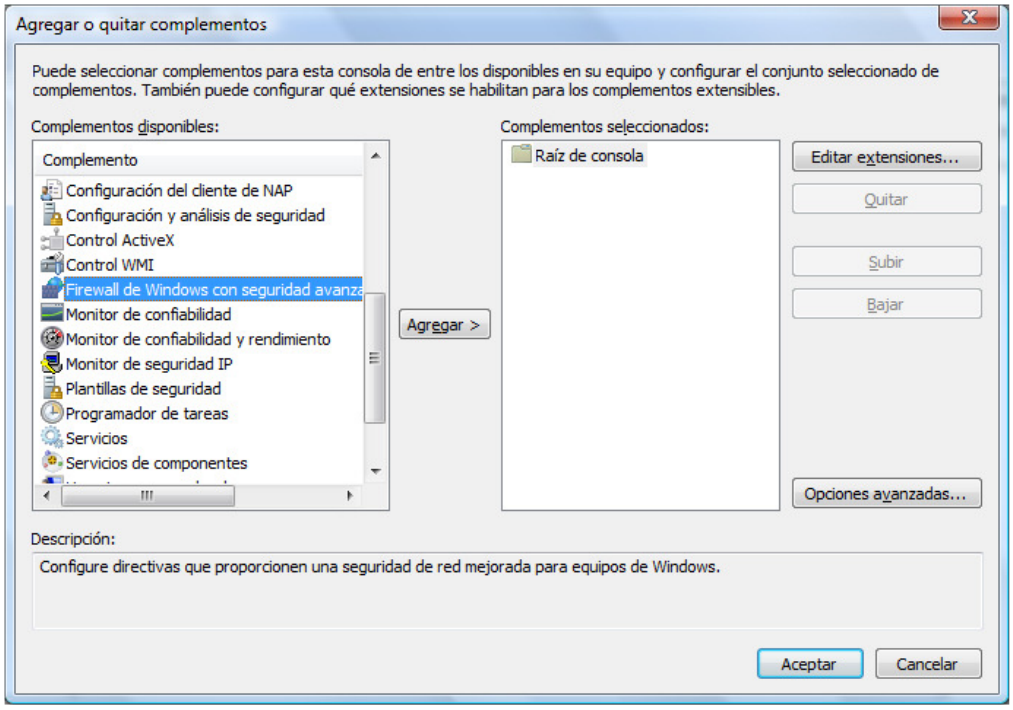

Fig.D.6 Agregar o quitar firewall de Windows.

En este firewall quedan más visibles las partes de reglas de entrada y reglas de salida del Firewall. Para añadir las reglas que interesan se debe pulsar en nueva regla del menú y se ha de seguir el asistente para crear la regla que permita el tráfico entrante en los puertos de los protocolos TCP y UDP necesarios (*Fig.D.7*).

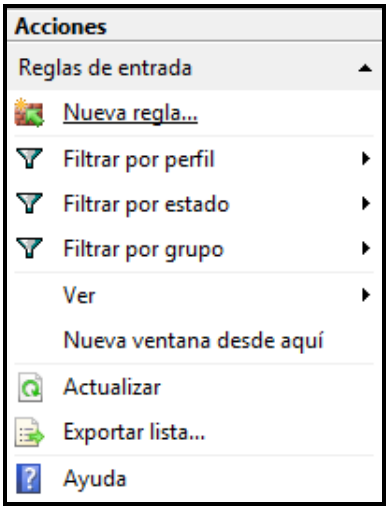

Fig.D.7 Añadir nueva regla en el firewall.

En Tipo de regla, seleccionar la opción de puerto (*Fig.D.8*).

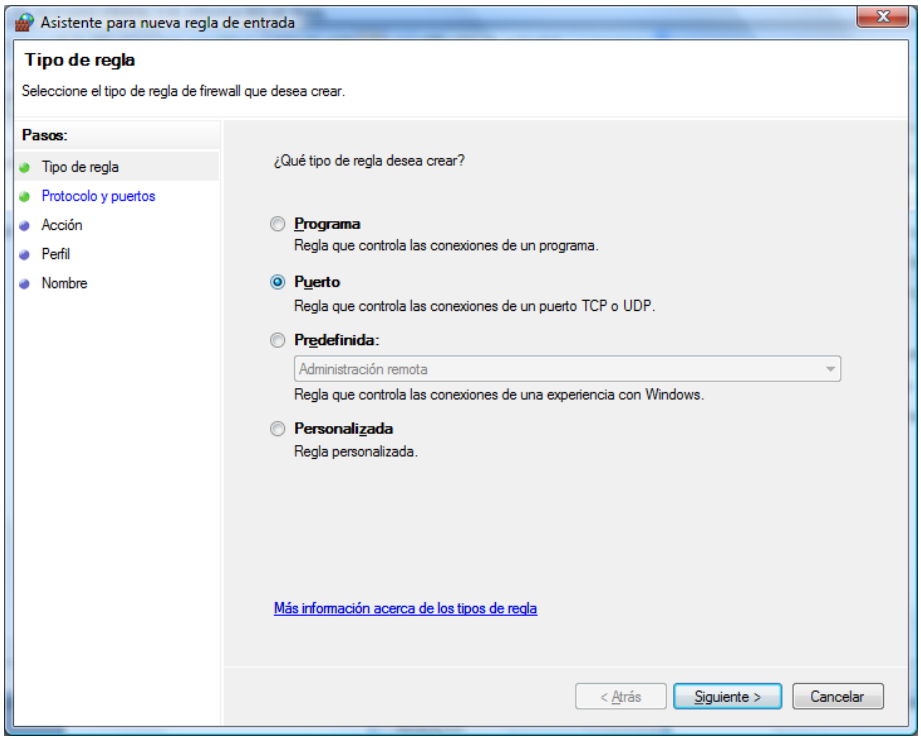

Fig.D.8 Tipo de regla en el firewall.

Seguidamente seleccionar el protocolo que se desea, e introducir el puerto en cuestión (*Fig.D.9*).

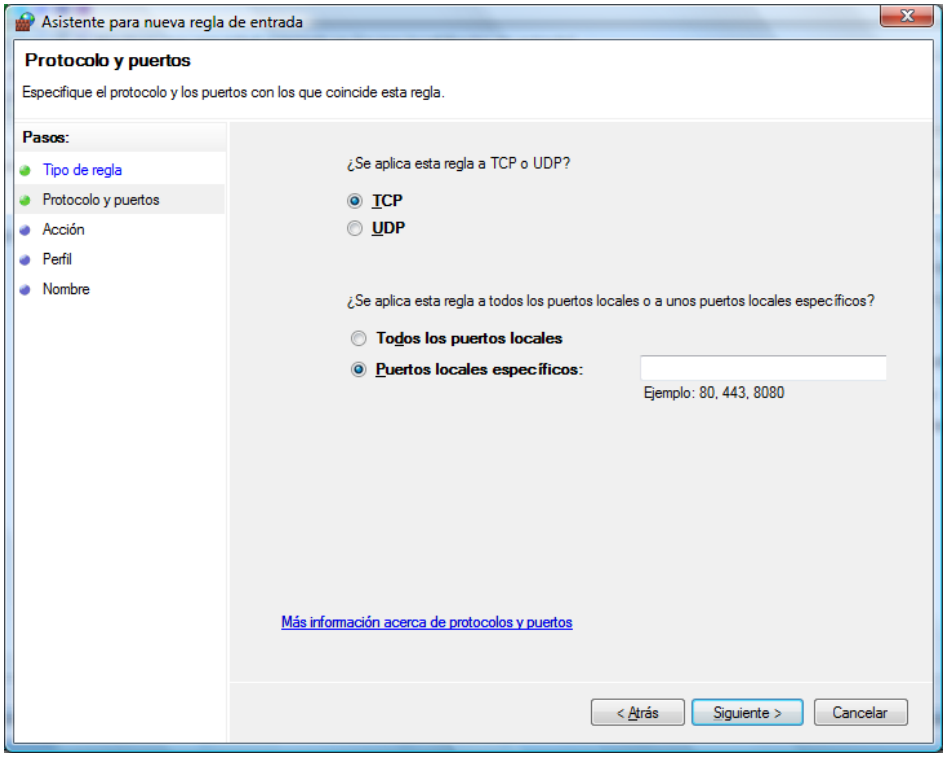

Fig.D.9 Protocolo y puerto de las reglas en el firewall.

