UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Redes Virtuales con soporte para IPv6 usando Software Libre

TESIS

QUE PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

PRESENTA:

LUIS ENRIQUE AMAYA GONZÁLEZ

DIRECTOR DE TESIS:
ING. JUAN JOSÉ CARREON GRANADOS

Índice

Índice de cuadros, imágenes y tablas

Introducción

El mundo está cambiando continua y exponencialmente desde las últimas décadas del siglo pasado. Donde anteriormente era posible tener un red conectada físicamente en una misma región o área, y dados los vertiginosos cambios en la actualidad junto a la globalización que se vive mundialmente, ésta extiende cada vez más sus límites y los supera continuamente.

Actualmente las telecomunicaciones están rebasando todos los límites conocidos y aceptados con anterioridad, como podrían ser los límites locales, estatales o nacionales. Aunado a esto, las telecomunicaciones siguen teniendo los mismos requerimientos de antaño, que seguramente los seguirá teniendo, como podrían ser un medio comunicaciones confiables, seguras y rápidas.

Hasta no hace mucho tiempo, era común que las empresas y las comunidades académicas tuvieran sus redes conectadas físicamente, esto es; tenían enlaces dedicados en cada una de sus instalaciones, para mantener una ***WAN*** (*Wide Area Network*) .Dichos enlaces, van desde los ***ISDN*** (128 Kbps) a la fibra ***OC-3*** (*Optical Carrier*, 155 Mbps) los cuales le permiten expandir su red privada más allá de sus límites geográficos inmediatos. Pero el mantenimiento de los enlaces dedicados se vuelve costoso e incluso llega a superar el beneficio del enlace de las áreas geográficas que se busca unir.

Con el crecimiento de la red de redes, esto es la *Internet*, el uso extensivo de las redes internas o *intranets* y las crecientes necesidades de interactuar con nuestros pares académicos o empresariales en lugares cada vez más remotos surge la necesidad de usar las redes virtuales privadas o *VPN*'s, que son enlaces virtuales, esto es que no son enlaces dedicados físicos, sino que usan la estructura actualmente existente de interconexión para poder crear enlaces virtuales en dicha infraestructura a fin de poder enlazar dichos puntos geográficamente distantes sin necesidad de tener que tender un cableado de ningún tipo físicamente.

Ahora más que nunca, los entornos de desarrollo tanto empresarial como académico se dedican a ganar eficiencia al hacer que su planta laboral sea mucho menos estática, capaz de llegar a la ubicación sin demoras y así obteniendo muchos más beneficios de estas situaciones. Ante la necesidad de tener un servicio que pueda proporcionar la conexión de internet e impulsar el uso e implementación del protocolo IPv6, se propone el uso de una red virtual privada, que adicionalmente dará un rango de seguridad al usuario de este servicio y para que dentro de la universidad ésta siga estando a la vanguardia en el uso de la nuevas tecnologías.

Las redes virtuales privadas pueden ser hardware o software, además las ***VPN***'s puede dar muchos beneficios, por ejemplo puede extender el área de comunicación, mejorar la seguridad, reducir los costos operativos con respecto al mantenimiento de las ***WAN*** tradicionales, reduce el tiempo de tránsito y el costo de transporte para los usuarios remotos, simplifica la topología de la red, provee oportunidades de red globales, facilita la compatibilidad con los protocolos de red usuales y retorno de inversión mucho más rápido que en las redes tradicionales.

Para un correcto desempeño de nuestra red deberemos de tomar en cuenta algunos de estos tópicos como son la seguridad, confiabilidad, escalabilidad, la administración de las redes y las políticas de las mismas.

Dado que las VPN's soportan en su mayoría el protocolo TCP/IP como medio de comunicación a través de las redes y que dicho protocolo está presentando una transición en estos momentos de la versión IPv4 a la IPv6. Esto es, el motivo de la transición se debe, entre otras razones, al agotamiento de direcciones disponibles en la versión IPv4 del protocolo de internet que teóricamente debería de tener unas 4,294,967,296 (2^{32}) direcciones, aunque existe una cantidad de dichas direcciones reservada para propósitos especiales como a las redes privadas (aprox. 18 millones) o para multicast (aprox. 16 millones) lo cual reduce la cantidad de direcciones que puedes ser colocadas públicamente. Además de la incorporación de nuevos medio y dispositivos que también están haciendo uso de las direcciones de internet lo que esta acelerando el agotamiento de dichas direcciones en la versión actual del protocolo. Por lo cual se está impulsando el uso de la nueva versión del protocolo, esto es IPv6 el cual tiene 128 bits, por lo que teóricamente debe de ofrecer unas 2^{128} direcciones (unos 340 sextillones) de direcciones.

Es necesario implementar dicho protocolo por las nuevas necesidades que se vienen gestando desde hace un tiempo, audio y video entre muchas otras, así como la gran cantidad de servicios que se utilizan actualmente. Existen varias implementaciones para las redes virtuales privadas (a) ***VPN***, en las cuales se contempla el diseño, la seguridad, el tipo de conexión, etc. Adicionalmente se pueden usar tecnologías auxiliares en el área de la seguridad como IPSec o los servidores AAA que nos ayudan a garantizar la seguridad de nuestras conexiones y la integridad de nuestros datos.

Actualmente por el amplio uso del NAT (*Network Address Translation*) se ha podido extender el plazo que se le había predicho a IPv4, pero esto no podrá ser sostenido mucho más tiempo, ya que las aplicaciones que no se adhieren estrictamente al modelo cliente-servidor (*tales como VoIP, video conferencia y aplicaciones peer to peer*) y para las cuales ya se está dando un uso masivo, exigen tener un tipo de conexión punto-a-punto, esto es que se respete aquel principio bajo el cual se construyó la red, de la transparencia de conexión.

En cuanto a la transición de IPv4 hacia IPv6, la cual todavía estamos atravesando, ésta no tiene una fecha definida como fue el caso de IPv4, y tiene más que ver con el desarrollo de políticas tanto en la iniciativa privada como en la actividad pública, donde se tienen que definir los caminos y las iniciativas a tomar para lograr que la transición sea los menos lenta posible. En cuanto al usuario final, la llamada *última milla*, éste tendrá que cambiar los dispositivos que le provean de conexión a la red o a través de su proveedor de servicio.

En la actualidad el software libre y su modelo de desarrollo han alcanzado un alto grado de madurez, por lo cual, las más grandes empresas a nivel mundial y prestigiadas entidades académicas, lo adoptan como parte de sus herramientas fundamentales para la generación de aplicaciones o software a la medida. Incluso se ha especulado sobre la desaparición del software cerrado como modelo viable para la industria. Aunque esto se llevará un periodo de transición, no dudamos de que éste es el camino. Es gracias al software libre y abierto que se puede generar la innovación en casi cualquier lado, desde las grandes empresas e industrias hasta los desarrolladores individuales.

Prueba de esto, son los parches que los desarrolladores independientes han aportado al proyecto OpenVPN, desarrolladores que no pertenecen a la rama estable y principal de OpenVPN, estos desarrolladores, por su cuenta han enriquecido al proyecto a fin de que ofrezca cualidades que originalmente no posee, pero que gracias a que es de código abierto y libre, es posible que sea modificado de acuerdo a las necesidades de quienes requieran capacidades diferentes de las que este software ofrece de manera estándar.

Así, el objetivo del presente trabajo de tesis es desarrollar una solución robusta y de sólida cobertura que permita poner al alcance de la comunidad universitaria esta nueva versión del IP. Para ello se escogió el uso del software libre para este desarrollo. Así en el capítulo 1 se hace una revisión del protocolo de Internet en su versión 6, se muestra su estructura interna en general y se hace mención somera de sus componentes. En el capítulo 2 se hace mención del complemento mandatorio del IPv6, este es IPSec, un conjunto de protocolos destinados a ofrecer seguridad intrínseca a las comunicaciones a través del IPv6. En el capítulo 3 se describen de manera general a las redes virtuales privadas, algunas de sus clasificaciones, así como las tecnologías de las que hacen uso.

En el capítulo 4 el objetivo a desarrollar con el software OpenVPN es la implementación de un servicio totalmente funcional de una red privada virtual con soporte para IPv6. La meta es que el servicio este a disposición de la comunidad universitaria a fin de que ayude a promover la adopción del protocolo de Internet de nueva generación, así como la generación de buenas prácticas de uso y la capacitación de todos aquellos que son parte importante al difundir esta tecnología; administradores de red, desarrolladores y usuarios en general.

Capítulo 1

Protocolo de Internet Versión 6

La nueva versión del protocolo de internet tiene características novedosas y actualizaciones con respecto a su versión predecesora que es la que seguimos usando hoy en día. Por lo cual es importante conocer dichas características internas y saber a qué retos se enfrenta este nuevo diseño, por lo cual en este capítulo se hará una exposición del mismo.

Capítulo 1 Protocolo de Internet versión 6 (IPv6)

Esta versión del protocolo de Internet está diseñada para cubrir las actuales limitaciones del protocolo en su versión 4 (*IPv4*) además de proveer mejoras como un mayor espacio para direcciones al pasar de 32 bits en IPv4 a los 128 de IPv6, simplificaciones en los formatos de encabezados, soporte mejorado para el campo de opciones, capacidades nativas de calidad de servicio, servicios de autenticación y cifrados incluidos, auto configuración de direcciones, mejor soporte a la movilidad para dispositivos y usuarios, tráfico multimedia en tiempo real, etcétera.

Además se han desarrollado mecanismos de transición que nos garantizan la coexistencia de ambas versiones del protocolo durante la transición a la última versión.

Actualmente la necesidad de intercambiar información, el crecimiento de la Internet como la red de redes más usada, así como los requerimientos de una mejor seguridad hacen que el uso del IP en su cuarta versión sea cada vez más difícil de manejar. Añadido a esto, la red de redes está inmersa en un medio de una rápida evolución, con tendencia a las modificaciones inmediatas e inseguras, con demandas cada vez más amplias de servicios que garanticen la seguridad y fiabilidad de uso. En nuestros días, donde se garantiza que el agotamiento del espacio disponible de direcciones IPv4 está más cercano, el uso de alternativas, como el *NAT (network addres translation)* y *CIDR (classless inter-domain routing),* además de consideraciones de orden político o de índole económicas no totalmente claras por parte de los actores involucrados en el área de las telecomunicaciones, tanto de la iniciativa privada como de la administración pública han provocado el aplazamiento de dicha transición.

Para propósitos históricos mencionaremos que los principales arquitectos de IPv6 fueron Steve Deering y Robert Hinden.

Un rol importante en el campo del desarrollo e implementación del IPv6 lo han jugado las redes académicas, las cuales han estado generalmente interesadas en el desarrollo y no tanto en las ganancias económicas. Esto ha traído experiencia y recursos humanos capacitados para el desarrollo futuro.

En los últimos años se ha visto un incremento en el número de dispositivos que incluyen capacidades de soporte para el IPv6. Otras regiones del mundo claramente están apoyando el uso de la próxima versión del protocolo, entre ellas Asia y Europa, en la primera región, su tardía inclusión en la red de redes provocó que le fuera asignada una menor cantidad de direcciones en IPv4, lo cual sumado al rápido crecimiento y evolución tecnológica de dicha parte del mundo provoca un agotamiento acelerado de sus direcciones disponibles, y en la segunda región, la Europea, la dorsal (*backbone*) académica del proyecto GÉANT provee soporte oficial para IPv6 desde enero de 2004.

En el mundo existen varias organizaciones que se han encargado del desarrollo del protocolo como son el IPv6 Forum, que es un consorcio formado por proveedores de soluciones, proveedores de servicio de Internet (*ISPs*) además de redes de académicas y de investigación, el cual tiene como misión promover a nivel mundial el uso del IPv6.

