IMPLEMENTACIÓN DE UNA VPN PARA LA INTERCONEXIÓN DE SEDES PRINCIPALES Y REMOTAS MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE

FABIAN LEONARDO GUERRA PLATA
FREDY MARTIN ARCHILA BARAJAS

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERIAS ELECTRICA, ELECTRONICA Y TELECOMUNICACIONES
ESPECIALIZACIÓN EN TELECOMUNICACIONES
BUCARAMANGA
2013

IMPLEMENTACIÓN DE UNA VPN PARA LA INTERCONEXIÓN DE SEDES PRINCIPALES Y REMOTAS MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE

FABIAN LEONARDO GUERRA PLATA
FREDY MARTIN ARCHILA BARAJAS

Monografía presentada como requisito final para optar por el título de:
ESPECIALISTA EN TELECOMUNICACIONES

Director:
Ingeniero Fredy Alfonso Beltrán Miranda.

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERIAS ELECTRICA, ELECTRONICA Y TELECOMUNICACIONES
ESPECIALIZACIÓN EN TELECOMUNICACIONES
BUCARAMANGA
2013

Dedico este trabajo, en primer lugar a Dios, porque en sus manos están puestos todos mis esfuerzos para lograr alcanzar nuestras metas, y a mis padres y hermanos, quienes con su apoyo, dedicación, desprendimiento y cuidado, han logrado impulsar mi crecimiento profesional y han hecho de mí el ser humano que soy hoy en día, que Dios lo siga bendiciendo.

Fredy

Dedico este trabajo a mi madre, a quien todo le debo; quien siempre me apoya en todos mis proyectos y me brinda su amor incondicional. A las personas que de una u otra forma han estado pendientes de esta etapa de mi vida. A mi alma mater, de la cual me siento orgulloso. A Jairito por ser un gran ejemplo de vida.

Fabián

AGRADECIMIENTOS

Los autores expresan su agradecimiento a:

Ing. FREDDY ALFONSO BELTRAN MIRANDA, director de este proyecto.

UNIVERSIDAD INDUSTRIAL DE SANTANDER, por brindarnos la oportunidad de ser parte de esa gran familia UIS, y por su apoyo a nuestra formación como mejores seres humanos y profesionales.

NUESTROS COMPAÑEROS DE ESTUDIO Y AMIGOS, por el apoyo brindado durante este proceso y por ofrecernos su amistad y apoyo incondicional.

CONTENIDO

pág.

LISTA DE TABLAS

pág.

LISTA DE FIGURAS

pág.

LISTA DE ANEXOS

pág.

RESUMEN

TITULO: IMPLEMENTACIÓN DE UNA VPN PARA LA INTERCONEXIÓN DE SEDES PRINCIPALES Y REMOTAS MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE *

AUTORES: FABIÁN LEONARDO GUERRA PLATA, FREDY MARTIN ARCHILA **

PALABRAS CLAVE: VPN, virtualización, Software libre, túneles, implementación, openvpn, protocolos.

En este trabajo se presenta la implementación de una red privada virtual (VPN) de bajo costo, basada en software libre y con el aprovechamiento de la tecnología de virtualización, para incorporar algunos servicios que podrían resultar de utilidad a la empresa. Esta implementación puede servir de solución para las pequeñas y medianas empresas, que estén interesadas en incorporar este tipo de tecnología y que no cuenten con los recursos económicos suficientes para adquirir todo el equipamiento necesario para montar su propia red privada virtual.

En el trabajo también se exponen los temas que soportan el conocimiento teórico que hay detrás de esta implementación, como los son: virtualización, redes privadas virtuales, protocolos, etc., igualmente se presentan algunas referencias de los posibles elementos de software que se utilizarán como base para la implementación y cuales, según el objetivo del proyecto, serían los más adecuados para llevarla a cabo.

La elección de los diferentes elementos que sustentan el proyecto no es una camisa de fuerza, por lo que esta implementación puede adaptarse a diversos sistemas operativos y utilizar la herramienta de virtualización que se prefiera, probablemente lo único que se deba mantener es el software VPN. Por último se describirá la implementación de la red privada virtual, especificando paso a paso como se debe realizar el montaje.

---

* PROYECTO DE GRADO

** ESCUELA DE INGENIERIAS ELECTRICA, ELECTRONICA Y TELECOMUNICACIONES. DIRECTOR: Ing. FREDDY ALFONSO BELTRAN MIRANDA.

ABSTRACT

TITLE: IMPLEMENTATION OF A VPN INTERFACE FOR MAIN AND REMOTE OFFICES INTERCONNECTION THROUGH THE USE OF FREE SOFTWARE *

AUTHORS: FABIÁN LEONARDO GUERRA PLATA, FREDY MARTIN ARCHILA **

KEYWORDS: VPN, Virtualization, Free Software, tunnels, implementation, openvpn, protocols.

This work introduces a low cost virtual private network implementation (VPN), based on free software and the use of virtualization technology to incorporate some services that could be useful for many companies. This implementation can serve as a solution for small and medium enterprises, which are interested in incorporating this technology and have no sufficient financial resources to purchase the necessary equipment to set up your own virtual private network.

The document expose topics that support the theoretical knowledge behind this implementation, as are: Virtualization, virtual private networks , protocols, etc. , also some references of possible elements of software to be use as basis for implementation and which , according to the objective of the project would be appropriate to carry it out .

The choice of the different elements that support the project is not a “straitjacket”, so this implementation can be adapted to different operating systems and using virtualization tool you prefer, probably the only thing that should be maintained is the VPN software. Finally the implementation of the virtual private network is described by specifying step by step how it must be mounting.

---

* GRADUATION PROJECT

** SCHOOL OF ELECTRONIC, ELECTRICAL AND TELECOMUNICATIONS ENGENEERING. Project Director: ING. FREDDY ALFONSO BELTRAN MIRANDA.

INTRODUCCIÓN

Cuando muchas empresas empezaron a tener éxito en el mercado y debieron empezar a expandir sus áreas de influencia, para así poder obtener mejores resultados en la distribución de sus productos o servicios, crearon sucursales en cada uno de estos lugares, las cuales, para poderse comunicar usaban el servicio telefónico. Luego a medida que la tecnología iba evolucionando, cada una de estas filiales comenzó a equipar sus edificios con aparatos que les permitieran ser más eficientes y obtener un mejor desempeño en sus labores, es así como estas empresas introdujeron los computadores; después, para dar mayor eficiencia en su trabajo, cada una de estas sedes comenzó a interconectar sus ordenadores dando como resultado final que cada sucursal contaba con su propia red LAN, la cual podía o no ser compatible con las demás redes LAN de las otras sedes de la empresa, ya que cada una de ellas operaba de forma aislada; es precisamente esta independencia de las sucursales lo que generaba un inconveniente en la comunicación de la organización, ya que esto limitaba el control sobre la información que se podía manejar entre la sede principal y la remota. Posteriormente las compañías, para superar dicha dificultad, decidieron interconectar cada una de las redes LAN de las filiales con la sede principal, para compartir los recursos internos de la empresa y obtener un mayor control sobre la información, utilizando para ello el único medio disponible, las líneas telefónicas; sin embargo la utilización de este servicio suponía varios problemas entre ellos el costo y la escalabilidad, ya que dependiendo de la ubicación de una sede, la conexión con esta podía resultar muy costosa, además si la compañía se expandía, eso representaba la necesidad de conectar una sucursal más, traduciéndose esto en un aumento de los costos. Finalmente cuando la era de las comunicaciones llegó y el Internet entro en auge, se dio un adelanto importante, que le permitía a estas organizaciones poder interconectar sus distintas sucursales con costes mucho más económicos y con posibilidades de crecimiento mucho mayores a las ofrecidas por las líneas telefónicas, a este desarrollo se le denomino VPN, por sus siglas en inglés (Virtual Private Network) o lo que es lo mismo Red Privada Virtual.

Lo que esta nueva tecnología ofrece, es la posibilidad de conectar dos o más redes utilizando una red pública (Internet). En realidad no podemos afirmar cuando ni donde se implementó la primera VPN, pero si diremos que la invención del Internet y toda la historia que rodea a este desarrollo han posibilitado la creación de estas soluciones.

Durante el transcurso del documento se hará referencia a varios temas que sirven como soporte teórico para el desarrollo de este proyecto; se tratarán temas como el de software libre, virtualización y por supuesto el de redes privadas virtuales y su funcionamiento, entre algunos otros.

Llegado el momento de la implementación, se observará que no es tan complicado, ni tan costoso realizar el montaje de una VPN como la que aquí se plantea, ya que lo que se necesita para llevar a cabo esto no es más que el software vpn, un computador de escritorio o portátil con características de rendimiento óptimo y una conexión a internet, nada más. La conexión a internet utilizada para este trabajo es proporcionada por un modem USB con tecnología 3.5G, debido a que en el lugar donde se desarrolla la implementación no se posee ningún otro tipo de acceso. Sin embargo esto no debe considerarse un impedimento, más bien puede tomarse como una ventaja ya que demuestran la flexibilidad y adaptabilidad de este tipo de soluciones.

Como se mencionó anteriormente, el desarrollo de este trabajo utiliza software libre, lo que plantea algunos retos para las personas particulares o las empresas que decidan montar una solución similar, estos desafíos se mencionaran más adelante en un apartado destinado a este tema.

Por último cabe aclarar que los contenidos tratados en este texto no serán tan detallados, ya que por la orientación del trabajo la pormenorización de estos hechos queda fuera de su alcance, claro está, a menos que se trate de temas relevantes para el desarrollo de la implementación, ya que la intención principal de este documento es guiar al lector en como poner en funcionamiento una VPN, para lo cual se detallará paso a paso y recurriendo a imágenes el modo en que se debe realizar este procedimiento.

1. ANTECEDENTES

1.1 PROBLEMÁTICA

Las grandes empresas implementan sus redes privadas virtuales adquiriendo equipos dedicados, routers VPN o firewalls VPN, que satisfacen los requerimientos que estas organizaciones necesitan, sin embargo estos equipos suelen ser muy costosos y por tanto algunas pequeñas y medianas empresas que requieren o desean acceder a esta tecnología no lo pueden hacer. Aunque hoy en día existen algunos dispositivos más económicos para el desarrollo de estas VPN's enfocados a PYMES, igualmente hay algunas de estas empresas que, por x o y motivo, lo piensan antes de invertir su dinero en alguno de estos dispositivos. Es a esta problemática que este trabajo pretende dar una alternativa de solución, ya que plantea la implementación de redes privadas virtuales basadas en software, para lo cual solo se necesita un computador normal, con algunas características de rendimiento óptimas, el software VPN y una conexión a internet.

1.2 JUSTIFICACIÓN

Este trabajo pretende ofrecer una alternativa, para aquellas empresas que por motivos económicos o de otra índole, aún no han decidido implementar una solución de este tipo.Igualmente procura ser una fuente de información para que aquellas empresas o personas que tienen poco o ningún conocimiento acerca de redes privadas virtuales, conozcan lo que son, para lo que pueden servir y si fuese el caso implementen la solución que aquí se expone;y para aquellas organizaciones que ya han montado una VPN, presentarles una propuesta que probablemente les resulte llamativa e interesante de aplicar.

Así mismo, este trabajo pretende mostrar las bondades del software libre, ya que gran parte de la implementación aquí propuesta, utiliza este tipo de aplicaciones para su desarrollo; más específicamente se trabaja sobre un sistema operativo GNU/Linux, con lo cual se desea ilustrar el manejo de este tipo de softwarepara aquellos que poco conocimiento tienen acerca de este tema.

1.3 OBJETIVOS

General:

Implementar una VPN basada en software libre, que represente una alternativa real de solución a las comunicaciones en las PyMES.

Especificos:

- Explicar conceptos que le permitan a una empresa o persona tomar la decisión de si está solución es adecuada o no para sus necesidades.

- Utilizar herramientas de hardware y software que permitan reducir los costos del montaje de este tipo de soluciones.

- Desarrollar el montaje de tal manera que se explique con claridad el proceso de implementación de este tipo de VPN's.

1.4 HERRAMIENTAS A UTILIZAR

En este apartado se describirán muy rápidamente algunas de las herramientas que se utilizarán para el desarrollo de este trabajo.

- *Sistema operativo*: Como ya se ha hecho mención, para el desarrollo de este trabajo se utilizara un sistema operativo basado en GNU/Linux[1]; este tipo de sistemas han estado ganando gran popularidad en los últimos años debido a que, es muy similar a Unix, es software libre, posee gran robustez, confiabilidad, seguridad y libertad para modificar el código.

- *Máquinas virtuales*: Es un tipo de software que se instala sobre un sistema operativo, llamado anfitrión, y que nos permite virtualizar otro sistema operativo, denominado huésped.

- *Software VPN*: Es un tipo de aplicación que permite crear redes privadas virtuales, a través de técnicas de entunelamiento y con el uso de protocolos específicos para este fin.

Todos estos temas serán ampliados un poco más en el siguiente capítulo.

[1] Es el nombre utilizado para designar a las herramientas del sistema operativo GNU, que utilizan como base para su funcionamiento el núcleo Linux.

2. BASE TEÓRICA

2.1 SOFTWARE LIBRE[(1),(2)]

Según la Free Software Foundation[2], "*«Software libre» significa que el software respeta la libertad de los usuarios y la comunidad. En términos generales, los usuarios tienen la libertad de copiar, distribuir, estudiar, modificar y mejorar el software*". Esto quiere decir que cualquier persona o empresa que adquiera un software denominado "libre", estará en la libertad de instalarlo en la cantidad de equipos que desee, copiarlo y distribuirlo dentro de su entorno personal o laboral o más allá si lo desea, modificarlo para adaptarlo a las necesidades personales o de su empresa, todo esto sin tener que pedir permiso al o los desarrolladores del software y sin los perjuicios legales que conllevaría hacer algo como esto en un software propietario.

Para que un programa sea denominado "software libre" debe poder permitir a los usuarios cuatro libertades esenciales[3]:

- Libertad 0: la libertad de ejecutar el programa para cualquier propósito.
- Libertad 1: la libertad de estudiar como funciona el programa y cambiarlo para que haga lo que usted quiera.
- Libertad 2: la libertad de redistribuir copias para ayudar a su prójimo.
- Libertad 3: la libertad de distribuir copias de sus versiones modificadas a terceros. Esto le permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones.

Las libertades 1 y 3 necesitan acceso al código fuente, por lo tanto es condición necesaria que un "software libre" venga siempre acompañado de este.

"Libre no significa gratis", libre significa que el usuario está en la libertad de hacer lo que desee con este tipo de software, no que no deba pagar por él, aunque mucho del software libre se puede conseguir gratis, no es una norma. Como se mencionó anteriormente un "software libre" debe permitirle a un usuario las cuatro libertades señaladas, mientras que un software gratuito, es aquel que usted consigue sin pagar nada por él, sin embargo puede que le brinde algunas o ninguna de las libertades ya mencionadas.
Aunque el software es libre, para que pueda cumplir con esta característica, debe enmarcarse dentro de un contexto legal, por lo tanto a este tipo de software también se les crean licencias con el fin de que puedan ser utilizados de la manera en que se desea

[2]Fundación para el software libre. Es una organización creada en octubre de 1985 por Richard Stallman y otros entusiastas del software libre con el propósito de difundir este movimiento.
[3] Tomadas de la página web http://www.gnu.org/philosophy/free-sw.es.html

que se lleve a cabo su uso. Existen varios tipos de licencias para el software libre, aquí mencionaremos algunas[4]:

- *Licencia GNU GPL (Licencia Pública General de GNU)*: es una de las más utilizadas, el autor conserva los derechos de autor (copyright),permite la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia licencia.

- *Licencia AGPL (Licencia Pública General de Affero)*: es una licencia copyleft derivada de la Licencia Pública General de GNU diseñada específicamente para asegurar la cooperación con la comunidad en el caso de software que corra en servidores de red. La Affero GPL es íntegramente una GNU GPL con una cláusula nueva que añade la obligación de distribuir el software si éste se ejecuta para ofrecer servicios a través de una red de ordenadores.

- *Licencias estilo BSD*: Llamadas así porque se utilizan en gran cantidad de software distribuido junto a los sistemas operativos BSD. El autor, bajo tales licencias, mantiene la protección de copyright únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, incluso si dichos trabajos tienen propietario. Son muy permisivas, tanto que son fácilmente absorbidas al ser mezcladas con la licencia GNU GPL con quienes son compatibles.

- *Licencias estilo MPL (licencia Pública de Mozilla)*: Se utilizan en gran cantidad de productos de software libre de uso cotidiano en todo tipo de sistemas operativos. Fue la primera licencia nueva después de muchos años, que se encargaba de algunos puntos que no fueron tenidos en cuenta por las licencias BSD y GNU. En el espectro de las licencias de software libre se la puede considerar adyacente a la licencia estilo BSD, pero perfeccionada.

- *Copyleft*: es un método general para hacer que un trabajo específico sea libre, y requiere que todas las modificaciones y versiones extendidas del programa sean libres también. La efectividad de ejercerlo puede depender de la legislación particular de cada país, pero en principio se puede utilizar para programas informáticos, obras de arte, cultura, ciencia, o cualquier tipo de obra o trabajo creativo que sea regido por el derecho de autor.

Algunos de los retos a los que se enfrenta la gente al utilizar software libre son: cambiar su forma de pensar, pasar de algo que ya conocen y manejan bien, a algo desconocido y probablemente difícil de aprender; elevar su conocimiento informático, pues mucho del software libre exige un conocimiento intermedio o avanzado sobre este tema y más al momento de resolver problemas.

Para terminar con este tema vamos a hacer una comparación entre el software libre y el software propietario, con el fin de que el lector tenga en cuenta estos aspectos al

[4] Tomadas de la página web http://es.wikipedia.org/wiki/Software_libre, sección tipos de licencias.

momento de querer implementar una solución como la que aquí se plantea y pueda ver las ventajas y desventajas de utilizar uno u otro tipo de software.

Tabla 1. Software libre versus software propietario.

Software libre	Software propietario
– El usuario puede utilizar el software de la forma que lo desee. – El usuario puede copiar, distribuir y modificar el software. – Bajo costo o gratuito. – Poca o ninguna restricción legal para su uso por lo tanto inexistencia de piratería. – Menos propenso a ataques informáticos. – Limitada compatibilidad con el hardware. – Garantía inexistente por parte del desarrollador. – El soporte depende de si el problema ya ha sido resuelto por alguien, de lo contrario puede tornarse difícil encontrar una solución. – Los conocimientos informáticos deben ser de grado intermedio o avanzado (por lo anteriormente expuesto) si desea sacarse el mayor provecho de este software. – La administración del sistema requiere de profesionales capacitados. – Interfaces graficas menos amigables. – Actualizaciones periódicas.	– El usuario solo pude usar el software para lo que el propietario especifique. – El software está restringido para su copia y distribución y no puede modificarse. – Alto costo, algunos son gratis. – Legalmente muy protegido y por lo mismo muy pirateado. – Vulnerable a ataques informáticos. – Mayor compatibilidad con el hardware. – Garantía limitada a fallos en el software o a compatibilidades de hardware. – Soporte exclusivo por parte del propietario. – Se deben poseer ciertos conocimientos informáticos para su uso, por lo general conseguidos mediante cursos. – La administración de los sistemas requiere profesionales capacitados pero resulta un poco más sencilla que con el software libre. – Interfaces graficas más amigables y mejor acabadas. – Actualizaciones cada cierto tiempo o hasta cuando se descubra alguna vulnerabilidad.

2.2 VIRTUALIZACIÓN[(3),(4),(5)]

La virtualización es la técnica mediante la cual se logra crear una versión virtual de algún elemento de hardware, software o red. Esto se consigue utilizando un software específico para este fin llamado Hypervisor o VMM (del inglés Virtual Machine Monitor) que lo que hace es establecer una capa lógica que permite abstraer los recursos de una computadora y traspasarlos a una interfaz externa a la que llamamos Máquina Virtual. La VMM permite manejar y gestionar los cuatro recursos básicos de una computadora (CPU, memoria, almacenamiento y conexiones de red).

2.2.1 Tipos de hypervisores

- *Hypervisores tipo 1*: También llamado nativo o unhosted, es software que se ejecuta directamente sobre el hardware del equipo y se encarga de controlar este hardware y monitorear los sistemas operativos virtualizados instalados sobre este VMM.

- *Hypervisor tipo 2*: También denominados hosted, es una aplicación que se ejecuta sobre un sistema operativo para poder virtualizar otros sistemas operativos, lo que hace que el rendimiento de estos sistemas virtualizados sea menor en comparación con los virtualizados en los Hypervisores tipo 1.

En la siguiente imagen se presenta un modelo donde se visualizan estos dos tipos de hypervisores.

Figura 1. Tipos de hypervisores o VMM.

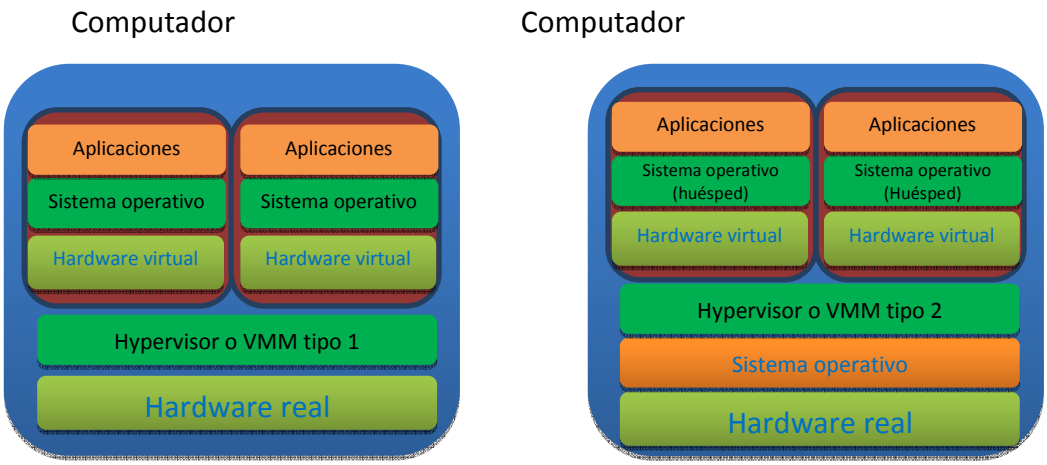


Imagen basada en la figura de http://blog.0balla.net/2012/04/introduccion-a-la-virtualizacion

2.2.1.1 Tipos de virtualización

Existen varios métodos de virtualización, los cuales se pueden catalogar en uno de estos dos tipos:

- *Virtualización de plataforma*: En esta técnica se trata de simular un computador real, ya sea este un servidor o PC, con todos o casi todos sus componentes (ya que depende de si el hardware de la maquina se puede o no virtualizar) y prestarle todos los recursos necesarios para su funcionamiento. Por tanto, existe un software anfitrión (Hypervisor) el cual se encarga de controlar y administrar los recursos de la maquina física y transferirlos de forma tal que todas las máquinas virtuales que corran sobre dicho equipo funcionen adecuadamente. Es dentro de este esquema que se encuentran la mayoría de los métodos de virtualización conocidos, incluida la virtualización de sistemas operativos y de aplicaciones.

- *Virtualización de recursos*: esta técnica permite agrupar varios dispositivos para que sean vistos como uno solo o viceversa, haciendo que un medio sea dividido en múltiples recursos independientes. Generalmente este método se aplica a medios de almacenamiento, es muy utilizado en soluciones

SANS, pero también se aplica a otros recursos como los de red, en donde encontramos las VLAN (Virtual Local Area Network) las cuales se rigen por el estándar 802.1Q de la IEEE; y las VPN (Virtual Private Network), las cuales no tienen un estándar definido.

Esta tecnología ha empezado a cobrar cada vez más fuerza en las soluciones informáticas de hoy en día, aunque su desarrollo y utilización viene desde hace mucho tiempo (aproximadamente desde la década de los 60)es ahora cuando ha aumentado su popularidad debido a que, para todos los tipos de empresas e incluso para las personas del común, su utilización trae más ventajas que desventaja, veamos algunas de estas:

2.2.2 Ventajas y desventajas

2.2.2.1 Ventajas

- *Ahorro de costos*: Es una de las mayores ventajas ya que se reduce el número de equipos físicos necesarios, lo que permite ahorro de espacio, así como del costo de su administración y el de su mantenimiento.

- *Flexibilidad de crecimiento*: Instalar nuevos servidores es mucho más rápido y sencillo que hacerlo con un servidor físico.

- *Administración simplificada*: Debido a que los servidores se encuentran en un solo equipo físico, la tarea de administrar cada uno se vuelve más sencilla.

- *Mejor aprovechamiento de aplicaciones y equipos antiguos*: En nuestras empresas u hogares poseemos equipos y aplicaciones que no son compatibles con las nuevas tecnologías y que no actualizamos debido al alto costo que esto representaría. Sin embargo la virtualización nos permite seguir utilizando estos recursos, mientras actualizamos los equipos necesarios.

- *Mejor gestión de recursos*: Si el equipo anfitrión pierde rendimiento, solo deberemos adquirir hardware para repotenciar dicho equipo, con lo cual este aumento de rendimiento será transferido a todas las máquinas virtuales instaladas en este. De igual manera los recursos del equipo anfitrión son mejor aprovechados y no son tan desperdiciados.

2.2.2.2 Desventajas

- *Rendimiento inferior*: Un sistema virtualizado no alcanzará el mismo rendimiento que si estuviera instalado en forma nativa.

- *Hardware no soportado*: La utilización de un hardware está condicionada a si el hypervisor lo soporta o no.

- *Fallos en el equipo anfitrión*: Esta es quizás una de las desventajas mas críticas, dado que si el equipo anfitrión falla (ya sea por hardware o por software) también fallarán todas las máquinas virtuales instaladas en este. Por lo tanto se hace necesario tener medidas de respaldo.

- *Abuso de máquinas virtuales*: Debido a que ya no se hace necesario comprar equipos, se empieza a exagerar en el uso de esta tecnología, por lo cual se hace necesario dimensionar muy bien los equipos para saber que cantidad de máquinas virtuales puede soportar el computador anfitrión.

En cuanto a este trabajo se refiere, se va a utilizar la virtualización de plataforma, para simular una red LAN, a través de la utilización de máquinas virtuales; además se utilizara la virtualización de recursos, ya que será indispensable para la implementación de la VPN.

2.3 REDES PRIVADAS VIRTUALES (VPN's)[(6), (7), (8)]

Para entender lo que es una VPN primero recordemos un poco de historia en cuanto a redes.

En informática y comunicaciones se ha hablado de tres tipos de redes: redes LAN, MAN y WAN, y se denominan de una u otra forma dependiendo de la extensión geográfica que abarque dicha red, por ejemplo: la red de un edificio es considerada LAN; una red que interconecta dos o más edificaciones (redes LAN) entre municipios cercanos es denominada MAN (aunque algunos prefieren acuñar el término de LAN extendidas) y una red que enlace varias redes MAN a nivel nacional o internacional es llamada WAN. Ahora bien, otro concepto que se enlaza con estas definiciones es el de privado o público; una red LAN, MAN o WAN es de carácter público si cualquiera puede acceder a todos los recursos de esta con total libertad. Ya que todas las organizaciones, sin importar si son públicas o privadas, manejan información confidencial que no quieren que sea conocida más allá de sus propias empresas, podemos decir que todas las redes LAN, MAN y WAN corporativas son privadas, aunque crear y mantener una red WAN privada es muy costoso y pocas empresas lo hacen.

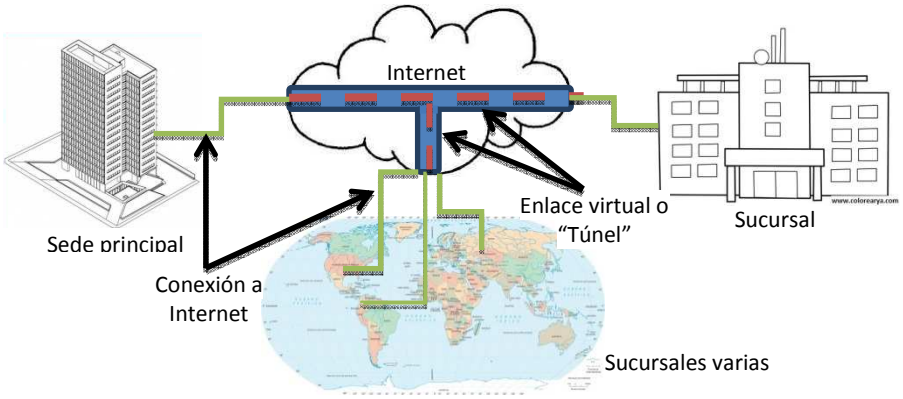
Remontémonos ahora a la década de los 80, cuando empezó el auge de los computadores, muchas empresas compraron estas máquinas para optimizar su procesos y adquirieron, además, tecnología para interconectar cada uno de estos computadores, creando así las primeras redes LAN. Es más, muchas de estas compañías crearon redes LAN para cada una de sus sucursales, las cuales eran independientes entre sí. Cuando surgió la necesidad de interconectar dichas sucursales (para crear así redes MAN), las empresas contaban con uno de estos dos medios: tender por su cuenta un cable físico (Coaxial, Par trenzado o de fibra óptica) desde la sede principal hasta cada una de las filiales, con los consiguientes costos de tendido (dependiendo del tipo de cable) y mantenimiento de los mismos o, contratar con la compañía de teléfonos para

interconectarlas por dicho medio, ya que esta red pública era una de las más extendidas a nivel nacional e internacional (por lo que se considera una red WAN). Si bien a muchas empresas les resultaba más económico tender su propio cable para enlazar sus sucursales en la misma ciudad o zonas cercanas, no lo era así para conectar con sus filiales más alejadas, por tanto contrataban con la empresa de teléfonos para arrendar un enlace privado y de esta forma interconectar dichas sedes (creando redes WAN).

Para esa época existían solo dos tecnologías para crear interconexiones WAN, los enlaces dedicados o los enlaces conmutados. Dentro de los enlaces dedicados se encontraban topologías como la Clear Channel, Frame Relay y ATM, dentro de los enlaces conmutados se contaba con los análogos y los digitales. Muchos de estos servicios eran ofrecidos por las compañías de teléfonos, ya que su red se encontraba por casi todo el mundo y al ser empresas públicas tenían los recursos económicos para llevar a cabo estos enlaces. Sin embargo para las empresas que adquirían estos servicios el costo podía llegar a ser abrumador, dependiendo de la distancia y duración de la conexión con la otra sede.

A mediados de los años 90 y en particular a principios del siglo 21, el auge de Internet comenzó a cambiar muchas de las formas en que nos comunicamos, permitiendo el desarrollo de nuevas tecnologías en este campo, una de ellas ha sido las VPN's, las cuales han incursionado con fuerza en las empresas, ya que les han permitido reducir el costo de sus conexiones MAN o WAN en un gran porcentaje, ya que solo tienen que pagar por el acceso a Internet, además su funcionamiento es en base a protocolos de red, como IP, sin importar la tecnología WAN que lo soporte. A continuación se presenta una imagen que representa lo que es una red privada virtual.

Figura 2. Red Privada Virtual (VPN).



Dicho lo anterior podemos decir que una VPN es un enlace virtual, que asemeja el tendido de un cable físico, para conectar dos o más sedes, y que utiliza la red pública de internet para llevar a cabo esta acción, haciendo que estas filiales se comporten como si estuvieran

dentro de una misma red LAN y de esta forma poder disfrutar de los servicios y ventajas que ofrecen las redes privadas.

2.3.1 Ventajas y desventajas [(6), (7), (8)]

2.3.1.1 Ventajas

Costo: es quizás el principal motivo del uso y masificación de esta tecnología. El porcentaje de ahorro de las empresas en costos de comunicación puede llegar a ser considerable, todo esto debido a que se utiliza internet como medio de comunicación.

- *Conexiones remotas*: Es para lo que se han diseñado, para conectar remotamente, ya sea sucursales o empleados individuales, brindando la posibilidad a estos de laborar desde su hogar, o desde el lugar donde se encuentren (trabajadores móviles), haciendo que ellos se conecten a la empresa de una forma más segura.

- *Independencia*: La tecnología VPN es independiente, puede implementarse en diversos sistemas operativos como Unix, Linux, Windows, OS X, etc. Igualmente puede utilizarse sobre líneas rentadas, enlaces Frame Relay, ATM, RDSI, T1 o Wireless.

- *Flexibilidad y escalabilidad*: es un punto a favor muy fuerte, ya que si se desea agregar otro sitio, solo basta con comprar equipos y contratar un acceso a internet.

- *Seguridad*: Las VPN's ofrecen variedad de elementos para asegurar la información y previenen contra algunos tipos de ataques conocidos.

2.3.1.2 Desventajas

- *Uso de medios externos*: las VPN's se transmiten sobre recursos externos a los de la empresa (por lo general se utiliza Internet), por lo que esta no tienen el control sobre estos medios.

- *Personal calificado en seguridad de redes*: esto no se refiere a que no exista personal calificado en este aspecto, sino a que las empresas deben contratar este tipo de personal para garantizar que las VPN's no se conviertan en un foco de inseguridad al interior de las compañías. Esto por lo general no es inconveniente en las grandes organizaciones ya que estas poseen este tipo de personal, sin embargo para las pequeñas empresas es un costo que se debe tener en cuenta.

- *Estandarización*: debido a que existe una gran variedad de equipos para implementar VPN´s y que esta es una tecnología no estandarizada, sino que trabaja sobre topologías y protocolos ya existentes, pueden encontrarse incompatibilidades al momento de implementar este método.

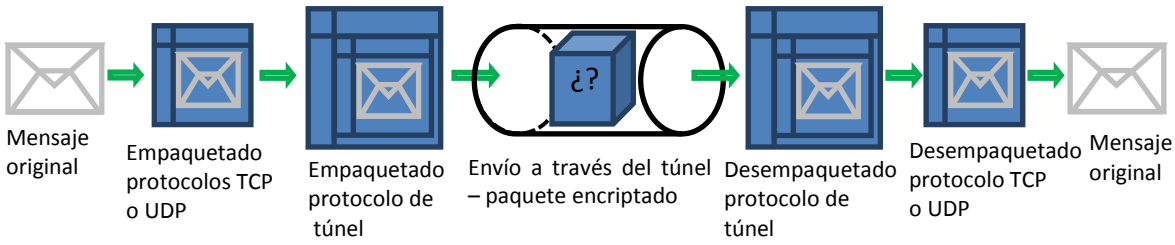
- *Equipos potentes*: debido a que dentro de la VPN continuamente se están cifrando, descifrando o encapsulando o desencapsulando datos, los equipos que soportan las VPN's deben ser potentes, de lo contrario se podrían ocasionar cuellos de botella en esta red virtual.

2.3.2 Funcionamiento de una VPN [(6), (7), (8)]

Como se ha venido diciendo, una VPN funciona, por lo general, utilizando una infraestructura de red pública, comúnmente Internet, sobre la cual se crea una conexión que permite la comunicación entre dos puntos, a este enlace se le denomina "túnel" (ver figura 2-2).

Como la conexión se hace a través de Internet, que es un medio inseguro, los túneles son un modo de transferir los datos de forma segura a través de esta red pública. La forma en que se hace esto es encapsulando el paquete de datos original dentro del paquete de datos de algún protocolo propio para esta tarea. Al llegar al destino, el paquete original es desempaquetado volviendo así a su estado original. En el traslado a través de Internet, los paquetes viajan encriptados. El túnel es por tanto el canal virtual que se crea cuando se establece la conexión entre dos puntos remotos que se encuentran en diferentes redes con el fin de garantizar la seguridad de los datos enviados, desaparece cuando se corta la comunicación entre estos .La técnica de tunelizar se suele utilizar para trasportar un protocolo determinado a través de una red que, en condiciones normales, no lo aceptaría.