Hay que marcar la opción permitir conexión (*Fig.D.10*).

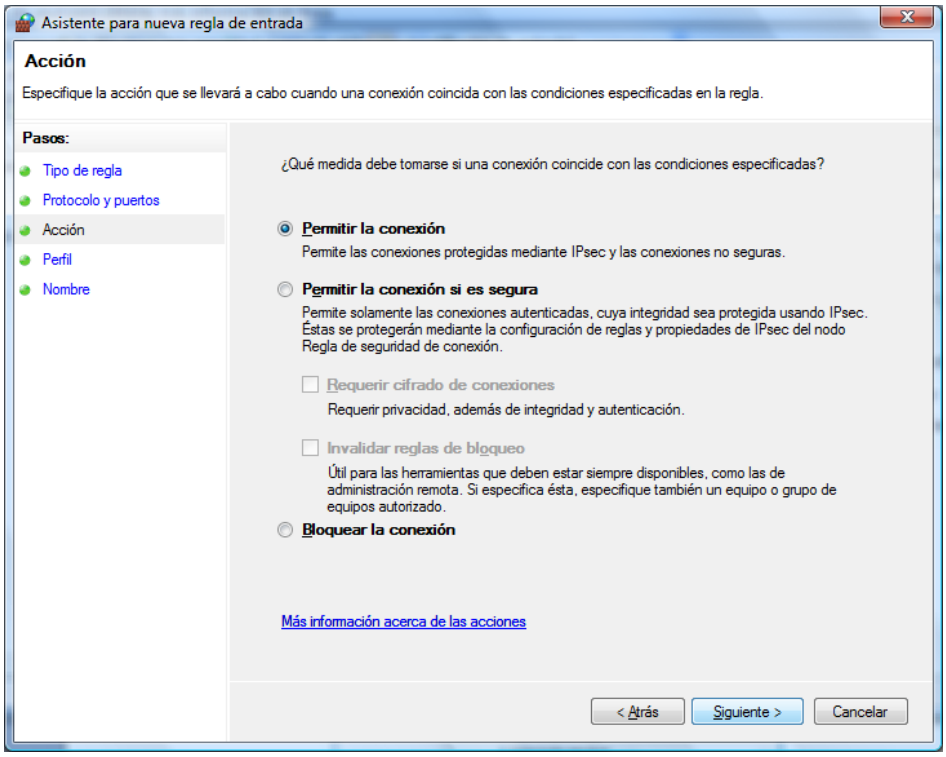

Fig.D.10 Permiso de conexión en el firewall.

Posteriormente nos solicitará el nombre de la regla que se acaba de crear. Hay que asignar el nombre de la aplicación a la cual se ha dado acceso.

Una vez creadas las reglas el aspecto es el que se muestra en la *Fig.D.11*:

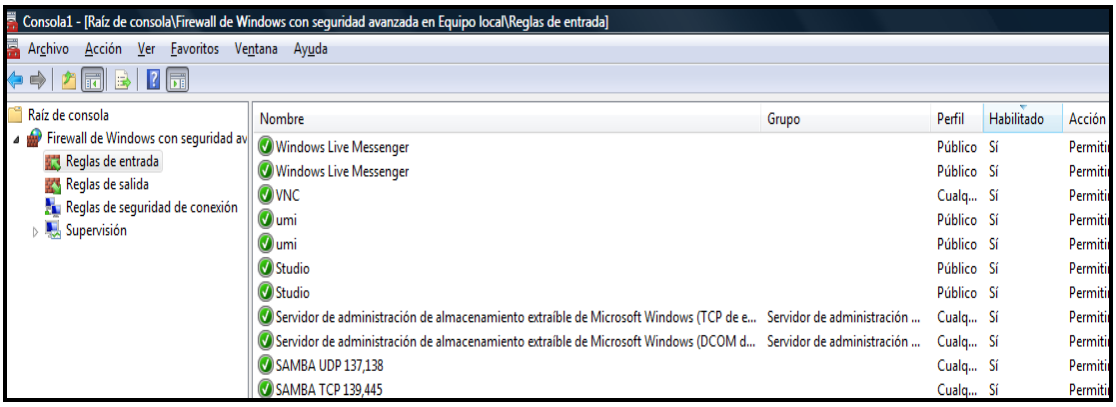

Fig.D.11 Reglas creadas en el firewall.

A modo de comprobación se revisa el del VNC pulsando doble click sobre la regla (*Fig.D.12*):

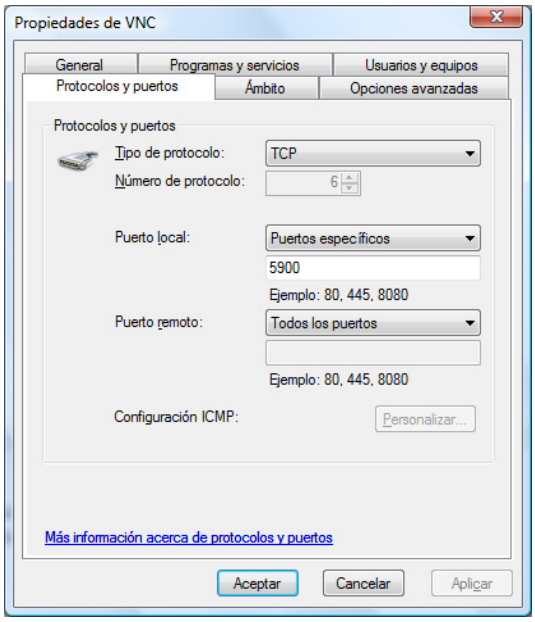

Fig.D.12 Regla en el firewall para VNC.

Una vez creadas las reglas hay que limitar el acceso al ámbito del rango perteneciente a la VPN, tal y como se muestra en la *Fig.D.13*:

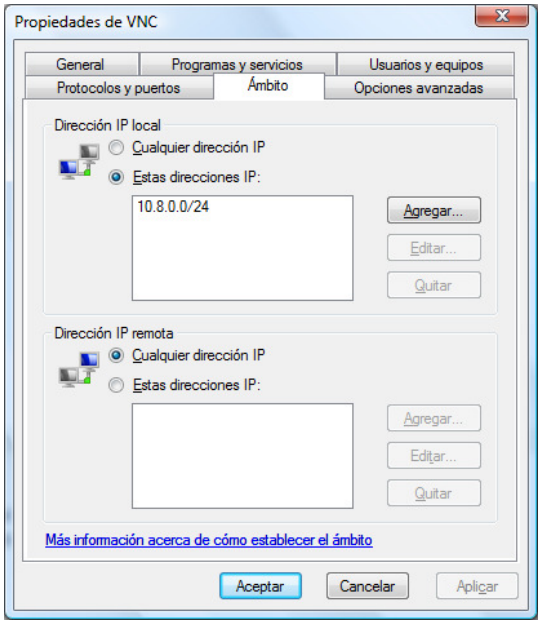

Fig.D.13 Limitar rango de IP's para una regla creada.