En la UNAM existe un grupo de trabajo encargado de promover IPv6, coordinando a su vez a los grupos correspondientes en la Corporación de Universitaria para el desarrollo de Internet (*CUDI*) cuya misión es la de promover y coordinar el desarrollo de redes de telecomunicaciones y cómputo enfocadas al desarrollo científico y educativo en México, así como en la Cooperación Latinoamericana de Redes Avanzadas (*CLARA*) con objetivos similares en toda la región.

En estos momentos IPv6 está siendo considerado clave en el desarrollo de tecnologías como las comunicaciones ubicuas, los servicios multimedia VoIP, las redes punto a punto, etcétera. IPv6 tendrá unos 15 años muy pronto, en los cuales se ha estado robusteciendo y expandiendo sus campos de aplicación hasta abarcar varios como son los ambientes de colaboración a distancia, tecnologías de ubicuidad como los servicios de paquete vía radio[transferencia] (*GPRS*, por sus siglas en inglés), televisión de alta fidelidad (*HDTV*), control remoto de distintos dispositivos, aplicaciones sobre protocolos inalámbricos, enlaces mediante cable eléctrico (*PLC*, por sus siglas en inglés), conexiones domésticas por vía telefónica o fibra, aplicaciones sobre demanda en línea (juegos, colaborativas, etc.), tecnologías always-on (tales como *xDSL*, fibra y más), todo lo anterior son solo algunos de los campos sobre los que se han desarrollado aplicaciones para el protocolo de Internet en sus 2 versiones.

En nuestros días, la primer década del siglo XXI, las redes experimentales han dejado de serlo para volverse redes de producción, madurando la nueva actualización de la pila del protocolo TCP/IP de tal manera que ya se pueden tener una variedad de servicios para astronomía, bibliotecas digitales, educación a distancia, ciencias de la tierra y de la vida, campos colaborativos, laboratorios remotos, robótica, súper cómputo, telemedicina, visualización, cómputo científico y muchos más.

1.1 Datagrama en IPv6

El método por el cual IPv6 encapsula el tráfico recibido a través de los protocolos de las capas inferiores es, básicamente, el mismo que se utiliza para IPv4. Mientras que el uso de datagramas no ha cambiado, para IPv6 se han realizado algunas modificaciones a la estructura y al formato. El incremento de tamaño en las direcciones IP de 32 a 128 bits, llevó a un incremento de información en el encabezado. Todo lo cual condujo a promover el retiro de los campos que en el protocolo eran redundantes o que no eran estrictamente necesarios. Además se introdujeron cambios y nuevas características para mejorar la interacción del protocolo con la interconexión actual. Los cambios más significativos en el encabezado IPv6 son los siguientes:

•Estructura de encabezados múltiples

En vez de contener un solo encabezado que contenga todos los campos para un datagrama (que posiblemente incluyera opciones), el datagrama IPv6 soporta un encabezado principal y encabezados de extensiones para la información adicional, cuando se requiera.

•Formato simplificado de encabezamiento

Algunos campos han sido removidos del encabezado principal para reducir su tamaño y mejorar su eficiencia. Sólo los campos que son requeridos por la mayoría de los datagramas permanecen en el encabezado principal; otros has sido trasladados a los encabezados de extensión y serán utilizados bajo demanda; y otros fueron removidos porque ya no eran necesarios.

•Campos Renombrados

Algunos campos han sido renombrados para reflejar mejor su uso actual en las redes modernas.

•Mayor Flexibilidad

Los encabezados de extensión permiten una mayor cantidad de información adicional cuando sea requerida.

•Eliminación del cálculo de la suma de verificación

Para esta actualización del protocolo ya no se calculará la suma de verificación en el encabezado. Esto ahorra tanto el tiempo de cálculo gastado por cada dispositivo que empaquetaba datagramas IP (hosts y ruteadores) y el espacio que el campo de la suma de verificación ocupaba en el encabezado IPv4.

•Calidad de Servicio mejorada

Un nuevo campo, el campo de flujo, se ha agregado para ayudar en la jerarquización de tráfico.

El encabezado principal en IPv6 contiene información de control y localización que será usada en el procesamiento y ruteo de un datagrama, y su longitud final es de 40 bytes. Los campos que conforman el encabezado principal son los siguientes:

•*Versión (4 bits):* indica la versión IP usada para generar el datagrama.

•*Clase de tráfico(8 bits):* este campo reemplaza al de tipo de servicios (*TOS*) del encabezado IPv4. No se usa de la manera en que se usaba TOS (con precedencia, bits D, T y R), sino usando ahora el método de los servicios diferenciados (*DF)*, definidos en el RFC 2474, para distinguir e identificar entre diferentes clases o prioridades en los paquetes.

•*Etiqueta de flujo (20 bits):* fue creado para proveer soporte adicional para los datagramas que utilizan las características de tiempo real y calidad de servicio.

•*Longitud de la carga útil(16 bits):* mide la longitud de los encabezados de extensión, ya no se toma en cuenta el encabezado principal como en IPv4.

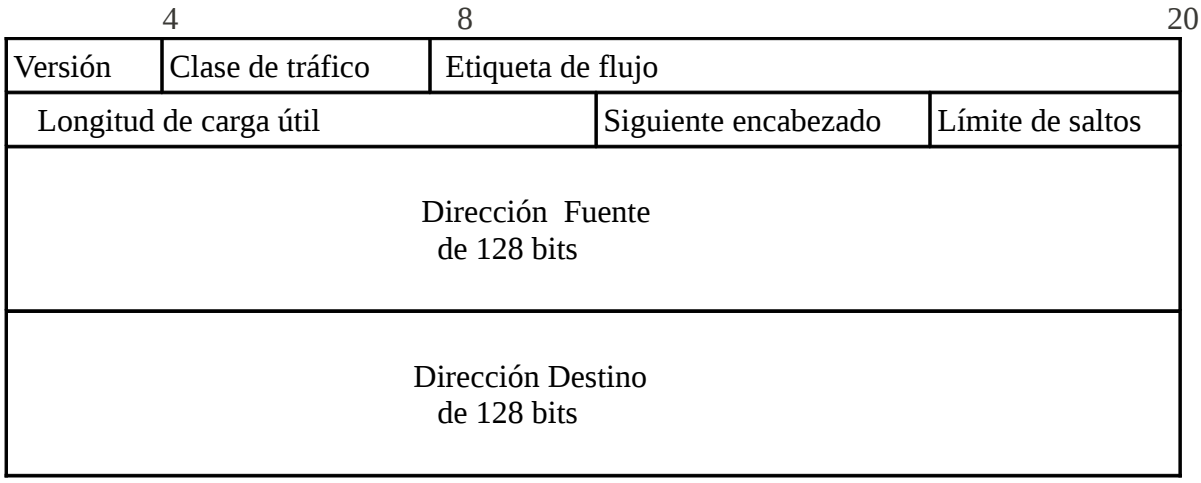
•*Siguiente encabezado(8 bits):* sirve para identificar al siguiente encabezado que sigue al encabezado principal.

•*Límite de saltos(8 bits):*reemplaza al campo *TTL* del encabezado IPv4, con su nueva denominación refleja mejor el objetivo de su trabajo, en tanto que cuenta saltos, no tiempo.

•*Dirección origen(128 bits):* la dirección IPv6 de 128 bits del origen del datagrama.

•*Dirección destino(128 bits)*:la dirección IPv6 de 128 bits del destinatario del datagrama, unicast, multicast o anycast.

Todo lo cual se puede observar en el cuadro 1, la imagen del datagrama que se tiene para IPv6.



Cuadro 1. “Datagrama IPv6”

1.2 Encabezados de IPv6

En esta actualización de la pila del IP, se tiene sólo un encabezado principal y toda la información adicional que requiera un paquete estará contenida en los llamados encabezados de extensión, los cuales fueron creados en un intento de proveer de eficiencia y flexibilidad a los datagramas IPv6. De acuerdo con el RFC 2460 los encabezados de extensión (EH) deberán ir en el siguiente orden y con el código asignado en el encabezado de *siguiente encabezado*. En la Tabla 1, mostramos los encabezados que se están manejando actualmente para IPv6, así como la existencia de

encabezados experimentales.

	Tipo de encabezado	Código asignado.
1	Encabezado principal IPv6.	-
2	Encabezado de opciones salto a salto.	0
3	Encabezado de opciones de destino (con opciones de enrutamiento).	60
4	Encabezado de enrutamiento.	43
5	Encabezado de fragmentación.	44
6	Encabezado de autenticación.	51
7	Encabezado de carga de seguridad de encapsulamiento.	50
8	Encabezado de opciones de destino.	60
	Encabezado de movilidad (*RFC5096 experimental*).	135
	Encabezado de sin siguiente encabezado .	59
9	Encabezado de protocolo de capa superior (TCP, UDP, ICMPv6).	(6, 17, 58)

Tabla 1. “Encabezados en IPv6”

Encabezado de extensión salto a salto

Este encabezado se usa para dar soporte a los jumbo-datagramas o, con la opción de alerta de enrutamiento, en parte integral de la operación MLD (*Multicast Listener Discovery*) RFC2710 MLDv1 1999, RFC3810 MLDv2 2004 y RFC4604 IGMPv3 & MLDv2 2006

Encabezado de opciones de destino

Se usa para la movilidad en IPv6, así como para otras aplicaciones. Únicamente el nodo destino del paquete lo examina y no todos los nodos intermedios en la ruta.

Encabezado de enrutamiento

Se usa en la movilidad IPv6 y en enrutamiento por origen. Podría ser necesario deshabilitar en los enrutadores la opción de enrutamiento por origen IPv6 para proteger contra ataques de denegación [de servicios] distribuida (*DDoS*).

Encabezado de fragmentación

Es fundamental en el uso de comunicaciones que usan paquetes fragmentados (en IPv6, el tráfico origen debe realizar la fragmentación, los nodos intermedios no.)

Encabezado de autenticación

Es similar en formato y uso al encabezado de autenticación usado en IPv4, definido en el RFC2406. No proporciona una garantía de confidencialidad de los datos ya que no provee el cifrado de los mismos, se usa para garantizar la autenticación e integridad, además de la anti-réplica de los paquetes IP.

Encabezado de carga de seguridad de encapsulamiento

Toda la información que lleve el encabezado de seguridad de encapsulamiento (*ESP*) es cifrado y por tal razón es inaccesible para enrutadores intermedios de la red. Se usa para proporcionar confidencialidad, especificando el modo de cifrar los datos y cómo se incluirá el contenido en el paquete IP. También tiene la capacidad de ofrecer autenticación del origen, integridad y anti-réplica de la información contenida.

1.3 Direccionamiento IPv6

Una de las principales motivaciones que llevaron a la creación de IPv6 fue la rectificación de problemas creados por la asignación de direcciones en IPv4.

Se requerían más direcciones, pero mucho más que eso, se deseaba que fuera una manera contemporánea y que reflejara un manejo de redes actuales. En base a esto, no es sorprendente los múltiples cambios que se dieron en el direccionamiento IP. El esquema de direccionamiento es similar en concepto al que se manejó para IPv4, pero fue completamente rediseñado para soportar una red en continuo crecimiento y con nuevas aplicaciones en el porvenir.

Algunos de los aspectos que se conservan, con respecto al modelo anterior, son:

- El ruteo y la identificación de la interfaz de red. El ruteo se lleva a cabo a través de la estructura de direcciones en la red.

- Las direcciones IPv6 están asociadas con la capa de red , del modelo OSI, en las redes TCP/IP, que son distintas de las direcciones físicas de la capa de enlace.