Figura 3. Envío de datos a través de una VPN.



La información que viaja dentro de una VPN debe ser segura, por lo tanto al crear una red privada virtual debe garantizarse la autenticación, integridad, confidencialidad y no repudio de la comunicación, veamos como se logra esto:

- *Autenticación*: Para autenticar la comunicación existen varios métodos que se pueden implementar en una VPN como lo son claves precompartidas, emisión de certificados, autenticación centralizada. La autenticación se realiza al inicio de la sesión y luego aleatoriamente durante la misma, esto con el fin de evitar intrusiones.

- *Integridad*: es garantizar que los datos enviados durante la comunicación no han sido alterados, para ello se emplean funciones tipo HASH como lo son los MD5 (del inglés MessageDigest) y el Secure Hash Algorithm (SHA).

- *Confidencialidad*: es asegurar que los datos que viajan a través de internet, no puedan ser interpretados en caso de ser interceptados; esto se logra utilizando algoritmos de cifrado como DES (Data Encryption Standard), Triple DES (3DES), AES (Advanced Encryption Standard) y DH (Diffie-Hellman).

- *No repudio*: es certificar que el emisor de un mensaje es quien dice ser, y que no se pueda negar este hecho. Para esto se combinan las diversas técnicas vistas antes, como la encriptación y la autenticación.

2.3.3 Proceso de conexión de una Red Privada Virtual

El proceso de conexión de una VPN se describe en los siguientes pasos[5]:

1) Un cliente VPN hace una conexión a un servidor VPN que está conectado a Internet. El Servidor VPN actúa como pasarela y normalmente está configurado para proveer un acceso entero a la red.

2) El servidor VPN responde la llamada virtual.

3) El servidor VPN autentifica la llamada y autoriza la conexión con el cliente.

4) El servidor VPN transfiere la información entre el cliente VPN y la organización.

2.3.4 Protocolos de tunelización [(6), (7)]

Los túneles son la base de las VPN's, para que este se pueda establecer, tanto el cliente del túnel como el servidor del túnel deben utilizar el mismo protocolo de tunelización. Existen varios protocolos comúnmente utilizados para la creación de túneles en VPN's, aquí mencionaremos los más comunes, PPTP, L2TP,L2F, GRE, MPLS, SSH, SSL/TLS e IPsec. PPTP, L2TP, L2F y GRE son protocolos de nivel dos (capa de enlace) mientras que MPLS es un protocolo que trabaja entre las capas dos y tres, IPsec es un protocolo de nivel tres (capa de red), SSL/TLS son protocolos de nivel cuatro (capa de transporte) mientras que SSH es un protocolo de nivel siete (capa de aplicación), estos niveles corresponden al modelo de referencia OSI. Los protocolos de nivel 2 utilizan tramas como su unidad de intercambio mientras que los de nivel 3 utilizan paquetes, los de nivel 4 utilizan segmentos o datagramas (dependiendo si se usa TCP o UDP respectivamente) y los de nivel 7 usan APDU (Unidad de datos de aplicación).

[5] Pasos tomados de la página web http://moodle.unid.edu.mx/dts_cursos_mdl/maestria_en_tecnologias_de_informacion/tem_sel_redes/sesion8/actividades/RPV_I.pdf

A continuación se mencionaran algunas de las características de estos protocolos, sin embargo no se entrara a detallar en profundidad todos los aspectos ya que no se considera necesario para el desarrollo de esta monografía.

2.3.4.1 GRE (Generic Routing Encapsulation) [(7), (8)]

Protocolo originalmente desarrollado por CISCO Systems, por lo cual esta implementado en muchos de sus productos. Diseñado para la implementación de túneles a través de Internet, puede transportar hasta 20 protocolos de nivel de red. Debido a que es un protocolo creado para el establecimiento de túneles, suele acompañarse de otros estándares de autenticación y cifrado para brindar la seguridad necesaria. Este protocolo se implementa comúnmente junto con otros protocolos como PPTP e IPsec. Posee como características principales las siguientes:

- Permite emplear protocolos de encaminamiento especializados que obtengan el camino óptimo entre los extremos de la comunicación.
- Soporta la secuencialidad de paquetes y la creación de túneles sobre redes de alta velocidad.
- Permite establecer políticas de encaminamiento y seguridad.

Suele utilizarse con protocolos que no son enrutables, como NetBIOS, o con protocolos enrutables diferentes de IP a través de una red IP. Actualmente el protocolo GRE se utiliza como un mecanismo de transición para la implementación de redes IPv6, es decir se pueden conectar dos redes IPv6 a través de un túnel Ipv4. GRE está definido por los RFC (Request for Comments) 1701, 1702, 2637 y 2784.

2.3.4.2 MPLS (Multi-Protocol Label Switching) [(7), (9), (10)]

MPLS es un estándar IP de transporte de datos desarrollado por la IETF, que trata de proporcionar algunas de las características de las redes orientadas a conexión a las redes no orientadas a conexión. Opera entre las capas 2 y 3 del modelo OSI. Puede ser utilizada para transmitir diferentes tipos de datos, incluyendo voz y paquetes IP.

Las principales aplicaciones de esta tecnología son: en soporte para redes VPN, en ingeniería de tráfico, en servicios que requieren soporte de QoS (Quality of Service), en soluciones multiprotocolo.

En cuanto a la implementación de VPN sobre MPLS, podemos decir que la mayor ventaja de esta tecnología sobre otras es la forma en como se crea dicha VPN. Como se mencionó

anteriormente la base de una VPN es la creación de túneles, pues bien MPLS también crea dicho entorno, pero de manera diferente a la convencional.

Tradicionalmente una VPN está construida sobre tecnología IP, el objetivo de un túnel sobre IP es crear una asociación permanente entre dos extremos, de modo que funcionalmente aparezcan conectados. El problema que plantean estas IP VPN's es que están basadas en un modelo topológico superpuesto sobre la topología física existente, a base de túneles extremos a extremo.

Con una arquitectura MPLS se obvian estos inconvenientes ya que el modelo topológico no se superpone sino que se acopla a la red del proveedor. En el modelo acoplado MPLS, en lugar de conexiones extremo a extremo entre los distintos emplazamientos de una VPN, lo que hay son conexiones IP a un entorno común al que solamente pueden entrar los miembros de la misma VPN.

En el protocolo MPLS la creación de los túneles está a cargo de la función LSP (Label Switched Path).

La diferencia entre los túneles IP convencionales y los "túneles MPLS" (LSPs) está en que éstos se crean dentro de la red, a base de LSPs, y no de extremo a extremo a través de la red.

Algunas de las ventajas de utilizar MPLS en la implementación de VPN's son:

- Escalabilidad y flexibilidad ante el crecimiento de la red.
- Actualización transparente para el usuario.
- Utilización óptima de los recursos de red.
- Diferenciación entre servicios.
- Uso de tecnologías de vanguardia, entre otras.

Sin embargo MPLS no se construyó como una arquitectura segura, por lo que su uso en VPN debe estar acompañado de otras tecnologías que le brinden estas propiedades.

MPLS se posiciona en la actualidad como una tecnología que está desplazando rápidamente a Frame Relay y ATM en el transporte de datos de alta velocidad y voz digital. MPLS está referenciado en las RFC 3031 y 3032.

2.3.4.3 L2F (Layer 2 Forwarding) [(7), (11)]

Es un protocolo diseñado por cisco para establecer túneles desde usuarios remotos hasta sus sedes principales. En esto es similar al protocolo PPTP, sin embargo la principal diferencia con este, es que L2F puede trabajar directamente con otros medios como ATM o Frame Relay ya que no depende del protocolo IP para el establecimiento de túneles y además permite más de una conexión por túnel. L2F, al igual que PPTP, utiliza el protocolo PPP (Poin-to-Point-Protocol) para la autenticación del usuario remoto por lo que implementa consigo el Protocolo de Autentificación por Clave (PAP) y Protocolo de Autentificación de Desafío Mutuo (CHAP), pero también implementa otros sistemas de autenticación como TACACS+ (Terminal Access Controller Access Control System) y RADIUS (Remote Authentication Dial-In User Service).

L2F trabaja con un servicio de enlace llamado Virtual Dial-Up (VDU), que nos permite acceder a la utilización de toda la infraestructura de Internet, no solo para conectar a través de protocolos diferentes al IP sino también cuando las direcciones IP no son reconocidas. El protocolo L2F es capaz de encapsular payloads PPP o payloads SLIP que serán enviados a sus destinos.

Como L2F es un protocolo de Nivel de enlace de datos según el Modelo de Referencia OSI, ofrece a los usuarios la misma flexibilidad que PPTP para manejar protocolos distintos a IP, como IPX o NetBEUI.L2F está referenciado por el RFC 2341.

2.3.4.4 Protocolo PPTP (Point-to-Point Tunneling Protocol) [(6), (7), (12)]

Es un protocolo de comunicaciones desarrollado por un conjunto de empresas (llamado foro PPTP) entre las cuales se encuentran Microsoft, U.S. Robotics, Ascend Communications, 3Com/Primary Access, ECI Telematics, para implementar redes privadas virtuales o VPN.

Esta tecnología que hace posible el PPTP es una extensión del acceso remoto del protocolo PPP. La tecnología PPTP encapsula los paquetes ppp en datagramas IP para su transmisión bajo redes basadas en TCP/IP. Por el momento, PPTP únicamente soporta los protocolos de red IP, IPX, y NetBEUI. El protocolo PPTP específica además una serie de mensajes de control con el fin de establecer, mantener y destruir el túnel PPTP. PPTP utiliza túneles GRE para implementar el túnel de una VPN.

Una característica importante en el uso del PPTP es su soporte para VPN. La mejor parte de esta característica es que soporta VPN’s sobre Public-Switched Telephone Networks (PSTNs) que son los comúnmente llamados accesos telefónicos a redes.

La autenticación remota de clientes PPTP es realizada empleando los mismos métodos de autenticación utilizados por cualquier otro tipo de servidor de acceso remoto (RAS).La

autenticación utilizada para el acceso a los RAS soporta los protocolos CHAP(Challenge Handshake Authentication Protocol), EAP (Extensible Authentication Protocol), MS-CHAP (Microsoft CHAP) y PAP(Password Authentication Protocol). En cuanto a la encriptación de datos, PPTP utiliza el proceso de encriptación de secreto compartido en el cual sólo los extremos de la conexión comparten la clave. Dicha clave es generada empleando el estándar RSA RC-4 (bastante débil para usuarios fuera de Estados Unidos, ya que solo es de 40 bits) a partir del password del usuario.

Dentro de las desventajas de usar este protocolo para la creación de VPN es su seguridad, ya que esta ha sido completamente rota, el fallo de PPTP es causado por errores de diseño en la criptografía en los protocolos handshake LEAP de Cisco y MSCHAP-v2 de Microsoft y por las limitaciones de la longitud de la clave en MPPE (Microsoft Point-to-Point Encryption). Otra desventaja es que solo se permite una conexión sobre un túnel PPTP, además este protocolo suele usar más de un estándar para la autenticación y encriptación, por lo que dos clientes PPTP pueden ser incompatibles entre ellos si encriptan los datos de manera diferente. El protocolo PPTP está referenciado en el RFC 2637, aunque se debe aclarar que no es un estándar aprobado por la IETF.

2.3.4.5 L2TP (Layer 2 TunnelingProtocol) [(6), (7), (13)]

L2TP es el producto de una colaboración entre los miembros del foro PPTP, Cisco, y el Grupo de Tareas de Ingeniería de Internet (IETF). Este protocolo surge como el heredero de L2F y PPTP, ya que fue creado para corregir las deficiencias de estos protocolos y establecerse como un estándar aprobado por el IETF. Se trata de un estándar abierto y disponible en la mayoría de las plataformas.

Como este protocolo está basado en PPTP, utiliza el protocolo PPP para proporcionar acceso telefónico que puede ser dirigido a través de un túnel por Internet hasta un punto determinado. Igualmente incluye los mecanismos de autenticación PAP y CHAP y soporta la utilización de estos protocolos de autenticación, como RADIUS.

En L2TP la creación del túnel no depende de IP y GRE (como si lo es en PPTP), L2TP define su propio protocolo de establecimiento de túneles, basado en L2F, permitiéndole transportar una gran variedad de protocolos y trabajar con otros medios físicos incluyendo X.25, Frame Relay y ATM.

Dentro delas ventajas que se podrían conseguir al implementar una VPN basada en L2TP es que se puede usar certificados de seguridad de clave pública para cifrar los datos y garantizar la autenticidad de los usuarios de la VPN, además ofrece un acceso económico, con soporte multiprotocolo y acceso a redes de área local remotas, por otra parte permite que un único túnel soporte más de una conexión, con lo cual se podrían aplicar ciertas

técnicas de Calidad de servicios (QoS). Sin embargo una de sus mayores desventajas es que no presenta unas características criptográficas especialmente robustas, por lo cual:

- Sólo se realiza la operación de autenticación entre los puntos finales del túnel, pero no para cada uno de los paquetes que viajan por él. Esto puede dar lugar a suplantaciones de identidad en algún punto interior al túnel.

- Sin comprobación de la integridad de cada paquete, sería posible realizar un ataque de denegación del servicio por medio de mensajes falsos de control que den por acabado el túnel L2TP o la conexión PPP subyacente.

- L2TP no cifra en principio el tráfico de datos de usuario, lo cual puede dar problemas cuando sea importante mantener la confidencialidad de los datos.

- A pesar de que la información contenida en los paquetes PPP puede ser cifrada, este protocolo no dispone de mecanismos para generación automática de claves, o refresco automático de claves. Esto puede hacer que alguien que escuche en la red y descubra una única clave tenga acceso a todos los datos transmitidos.

Para poder solventar esta situación L2TP suele implementarse junto con IPsec, para garantizar una autenticación y cifrado más potente.

Un túnel L2TP se crea encapsulando una trama L2TP en un paquete UDP, el cual es encapsulado a su vez en un paquete IP, cuyas direcciones de origen y destino definen los extremos del túnel. Siendo el protocolo de encapsulamiento más externo IP, los protocolos IPsec pueden ser utilizados sobre este paquete, protegiendo así la información que se transporta por el túnel. El protocolo L2TP se encuentra referenciado en el RFC 2661.

2.3.4.6 SSH (SecureSHell) [(7), (14)]

Es un protocolo y a su vez el nombre del programa que lo implementa, sirve para acceder a maquinas remotas a través de una red. Este protocolo fue concebido para trabajar de forma similar a Telnet, pero con la ventaja de que añade seguridad a la conexión ya que hace que la información que viaja por el medio de comunicación vaya cifrada. El protocolo SSH se utiliza con frecuencia para tunelizar tráfico confidencial sobre Internet de una manera segura.

Aunque con este protocolo no se pueden crear VPN's en todo el sentido de la palabra (ya que hacer esto es complicado y no muy recomendado), si logramos crear túneles y de cierta manera una VPN punto a punto (entre dos equipos). Para esto SSH utiliza una

técnica de redirección de puertos (port-forwarding[6]). El mayor inconveniente es que, al contrario que en las VPN, cada aplicación debe ser configurada individualmente para que utilice el túnel, de modo que no podemos garantizar que todo el tráfico circule únicamente a través del mismo.

2.3.4.7 IPsec (InternetProtocolsecurity) [(6), (7), (15), (16)]

Es un estándar que proporciona servicios de seguridad a la capa IP y a todos los protocolos superiores basados en IP (TCP y UDP, entre otros). IPsec es un conjunto de protocolos cuya función es asegurar las comunicaciones sobre el Protocolo de Internet (IP), proporciona confidencialidad, integridad y autenticidad de datagramas IP, combinando tecnologías de clave pública (RSA), algoritmos de cifrado (DES, 3DES, IDEA, Blowfish), algoritmos de hash (MD5, SHA-1) y certificados digitales X509v3.

Dentro de IPsec se distinguen los siguientes componentes:

- Dos protocolos de seguridad; IP Authentication Header (AH) e IP Encapsulating Security Payload (ESP) que proporcionan mecanismos de seguridad para proteger tráfico IP.

- Un protocolo de gestión de claves Internet Key Exchange (IKE) que permite a dos nodos negociar las claves y todos los parámetros necesarios para establecer una conexión AH o ESP.

El protocolo AH es el procedimiento previsto dentro de IPsec para garantizar la integridad y autenticación de los datagramas IP, sin embargo no proporciona ninguna garantía de confidencialidad. Este procedimiento calcula un Hash Message Authentication Code (HMAC) a través de algún algoritmo hash operando sobre una clave secreta, el contenido del paquete IP y las partes inmutables del datagrama. Este proceso restringe la posibilidad de emplear NAT ya que si se modifican las direcciones IP el control de integridad fallaría. AH puede proteger opcionalmente contra ataques de repetición utilizando la técnica de ventana deslizante[7] y descartando paquetes viejos. AH protege la carga útil IP y todos los campos de la cabecera de un datagrama IP excepto los campos que pueden ser alterados en el tránsito. AH opera directamente por encima de IP, utilizando el protocolo IP número 51.

---

[6]Es la acción de redirigir un puerto de red de un nodo de red a otro. Esta técnica puede permitir que un usuario externo tenga acceso a un puerto en una dirección IP privada (dentro de una LAN) desde el exterior vía un router con NAT activado.

[7]Es un mecanismo dirigido al control de flujo de datos que existe entre un emisor y un receptor pertenecientes a una red informática. La ventana deslizante es un dispositivo de control de flujo de tipo software, es decir, el control del flujo se lleva a cabo mediante el intercambio específico de caracteres o tramas de control, con los que el receptor indica al emisor cuál es su estado de disponibilidad para recibir datos.

El protocolo ESP proporciona autenticidad de origen, integridad y protección de confidencialidad de un paquete, para ello específica el modo de cifrar los datos que se desean enviar y cómo este contenido cifrado se incluye en un datagrama IP. Adicionalmente, puede ofrecer los servicios de integridad y autenticación del origen de los datos incorporando un mecanismo similar al de AH. Al contrario que con AH, la cabecera del paquete IP no está protegida por ESP. ESP opera directamente sobre IP a través del protocolo IP número 50.

Para poder entender el funcionamiento del protocolo IKE debemos primero conocer un concepto fundamental en IPsec como lo es el de asociación de seguridad (SA).
Un SA es un canal de comunicación unidireccional que conecta dos nodos, a través del cual fluyen los datagramas protegidos mediante mecanismos criptográficos acordados previamente. Estos nodos intercambian mensajes para determinar los parámetros que configurarán su enlace seguro; es necesario que ambos nodos estén de acuerdo tanto en los algoritmos criptográficos a emplear como en los parámetros de control. Estos parámetros son necesarios para la puesta en marcha de los mecanismos de seguridad que construyen la asociación de seguridad. Al identificar únicamente un canal unidireccional, una conexión IPsec se compone de dos SAs, una por cada sentido de la comunicación. Las asociaciones de seguridad, a su vez, se almacenan en bases de datos de asociaciones de seguridad (SAD - Security Asocciation Databases).

El IETF ha definido el protocolo IKE para realizar tanto esta función de gestión automática de claves como el establecimiento de las SAs correspondientes. IKE es un protocolo híbrido que ha resultado de la integración de dos protocolos complementarios: ISAKMP y Oakley. ISAKMP define de forma genérica el protocolo de comunicación y la sintaxis de los mensajes que se utilizan en IKE, mientras que Oakley especifica la lógica de cómo se realiza de forma segura el intercambio de una clave entre dos partes que no se conocen previamente. El objetivo principal de IKE consiste en establecer una conexión cifrada y autenticada entre dos entidades, a través de la cual se negocian los parámetros necesarios para establecer una asociación de seguridad IPsec. Una característica importante de IKE es que su utilidad no se limita a IPsec, sino que es un protocolo estándar de gestión de claves que podría ser útil en otros protocolos, como, por ejemplo, OSPF o RIPv2.

El protocolo IPsec permite dos modos de funcionamiento: el modo transporte y el modo túnel, tanto AH como ESP proporcionan estos dos modos.

- *Modo Transporte*: en este modo se establece una seguridad extremo a extremo entre cliente a servidor, servidor a servidor o cliente a cliente. En cada extremo de la comunicación se requiere la implementación de IPsec. En este modo el contenido transportado dentro del datagrama protegido AH o ESP son datos de la capa de transporte (por ejemplo, datos TCP o UDP).

- *Modo túnel*: En este modo el contenido del datagrama AH o ESP es un datagrama IP completo (datos más cabeceras del mensaje). El paquete IP se encapsula completamente dentro de un nuevo paquete IP que emplea el protocolo IPsec y al que se le añade una copia de la cabecera del paquete IP original. El modo túnel es empleado principalmente por los gateways IPsec, con objeto de identificar la red que protegen bajo una misma dirección IP y centralizar de este modo el procesado del tráfico IPsec en un equipo. Otra aplicación del modo túnel, tanto con ESP como con AH, es poder establecer Redes Privadas Virtuales (VPN).

IPsec puede ser implementado tanto en un computador como en un router o firewall, al router o firewall que implementa IPsec se le denomina Gateway IPsec. Entre las ventajas de utilizar IPsec destacan:

- Está basado en estándares del IETF lo cual proporciona un nivel de seguridad común y homogéneo a todas las aplicaciones.
- Independencia de la tecnología física empleada (Frame Relay, PPP, xDSL o ATM).
- Integración en la versión actual de IP (IP versión 4) y, lo que es todavía más importante, se incluye por defecto en IPv6.
- Diseño modular, de modo que se pueda seleccionar el conjunto de algoritmos deseados sin afectar a otras partes de la implementación.
- Interoperabilidad entre fabricantes, lo cual constituye una garantía para los usuarios.
- Incorporación en muchos de los equipos de comunicaciones, así como en los sistemas operativos más comunes.
- Las VPN's con IPsec soportan todos los tipos y servicios IP como son: ICMP, VoIP, FTP, SQL, etc.
- Múltiples conexiones a través del túnel ya establecido.

Dentro de las desventajas de esta tecnología podemos mencionar las siguientes:

- La configuración de IPsec es complicada, debido a sus múltiples características.
- No se recomienda el uso de este protocolo en conexiones que utilicen NAT.
- La conexión entre el cliente y la VPN puede verse afectada si el firewall no permite el uso IPsec o IKE.
- Si no se implementan las opciones de seguridad correctamente en los túneles, la VPN puede quedar expuesta a ataques.

El protocolo IPsec y sus componentes se referencia en muchas RFC, entre las cuales se destacan 2403, 2404, 2405, 2451, 2857, 3948, 4301, 4302, 4303…

2.3.4.8 ProtocolosSSL/TLS (Secure Socket Layer/TransportLayer Security) [(7), (17), (18)]

Para empezar diremos que el protocolo TLS es la versión actual modificada del protocolo SSL, estos protocolos son utilizados para brindar seguridad, en la capa de transporte del modelo OSI, a través de métodos criptográficos, con lo cual se logran comunicaciones seguras a través de una red. Por sus características seguras, el uso de estos protocolos está muy extendido en Internet, sobre todo en lo relacionado con comercio electrónico y páginas seguras (comúnmente referenciadas como HTTPS), pero también se utilizan para la creación de redes privadas virtuales.

El protocolo SSL fue diseñado por Netscape en 1996, cuenta con tres versiones, la última es la 3.0, sin embargo ninguna versión de estas ha sido estandarizada por el IETF, por el contrario, ha sido el IETF quien ha desarrollado el protocolo TLS, basándose en SSL, pero incluyendo algunas leves modificaciones, TLS tiene también tres versiones, la 1.0, 1.1 y 1.2.Aunque el funcionamiento de estos protocolos es similar, las modificaciones hechas en TLS impiden la interoperabilidad entre TLS 1.0 y SSL 3.0, no obstante, existe compatibilidad entre las versiones superiores de TLS, sin embargo existe una recomendación para que cuando se inicien sesiones en SSL y TLS nunca se negocie el uso de SSL 2.0, ya que este protocolo contiene muchas fallas de seguridad.

Los objetivos del protocolo son varios:

- *Confidencialidad*: El protocolo se debe emplear para establecer una conexión segura entre dos partes utilizando para ello algoritmos criptográficos que garanticen la seguridad de la comunicación.
- *Integridad*: El protocolo provee mecanismos de autenticación que permiten certificar la autenticidad de las partes.
- *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 costosos computacionalmente, 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).

Para comprender mejor como SSL/TLS incorpora la eficiencia en su diseño, debemos entender los conceptos de sesión y conexión: una sesión es la que se realiza cuando se negocian los parámetros (algoritmos que utilizarán para el intercambio de claves, de autenticación, de cifrado y de compresión) de la comunicación entre las partes (cliente y servidor), la conexión es la que inicia las negociaciones y la que se establece después de estas. En el establecimiento de cada conexión se especifica un identificador de sesión, que

permite saber si la conexión empieza una nueva sesión o es continuación de otra. Las conexiones son transitorias y están asociadas a una única sesión, mientras que una sesión puede tener múltiples conexiones. Un ejemplo de lo útil de esta característica sería si un cliente inicia y luego termina la conexión, pero después quiere volver a conectarse, el sistema puede hacer uso de esos parámetros previamente acordados (cache de sesiones) para establecer la nueva conexión, con lo cual se considera que la nueva conexión hace parte de la misma sesión iniciada anteriormente.

Hemos dicho que SSL/TLS provee seguridad a la capa de transporte, sin embargo estos protocolos se ejecutan entre la capa de aplicación y por encima del protocolo de transporte TCP.

SSL/TLS se compone de cuatro protocolos. Estos protocolos tienen el mismo funcionamiento tanto en SSL como en TLS, pero en este último, incorporan algunos detalles para su mejor funcionamiento.

- ***Protocolo de registro(Record Protocol)***: se implementa sobre un protocolo de transporte fiable como el TCP. 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. Encapsula los protocolos de nivel más alto y construye un canal de comunicaciones seguro.

El protocolo de registro es un protocolo por capas. En cada nivel los mensajes incluyen campos para el tamaño, descripción y contenido. El protocolo toma un mensaje para ser transmitido, lo divide en bloques, comprime los datos (opcionalmente), los encripta, genera un MAC y transmite el resultado. En el lado del receptor se sigue un proceso inverso: descifrado, verificación, descompresión y reensamblaje. Este protocolo es el que permite que los datos protegidos sean convenientemente codificados por el emisor e interpretados por el receptor.

El protocolo de registros se emplea para encapsular varios protocolos de más alto nivel, uno de ellos, el protocolo de mutuo acuerdo.

- ***Protocolo de mutuo acuerdo (Handshake Protocol)***: se encarga de gestionar la negociación de los algoritmos de cifrado y la autenticación entre cliente y servidor. Define las claves de sesión utilizadas para cifrar. Se podría decir que es un protocolo de autenticación, que opera sobre el protocolo de registro.

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.

El protocolo de mutuo acuerdo consta de los siguientes pasos:

1) Intercambio de mensajes de saludo (*hello messages*) para acordar los algoritmos a emplear, intercambiar valores aleatorios y verificar si es una sesión reanudada.

2) Intercambiar los parámetros criptográficos necesarios para permitir que el cliente y el servidor acuerden un pre-secreto.
3) Intercambio de certificados e información criptográfica para permitir que cliente y servidor se autentifiquen.

4) Generar un secreto principal a partir del pre-secreto e intercambiar valores aleatorios.

5) Proporcionar los parámetros de seguridad a la capa de registro.

6) Permitir al cliente y al servidor verificar que su interlocutor ha calculado los mismos parámetros de seguridad y que el acuerdo se produjo sin alteraciones por parte de un tercero.

7) Protocolo de datos de aplicación

Los mensajes de datos de la aplicación son transportados por la capa de registro y son fragmentados, comprimidos y encriptados basándose en el estado actual de la conexión. Los mensajes se tratan como datos transparentes para la capa de registro.

- ***Protocolo de cambio de especificaciones criptográficas (Change Cipher Spec Protocol)***: Este protocolo marca las transiciones entre distintas estrategias de cifrado. Consta de un mensaje que se encripta y comprime con las especificaciones actuales de la conexión (no las pendientes).

Cuando el destinatario recibe este mensaje la capa de registro copia el estado de lectura pendiente al estado de lectura actual. De forma similar, el emisor cambia su estado de escritura al enviar este mensaje.

Este mensaje se envía durante el acuerdo, después de haber acordado los parámetros de seguridad pero antes de que se envíe el mensaje de verificación finalizada. Es un mensaje de un byte para notificar cambios en la estrategia de cifrado.

- ***Protocolo de alerta (Alert Protocol)***: Uno de los tipos de mensaje que soporta la capa de registro es el de alerta. Estos mensajes incluyen la severidad de la alerta y una descripción de la misma. Los mensajes de alerta con nivel de fatal provocan la inmediata terminación de la comunicación.

Existen distintos tipos de alertas:

- *Alerta de cierre*: El cliente y el servidor deben saber que la conexión se está cerrando para evitar un ataque de truncado. Cualquiera de los dos puede iniciar el intercambio de mensajes de cierre. Cualquier información recibida después de la alerta de cierre es ignorada.

- *Alerta de error*: La gestión de errores el protocolo de mutuo acuerdo es muy simple, cuando uno de los interlocutores detecta un error lo envía al otro y, si se trata de un error fatal, cierran la conexión.

Dentro de las ventajas de utilizar el protocolo SSL/TLS tenemos:

- SSL/TLS se encuentra ampliamente difundido en Internet.

- Este protocolo proporciona un canal de comunicaciones seguro entre el servidor y los clientes, al proporcionar confidencialidad, autenticación e integridad, por lo que es el preferido para la implementación de seguridad en servidores web y transacciones electrónicas.

- Puede proporcionar seguridad a otros servicios como FTP, POP3, SMTP… debido a que funciona entre los niveles de aplicación y transporte.

- Existen en Internet numerosas aplicaciones de libre distribución para implementar VPN's.

- Al implementar una VPN con SSL/TLS, los costos son mínimos ya que no se requiere mantenimiento de los clientes.

- Las VPN basadas en SSL/TLS no se ven afectadas por la existencia de firewalls entre el cliente y el servidor.

- Muchas de las VPN basadas en este protocolo proporcionan mecanismos de protección frente a ataques de tipos "man in the middle" y "denegación de servicios (DoS)".

Dentro de las desventajas de su utilización encontramos:

- Aunque SSL/TLS proporciona seguridad, esta protección es parcial, ya que está limitada solo al canal de comunicaciones, es decir mientras los datos viajan por el medio, pero después de que estos son entregados al servidor la seguridad desaparece.

- Debido a lo anterior, SSL/TLS carece de mecanismos que puedan garantizar por completo las actividades comerciales debido a que los datos dentro del servidor están expuestos, lo que podría generar diversos tipos de fraudes.

- La mayoría de servidores no autentican al cliente, por lo que cualquiera podría realizar transacciones electrónicas con los datos de las tarjetas robadas.

- SSL/TLS no soporta aplicaciones en tiempo real (Voz y video).

- No es totalmente transparente para el usuario.

- Tiene problemas con algunos protocolos de transporte y sobre todo con protocolos no orientados a conexión, así como con algunas aplicaciones.

- Una VPN SSL/TLS solo soporta servicios TCP y aquellos que trabajen en conjunto con SSL como HTTP, POP3, SMTP, etc.

- La eficiencia de una VPN puede verse afectada por la creación de múltiples llaves para una sola sesión.

El protocolo SSL v 3.0 se encuentra especificado en el RFC 6101, TLS v 1.0 en el RFC 2246, TLS v 1.1 en el RFC 4346 y el TLS v 1.2 en el RFC 5246, este último es el estándar actual, por lo que reemplaza a las versiones anteriores, incluyendo a SSL v 3.0.

2.3.5 Arquitecturas de VPN's [(6), (8)]

En este apartado se citarán los tipos de VPN's que se pueden implementar. Estas arquitecturas están determinadas por las infraestructuras de red y las necesidades de cada empresa.

2.3.5.1 VPN de acceso remoto

Fue, quizás, el primer uso que se le dio a la tecnología VPN, surgió de la necesidad de poder acceder a la red corporativa desde cualquier lugar. Actualmente es uno de los modelos más usados, ya que permite a diferentes tipos de usuarios autorizados (empleados, proveedores y clientes) que se conectan con la empresa desde sitios remotos (oficinas comerciales, domicilios, hoteles, etc.) utilizando Internet como vínculo de acceso. Una vez autenticados tienen un nivel de acceso muy similar al que tienen en la red local de la empresa.

El empleo de esta tecnología por parte de las empresas ha hecho que estas dejen de lado sus RAS corporativos, aunque no los desechan del todo ya que por razones de contingencia pueden resultar de utilidad.

2.3.5.2 VPN sitio a sitio o LAN to LAN

A diferencia del modelo anterior, en el que un usuario se conecta a la LAN de la empresa para acceder a algún recurso específico, la arquitectura VPN sitio a sitio se utiliza para

conectar oficinas remotas con la sede central de la organización, esto quiere decir que cualquier computador (si está autorizado) de la LAN de la sucursal está en capacidad de conectarse con la LAN de la sede principal y acceder a los recursos de esta que le estén permitidos. Igualmente en el esquema de acceso remoto no se ve la necesidad de conectar los equipos de la compañía con el computador del usuario; por el contrario en el modelo LAN to LAN puede surgir la necesidad de que la sede principal necesite acceder a datos que se encuentren en la LAN de la filial, lo que es factible al utilizar este modelo de conexión VPN.

En esta arquitectura el servidor VPN, que posee un vínculo permanente a Internet, acepta las conexiones vía Internet provenientes de los sitios y establece el túnel VPN. Los servidores de las sucursales se conectan a Internet utilizando los servicios de su proveedor local de Internet, típicamente mediante conexiones de banda ancha. Esto permite eliminar los costosos vínculos punto a punto tradicionales (realizados comúnmente mediante conexiones de cable físicas entre los nodos), sobre todo en las comunicaciones internacionales.

Si analizamos un poco más este esquema, podemos visualizar otra posible arquitectura que podríamos denominar extranet VPN, ya que no solo podemos brindar acceso a las sedes remotas de la empresa, sino a sedes de otras empresas, como las de los proveedores o clientes, que necesiten acceder a información o servicios de nuestra red LAN y viceversa, haciendo que ellos nos brinden acceso a sus propias redes LAN y a los servicios que se necesiten.

2.3.5.3 VPN sobre la LAN o VPN interna

Este esquema permite a las empresas la posibilidad de crear una VPN sobre su propia red LAN, sin la necesidad de utilizar Internet, este es, tal vez, el motivo por el cual este tipo de arquitectura VPN no se encuentra tan difundido a nivel empresarial, ya que como los datos viajan sobre la propia red LAN de la empresa, no se percibe ningún riesgo para dicha información. Además, las redes privadas virtuales se crean con el propósito de proteger los datos que viajan por medios potencialmente inseguros (como lo es Internet) y dado que asumimos que la red de la compañía es segura, no se hace necesario implementar ningún otro método de seguridad adicional. Sin embargo, analizado con un poco más de detalle, este modelo puede resultar sumamente útil dentro de la empresa, ya que serviría para aislar zonas y servicios de la red interna, lo que lo hace muy conveniente para mejorar las prestaciones de seguridad de las redes inalámbricas (WiFi).

2.3.6 Tipos de implementaciones VPN's [(6), (8), (19)]

Ya se mencionaron las distintas alternativas que existen al momento de querer implementar una red privada virtual. A continuación se mencionaran las formas en que se

puede poner en funcionamiento una VPN, existen básicamente dos tipos de implementaciones: basadas en hardware o basadas en software.

2.3.6.1 Basadas en hardware

Las VPN's basadas en hardware son aquellas que se implementan sobre dispositivos dedicados; equipos que se encargan exclusivamente de la creación y gestión de una red privada virtual.