D.3. Configuración por comando de IPTables.

Así pues se pasa a explicar como ha quedado configurado en Linux IPtables a través de su interfaz gráfica, que de manera directa traduce la configuración a

comandos y los escribe sobre este fichero. No se entrará en el detalle de configuración a nivel de comandos de iptables, puesto que desde la interfaz gráfica es mucho más sencillo de realizar. No obstante se explica brevemente como funciona iptables.

El comando iptables tiene las siguientes formas de invocación: los ítems entre claves, {...|...|...}, son requeridos, pero solo se puede indicar uno en cada caso. En cuanto a los ítems entre corchetes, [...], son opcionales.

iptables { -A | --append | -D | --delete } cadena especificación-de-regla [opciones]

Esta forma de invocación del comando, agrega (-A o --append) o elimina (-D o --delete) una regla de la cadena especificada. Por ejemplo, para agregar una regla a la cadena de INPUT en la tabla *filter* que descarte todos los paquetes UDP, se usa este comando:

iptables -A INPUT -p udp -j DROP

Para borrar la regla agregada por el comando anterior, se usa este comando:

iptables -D INPUT -p udp -j DROP

D.4. Configuración gráfica del Firewall en OPENSUSE.

Desde el acceso al software de OpenSuse tal y como se muestra en la *Fig.D.14*, se accede a la configuración del firewall.

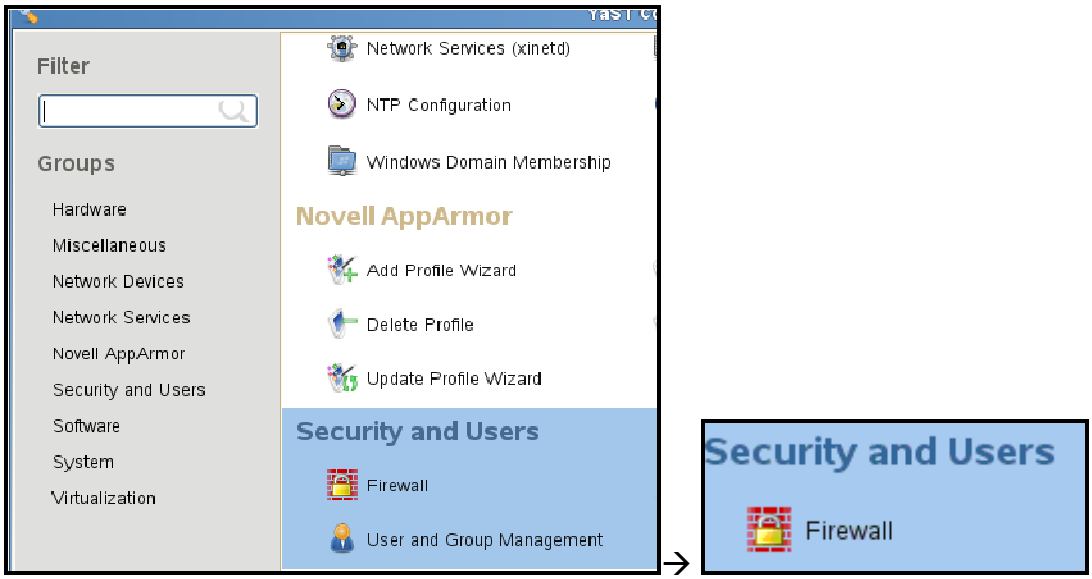

Fig.D.14 Firewall en OpenSuse.

Una vez en la pantalla principal se ha de habilitar el inicio automático del servicio marcando la opción “Enable Firewall Automatic Starting”, tal y como se muestra en la *Fig.D.15*.

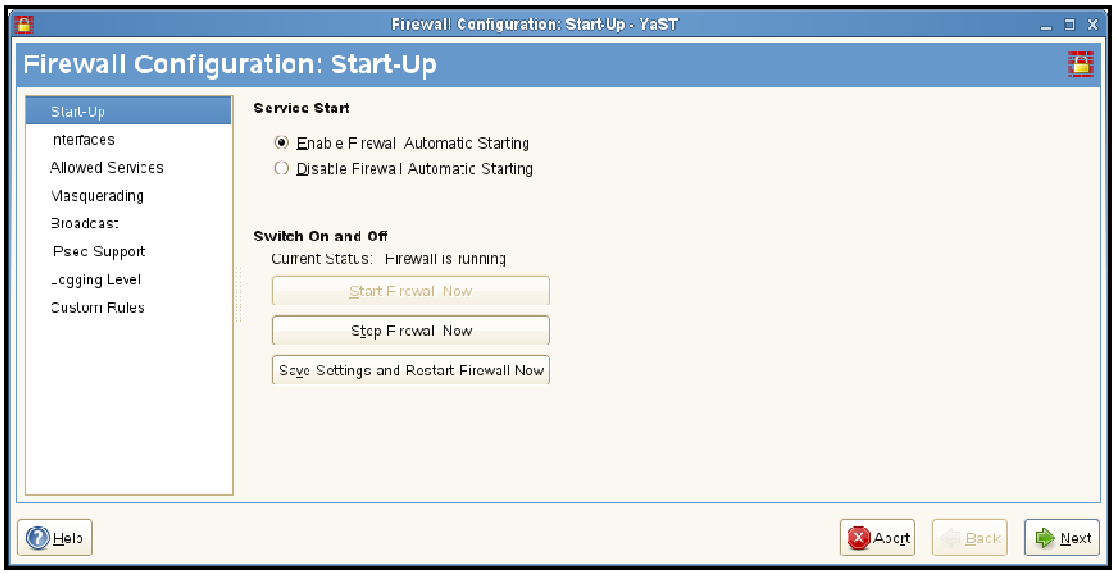

Fig.D.15 Configuración del firewall.

Para crear las reglas de acceso hay que acceder a la parte de “Custom Rules” del menú izquierdo y pulsar en “Add” (*Fig.D.16*).

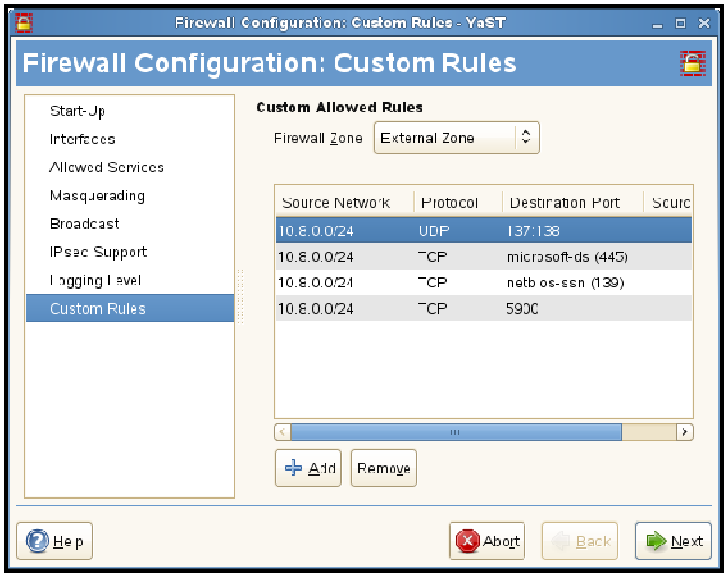

Fig.D.16 Configuración del firewall.

En la ventana siguiente hay que seleccionar el protocolo sobre el que se quiera realizar la regla, el rango de ordenadores a los que se permitirá este acceso, y

el puerto necesario. Para el caso del software VNC, a modo de ejemplo, ver la *Fig.D.17*.