- La interpretación de direcciones y la representación de los prefijos, esto es, las direcciones IPv6 son como las direcciones sin clase (*classsles*) en IPv4, en que son interpretadas con un identificador de red *(NetID)* y un identificador de cliente *(HostID*), un número de prefijo de longitud, usando una notación parecida a la usada en *CIDR* usada para identificar la longitud del identificador de red.

De los cambios que se introdujeron en IPv6, el más celebrado es el incremento en tamaño de las direcciones IP, y por resultado el incremento en el espacio de direcciones disponibles.

En IPv4, las direcciones se componen de 32 bits, los cuales están agrupados en cuatro octetos de bits. El incremento de 32 a 128 bits incrementa este espacio hasta cantidades astronómicas, y con lo cual se pierde la facilidad nemotécnica que se tenia en la versión anterior del IP. Un problema que surge al querer usar la anterior notación , que agrupaba octetos, en notación decimal, es que en esta nueva versión del IP en vez de tener 4 de estos octetos tendríamos 16, con lo que se hace *humanamente* imposible de manejar.

Es por esto que se eligió la notación hexadecimal para la representación de las direcciones IPv6, con lo cual se logra que sean *humanamente* más manejables. En el cuadro 2, se muestran las notaciones existentes y sus variantes.

	0		32		64		96				128
únicamente hexadecimal	805B	2D9D	DC28	0000	0000	FC57	D4C8		1FFF		
ceros suprimidos	805B	2D9D	DC28	0	0	FC57	D4C8		1FFF		
ceros comprimidos	805B	2D9D	DC28	::		FC57	D4C8		1FFF		
notación hibrida	805B	2D9D	DC28	::		FC57	212	200	31	255	

Cuadro 2. “Distintas notaciones existentes para IPv6”

Como se puede apreciar en el cuadro 2, la notación utilizada para las direcciones en IPv6 dista mucho de ser fácilmente manejable, pero ahorrará muchos problemas que se tendrían si se siguiera manejando la notación decimal o binaria. Se muestran las notaciones que se permiten utilizar, los atajos de los que se puede hacer uso si se presentan los casos que correspondan y la última es la notación híbrida que dejaría utilizar la notación decimal habitual en IPv4 embebida en la notación para las direcciones en IPv6.

Los motivos principales que se tomaron en cuenta para dividir el espacio de direcciones en IPv6 fueron asignaciones y ruteo. Esto es, se intento hacer la asignación de direcciones lo mas sencillo posible, ya sea a los proveedores de servicio (*ISP's*), a las organizaciones o a los usuarios finales.

Inicialmente se tenía pensado usar un formato de prefijo (*FP*), se quería dividir el espacio de direcciones IPv6 en bloques variables para diferentes propósitos, aunque rápidamente se dieron cuenta que se empezaría a considerar que el uso de los formatos de prefijo sería el equivalente a las clases de direcciones en IPv4.

Entonces se decidió que los implementadores de hardware con soporte para IPv6 realizarán las decisiones de ruteo en base a los primeros bits de las direcciones. Esto es como IPv6 se suponía que ***no*** debía trabajar, porque se supone que las ubicaciones de estas direcciones están sujetas a cambios y modificaciones. Asi que se decidió remover el término “formato de prefijo” de la norma.

Formato de direcciones IPv6 Unicast Globales

Se anticipaba que las direcciones que más se usarían en IPv6 fueran las direcciones unicast, y es por esta razón que la vasta mayoría del espacio de direcciones IPv6 se dedica a este tipo de direcciones. Un octavo del enorme espacio de direcciones IPv6, las cuales son indicadas por la cadena “001” en los primeros tres bits de la dirección. Surgió la cuestión de como utilizar los 125 bits restantes de las direcciones. Cuando IPv4 fue diseñado, el modelo de asignación de direcciones se basó en una entidad central: la *IANA*. Cualquier organización que deseara obtener un bloque de direcciones se lo tenía que solicitar a esta autoridad central. Pero esto se volvió rápidamente impráctico. Los diseñadores de la nueva versión de la pila de protocolos TCP/IP tuvieron esto en cuenta y se implementaron ventajas en el diseño de las direcciones en IPv6 para reflejar mejor la topología de la red de redes. Algunas de estas fueron:

- Facilidad de ubicación de bloques de direcciones a varios niveles de la jerarquía topológica de Internet.

- Direcciones IP que reflejen automáticamente la jerarquía en que mueven información los enrutadores, permitiendo a los mismos ser facilmente agregables para un ruteo más eficiente.

- Flexibilidad para que las organizaciones, como los proveedores de servicio (*ISP's*), puedan subdividir sus bloques de direcciones para los usuarios.

- Flexibilidad para las organizaciones de usuarios finales para subdividir sus bloques de direcciones para adaptarse a redes internas, tal como hicieron las subredes en IPv4.

- Mayor significado a las direcciones IP, en vez de ser solo una cadena de 128 bits sin significado, es posible observar una dirección y obtener cierta información de ella.

En IPv4, la dirección IP no tiene relación alguna con las tecnologías de red subyacentes de la capa de enlace. Un cliente se puede conectar a una red TCP/IP usando una interfaz de red Ethernet (*NIC*) la cual tiene una dirección física (*MAC*) y una dirección lógica IP, y se puede observar que ambos números no tienen relación alguna de ningún tipo.

Asi con la completa remodelación practicada en las direcciones para IPv6, se aprovechó para poder incluir este nuevo esquema de mapeo de las direcciones físicas como parte de las direcciones lógicas. Con 128 bits, y aún con 45 bits reservados para el prefijo de red y 16 bits para las subredes, quedan 64 bits para usarlos en el identificador de interfaz, análogo este al *hosts ID* bajo IPv4.

Asi como en IPv4 existen direcciones que están reservadas, o son privadas, una pequeña parte del espacio de direcciones IPv6 ha sido designada para direcciones especiales. El propósito de estas direcciones y bloques de direcciones es el de proveer direcciones para propósitos especiales y privados en redes IPv6. Algunos tipos de estas direcciones especiales suelen ser :reservadas, privadas , no específicas y el loopback.

Una porción del espacio de direcciones IPv6 fue reservado por el IETF. A diferencia de IPv4, el cual tiene reservados varios bloques en distintas ubicaciones a lo largo de su espacio de direcciones, en IPv6 el bloque reservado está al principio del espacio de direcciones, aquellos que comienzan con “0000 0000” (o 00 para el primer octeto hexadecimal). Esto representa $\frac{1}{256}$ del total del espacio de direcciones. Las direcciones *IPv4 embebidas* también se encuentran aquí.

IPv6 es compatible con su antecesor IPv4, por ejemplo, para comunicar “islas” de IPv6 con redes IPv4, se pueden emplear túneles. Para soportar la compatibilidad IPv4/IPv6, se desarrolló un esquema para permitir las *direcciones embebidas IPv4* en la estructura de direcciones IPv6. Se usan dos formatos de direcciones embebidas para poder indicarle las capacidades del dispositivo que usa las direcciones embebidas.

Las direcciones IPv4 compatibles con IPv6 son direcciones especiales asignadas a los dispositivos con capacidades IPv6, como aquellos que son de pila doble (*dual stack*) los cuales tienen la capacidad de manejar ambas versiones del protocolo. Asi se podría ver una dirección que se formara con la parte decimal que es la dirección en IPv4 más la parte que se formaría con la dirección en IPv6, como se observa en el cuadro 3.

	0	32	64	96	128
dirección IPv6 en notación hibrida	0 0	0 0	0 0	132.248	59.73
dirección IPv6 en notación comprimida	::132.248.59.73				

Cuadro 3. “Dos formas de notaciones en direcciones IPv6”

Uno de los cambios más significativos en el modelo general de direcciones IPv6 fue la modificación a los tipos básicos de direcciones y como son usadas. Las direcciones *unicast* son la opción para la mayoría de las comunicaciones, así como en IPv4, pero los métodos para las direcciones son diferentes en IPv6. Las direcciones tipo *broadcast* como tipo especial han sido eliminadas. En contraparte, el soporte para las direcciones tipo *multicast* se ha expandido y también aparece un tipo de direcciones nuevo, las direcciones *anycast* que también se pueden usar en IPv4.

Las direcciones tipo *multicast* permiten a un sólo dispositivo mandar datagramas a un grupo de receptores. IPv4 también ofrece soporte a las direcciones tipo *multicast* usando el bloque de direcciones de clase D. En IPv6 este tipo de direcciones se ubican en el bloque *multicast*. Este es $\frac{1}{256}$ del espacio de direcciones que tiene IPv6, que consisten de las direcciones que comienzan con “1111 1111”. Asi cualquier dirección que comience con FF en notación hexadecimal es una dirección *multicast* IPv6.

El formato para las direcciones multicast se observa en la tabla 2.

Campo	Tamaño	Descripción
Indicador	8	Los primeros ocho bits son siempre “1111 1111” para indicar una dirección *multicast*
Banderas	4	Cuatro bits son reservados para el uso de las banderas que pueden indicar la naturaleza de la dirección multicast. En la actualidad los primeros tres de estos bits están en desuso y se ponen en ceros. El cuarto en la bandera “T” (*Transient*). Si está en cero esto indica que la dirección multicast está permanentemente asignada. Si está en uno, significa que una dirección multicast no permanente.
Scope ID	4	Estos cuatro bits permiten tener 16 diferentes valores. Los cuales nos darán una variedad de opciones de alcance, como pueden ser las direcciones globales para toda la Internet o restringidas a una pequeña esfera en particular como pudiera ser una organización.
ID de Grupo	112	Define un grupo en particular

Tabla 2. “Formato para direcciones multicast”

Otra de las características interesantes y de importancia en IPv6, es la capacidad que se le otorga a los dispositivos para autoconfigurarse. En la anterior versión del protocolo, IPv4, es necesaria una configuración manual o en su defecto utilizar mecanismos como DHCP que permiten asignar direcciones, en IPv6 esto se lleva varios pasos más adelante y se permite configurar la dirección IP automáticamente y otros parámetros sin la necesidad de un servidor. También se tiene el mecanismo para reasignar las direcciones (*renumbering*).

Otra de las muchas mejoras introducidas en IPv6 es el mecanismo para administrar dispositivos IP, incluyendo la configuración del cliente. Los dos mecanismos son :

1. ***Autoconfiguración sin estado(stateless)***: un método que permite que el dispositivo se configure así mismo sin intervención de un servidor y mediante un ruteador.

2. ***Autoconfiguración con estado(statefull)***: en este método la configuración del cliente es provista por un servidor.

La *autoconfiguración sin estado (stateless) y la reasignaciones* se define en el RFC 2462, además implementa varias de las nuevas características que trae consigo el IPv6, las cuales incluyen las direcciones locales *(link-local, que es una de los dos tipos de direcciones locales que se tienen), multicast, el protocolo de descubrimiento de vecinos (neighbor discovery [ND] protocol) y la capacidad de generar el identificador de interfaz por medio de la dirección perteneciente a la capa de enlace*, la MAC.

Lo siguiente es un sumario de los pasos tomados cuando se usa la *autoconfiguración sin estado*:

- Generación de direcciones de enlace local (***link-local***): Para el caso de las direcciones de autoconfiguración sin estado estas tienen en sus primero diez bits una cadena del tipo “1111 1110 10”. La dirección generada usa estos diez bits seguidos de 54 ceros y los 64 bits pertenecientes al identificador de interfaz. Usualmente esto se derivará de la capa de enlace, de la dirección MAC o será un “token” generado de alguna otra manera.