La implementación de una VPN basada en hardware puede resultar sencilla de realizar, ya que, dejando de lado el hecho de que estos dispositivos solo deben conectarse a la red local para empezar a funcionar, incorporan interfaces gráficas que permiten gestionar de manera fácil los recursos que proporciona el dispositivo. Sin embargo, puede que la configuración de estos equipos no resulte del todo sencilla, ya que se debe tener en cuenta el protocolo sobre el cual se trabajará y creará la VPN.

Como estos dispositivos trabajan sobre la capa dos o tres del modelo OSI, pueden proporcionar el uso de uno o varios de los protocolos que se utilizan para la creación de VPN's como PPTP, L2TP o IPsec, algunos también pueden brindar la posibilidad de utilizar el protocolo SSL/TLS. Debido a que el estándar actual recomendado por la IETF es IPsec, la gran mayoría de los dispositivos que integran funciones de VPN traen incorporado este protocolo. Por lo tanto, si se escoge este protocolo como base para la creación de la red privada virtual, puede que la implementación no resulte tan fácil.

Otro aspecto a tener en cuenta al querer implementar una VPN sobre estos aparatos es la cantidad de usuarios que puede soportar el dispositivo y el total de redes privadas virtuales que se pueden crear sobre este, esto con el fin de poder evaluar posibles ampliaciones de la red virtual.

Dentro de las ventajas de utilizar este tipo de elementos para crear una VPN se encuentra el rendimiento, ya que todos los recursos de este equipo están dicados al funcionamiento de la red, igualmente son dispositivos seguros y fáciles de instalar y administrar.

Dentro de las desventajas de este tipo de solución podemos mencionar que: son equipos que pueden llegar a ser muy costosos, provienen de empresas desarrolladoras de soluciones, que les instalan sistemas operativos propios y también protocolos propietarios por lo que pueden resultar incompatibles al momento de usarse con otros equipos, igualmente son poco flexibles ya que no brindan la posibilidad de seleccionar el tráfico a enrutar, todo los datos se envían por el mismo túnel.

Además de los equipos VPN dedicados, encontramos dispositivos a los que podemos denominar “hibridos” debido a que, además de cumplir con su tarea principal, integran funcionalidad VPN.

Por lo general la funcionalidad de VPN se incluye dentro de dispositivos router (enrutadores) o firewall (muros de fuego o cortafuegos) debido a que son equipos por los cuales atraviesa todo el tráfico que entra de otras redes y sale de la propia red, lo que los hace idóneos para implementar esta funcionalidad.

Debido a esto, dentro de las redes privadas virtuales basadas en hardware podemos encontrar aquellas que utilizan routers o firewalls con funcionalidad VPN.

2.3.6.2 VPN’s basadas en routers

El router es un dispositivo que permite enrutar el tráfico de datos desde y hacia la red de una persona u organización.

Para crear una red privada virtual basada en routers se debe tener en cuenta que este dispositivo provea o se le pueda incorporar la funcionalidad VPN. En cuanto a esto, se debe considerar que si el enrutador no posee, pero se le puede agregar un módulo VPN, el rendimiento del equipo puede verse afectado, ya que se le está agregando carga extra al procesamiento de los datos que antes no tenía. En los routers que ya vienen con este servicio integrado no se posee este inconveniente, puesto que están desarrollados para cumplir con estas funciones simultáneamente.

Dentro de las ventajas y desventajas de este tipo de implementación podemos mencionar las mismas que para las VPN basadas en hardware.

2.3.6.3 VPN’s basadas en firewalls

Los firewalls son dispositivos de protección que se encargan de defender la red interna de las empresas de ataques provenientes de otras redes, como en el caso de internet. Es por eso que todas las empresas con conexiones a internet cuentan con estos dispositivos, sin embargo no todas cuentan con firewalls con funcionalidad VPN incorporada.

A diferencia del modelo basado en routers, no es necesario que el firewall incorpore funcionalidad VPN, si la incorpora o se le puede adaptar, la conexión resulta sencilla, pues solo se deben configurar los dispositivos para que acepten la conexión, pero si no, existe una configuración típica en la cual se usa el cortafuegos como un complemento de la red privada virtual para fortalecer la seguridad en la propia red.

En este tipo de configuración se utiliza un servidor VPN ubicado atrás de un firewall dedicado, con lo cual se garantiza que todos los paquetes que atraviesen el muro de fuego estén protegidos.

Dentro de las ventajas de utilizar este tipo de implementación tenemos:

- Proporciona niveles altos de seguridad.
- La VPN está completamente protegida frente a Internet al incorporar los sistemas de seguridad que proporciona el firewall.
- Si se cuenta con un cortafuego dedicado VPN se logra crear un solo punto de acceso a la red.
- Si se ubica el servidor VPN tras un firewall entonces sólo habrá un equipo que controle todo el acceso a y desde Internet.
- Las restricciones de red del tráfico VPN están ubicadas únicamente en el servidor VPN lo que facilita crear conjuntos de normas.
- Dentro de las desventajas se puede decir que son las mismas que para las VPN basadas en hardware.

2.3.6.4 Basadas en software

Como su nombre lo indica, es un tipo de software desarrollado para implementar VPN's de host a host, existe una gran variedad de software VPN que, dependiendo de su desarrollo, puede o no instalarse sobre cualquier sistema operativo. Generalmente utilizan el modelo cliente – servidor.

En este tipo de implementación el tráfico sale del anfitrión, se cifra o encapsula, dependiendo de la VPN instalada, y se enruta a su destino. Otra característica de este tipo de redes privadas virtuales es que utilizan como base el protocolo SSL/TLS para el establecimiento de las conexiones y basan su funcionamiento en certificados digitales y sistema de claves y llaves públicas.

Este tipo de implementación es recomendable en el caso donde las empresas no posean control sobre los dispositivos de comunicación y seguridad (routers y firewalls), cuando existen problemas de interoperabilidad entre los dispositivos VPN o cuando las conexiones se realizan desde diferentes ubicaciones (en el caso de trabajadores móviles).
Dentro de este sistema también se puede mencionar una configuración basada en firewall, similar a la vista anteriormente, con la diferencia de que el firewall es basado en software y que se instala dentro del mismo equipo que funciona como servidor VPN.

Dentro de las ventajas de este sistema de redes privadas virtuales está el que son altamente configurables debido a que el tráfico puede ser enviado a través de un túnel, en función de las direcciones o protocolos; su implementación resulta más económica,
Dentro de las desventajas se puede mencionar que poseen menos rendimiento que las basadas en hardware, su configuración es más delicada pues se debe tener en cuenta la seguridad de todo el equipo.

2.4 ELECCIÓN DE LOS ELEMENTOS PARA LA IMPLEMENTACIÓN

En el capítulo anterior se describieron a grandes rasgos los elementos necesarios para realizar la implementación; en este se hará una explicación de cuales son en sí los componentes sobre los cuales se desarrollará este trabajo, teniendo como base los fundamentos sobre los que se redacta esta monografía.

Para recordar un poco, esta obra se enfoca en presentar una solución de conectividad de acceso remoto de bajo costo y centrada en la utilización de software libre, que podría ser útil a pequeñas y, tal vez, a medianas empresas, por tanto el uso de dispositivos dedicados no está considerado dentro de este proyecto.
La solución que aquí se plantea tiene como base la utilización de un sistema operativo, un software de virtualización y una solución VPN basada en software, todo instalado en un PC convencional con acceso a internet.

2.4.1 ELECCIÓN DEL SISTEMA OPERATIVO

Dentro de la gran variedad de sistemas operativos desarrollados bajo el concepto de software libre encontramos muchas distribuciones basadas en Linux como Debian, Ubuntu, Red Hat, Centos, Fedora, por mencionar las distribuciones más populares; sin embargo debemos tener en cuenta que para nuestro montaje sería mejor utilizar versiones orientadas a servidores y no a escritorios, dentro de las distros basadas en Linux y orientadas a servidores tenemos: Debian, Red Hat, Ubuntu y Centos; para poder tomar una decisión sobre qué sistema elegir se va a dar una pequeña descripción de cada uno de ellos, mencionando algunos aspectos importantes como seguridad, compatibilidad, soporte, facilidad de uso y costo.

Antes de comenzar con esta comparación hay que decir que todas las distribuciones Linux manejan el mismo kernel[8], por tanto las diferencias son más de tipo administrativo.

[8] Software responsable de gestionar los recursos de un computador y de brindar a los distintos programas acceso al hardware de forma segura o básica.

2.4.1.1 Debian(20)

Una de las distribuciones más antiguas, Debian fue iniciada en 1993 por Ian Murdock. Es la única entre las versiones aquí mostradas en la que no hay ninguna compañía detrás, se mantiene y actualiza gracias al apoyo de los voluntarios del proyecto Debian, vinculados por el contrato social Debian.

No es una distribución para el usuario promedio ya que requiere de conocimientos avanzados en el manejo del sistema. Es una de las distros más estables. El soporte es proporcionado por los miembros del proyecto Debian, mediante correos electrónicos o con información publicada en internet en su sitio oficial, también hay una comunidad de Debian que se encarga de ayudar a solucionar problemas con este sistema, sin embargo la documentación existente en internet sobre esta distribución no es tan abundante, por lo cual la resolución de problemas puede tornarse demorada. Es compatible con gran variedad de hardware y software.

Las versiones de Debian se publican promedio cada dos años y se les brinda soporte por aproximadamente 3 años. Debian cuenta con versiones para escritorio y servidores, aunque es más popular en este último.

Esta distribución se pude conseguir gratuitamente en internet, aunque los desarrolladores del proyecto permiten a personas o empresas distribuirlo comercialmente mientras se respete su licencia.

2.4.1.2 Ubuntu[(21), (22)]

Distribución basada en Debian, que proporciona un sistema operativo estable y actualizado para el usuario promedio. Es la distro que más auge ha tenido en los últimos años, con cada vez más usuarios y que más rápido se ha adaptado a las necesidades de los mismos, esto debido al fuerte enfoque en la facilidad de uso e instalación del sistema que los desarrolladores le han dado.

El soporte de esta versión de Linux es proporcionado por Canonical, una empresa dedicada al desarrollo de software, así como por una gran comunidad de usuarios. Canonical cobra por el soporte a empresas que tienen instalado Ubuntu. Cada seis meses se publica una nueva versión de Ubuntu la cual recibe soporte por parte de Canonical, durante dieciocho meses, por medio de actualizaciones de seguridad, parches para bugs críticos y actualizaciones menores de programas. Las versiones LTS (Long Term Support), que se liberan cada dos años, reciben soporte durante tres años en los sistemas de escritorio y cinco para la edición orientada a servidores.

Ubuntu es de las distribuciones sobre las que más se encuentra información en foros y sitios de internet. Es muy compatible con un gran número de dispositivos hardware y cuenta con una gran cantidad de repositorios que permiten la instalación de muchos programas.

Aunque gran parte del énfasis de la distribución está en el escritorio, también hay una versión de servidor, y es que Canonical ha atraído el apoyo de los agentes comerciales más tradicionales de UNIX, como Oracle.

En cuanto al costo, el sistema operativo se consigue de forma gratuita, el desarrollo del programa esta sostenido por donaciones que se pueden realizar al momento de querer bajar el S.O de su sitio web.

2.4.1.3 Red Hat Enterprise(23)

Red Hat Enterprise Linux (RHEL) es probablemente la distribución de Linux más conocida y más popular en cuanto a servidores, es desarrollado por la empresa Red Hat Inc. Es una distro muy estable y compatible con gran cantidad de hardware y software.

RHEL es una distribución comercial, por tanto la adquisición de este sistema tiene un costo, más que por el software en sí, se cobra por la asistencia, por tanto el acceso a soporte y actualizaciones de seguridad requiere que los clientes paguen un honorario por estos servicios.

El manejo de este sistema es variable, ya que como se enfoca a servidores cuenta con diferentes alternativas para su inicio, desde ejecutarlo en modo consola (para usuarios avanzados), hasta cargar el ambiente gráfico que posee, el cual es más amigable y de fácil uso para usuarios promedio.

2.4.1.4 CentOS(24)

CentOS (abreviatura de Community Enterprise Operating System) es una versión de libre disposición compilada por voluntarios a partir del código fuente liberado por Red Hat. Esta distribución está enfocada a la parte de servidores.

Debido a que está basado RHEL es un sistema bastante estable, y al igual que Red Hat, presenta diferentes alternativas de uso. Dado que es una copia casi exacta de RHEL, las aplicaciones diseñadas para aplicaciones comerciales de Red Hat se ejecutarán sin modificaciones y con total compatibilidad.

En cuanto al soporte para CentOS, diremos que no es proporcionado por Red Hat, sino por la comunidad que creo esta distribución, igualmente es compatible con cantidad de hardware, sin embargo, Centos sólo ejecuta las versiones más básicas y estables de programas, reduciendo el riesgo de bloqueos del sistema. En el lado negativo, tiene como resultado un menor grado de funcionalidad comparado con versiones de software avanzados compatibles con otros sistemas de Linux.

Como abran notado, no se ha mencionado que tan seguro es cada uno de estos sistemas, pues bien, la razón es porque, como se mencionó al principio, todos estos sistemas trabajan bajo el mismo kernel, por tanto no hay una mayor diferencia entre ellos en este sentido, dado que todas las distribuciones incorporan las mismas características de desarrollo provenientes de Linux. La principal diferencia en este sentido será la forma en que se administra el sistema, por tanto la mayor parte de responsabilidad de la seguridad del sistema recae sobre el usuario que lo controla.

Habiendo mencionado esto y comparando las alternativas se ha optado por escoger como sistema base para el desarrollo de la implementación Ubuntu Server 12.04 LTS, debido más que todo a la gran cantidad de información que se puede conseguir, a su alta compatibilidad con hardware y software y porque es una distribución muy popular que continua en ascenso y que provee soporte hasta el 2017, además está enfocada a hacer más fácil el tránsito de plataformas Windows a entornos Linux.

2.4.2 ELECIÓN DEL SOFTWARE PARA VIRTUALIZAR

Como se mencionó en el apartado sobre virtualización (1.2), este trabajo contempla la virtualización de plataforma en lo concerniente a sistemas operativos invitados.

Este tipo de virtualización permite instalar sobre un sistema operativo anfitrión otro(s) sistema(s) operativo(s) denominados invitados, los cuales se instalan con todas las características y funcionalidades, como si estuviera instalado de forma nativa en el computador, estos sistemas corren totalmente independientes unos de otros y del sistema operativo anfitrión.

A continuación describiremos algunos de los productos que más se destacan en este ámbito de la virtualización.

2.4.2.1 VMware Workstation(25)

Es desarrollado por la empresa VMware Inc., filial de EMC Corporation, muy reconocida en el mundo del software para virtualización compatible con arquitecturas i686, x86 y x64.

VMware es una herramienta de virtualización muy común y con muchas características al momento de virtualizar sistemas operativos. VMware puede funcionar tanto en sistemas Windows como Linux y en Mac OS X que corre en procesadores Intel. Soporta la gran mayoría de versiones desktop y para servidores de Microsoft Windows, Linux, Solaris, Netware, Novell, FreeBSD, tanto en 32 bits (x86) como 64 bits (x64).

VMware virtualiza la plataforma que se instala, de forma que la mayor parte de las instrucciones en VMware se ejecutan directamente sobre el hardware físico.

VMware es muy estable, goza de buen soporte por parte del fabricante, el rendimiento de los sistemas virtualizados es bueno y rápido. Necesita módulos propietarios para el kernel.

Este software es propietario, por lo que se debe pagar una suma de dinero para adquirir su licencia, sin embargo también tiene versiones gratuitas como VMware Server y VMware Play.

2.4.2.2 VirtualBox(26)

Software de virtualización para plataformas i686, x86 y x64, desarrollado originalmente por la empresa alemana innotek GmbH, actualmente es desarrollado por la empresa Oracle Corporation.

Es una herramienta de virtualización muy popular entre ambientes Gnu/Linux, aunque también puede ser instalado en otras plataformas como Windows, Mac OS X, Solaris/OpenSolaris y OS/2 Warp.

VirtualBox es un software que posee una versión comercial denominada Oracle VM VirtualBox y una versión libre llamada VirtualBox OSE.

Si se compara VirtualBox con otras aplicaciones privadas de virtualización como VMware Workstation o Virtual PC, VirtualBox carece de algunas funcionalidades, sin embargo incorpora otras como la ejecución de máquinas virtuales de forma remota, por medio del Remote Desktop Protocol (RDP), además agrega soporte iSCSI.

VirtualBox almacena los discos duros de los sistemas invitados como archivos individuales dentro del sistema anfitrión, utilizando para ello un contenedor llamado Virtual Disk Image, incompatible con los demás software de virtualización.

2.4.2.3 Virtual PC(27)

Es la solución de virtualización propuesta por Microsoft, por lo cual tiene una gran acogida en una amplia parte de los usuarios Windows. Su función es emular mediante virtualización un hardware sobre el que funciona el sistema operativo. Aunque es un software propietario se distribuye de forma gratuita.

Virtual PC puede instalarse en la mayoría de versiones de Windows 7, Vista y XP, en sus versiones de 32 y 64 bits, aunque en algunas solo en las versiones de 32 y en otras en las de 64, igualmente puede instalarse en Windows Server 2003.

Soporta los siguientes sistemas operativos como invitados: XP SP3 Professional, Windows Vista Enterprise SP1, Windows Vista Ultimate SP1, Windows Vista Busines SP1, Windows 7 Professional, Windows 7 Ultimate, Windows 7 Enterprise.

La instalación de sistemas Linux como huésped en Virtual PC es posible, pero no es fluido. Windows Virtual PC no soporta Linux como huésped oficialmente, lo que quiere decir que Microsoft no brinda soporte para resolver los problemas que se presenten; sin embargo existen grupos en internet que brindan apoyo para resolver algunas situaciones presentadas al instalar este tipo de configuraciones.

2.4.2.4 Parallels Desktop(28)

Software de virtualización para plataformas Mac basados en tecnología Intel. Permite emular el hardware del sistema anfitrión. Es un software propietario por lo tanto adquirirlo tiene un costo. Muchos de los usuarios de este software lo utilizan para virtualizar sistemas Windows, sin embargo esto no quiere decir que Parallels soporte solo esta plataforma.

Las últimas versiones de Parallels Desktop permiten virtualizar varias plataformas en sus arquitecturas X86 de 32 y 64 bits como: la ya mencionada Windows en varias de sus versiones, OS/2, Solaris, diversas distribuciones de Linux, FreeBSD y, como es lógico, algunas versiones de Mac OS X.

Parallels solo puede instalarse en plataformas Mac, la última versión (9), es soportada por las siguientes versiones Mac OS X y posteriores: Snow Leopard , León, León de Montaña y Mavericks, con uno de estos procesadores intel: Core 2 Duo, Core i3, i5 e i7 y Xeon.

Aquí se han mencionado solo algunas de los sistemas de virtualización que existen, sin embargo hay algunos otros, como Xen, KVC, Quemu, etc., cada uno con sus propias características para realizar este tipo de labor.

En cuanto al sistema de virtualización escogido para contribuir al desarrollo de esta implementación se ha optado por VirtualBox, debido a que es bastante compatible con ambientes Linux, que es el sistema operativo elegido, su manejo es muy intuitivo y presenta un entorno gráfico amigable, por otro lado puede ejecutarse en modo consola para realizar otra serie de actividades que no se describirán en este trabajo, ya que no se pretende montar un sistema virtual muy elaborado, igualmente está acorde con la orientación de este proyecto, ya que es software libre.

2.4.3 ELECCIÓN DEL SOFTWARE VPN

Ya se han seleccionado el sistema operativo y el software de virtualización, necesarios para el desarrollo de esta propuesta, ahora seleccionaremos el que es el componente más importante para la realización de esta implementación, el software VPN; como ya se mencionó, se optó por crear una red privada virtual basada en software, ya que por las características de este proyecto es el sistema más adecuado.

Dada la popularidad de las redes privadas virtuales, existe gran cantidad de software para la creación de VPN, incluso muchos de los sistemas operativos actuales incorporan dentro de sus opciones de red la posibilidad de crear este tipo de redes.

A continuación mencionaremos algunos de los desarrollos de software más populares para la creación de VPN.

2.4.3.1 LogMeIn Hamachi(29)

Es una aplicación gratuita utilizada para crear redes privadas virtuales, sin la necesidad de ninguna o casi ninguna configuración adicional en el cliente. El atributo principal de este cliente es la facilidad de uso. Para instalar esta aplicación solo es necesario descargar el aplicativo desde un sitio seguro en Internet, configurarlo y conectarse.
El software cliente crea una interfaz de red virtual, con una dirección IP proporcionada por el servidor VPN; por esta interfaz entra y sale todo el tráfico VPN cifrado y autenticado por el cliente hamachi.

Si bien la aplicación es gratuita, es administrada en su totalidad por la empresa proveedora, por lo cual no se puede ejercer ningún control o configuración sobre la VPN creada. Con la aplicación gratuita no podemos conectar más de 16 ordenadores (Aunque

la cifra puede ser menor). Si necesitamos conectar una cantidad mayor de equipos, la empresa proveedora cuenta con una versión comercial que permite conectar un numero mayor de computadores.

LogMeInHamachi es una solución multiplataforma que tiene una versión estable para Windows y una versión beta para Linux y Mac.

2.4.3.2 NeoRouter(30)

Es una aplicación VPN gratuita, segura, fácil de configurar y usar, que proporciona una solución de acceso remoto.

Se puede instalar en los sistemas operativos más comunes, Windows, Linux, Mac e incluso en Android e iOS.

Trabaja en el modo cliente – servidor, soporta conexiones P2P, se ejecuta como servicio dentro del sistema, permite el manejo de escritorio remoto y puede ser ejecutado desde una memoria USB.

2.4.3.3 Tinc(31)

Es un demonio de red privada virtual (VPN) que utiliza un túnel y cifrado para crear una red privada segura entre hosts en Internet. Tinc es un software libre y licenciado bajo la Licencia Pública General de GNU versión 2 o posterior.

Actualmente Linux, FreeBSD, OpenBSD, NetBSD, MacOS / X, Solaris, Windows 2000, XP, Vista y Windows 7, y 8 plataformas más. Tinc también tiene soporte completo para IPv6, proporcionando la posibilidad de tunelizar el tráfico IPv6 a través sus túneles y de crear túneles IPv6 a través de redes existentes.

2.4.3.4 OpenVPN [(7), (32)]

Al igual que los demás productos descritos, es una solución para la implementación de redes privadas virtuales que implementa conexiones de capa 2 o 3, usa los estándares de la industria SSL/TLS para cifrar.

Es una solución multiplataforma que ha simplificado la configuración de VPN's frente a otras soluciones más antiguas y difíciles de configurar como IPsec y haciéndola más

accesible para gente inexperta en este tipo de tecnología, incluye características que permiten configuraciones simples para túneles Punto a Punto, Acceso Remoto, VPN's sitio-a-sitio, seguridad para redes Wi-Fi, además incluye funcionalidades de nivel empresarial para proveer balanceo de cargas, failover, y controles de acceso refinados.

Es de las aplicaciones más utilizadas para la creación de VPN's basadas en software, por lo tanto hay gran cantidad de información sobre este aspecto.Esta liberado bajo la Licencia Pública General GPL versión 2.

Después de observar las alternativas planteadas, se considera que la aplicación mas adecuada para llevar a cabo la implementación propuesta es OpenVPN, ya que cuenta con características de seguridad, facilidad de configuración y uso, que lo hacen una buena herramienta para este tipo de proyectos, además de ser software libre. así que miremos un poco más esta aplicación.

2.5 OPENVPN [(7),(33),(34),(35)]

OpenVPN es una herramienta software multiplataforma para la creación y administración de redes privadas virtuales la cual ofrece seguridad por medio del protocolo SSL/TSL y librerías criptográficas. Esta herramienta fue creada en el año 2001 y combina facilidad de uso con alto grado de seguridad. Además, está publicada bajo licencia GPL. Estas características la hacen una muy buena opción en conectividad frente a otras herramientas de su tipo, como es el caso de su principal competidor IPsec, las cuales son mucho más complejas de usar para un usuario no experto en redes.

Dentro de las plataformas que soportan OpenVPN están Linux, Windows 2000/XP y Vista, OpenBSD, FreeBSD, NetBSD, Mac OS X y Solaris. En cuanto a seguridad se refiere, soporta encripción mediante claves privadas pre-compartidas (Llave estática) o seguridad por llave pública (Modo SSL/TSL) usando certificados digitales. Adicionalmente también soporta túneles TCP/UDP.

Para los modos de conexión utiliza el driver TUN/TAP para crear dispositivos virtuales, los cuales pueden ser vistos tanto como dispositivos punto a punto (Modo Túnel, utilizando dispositivos virtuales TUN), como dispositivos Ethernet (Modo Puente, mediante dispositivos virtuales TAP). Esto permite utilizar OpenVPN para conexiones Host to Host, para interconexión de estaciones de trabajo; en conexiones Road Warrior, comúnmente utilizada para accesar desde estaciones de trabajo remotas a la LAN; así como también en conexiones LAN to LAN, que sirve para comunicar sedes remotas como si estuviera físicamente contiguas.

2.5.1 Ventajas y desventajas

Dentro de las ventajas de OpenVPN se tiene:

- Posibilidad de trabajar sobre Capa 2 y Capa 3 mediante sus dos modos de trabajo (Túnel y Puente), lo cual permite enviar tramas Ethernet, paquetes IPX y paquetes NETBIOS.
- Protección de los usuarios remotos por el firewall interno. Al establecerse el túnel de comunicación entre el usuario remoto y la LAN, el firewall de la red toma al usuario como un usuario de la LAN más, protegiéndolo de cualquier amenaza potencial mientras permanezca conectado en la VPN.
- Desde la versión 2.0. se permite la utilización del mismo puerto TCP o UDP para comunicaciones entrantes, mientras se realicen configuraciones independientes para cada conexión. Esta funcionalidad está habilitada en el modo especial del servidor.
- Posibilidad de conexión remota a través de cualquier firewall si se cuenta con conexión a internet y acceso a sitos Https.
- Configuraciones de las reglas de firewall independientes para cada interfaz virtual. Todos los conceptos de NAT pueden ser aplicados a los túneles virtuales.
- OpenVPN tiene soporte para proxy. Puede ser configurado para correr como servicio TCP o UDP. También se puede configurar como servidor o como cliente (estableciendo o esperando conexiones, respectivamente).
- Gran flexibilidad gracias a la opción de ejecutar diferentes scripts durante la configuración de la conexión. Estos scripts pueden ser utilizados para fines que van desde autenticación hasta recuperación de errores.
- Posibilidad de utilizar OpenVPN a nivel de aplicación. Para conectarse a la VPN de la organización el usuario solo necesita realizar una conexión desde un navegador a un sitio HTTPS.
- Es una herramienta de código libre lo que permite a los desarrolladores hacer modificaciones a su código para adaptarlas a las necesidades particulares del usuario.
- Su diseño modular hace de esta herramienta excepcional en cuanto a seguridad se refiere. Ninguna otra solución de este tipo ofrece tantas ventajas en este aspecto.

En cuanto a desventajas tenemos las siguientes:

- No tiene compatibilidad con el IPSec, el cual es el estándar para la mayor parte de soluciones VPN y de fabricantes de hardware.

- Hasta ahora no hay muchos fabricantes que incluyan esta solución en sus dispositivos.
- Su uso no es ampliamente difundido.

2.5.2 Historia de OpenVPN (35)

OpenVPN surgió durante la coyuntura del 11 de Septiembre de 2001, cuando James Yonan pensó en la necesidad de encontrar una forma de conexión remota a través de internet sin tener que preocuparse porque esta no fuese segura. Hasta ese momento las soluciones VPN existentes se separaban en dos clases: Una centrada en la seguridad y otra centrada en la facilidad de uso. En ese entonces no había una herramienta que lograra combinar estos dos aspectos de una manera eficiente. IPSec, la principal solución VPN, a pesar de ser difícil de configurar proveía un nivel aceptable de seguridad. Pero a pesar de esto, su estructura compleja lo hacía vulnerable a diferentes tipos de ataques. De esta forma Yonan llegó a un diseño de una solución modular basada en el enfoque de usabilidad y utilizando los drivers TUN/TAP del kernel Linux. Esto inmediatamente proveyó la flexibilidad que Yonan buscaba con respecto a otras opciones basadas en SSL/TSL; Así OpenVPN ofrecería todas las opciones que los dispositivos de red podían realizar. El nombre de OpenVPN proviene del proyecto OpenSSL con su mensaje “Esto es código abierto y software libre”.

2.5.2.1 OpenVPN versión 1

Open VPN inicio operaciones el 13 de Mayo de 2001 con la versión 0.9, la cual contaba únicamente con Tunelización de paquetes IP a través de UDP con cifrado Blowfish y firmas SHA HMAC. Ya en la versión 0.91 se incorpora el soporte SSL/TSL. Para la versión 1.0 lanzada un año después ya se cuenta con autenticación basada en SSL/TSL e intercambio de claves. Dicha versión es la primera con documentación en forma de ManPage (documentación en modo consola).

Poco después en la versión 1.0.2 se introdujo la primera adaptación a sistemas basados en RPM (RedHat Package Manager). Desde este momento el desarrollo de OpenVPN se incrementó vertiginosamente al lanzarse nuevas actualizaciones cada 4 u 8 semanas.

2.5.2.2 OpenVPN versión 2

Paralelamente al desarrollo de la versión 1, se desarrolló la versión 2, saliendo a la luz en febrero de 2004 la versión 2.0-test3, la cual buscó realizar un primer acercamiento a los servidores multicliente en OpenVPN; esta hasta la actualidad es una de las principales

características de la herramienta al poderse conecta varios clientes por el mismo puerto. En febrero 22 de 2004 las dos líneas trabajadas en las versiones 1.6-beta7 y 2.0-test3 se fusionaron para continuar trabajando sobre la versión 2.

Hubo al menos 29 versiones nombradas como “test”, 20 versiones “beta” y 21 “reléase candidate” (rc) hasta abril de 2005. Esta gran cantidad de versiones fueron posible gracias al trabajo de un gran número de desarrolladores que aportaron al proyecto incrementando la estabilidad y el desempeño del mismo.

Los siguientes son las principales características de la versión 2:

- Soporte multicliente
- La configuración de los clientes puede ser controlada por el servidor.
- Adición de una interfaz de administración (Telnet).
- Se trabaja ampliamente el driver para Windows.

Para mayor detalle de la evolución de OpenVPN versión a versión, visitar: http://openvpn.net/index.php/open-source/documentation/change-log/71-21-change-log.html.

2.5.3 Comparación entre OpenVPN e IPSec (35)

A continuación se hace un paralelo entre OpenVPN e IPSec en el cual se muestran los puntos clave de cada herramienta.

Tabla 2.Comparativa entre OpenVPN e IPsec.

IPsec	OpenVPN
El estándar en tecnologías VPN.	Todavía no muy conocida, no compatible con IPSec.
Múltiples de hardware (dispositivos).	Solo disponible para computadoras, pero en la mayoría de los sistemas operativos disponibles.
Tecnología ampliamente probada.	Tecnología aún en desarrollo.
Diferentes interfaces de usuarios para administración.	No existe una interfaz gráfica profesional, aunque si existen varios proyectos al respecto.

Modificación compleja de la pila IP	Tecnología Sencilla
Necesidad de modificaciones críticas al kernel.	Interfaces de red y paquetes estandarizados.
Se requieren privilegios de administrador.	Se ejecuta en el espacio de usuario y puede usarse el chroot para su ejecución.
Las implementaciones de diferentes fabricantes pueden ser incompatibles entre sí.	Tecnologías de cifrado estandarizadas.
Tecnología y configuración complejas.	Fácil, bien estructurado, tecnología modular, fácil configuración.
Curva de aprendizaje pronunciada para los nuevos en esta tecnología.	Fácil de aprender, rápido éxito para los novatos.
Requiere varios puertos y protocolos en firewall.	Solo necita un puerto en el firewall.
Problemas en el direccionamiento dinámico en ambos extremos de la conexión.	El direccionamiento dinámico funciona impecablemente. Rápida reconexión.
	Control de tráfico.
	Veloz (Más de 20 Mbps en máquinas de 1 GHz)
	Ningún problema con NAT en ambos extremos de la comunicación.
	Compatible con firewalls y proxys
	Posibilidad de conexión para “Road Warrios” (Agentes viajeros).

Fuente http://es.wikipedia.org/wiki/OpenVPN

2.5.4 Componentes de OpenVPN[(7), (34)]

Después de haber detallado las múltiples ventajas de OpenVPN se continuará con una descripción de los componentes con las que cuenta la herramienta para la implementación de las conexiones VPN. Cabe señalar que OpenVPN no cuenta con una versión específica para cliente o para servidor. De esta forma, al instalarlo éste puede desempeñar cualquiera de los dos roles según la configuración que se le aplique a la herramienta. Al ejecutarse en el espacio del usuario, no requiere ningún componente adicional a los drivers TUN/TAP para su correcto funcionamiento, que dependiendo de la versión del sistema operativo puede ya venir incluidos en el kernel del mismo.

La seguridad está dada por SSL y las librerías Crypto de OpenSSL, las cuales pueden utilizarse alternativamente para ofrecer diferentes niveles de seguridad. Mediante el uso de ambas pude obtenerse autenticación con uso de certificados, cifrado con clave pública e intercambio dinámico de claves mediante TLS. Utilizando solo la librería Crypto se puede utilizar un nivel básico de autenticación mediante el uso de claves estáticas compartidas.

También se deja al usuario la opción de no utilizar ninguna técnica de cifrado, aunque es la opción no recomendada.

Adicional a la seguridad, se ofrece también la opción de compresión de datos con la librería LZO que mejora el rendimiento en el flujo de datos del túnel. Opciones adicionales pueden ser adheridas a la configuración de la herramienta mediante la manipulación de los scripts de OpenVPN.

2.5.4.1 Archivos de configuración (7)

Después de la instalación de OpenVPN es necesaria la ejecución de los scripts de configuración en los cuales se definen las características de las conexiones a establecer en el túnel. Estos archivos presentan ligeros cambios si se habla de sistemas operativos Windows, o si se trata de sistemas Linux. Inicialmente, esas diferencias en los archivos se remiten a sus extensiones, siendo la extensión para los sistemas Windows “ovpn” y “conf” para los sistemas Linux. Adicional a esto, los comandos incluidos en ellos pueden contener (o carecer) de ciertas directivas para determinadas instrucciones.

Un ejemplo de esto es la ejecución del archivo de configuración, para el cual se usa la directiva “--config”, que en el caso de Windows se omite:

- Para Linux: openvpn --config Nombre_Archivo.conf
- Para Windows: openvpn Nombre_Archivo.ovpn

Según la estructura de la VPN que se quiera implementar se debe crear un archivo de configuración en modo cliente, servidor, o sin ningún rol; si se quiere crear una conexión en modo túnel, modo servidor o como extremo de comunicación, respectivamente. En cualquier caso se ejecuta este archivo independientemente de los demás scripts y complementos que se requieran adicionalmente.

A continuación se muestran los comandos necesarios para configurar un túnel sin cifrado, ni autenticación, utilizando solo comandos de consola:

- openvpn --remote domain1.com --dev tun0 --ifconfig 10.8.0.1 10.8.0.2
- openvpn --remote domain2.com --dev tun0 --ifconfig 10.8.0.2 10.8.0.1

Según se observa se configura una conexión entre los extremos de comunicación con interfaces de red de dirección 10.8.0.1 y 10.8.0.2, trabajando en modo túnel al definirse el driver como “tun” en ambas máquinas.