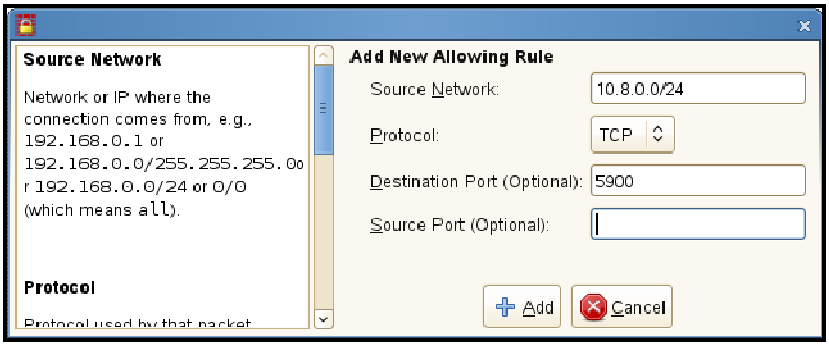

Fig.D.17 Añadir nueva regla en el firewall.

Una vez añadidas todas las reglas, el aspecto es el de la *Fig.D.18*:

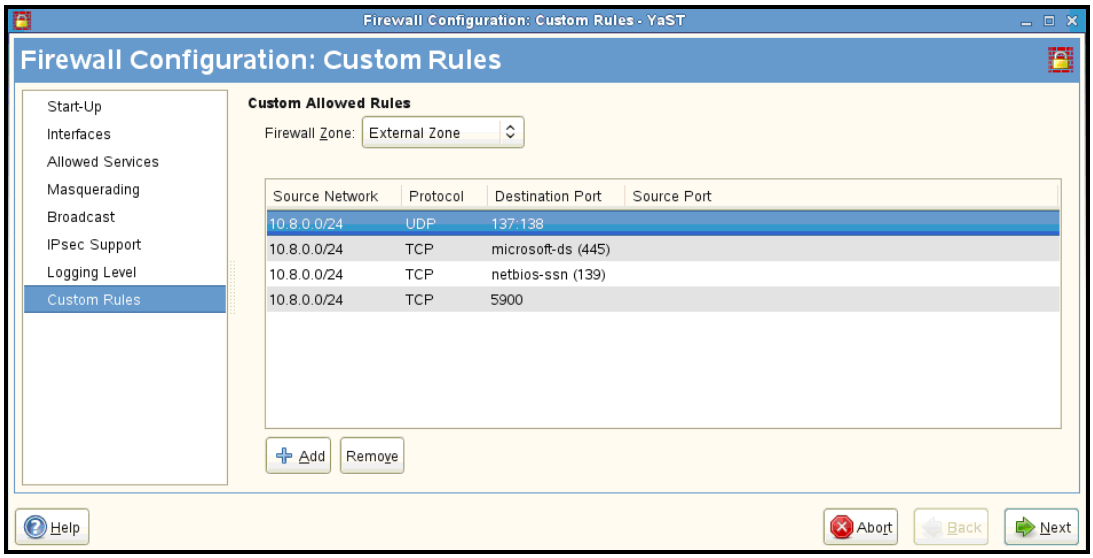

Fig.D.18 Reglas creadas en el firewall.

Para abrir el puerto correspondiente a OPENVPN, puerto número 1194, hay que hacerlo a través de “Allowed Services” y pulsar en “Advanced” tal y como se muestra en la *Fig.4.19* y *Fig.4.20*. De esta manera se puede permitir su entrada desde cualquier ordenador que en el futuro quiera conectarse a la VPN.

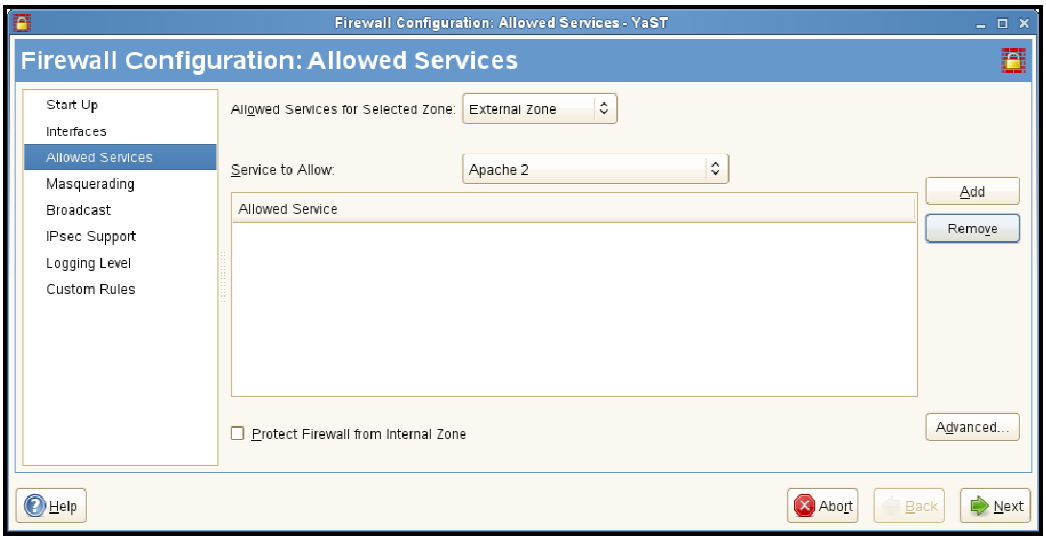

Fig.4.19 Abrir puerto en el firewall.

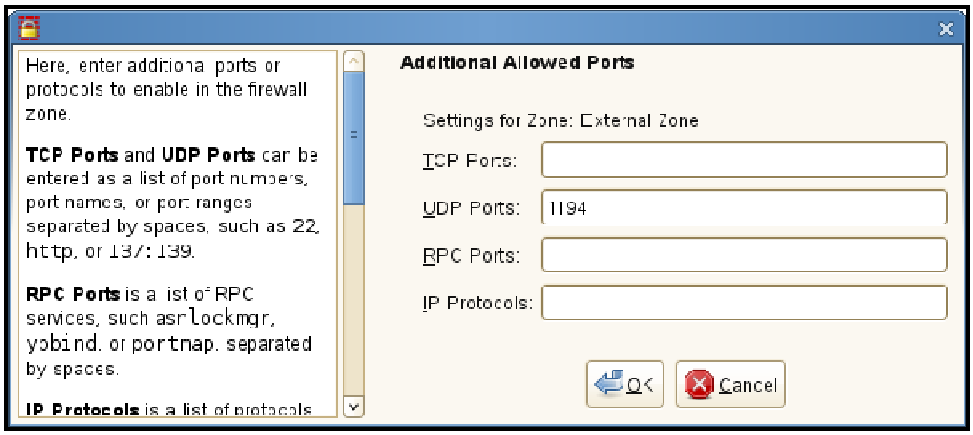

Fig.4.20 Abrir puerto en el firewall.

Esta configuración se ve reflejada en el fichero ubicado en *etc/sysconfig/susefirewall2*. Al realizar la configuración anterior se añaden las líneas siguientes:

- Puertos para SAMBA Y VNC.

FW_SERVICES_ACCEPT_EXT="10.8.0.0/24,udp,137:138 10.8.0.0/24,tcp,445 10.8.0.0/24,tcp,139 10.8.0.0/24,tcp,5900"

- Puerto para OPENVPN.

FW_SERVICES_EXT_UDP="1194"

D.5. Configuración gráfica del Firewall en UBUNTU

De igual manera que en el caso anterior, para este S.O. se debe acceder a la configuración del Firewall. En la ventana principal seleccionar “Deny incoming traffic” (denegar el tráfico entrante a todo, referenciado a la opción anteriormente explicada de firewall conocida como restrictiva), tal y como aparece en la *Fig.4.21*.