- Prueba de unicidad de las direcciones de enlace local (***link-local***): El nodo verifica que la dirección generada no está siendo ocupada por algún otro dispositivo en la red local. Se manda un mensaje de solicitud de vecindad (*neighbor solicitation*) usando el protocolo *neighbor discovery*, el cual escucha por una advertencia de vecindad (*neighbor advertisement*) en respuesta a la verificación que otro dispositivo ya esta usando dicha dirección de enlace local, con lo cual, otra dirección deberá ser generada, o la autoconfiguración falla y otro método deberá ser utilizado.

- Asignación de dirección de enlace local(***link-local***) : asumiendo que se pase una prueba de unicidad de dirección de enlace local, el dispositivo asignará la dirección de liga local a su inetrfaz IP. Esta dirección puede ser usada para comunicación en la red local, pero no hacia la Internet, porque las direcciones de enlace local no son ruteables.

- Enrutador de contacto(***Router Contact)***: el siguiente nodo intentará realizar contacto con el enrutador local para obtener más información acerca de la configuración. Esto se realiza tanto escuchando los mensajes de anuncio de enrutador (*router advertisement*) enviados periódicamente por los enrutadores.

- Enrutador de dirección(***Router Direction***): este enrutador provee de dirección al nodo para proceder con la autoconfiguración, si en esta red se usa la autoconfiguración con estados y/o advertir la dirección del servidor DHCP que si se está usando. Opcionalmente le debe de indicar al cliente como determinar su dirección global de Internet.

- Configuración de la dirección global: asumiendo que la autoconfiguración sin estado esté en uso en la red, el cliente se autoconfigurará con su dirección de Internet global única. Esta dirección está formada por un prefijo de red, que proveerá el ruteador al cliente, combinada con el identificador del dispositivo como se comentó anteriormente.

Este método tiene muchas ventajas sobre las configuraciones manuales o en base a un servidor, y es particularmente útil en el soporte de dispositivos móviles, los cuales se desplazan por varias redes y pueden adquirir direcciones válidas sin necesidad de conocer los servidores locales o prefijos de red.

La reasignación de direcciones (***renumbering***) es un método relacionado con la autoconfiguración. Como la configuración de un cliente, se puede implementar usando protocolos como DHCP, a través del uso de direcciones IP otorgadas temporalmente que expiren después de un tiempo. Bajo IPv6, las redes pueden ser reasignadas teniendo en los enrutadores intervalos que expiren para prefijos de red cuando la auto-configuración este realizada.

Dada la importancia de la pila de protocolos TCP/IP y lo significativos que están siendo los cambios realizados en IPv6, este cambio no está ocurriendo de una una sola vez, se está dando una transición y la coexistencia de ambas versiones del IP.

1.4 Mecanismos de transición de IPv4 a IPv6

Pila Dual (*Dual Stack*):

Este es uno de los mecanismos de transición que se tienen, en el cual se soportan ambas versiones del protocolo tanto IPv4 como IPv6.

Túneles (Tunnels): Para este mecanismo de transición se permiten dos tipos de túnel

- Automáticos

  Las direcciones IPv6 de los hosts alcanzables deben ser compatibles con IPv4. Asi mismo las direcciones IPv4 que se usen para formar direcciones IPv6 deberán ser enrutables. Establecer túneles automáticos entre cualquier par de hosts IPv6, cuyas direcciones IPv6 sean compatibles con IPv4 permite a los hosts el encapsulamiento IPv6 en IPv4 en puntos finales, sin que se requiera a los enrutadores que estén en la ruta que soporten IPv6.

- Manualmente configurados

  Estos túneles son establecidos y configurados manualmente. Los túneles configurados manualmente no requieren de las direcciones compatibles con IPv6 ni compatibilidad con IPv4.

1.5 Ruteo en IPv6

En el enrutamiento en IPv6 es de notar que el anuncio de enrutador (***router advetisement***) es una de las áreas más importantes de la operación del protocolo y que no tiene análogo en IPv4. Es un mecanismo que provee información de configuración a los hosts de la red. Se pueden ajustar algunos parámetros para que respondan a situaciones específicas, como podría ser el uso de dispositivos inalámbricos, donde puede ser deseable el ajuste del tiempo de vida de los prefijos a fin de que pasado un tiempo después de dejarlos de detectar a los primeros, sean dados de baja.

En caso de que se estuvieran usando múltiples enrutadores, pueden surgir algunos problemas ya que dichos enrutadores tratarán de asignarle a un dispositivo distintas direcciones, y a pesar de que se ha implementado la preferencia alta, media y baja para este tipo de situaciones, esto no fue incluido en el *RFC 2461*, y por lo tanto no necesariamente aparecerá implementado como parte del protocolo.

En el caso de los protocolo de enrutamiento, el problema es menor ya que estos protocolos (*BGP, OSPF y demás protocolos*) trabajan de manera similar que en IPv4. Aunque en ocasiones podrían convertirse en una pesadilla los problemas que surgen si es que uno de los protocolos existentes en una red de pila dual, toma decisiones distintas, o podría ser que esto provea una ventaja al ofrecer redundancia a la hora de buscar solución en algún conflicto causado por alguna de las dos versiones.

De momento el uso para el enrutamiento del tipo *multicast,* sigue siendo experimental.

1.6 Seguridad en IPv6

En IPv6, la seguridad es parte del protocolo, en la forma de seguridad IP (***IPSec***). Se había estado buscando un mecanismo que permitiera separar la seguridad de la capa de aplicación, y esta fue la opción que se tomó. En IPv4 no es nativo el soporte, pero para IPv6 esto si lo es.

IPSec es un conjunto de protocolos que son parte de la especificación del IPv6 y dado la fuerte necesidad de seguridad en IPv4 éste fue adaptado al mismo. Aunque el soporte para IPv4 es opcional y las soluciones propietarias son lo que prevalece. Del otro lado en IPv6, IPSec provee la seguridad de punto final a punto final, se define en el *RFC 2401*.

Dicho de otra manera, la seguridad para IPv6 no es diferente de la seguridad para IPv4. Los ataques más conocidos para IPv4 también se pueden llevar a cabo en IPv6, significando esto que el concepto de seguridad es similar.

En las implementaciones nativas de IPv6 en el *RFC 4301* se hace las especificaciones de los requerimientos de IPsec en el IPv6, pero este no cubre como debe ser el intercambio de llaves *PKI*. También introduce nuevos problemas con los IDS/IPS existentes.

Direcciones Privadas

IPv6 ofrece varias opciones para las direcciones ip que pueden ayudar a los arquitectos de seguridad. Las direcciones privadas pueden ser usadas por aplicaciones cliente para inhibir el rastreo a usuarios (*user tracking*), el cual puede ser útil para proteger comunicaciones externas.

De acuerdo al *RFC 4291*, las interfaces que usan las direcciones de autoconfiguración sin estado (*stateless address autoconfiguration*) generan identificadores de interfaz basados en su identificador ***IEEE EUI-64***. Esto provee un fuerte soporte para la diferenciación única (*uniqueness*), pero esto permite rastrear a una interfaz, incluso si esta cambia de una red a otra, o si el prefijo de la red es cambiado.

Considerese, por ejemplo, un dispositivo móvil que se conecte a diferentes redes inalámbricas en diferentes ubicaciones. Usando IPv4, el dispositivo en cuestión usará DHCP en las diferentes redes y recibirá direcciones completamente diferentes. Si este mismo dispositivo usa la autoconfiguración IPv6 , su dirección de interfaz inalámbrica tendría el mismo identificador de interfaz en cada red. Aún más, el identificador ***IEEE EUI-64***, por estar basado en la dirección MAC del hardware, revelaria qué tipo de dispositivo es.

Una interfaz que acepte conecciones entrantes y que tenga una denominación DNS, no puede tener una dirección privada, pero es posible usar diferentes direcciones para las conexiones de salida. Para el *RFC 494*, *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*, se definió la manera de generar y cambiar dichas direcciones temporales. Los requerimientos importantes son que la secuencia de direcciones temporales e interfaces escogidas deben ser impredecibles y tener baja probabilidad de colisiones con las opciones seleccionadas por otras interfaces.

El método recomendado en el *RFC 4941* trabaja aproximadamente como se menciona a continuación:

1. Obtener el identificador de interfaz que será usado sin este esquema.

2. Aplicar una función criptográfica *hash* a éste valor y también a un valor salvado en históricos o a un número de 64 bits escogido añeatoriamente.

3. Use la salida de la función *hash* anterior, para seleccionar el identificador de interfaz y para actualizar el valor de los históricos.

4. Ejecutar la detección de direcciones duplicadas (DAD).

5. Ajuste los tiempos de vida (*lifetime*) apropiados y agregue el nodo de grupo multicast correspondiente al identificador de interfaz.

6. Continúe usando los identificadores de interfaz prioritarios para las conexiones establecidas pero no para las nuevas.

7. Repita este proceso cuando se conecte a una nueva red o cuando los tiempos ajustados en la iteración previa expiren.

Link encap:Ethernet Hwaddr 00:0C:29:6F:8F:98

inet addr: 192.168.1.3 Bcast: 192.168.1.255 Mask:255.255.255.0

inet6 addr: 2002:2::9048:b971:277c:e16c/64 Scope: Global

inet6 addr: 2002:2::20c:29ff:fe6f:8f98/64 Scope: Global

inet6 addr: fe80::20c:29ff:fe6f:8f98/64 Scope: Global

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

Cuadro 4 “Ejemplo de direcciones IPv6 privadas. “

En el ejemplo del cuadro 4 se ve una dirección privada IPv4 no ruteable /24, una dirección global enrutable /64 IPv6, una segunda dir. global enrutable /64 IPv6 basada en su dir. IEEE y la dirección local link basada en su dir. IEEE

Por distintas razones, este mecanismo de extensiones privadas deberá ser usado con cuidado si es que se llega a usar. Algunas de las razones son las siguientes:

- Que tanta privacía es provista actualemte es cuestionable. En redes pequeñas que no cambian mucho, cualquiera que pueda observar el tráfico de red podrá correlacionar las actividades con mucha precisión, sin importar si las direcciones cambian periódicamente o no. Un observador podría determinar que tan seguido las interfaces están generando estas nuevas direcciones.
- En algunas redes, los administradores pueden querer tener control de que está conectado y por consiguiente de las direcciones usadas. Las políticas locales de seguridad podrían dictar que para propositos de auditoría o computo forense, todas las direcciones deberán ser asignadas centralizadamente y guardadas en bitácora. En tal caso, es mejor no permitir las direcciones privadas ni la autoconfiguración sin estado de direcciones sino requerir usar *DHCPv6* para la asignación de direcciones.
- En general, las políticas de seguridad empresariales no extienden el privilegio de comunicaciones privadas a usarios que estan en equipos de la empresa o que están accediendo a la red empresarial. En estos casos, la meta del análisis forense y la seguridad pueden ser vistas con más importancia que proteger la privacía del usuario que navega por la Interet. Esta será una decisión que deberá dejarse en manos del departamento encargado.
- Una buena práctica de administración de redes es aplicar filtrado, esto es, no permitir paquetes sin una dirección válida de origen dentro de la red administrada. Algunos ataques distribuidos de denegación de servicios (DDoS) han usado direcciones de origen forjadas con prefijos válidos. Las direcciones privadas pueden ser difícies de distinguir de las direcciones usadas en estos ataques sin medidas adicionales, como la limitación de transferencia o la verificación de ruta en reversa completa.

Direcciones Generadas Criptográficamente

Las direcciones generadas criptográficamente (CGA), también llamadas direcciones basadas en *hash*, proveen un método de comprobar la pertenencia de una dirección de origen en un paquete. La idea se basa en escoger un par de llaves, pública y privada, capaces de crear una firma digital con la llave privada y verificarla con la pública. Entonces, la llave pública (y junto con otros parámetros), es usada para generar un identificador de interfaz, esta llave es insertada dentro del paquete, y el paquete es firmado con la llave privada. A la recibimiento del paquete, la llave pública puede ser usada para verificar la firma y la dirección. Un atacante no puede firmar un paquete forjado sin la llave pública.