El mismo ejemplo basado en archivos de configuración para entornos Linux quedaría de la siguiente manera:

Tabla 3. Ejemplo sobre archivos de configuración en OpenVPN.

ExtremoA.conf	ExtremoB.conf
Remote domain1.com Dev tun0 Ifconfig 10.8.0.1 10.8.0.2	Remote domain2.com Dev tun0 Ifconfig 10.8.0.2 10.8.0.1

Después de creados los archivos, se ejecutan mediante OpenVPN en una terminal de consola en ambos extremos de la conexión:

openvpn --ExtremoA.conf

openvpn --ExtremoB.conf

2.6 Drivers TUN/TAP [(7),(36)]

La creación de los túneles en OpenVPN depende de los drivers TUN/TAP para su implementación en ambos extremos del túnel. Estos fueron desarrollados en 1999 por Maxim Krasnyansky y hacían parte de la herramienta para creación de túneles virtuales VTUN, de libre distribución. En ese entonces para las implementaciones de VPN's se debía instalar la VTUN. Los túneles virtuales TUN/TAP fueron incluidos en el kernel de Linux a partir de las versiones 2.4.x. En el caso de sistemas Windows, en el 2003 se derarrolló una herramienta que permitía esta implementación llamada TAP-Win32, la cual ha estado en constante desarrollo desde ese entonces.

Cuando la interfaz virtual se llama "tun" significa que está trabajando en modo túnel enlazando con otra interfaz del mismo tipo en el otro extremo de la conexión virtual. Si la interfaz recibe el nombre de "tap" significa que está trabajando en modo puente, simulando una interfaz ethernet en vez de punto a punto.

Una de las ventajas de utilizar drivers TUN/TAP es que estos permiten que aplicar reglas de firewall como si se tratase de una interfaz ethernet real. De esta manera se pueden aplicar a la inertaz virtual las mismas reglas que a la interfaz real. Así mismo, las interfaces virtuales desplazan la complejidad de las de la configuración de los componentes de seguridad y cifrado, de los componentes de red, al trabajar en el espacio del usuario.

Dentro de las características más sobresalientes de los controladores TUN/TAP se tienen:

- Facilidad de uso para la creación de túneles virtuales en redes TCP/IP.
- Permite diferentes opciones de configuración de las VPN.
- Utiliza las librerías “Zlib” para compresión en TCP y la “LZO” para compresión en TCP/UDP.
- Permite el cifrado por medio de Blowfish y funciones de integridad por hash MD5.
- Permite crear túneles TCP/UDP.
- Soportado por diferentes sistemas operativos (Linux, FreeBSD, OpenBSD, Apple OS, Solaris).

Entre las opciones de túneles que se pueden implementar están:

- Túneles IP (Tun): Para conexiones punto a punto.
- Túneles Ethernet (Tap): Para conexiones LAN to LAN, soportando protocolos de red (IP,IPX, AppleTalk,etc).
- Túneles Serie (Tty): Soporta las conexiones que utilizan protocolos de serie (PPP,SLIP,etc).
- Tuberías (Pipes): Soporta todos los programas que trabajan sobre tuberías de Unix, como SSH.

2.6.1.1 OpenSSL [(7),(37),(38)]

OpenSSL es un proyecto de software libre basado en la librería SSLeay y desarrollado en C, por Eric Young y Tim Hudson. La finalidad principal de OpenSSL es proveer niveles de cifrado y seguridad a las comunicaciones en internet implementando SSL (version 2 y 3) y TSL (vesrion 1), que ofrecen funcionalidades importantes de seguridad, entre las que se resaltan la generación de llaves RSA y certificados digitales. Sobre este proyecto hay una extensa comunidad de desarrolladores que unen sus esfuerzos para ofrecer mejoras constantes a la herramienta y a su documentación.

Dentro de los diferentes algoritmos criptográficos soportados por OpenSSL están:

- Algoritmos para Cifrado: AES, Blowfish, SEED, Camellia, DES, 3DES, CAST-128, RC2, RC4, RC5, GOST 28147-89.

- Algoritmos para funciones Hash: MD2, MD5, SHA-1, SHA-2, RIPEMD-160, MDC-2, GOST R 34.11-94.

- Algoritmos para criptografía de clave pública: RSA, DSA, Diffie-Hellman, Curva elíptica, GHOST R 34.10-2001.

Para su funcionamiento OpenSSL utiliza una interfaz de línea de comandos mediante la cual se pueden ejecutar gran cantidad de instrucciones provistas entre otras por sus librerías Crypto y SSL. Mientras la librería SSL provee funcionalidades para la implementación de SSL/TSL, la librería Crypto ofrece las funciones de criptográficas y todo lo relacionado con cifrado. Mediante esta interfaz y las opciones soportadas por sus librerías se pueden realizar las siguientes operaciones:

- Operaciones criptográficas de clave pública.

- Administración de llaves públicas y privadas.

- Creación de certificados X.509.

- Cálculos de resumen de mensajes.

- Encripción y desencripción mediante múltiples cifradores.

- Pruebas de cliente y servidor SSL/TSL.

- Manejo de correo cifrado firmado por S/MIME.

- Generación y verificación de solicitudes de marca de tiempo.

2.6.1.2 LZO [(7),(39)]

LZO (Lempel–Ziv–Oberhumer) es una librería multiplataforma desarrollada en C y publicada mediante licencia GPL, que implementa múltiples algoritmos para la compresión/descompresión de datos en tiempo real, siendo bastante notoria su velocidad de descompresión.

Dentro de las características principales de LZO se encuentran:

- Decompresión simple y bastante rápida.

- No se presenta gasto de memoria para la descompresión.
- No necesita buffers adicionales para la descompresión a los de fuente y destino.
- Compresión algo rápida. (64KB de RAM requeridos).
- Permite extra compresión con costo en velocidad. La velocidad de decompresión no se afecta.
- Permite ajustar el balance entre la velocidad y la capacidad de compresión, sin afectar la velocidad.
- Permite un uso de memoria de solo 8 KB para ciertos niveles de compresión.
- El algoritmo implementa seguridad en hilos.
- El algoritmo no presenta perdidas de datos en la descompresión.

Existen múltiples variables del algoritmo LZO, como lo son: LZO1, LZO1A, LZO2A, LZO1B, LZO1F, LZO1X, LZO1Y, LZO1Y. Pero de todas estas, la mejor opción es la LZO1X, pues da una mejor rata de compresión que las demás.

Cada algoritmo, sin importar si se usa para compresión o descompresión, requerirá un archivo de cabecera el cual contiene los prototipos para ese algoritmo. Esta cabecera es el nombre del algoritmo junto con la extensión ".h". De esta manera, para el algoritmo LZO1X la cabecera sería "lzo1x.h".

En cuanto al proceso de compresión y descompresión se requerirán de los siguientes pasos:

- Añadir la cabecera para el algoritmo LZO seleccionado (lzo1x.h, por ejemplo).
- Inicializar la librería mediante la función lzo_init().
- Llamar la función de compresión/descompresión requerida.
- Vincular la aplicación con la librería LZO.

2.6.1.3 Autenticación adicional [(7),(40)]

Una manera de proveer políticas de autenticación eficientes en entornos Linux es la utilización del Pluggable Authentication Module (PAM). Este módulo permite establecer políticas de autenticación para los diferentes niveles de acceso que se requieran, separando totalmente este proceso del resto de funciones del sistema. De esta forma, se pueden desarrollar módulos y aplicaciones independientemente del método de autenticación a utilizar.

El módulo PAM descompone el método de autenticación en 4 niveles independientes: autenticación, cuentas de usuario, gestión de sesiones y actualización de passwords. Los cuales se definen para cada aplicación que se ejecute en la máquina, dando como resultado una aprobación o denegación del servicio, según los parámetros definidos por el administrador.

Los niveles de autenticación están definidos de la siguiente manera:

- *Autenticación (Authentication)*: Este servicio busca establecer quién es el usuario mediante la solicitud de credenciales de acceso, asignándole al usuario una serie de permisos al comprobar su identidad.

- *Cuentas de usuario (Account)*: Una vez que se verifica y se da acceso al usuario, este módulo comprueba qué permisos efectivos se tienen en su cuenta. De esto manera se comprueba a qué aplicaciones tiene permiso el usuario, si su cuenta ha expirado, qué privilegios tiene la cuenta, etc.

- *Gestión de sesiones (Session)*: Este nivel contiene las tareas a ejecutarse antes y después de que el usuario inicia el servicio o aplicación. El módulo realiza tareas tales como cargar el perfil del usuario (directorio home), guardar datos de navegación, traer información de cuentas de correo.

- *Actualización de password (Password)*: Administración de todo lo relacionado con la información necesaria para realizar el proceso de autenticación.

En PAM es posible asignar una configuración por defecto o realizar la configuración de diferentes permisos de acceso, según el rol del usuario en el sistema. En este punto es imprescindible que el administrador conozca a la perfección que accesos se requiere para cada perfil pues de no aplicar adecuadamente una política puede comprometer seriamente la seguridad del sistema y/o generar falso rechazo o falsa aceptación a los usuarios del mismo.

Dentro de las ventajas que ofrece PAM tenemos:

- Brinda un método de autenticación centralizado.

- Posibilita al programador desentenderse de las rutinas de autenticación.

- Facilita el mantenimiento de las aplicaciones.

- Hace posible a desarrolladores y administradores controlar de manera flexible las diferentes opciones del módulo.

2.6.1.4 Scripts y customización

Una de las principales ventaja de OpenVPN es su flexibilidad de configuración. Para esto el usuario puede definir las configuraciones que requiera mediante el uso de Scripts. Con estos scripts es posible definir tantos aspectos del funcionamiento del túnel como el usuario lo necesite. De esta forma es posible definir esquemas para inicio o arranque del sistema, enrutamiento, comprobación de certificados, iptables, etc.

Dentro de los plugins se pueden encontrar múltiples opciones para autenticación, interfaz de usuario, firewall, generación de gráficos de parámetros, etc. Todo esto gracias a la comunidad de desarrolladores y entusiastas del proyecto que constantemente lanzan diferentes complementos para OpenVPN. Haciendo que cada vez sea más robusta, amigable y accesible. De esta manera el nivel de personalización de la herramienta es bastante alto, comparado con muchas de las soluciones existentes. En la página del fabricante se pueden encontrar gran cantidad de fuentes en las cuales descargar complementos.

2.6.2 Modos de funcionamiento (7)

Como ya se dijo anteriormente, OpenVPN permite trabajar en varios modos de funcionamiento dependiendo de las necesidades del usuario y de las especificaciones que requiera la red para funcionar correctamente. Básicamente se utilizan dos modos de conexión, modo túnel o modo puente. La diferencia entre estos dos radica en que el primero es para conexión de equipos Host-Host, trabajando sobre protocolo IP y el otro trabaja con el estándar Ethernet, ideal para conexiones LAN-LAN con soporte para protocolos No-IP (como IPV4, IPV6, IPX, AppleTalk).

Un modo de conexión adicional es el modo Road Warrior (Host-LAN), el cual consiste en uno o varios equipos acceden a la red mediante la conexión a un servidor en el cual se autentica y desde el tienen el acceso a los recursos de la red. Este modo en un principio se basa en el modo de conexión túnel (aunque es posible configurarlo como puente) y en su driver virtual correspondiente. Este es el modo de comunicación más usado y permitiendo la encripción de la comunicación entre el Host y el servidor VPN. Para esto se requiere una serie de claves y certificados digitales.

Para poder configurar un modo u otro, es necesario definir en el fichero de configuración de la conexión qué driver virtual configurar. Para modo túnel se usa el driver TUN y para modo puente se usa el driver TAP. Es importante que a ambos extremos de la conexión se defina el mismo driver para que la conexión sea exitosa. En ocasiones se le asigna un número a la interfaz virtual definida, pero es necesario, que este al igual que el tipo de driver, coincida en ambos extremos de la conexión.

2.6.2.1 Modo Túnel:

Este modo es el más usado para establecer conexiones entre dos máquinas remotas. Dentro de sus principales ventajas están:

- Es el método más sencillo de conexión.
- Eficiente y escalable.
- Permite establecer una MTU de una forma más eficiente.

Dentro de sus desventajas se tienen:

- En ocasiones requiere de un servidor WINS, según la configuración del cliente.
- Es necesario definir las rutas de las subredes en los extremos de comunicación.
- Las aplicaciones con funciones de broadcast no podrán ver las demás máquinas de la subred.
- No soporta el estándar Ethernet.

2.6.2.2 Modo Puente:

Utilizado para conexión de Subredes. Sus principales ventajas son:

- Permite que las aplicaciones que implementan broadcast puedan enviar paquetes a las subredes.
- Trabaja con protocoles sobre Ethernet.
- Comunicación entre cualquier par de máquinas de la subred.

- Si se quiere configurar la conexión para Road Warrior con el driver TAP, es relativamente sencillo.

Desventajas:

- No tan eficiente y escalable comparado con el modo Túnel.

Ya para terminar este recuento sobre OpenVPN, se incorpora una matriz DOFA sobre este proyecto, el cual presenta grandes expectativas para el futuro cercano y de lo que sus desarrolladores podían llegar a proporcionar más adelante.

En esta matriz se puede apreciar como este aplicativo tiene el potencial suficiente para convertirse en una tecnología de creación de redes privadas virtuales muy fuerte, si los desarrolladores del producto implementaran algunos de los soportes de los que adolece actualmente OpenVPN y promovieran más el uso de este producto.

Tabla 4. Matriz DOFA del proyecto OpenVPN.

ANÁLISIS INTERNO	LISTA DE FORTALEZAS	LISTA DE DEBILIDADES
	F1. Altamente seguro. **F2.** Configurable y flexible, pudiéndose utilizar scripts para facilitar su implementación. **F3.** Se pueden realizar conexiones a través de dispositivos de seguridad (Firewall, Proxies). **F4.** Implementa conexiones en capa 2 o 3 para transporte de protocolos distintos de IP. **F5.** Se puede implementar en redes NATeadas. **F6.** Soporte para IPs dinámicas. **F7.** Facilidad de instalación e implementación. **F8.** Utilización de tecnologías estandarizadas.	**D1.** No compatible con IPsec. **D2.** Disponible solo para instalación en computadores. **D3.** Poco conocimiento sobre su utilización. **D4**. Falta de mejores Interfaces gráficas.

	F9. Software libre, bajo licencia GPL. **F10.** Gran cantidad de información en la web.	
LISTA DE OPORTUNIDADES **O1.** Empresas desarrolladoras de hardware de comunicación y seguridad. **O2.** Empresas comerciales interesadas en implementar este tipo de solución VPN. **O3.** De uso libre, permitiendo así una mayor difusión y conocimiento del producto. **O4.** Grancantidad de desarrolladores.	ESTRATEGIAS PARA MAXIMIZAR LAS **F** Y **O** 1. Instalar Clientes OPENVPN en distintos dispositivos (Switches, router, Firewall) (F7+F9+O1+O2+O4) 2. Convertirlo en un estándar (F8+O1+O2+O4) 3. Desarrollar el soporte para transmisión de audio y video y mejorar las interfaces gráficas. (O1+O2+O4/D2+D5)	ESTRATEGIAS PARA MAXIMIZAR LAS **O** Y MINIMIZAR LAS **D** 1. Incorporarlo en los distintos dispositivos de red (O1+O3/D2) 2. Realizar cursos sobre el manejo de dicha aplicación (O3+O4/D3). 3. Desarrollar mejores interfaces gráficas. (O1+O2+O4/D4)
LISTA DE AMENAZAS **A1.** Desarrollo de mejores soluciones VPN. **A2.** Falta de aceptación del producto.	ESTRATEGIAS PARA MAXIMIZAR LAS **F** Y MINIMIZAR LAS **A** 1. Mejoramiento constante de las actualizaciones y BD de conocimiento. (**TF / TA)**	ESTRATEGIAS PARA MINIMIZAR LAS **D** Y **A** 1. Mejoramiento constante del producto. (**TA/ D1+D2+D3+D5)** 2. Mayor publicidad y capacitación sobre la aplicación. (**D4+A2**)

3. IMPLEMENTACIÓN DE LA RED PRIVADA VIRTUAL

Para empezar con esta implementación primero definiremos el escenario sobre el cual se desarrollará la implementación, así como los requisitos necesarios para su montaje.

Nota: Aclarar que todas las imágenes presentadas en este capítulo, a excepción de la siguiente, corresponden a capturas de pantallas tomadas durante el montaje de la implementación.

3.1 ESCENARIO

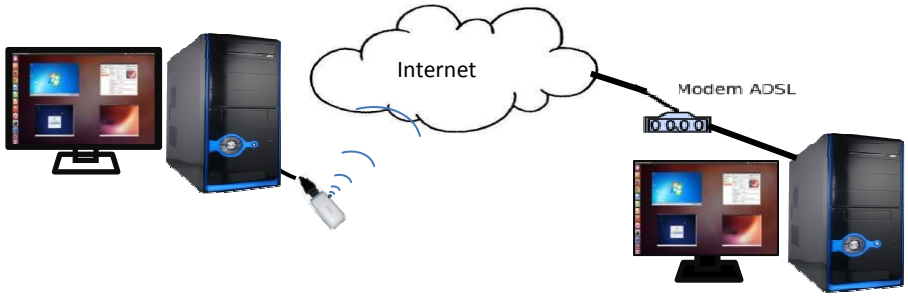
Esta implementación considera el montaje de una red privada virtual "sitio a sitio" para conectar dos sedes (A y B) distantes de forma segura, las cuales pertenecen a la empresa PST-Experts. Advertir que esta compañía es ficticia y sus sedes serán simuladas en dos equipos mediante software de virtualización.

Una oficina ubicada en un lugar remoto con acceso a internet mediante tecnología 3.5G (modem USB), ya que no existe ningún otro ISP que provea otro tipo de enlace. La conexión a internet es de 1,5 Mbps aproximadamente, con direccionamiento IP dinámico.
La oficina cuenta con un computador, el cual se utilizará como servidor VPN y que posee las siguientes características: procesador core i7-2600 de 3.4 a 3.8 GHz, tarjeta video ATI de 1Gb, memoria RAM de 8Gb y un disco duro SATA de 1,5 Tb. En el equipo se encuentra instalado el sistema operativo Ubuntu Server 12.04 LTS con entorno gráfico, el software OpenVPN y el software de virtualización VirtualBox v 4.2 que se utilizará para simular una pequeña red LAN, la cual contará con tres equipos virtualizados: Centos 6.3, el cual actuará como servidor web y DNS, Windows 7 ultimate y Ubuntu desktop 13, los cuales simularan dos estaciones de trabajo en la red. Esta red cuenta con el siguiente direccionamiento IP: 192.168.25.0

El equipo que simula la otra sucursal cuanta con características similares al ya descrito, excepto en que solo cuenta con una máquina virtual y el direccionamiento es 192.168.1.0

La siguiente imagen dará una visión más clara de lo que se pretende lograr:

Figura 4. Escenario planteado.

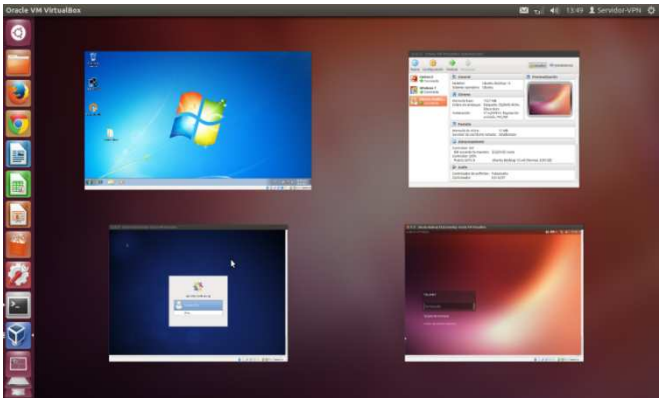


3.2 CONFIGURACIÓN DE LA RED VIRTUAL

En este apartado mencionaremos como se realizó la configuración del entorno virtual para simular la red interna de las sedes A y B.

1. Instalación de los diferentes sistemas operativos en Virtualbox.

Figura 5. Máquinas virtuales instaladas.



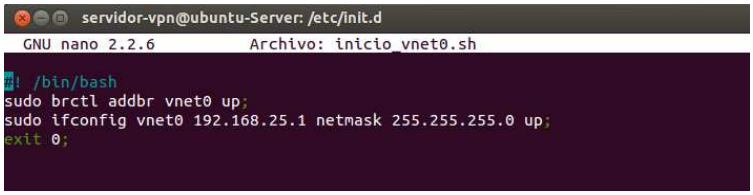
2. Desde el sistema anfitrión se virtualiza una tarjeta de red para que las máquinas virtuales reconozcan dos adaptadores de red, el físico y el virtual. Para ello digitamos el siguiente comando en la terminal del servidor VPN:

   $ Sudo brctl addbr vnet0

   $ ifconfig vnet0 192.168.25.1 netmask 255.255.255.0

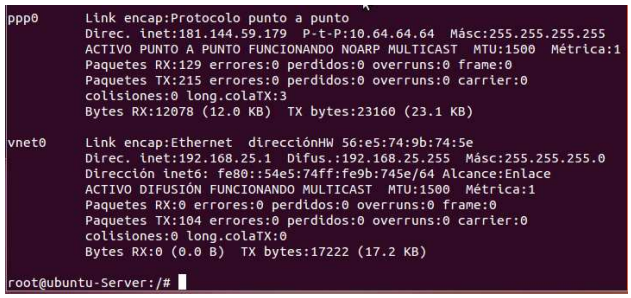
Debido a que esta configuración se borra cuando se reinicia la máquina, es necesario crear un script, dentro de la carpeta */etc/init.d*, que ejecute estas instrucciones al inicio del sistema, tal y como el que se muestra en la imagen.

Figura 6. Script creación automática interfaz *vnet0*.



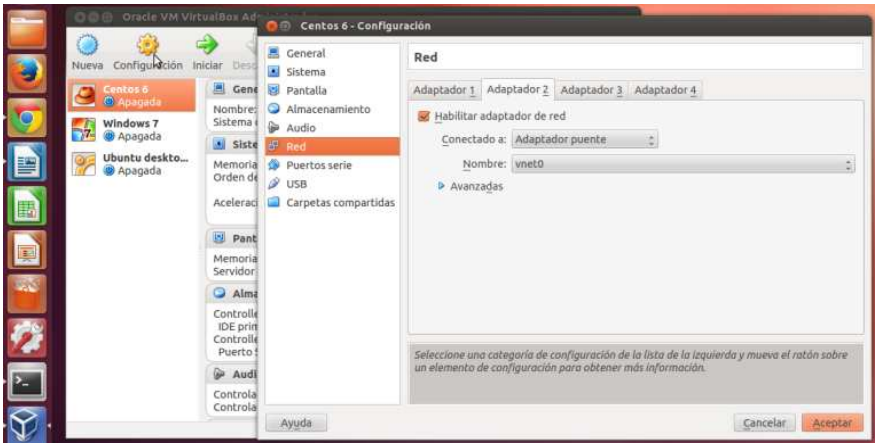
Después de realizado este procedimiento comprobamos que se haya creado una nueva interfaz de red vnet0, utilizando el comando *ifconfig*.

Figura 7. Visualización de la creación de la interfaz *vnet0*.



3. Verificada la existencia de la interfaz vnet0, abrimos la aplicación VirtualBox y procedemos a seleccionar cada una de las máquinas virtuales instaladas y a indicarles que trabajen con dicha interfaz de red para ello: seleccione una de las máquinas virtuales instaladas, seguidamente vaya a la opción configuración, seleccione la opción Red, luego en la pestaña adaptador en la opción interfaz, seleccione en la opción *conectado a* la alternativa *Adaptador* puente y en la opción Nombre elija vnet0. Tal y como lo muestra la imagen siguiente:

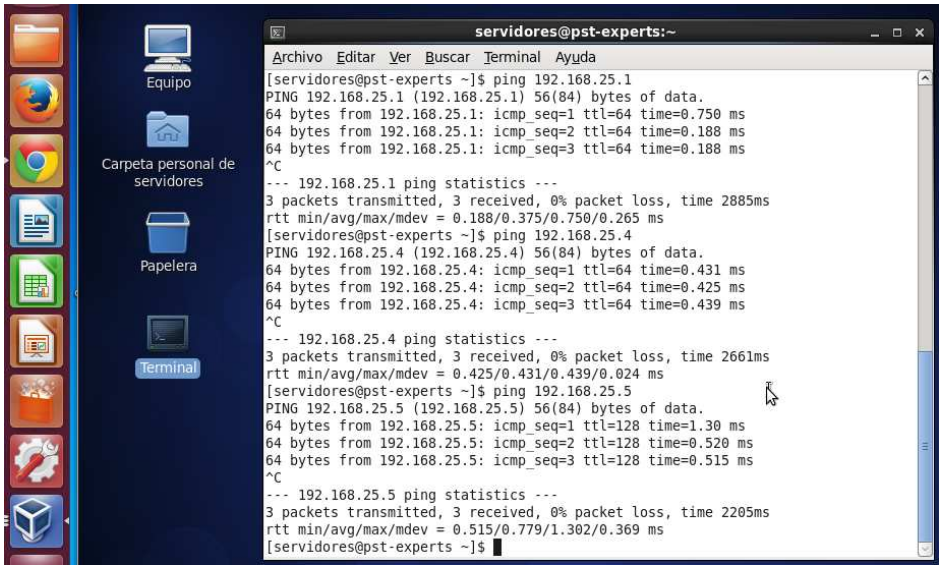
Figura 8. Asignar interfaz *venet0* a las máquinas virtuales.



4. Por último configure en cada sistema las direcciones de red que desee que lleven cada una de las máquinas.

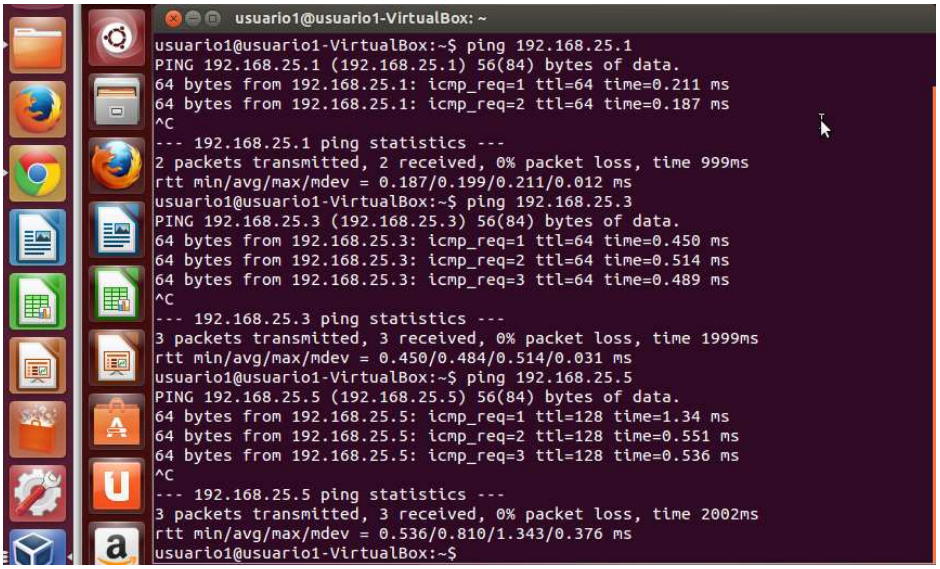
5. Comprobación de la conexión mediante ping.

Figura 9. Comprobación de la conexión entre las diferentes máquinas virtuales.



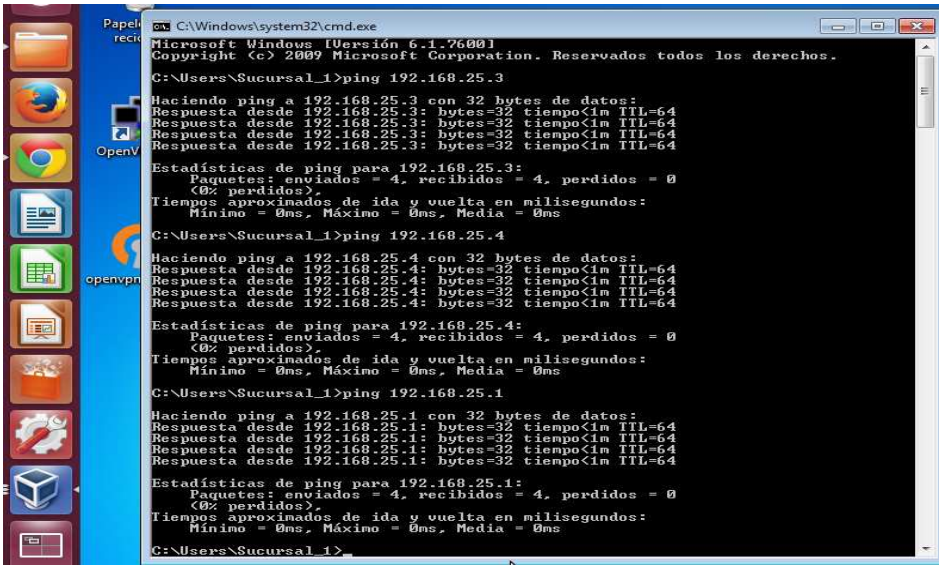
Ping desde MV Centos a toda la red

Figura 10. Comprobación de la conexión entre las diferentes máquinas virtuales.



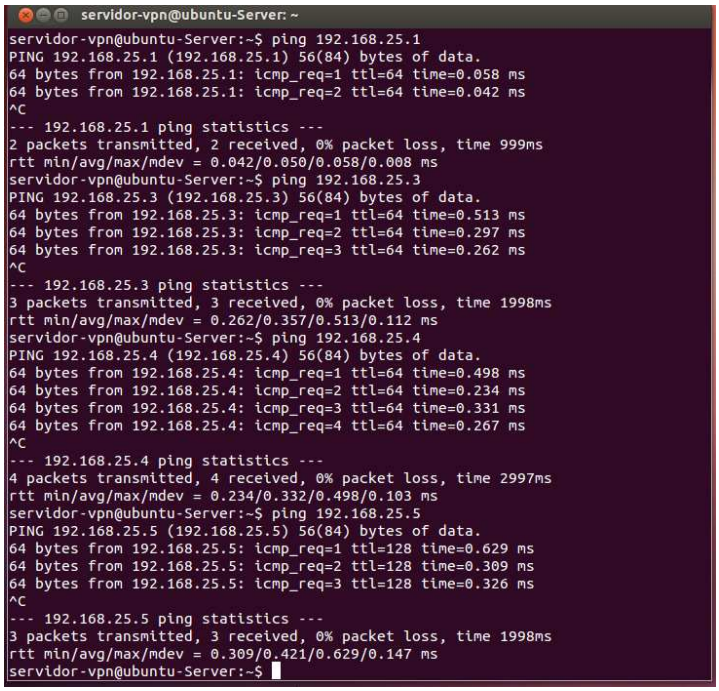
Ping desde MV Ubuntu 13

Figura 11. Comprobación de la conexión entre las diferentes máquinas virtuales.



Ping desde MV Windows a la red

Figura 12. Comprobación de la conexión entre las diferentes máquinas virtuales.



Ping desde equipo anfitrión a todas las demás.

3.3 CONFIGURACIÓN DE OPENVPN (41)

De aquí en adelante se describirán las diferentes acciones que se emplearon para la instalación y configuración de la red privada virtual con OpenVPN.

3.3.1 Instalación de OpenVPN en el servidor

Como ya se mencionó, el sistema operativo sobre el cual se trabajará e instalará OpenVPN es Ubuntu Server 12.04 LTS, sin embargo los pasos son similares y se pueden adaptar a cualquier otra distribución GNU/Linux sin mayores inconvenientes, solo se debe tener en cuenta reemplazar los comandos por los comandos propios para cada sistema.

Antes de empezar con la instalación, primero debemos observar que el sistema cumpla con algunos requisitos necesarios para poder ejecutar OpenVPN de forma adecuada.

- Privilegios de root: tanto la instalación como la configuración de OpenVPN deben hacerse con estos privilegios ya que el programa requiere abrir puertos UDP/TCP y cargar el soporte TUN.

- Drivers TUN/TAP: Son necesarios para la creación de los túneles. En las distribuciones Ubuntu 10.04 y superiores estos drivers ya vienen incluidos dentro del Kernel, por lo que ya no se hace necesario compilarlos, ni preocuparse de que se carguen al inicio del sistema.

- Conexión a internet.

Después de autenticarnos como root podemos realizar la descarga de la aplicación OpenVPN, para ello abra una terminal y digite el siguiente comando:

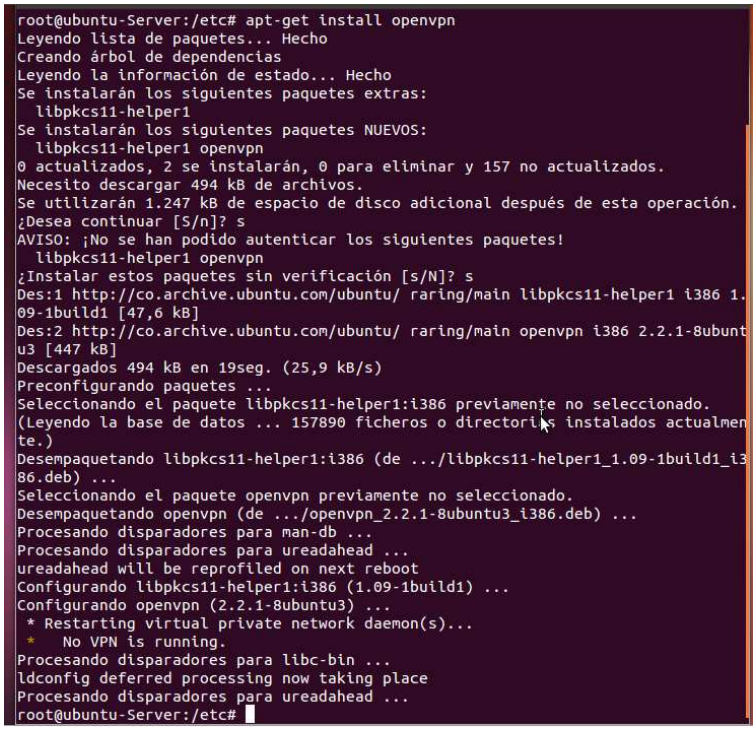
 # apt-get install openvpn

Comenzará la instalación del programa junto con todas las dependencias necesarias para su funcionamiento, observe la imagen que se presenta a continuación.

Cuando se termina de instalar el paquete de OpenVPN, se instalan junto con este las siguientes librerías:

- **Libssl:** Es la librería encargada de proporcionar el paquete OpenSSL, el cual es necesario para brindar el cifrado y seguridad a los paquetes que viajan por el túnel.

Figura 13. Instalación de OpenVPN.



- **liblzo2:** Instala el paquete LZO que es el encargado de brindar la compresión a los paquetes y lograr una mayor eficiencia en el transporte de los mismos.