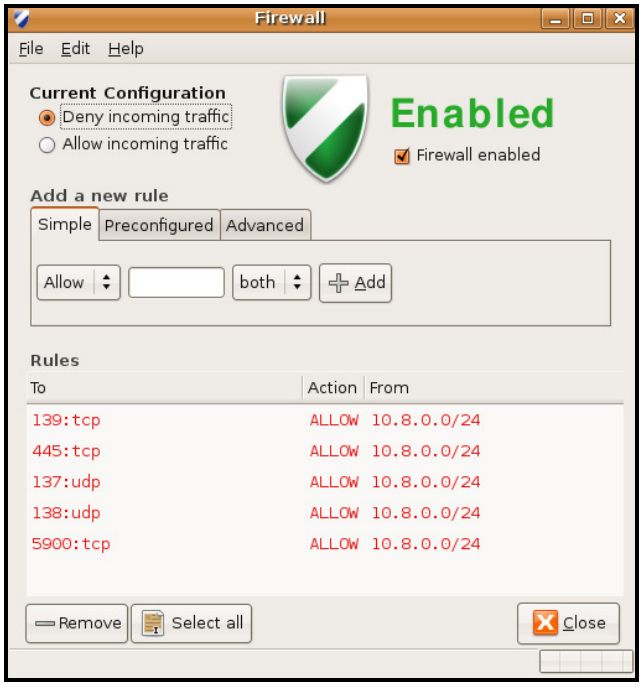

Fig.4.21 Configuración gráfica del firewall.

Para añadir nuevas reglas ir a la pestaña “Advanced” e introducir en el campo From el rango perteneciente a la VPN, en el puerto especificar el de la aplicación correspondiente, escogiendo el protocolo al que pertenece. En la *Fig.4.22* se muestra como añadir una entrada nueva a modo de ejemplo. Se debe seleccionar el protocolo, escribir la información correspondiente al rango de IP’s permitidos en el campo From y el número de puerto de la regla a crear. El resto de opciones se dejan como aparecen por defecto.

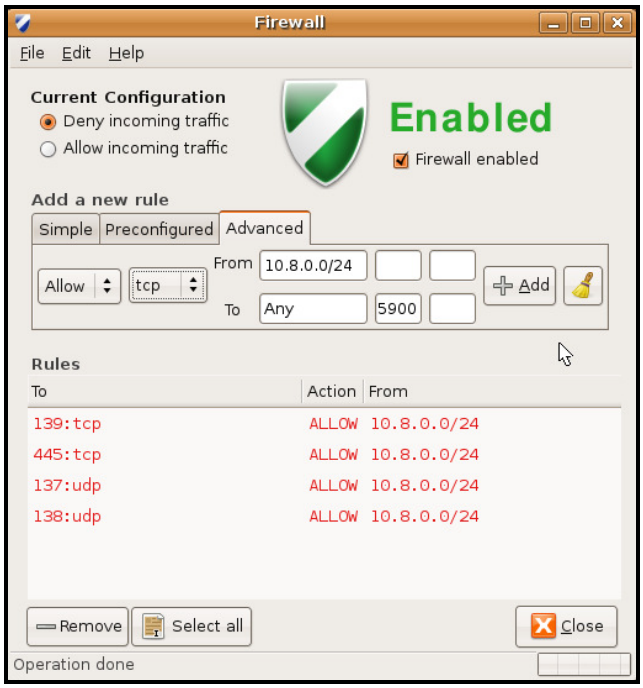

Fig.4.22 Añadir una nueva regla en el firewall.

Tal y como se ha comentado anteriormente para añadir el puerto 1194 para OPENVPN se realiza a través de la pestaña “Simple” (*Fig.4.23*).

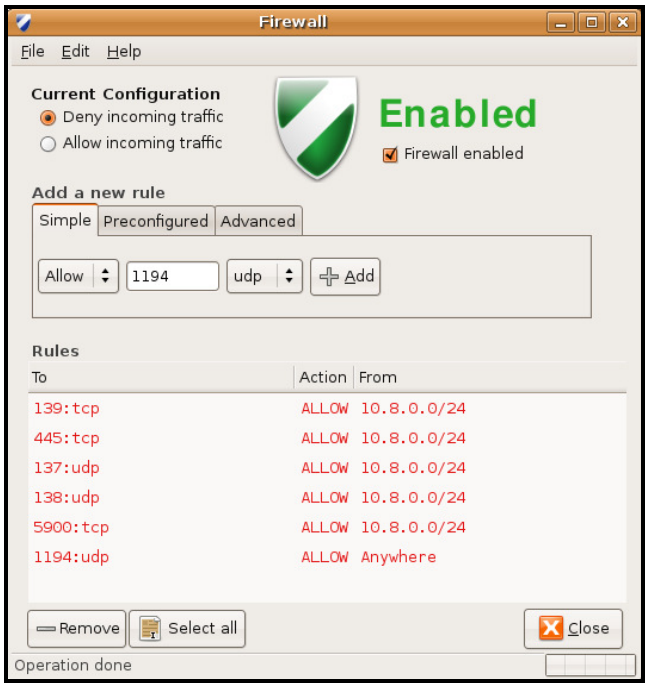

Fig.4.23 Permitir un puerto en el firewall.

Es posible ejecutar varios comandos de este firewall mediante terminal. A continuación se muestran los más básicos:

- Mostrar las reglas aplicadas → *sudo ufw status*.
- Habilitar el firewall → *sudo ufw enable*
- Deshabilitar el firewall →*sudo ufw disable*

E. Diffie Hellman.

El protocolo Diffie Hellman desarrollado por Whitfield Diffie y Martin Hellman en 1976, permite intercambiar claves entre dos extremos en un canal inseguro, de forma anónima.

Este protocolo fue el punto de partida para los sistemas asimétricos.

Su seguridad se basa en la complejidad en el cálculo de sus algoritmos.

A continuación se puede ver como se realiza el cálculo de las claves de ambos extremos y un ejemplo de ello.

El usuario A escoge un número q primo muy grande, y un número Xa que será la clave privada y deberá ser tal que Xa<q.

El usuario B escogerá un número p que ha de ser p<q siendo p raíz primitiva de q, y un número Xb que será la clave privada y deberá ser tal que Xb<q.
.
Se dice que p es raíz primitiva de q si las potencias de p generan todos los enteros desde 1 hasta q-1:

p mod q
p^2 mod q
...
p^(q-1) mod q

A enviará a B su número q, y B enviará a A su número p. El envío es público, y por tanto visible para cualquiera.

Ahora se generarán las claves públicas de usuario:

A calculará su clave pública Ya de la siguiente forma:

Ya = p^Xa mod q

B calculará su clave pública Yb de la siguiente forma:

Yb = p^Xb mod q

Ahora A enviará a B su clave pública Ya, y B enviará a A su clave pública Yb. El envío es público, y por tanto visible para cualquiera.

Una vez ambos usuarios tienen la clave pública del otro extremo, calculan la clave compartida que usarán.

El usuario A calcula:

K = (Yb)^Xa mod q

Y el usuario B calcula:

K = (Ya)^Xb mod q

Este número K es idéntico en los dos usuarios. K es la clave simétrica que comparten ambos usuarios, y que pueden usar para establecer una comunicación cifrada.

Se puede hacer una comprobación de K de la siguiente forma:

(Yb)^Xa mod q = (p^Xb)^Xa mod q = p^(Xb*Xa) = (p^Xa)^Xb mod q = Ya^Xb mod q

F. Funciones de Hash

Las funciones de hash son operaciones que se realizan sobre un texto.