Cuatro procesos son necesarios para hacer funcionar esto:

El emisor debe:

1. Generar un par de llaves (pública y privada) y la dirección correspondiente.
2. Insertar la llave pública dentro de un paquete y firmarlo con la llave privada.

El receptor debe:

1. Verificar que la dirección de origen corresponde a la llave pública.
2. Validar la firma con la llave pública.

Notese que usando CGA no se prueba la identidad de uno, pero muestra que la misma entidad (aquella con la llave privada firmante) generó cada paquete y que los paquetes no fueron subsecuentemente modificados por otras entidades. Este punto trata de que nadie sin la llave privada puede usar CGA legitimamente.

CGA ha sido estandarizado como el principal bloque para el aseguramiento del protocolo *RFC 3971* de descubrimiento de vecindad en IPv6 (*IPv6 Secure Neighbor Discovery SEND*), y han sido propuestos para el uso con el protocolo *SHIM6* (*Site Multihoming for IPv6*). En todos los casos el algoritmo hash especificado para CGA es SHA-1, el algoritmo de firma es RSA, y el formato de firma sigue el estandar de cifrado de llave pública (PKCS) #1, versión 1.5, descrito en el *RFC 3447*.

Las CGAs se especificaron en los RFC 3972 y RFC 4581. Las implementaciones necesitan generar y almacenar valores criptograficos seguros para usar estos protocolos con seguridad. Lease el RFC 4086 para una discusión acerca de generar valores pseudo-aleatorios.

Algunas vulnerabilidades en IPv6

La IETF manejo la seguridad como una parte importante en el diseño de los estandares para IPv6. El trabajo para agregar confidencialidad e integridad en IPv4 llevó al desarrollo de IPSec. IPv6 junto con IPSec resuelve este problema, pero IPSec no ha manejado otras debilidades encontrada en la pila del protocolo TCP/IP.

IPv6 no resuelve muchos ataques de capa 2 tradicionales, tales como el sniffeo de tráfico, *traffic flooding*, *man-in-the-middle*, *rogue devices* y ataques de desbordamiento de tabla ARP. Algunos de estos ataques se manejan en IPv6 parcialmente, mientras que otros ataques, similares en naturaleza, eplotan diferentes caracteristicas.

Mientras que la mayoría de los sistemas operativos soportan IPv6 desde 2003, la pila de protocolos de estos sistemas operativos no han sido probados totalmente. Una revisión en la NVD (*National Vulneratibilities Database*) muestra vulnerabilidades en la pila de protocolos de los sistemas operativos mas populares. Cuando las nuevas vulnerabilidades son expuestas, los vendedores realizan actualizaciones. Como cualquier código nuevo, los vendedores necesitan tiempo para estabilizar y *endurecer* (*harden*) el código.

Los ataques de capa 2 y 3 en IPv4 son posibles porque se asumió que todos los nodos de red se comportarían de manera *confiable.* IPv4 usa ARP para asociar las direcciones físicas a las direcciones lógicas. Este presuposición permitió que los ataques interfirieran con la resolución de direcciones IP y la asociación de direcciones lógicas y físicas. IPv6 no usa ARP para mapear direcciones IP con interfaces físicas, en vez de eso usa ICMPv6. IPv6 usa ICMP para el descubrimiento de vecindad (ND) y lel proceso de autoconfiguración de direcciones sin estadoque asocia direcciones físicas y lógicas. Pero IPv6 es aún vulnerable a los ataques de capa 3.

El tamaño de las subredes en IPv6 pueden presentar su propio desafío de seguridad. Estos desafíos se discuten en el *RFC 5157 IPv6 Implications for Network Scanning.* El tamaño de las subredes es mucho mayor de lo que fue en IPv4; una subred por default en IPv6 puede tener 2^{64} direcciones.

Capítulo 2

IPSec

Una de las principales preocupaciones en el IPv6 es la seguridad. Durante la existencia del IPv4 se manifestaron sus debilidades y flaquezas en su diseño puesto que nunca se contempló la seguridad como aspecto fundamental. Esto ha cambiado y para esta versión se hace imperativa la implementación de un conjunto de protocolo destinados a ofrecer seguridad desde el inicio de la conexión hasta el punto final de la misma, y a esto se le ha llamado IPSec. Por lo cual en este capítulo haremos una breve exposición de dicho conjunto de protocolos y sus características.

Capítulo 2 IPSec

El conjunto de protocolos en el IPSec (***Internet Protocol Security***) tienen como finalidad hacer más seguras las comunicaciones dentro del IP, autenticando y cifrando cada paquete del flujo de datos. Es un protocolo diseñado por la IETF, que se definió en el *RFC4301*. También incluye protocolos para establecer la autenticación mutua entre agentes al inicio de la sesión y la negociación de llaves criptográficas durante la sesión. IPSec puede ser usado para proteger flujos de datos entre un par de hosts, ya sean servidores o clientes, entre un par de puertas de enlace (*gateway*) de seguridad, ya sean firewalls o ruteadores, o entre una puerta de enlace un hosts. IPSec es un esquema de seguridad en modo dual, de punto a punto, operante en la capa 3 del modelo *OSI*.

2.1 Componentes

Los protocolos que usa IPSec son:

IKE (***Internet Key Exchange***) para llevar a cabo una asociación de seguridad (***SA***, *security asociation*) al llevar las negociaciones de los protocolos y algoritmos, además de generar las llaves de cifrado y autenticación que serán usadas por IPSec.

Encabezado de Autenticación (***AH*** *authentication header*) para proveer integridad y autenticación de origen de datos para los datagramas IP y para proveer protección contra los ataques de respuesta (*reply attacks*).

El encabezado de carga de seguridad de encapsulamiento (***ESP*** *encapsulating security payload*) para proveer confidencialidad, autenticación de origen de datos, integridad sin conexión (*conectionless*), un servicio anti-respuesta, un tipo de secuencia parcial de integridad y una limitada confidencialidad de flujo de tráfico. **ESP** también soporta configuraciones de sólo cifrado y sólo autenticación, aunque esto se desaconseja. A diferencia del encabezado de autenticación ***ESP*** no protege el encabezado de paquete IP. En el modo de túnel, donde el paquete completo original es encapsulado con un nuevo encabezado de paquete, la protección que ofrece ***ESP*** abarca a todo el paquete, incluyendo el encapsulado, mientras que el encabezado exterior continua sin protección. Este encabezado opera directamente al principio del IP, usando el número 50 del IP. Observamos la forma del encabezado de carga seguridad en el cuadro 4 de la siguiente página.

0-7 bits	8-15 bits	16-23 bits	24-31 bits
Índice de parámetros de seguridad (*SPI*)			
Números de Secuencia.			
Carga útil de datos (variable)			
	Relleno (0-255 bytes)		
		Long. de relleno	Prox. encabezado
Autenticación de datos (variable)			

Cuadro 5 "Encabezado de Carga de Seguridad"

2.2 Trama IPSec

La trama del conjunto de protocolos IPSec se compone de los encabezados de autenticación y el encabezado de próximo encabezado.

Encabezado de Autenticación (***AH***)

El encabezado de autenticación es parte del conjunto de protocolos Elipse. Este garantiza la integridad sin conexión (*conectionless*) y autenticación del origen de los paquetes IP. Además puede, opcionalmente, proteger contra los ataques de respuesta (*reply attacks*) usando la técnica de deslizamiento de ventanas (*sliding windows*) y descartando paquetes viejos. El encabezado de autenticación protege la carga IP y todos los campos de encabezados de un datagrama IP con excepción de los campos que vayan a modificarse durante su trayectoria.

En IPv4, los campos de encabezados variables y por lo tanto inautenticados, incluyen al campo (***DSCP*** *differentiated services code point) de apuntador de código de servicios diferenciados, tipo de servicios (TOS type of servicie*), banderas, fragment *offset, t*iempo de vida (***TTL*** *time to live*) y el encabezado de suma de verificación *(Checksum).*

El encabezado de autenticación trabaja directamente al principio del *IP*, usando el número de protocolo *IP* 51.

0-7 bits	**8-15 bits**	**16-23 bits**	**24-31 bits**
Prox. Encabezado	Long. carga útil	Reservado	
Índice de parámetros de seguridad (*SPI*)			
Número de secuencia.			
Autenticación de datos (variable)			

Cuadro 6 " Encabezado de Autenticación "

El encabezado de Próximo encabezado, que se muestra en el cuadro 5, es un campo de ocho bits que identifica el tipo de la próxima carga útil después del encabezado de autenticación (*AH*). El valor de este campo se escoge del conjunto de números de protocolo IP que se definió en el más reciente *RFC* de asignación de números del *IANA* (*Internet Assigned Number Authority*).

- La longitud de carga útil es el tamaño de del paquete del encabezado de autenticación.

- La parte de reserva se utilizará en el futuro, se llenará de ceros hasta entonces

- El índice de parámetros de seguridad (*SPI*) identifica a los parámetros de seguridad, los cuales en combinación con la dirección ip identifica la asociación de seguridad implementada en este paquete.

- El número de secuencia es un número monotónico incremental, usado para prevenir ataques de respuesta.

- La autenticación de datos contiene el valor de chequeo de integridad (*ICV integrity check value*) necesario para autenticar el paquete.

2.3 Arquitectura IPSec

La arquitectura de IPSec especifica la base en la cual todas las implementaciones serán construidas y define los servicios de seguridad proveídos por IPSec, cómo y dónde pueden ser usados, cómo serán los paquetes construidos y procesados, y la interacción de procesamiento de IPSec con las políticas de seguridad.

Esta arquitectura define hasta que nivel podrá ser definido y usado por el usuario de acuerdo a las políticas de seguridad, permitiendo que cierto tráfico sea identificado para recibir el nivel de protección deseable.

IPSec fue definido para proveer un alto nivel de seguridad criptográfica, para ambas versiones del IP. Componiéndose de los siguientes encabezados que proveen seguridad en el tráfico: el encabezado de autenticación **AH** (***Autentication Header***) y el encabezado de encapsulamiento de carga útil de seguridad **ESP** (**Encapsulating Security Payload**), incluyendo los protocolos que generan y administran las llaves de cifrado, IKE (**Internet Key Exchange**) e ISAKMP (**Internet Security Key Association and Key Management Protocol**).

IPSec maneja a través de las asociaciones de seguridad SA (**Security Association**) su esquema de interoperabilidad, controladas por el índice de parámetros de seguridad SPI (**Security Parameter Index**), que se norman por las políticas de seguridad SP (**Security Policy**); las cuales se almacenan en bases de datos. También se define como dichas bases de datos se relacionarán con las distintas funciones de procesamiento de IPSec, y como distintas implementaciones del IPSec pueden coexistir.

Las políticas establecidas pueden tener dos vertientes, las estáticas y las dinámicas. Las políticas estáticas serán previamente establecidas y contendrán valores fijos, los cuales establecerán los parámetros que se negociarán; estableciendo canales seguros. O pueden ser dinámicas, en cuyo caso se podrán usar protocolos de administración de llaves como ISAKMP.

2.4 Protocolo del encabezado de autenticación

El encabezado de encapsulamiento de carga de seguridad, se define en el *RFC 4303*, tiene como objetivo principal proporcionar confidencialidad, especificando el modo de cifrar los datos que se desean enviar y como este contenido cifrado se incluye en un datagrama IP. Puede ofrecer servicios de antiréplica, integridad y autenticación del origen de los datos incorporando un mecanismo similar al a AH. El encabezado ESP se inserta después del encabezado IP y antes del encabezado del protocolo de capa superior (modo transporte) o antes del encabezado IP encapsulado (modo túnel).