OpenVPN también necesita de las librerías **libpam0g**, **libpam-modules** y **libpam-runtime**, las cuales instalan el paquete PAM, el cual es necesario si decide crear mecanismos de autenticación adicionales como solicitud de usuario y contraseña para usuario que se conecte a la VPN. Al instalar OpenVPN no se instalan estas librerías, sin embargo en Ubuntu estos paquetes están incluidas por defecto, así que no hay nada de que preocuparse. Para saber si todas las dependencias necesarias para el buen funcionamiento de OpenVPN han sido instaladas puede utilizar el comando:

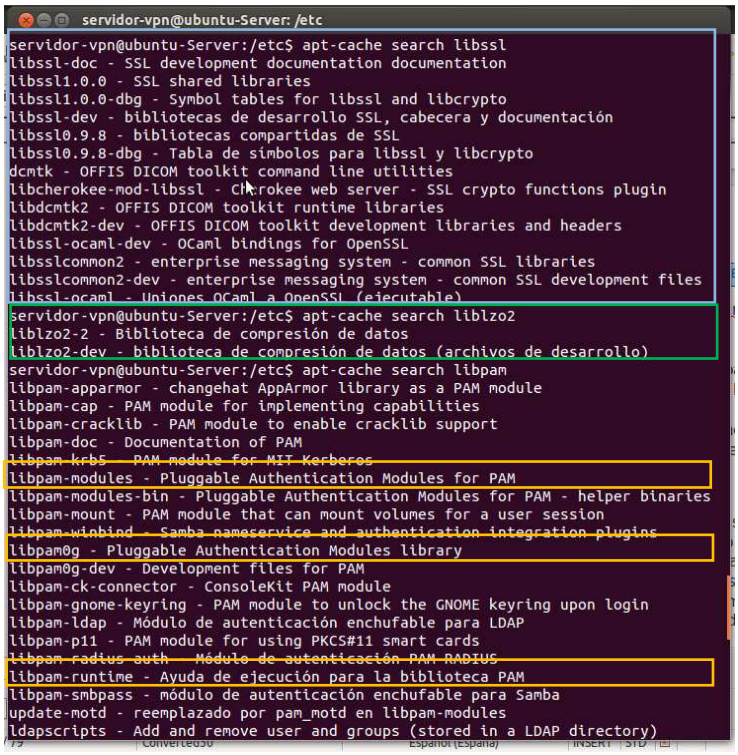
apt-cache search **nombre_del_paquete**

En la siguiente imagen se aprecia la verificación de todas las dependencias necesarias.

Cuando termina la instalación de OpenVPN, se agrega el *servicioopenvpn* para que este sea iniciado automáticamente al arranque del sistema; si no deseamos que el servidor OpenVPN sea iniciado al arranque del sistema usaremos el comando:

 # update-rc.d -f openvpn remove

Figura 14. Comprobación de dependencias del programa OpenVPN.



Como la instalación crea un *servicio openvpn*, podemos, como a cualquier servicio, detenerlo o iniciarlo, para este caso se usará el script de control de ejecución /etc/init.d/openvpn.

El script requiere que exista un archivo de configuración de OpenVPN en el directorio /etc/openvpn con terminación .conf, por ejemplo /etc/openvpn/servidor.conf (la creación de este archivo será detallada más adelante).

Para iniciar el servicio de OpenVPN use el comando:

 # /etc/init.d/openvpn start o *#service openvpn start*

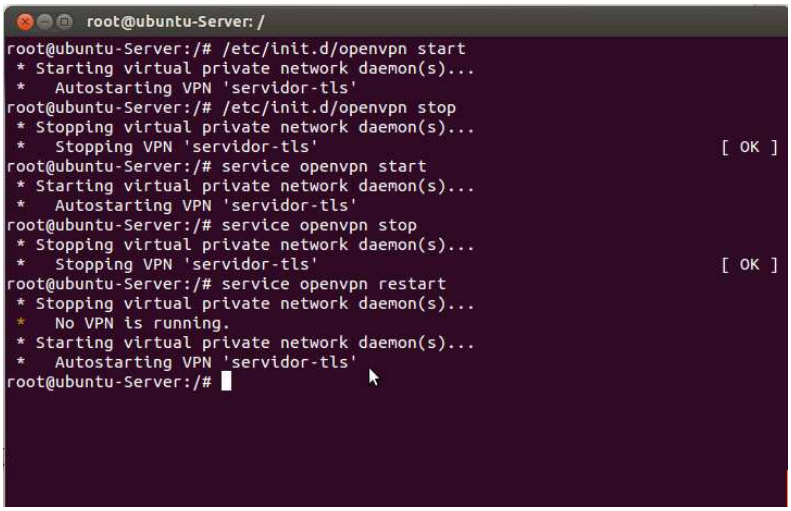
Para detener el servicio de OpenVPN use el comando:

*# /etc/init.d/openvpn stop*o *#service openvpn stop*

Nota: el script ejecuta el programa openvpn con el parametro --daemon por cada archivo de configuración en el directorio /etc/openvpn/ con terminación .conf, el proceso openvpn se ejecuta en segundo plano.

A continuación se presenta una imagen de lo que sucede cuando ejecuta los comandos mencionados.

Figura 15. Ejecución de del servicio openvpn mediante comandos.



Si se realizan cambios significativos en la configuración del servidor o cliente OpenVPN (archivos *.conf*), como cambio de dirección IP, protocolo, puertos o directorios de configuraciones, se aconseja reiniciar por completo el servicio OpenVPN con el comando:

 # /etc/init.d/openvpn restart o *#service openvpn restart*

Si está realizando pruebas en las configuraciones o conexiones VPN se recomienda que ejecute el programa OpenVPN en primer plano para que de esta manera pueda localizar mensajes importantes o de error con mayor rapidez, por ejemplo:

 # openvpn /etc/openvpn/hosta-debug.conf

Si se ha ejecutado OpenVPN en primer plano, teclee *Ctrl+C* para detener la ejecución.
OpenVPN crea durante la instalación el directorio /etc/openvpn en donde se recomienda guardar todos los archivos referentes a la configuración de esta aplicación.

3.3.2 Configuración del servidor VPN

Un aspecto a tener en cuenta después de instalar OpenVPN es habilitar en el firewall los puertos que se utilizan para realizar las conexiones, OpenVPN utiliza por defecto el puerto UDP 1194, oficialmente aprobado por *IANA*, es necesario configurar dicho firewall para que permita conexiones por ese puerto.

Como Ubuntu trae por defecto un firewall denominado IPtables, utilizamos la siguiente instrucción para permitir conexiones al puerto 1194 desde cualquier origen.

 # iptables -A INPUT -i ppp0 -p upd -s 0.0.0.0/0 --dport 1194 -j ACCEPT

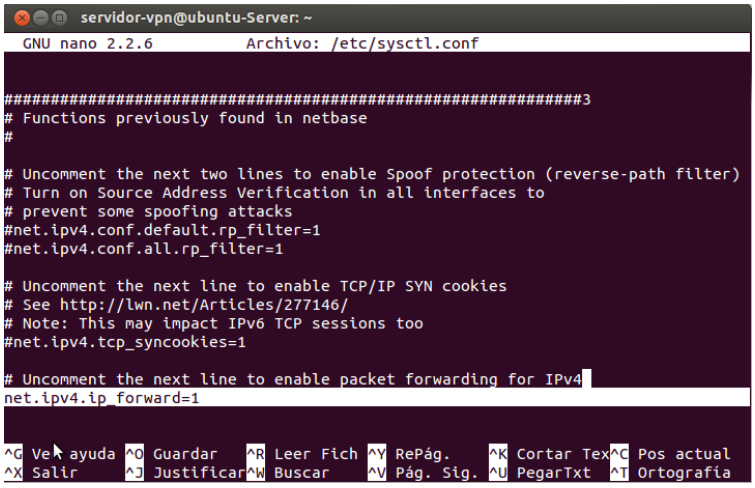
Nota: En nuestro caso se utiliza la interfaz ppp0 ya que es la que nos permite el acceso a Internet, si la conexión que se maneja es diferente, pues simplemente reemplazamos ppp0 por eth0 en caso de una tarjeta de red.

Otro requerimiento a recordar es el enrutamiento para las conexiones VPN, para ello se debe NAT de tipo Forward hacia el servidor OpenVPN, para activar el forwarding de manera que siempre quede activo, modificamos el archivo */etc/sysctl.conf*, para ello digite:

 # nano /etc/sysctl.conf

Se ubica la línea *#net.ipv4.ip_forward=1* y se descomenta (se borra el símbolo #), debe quedar como aparece en la imagen.

Figura 16. Activación permanente del ruteo forwarding.



OpenVPn soporta diferentes tipos de autenticación, desde el método de llaves secretas precompartidas (modo Statik Key), utilizadas muy a menudo en conexiones punto a punto, hasta la técnica de llaves públicas (modo SSL/TLS) a través del uso de certificados X.509 para autenticar servidores y clientes, este método es usado en conexiones sitio a sitio.

En este proyecto se usará el método de llaves públicas y certificados para realizar la autenticación de los usuarios a la VPN.

Cuando se instala OpenVPN, se crea la carpeta */etc/openvpn*, esta carpeta contiene un único archivo llamado *update-resolv-conf*, para comenzar con la configuración del servidor nos ubicamos en esta carpeta, luego creamos una carpeta donde copiaremos los archivos necesarios para la creación de autoridades certificadoras y llaves públicas, nombre a esta carpeta como se desee, para esta implementación se nombrará *Certificados_AC*, para lo anterior digite:

 # cd /etc/openvpn

 # mkdir Certificados_AC

En la instalación OpenVPN crea otra carpeta en /usr/share/doc/openvpn/*examples/easy-rsa* que contiene todo lo necesario para trabajar, dentro de este directorio encontramos dos carpetas, una llamada 1.0 y la otra 2.0, estas son las versiones de los scripts que incorpora OpenVPN para la creación y manejo de certificados y autoridades certificadoras. Lo que se debe hacer es copiar los archivos contenidos en la carpeta 2.0 a la carpeta que se creó anteriormente, digite:

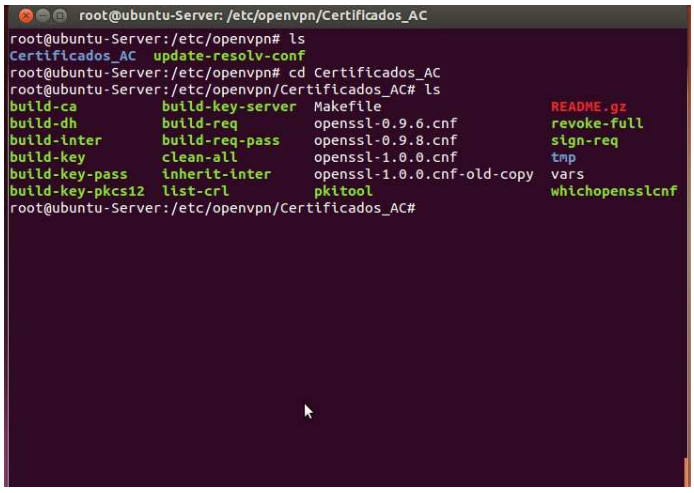
cp -va /usr/share/doc/openvpn/examples/easy-rsa/2.0/* etc/openvpn/Certificados_AC

Hecho esto, nos desplazamos a este directorio

/openvpn # cd Certificados_AC

A continuación se presenta una imagen con el contenido de esta carpeta después de copiar los archivos.

Figura 17. Contenido directorio openvpn y Certificados_AC.



3.3.2.1 Configuración de la autoridad certificadora (AC)

Dentro del directorio *Certificados_CA* ubicamos el archivo *vars*, este archivo contiene los parámetros globales para la generación y administración de la AC. Editaremos este archivo para definir los parámetros que necesitará la autoridad certificadora para funcionar.

*/Certificados_AC # **nano** vars*

Se ha utilizado *nano* como el editor para este trabajo, se puede utilizar cualquier otro editor como *vi* o *vim* o el que se prefiera.

Un parámetro importante en este archivo es *export KEY_CONFIG*, el cual nos permite elegir la ubicación del archivo *openssl* con el que se va a trabajar, en este caso se recomienda utilizar el archivo *openssl-1.0.0.cnf*, por tanto debemos modificar esta línea para que quede de la siguiente manera:

Export KEY_CONFIG=//etc/openvpn/Certificados_AC/openssl-1.0.0.cnf

Otro parámetro que es necesario revisar es KEY_DIR, el cual apunta al directorio en donde residen todos los archivos de los certificados. Es en este directorio donde se almacenaran las llaves privadas (.key), los archivos de solicitud de certificado (.csr), los certificados (.crt) y otros archivos como el serial y el index.txt. La variable KEY_DIR apunta por defecto al directorio /etc/openvpn/ExamplesCA/keys, pero en este proyecto se ha cambiado por:

export KEY_DIR="$EASY_RSA/keys"

Es bueno aclarar que EASY_RSA es una variable que apunta al directorio actual, por lo que es importante cambiarse al directorio para que los scripts localicen el directorio de las llaves.

Cuando un certificados es emitido se le asigna una fecha de emisión y una fecha de expiración, la fecha de expiración del certificado raíz de la AC está definidapor la variable CA_EXPIRE, por defecto viene configurada para que el certificado dure 10 años (3650 días), si desea definir otra fecha, mayor o menor, edite esta variable; por ejemplo que el certificad expire en 20 años:

export CA_EXPIRE=7300

El tiempo de expiración para los certificados de servidores y clientes se definen en la variable KEY_EXPIRE, por defecto los certificados generados para clientes tendrán un tiempo de expiraciónde 10 años, sin embargo, es conveniente disminuir este tiempo para

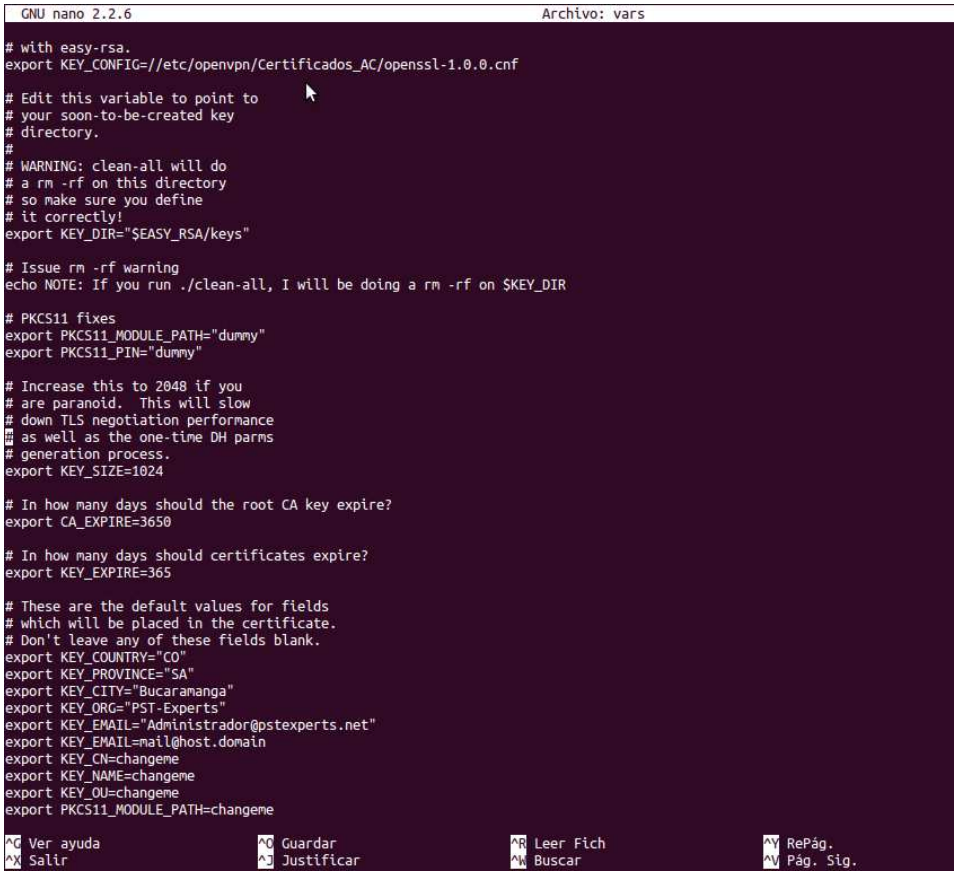
tener mayor control sobre estos clientes; ejemplo, certificado valido por un año (365 días):

export KEY_EXPIRE=365

Hecho lo anterior, pasamos a configurar los parámetros de nuestra AC, estos parámetros van desde *export KEY_COUNTRY* hasta *export KEY_EMAIL*, cambie los valores que traen por defecto, por los datos que representen a la autoridad certificadora.

Para dar a entender un poco mejor lo que se debe hacer, en la siguiente imagen se presenta la configuración que se le ha dado al archivo *vars* de este trabajo.

Figura 18. Contenido y configuración del archivo vars.



Una vez terminada la configuración del archivo guarde los cambios y salga del editor, volverá al prompt del sistema; para que los cambios surtan efecto se deben exportar las variables definidas en el archivo *vars* al Shell del sistema, esto debido a que los scripts utilizados por easy_rsa usan los valores establecidos dentro de estas variables. Para realizar esta operación existen dos formas:

1. Ejecute el comando *source vars*.

   /Certificados_AC # source vars

2. O digite la siguiente instrucción *. ./vars* (punto, espacio, punto y slash)

   /Certificados_AC #. ./vars

Realizado lo anterior se muestra un mensaje solicitando que se ejecute el script clean-all, digite lo siguiente:

/Certificados_AC #./clean-all (un punto y seguido slash)

Al ejecutar este script lo que se hace, y como se podrá suponer, es borrar todos los archivos y certificados existentes, para comenzar con un entorno limpio, además crea el directorio *keys*, el cual contiene los archivos *index.txt* y *serial*.

- *Keys*: es el directorio en el cual se almacenan las llaves y certificados.
- *Index.txt*: es el archivo en el cual se registra el índice de archivos creados y su estado.
- *Serial*: es el archivo en el que se guarde el número de serie del último certificado creado.

Nota: después de crear los certificados para la AC, el servidor y los clientes no vuelva a ejecutar el script clean-all, ya que borrará todos los archivos creados; a menos que quiera volver a configurar otros certificados o que el procedimiento anterior le haya fallado.

Ejecutado todo esto, procedemos a inicializar la autoridad certificadora, ejecutando el script *pkitool*, este script lo ejecutamos junto con los argumentos *–initca* y *–pass* (opcional), la opción *pass* solicita se introduzca una clave, esto con el fin de fortalecer la seguridad de la llave privada raíz con una clave de paso (*passphrase*).

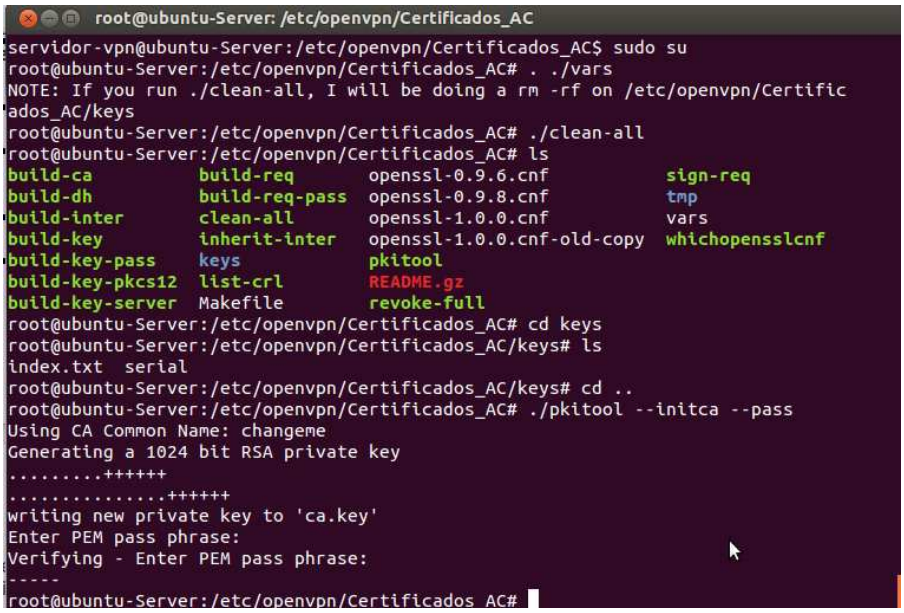
/Certificados_AC # ./pkitool –initca –pass (un punto y seguido slash)

Si ejecutó el script con la opción *–pass*, se le solicitara que introduzca una clave, asegúrese de que esta clave sea lo suficientemente segura ´para garantizar que no sea adivinada o decodificada fácilmente.

Nota: recuerde muy bien esta clave, ya que será solicitada al momento de crear o revocar certificados para clientes y servidores.

A continuación se presenta una imagen que condensa todo lo que se ha explicado hasta el momento:

Figura 19. Inicialización de la autoridad certificadora (AC).



El script *pkitool* crea dos archivos en el directorio *keys*, *ca.key* y *ca.crt*

- *ca.key*: es el archivo de la llave privada raíz, la cual está protegida por el *pass phrase*; es de caracter privado y por tanto debe permanecer únicamente en el servidor VPN, en un lugar seguro, ya que si fuera hurtado pondría en riesgo la seguridad de la red privada virtual y de toda la infraestructura de llave pública.

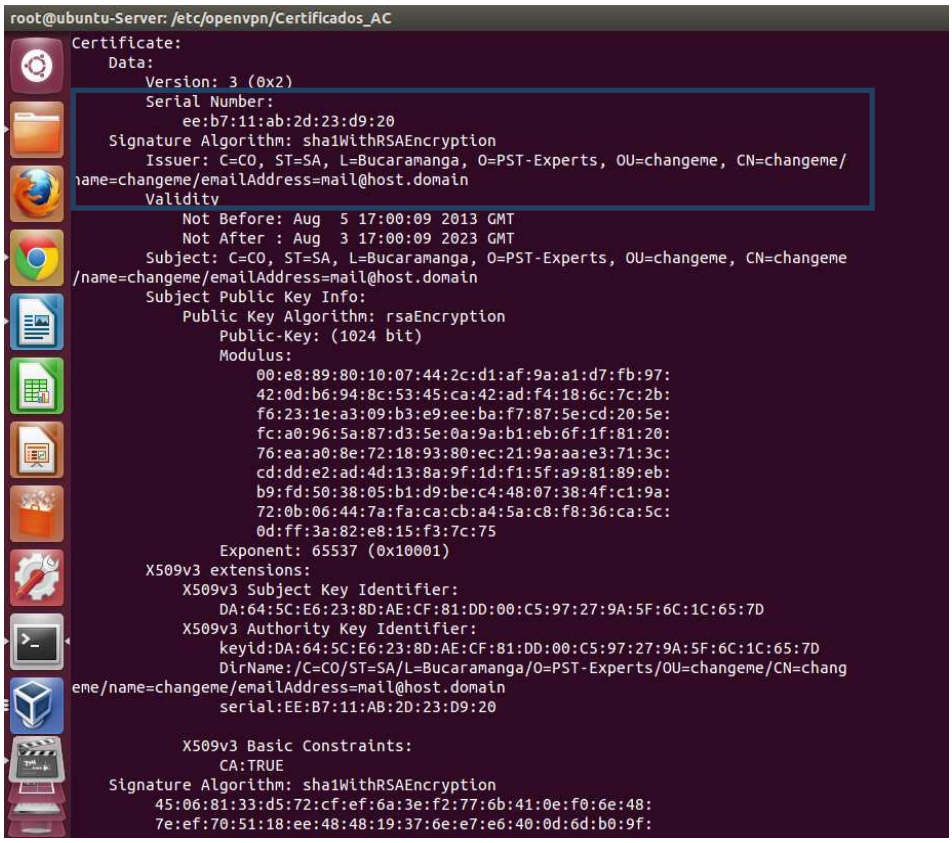
- *ca.crt*: es el archivo del Certificado Raíz de la AC, este archivo es de carácter público y debe ser distribuido a todos los miembros de la VPN, este certificado será usado por los clientes y servidores VPN para validar la autenticidad de las conexiones.

Podemos ver el contenido del certificado raíz usando comandos *oppenssl*, para este caso digiete:

/Certificados_AC # openssl x509 -in keys/ca.crt -noout -text

Se nos muestra el contenido del certificado, tal y como aparece en la siguiente imagen:

Figura 20. Certificado raíz de la AC.



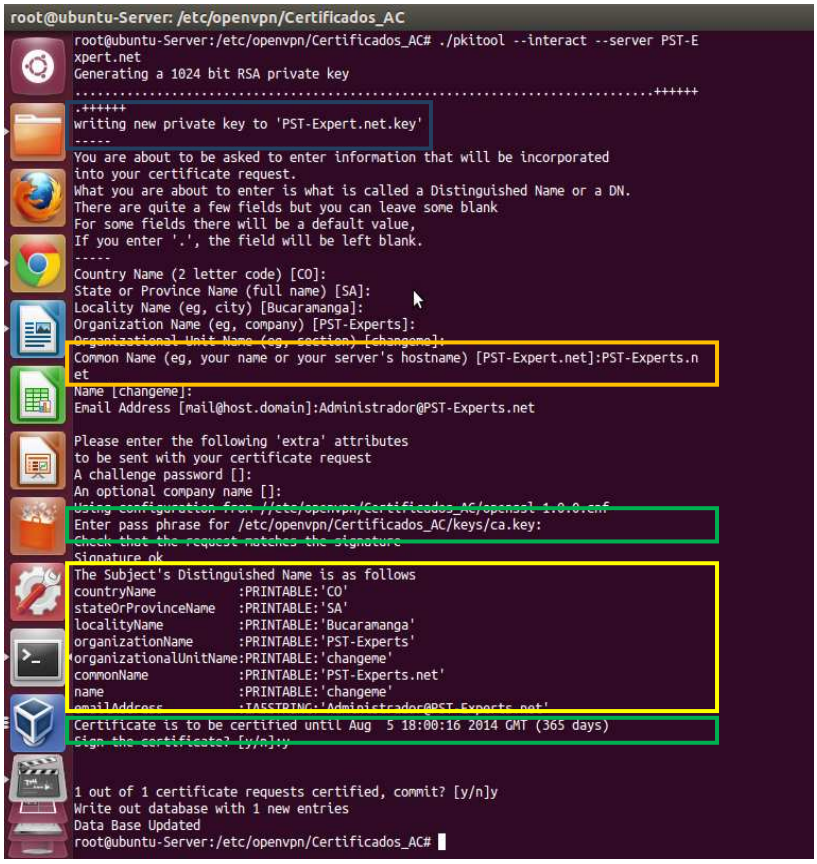
Observe la información seleccionada, en esta se indican los parámetros introducidos para la AC y las fechas de emisión y expiración del certificado, como se aprecia el certificado tiene una duración de 10 años.

3.3.2.2 Generación de la llave privada y el certificado para el servidor OpenVPN

Para este procedimiento también se usará el script *pkitool*, pero con la opción *–interact* (modo interactivo) y la opción *–server* (indica que es el certificado para el servidor), seguido del nombre que se dará al servidor y al certificado, en este caso se escogió PST-Experts.net por ser la empresa a la cual se le está creando la red privada virtual. Por tanto digite lo siguiente y observe la imagen.

/Certificados_AC # ./pkitool --interact --server ***PST-Experts.net***

Figura 21. Generación del certificado para el servidor OpenVPN.



Mire como se especifica que se va a crear una nueva llave privada para *PST-Experts.net*; luego se solicitara que introduzca algunos valores para la generación del certificado, estos argumentos son tomados del archivo *vars*, por tanto, cambie solo aquellos valores que necesiten ser cambiados. Note como el campo *Common Name* (CN) toma el nombre *PST-Experts.net* (el que se escribió en la ejecución del script)como valor.

Dentro de esta generación del certificado, preste atención a la parte donde se solicita el *pass phrase,* que es el password que se dio cuando se creó el certificado de la AC, si no recuerda esta contraseña, no podrá continuar con la generación del certificado para el servidor. Note también la duración de este certificado, la cual es de 1 año.

Si se decide expedir un certificado con fecha de expiración de un año o menor, se debe tener esto en cuenta para renovar el certificado antes de que este finalice, ya que de lo contrario los clientes quedaran sin conexión a la VPN.

Por último se realizan dos preguntas a las cuales se debe responder que sí (y), claro está, si la información que se introdujo es correcta, si se tiene duda digite N (no), si responde sí,

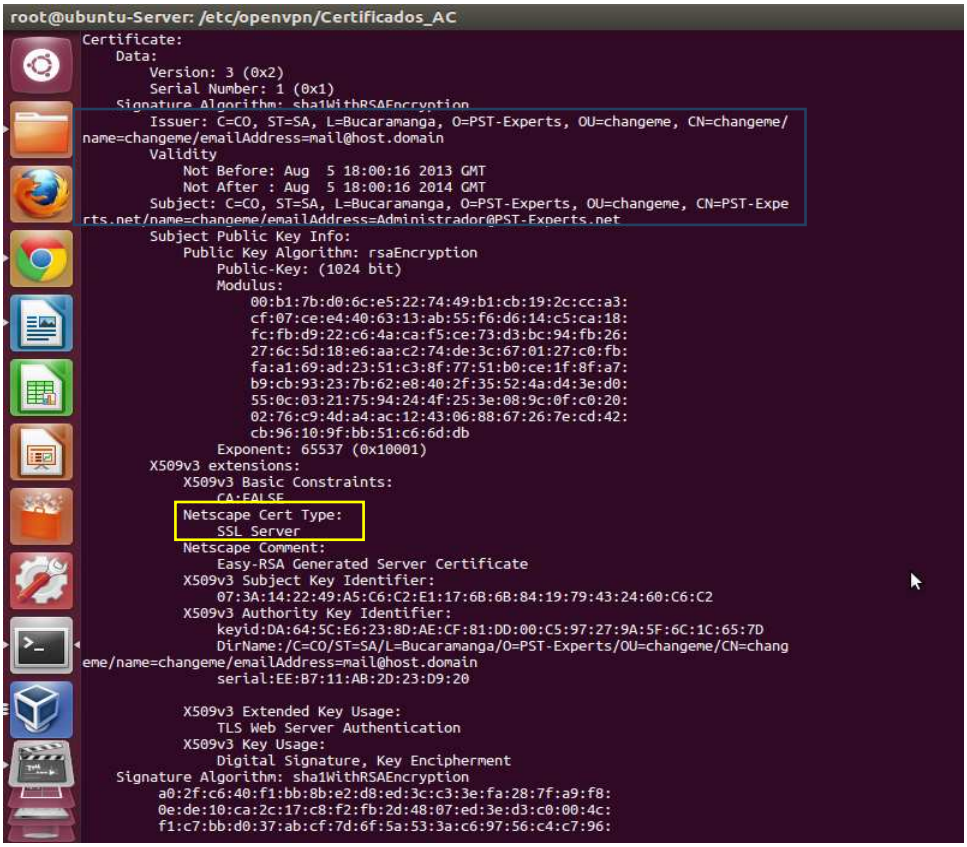
aparece un mensaje diciendo que se creó y actualizo un nuevo registro en la base de datos.

La ejecución de este script crea cuatro archivos dentro de la carpeta *keys*, estos son: *PST-Experts.net.csr*, que es el archivo de petición del certificado (Certificate Signing Request), *PST-Experts.net.crt*, que es el fichero de certificado (llave pública),*PST-Experts.net*.key, que es el archivo de llave privada y el archivo *01.pem*, el cual es el fichero correspondiente al certificado público del servidor en formato PEM. El nombre del fichero proviene de su número de serie del certificado, el cual es 01.

Al igual que con el certificado raíz, también podemos ver el contenido del certificado del servidor, para ello digite el siguiente comando y observe la imagen donde se muestra dicho resultado:

/Certificados_AC #openssl x509 -noout -text -in keys/PST-Experts.net.crt

Figura 22. Contenido del certificado del servidor.



Si observamos el certificado de la AC y el del servidor, notamos que son similares en su estructura, sin embargo en el certificado del servidor los argumentos del campo *Subject*cambian, al menos en lo que al *CN* se refiere y aparecen unos nuevos valores, de los cuales se resalta uno, que será importante para reforzar la seguridad de la VPN.

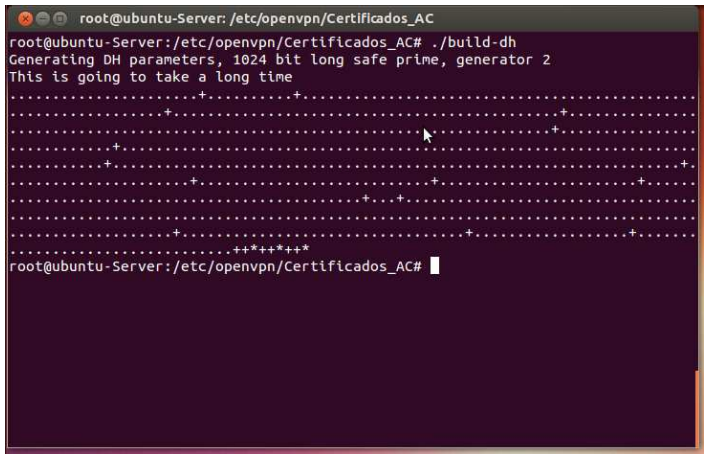
3.3.2.3 Generación del cripto-sistema de clave pública

Para que se pueda establecer un canal seguro, se debe crear en el servidor un método que permita compartir las claves de forma segura, sin la necesidad de que tanto el servidor como el cliente tengan una clave precompartida. Para ello se emplea el algoritmo *Diffie-Hellman*.
Este algoritmo permite establecer una clave secreta entre dos partes desconocidas, que desean comunicarse de forma secreta a través de un canal inseguro. La clave secreta establecida puede ser usada para cifrar comunicaciones subsecuentes mediante algoritmos de cifrado simétrico.

Para generar el algoritmo se utiliza el script *build-dh*, que crea un archivo con los parámetros *Diffie-Hellman*, para ello digite lo siguiente y observe la imagen que muestra el proceso.

/Certificados_AC # ./build-dh

Figura 23. Generación archivo Diffie-Hellman.



El archivo creado por este proceso es exclusivo para uso del servidor, por tanto nunca debe copiarse a los clientes.

Terminado estos procesos deberá copiar los siguientes archivos, *ca.crt*, *PST-Experts.net.key*, *PST-Experts.net.crt* y el archivo *dh1024.pem* a la carpeta */etc/openvpn*, por tanto digite el siguiente comando:

/Certificados_AC #cp keys/ca.crt keys/PST-Experts.net.{crt,key} keys/dh1024.pem /etc/openvpn/

OpenVPN crea los archivos de llave privadas con permisos de ejecución *600*, se recomienda que estos archivos permanezcan con estos privilegios, con el fin de que ningún otro usuario tenga acceso a estos archivos.

3.3.2.4 Generación de llaves privadas y certificados para los clientes

Cuando se está a punto de crear los certificados para los clientes de la red privada virtual, es recomendable y se podría decir que exigible, tener a mano la información de los clientes para los que se emitirán los certificados; como mínimo se deben tener los siguientes datos: *Nombre de la persona o equipo*, *Nombre de la sección (departamento) o empresa externa donde labora*, *correo electrónico* y *nombre del certificado* (Common Name). También se debe disponer del pass phrase del certificado raíz de la AC (si creo dicho certificado con ese argumento).

Para crear los certificados de los clientes se usa nuevamente el script *pkitool* con la opción *interact,* seguido del nombre que se dará al cliente,así que digite:

/Certificados_AC # ./pkitool –interact ***fabianguerra.PST-Experts.net***

Fíjese en no agregar la opción *server* al ejecutar el script, porque quedaría con dos archivos de servidor en vez de uno de cliente.

A medida que se ejecuta el script se preguntarán algunos datos sobre el cliente que se está creando, llene los campos solicitados con la información correspondiente. En la siguiente imagen se muestra la ejecución del script.

Observe que el proceso es similar al que se efectuó en la creación del certificado del servidor, lo único diferente es el nombre para quien se crea la llave y el nombre del nuevo certificado.

Al terminar la ejecución del script se crean cuatro archivos: *fabianguerra.PST-Experts.net.csr*, que es el archivo de petición del certificado (Certificate Signing Request), *fabianguerra.PST-Experts.net.crt*, que es el archivo de certificado (llave pública), *fabianguerra.PST-Experts.net*.key, que es el archivo de llave privada y *02.pem*.