En nuestro caso se han usado funciones hash para garantizar la integración de los datos. Para ello se usan códigos hash de autenticación de mensajes (HMAC) para comprobar que la información recibida en el receptor corresponde con la información enviada por el emisor.

En primer lugar el emisor realiza un cálculo matemático sobre el mensaje, utilizando el código ASCII que asigna un número a cada letra (*Fig.F.1*).

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Fig.F.1 Código ASCII.

Si se coge el mensaje:

"Crear una VPN"

Se calcula el valor de cada palabra según la tabla ASCII:

Crear = 67+114+101+97+114 = 493
una = 117+110+97 = 324
VPN = 86+80+78 = 244

Y ahora se realiza una función matemática, en la cual se han puesto previamente de acuerdo ambos extremos de la comunicación, como por ejemplo:

(primer número * segundo número) – tercer número = 493 * 324 – 244 = 159488

Si el mensaje es más grande, se puede dividir en bloques de palabras.

Finalmente el emisor envía el mensaje, añadiendo al final el valor de hash obtenido.

Al recibir el mensaje el receptor, realizará la misma operación para calcular el valor de hash. Si coincide con el valor del mensaje, quiere decir que la información no ha sido alterada y es correcta.

G. SSL/TLS

G.1. Introducción

- Antes de comenzar a explicar este punto, es importante definir el concepto de criptografía. Su origen griego significa escritura oculta, y es la ciencia de cifrar y descifrar información utilizando técnicas especiales que permiten que los mensajes sean leídos sólo por las personas a las que realmente van dirigidos. Estas personas, disponen de los medios adecuados para descifrarlo, ya sea por cifrado simétrico o asimétrico (capítulo 4.4 de la memoria).

G.2. Protocolos.

Algunos protocolos que usan los cifrados simétricos o asimétricos son:

- DSS ("Digital Signature Standard") con el algoritmo DSA ("Digital Signature Algorithm")
- PGP
- SSH
- SSL, ahora un estándar del IETF
- TLS

Herramientas como PGP, SSH o la capa de seguridad SSL para la jerarquía de protocolos TCP/IP utilizan un híbrido formado por la criptografía asimétrica para intercambiar claves de criptografía simétrica, y la criptografía simétrica para la transmisión de la información.

El protocolo TLS es una mejora de otro protocolo anterior llamado SSL. Se comenta brevemente este último para comprender mejor a su sucesor, TLS. El protocolo SSL, permite establecer una conexión segura por medio de un canal cifrado entre un cliente y un servidor, lo que posibilita que la comunicación se realice en un entorno seguro. Su diseño está basado en poder trabajar de una manera modular (extensible), con compatibilidad total entre sus versiones (anteriores y posteriores), y negociación entre las partes (peer-to-peer). Normalmente, el servidor es el único que es autenticado, asegurando así su identidad, pero el cliente se mantiene sin autenticar, ya que para la autenticación mutua se necesita una infraestructura de claves públicas o PKI para los clientes. Permite prevenir escuchas secretas (termino conocido en inglés como eavesdropping), evita la suplantación de identidad del emisor y mantiene la integridad del mensaje.

La versión 3.0 de SSL fue desarrollada por Netscape en el año 1996 y sirvió de base para desarrollar la versión 1.0 de TLS, protocolo estándar IETF definido en el RFC 2246. Cabe comentar que un RFC es un documento con una propuesta oficial para un nuevo protocolo de Internet, en el que se explica con todo detalle. Si resulta lo bastante interesante para la IETF (Internet Engineering Task Force → Grupo de Trabajo en Ingeniería de Internet, que es una organización internacional abierta de normalización con el objetivo de

contribuir a la ingeniería de Internet) que puede llegar a convertirse en un estándar de Internet.

Las primeras implementaciones de SSL sólo podían usar claves simétricas de 40 bits como máximo, ya que el gobierno de los EEUU imponía restricciones sobre la exportación de tecnología criptográfica. Esta clave era de 40 bits ya que las agencias de seguridad nacional americanas podían atacarla mediante fuerza bruta y poder leer así el tráfico cifrado, mientras que los posibles atacantes con menores recursos no podrían leerlo. Tras la aparición de mejores productos criptográficos diseñados en otros países, se rebajaron las restricciones de exportación de tecnología criptográfica que finalmente desembocó en la desaparición casi por completo la limitación. Hoy día se usan claves de 128 bits, o incluso de más, para las claves de cifrado simétricas.

G.3. DESCRIPCIÓN.

El protocolo SSL/TSL se basa en tres fases:

G.3.1 NEGOCIACIÓN.

Los extremos de la comunicación (cliente y servidor) negocian los algoritmos criptográficos que utilizarán para autenticarse y cifrar la información. Las opciones disponibles son:

- **Criptografía de clave pública**: RSA, Diffie-Hellman, DSA (Digital Signature Algorithm) o Fortezza.
- **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.

G.3.2 AUTENTICACIÓN Y CLAVES.

Según la negociación anterior, se realiza la autenticación mediante certificados digitales y se intercambian las claves para el cifrado.

G.3.3 TRANSMISIÓN SEGURA.

Comienza el tráfico de información cifrada y segura entre ambos extremos de la comunicación.

G.4. OBJETIVOS DEL PROTOCOLO TLS.

A continuación se detallan:

- **Seguridad criptográfica**. El protocolo se debe emplear para establecer una conexión segura entre emisor y receptor.
- **Interoperabilidad**. Aplicaciones distintas deben poder intercambiar parámetros criptográficos sin necesidad de que ninguna de las dos conozca el código de la otra.
- **Extensibilidad.** El protocolo permite la incorporación de nuevos algoritmos criptográficos.
- **Eficiencia**. Los algoritmos criptográficos son computacionalmente costosos, por lo que el protocolo incluye un esquema de *cache de sesiones* para reducir el número de sesiones que deben inicializarse desde cero (usando criptografía de clave pública).

G.5. FUNCIONAMIENTO DEL PROTOCOLO TLS

El protocolo está dividido en dos niveles:

- Protocolo de registro TLS (*TLS Record Protocol*).
- Protocolo de mutuo acuerdo TLS (*TLS Handshake Protocol*).

El de más bajo nivel es el *Protocolo de Registro*, que se implementa sobre un protocolo de transporte fiable como el TCP. Un protocolo de bajo nivel controla el acceso al medio físico, lo que se conoce como MAC (Media Access Control) y, además, parte del nivel de transmisión de datos, ya que se encarga también de las señales de temporización de la transmisión. El protocolo proporciona seguridad en la conexión con dos propiedades fundamentales:

- La conexión es privada. Para encriptar los datos se usan algoritmos de cifrado simétrico. Las claves se generan para cada conexión y se basan en un secreto negociado por otro protocolo (como el de mutuo acuerdo). El protocolo también se puede usar sin encriptación.
- La conexión es fiable. El transporte de mensajes incluye una verificación de integridad.

El Protocolo de mutuo acuerdo**,** *proporciona seguridad en la conexión con tres propiedades básicas:*

- La identidad del interlocutor puede ser autentificada usando criptografía de clave pública. Esta autentificación puede ser opcional, pero generalmente es necesaria al menos para uno de los interlocutores.
- La negociación de un secreto compartido es segura.
- La negociación es fiable, nadie puede modificar la negociación sin ser detectado por los interlocutores.