La función de cifrado dentro del protocolo ESP es desempeñada por un algoritmo de cifrado de llave simétrica. Típicamente se usan algoritmos de cifrado por bloques (DES), de modo que la longitud de datos a cifrar tenga que ser un múltiplo de tamaño de bloque (8 o 16 bytes, en la mayoría de los casos). Por esta razón existe un campo de relleno cuya finalidad es añadir caracteres de relleno al campo de datos para ocultar así su longitud real, y por lo tanto las características del tráfico.

El encabezado de autenticación de definió en el RFC 4302, es un encabezado de IPSec usado para proporcionar integridad en los datos, autenticación en el origen de los datos y opcionalmente servicios de anti-réplica en los datagramas IP. No proporciona ninguna garantía de confidencialidad.

Se suele situar justo antes de los datos, de forma que los proteja de posibles atacantes. Ha sido diseñado de forma muy versátil, de manera que puede incluirse antes que otros encabezados para asegurar que las opciones que acompañan al datagrama sean las correctas.

2.5 Métodos de autenticación

La autenticación en IPSec se logra a tráves de algoritmos de autenticación, tales como MD5, SHA-1, HMAC, RIPEMD-160.

2.6 Asociaciones de seguridad y políticas

Una SA es la forma básica de comunicación en IPSec refiriéndose a un contrato entre dos entidades que desean comunicarse en forma segura. Las SA determinan los encabezados de IPSec a utilizar, las transformaciones, las llaves y la duración de validez de dichas llaves. Las SA son de un solo sentido, cada entidad con IPSec tendrá una SA para el tráfico entrante y otra para el trafico saliente; además de que son específicas para cada encabezado. Cuando se implementa IPSec, se crea una base de datos de las SA denominada SAD donde se almacenan todas las SA de dicha implementación.

La manera que tiene las SA de identificarse de manera única, es a través de los SPI. Estos son entidades de 32 bits, por los cuales se comunican dos entidades de manera segura e indicarán los parámetros usados, tales como llaves y algoritmos.

Para el manejo de las SA's se establecen dos tareas principalmente: creación y eliminación; que a su vez se pueden ejecutar de manera manual o dinámica a través de un protocolo de intercambio de llaves como IKE. La creación de llaves se lleva a través de la negociación de los parámetros de las SA y la correspondiente actualización de la SAD. El manejo manual de llaves es obligatorio en toda implementación, el proceso de asignación del SPI y la negociación de parámetros es totalmente manual y permanecerán hasta que sean manualmente borrados. Para el manejo dinámico de las llaves se utiliza IKE.

2.7 Protocolo de intercambio

El conjunto de protocolos que forman IPSec están diseñados con capacidad de expansión que dan servicios de seguridad como el control de acceso, integridad, confidencialidad y autenticación. El mismo es capaz de proteger paquetes IP entre hosts y gateways, gaterías, hosts, etc. Este conjunto de protocolos puede ser implementado en IPv4 de manera opcional, y aunque en IPv6 se debería implementar de manera obligada, esto también puede no serlo.

Una de las características de IPSec es su posibilidad de acoplamiento a otras tecnologías, aunado al hecho de que es posible cambiar los algoritmos criptográficos estándar por otros más robustos.

IPSec maneja una especificación de arquitectura que deberá ser la base de todas las implementaciones, definiendo los servicios que esta proveerá, como se procesara la información y dónde, además de como se deben de definir las políticas de seguridad a usar en la misma. Ha sido diseñado para proveer seguridad criptográfica para ambas versiones del IP. Tiene dos encabezados que proveen la seguridad del tráfico de información (*AH* [authenticaction header] y *ESP [encapsulating security payload]*), protocolos que generan y administran las llaves (ISAKMP [internet security association and key managment protocol] e IKE [intenet key exchange].

El esquema de interoperabilidad de IPSec se maneja a través de SAs (Security Associations) las cuales son controladas por un SPI (Security Parameter Index), y regidas por un SPs (Security Policy) que se configuran previamente; tanto las SAs como las SPs se almacenan son almacenadas en sus respectivas bases de datos: SAD para las asociaciones de seguridad y SPD para las políticas. La arquitectura de IPSec también define la interacción que hay entre estas bases de datos con las diferentes funciones de procesamiento de IPSec (cifrado y descifrado) y define cómo varias implementaciones de IPSec pueden existir.

Los parámetros que se negocian para establecer los canales seguros se indican bajo las políticas preestablecidas dentro de un esquema de funcionamiento estático con valores fijos y previamente establecidos, o bien, en un esquema de funcionamiento dinámico utilizando un protocolo de administración de llaves como ISAKMP (Internet Security Association and Key Managment Protocol). Estas políticas determinan si dos entidades son capaces de comunicarse entre sí y cuál sería la transformación a usar en un caso dado.

IPSec proporciona los siguientes servicios de seguridad:

Control de acceso

Previene el uso no autorizado de recursos, garantizando que sólo acceden a la información y a los recursos los usuarios que tiene permiso para ellos.

Integridad

Implica que los datos no puedan ser modificados o corrompidos de manera alguna desde su transmisión hasta su recepción en una comunicación.

Autenticación

Definen mecanismos para garantizar la procedencia de la información, de modo que se puede verificar que realmente es el remitente autorizado quien lo envió.

Protección a la réplica

Asegura que una transacción sólo se pueda llevar a cabo una vez, a menos que se autorice una repetición de la misma. Nadie debería poder grabar una transacción para luego replicarla para aparentar múltiples transacciones del remitente original, por ejemplo.

Confidencialidad

Implica que la información sea accesible únicamente por las entidades, sistemas o personas autorizadas, asegurando la privacidad de la información al no ser consultada por terceras personas.

Confidencialidad limitada en el flujo de tráfico

Este servicio se refiere a ocultar las direcciones fuente y destino, la longitud del mensaje, o la frecuencia de la comunicación. En el contexto de IPSec, usando ESP en modo túnel, especialmente en un gateway de seguridad, puede proporcionar un cierto nivel de confidencialidad en el flujo de tráfico.

IPSec proporciona un nivel de seguridad común y homogéneo para todas las aplicaciones, además de ser independiente de la tecnología física empleada. Cuando se implementa IPSec en un firewall o enrutador, éstos proporcionan una fuerte seguridad que puede ser aplicada a todo el tráfico que cruza el perímetro.

Por otro lado, al estar implementado en la capa de red, debajo de los protocolos TCP/UDP resulta “transparente” para las aplicaciones, es decir, no hay necesidad de realizar alguna configuración desde el punto de vista de usuario final ni del servidor. También IPSec tiene la capacidad de ofrecer seguridad individual si ésta fuese indispensable, resultando útil para los empleados que accedan a la red desde el exterior vía telefónica. Además, es posible asegurar una subred virtual dentro de una organización para las aplicaciones más sensibles.

Facilita el comercio electrónico de negocio a negocio al proporcionar una infraestructura segura sobre la cual realizar transacciones usando cualquier aplicación, por ejemplo las extranets.

Los algoritmos permitidos para la protección con IPSec, tanto los usados para autenticación como los usados para cifrado, idealmente desempeñan dos metas incompatibles: proveer máxima protección contra una gran variedad de ataques matemáticos, de análisis criptológico y de fuerza bruta; y por otro lado, requerir un procesamiento mínimo en el lado de cada participante dentro de la comunicación. Aunque los documentos de IPSec decretan algoritmos específicos para proveer un grado estándar, con seguridad interoperable, se pueden implementar algoritmos adicionales ya sea para dominio público o privado.

Todos los algoritmos son algoritmos de bloque, empiezan en el inicio del mensaje y cada bloque es procesado uno a la vez. El tamaño del bloque es parte de la definición de cada algoritmo, donde el más común es de 8 bytes (64 bits). Cada bloque pasa de cierto modo por algún procesamiento repetitivo donde cada iteración de ese procesamiento es conocido como ciclo. El número de ciclos es algunas veces considerado como una característica importante en la criptografía de un algoritmo. Cada ciclo, en turno, consiste de una función de ciclo, la cual es un procesamiento que constituye cada ciclo del cifrado. La función de ciclo puede ser simple y sencilla, o extremadamente compleja. Algunos algoritmos tienen múltiples funciones de ciclo que se pueden aplicar a uno o más ciclos.

En muchos algoritmos, la llave secreta más completa no es usada como función hash o para cifrar cada bloque, sino para generar múltiples sub-llaves, o ciclos de llave donde a su vez cada ciclo puede incorporar una o más sub-llaves. Si cada bloque fuera cifrado o manejado por una función hash separadamente, se presentarían ataques más fáciles, ya que el contenido de algunas partes del paquete de Internet serían conocidas. En el caso de una función hash, el hash final se debe reflejar en todos los bits de todo el bloque de entrada, no solo en el último bloque. En el caso de un algoritmo de cifrado, si cada bloque es descifrado separadamente, sin hacer referencia a ningún otro bloque, los bloques previsibles pueden ser atacados más fácilmente una vez que la llave fue conocida y todo el bloque puede ser descifrado. Por esta razón, todo algoritmo de manera obligatoria en IPSec incorpora dentro de su definición un mecanismo de retroalimentación, es decir, el cifrado o autenticación de cada bloque tiene como una de sus entradas la salida calculada criptográficamente del bloque previo.

La seguridad de los algoritmos criptográficos dependerá de la complejidad de su criptografía y de su robustez. Sin embargo, un algoritmo criptográfico no es suficiente para garantizar la seguridad de las comunicaciones debido a que varios factores juegan un papel muy importante, como por ejemplo, la implementación en hardware o software, o bien, la generación de llaves secretas que deberán tener una apropiada longitud, complejidad y ser generadas, intercambiadas, administradas y almacenadas de una manera segura.

El protocolo IPSec ha sido diseñado en forma modular de modo que se puedan seleccionar determinados algoritmos para cifrado y autenticación sin afectar a otras partes de la implementación.

Sin embargo, han sido definidos algunos algoritmos de manera estándar para soportar todas las implementaciones y asegurar la interoperabilidad en el mundo global de Internet, como son AES (en etapa de evaluación) para sustituir a DES y 3DES, considerados actualmente para cifrado, así como MD5 y SHA-1 como funciones hash para autenticación. Además es perfectamente posible usar otros algoritmos que se consideren más seguros o más adecuados para un entorno específico, como por ejemplo IDEA o Blowfish.

Para los algoritmos de autenticación se utilizan las funciones hash (o primitivas hash), cuya funcionalidad es usada principalmente para resolver el problema de integridad y autenticidad del origen de los mensajes.

Una función hash o "función resumen" es un algoritmo que, aplicado a un mensaje determinado, crea una representación digital o hash de una longitud fija mucho menor que el mensaje original, pero substancialmente único a él, de tal manera, que no sea factible, dado solamente el hash, reconstruir el mensaje original, es decir, las funciones hash son de una sola dirección. Un simple ejemplo de una función hash sería contar el número de letras del mensaje, si es par asociamos un 0 y si es impar un 1. El principal inconveniente de este sistema es que pueden existir colisiones (dos mensajes diferentes producen la misma salida) por lo que conviene que las funciones tengan un rango de salida lo suficientemente grande (128 bits o más) para poder considerarlas libres de colisión.

2.8 Implementaciones

IPSec puede ser implementado en hosts, en conjunto con un enrutador, o con un firewall (para crear gateways de seguridad). La implementación es configurada dependiendo de los requerimientos de seguridad de los usuarios. A continuación se menciona la implementación de IPSec en varios dispositivos de red (hosts y enrutadores). La implementación en hosts es más útil cuando se desea una seguridad punto a punto; sin embargo, en casos cuando la seguridad se desea sobre una parte de la red, es mejor la implementación en enrutadores que incluyen VPNs e intranets.