Transfiera de forma segura al cliente los siguientes archivos *fabianguerra.PST-Experts.net.crt* y *fabianguerra.PST-Experts.net*.key, así como el archivo del certificado raíz

de la Autoridad Certificadora, *ca.crt*. El archivo *fabianguerra.PST-Experts.net.csr* no es necesario trasladarlo al cliente, puesto que no será utilizado por este para ningún fin.

Figura 24. Creación del certificado para el cliente.

Ejecute este script cada vez que desee crear un nuevo cliente. Para el caso de esta implementación se han creado otros dos clientes.

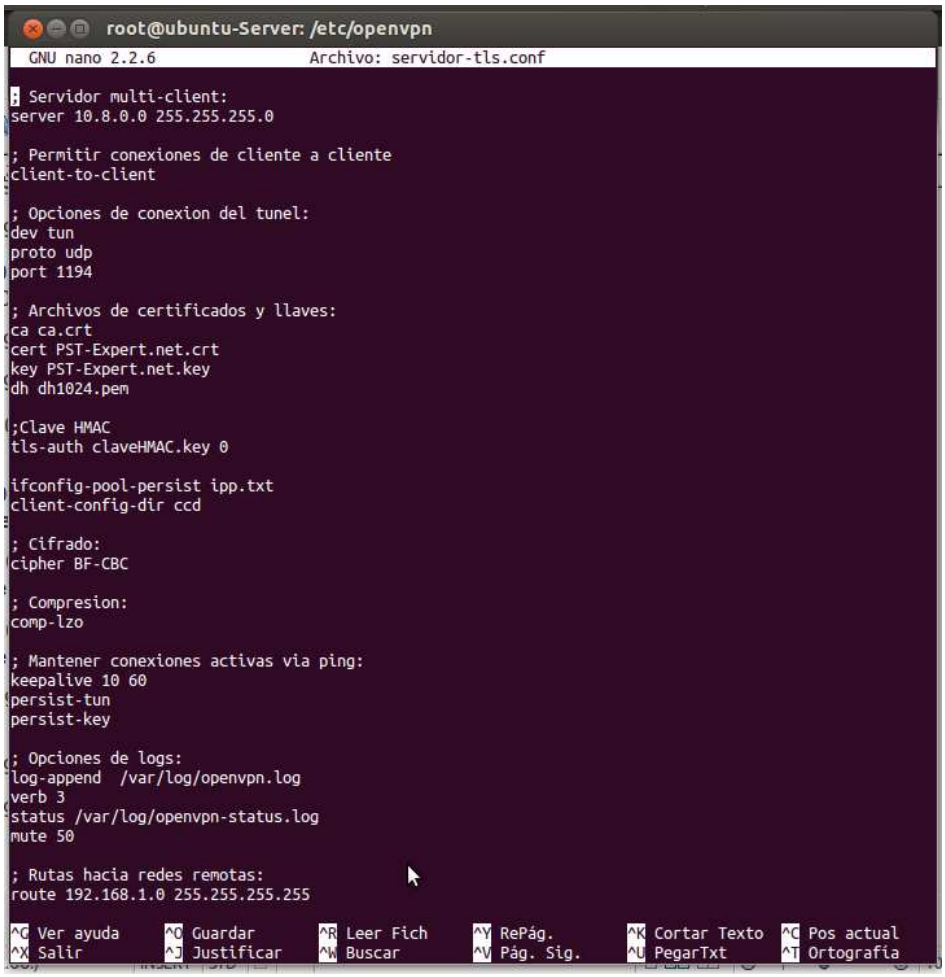
3.3.2.5 Creación del archivo de configuración para el servidor

Vamos a crear un archivo de configuración para una conexión VPN *sitio a* sitio, usando el sistema de autenticación basado en certificados y llaves públicas emitidos por la Autoridad Certificadora local, la que se creó.

La configuración del servidor OpenVPN para conexiones de sitio a sitio se realiza configurando OpenVPN en modo *server* (tls-server), este modo soporta la conexión de varios clientes TLS de forma simultánea, por lo que, con una sola instancia de OpenVPN en modo server se podrá conectar, no solamente, más de una oficina remota, sino también usuarios móviles (Road Warriors) para brindarles acceso seguro a los recursos internos de la empresa.

Para crear el archivo, estando en la carpeta */etc/openvpn*, se abre un editor de texto, el que se prefiera, se le da un nombre al archivo, el cual debe terminar en *.conf*, para el caso de este proyecto se le dio el nombre de *servidor-tls* y digite lo que se encuentra en la siguiente imagen, la cual muestra el archivo de configuración del servidor que se ha configurado para este proyecto.

Figura 25. Archivo para configuración del servidor OpenVPN.



A continuación se describirá como OpenVPN utiliza cada una de las instrucciones contenidas en el archivo.

- *Server 10.8.0.0 255.255.255.0*: indica que esta máquina va a actuar como un servidor y asignará direcciones IP en el rango de red de la VPN 10.8.0.0 a los clientes que se vayan conectando. Esta directiva simplifica la configuración del archivo, ya que condensa en esta sola instrucción el uso de muchas otras como: mode server, tls-server, ifconfig, entre otras.

- *client-to-client*: (opcional) se puede usar cuando se desea que los clientes se vean entre sí, lo que esta opción hace es enrutar el tráfico entre los clientes conectados a la VPN, en lugar de enviar todo el tráfico originado por el cliente a la interfaz TUN/TAP. Esta directiva suele utilizarse cuando OpenVPN está configurado en modo server, permitiéndole manejar múltiples clientes a través de una sola interfaz TUN, lo que lo convierte en un router.

- *dev tun*: indica el tipo de driver que se va a utilizar para realizar la conexión, aquí se puede utilizar o tun o tap dependiendo del tipo de enlace que se desee crear.

- *proto udp*: define el tipo de protocolo que se puede usar, OpenVPN permite utilizar tanto el protocolo UDP como TCP, es recomendable por cuestiones de seguridad y eficiencia utilizar el protocolo UDP

- *port 1194*: especifica el puerto que se desea utilizar, dado que OpenVPN utiliza el puerto 1194 para sus conexiones, es este el que se va a utilizar en este proyecto, sin embargo si se quiere escoger otro puerto se puede hacer, solo recuerde asignarlo también en el cliente y abrirlo en los respectivos firewalls.

- *ca ca.crt*: en este campo se debe colocar el certificado público de la Autoridad Certificadora.

- *cert PST-Experts.net.crt*: aquí se especifica el nombre dell certificado del Servidor OpenVPN.

- *key PST-Experts.net.key*: se escribe el nombre de la clave privada del Servidor OpenVPN.

- *dh dh1024.pem*: agrega los parámetros Diffie-Hellman.

- *tls-auth ClaveHMAC.key 0*: La directiva tls-auth agrega una firma HMAC adicional a todas las transacciones de paquetes SSL/TLS para verificar la integridad de los paquetes transmitidos entre el cliente y servidor OpenVPN. Esta instrucción se detallara más adelante en el apartado sobre seguridad.

- *ifconfig-pool-persist ipp.txt*: Se crea un archivo llamado “ipp.txt” en el que se registran las direcciones IP de los clientes que están conectados a la red privada virtual.

- *client-cofig-dir ccd*: Indica la ruta del *Directorio de Configuraciones de Clientes* (*Client Config Directory*), en este caso usaremos el directorio *ccd*, este directorio debe crearse en la misma carpeta donde se encuentra el archivo de configuración, es decir, en

/etc/openvpn, este directorio se utilizará para almacenar los archivos de configuración creados para los clientes.

- *Cipher BF-CBC*: especifica el tipo de cifrado que se va a utilizar, por defecto OpenVPN utiliza el cifrado Blowfish, más adelante se enunciaran los tipos de cifrado soportados por OpenVPN

- *comp-lzo*: indica que se va a utilizar la librería de compresión LZO.

- *keepalive 10 120*: esta opción sirve para comprobar la red, para lo cual el servidor OpenVPN envía un ping cada 10 segundos y esperará 120 segundos máximo para recibir contestación, si el otro extremo no contesta, se considera que dicho cliente esta caído. Puede configurar estos valores de acuerdo a su necesidad.

- *persist-key*: permite que las claves no tengan que ser re-leidas cuando el Servidor OpenVPN es reiniciado.

- *persist-tun*: permite que el túnel no tenga que ser cerrado y re-abierto de nuevo tras reiniciar el Servidor OpenVPN.

- *log-append/var/log/openvpn.log*:cuando se ejecuta OpenVPN como demonio, define la ruta al archivo de log para almacenar los eventos.

- *verb 3*: especifica el nivel de detalle con el que se almacenarán los logs, entre más grande el número dado mayor será la información almacenada, por ejemplo si se escoge 0, solo se mostraran los errores fatales, si se coloca 1, se mostraran algunos aspectos, además de los errores fatales, 3 es el nivel recomendado para una buena información y 9 es el que muestra más cantidad de detalle respecto de la conexión.

- *status va/log/openvpn-status.log*: indica la ruta y el nombre del fichero donde se registra la información del estado de la conexión. En este log se registran las conexiones activas, truncadas, información del cliente como: IP pública de origen, certificado, IP virtual y la cantidad de bytes enviados y recibidos por este. Este archivo se actualiza cada minuto.

- *Mute 50*: Si se registran un número determinado de mensajes consecutivos de la misma categoría de log estos serán omitidos, el valor puede cambiar de acuerdo a la necesidad del administrador.

- *Route 192.168.1.0*:Agrega ruta local hacía la red remota.

Nota: Recuerde que para que el servidor funcione adecuadamente, se debieron copiar los archivos de certificados y llaves creados para el servidor a la carpeta ***/etc/openvpn****, para este proyecto serán los archivos* ***PST-Experts.net.key****,* ***PST-Experts.net.crt****,* ***ca.crt*** *y el archivo Diffie-helman* ***dh1024.pem***

Terminada la creación del archivo de configuración para el servidor, debemos crear el directorio *ccd*, especificado en la instrucción *client-cofig-dir*, y ubicarnos dentro de este; si se le dio otro nombre, pues cree el directorio con dicho nombre, cree esta carpeta con permisos de ejecución 700, entonces digite:

etc/openvpn# mkdir –m 700 ccd y luego

etc/openvpn# cd ccd

Dentro de este directorio se creará un archivo por cada cliente al cual se quiera permitir acceso a la VPN, un requisito fundamental que debe cumplir este fichero, es que se debe llamar exactamente igual que el nombre del certificado dado al cliente, en otras palabras debe tener el mismo *Common Name*. En nuestro caso *fabianguerra.PST-Experts.net*. A continuación se presenta el contenido de este archivo.

; Archivo de configuración de cliente: fabianguerra.PST-Experts.net

Dirección IP fija
ifconfig-push 10.8.0.9 10.8.0.10

Ruta interna para OpenVPN hacía la subred 192.168.200.0/24
iroute 192.168.25.0 255.255.255.0

No heredar la lista push global
push-reset

Exportamos la subred LAN de Matriz 192.168.99.0/24
push "route 192.168.99.0 255.255.255.0"

Nota: Asegúrese que las opciones push estén entre comillas como se muestra en el ejemplo.

En el siguiente listado se describen los parámetros usados en el archivo de configuración del cliente fabianguerra.PST-Experts.net:

- *ifconfig-push 10.8.0.9 10.8.0.10*: mediante esta opción se le asigna una dirección estática al cliente. En este caso el cliente toma la primera dirección, es decir la 10.8.0.9.

- *iroute 192.168.1.0 255.255.255*: **c**rea una ruta interna en el servidor OpenVPN hacia la red de la oficina remota, *-iroute* siempre debe de ir acompañado de un *route* correspondiente en el archivo de configuración del servidor, por ejemplo: *route 192.168.1.0 255.255.255.0.*

- *push-reset*: esta opción impide que se agreguen rutas del archivo de configuración del servidor al cliente.

- *push "route 192.168.25.0 255.255.255.0"*: añadimos una ruta en la tabla de rutas del Cliente OpenVPN. Mediante esta línea indicamos al Cliente OpenVPN que envié los paquetes que tengan como destino la red 192.168.25.0 por la interfaz del túnel, por lo que el Cliente OpenVPN podrá comunicarse con cualquier PC de esta red, además de poder comunicarse con el Servidor OpenVPN.

3.3.3 Instalación de los clientes VPN

En esta sección se verá como instalar el cliente OpenVPN; en este trabajo se instaló el cliente tanto en sistemas GNU/Linux, específicamente en Ubuntu desktop y en Windows 7.

3.3.3.1 Instalación del cliente en sistemas GNU/Linux

Básicamente la instalación del cliente en sistemas GNU/Linux como Ubuntu, Debian, MintLinux etc., es idéntica a la vista para el servidor, es decir, ejecutamos:

 # apt-get update

 # apt-get install openvpn

Al igual que lo hecho con el servidor si queremos verificar que todas las dependencias (paquete Openssl, LZO y PAM) se hayan instalado correctamente utilizamos el comando:

*# apt-cache search **nombre_del_paquete***

Al igual que en el servidor, en el cliente se instala el *servicio openvpn* que ejecuta automáticamente el inicio de la aplicación, sin embargo, en el cliente puede que no necesitemos de esta opción, por tanto, si no deseamos que el cliente OpenVPN sea iniciado al arranque del sistema usaremos el comando:

 # update-rc.d -f openvpn remove

Como la instalación crea un *servicio openvpn*, podemos, como a cualquier servicio, detenerlo o iniciarlo, para este caso se usará el script de control de ejecución /etc/init.d/openvpn.

El script requiere que exista un archivo de configuración de OpenVPN en el directorio /etc/openvpn con terminación .conf, por ejemplo /etc/openvpn/cliente.conf (la creación de este archivo será detallada más adelante).

Para iniciar el servicio de OpenVPN use el comando:

 # /etc/init.d/openvpn start

Para detener el servicio de OpenVPN use el comando:

 # /etc/init.d/openvpn stop

Nota: el script ejecuta el programa openvpn con el parametro --daemon por cada archivo de configuración en el directorio /etc/openvpn/ con terminación .conf, el proceso openvpn se ejecuta en segundo plano.

Si se realizan cambios significativos en la configuración del servidor o cliente OpenVPN (archivos *.conf*), como cambio de dirección IP, protocolo, puertos o directorios de configuraciones, se aconseja reiniciar por completo el servicio OpenVPN con el comando:

 # /etc/init.d/openvpn restart

OpenVPN crea durante la instalación el directorio /etc/openvpn en donde guardaremos los archivos de certificado y llaves creados para el cliente.

3.3.3.2 Instalación y ejecución del cliente OpenVPN en Windows 7

Para poder instalar el cliente en Windows se requiere que el usuario tenga privilegios de administrador, para ejecutar el cliente, el usuario ha de ser *administrador* o pertenecer al grupo *operadores de configuración de red*.

Otro requerimiento a tener en cuenta es que si el servidor OpenVPN envía información de red al cliente VPN, como la dirección IP del servidor DNS para usar en la VPN, deberá de asegurarse que los servicios *Cliente DNS y Cliente DHCP* estén habilitados y en ejecución.

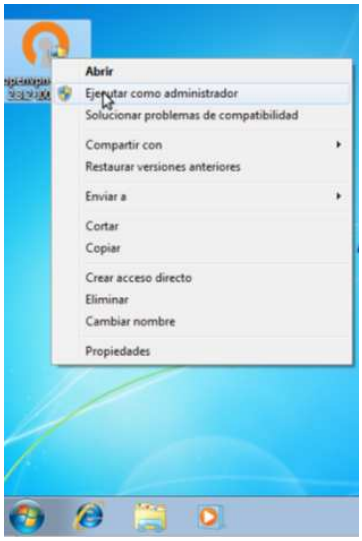
Aclarados estos requerimientos, para instalar el cliente OpenVPN en Windows 7, Vista o XP profesional, siga estos pasos:

1. Descargue la última versión del cliente para Windows desde la siguiente dirección: http://openvpn.net/index.php/download/community-downloads.html, al abrirse la

página, encontrará una tabla donde podrá elegir el driver de instalación para arquitecturas de 32 y 64 bits, a la fecha la última versión de este instalador es la 2.3.2 del 6/03/2013.

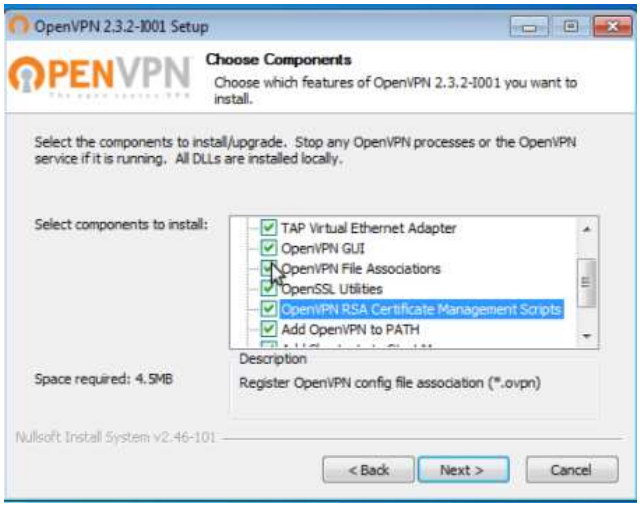
2. Vaya a la carpeta donde descargo el instalador, haga clic derecho y elija la opción *ejecutar como administrador*, como se muestra en la siguiente imagen:

Figura 26. Instalación del cliente OpenVPN en Windows.



3. Cuando aparezca la ventana de instalación, haga clic en siguiente, luego acepte la licencia y en la siguiente ventana seleccione los elementos que desea instalar; por defecto vienen seleccionados todos, y es recomendable dejarlos así, a menos que se sepa que componentes no se requieren, observe la imagen.

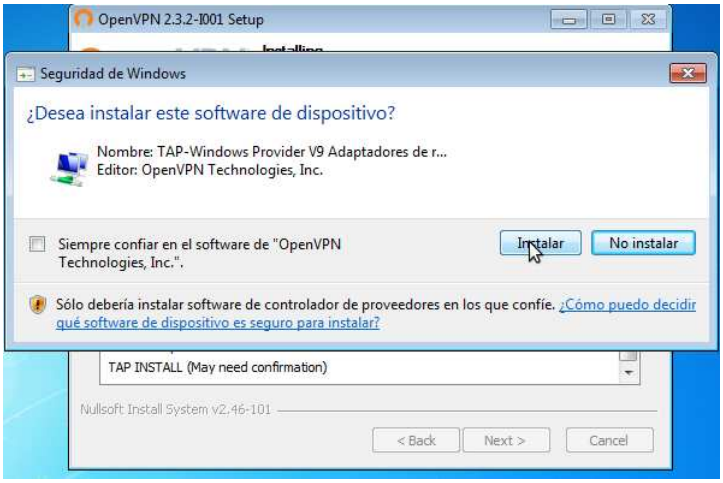
Figura 27.Instalación del cliente OpenVPN en Windows.



4. Luego seleccione la carpeta donde se instalará el cliente, por defecto se instala en *c:/Archivos de programas/OpenVPN, por*obvias razones se recomienda dejar la ubicación que trae por defecto el instalador. Enseguida se inicia el proceso de instalación del cliente.

5. Durante el proceso de instalación en Windows 7, se le mostrará un mensaje donde se le preguntará que si quiere instalar el driver TAP, haga clic en instalar. En Windows XP se le mostrará un mensaje sobre el driver TAP, se le muestra este mensaje ya que este driver no está firmado por Microsoft, haga clic en continuar para seguir con la instalación.

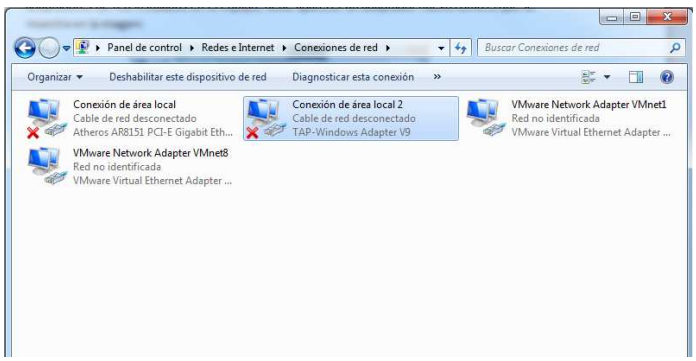
Figura 28. Instalación del cliente OpenVPN en Windows.



6. Haga clic en finalizar para terminar con la instalación.

Terminada la instalación del cliente, podemos verificar la instalación del driver TAP, para ello vaya al *centro de redes y recursos compartidos*, *cambiar configuracióndel adaptador* y vea los adaptadores de red instalados en el equipo, debe aparecer un adaptador nuevo como el que se muestra en la imagen:

Figura 29. Comprobación de la creación del driver TUN/TAP.

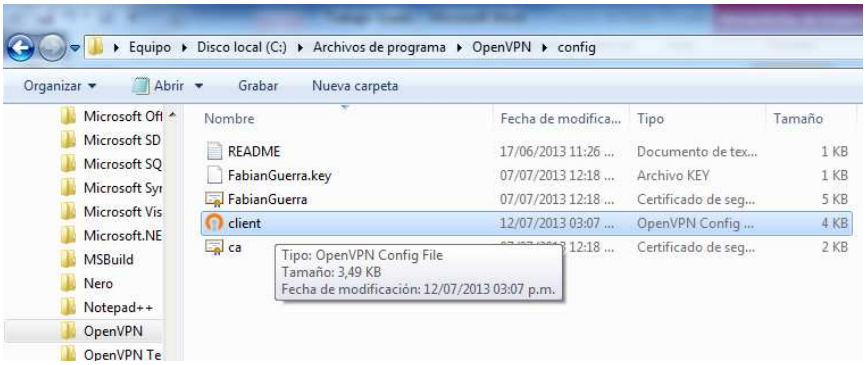


Al finalizar la instalación también se crea un acceso directo en el escritorio con el nombre *OpenVPN GUI*, el cual nos permitirá realizar la conexión entre el cliente y el servidor, pero antes de hacer esto, debemos copiar los archivos necesarios para la ejecución a la carpeta *c:/Archivos de programas/OpenVPN/config*, los archivos que se copian a esta carpeta son las llaves y certificados creados para este cliente, así como el archivo de configuración del cliente OpenVPN, siguiendo con nuestra implementación transferimos los archivos:

fabianguerra.PST-EXperts.net.crt, fabianguerra.PST-EXperts.net.key, ca.crt, archivo_configuración_cliente.conf

Con respecto a este último archivo, como Windows no procesa los archivos con extensión .conf, se debe cambiar la extensión a *.ovpn* con lo cual debe quedar *archivo_configuración_cliente.ovpn*, también se debe aclarar que cuando se copia un archivo *.conf* desde Linux a Windows, este lo reconoce como un archivo *.txt*, para que el cliente OpenVPN entienda que este es el archivo de configuración del cliente, se debe cambiar la extensión de *.txt* a *.ovpn*, si el cambio de extensión fue exitoso, el archivo cambiara a un icono similar al del instalador del cliente,observe la siguiente imagen para que pueda comprender mejor.

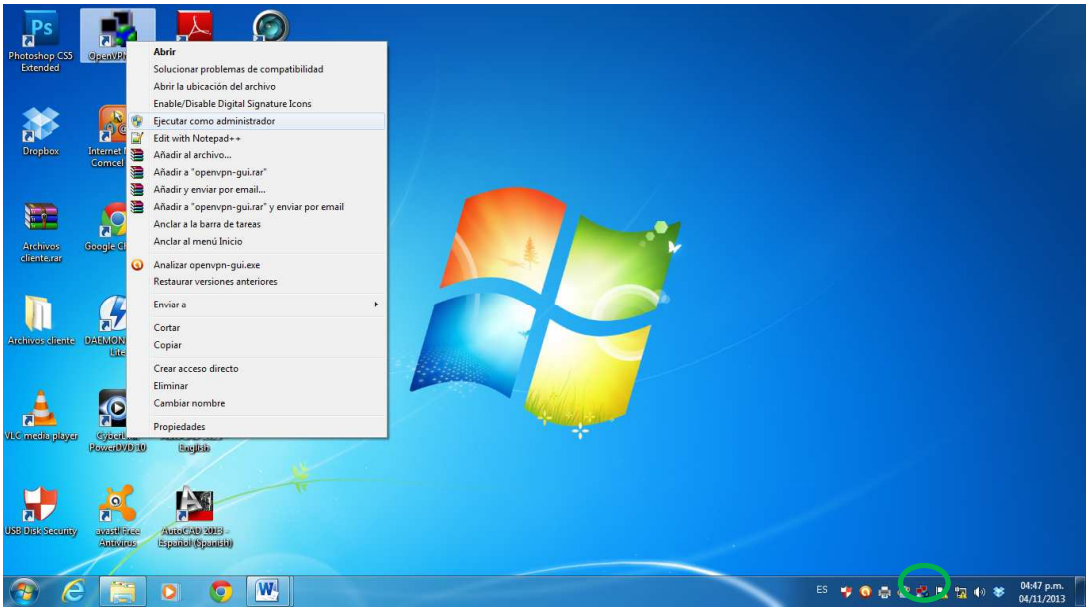
Figura 30. Carpeta de configuración del cliente OpenVPN en Windows.



*Nota: se requieren permisos de administrador para realizar cambios en la carpeta **config**, sin embargo, si lo que se desea es modificar el archivo de configuración, lo recomendable es hacerlo desde el cliente de conexión, el cual se verá más adelante, esto debido a que si realizan los cambios directamente sobre el archivo estando dentro de la carpeta**config**, OpenVPN no dejara que modifique nada de este archivo y denegará la solicitud.*

Copiados los archivos a la ubicación mencionada, ya se puede ejecutar el cliente, ubíquese en el escritorio y haga clic derecho sobre el acceso directo *OpenVPN GUI*, elija la opción *ejecutar como administrador*, aparecerá un mensaje de advertencia sobre si permitir o no la ejecución de este programa, hacer clic en *sí*, hecho esto se colocará un icono en el área de notificaciones, el cual será utilizado para realizar la conexión. En la siguiente imagen se ilustra este proceso.

Figura 31. Ejecución del cliente OpenVPN en Windows.



Nota: En sistemas Windows 7 siempre se debe ejecutar el cliente con la opción ejecutar como administrador, con el fin de evitar posibles problemas al momento de iniciar la conexión. En Windows XP solo se debe hacer doble clic.

Para predefinir que el cliente OpenVPN se ejecute siempre con privilegios de administrador, puede realizarse el siguiente procedimiento:

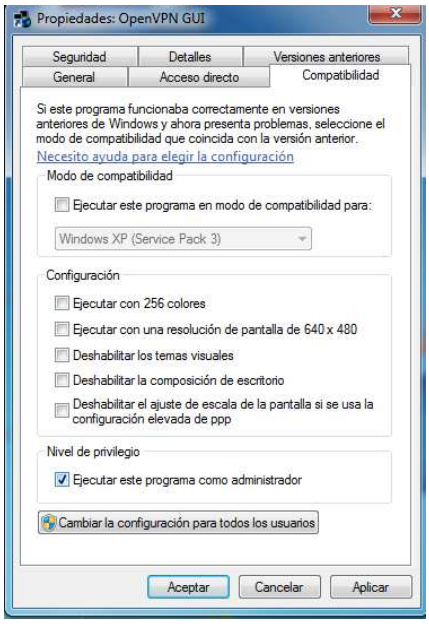
1. Hacer clic derecho sobre el icono de acceso directo de OpenVPN.

2. Hacer clic sobre la opción propiedades.

3. Hacer clic sobre la *pestaña compatibilidad*.

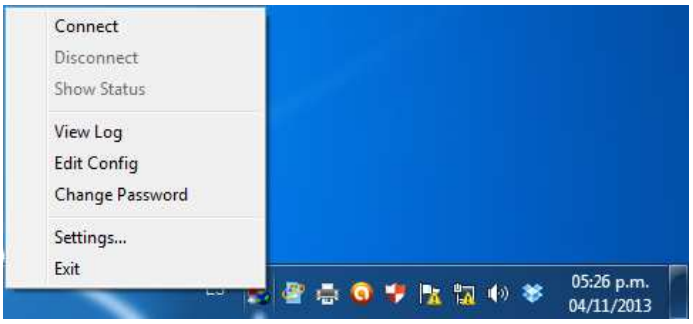
4. Ubicarse sobre la sección *nivel de privilegio* y activar la opción *ejecutar este programa como administrador* (Ver imagen).

Figura 32. Configuración del cliente OpenVPN en modo administrador.



Una vez iniciado el cliente OpenVPN, para realizar la conexión hacemos clic derecho sobre el icono en el área de notificaciones, en el menú desplegable hacemos clic sobre la opción *connect*. Observe la siguiente imagen donde se muestran las demás opciones de que dispone este cliente.

Figura 33. Opciones del cliente OpenVPN.



El icono del *cliente de conexión* en el área de notificaciones nos muestra el estado de la conexión, rojo  , indica que el cliente esta desconectado, amarillo  , indica que el cliente se está conectando y verde  , indica que el cliente se encuentra conectado a la VPN. Además de estas indicaciones, dentro del icono de conexión podemos encontrar

otras opciones como: desconectarse de la VPN (*Disconnect*), ver el estado de la conexión (*Show status*), ver el log de conexión (*View Log*), el cual resulta muy útil en caso de presentarse problemas de conexión, se puede editar el archivo de configuración del cliente OpenVPN (*Edit Config)* y cambiar algunas configuraciones básicas del cliente (*Settings*).

3.3.3.3 Creación archivo de configuración para el cliente

A continuación se presenta el archivo de configuración para el cliente en una configuración VPN sitio as sitio. Este mismo archivo puede ser utilizado tanto en clientes Linux como Windows.

Como se puede apreciar en la imagen, muchos de los parámetros utilizados ya han sido definidos en la sección "*creación del archivo de configuración para el servidor*", así que en este apartado solo se definirán aquellas variables que no hayan sido explicadas.

- *client*: Es una variable, que al igual que la variable *server*, permite simplificar la configuración del archivo al condensar en esta sola instrucción el uso de otras dos: *tls-client* y *pull*. Indica que este certificado asumirá un rol de cliente durante las transacciones TLS handshake y permite al servidor la posibilidad de recibir multiples conexiones de este tipo.
- *remote pst-experts.**no-ip.org***: especifica la dirección IP o nombre de host al cual el cliente debe conectarse para establecer el túnel, opcionalmente, después de la dirección se puede colocar el puerto por el cual debe realizar dicha conexión. Se debe aclarar que, para realizar la conexión planteada en este trabajo, se utiliza un servidor de DNS dinámicos, ya que como se ha dicho, el servidor OpenVPN se conecta a Internet a través de un proveedor que le asigna direcciones IP dinámicas, por lo cual no se puede colocar una IP fija en esta opción dado que el cliente nunca podrá conectarse a dicha dirección. En el anexo # se explicara mejor este detalle.

Figura 34. Archivo de configuración del cliente Windows.

```
; Modo cliente TLS:
client

; Opciones de conexion del tunel:
dev tun
proto udp
port 1194


; Servidor remoto:
remote pst-experts.no-ip.org 1194

nobind
resolv-retry infinite

;Autenticaciòn del servidor
ns-cert-type server

; Archivos de certificados y llaves:
ca ca.crt
cert fabianguerra.PST-Experts.net.crt
key fabianguerra.PST-Experts.net.key

;Autenticaciòn con clave HMAC
tls-auth claveHMAC.key 1

; Cifrado:
cipher BF-CBC

;Compresion:
comp-lzo

; Mantener conexiones activas via ping:
persist-tun
persist-key

; Opciones de logs:
;log-append  /var/log/openvpn.log
verb 3
;status /var/log/openvpn-status.log
mute 50
```

- *nobind*: Le indica a OpenVPN que no se vincule ni al puerto ni a la dirección IP local de esta máquina, es usada cundo se utiliza la directiva *remote*.

- *resolv-retry infinite*: le Indica al cliente que intente resolver de manera indefinida la dirección IP o el nombre de host dado en la directiva *remote*.

- *ns-cert-type server*: directiva utilizada para realizar la validación del certificado del servidor como un "certificado solo de servidor", esta directiva será aclarada más adelante en la sección que trata sobre seguridad en su apartado "previniendo ataques Man in the Middle".

- *tls-auth ClaveHMAC.key 1*: parámetro que indica al cliente que existe una clave HMAC que será solicitada por el servidor para poder realizar la conexión. Al igual que la opción anteriormente expuesta, esta variable también será detallada más adelante en el apartado "previniendo ataques Man in the Middle".

3.4 OPCIONES PARA REFORZAR LA SEGURIDAD EN NUESTRA RED PRIVADA VIRTUAL

En esta sección se mencionarán algunos métodos que OpenVPN utilizan para reforzar la seguridad de las conexiones VPN y garantizar la privacidad y confidencialidad de los datos transmitidos por el túnel, algunos de estos mecanismos ya se han implementado en los archivos de configuración del servidor y del cliente creados anteriormente para este

proyecto, otros, se mencionan a modo de información. Algunas de las técnicas que se mencionarán será la implementación de firmas HMAC para la autenticación de paquetes, como prevenir ataques Man in the Middle, revocar certificados y algunos otros métodos más.

3.4.1 Restringiendo permisos y directorios

OpenVPN permite que cuando se ejecute su demonio, se puedan transferir los privilegios de operación del túnel a usuarios con permisos limitados, garantizando de esta manera que en caso de una ruptura de la seguridad, el posible atacante no tenga los privilegios suficientes para hacer demasiado daño. Si queremos que el demonio de OpenVPN no tenga casi privilegios se ha dehacer que dicho demonio funcione con el usuario *nobody* y el grupo *nogroup*. El usuario n*obody* y el grupo *nogroup*poseen los privilegios justos para hacer funcionarOpenVPN, pero no para hacer mucho más. Para aplicar esta técnica de reducciónde privilegios,se deberán ingresar las directivas*user* y *group*, en cada uno de los archivos deconfiguración tanto del servidor como de los clientes, tal y como se muestra a continuación.

user nobody

Usuario del sistema al que se pasaran los privilegios de operación del túnel cuando el servicio OpenVPN esté ejecutándose.

group nogroup

Grupo del sistema al que se pasaran los privilegios de operación del túnel cuando el servicio OpenVPN esté ejecutándose.

Dentro de este mecanismo de seguridad, encontramos otro que se complementa con el anterior, este método consiste en utilizar la directiva *chroot*, la cual permite bloquear al demonio de OpenVPN en un único directorio, lo que seconoce como una jaula, donde el demonio no será capaz de acceder sino a la carpeta indicada por la opción *chroot*. Para realizar esto coloque la directiva *chroot* dentro del archivo de configuración del servidor y los clientes, seguido de la ruta o el nombre de la carpeta donde desee que el demonio quede enjaulado, tal como se muestra a continuación:

*chroot **restringido***

Si se introduce esta línea en el fichero de configuración hará que OpenVPN quedeencerrado en el directorio de nombre *restringido*por lo que dicho demonio nopodrá acceder a ningún fichero o carpeta que se encuentre fuera de este. Si esta variable se coloca de esta manera, hará que OpenVPN busque este directorio en *etc/*openvpn, por lo

tanto debe haber una carpeta con el nombre *restringido* dentro de este directorio. También puede, como ya se mencionó, colocar la ruta completa donde esté ubicado el directorio que quiere utilizar como jaula. Este método esimportante desde el punto de vista de la seguridad, ya que un atacante que ingrese al sistema y comprometa al servidor, no podrá acceder a ningún fichero ni directorio que no se encuentre dentro de loque hemos llamado directorio *restringido*.

3.4.2 Mejorando la autenticación TLS

Hasta ahora se ha mencionado como OpenVPN provee mecanismo de autenticación basado en la utilización de llave estática y certificados x509, sin embargo OpenVPN también permite utilizar otros mecanismos de autenticación TLS como el uso de firmas HMAC.