Esquema de operación del protocolo de mutuo acuerdo (TLS Handshake Protocol*), Fig.G.1.*

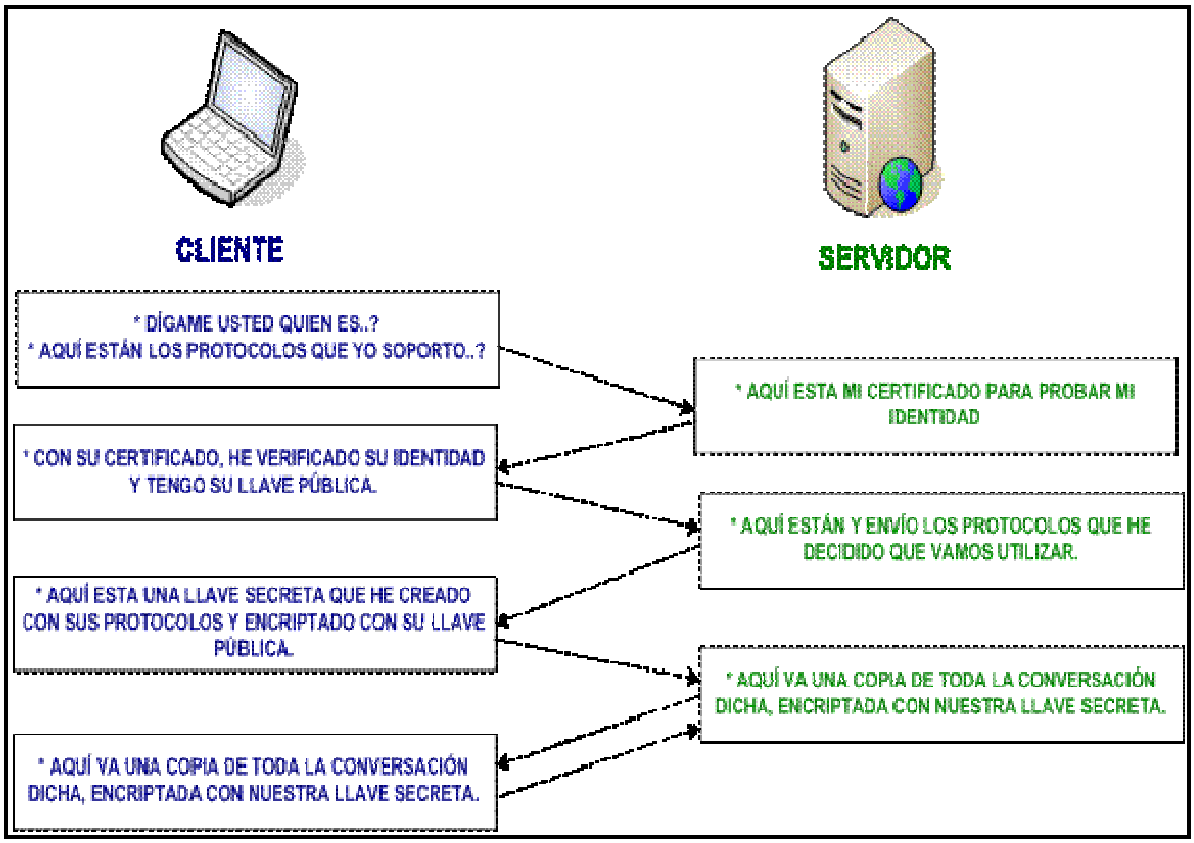

Fig.G.1 Intercambio de datos utilizando TLS/SSL.

La comunicación entre los nodos CLIENTE y SERVIDOR se basa en el intercambio de mensajes. En cada mensaje existe un campo (content_type) donde se especifica el protocolo de nivel superior utilizado. Estos mensajes pueden ser comprimidos, cifrados y empaquetados con un código de autentificación del mensaje (MAC).

G.6. EJEMPLOS

G.6.1 APLICACIONES.

El protocolo SSL/TLS se ejecuta en una capa entre los protocolos de aplicación como:

- HTTP sobre SSL/TLS es HTTPS, ofreciendo seguridad a páginas WWW para aplicaciones de comercio electrónico, utilizando certificados de clave pública para verificar la identidad de los extremos. Visa, MasterCard, American Express y muchas de las principales instituciones financieras han aprobado SSL para el comercio sobre Internet.
- SSH utiliza SSL/TLS por debajo.
- SMTP y NNTP pueden operar también de manera segura sobre SSL/TLS.
- POP3 e IMAP4 sobre SSL/TLS son POP3S e IMAPS.

También se puede ejecutar sobre el protocolo de transporte TCP, que forma parte de la familia de protocolos TCP/IP.

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

G.6.2 ESTÁNDARES.

- La primera definición de TLS apareció en el RFC 2246: "The TLS Protocol Versión 1.0" (El protocolo TLS versión 1.0) y está basada en la versión 3.0 de SSL, siendo prácticamente equivalentes. Posteriormente se extendió dicha definición.
- RFC 2712: "Addition of Kerberos Cipher Suites to Transport Layer Security (TLS)". Aparecen las familias de cifrados de 40 bits definidas, para advertir que ya han sido asignadas.
- RFC 2817: "Upgrading to TLS Within HTTP/1.1". Explica cómo usar el mecanismo de actualización en HTTP/1.1 para iniciar TLS sobre una conexión TCP existente, permitiendo al tráfico seguro e inseguro HTTP compartir el mismo puerto.
- RFC 2818: "HTTP Over TLS". Diferencia el tráfico seguro e inseguro HTTP usando un puerto de servidor diferente.
- RFC 3268: "AES Ciphersuites for TLS". Añade la familia de cifrado AES.
- RFC 3546: "Transport Layer Security (TLS) Extensions". Añade un mecanismo para negociar extensiones de protocolos durante la inicialización de sesión y define algunas extensiones.
- RFC 4279: "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)". Añade tres conjuntos de nuevas familias de cifrados para que el protocolo TLS permita la autenticación basada en claves previamente compartidas.

G.6.3 TLS 1.1.

Es la última versión aprobada del protocolo TLS. Es muy parecida a la versión anterior (TLS 1.0), pero la principal diferencia es la modificación del formato para el cifrado RSA anterior al uso de 'master secret', que es parte del mensaje de intercambio de claves del cliente (si se usa RSA). En TLS 1.0 se usaba la versión 1.5 de PCK#1, pasando a usar ahora la versión 2.1. Con este cambio se consigue protección ante ataques descubiertos por Daniel Bleichenbacher que podían lanzarse contra servidores TLS 1.0, usando PKCS#1 versión 1.5. También se incluyen recomendaciones para evitar ataques remotos programados. TLS 1.1 está actualmente implementado en el navegador Opera y en GnuTLS.

G.7. CONCLUSIÓN

La seguridad es un aspecto fundamental para muchas aplicaciones cliente-servidor, siendo un ejemplo muy importante, por su gran proyección en los

últimos tiempos, el negocio electrónico. Mediante el uso de SSL/TLS se ha conseguido aumentar la seguridad en este tipo de transacciones siempre partiendo de la base que la idea de "seguridad total" es una utopía. Es imposible conseguir una seguridad del 100 %.

El uso de SSL/TLS junto con otras técnicas como IPSec, cifrado RPC, etc, nos ayudan a mantener la confidencialidad e integridad de los datos durante la comunicación, protegiendo así datos confidenciales como números de tarjetas de crédito en las diferentes transacciones de comercio electrónico, envío de información privada, en una intranet o a través de Internet, de una organización, etc.

No hay que olvidar que existen múltiples ataques y cada vez más sofisticados. Esto obliga a una investigación permanente para mejorar los protocolos de seguridad. Podríamos decir que hoy en día se obtiene un nivel de seguridad muy aceptable haciendo un uso correcto de estos protocolos.

H. Controlador e interfaz de red.

H.1. Controlador de red.