Implementación en hosts:

La implementación en hosts tiene las siguientes ventajas:
- Provee una seguridad punto a punto.
- Capacidad de implementarse en todos los modos de IPSec.
- Proporciona seguridad en el flujo de datos.
- Capacidad para conservar la autenticación de los usuarios en las conexiones establecidas por IPSec.

Esta implementación puede ser clasificada en dos distintas subimplementaciones:

Implementación integrada con el Sistema Operativo (OS):

Como IPSec es un protocolo de nivel de red, puede ser implementado como parte del mismo, donde IPSec necesita los servicios del nivel IP para construir el encabezado IP. Este modelo es idéntico a la implementación de otros protocolos del nivel de red como ICMP.

Implementación que se coloca entre los niveles de red y de enlace de la pila del protocolo .

Se denomina implementación BITS (Bump in the Stack), y es utilizado para que las compañías encargadas de dar soluciones en VPN e intranets puedan proporcionan una solución completa, dado que la solución que se integra con el OS limita las capacidades para proporcionar soluciones avanzadas.

Implementación en enrutadores

La implementación en enrutadores tiene la capacidad de proporcionar seguridad al flujo de paquetes entre dos redes sobre una red pública, como lo es Internet, por medio de un túnel; además de autenticar y autorizar a los usuarios que entran a la red privada para comunicarse sobre Internet construyendo sus VPN o intranets.

Existen dos tipos de implementación en enrutadores:

1. Implementación nativa: Esta implementación es análoga a la implementación en hosts integrada con el OS. En este caso, IPSec es integrado con el software del enrutador

2.“Bump in the Wire” (BITW): Es similar a la implementación BITS, pero en este caso IPSec es implementado en un dispositivo de cifrado externo dedicado conectado a la interfaz física del enrutador. Este dispositivo normalmente no ejecuta ningún algoritmo de ruteo, sino solamente es usado para asegurar los paquetes.

A la fecha existen diversas implementaciones; sin embargo, la mayoría limitadas a la aplicación de VPNs únicamente de forma nativa, por lo que IPSec es denominado por algunos como el "protocolo VPN". En los últimos años han emergido proyectos para implementar seguridad en sistemas operativos, usando esquemas BITS, en busca de brindar una plataforma base de seguridad que sea independiente de las aplicaciones utilizadas por el usuario.

Capítulo 3

VPN

Las redes virtuales privadas han existido desde hace un tiempo, y sus usos se han diversificado conforme estas se adaptan a las distintas tecnologías emergentes. Parte de esto hace necesario que se expongan sus características y posibilidadaes, por lo que en el presente capítulo mencionaremos sus principales características, requerimientos y algunas de sus ventajas y desventajas.

Capítulo 3 VPN

Una red privada virtual **VPN** (***Virtual Private Network***), se le conoce al tipo de red que permite una extensión de una red local sobre una red pública o no controlada, como por ejemplo Internet.

Las VPNs surgen ante la necesidad de las organizaciones o corporativos de proveer a su personal con la capacidad de acceder a su infraestructura de red interna, o intranet desde cualquier lugar y en cualquier momento de forma segura. Así las VPNs aparecen como las redes privadas con la capacidad de utilización de la infraestructura de Internet para el transporte de datos, implementando técnicas de seguridad para mantener la confidencialidad de los datos que se manejan entre los usuarios.

Ahora es posible implementar distintos tipos de VPNs, como las VPN sitio-a-sitio, VPNs de acceso remoto, VPNs LAN-2-LAN, VPNs confiables, VPNs seguras, L1VPNs, L2VPNs, L3VPNs, VPNs VPWS, VPNs VPLS, VPNs IPLS, VPNs basadas en esquemas de red, VPNs basadas en C(P)E, VPNs multiservicio, VPNs suministradas por el usuario, VPNs Internet, VPNs Intranet, VPN extranets, VPNs punto-a-punto, VPNs multipunto-a-multipunto, VPNs orientadas a la conexión, VPNs connectionless, etc. Además hay redes virtuales basadas en L2TPv3, AToM, capa 3 MPLS, L2F, L2TPv2, PPTP y SSL.

Una **VPN** da la capacidad de proveer los servicios de redes privadas para organizaciones tales como los proveedores de servicio de Internet **ISPs** (*Internet Service Providers*) o los proveedores de red en la dorsal que es conocido como la VPN dorsal (V*PN backbone*) y es usado para transportar tráfico de múltiples VPNs, así como posible tráfico no VPN.

Las VPNs suministradas usando tecnologías tales como **Frame Relay** y circuitos virtuales **VC**-**ATM** (*virtual circuit-Asynchronous Transfer Mode*) han estado disponibles por mucho tiempo, pero en los años recientes las VPNs IP son más y más populares.

Es importante destacar la seguridad como factor importante al establecer las VPN, así como proporcionar y garantizar la autenticación, confidencialidad e integridad dentro del canal de comunicación.

Esta autenticación se resume a saber quien se encuentra en el otro extremo, así como el nivel y facultades de acceso que debe de tener. Garantizar la integridad, es decir, que la información no sufra alteraciones y para eso se utilizan algoritmos especializados.

Debido a una posible interceptación de datos a través del canal se debe de garantizar la confidencialidad de esta información, por eso es necesario establecer un cifrado de los datos, y así la información sólo se entenderá para las partes involucradas, siendo inútil para el intruso a la red.

3.1 Clasificación de las VPN

3.1.1 VPN de acceso remoto

Este modelo consiste en que los usuarios se conectan desde un sitio remoto y se utiliza internet como un vínculo de acceso y después de ser autenticados se puede decir que el nivel de acceso que poseen es como el de una red local.

3.1.2 VPN punto a punto

En este modelo la arquitectura a seguir es la de conectar los nodos remotos con la matriz o punto central. El servidor VPN siempre debe de tener un vínculo permanente con Internet y debe de aceptar las conexiones provenientes de los sitios y establecer el llamado túnel VPN; mientras que los puntos externos deben de utilizar los servicios de su proveedor local de internet por medio de banda ancha, a este fenómeno también se le conoce como túneleo (tunneling).

Lo anterior permite tener una cuenta de acceso seguro en la máquina con la que se quiere comunicar los datos.

3.1.3 VPN interna

Esta opción tiene las mismas cualidades de una VPN tradicional, la única diferencia es que en lugar de utilizar internet como medio de acceso, utiliza la red local del edificio donde se encuentra, con lo cual su nivel de seguridad es mayor que cualquier red WiFi.

3.1.4 VPN basada en firewall

Este tipo de VPN aprovecha los mecanismos de seguridad del servidor de seguridad, incluyendo la restricción del acceso a la red interna, realiza la traducción de direcciones, satisfaciendo los requisitos de autenticación. La mayoría de los firewalls comerciales también optimizan al núcleo del sistema operativo al despojar a los servicios innecesarios o peligrosos, proporcionando seguridad adicional para el servidor VPN. La desventaja de este tipo de tecnología es poder optimizar su desempeño de manera eficiente sin mermar las aplicaciones del sistema operativo.

3.1.5 VPN basada en software

Estas VPNs son ideales en casos donde ambos extremos de la VPN no están controlados por la misma organización o cuando diferentes firewalls y enrutadores se implementan dentro de la misma. Por el momento, las VPNs independientes ofrecen mayor flexibilidad en cómo se gestiona el tráfico de red. Muchos productos basados en software permiten que el tráfico de túnel se dependa de la dirección o protocolo, a diferencia de los productos basados en hardware, que en general encapsulen el tráfico que manejan, independientemente del protocolo.

Pero el software de los sistemas en que están basados generalmente son más difíciles de manejar que el cifrado de los enrutadores. Ellos requieren familiaridad con el sistema operativo del Host, la propia solicitud, y los mecanismos de seguridad adecuados. Y algunos paquetes de software de VPN requieren cambios en las tablas de enrutamiento y sistemas de direccionamiento de red.

Las VPNs también pueden clasificarse de acuerdo a criterios de función:

Por su punto de terminación, pueden estar basadas en las CE (overlay) o en el PE (peer-to-peer); por el tipo de tráfico de cliente transportado (nivel 2 y nivel3 del modelo OSI); por el tipo de red del proveedor (IP, IP/MPLS, ATM, Frame Relay, SONET/SDH, pppoe, etc); por tecnología de túnel (IPSec, L2TP, PPTP, MPLS-LPS, ATM-VP/VC, Frame Relay VC, SONET/SDH VT, PPP/Dial-up), y por el número de nodos conectados en multipunto y punto a punto.

3.2 Arquitecturas de las VPNs

Dentro de las posibles arquitecturas que encontramos en las VPN se pueden mencionar las siguientes:

- *Proporcionada por un servidor de Internet*: El proveedor de Internet puede instalar en su oficina un dispositivo que se encargará de la creación del túnel para la organización.

- *Basadas en firewalls*: De la misma forma en que las VPN trabajan en los niveles más bajos del modelo OSI, el firewall actuará de la misma forma.

- *Basadas en Caja Negra*: Básicamente es un dispositivo con software de cifrado. No provee seguridad en la organización pero si en los datos. Para suplir esta carencia se pueden utilizar un firewall en serie o paralelo al dispositivo de VPN.

- *Basadas en Enrutadores*: Puede ser en este caso que el software de cifrado se añada al enrutador ya existente o bien que se utilice una salida exclusiva de otro proveedor.

- *Basadas en acceso remoto*: El cliente tiene software por el cual se conecta al servidor de VPN de la corporación a través de un túnel cifrado.

- *Basadas en software*: Por lo general se utiliza de un cliente a un servidor de VPN que está instalado en alguna estación de trabajo. Es necesario tener procesos de administración de claves y un emisor de certificados.

3.3 Protocolos

Algunos de las tecnologías y protocolos usados para habilitar las VPNs sitio-a-sitio incluyen **IPSec**, **GRE** (Generic *Routing Encapsulating*), **L2TPv3** (Layer *Two Tunneling Protocol version 3*), **Draft Martini pseudowires** (circuitos emulados), **IEEE 802.1Q tunneling** (*Q-en-Q*) y **MPLS** (Multiple *Label Switched Paths*). A continuación se describen estos protocolos y tecnologías:

IPSec: consiste en un conjunto de protocolos diseñados para proteger el tráfico del IP entre puertas de enlace seguras. Mientras este transita entre redes intermedias.

GRE: puede ser usado para construir túneles y transportar tráfico multiprotocolo entre dispositivos CE en una VPN. *GRE* tiene una pequeña o ninguna seguridad, pero los túneles GRE pueden ser protegidos usando IPSec.

Draft Martini (cualquier transporte sobre **MPLS** [**AToM**]): el trasporte de datos tipo Draft Martini habilita un trasporte de datos del tipo punto-a-punto de protocolos del tipo Frame Relay, ATM, Ethernet , Ethernet VLAN (802.1 Q), HDLC (High-Level Data Link Control) y tráfico PPP sobre MPLS.

L2TPv3: permite el transporte punto-a-punto de protocolos tales como Frame Relay, ATM, Ethernet, Ethernet VLAN, HDLC, y tráfico PPP sobre IP.

MPLS LSPs: Una LSP es una ruta a través de una **LSR** (*Label Switch Routers*) en una red MPLS. Los paquetes son entregados en base a etiquetas agregadas al paquete. LSP puede ser señalizado usando **TDP** (*Tag Distribution Protocol*), **LDP** (*Label Distribution Protocol*), o **RSVP** (*Resource Reservation Protocol*).