Como ya se mencionó en la sección donde se habla de las diferentes opciones del archivo de configuración del servidor, el uso de la directiva *tls-auth* indica que se está utilizando un archivo que contiene una firma HMAC, el cual se usará para verificar la integridad de los paquetes SSL/TLS transmitidos entre el cliente y servidor OpenVPN.

Si esta instrucción está activa en el archivo de configuración del servidor y un cliente que no posee una llave HMAC intenta establecer un enlace, el servidor rechazará la conexión inmediatamente sin ninguna otra verificación. Este mecanismo provee- una capa adicional de seguridad más allá de la que SSL/TLS brindaría por sí misma, ya que refuerza la seguridad protegiéndonos contra:

- Ataques de Denegación de Servicio (DoS) o ataques de inundación en el puerto UDP de OpenVPN.

- Escaneo de puertos para determinar que puertos UDP del servidor están en estado de escucha.

- Vulnerabilidades de Buffer Overflow en la implementación SSL/TLS.

- Negociaciones SSL/TLS originadas desde maquinas no autorizadas, (aunque tales negociaciones resultarán fallidas al final, tls-auth puede rechazarlas desde una etapa temprana).

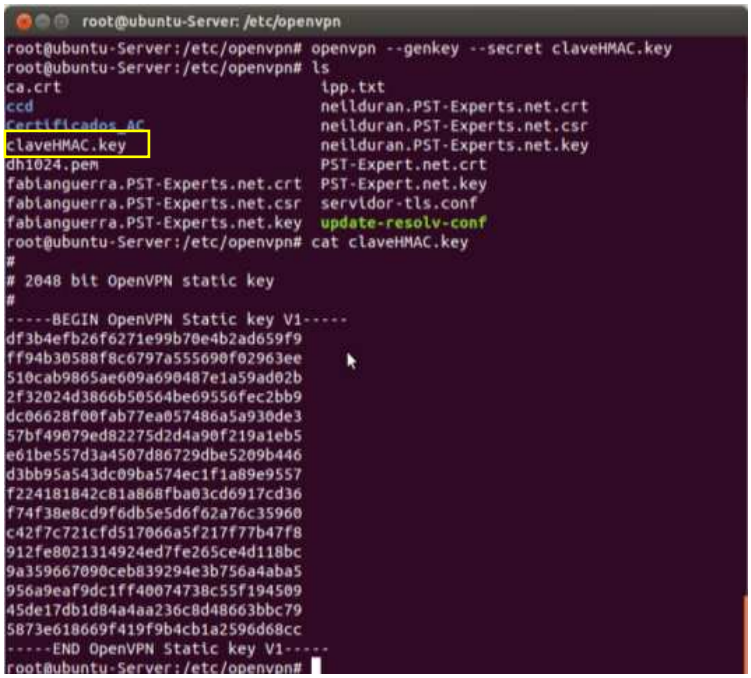
Para poder utilizar la directiva *tls-auth* en OpenVPN se debe generar una llave estática que se usará como la firma HMAC. Esta clave se genera en el servidor con el siguiente comando:

etc/openvpn# openvpn --genkey --secret ClaveHMAC.key

El comando anterior generará una llave estática de OpenVPN y la escribirá en el archivo *ClaveHMAC.key* en el directorio actual, es decir, /etc/openvpn/ta.key. Use un canal seguro para copiar esta llave a los clientes que deseen autenticarse. Este canal puede ser la misma VPN si ya está creada o mediante una conexión ssh o utilizando cualquier otro servicio que se considere protegido.

A continuación se presenta una imagen en donde se evidencia lo anterior, además de mostrar el contenido de la clave creada.

Figura 35. Creación y contenido de la firma HMAC.



Creada la clave, abrimos el archivo de configuración del servidor y lo editamos agregando el parámetro tls-auth de la siguiente manera:

tls-auth hmac.key direction

Como se puede apreciar la directiva *tls.auth* va seguida del nombre de la clave HMAC que se creó, si la clave no se creó dentro del directorio de configuración de OpenVPN, sino dentro de otra carpeta, debe indicar la ruta en donde se encuentra esta clave, pues de lo contrario se podrían presentar problemas al momento de autenticarse los clientes, por último se encuentra un argumento denominado *direction*, este argumento habilita el uso de 4 llaves (HMAC-send, cipher-encrypt, HMAC-receive, cipher-decrypt) para que cada

dirección del flujo de datos tenga su conjunto de llaves HMAC y de cifrado. La opción *direction* se asocia con un número, el cual debe ser complementario en cada lado de la conexión, es decir, si del lado del servidor usa 0, entonces del lado del o los cliente(s) use 1. La directiva en el archivo de configuración deberá quedar así:

*tls-auth **ClaveHMAC.key** 0*

Note que no se indicó ninguna ruta de la clave, por lo que se asume que su ubicación está en la carpeta de configuración de OpenVPN (*/etc/*openvpn).

Este proceso ya se habilito en el archivo de configuración del servidor y de los clientes de este proyecto, por lo que la VPN ya cuenta con un nivel de seguridad más elevado.

Si el servidor OpenVPN se encuentra en ejecución, puede optar por uno de estos métodos para aplicar los cambios: reiniciar el servidor,enviar una señal *SIGHUP* al servidor para que se reinicie y relea el archivo de configuración con los nuevos cambios o utilizar la opción reload. Estas instrucciones se especifican a continuación.

/etc/openvpn# /etc/init.d/openvpn restart

/etc/openvpn# /etc/init.d/openvpn reload

Como se mencionó anteriormente, esta clave HMAC debe ser transferida a los distintos clientes que estarán autorizados para conectarse a la VPN, en el equipo cliente guarde esta firma en un directorio que pueda ser leído por el proceso openvpn, de preferencia en el mismo lugar en que se almacenan los certificados. Para clientes GNU/Linux guárdela en /etc/openvpn y en clientes Windows en *c:/Archivos de programas/openvpn/config*.

Abra y edite el archivo de configuración de cada cliente agregando lo siguiente:

*tls-auth **ClaveHMAC.key**1*

Al igual que con el archivo del servidor, si agrego esta configuración a un cliente que este ejecutándose, entonces también deberá de mandar la señal **SIGHUP** o reiniciar por completo el proceso OpenVPN. En clientes GNU/Linux utilice alguno de los procedimientos descritos para el servidor, en clientes Windows deberá desconectarse y volverse a conectar.

3.4.3 Uso del protocolo UDP como medida de prevención de ataques

Como ya se ha mencionado a lo largo de este trabajo, OpenVPN permite el uso del protocolo UDP o TCP como medio de transporte para la VPN, sin embargo, es recomendable utilizar el protocolo UDP pues provee una mejor protección en contra de ataques DoS y escaneo de puertos que la que proporciona el protocolo TCP.

Como ya se ha visto al momento de la creación de los archivos de configuración tanto del servidor como de los clientes, para que OpenVPN utilice el protocolo UDP debe ser definido en la variable *proto*.Debe reiniciar el servidor y clientes openvpn para que el cambio tome efecto.

proto udp

3.4.4 Usando llaves RSA y de cifrado simétrico más grandes

Para usar llaves RSA más grandes que las que se han colocado para este trabajo, se debe volver a modificar el archivo *vars*, en este archivo se busca la variable *export KEY_SIZE*, el tamaño predefinido para este parámetro y así mismo para las llaves RSA es de 1024 bits, si se desea aumentar este tamaño (OpenVPN soporta claves RSA hasta de 2048 bits) defina en la variable KEY_SIZE el tamaño de la llave RSA, por ejemplo:

export KEY_SIZE=2048

Hay que recalcar que entre más grandes sea el valor de la llaves RSA, mayor el incremento en la carga del sistema cuando se realiza la negociación de la sesión SSL/TLS, la cual ocurre cada hora.Cuando se modifica la variable *export KEY_SIZE*, se afecta la creación de los parámetros Diffie-Hellman, esto quiere decir, siguiendo el ejemplo anterior, que cuando se ejecute el script *build-dh*,estos parámetros se crearan con un valor de 2048 bits, claro está, que este procedimiento solo se realiza una sola vez.

Si se cambian los valores de la variable *KEY_SIZE*, se deberá modificar el archivo de configuración del servidor en la línea donde se estipula el uso de Diffie_Hellman, en este lugar deberá colocar el valor que se le dio a esta variable, por lo tanto quedaría así:

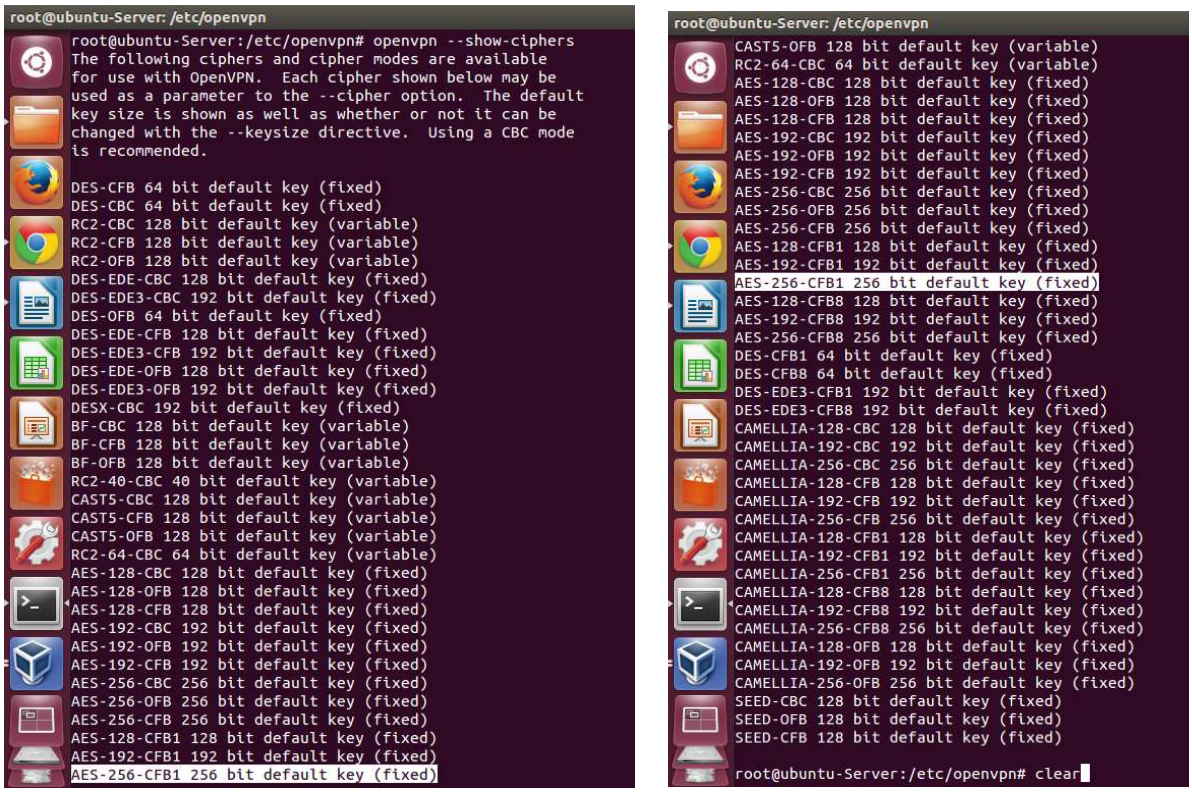
dh dh2048.pem

Si ya se ha generado todo el entorno de creación de la red privada virtual, para que estos cambios puedan efectuarse, se deberá volver a exportar el archivo *vars* y a generar los parámetros Diffie-Hellman. Se puede remitir al apartado sobre *creación de la autoridad certificadora* y al de *generación del cripto-sistema de clave pública* para realizar este procedimiento.

En cuanto al fortalecimiento del cifrado, OpenVPN utiliza por defecto el algoritmo *Blowfish* (128 bits), sin embargo, OpenVPN soporta todos los tipos de cifrado que estén soportados por la biblioteca OpenSSL, por lo que se pueden utilizar diferentes algoritmos de cifrado con llaves de diferentes tamaños.

Use el comando openvpn con el parámetro --show-ciphers para listar los cifrados soportados, a continuación se muestra una imagen con todos los tipos de cifrados soportados por OpenPN

Figura 36. Cifrados soportados por OpenVPN.



Si se desea modificar el tipo de cifrado utilizado para este proyecto, se debe modificar tanto el archivo de configuración del servidor como del o los cliente(s), para ello se ubica la variable *cipher* y se copia delante de esta el tipo de cifrado que se desee utilizar, por ejemplo
*cipher**DES-EDE3-CBC***

Nota: Se debe tener en cuenta que tanto en el cliente como en el servidor se debe utilizar el mismo tipo de cifrado

Para que este cambio tome efecto debe de reiniciar el servidor y el cliente, puede mandar una señal SIGHUP o usar los scripts de inicio ya mencionados.
Si realizo el cambio del cifrado en el servidor, pero olvido hacer el cambio en un cliente, es posible que no se pueda conectar y que en el log del cliente vera algunos mensajes, como los siguientes:

3.4.5 Revocando Certificados para clientes VPN

Cuando se crean los certificados se les asigna una fecha de inicio y una fecha de expiración, en tanto el certificado no haya expirado, los clientes lo podrán utilizar para autenticarse ante al servidor y conectarse a la VPN, sin embargo, puede ser que en algún caso se necesite invalidar el certificado de un cliente OpenVPN de tal manera que se le impida autenticarse ante el servidor y por ende conectarse a la red privada virtual. Las principales razones que se pueden dar para revocar el certificado de algún cliente pueden ser:

- Ya no se desea dar acceso a la VPN a un usuario en específico.
- La llave privada asociada con el certificado ha sido comprometida o robada.
- El usuario no recuerda la contraseña con la que se cifro la llave privada asociada a un certificado.

Para revocar el certificado de un cliente siga el siguiente procedimiento:

- Entrar al directorio de la CA.

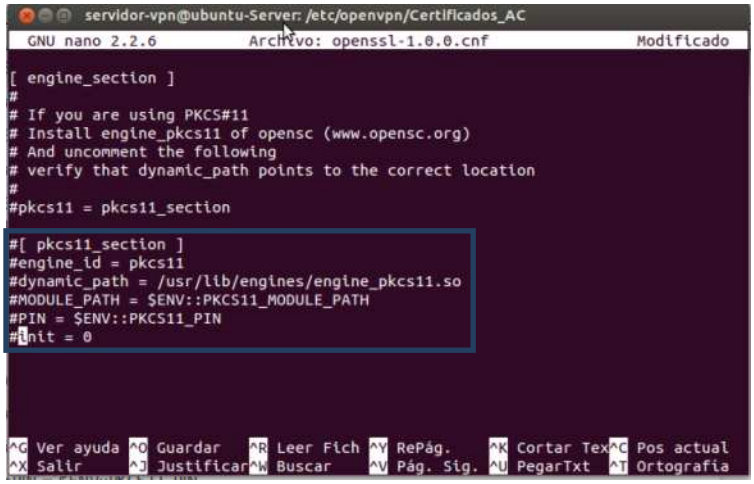
```
# cd /etc/openvpn/Certificados_AC/
```

- Exportar variables de entorno de la CA:

```
# source vars
```

- En el archivo /etc/openvpn/Certificados_AC/openssl-1.0.0.cnf se carga el módulo *pkcs11*, pero si no lo tiene cargado la revocación va a fallar, se recomienda que comente todo el bloque *[pkcs11 section]* en dicho archivo (esta sección se encuentra al final del mismo), tal como se muestra en la imagen:

Figura 37. Edición del archivo openssl-1.0.0.cnf.

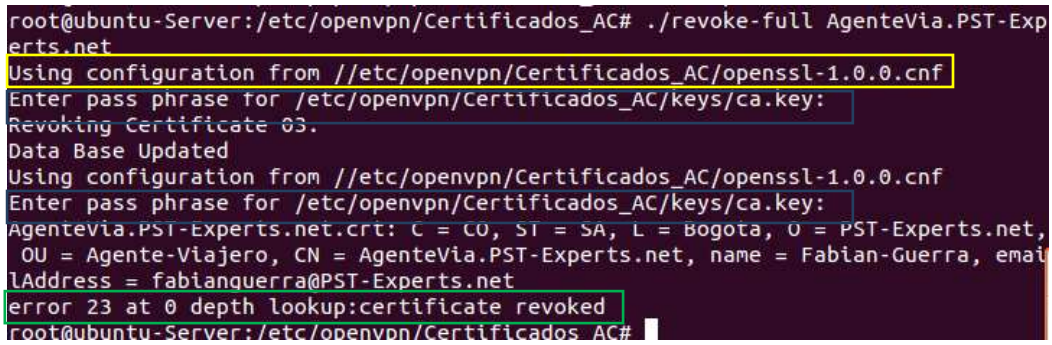


- Ahora ejecutamos el comando revoke-full con el nombre del certificado o mejor dicho con el common name como argumento, por ejemplo:

 # ./revoke-full ***nombre_del_common_name_del_certificado***

A continuación se presenta una imagen en donde se observa la ejecución de este script.

Figura 38. Revocando certificados.

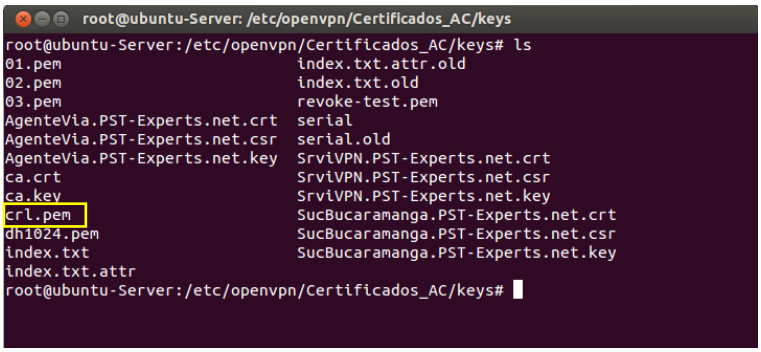


Para realizar la operación de revocación del certificado, el comando *revoke-full*solicitará el passphrase que se dio cuando se creó la llave privada de la Autoridad Certificadora *ca.key*. Después volverá a preguntar el passphrase para validar que el certificado haya sido revocado.Si no se acuerda de esta clave, se generará un error y no se podrá realizar la revocación del certificado del cliente.

Fíjese que el script indica que se va a utilizar la configuración del archivo de openssl-1.0.0.cnf. El error **23** en la última línea nos indica que la validación del certificado revocado fallo, esto quiere decir que el certificado fue revocado exitosamente.

- Si la ejecución del comando *revoke-full* fue exitosa, se crea dentro del directorio *keys* un archivo de *lista de revocación de certificados* (CRL) denominado *crl.pem* (observe la imagen).

Figura 39. Creación archivo crl.pem.



Copie este archivo a la carpeta de configuración de OpenVPN:

/Certificados_AC# cp key/crl.pem /etc/openvpn/

Nota: Si se equivoca en la digitación del pass phrase en alguna de las dos ocasiones en que se solicita, se creará el archivo crl.pem, pero la revocación del certificado no se llevará a cabo, por lo cual el cliente podrá seguir conectándose a la red privada virtual.

- Para que el servidor OpenVPN pueda saber que certificados de clientes se han revocado, se debe editar el archivo de configuración del servidor y habilitar la verificación de la CRL, para ello se digita:

crl-verify crl.pem

Se debe reiniciar el servidor OpenVPN para que el cambio tome efecto, después de esto, si un usuario listado en la CRL trata de conectarse a la VPN la conexión será rechazada.

Una vez que la opción *crl-verify* se encuentra habilitada en el servidor OpenVPN, el archivo CRL será re leído cada vez que un nuevo cliente se conecte, o una conexión existente renegocie la conexión SSL/TLS (por defecto cada hora). Esto significa que se puede actualizar el archivo CRL mientras el servidor OpenVPN está corriendo, y que la nueva lista de revocación tome efecto inmediatamente para nuevas conexiones.
Si un cliente para el cual serevocó un certificado está conectado, existen dos maneras por las cuales se puede desconectar a dicho cliente; la primera es reiniciar el servidor vía la señal (SIGUSR1 o SIGHUP) y desconectar a todos los clientes, la segunda es conectarse a la consola de administración del servidor OpenVPN y usar el comando *kill* para explicitamente matar la instancía del cliente en el servidor sin tener que interrumpir a los otros usuarios.

Para más información de la consola de administración de OpenVPN vea el apartado sobre administración del servidor OpenVPN (pág. 100).

3.4.6 Previniendo ataques de tipo Man In The Middle

Un ataque Man in the Middle (hombre al medio) consiste en la interceptación de los mensajes enviados entre dos puntos por parte de un atacante, sin que las víctimas se enteren de que el canal ha sido vulnerado. El ataque MitM es particularmente significativo en el protocolo Diffie-Hellman cuando se utiliza sin autenticación. OpenVPN cuenta con algunos métodos para evitar este tipo de ataques, principalmente reforzando la verificación del certificado del servidor en el cliente. A continuación se mencionan, en orden de preferencia, las técnicas utilizadas:

- *Creación del certificado del servidor con el atributo de servidor*:esto permite que el cliente valide que realmente se está conectadoa un servidor y no a un equipo cualquiera. para crear certificados de servidor puede utilizar el comando *pkitool* de la siguiente manera:

 # ./pkitool --interact --server nombre_certificado_servidor

Nota: este comando fue utilizado de esta manera en la creación del certificado del servidor para este proyecto, por lo que este certificado ya es de tipo servidor (remítase a la imagen 3.16 y observe el atributo seleccionado).

Para que los clientes hagan la validación de dicho atributo, se deberá agregar a cada cliente OpenVPN la siguiente línea en su archivo de configuración:

ns-cert-type ***server***

Deberá de reiniciar el cliente OpenVPN para que este cambio tome efecto, cuando un cliente trate de conectarse con el servidor OpenVPN, y si la validación del tipo de certificado del servidor es correcta, verá esto en los logs de conexión del cliente:

VERIFY OK: nsCertType=SERVER

- *Validando el nombre común (Common Name) del servidor*: se puede agregar una capa de seguridad adicional para la verificación del certificado del servidor, haciendo que un cliente acepte o rechace conexiones basándose en el *Common Name* del certificado del servidor. Para realizar esta tarea, se utiliza la directiva *tls-remote name*, donde *name* hace referencia al *Common Name* del servidor. Para habilitar esta validación edite el archivo de configuración del cliente e introduzca la siguiente línea:

tls-remote PST-Experts.net

Si no se acuerda del nombre que dio al *Common* Name del servidor, puede utilizar el siguiente comando openssl para averiguarlo.

/etc/openvpn# openssl x509 -noout -subject -in nombre_certificado_servidor.crt

- *Validando el contenido del sujeto (subject) en el certificado del servidor*: esta opción es muy similar a la anterior con la salvedad de que valida por completo el *subject* del certificado, el *subject* hace referencia a toda la información que se introdujo en la creación del certificado, como el nombre del país, del estado o departamento, el Common Name, el correo, etc. Para realizar dicha comprobación se hace mediante la directiva *tls-verify "subject"*.

Para verificar el *subject* se digita el mismo comando que para el *Common Name*. Si en el Subject del certificado del servidor se encuentran espacios en blanco, tendrá que convertirlos a "_" para que la validación funcione. Puede usar el siguiente comando para convertir el subject con espacios a "_".

/etc/openvpn# openssl x509 -noout -subject -in nombre_certificado_servidor.crt | tr ' ' '_'

Ahora en el archivo de configuración del cliente agregue la siguiente línea:

tls-verify "//C=CO/ST=SA/L=Ciudad/O=Mi_Empresa/OU=TI/CN=servidor_vpn.dominio.com/emailAddress=admin@dominio"

Deberá de reiniciar el cliente OpenVPN para que este cambio tome efecto, cuando un cliente trate de conectarse con el servidor OpenVPN, y si la validación del tipo de certificado del servidor es correcta, verá esto en los logs:

Sun Feb 22 23:36:01 2009 **VERIFY X509NAME OK**:
/C=CO/ST=SA/L=Ciudad/O=Mi_Empresa/OU=TI/CN=servidor_vpn.dominio.com/emailAddress=admin@dominio

3.4.7 Autenticación del cliente mediante usuario y contraseña

Con este método se pretende reforzar la autenticación del cliente, obligándolo a digitar un nombre de usuario y contraseña. Para poder realizar este procedimiento se necesitará que en el servidor OpenVPN se encuentre instalada la librería *libpam*, para que así se pueda cargar el módulo PAM.

Existen varios métodos para hacer que un cliente VPN se autentique utilizando un usuario y contraseña, pero en general se suele utilizar un plugin que trae OpenVPN para este propósito y que será el método explicado. Para hacer este procedimiento, en el archivo de configuración se inserta la siguiente línea:

plugin /usr/lib/openvpn/openvpn-auth-pam.soovpn

Mediante esta directiva se le está indicando al servidor OpenVPN que cargue el pluguin de autenticación indicado en dicha línea y que use el archivo de configuración *ovpn*.
El archivo de configuración *ovpn* no existe, así que se debe crear. Este archivo debe ser creado dentro de la carpeta */etc/pam.d* con el siguiente contenido:

;Archivo para la autenticación de un cliente VPN mediante usuario y contraseña.

auth required **pam_listfile.so** onerr=fail item=group sense=allow file=**/etc/security/grupos-red-virtual**

auth requisite pam_unix.so nullok_secure

auth optional pam_smbpass.so migrate missingok

Con las directivas utilizadas en este archivo, se le indica a OpenVPN que cargue el modulo *pam_listfile.so* para habilitar la autenticación basada en grupos y que busque los grupos que están autorizados para conectarse a la VPN en el archivo *grupos-red-virtual*.
Por lo anterior, se debe crear el archivo *grupos-red-virtual* en la carpeta */etc/security*, con el nombre del grupo o grupos permitidos para conectarse a la vpn.

/etc/security# nano grupos-red-virtual

vpn

Como prueba de funcionamiento de este proceso, se creó el grupo *vpn*, ahora como es lógico se deben agregar los usuarios que pertenecerán a este grupo y que estarán autorizados para conectarse a la red privada virtual, para este ejercicio se creó el usuario *Fabian*.

groupadd vpn

 # useradd -g vpn -c "Usuario VPN Fabian" -s /bin/false Fabian

Como se puede apreciar, se crea el grupo *vpn* y luego el usuario *Fabian*, el cual es asignado a este grupo (*-g vpn*), y se le restringe el uso de cualquier comando del sistema (*-s /bin/false*).

Terminado el proceso se debe reiniciar el servidor OpenVPn para que los cambios surtan efecto.

Para hacer que el cliente VPN se autentique mediante usuario y contraseña, en el archivo de configuración del cliente insertamos la siguiente línea:

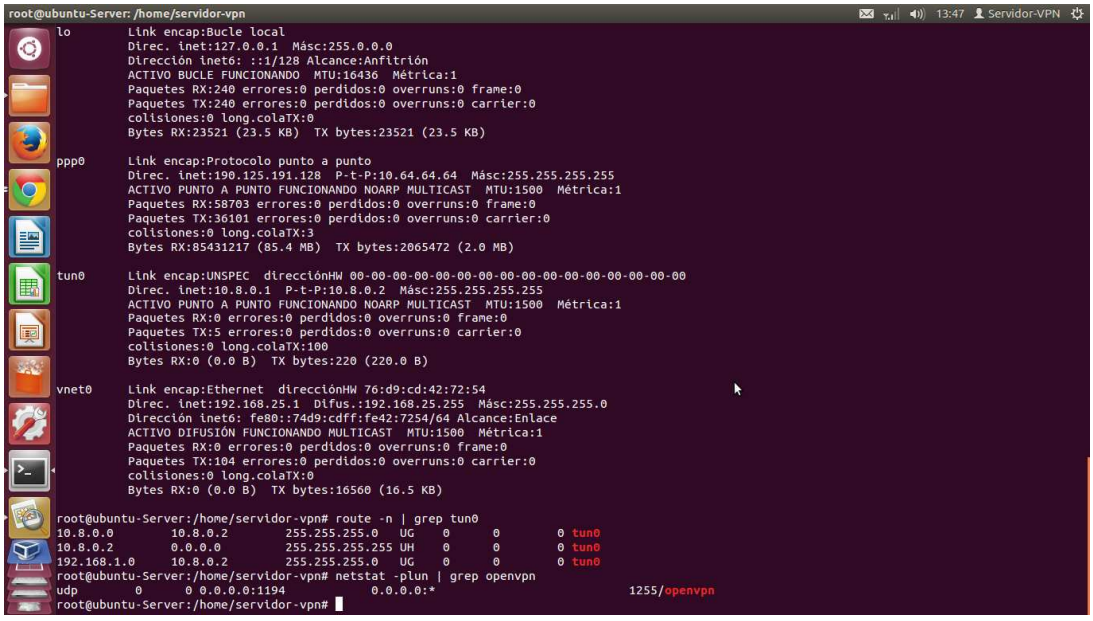
auth-user-pass

3.5 REALIZANDO PRUEBAS DE CONEXIÓN

A continuación se presentan las pruebas realizadas para verificar el funcionamiento de la red privada virtual OpenVPN.

1. Comprobación de la creación de las interfaces *tun*, de las rutas hacia la otra red y de la apertura del puerto UDP 1194.

Figura 40. Verificación de la creación de la interfaz *tun*.



Como se puede apreciar en la imagen, se utilizaron los siguientes comandos para la verificación respectiva:

- Para visualizar la interfaz *tun*.

Ifconfig o *Ifconfig tun0*

- Para verificar la creación de las rutas.

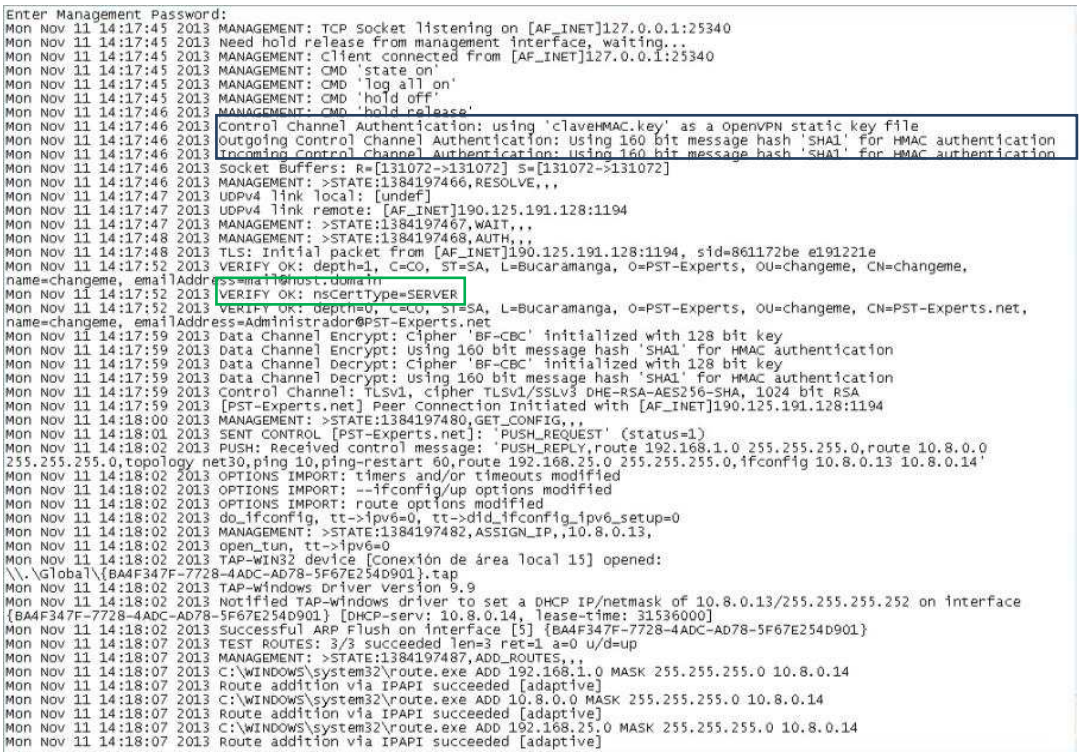
route – n | grep tun0

- Para verificar que se haya abierto el puerto UDP 1194.

netstat –plun | grep openvpn

2. Prueba de la conexión del cliente.

Figura 41. Conexión del cliente.



En esta imagen se pueden apreciar dos de las opciones de seguridad adicionales establecidas para reforzar la seguridad en la VPN. La primera es la validación de una clave HMAC y la segunda la confirmación de que el certificado sea solo de servidor.

3. Comprobación de la conexión del cliente a los equipos de la red detrás del servidor VPN.

Figura 42. Conexión desde el cliente a los equipos de la red 192.168.25.0.

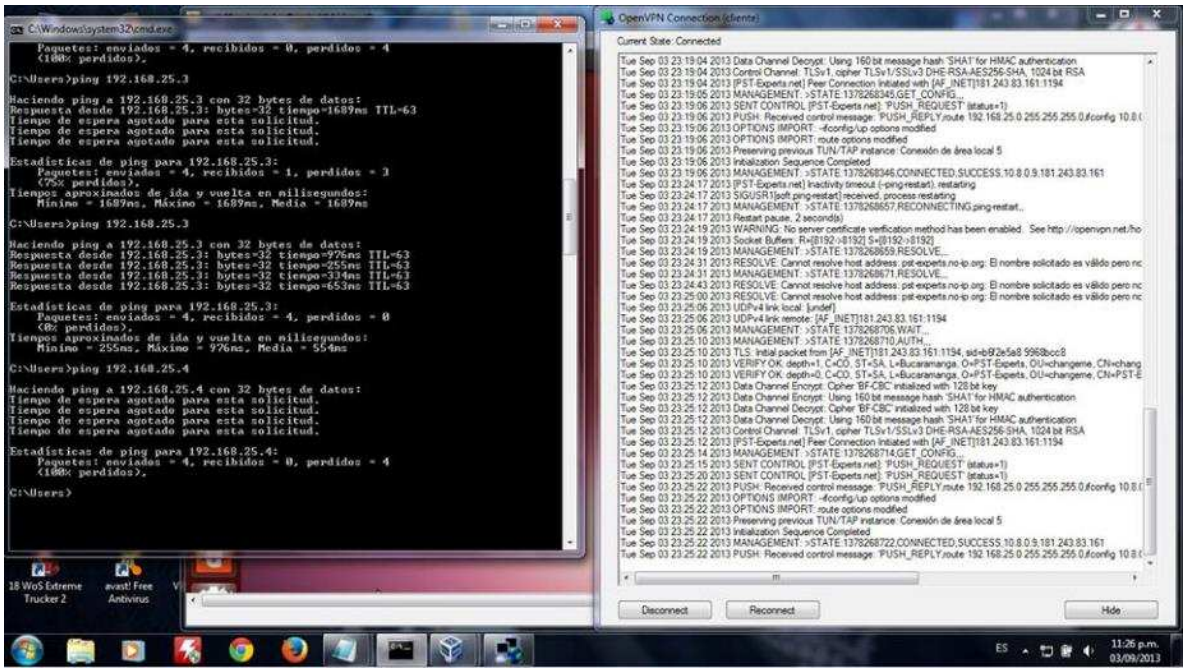
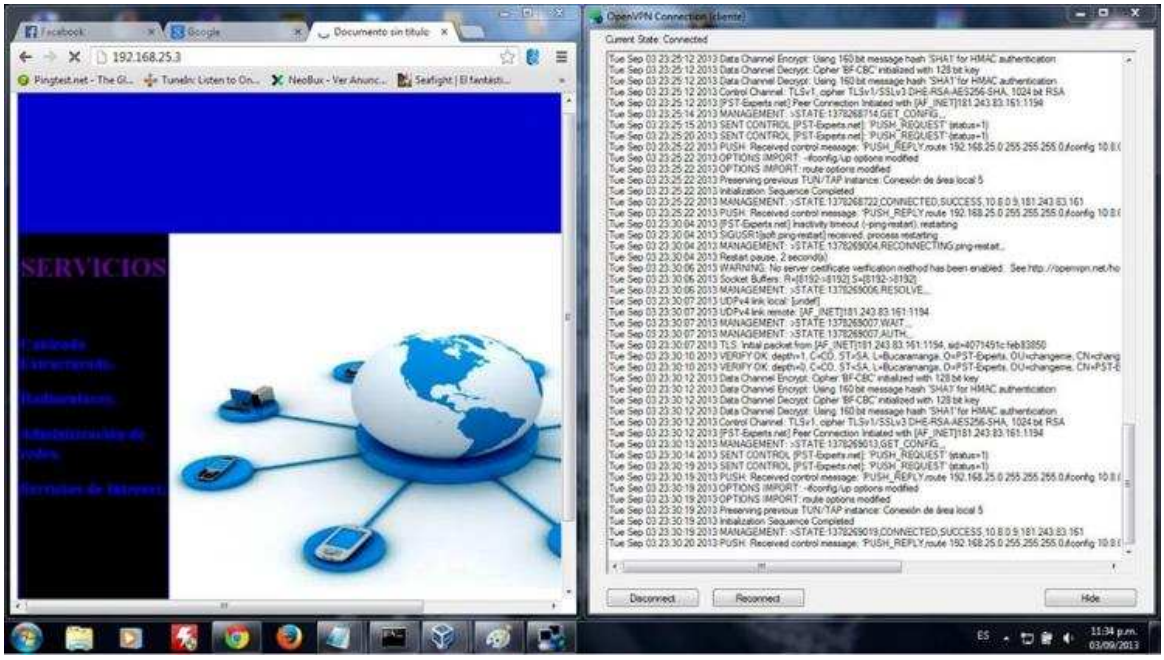


Figura 43. Conexión desde el cliente al servidor web de la red 192.168.25.0.