Uno o varios archivos que permiten que un S.O. interactúe con la tarjeta de red del ordenador. Algunos sistemas operativos como Windows suelen tener una gran base de datos de controladores para poder ser compatibles con diversos hardwares. Sin embargo a veces es necesario instalar los controladores propios del fabricante de la tarjeta (a menudo descargables desde la Web de este).

H.2. Interfaz de red (o tarjeta de red):

Es un tipo de tarjeta (que puede estar integrada en la placa base del ordenador o ser de expansión) que permite conectar un dispositivo a una red (desde un ordenador o impresora a un teléfono por ejemplo), permitiendo compartir así recursos de esta. El caso mas habitual hoy en día es la tarjeta con conector RJ-45. Cada tarjeta de red tiene un número de identificación único compuesto por 48 bits (6 octetos) en hexadecimal, llamado dirección MAC (administradas por el Institute of Electronic and Electrical Engineers → IEEE). Los tres primeros octetos del número MAC identifican al fabricante de la tarjeta, y los tres siguientes el identificador de la propia tarjeta.

I. NAT.

Corresponde a Network Address Translation - Traducción de Dirección de Red). Se trata de un mecanismo utilizado por routers IP para intercambiar paquetes entre dos redes que se asignan mutuamente direcciones incompatibles. Consiste en convertir en tiempo real las direcciones utilizadas en los paquetes transportados. También es necesario editar los paquetes para permitir la operación de protocolos, que incluyen información de direcciones dentro de la conversación del protocolo.

J. XLOCK.

Para aumentar la seguridad en el acceso de usuario del servidor OpenVPN, se ha instalado esta aplicación. Es un protector de pantalla que permite bloquear la sesión del usuario root de tal manera que para volver a abrir su sesión se debe escribir la contraseña de este usuario.

Para ejecutarlo, se puede pulsar sobre el icono de esta aplicación, o escribir xlock en la consola de comandos. Se ha instalado esta herramienta porque para el resto de usuarios existe un protector de pantallas pero que no pide contraseña.

Para instalarlo simplemente se debe escribir en una consola de comandos: *apt-get install xlockmore xautolock. Fig.J.1*

Fig.J.1 Icono Xlock.

K. OSI

La Organización Internacional para la Estandarización (ISO), en 1984 estableció la pila OSI (Open Systems Interconnection - Interconexión de Sistemas Abiertos). Esta pila OSI divide las comunicaciones de red en siete niveles, y consiste en una serie de reglas para que se puedan establecer la comunicación entre redes (*Fig.K.1)*, ya que hasta el momento de su creación cada red utilizaba sus propios protocolos, y la comunicación entre distintas redes no era posible.

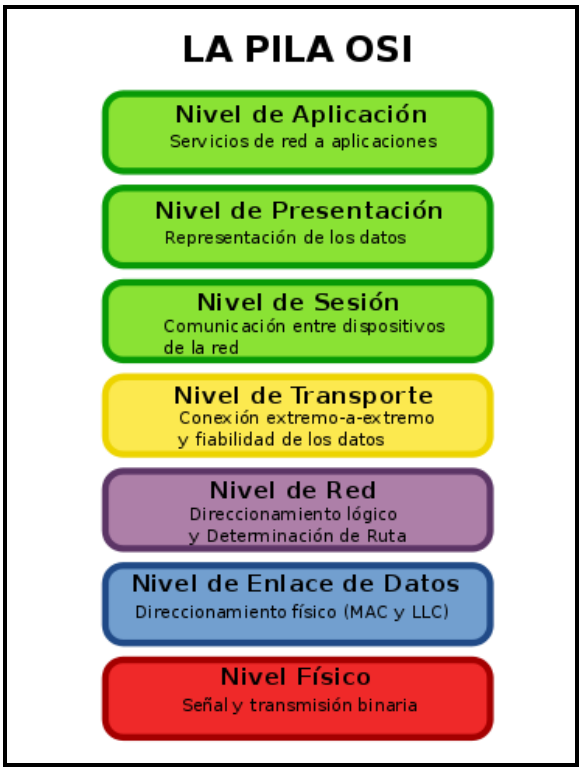

Fig.K.1 Capas del modelo OSI.

L. Bibliografía.

[1] Altadill Izura P.X, *IPTABLES Manual práctico,* Altadill Izura P.X, País Vasco, 2006, http://www.pello.info.

[2] Feilner M., *OpenVPN Building and Integrating Virtual Private Networks,* Packt Publishing Ltd, Birmingham, 2006, http://www.pdf-search-engine.com/openvpn-building-and-integrating-virtual-private-networks-pdf.html

[3] Tomás Canovas J.J., *Servicio VPN de acceso remoto basado en SSL mediante OpenVPN,* Tomás Canovas J.J, Cartagena, 2008, http://repositorio.bib.upct.es/dspace/handle/10317/758

[4] Evolución F., *Instalación del Servidor SAMBA,* Evolución F, México, 2004, http://www.linuxparatodos.net.

[5] Eckstein R., Collier–Brown D., Kelly P., *Usando Samba,* Eckstein R., Collier–Brown D., Kelly P, 1999, http://es.tldp.org/Manuales-LuCAS/USANDO-SAMBA/usando-samba.pdf.

[6] Morales D., *OpenVPN y Samba sobre openSUSE 10.3,* Morales D, Chile, 2007,http://metalklesk.blogspot.com/2007/12/openvpn-y-samba-sobre-opensuse -103.html.

[7] Merlos J.M., Configuración de una red privada virtual con OpenVPN usando Ethernet Bridging, Merlos J.M, 2007, http://www.merlos.org/documentos/linux/28-configuracion-de-una-red-privada-virtual-con-openvpn.html.

[8] Armando Medina J., *Como OpenVPN en Slackware Linux,* Armando Medina J., 2006, http://tuxjm.net/docs/openvpn-como4slack/index.html.

[9] Vialfa C., *Instalación de un servidor Samba,* Vialfa C., 2008, http://es.kioskea.net/faq/sujet-1155-instalacion-de-un-servidor-samba.

[10] Anónimo, ¿Qué es samba?, 2008, http://tecnoloxiaxa.blogspot.com/2008/11/instalar-y-configurar-samba-en-ubuntu-o.html

[11] mced, *RealVNC*, mced, 2008, http://www.adslayuda.com/servidores-real_vnc.html

[12] Anónimo, *SUSE: Compartiendo con SAMBA, h*ttp://blog.wunslov.com/2005/suse-compartiendo-con-samba/

[13] Fábrega Martínez, P.P., *Supuesto práctico de Apache, Samba, DNS y DHCP,* Fábrega Martínez, P.P*,* http://dns.bdat.net/documentos/entorno_publicacion_web/

[14] Vaughan E.,, *How To Samba With openSuse 10.3 And Windows XP,* http://www.tweakhound.com/linux/samba/page_1.htm

[15] Anónimo, *Configurar un cliente samba en Ubuntu O Debian,* 2008, http://tecnoloxiaxa.blogspot.com/2008/11/configurar-un-cliente-samba-en-ubuntu-o.html

[16] Ximo A., *Com modificar el permisos de la carpeta samba,* Ximo A., 2006, http://lliurex.net/home/va/node/1048

[17] Pérez Estévez E., *Conexión Host a Host con OpenVPN,* Perez Estevez E*., Ecuador, 2007,* http://www.ecualug.org/2007/02/06/comos/1_conexion_host_host_con_openvpn

[18] Cabrales A., *VNC e IPTABLES en suse11*, Cabrales A., México, 2009, http://www.ecualug.org/2009/02/16/forums/vnc_e_iptables_en_suse11

[19] Anónimo, *SuSEfirewall2*, 2009, http://en.opensuse.org/SuSEfirewall2.