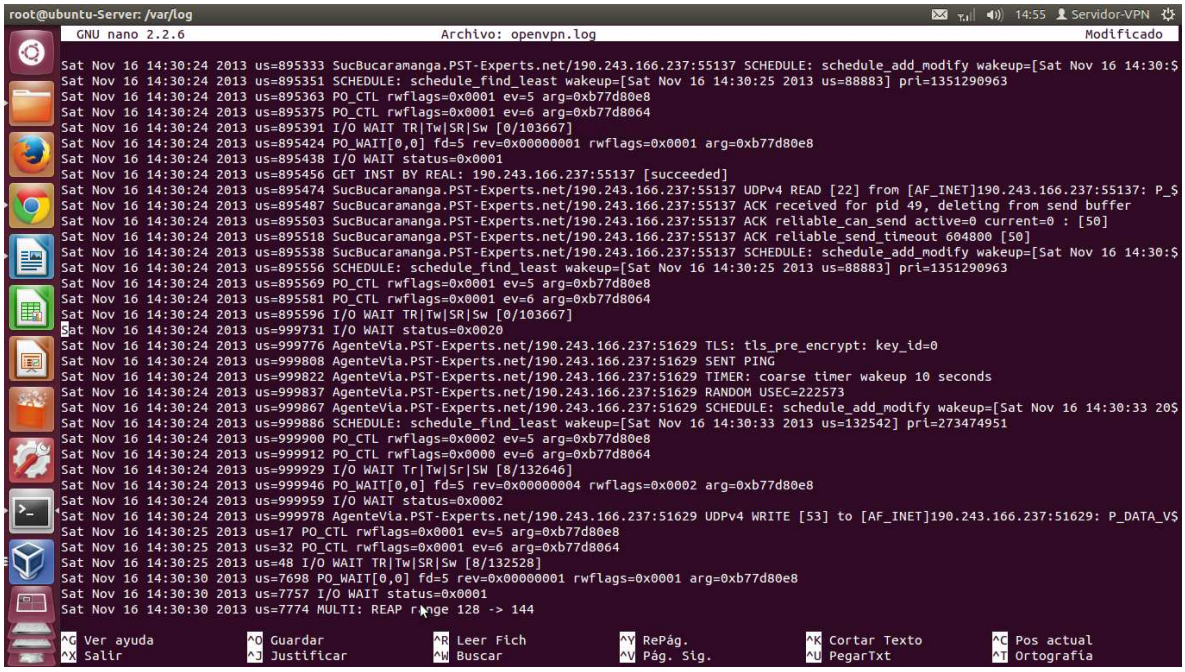


Para este punto se debe informar que dentro de esta red se creó un servidor web sencillo, ubicado en la maquina 192.168.25.3, con el fin de comprobar que un cliente conectado a la VPN pueda acceder a dicho servicio, este servidor web no está disponible para

accederlo desde Internet, pues como se ha dicho solo es para probar que realmente OpenVPN puede establecer conexiones sitio a sitio permitiéndole a un cliente conectarse a la red interna atrás del servidor OpenVPN y a los servicios que esta proporciona.

4. Verificación de la conexión de múltiples clientes.

Figura 44. Conexión multicliente a servidor VPN.

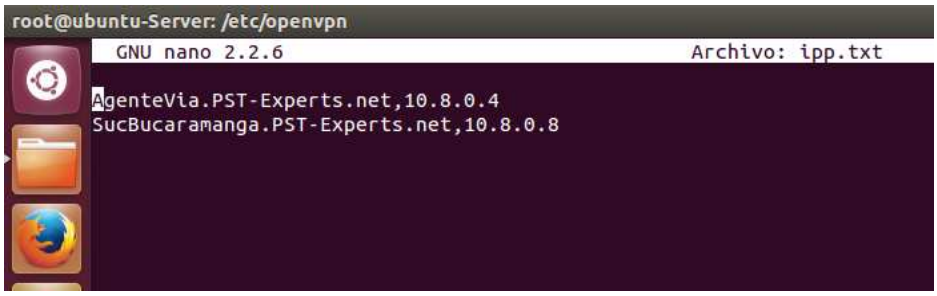


En esta imagen se aprecia como están conectados a la VPN los usuarios *SucBucaramanga.PST-Experts.net* y *AgenteVia.PST-Experts.net*

5. Confirmación de la asignación de redes a los clientes conectados

En la siguiente imagen se observa como se les asignan direcciones a los clientes a medida que se conectan a la VPN.

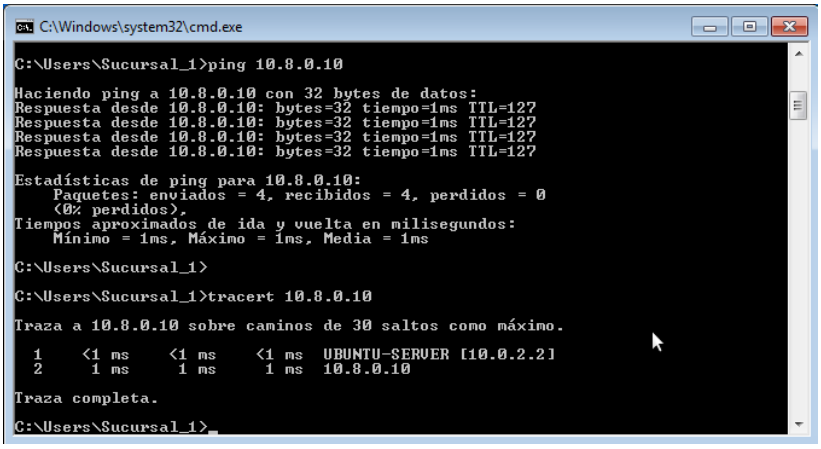
Figura 45. Asignación de subredes a los clientes conectados.



En este caso el cliente *AgenteVia.PST-Experts.net* puede tomar la dirección 10.8.0.5 o .6 de esta subred, mientras que el cliente *SucBucaramanga.PST-Experts.net* puede tomar la dirección 10.8.0.9 o .10

6. Comprobación del tracer entre los usuarios de la VPN para determinar el canal que toman los paquetes transmitidos.

Figura 46. Utilización de los comandos ping y tracertpara comprobar la conectividad entre clientes de la red VPN y determinar el camino que toman los paquetes.



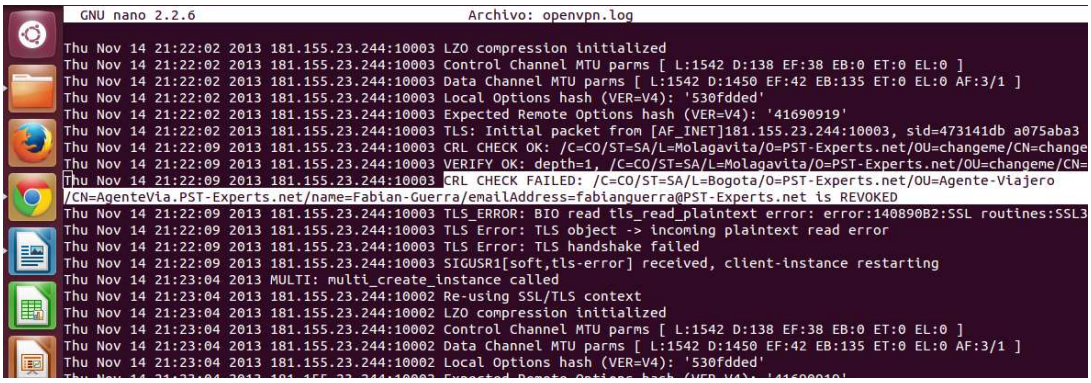
En esta imagen se puede apreciar como el cliente *AgenteVia* puede conectarse a la maquina del cliente *SucBucaramanga* y además el comando *tracert*nos muestra que para llegar desde el cliente *AgenteVia* hasta el cliente *SucBucaramanga* solo se necesitan dos saltos, uno de los cuales atraviesa por el servidor VPN.

En la siguiente imagen se utiliza traceroute en el servidor para determinar el camino hacia el cliente 10.8.0.6 (AgenteVia.PST-EXperts.net).

Figura 47. Utilización del comando traceroute en el servidor.

7. Probando la revocación de certificados.

Figura 48. Revocación de certificados.



Este log es del Servidor VPN, en él se puede observar como cuando un cliente que está en la lista de certificados revocados intenta acceder al servidor, su conexión es terminada. En el log de conexión del cliente no aparece ninguna información sobre este hecho, solo se muestra un error de conexión..

3.6 ADMINISTRACIÓN DEL SERVIDOR OPENVPN

Terminado el montaje de la red privada virtual, se hace necesario para el administrador poder monitorear la VPN para determinar posibles fallas o deficiencias en el servicio. OpenVPN provee una herramienta denominada *consola de administración*, que cuenta con los comandos necesarios para poder monitorear la VPN. Esta interfaz utiliza un modelo cliente-servidor basado en una conexión TCP, en la cual el servidor OpenVPN escucha en una dirección IP y puerto específico para las conexiones de administración.

Para poder utilizar *la consola de administración*, se debe agregar al archivo de configuración del servidor OpenVPN la siguiente directiva:

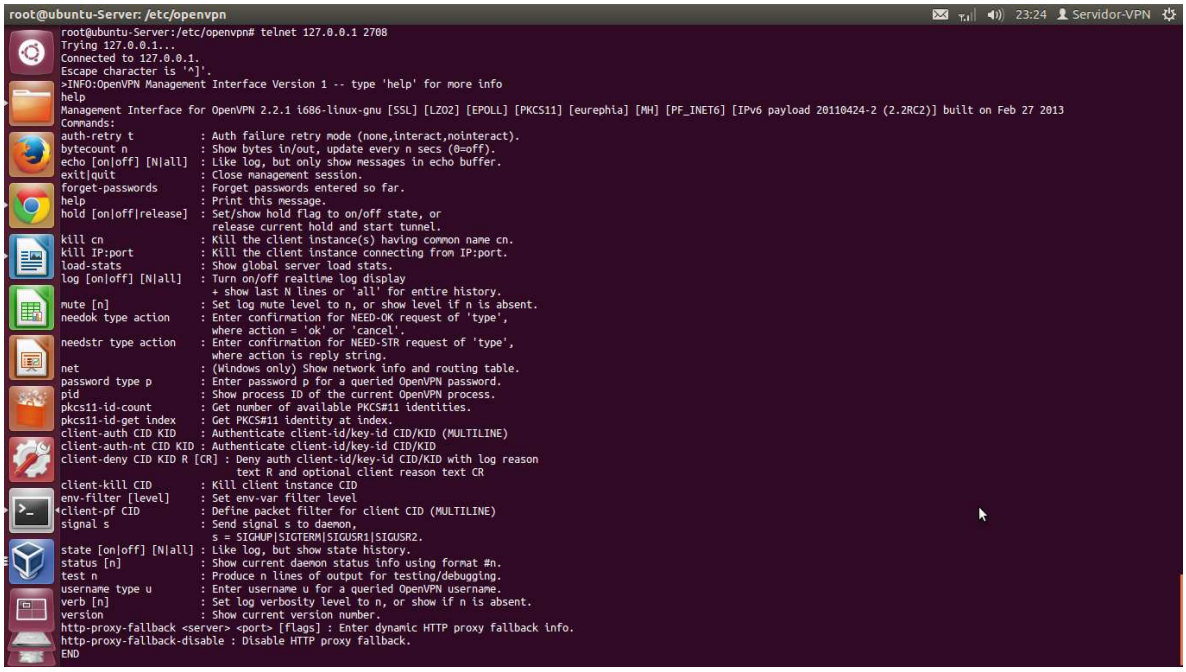
management dirección_IPPuerto

Dado que la *consola* no posee mecanismos de autenticación por contraseña, se recomienda que se active la conexión solo en la dirección lopback o localhost, el puerto puede ser designado libremente.

management 127.0.0.1 2708

Después de realizar este procedimiento, se debe reiniciar el servidor para habilitar el uso de la *consola*. Para conectarse a la *consola de administración* se puede hacer desde un cliente telnet, como se muestra en la imagen:

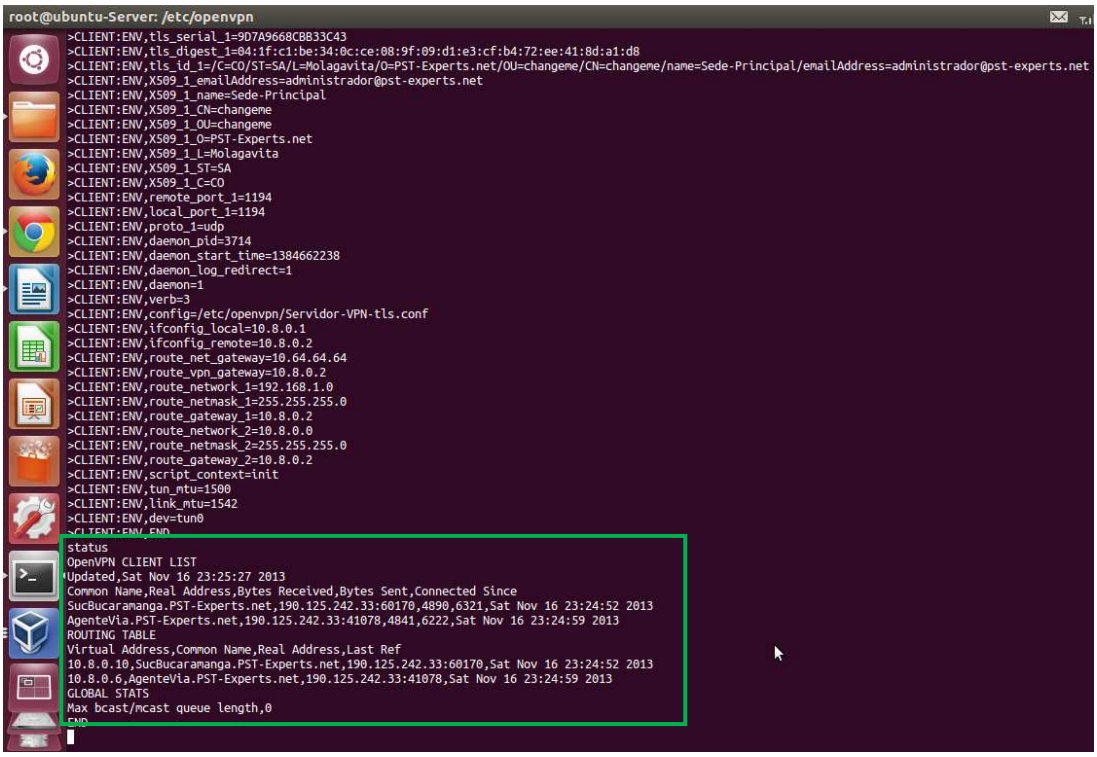
Figura 49. Conexión a la consola de administración de OpenvPN.



Cuando se conecta con *la consola de administración*, lo primero que se muestra es un mensaje donde se le dice que digite el comando *help* para obtener más información, luego de lo cual se muestra en pantalla una lista de comandos que podrán ser utilizados para monitorear y administrar el servidor y la VPN.

Por ejemplo, si se quiere saber cuantos clientes están conectados a la red privada virtual se digita el comando status, como se ve en la siguiente figura.

Figura 50. Utilizando la consola de administración.



Si se observa la imagen, antes de ejecutar el comando *status*, se puede ver que en la consola de administración también muestra el proceso de conexión de los distintos clientes que están conectados a la VPN. En la imagen se aprecia que en ese momento están conectados los clientes *SucBucaramanga.PST-Experts.net* con su dirección IP 10.8.0.10 y el cliente *AgenteVia.PST-Experts.net* con la dirección IP 10.8.0.6.

Para salir de la consola de administración se digita *quit*.

4.1 CONCLUSIONES

OpenVPN se posiciona como una aplicación a tener en cuenta para la transmisión de datos de forma segura en VPN's, ya que como se vio es fácil de instalar, configurar y administrar.

El proyecto aquí desarrollado evidencia que las pequeñas y medianas empresas pueden acceder a tecnologías de red que les pueden facilitar la implementación de servicios de internet como servidores web, de correo, FTP, entre otros, y a su vez, compartir estos servicio con otras sedes o corporaciones que se considere necesario a muy bajo costo. Lo que era el objetivo de este trabajo.

La utilización de tecnologías de virtualización en las empresas se presenta como un aliado en la disminución de los costos y centralización de la administración.

Este trabajo deja ver a grandes rasgos el potencial que la tecnología de redes privadas virtuales, junto con la de virtualización, pueden llegar a ofrecer para las PYMES.

4.2 RECOMENDACIONES

Si bien este proyecto plantea la idea de una VPN de acceso remoto a través de IP's dinámicas, es recomendable que para montar una VPN, se cuente con direcciones IP fijas, ya que eso permite un mayor control por parte del administrador de la VPN y elimina el problema de posibles fallos en el servidor de DNS dinámicos que se esté utilizando y que puedan afectar la estabilidad de la red privada virtual y los routers instalados por los proveedores en modo Bridge.

Anexo A: INSTALACIÓN DEL CLIENTE DE DNS DINAMICO NO-IP

Como se mencionó en el trabajo en el apartado sobre creación del archivo de configuración del cliente, específicamente en la variable *remote*, se introdujo como valor la dirección del servidor al cual se debería conectar dicho cliente, *pst-experts.no-ip.org*, sin embargo, también se aclaró que en este campo no se podía incluir una IP fija, ya que el servidor cuenta con una conexión a internet dinámica, por lo cual se debe buscar el mecanismo para que el cliente siempre encuentre el servidor aunque la IP de este cambie constantemente.

La solución fue crear una cuenta en un servidor que proporcionara el servicio de DNS Dinámica; existen varias alternativas al respecto, entre las cuales las más conocidas son Dyns DNS, dinahosting, No-IP, entre otros, se escogió No-IP por una sencilla razón, es gratis, bueno al menos para un dominio y tres equipos, pero es más que suficiente para poder desarrollar este proyecto.

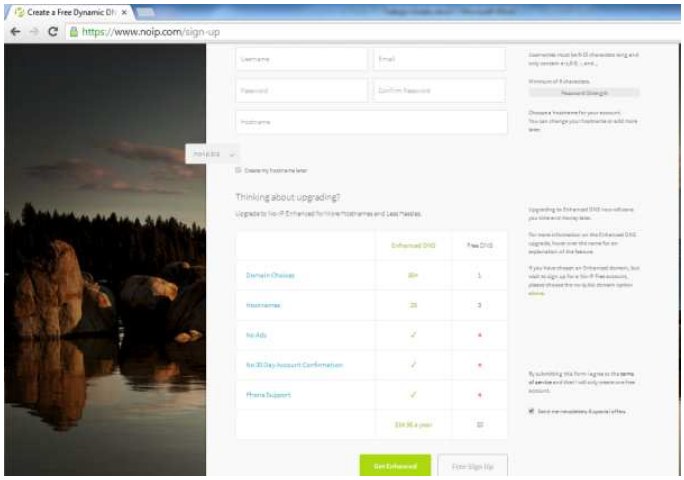
Aclarado él porque del uso de este servicio, se pasará a explicar cómo configurar este producto para que funcione en Ubuntu y así el servidor OpenVPN pueda conectar a sus clientes a través del túnel.

1. Primerase registra el equipo en la siguientepágina: www.noip.com.

2. Hacer clic en la opción *Sign UP*.

3. Rellene el formulario con la información solicitada y luego haga clic en la opción *Free sign up* (recuerde la clave introducida pues se solicitará en la instalación del cliente).

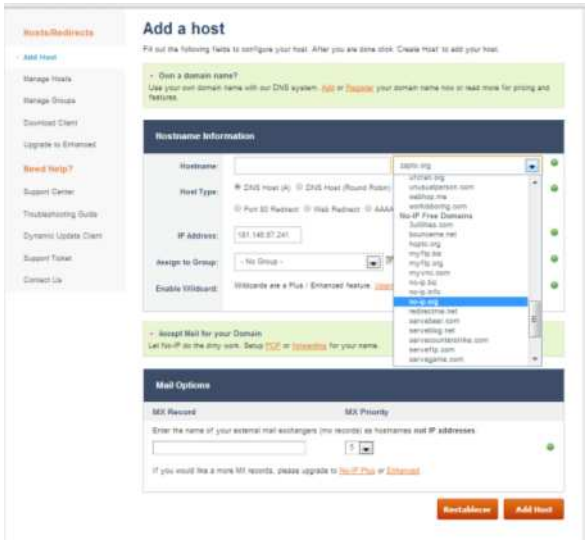
Figura 51. Registro en la web noip com.



4. Al ingresar al sitio, haga clic en la opción *add a Host*.

5. Se mostrara una página en donde aparecerá la información del equipo, como su nombre y dominio, puede cambiarlos si lo desea.

Figura 52. Registrar o cambiar el nombre del host.



6. Terminado este proceso puede utilizar la opción *Manage Host* para administrar los equipos registrados.

7. Una vez terminado el proceso de registro del equipo, se debe descargar el cliente, el cual es el encargado de actualizar la información sobre la IP del equipo. Para ello haga clic sobre la opción *Dynamic Update Client*.

Figura 53. Descargar el cliente No-IP.

8. Se abre una página en donde se mostrará el sistema operativo para el que se desea descargar el cliente, en este caso Linux, también se informa de la versión del cliente la cual es la 2.1.9, haga clic en Download para descargar el cliente. Si solicita un directorio para descargar, se puede elegir cualquiera, si no, lo más probable es que se haya descargado a la carpeta *Descargas*.

En la información suministrada en la página de descarga, dice que los usuarios de Ubuntu pueden descargar el cliente a través de la opción *apt-get*, sin embargo esto no es posible ya que el paquete ha sido eliminado de los repositorios.

Una vez descargado el cliente debemos instalarlo:

9. Ubicamos el directorio en el que se descargó el cliente y lo descomprimimos con:

tar xvzf noip-duc-linux.tar.gz

10. Se crea la carpeta noip-2.1.9-1, accedemos a ella.

11. Se digita el comando:

Sudo Make o solo *Make* si se es usuario *root*

12. Luego digite:

Sudo Make install

13. Después de que se instala se debe configurar con el comando:

Sudo /usr/local/bin/noip2-C

Al ejecutar esta opción, el cliente intentará conectarse con el servidor No-IP, si el equipo cuenta con más de una interfaz de red que pueda estar conectada a internet, solicitará que se escoja una de las interfaces, para ello dará opciones numeradas, se selecciona el número que indica la interfaz por la cual se sale a Internet.

Después de que el cliente contacta con el servidor se le solicitara el nombre de usuario y contraseña que se ingresaron al momento del registro, validada la información el cliente, preguntará por el tiempo que se desea transcurra entre cada actualización, por defecto se hace cada 30 minutos, pero si se desea puede aumentarse o disminuirse ese tiempo. Por último realizará una pregunta, a la cual respondemos no, además es la opción que el cliente ofrece por defecto.

14. Terminada la configuración iniciamos el cliente *no-ip* con la siguiente instrucción:

sudo /usr/local/bin/noip2

15. Para saber si el cliente no-ip se está ejecutando se digita:

sudo /usr/local/bin/noip2 –S

16. Para observar una lista de opciones que se pueden ejecutar en el cliente no-ip.

17. sudo /usr/local/bin/noip2 -h

BIBLIOGRAFIA

1. **Wikipedia.** Software libre. *WikipediA, la enciclopedia libre.* [En línea] Fundación Wikimedia, Inc., 05 de 18 de 2013. [Citado el: 05 de 08 de 2013.] http://es.wikipedia.org/wiki/Software_libre.

2. **GNU.** GNU.org. *Sistema operatvio GNU.* [En línea] 28 de 06 de 2013. [Citado el: 05 de 08 de 2013.] http://www.gnu.org/philosophy/free-sw.es.html.

3. **EISEN, Morty.** Introduction to Virtualization. *IEEE Long Island Section.* [En línea] 28 de 04 de 2011. [Citado el: 6 de 08 de 2013.] http://www.ieee.li/pdf/viewgraphs/introduction_to_virtualization.pdf.

4. **Wikipedia.** Virtualización. *WikipediA, La enciclopedia libre.* [En línea] Fundación Wikimedia, Inc., 13 de 07 de 2013. [Citado el: 7 de 08 de 2013.] http://es.wikipedia.org/wiki/Virtualización.

5. **VILLAR FERNÁNDEZ, Eugenio Eduardo.** Virtualización de servidores de telefonía IP en GNU/Linux. *Administración de sistemas operativos.* [En línea] 6 de 2010. [Citado el: 8 de 08 de 2013.] http://www.adminso.es/images/d/dc/PFC_eugenio.pdf.

6. **BROLLO, Gerardo G.** Aplicación e-Learning para el Aprendizaje de redes privadas virtuales. *Universidad Nacional del Nordeste.* [En línea] 2010. [Citado el: 8 de 08 de 2013.] http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/tfgerardobrollo.pdf.

7. **CANOVAS, Juan Jose T.** Servicio VPN de acceso remoto basado en SSL mediante OpenVPN. *Repositorio digítal Universidad Politécnica de Cartagena.* [En línea] 10 de 2008. [Citado el: 9 de 08 de 2013.] repositorio.bib.upct.es/dspace/bitstream/10317/758/1/pfc2873.pdf.

8. **Wikipedia.** Red privada virtual. *WikipediA, La enciclopedia libre.* [En línea] 21 de 07 de 2013. [Citado el: 10 de 08 de 2013.] http://es.wikipedia.org/wiki/Red_privada_virtual.

9. **MORALES DIBILBOX, Luis.** Investigación de Redes VPN con Tecnología MPLS. *Colección de tesis deigitales Universidad de las Americas Puebla.* [En línea] 2006. [Citado el: 14 de 08 de 2013.] http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/morales_d_l/capitulo2.pdf.

10. **ANGULO, Jenny M., HERNÁNDEZ, Jorge R. y MORENO, Deibis A.** MPLS. *Monografias.com.* [En línea] 21 de 04 de 2005. [Citado el: 14 de 08 de 2013.] http://www.monografias.com/trabajos29/informacion-mpls/informacion-mpls.shtml.

11. **WikipediA.** L2F. *WikipediA, La enciclopedia libre.* [En línea] 5 de 08 de 2013. [Citado el: 15 de 08 de 2013.] http://es.wikipedia.org/wiki/L2F.

12. --------. PPTP. *WikipediA, La enciclopedia libre.* [En línea] 07 de 08 de 2013. [Citado el: 15 de 08 de 2013.] http://es.wikipedia.org/wiki/PPTP.

13. --------. L2TP. *WikipediA, La enciclopedia libre.* [En línea] 24 de 07 de 2013. [Citado el: 16 de 08 de 2013.] http://es.wikipedia.org/wiki/L2TP.

14. --------. Secure Shell. *WikipediA, La enciclopedia libre.* [En línea] 16 de 07 de 2013. [Citado el: 18 de 08 de 2013.] http://es.wikipedia.org/wiki/Secure_Shell.

15. --------. IPsec. *WikipediA, La enciclopedia libre.* [En línea] 10 de 08 de 2013. [Citado el: 19 de 08 de 2013.] http://es.wikipedia.org/wiki/IPsec.

16. **PÉREZ IGLESIAS, Santiago.***Análisis del protocolo IPSec: el estándar de seguridad en IP.* [Revista] [ed.] Telefónica I+D. s.l., España : Fundación Dialnet, 11 de 2001. Comunicaciones de Telefónica I+D, Vol. 23. ISSN 1130-4693.

17. **WikipediA.** Transport Layer Security. *WikipediA, La enciclopedia libre.* [En línea] Fundación Wikimedia, Inc., 10 de 08 de 2013. [Citado el: 20 de 08 de 2013.] http://es.wikipedia.org/wiki/Transport_Layer_Security.

18. **TALENS-OLIAG, Sergio.** Seguridad en Java. *Universidad de Valencia.* [En línea] 12 de 1999. [Citado el: 20 de 08 de 2013.] http://www.uv.es/sto/cursos/seguridad.java/html/sjava-23.html.

19. **Universidad Interamericana para el Desarrollo.** Principales Tipos de Redes Privadas Virtuales . *Universidad Interamericana para el Desarrollo.* [En línea] 17 de 01 de 2006. [Citado el: 23 de 08 de 2013.]http://moodle.unid.edu.mx/dts_cursos_mdl/maestria_en_tecnologias_de_ informacion/tem_sel_redes/sesion9/actividades/RPV_II.pdf.

20. **WikipediA.** Debian. *WikipediA.* [En línea] 24 de 07 de 2013. [Citado el: 22 de 08 de 2013.] http://es.wikipedia.org/wiki/Debian.

21. --------. Ubuntu. *WikipediA, La enciclopedia libre.* [En línea] 24 de 07 de 2013. [Citado el: 22 de 08 de 2013.] http://es.wikipedia.org/wiki/Ubuntu.

22. **Canonical Ltd.** Ubuntu Server — for scale-out computing. [En línea] 2013. [Citado el: 23 de 08 de 2013.] http://www.ubuntu.com/server.

23. **WikipediA.** Red Hat. *WikipediA, La enciclopedia libre.* [En línea] 26 de 07 de 2013. [Citado el: 25 de 08 de 2013.] http://es.wikipedia.org/wiki/Red_Hat_Linux.

24. --------. CentOS. *WikipediA, La enciclopedia libre.* [En línea] 02 de 08 de 2013. [Citado el: 25 de 08 de 2013.] http://es.wikipedia.org/wiki/CentOS.

25. --------. VMware. *WikipediA, La enciclopedia libre.* [En línea] Fundación Wikimedia, Inc., 11 de 07 de 2013. [Citado el: 26 de 08 de 2013.] http://es.wikipedia.org/wiki/VMware.

26. --------. VirtualBox. *Wikipedia, La enciclopedia libre.* [En línea] Fundación Wikimedia, Inc., 16 de 08 de 20013. [Citado el: 26 de 08 de 2013.] http://es.wikipedia.org/wiki/VirtualBox.

27. --------. Windows Virtual PC. *WikipediA, La enciclopedia libre.* [En línea] Fundación Wikimedia, Inc., 06 de 08 de 2013. [Citado el: 26 de 08 de 2013.] http://es.wikipedia.org/wiki/Windows_Virtual_PC.

28. --------. Parallels Desktop para Mac. *WikipediA, La enciclopedia libre.* [En línea] Fundación Wikimedia, Inc., 14 de 07 de 2013. [Citado el: 27 de 08 de 2013.] http://en.wikipedia.org/wiki/Parallels_Desktop_for_Mac.

29. --------. Hamachi. *WikipediA, La enciclopedia libre.* [En línea] Fundación Wikimedia, Inc., 17 de 08 de 2013. [Citado el: 27 de 08 de 2013.] http://es.wikipedia.org/wiki/Hamachi.

30. **NeoRouter Inc.** NeoRouter. [En línea] NeoRouter Inc., 2013. [Citado el: 27 de 08 de 2013.] http://www.neorouter.com/.
31. **tinc.** tinc. [En línea] 23 de 05 de 2013. [Citado el: 29 de 08 de 2013.] http://tinc-vpn.org/.

32. **WikipediA.** OpenVPN. *WikipediA, La enciclopedia libre.* [En línea] Fundación Wikimedia, Inc., 03 de 08 de 2013. [Citado el: 30 de 08 de 2013.] http://es.wikipedia.org/wiki/OpenVPN.

33. **FEILNER, Markus**. OPENVPN En: OpenVPN: Building and Integrating Virtual Private Networks. Capitulo 3. [En línea] http://books.google.com.co [Citado el: 28 de 10 de 2013.] http://books.google.com.co/books?id=hKXEz92wtlMC&pg=PA27&hl=es&source=gbs_toc_r&cad=4#v=onepage&q&f=false

34. **SANS INSTITUTE**. OpenVPN and the SSL Revolution. [En línea] 16 de 05 de 2006. [Citado el: 28 de 10 de 2013.] http:// http://www.sans.org/reading-room/whitepapers/vpns/openvpn-ssl-vpn-revolution-1459.

35. **OPENVPN.** Change Log. *OpenVPN.* [En línea] OpenVPN Technologies, Inc., 2013. [Citado el: 2 de 09 de 2013.] http://openvpn.net/index.php/open-source/documentation/change-log/71-21-change-log.html.

36. **KRANSYANSKY, Maxim.** Vtun. [En línea] [Citado el: 07 de 11 de 2013.] http://vtun.sourceforge.net/features.html

37. **THE OPENSSL PROJECT.** OpenSSL. [En línea] [Citado el: 07 de 11 de 2013.] http://www.openssl.org/

38. **OPENSSL.** OpenSSL. *OpenSSL Cryptography and SSL/TLS toolkit.* [En línea] Ralf S. Engelschall, 2013. [Citado el: 05 de 09 de 2013.] http://www.openssl.org/.

39. **OBERHUMER, Markus F.X.J.** LZO. [En línea] 08 de 2011 [Citado el: 09 de 11 de 2013.] http://www.oberhumer.com/opensource/lzo/

40. **GONZÁLEZ SOSA, Imobach y PADRÓN MARTÍNEZ, Manolo.** Pluggable Authentication Modules (PAM). *La web de Sistemas Operativos (SOPA).* [En línea] 04 de 2003. [Citado el: 6 de 09 de 2008.] http://sopa.dis.ulpgc.es/ii-aso/portal_aso/leclinux/seguridad/pam/pam_doc.pdf.

41. **MEDINA, Jorge Armando.** Creación de Redes Privadas Virtuales en GNU/Linux con OpenVPN. *Tuxjm.* [En línea] 25 de 06 de 2011. [Citado el: 7 de 09 de 2013.] http://tuxjm.net/docs/Creacion_de_Redes_Privadas_Virtuales_en_GNU_Linux_con_OpenVPN/html-onechunk/.