También se requieren otros protocolos y tecnologías para permitir el acceso remoto, tales como:

L2F (*Layer Two Forwarding*) : L2F es un protocolo propietario de Cisco que fue diseñado para permitir encapsulamiento de tramas **PPP** (o **SLIP** [*Serial Line Interface Protocol*]) entre un sistema ***NAS*** y un dispositivo de puerta de enlace VPN ubicado en un sitio central. Los usuarios de acceso remoto conectados a un sistema ***NAS***, y las tramas PPP de los usuarios de acceso remoto son entonces encapsulados sobre la red hacia la puerta de enlace VPN de origen y destino.

PPTP (*Point-to-Point Tunneling Protocol*): PPTP es un protocolo que fue desarrollado por un grupo de empresas, incluyendo Microsoft, 3Com, y Ascend Communications. Como *L2F*, *PPTP* permite el encapsulamiento de tramas PPP de clientes de acceso remoto entre sistemas ***NAS*** y una ***VPN*** *gateway*. Los paquetes encapsulados PPP llevados sobre túneles PPTP son usualmente protegidos usando ***MPPE*** (*Microsoft Point-to-Point Encryption*).

L2TPv2/L2TPv3 (*Layer 2 Tunneling Protocol versions 2 and 3*): L2TP es una norma de la **IETF** (*Internet Engineering Task Force*) que combina las mejores cualidades de *L2F* y *PPTP*. En un ambiente de acceso remoto, *L2TP* permite tanto encapsulamiento de las tramas PPP de los clientes de acceso remoto a través de sistemas ***NAS*** a una puerta de enlace *VPN* como encapsulamiento de tramas *PPP* directamente desde el cliente de acceso remoto al concentrador/puerta de enlace *VPN*. *L2TP* tiene una seguridad intrínseca limitada por lo cual los túneles *L2TP* son usualmente protegidos con IPSec.

IPSec: Así como se habilitan VPNs sitio-a-sitio, IPSec también puede ser usado para asegurar tráfico de datos a través de túneles entre usuarios tanto de acceso remoto como usuarios móviles y un concentrador o puerta de enlace VPN.

SSL (*Secure Sockets Layer*): es un protocolo de seguridad que originalmente fue desarrollado por *Netscape Communications* (SSL versiones 1, 2, y 3), y provee de acceso remoto seguro para usuarios móviles y usuarios. Puede estar limitado funcionalmente (comparado con *L2F, PPTP, L2TPv2,* o *IPSec*) si son desplegadas VPNs *clientless* con SSL de acceso remoto.

TLS (*Transport Layer Security*), que es un estándar IETF muy similar a *SSLv3*.

Una ventaja es que no se requiere ningún tipo de software adicional porque SSL es incluido en cualquier navegador Web.

3.4 Requerimientos

Una VPN es una versión modificada de una red privada que permite incrementar la tradicional red de área local o la configuración de la Intranet a lo largo de la Internet y otras redes públicas. Para comunicarse de una manera económica y segura.

Como resultado, muchos de los requerimientos de una VPN y las redes privadas tradicionales son esencialmente los mismos. Los siguientes son requerimientos específicos de las VPNs:

- Seguridad
- Disponibilidad
- Calidad de Servicio
- Confiabilidad
- Compatibilidad
- Manejabilidad

A) Seguridad

Las redes privadas y las Intranets ofrecen un entorno altamente seguro porque los recursos de la red no están accesibles al público en general. Por lo tanto, la probabilidad de accesos desautorizados a sus recursos es altamente improbable. Pero esta certeza no es totalmente cierta para las VPNs, ya que estas hacen uso de los recursos públicos de la Internet y de las redes compartidas. Por lo tanto la seguridad en las VPNs no deberá tomarse a la ligera y las medidas de protección deberán plantearse muy seriamente.

Los recursos y la información localizados en la red pueden asegurarse de las siguientes maneras mediante:

Implementación de mecanismos de defensa periféricos.- que permitan sólo tráfico autorizado de fuentes confiables al interior de la red y que bloqueen el demás tráfico. *Firewalls* y *NAT's* son ejemplos de mecanismos de defensa que son implementados en los puntos donde una red privada o Intranet es conectada a la red en general. Los *firewalls* no sólo analizan el tráfico entrante, sino también el saliente. Los NAT's impiden revelar la IP real de los recursos localizados dentro de la red. Con el resultado de que los atacantes no pueden focalizar un recurso en específico ni los datos ahí almacenados.

Implementación de autenticación de usuarios y paquetes.- sirve para establecer la identidad del usuario y determinar si él o ella serán autorizados a ingresar a los recursos accesibles de la VPN. El modelo AAA (*Authentication Authorization Accounting*) es un ejemplo de uno de esos sistemas de autenticación de usuarios. Primero se autentica al usuario en la red, después de que el usuario ha sido autenticado exitosamente, el usuario puede acceder sólo a esos recursos que ha sido autorizado a usar. Adicionalmente, una detallada bitácora de actividades de todos los usuarios de la red es mantenida, lo cual permite a los administradores de la red descubrir y seguir los accesos no autorizados.

Implementación de mecanismos de cifrado.- se utiliza para garantizar la autenticidad, integridad y confidencialidad de la información cuando la misma es transmitida a través de las redes no autorizadas. IPSec ha emergido como uno de los más poderosos mecanismos de cifrado de datos. Este no solo cifra la información transmitida usando el encabezado ESP, también permite la autenticación de cada usuario y de cada paquete.

B) Disponibilidad y Confiabilidad

La *disponibilidad* se refiere al tiempo total que el sistema está disponible, en las redes privadas y las Intranets el tiempo de producción es relativamente alto porque toda la infraestructura es particular y está en completo control de la organización. Las VPNs usan redes intermedias como la Internet, por lo que las redes basadas en VPNs son altamente dependientes de las mismas.

Es en este tipo de escenarios que el factor de disponibilidad es altamente dependiente del proveedor de servicio de Internet. Si alguna organización está buscando una alta disponibilidad, tiene que contactar un ISP que ofrezca una infraestructura de intercambio altamente recuperable que incluya:

Poderosas capacidades de enrutamiento.- útiles para realizar el re-enrutamiento de tráfico a través de una ruta alternativa en caso de que la ruta principal falle o esté congestionada. Para garantizar la máxima eficiencia, esta capacidad de enrutamiento deberá soportar opciones para designar rutas preferenciales cuando sean requeridas.

Redundancia en las líneas de acceso, la cuál puede ser utilizada para acomodar el incremento en la demanda de ancho de banda.

Infraestructura redundante completa con recuperación automática,

Ésta infraestructura no solo debe de incluir dispositivos de intercambio como servidores y dispositivos de almacenamiento y de acceso, sino también plantas de engría y sistemas de enfriamiento.

La *confiabilidad* es otro de los requerimientos importantes de las VPNs y está íntimamente ligado al factor de *disponibilidad*. La confianza en las transacciones de las VPNs asegura la entrega de la información en los puntos finales en todas las situaciones. Como casi todas las otras configuraciones de red, la *confiabilidad* en los entornos VPNs puede ser logrado al intercambiar los paquetes de información por distintas rutas, si el dispositivo o vínculo en la ruta pudiera fallar. Este proceso por completo es transparente para el usuario y puede lograrse implementando redundancia en los vínculos así como hardware dedicado.

C) Calidad de servicio

La calidad de servicio (QoS) es la capacidad de la red para responder a situaciones críticas asignando un alto porcentaje del ancho de banda y recursos a las aplicaciones sensibles a los retrasos y de misión crítica. Las aplicaciones, tales como transacciones financieras y procesamiento de peticiones, son más importantes desde el punto de vista financiero que las actividades del usuario que incluyen el navegar por la red. Similarmente, aplicaciones tales como las videoconferencias son extremadamente sensibles a los retardos y requieren el suficiente ancho de banda para evitar la pobre calidad en la transmisión y la desincronía.

La calidad de servicio está comprometida con dos dimensiones, *latencia* y *rendimiento.* La *latencia* es el retraso en una comunicación saliente y es extremadamente importante para aplicaciones de audio y vídeo. El rendimiento (*throughput*) se refiere a la disponibilidad del apropiado ancho de banda para todas las aplicaciones, especiales de misión crítica y de uso intensivo de ancho de banda.

Dependiendo del nivel de latencia y del rendimiento, la calidad de servicio puede ser definida en alguna de las siguientes tres categorías:

1) *Mejor esfuerzo en calidad de servicio*: este tipo de servicio, en el mejor de los casos, indica la ausencia de calidad de servicio, porque el proveedor de servicios no garantiza la ausencia de latencia y de rendimiento en ningún caso. Es por esto que es el servicio menos costoso y no debe de ser usado para tráfico sensible al retraso o de uso intensivo de conexión.

2) *Calidad de servicio relativa*: esta clase de servicio es capaz de priorizar el tráfico de información. Por esta razón, al menos el rendimiento se garantiza. De cualquier manera, esta garantía no es absoluta y depende de la carga en la red y el porcentaje del tráfico que se necesita priorizar en algún momento dado. Adicionalmente, esta clase de servicio no tiene priorización para minimizar la latencia. Esta clase de servicio es moderadamente costoso para aplicaciones de uso intensivo de ancho de banda.

3) *Calidad de servicio absoluta*: esta clase garantiza ambas propiedades, *latencia* y *rendimiento*. Por lo tanto es el tipo de servicio más costoso y que soporta uso intensivo de ancho de banda y aplicaciones sensibles al retardo.

D) Manejabilidad

El control completo de los recursos de la red y sus operaciones, junto con la administración adecuada, han sido temas muy importantes para todas las organizaciones que tienen redes por todo el mundo. En este escenario la mayoría de las organizaciones están conectadas a sus servicios mundiales por medio de los proveedores de servicios (*ISP's*) como resultado, el control de una Intranet en el punto final no es posible por la presencia de intermediarios.

Con la actual disposición de dispositivos y software de VPNs, ha sido posible eliminar los límites tradicionales en el manejo de recursos y la administración de la red privada así como la parte pública de la VPN en sus puntos finales.

Una organización puede ahora administrar, monitorear, probar y localizar fallas, y mantener su red con el paradigma tradicional. La organización tiene el completo control del acceso a la red, y puede monitorear en tiempo real el estado de la red, ajustar la configuración de la VPN, etcétera.

E) Compatibilidad

Como ya se ha mencionado las VPNs usan las redes públicas como una extensión de la propia infraestructura, y esas redes intermedias pueden estar basadas en IP o en otras tecnologías de redes, tales como FR (*Frame Relay*) y ATM (*Asynchronous Transfer Mode*). Como resultado las VPNs deberían de ser capaces de hacer uso de todos los tipos de protocolos y tecnologías.

Para garantizar la compatibilidad con la infraestructura basada en IP, los siguientes métodos pueden ser integrados a las VPNs:

Uso de puertas de enlace IP:

Las puertas de enlace IP convierten o traducen los protocolos no-IP en IP y viceversa. Estos dispositivos pueden ser dispositivos de red dedicados o pueden ser soluciones basadas en software. Como dispositivos de hardware, las puertas de enlace son implementadas en las orillas de la Intranet de la organización. Como soluciones de software, las puertas de enlace son instaladas en cada servidor y son usadas para convertir de protocolos no-IP a IP.

Uso de túneles:

Los túneles se basan en la técnica de encapsulamiento de paquetes no-IP o IP en paquetes IP para su transmisión a través de la infraestructura IP existente. En el punto final receptor, donde se reciben estos paquetes mandados por el túnel, se remueve el encabezado IP para recuperar la información original, que en la terminología de los túneles se refiere como la carga útil (*payload*).

En la imagen 1 se esquematiza la forma en que las redes privadas virtuales hacen uso de la infraestructura de la Internet, como media de transmisión a través de ella y conservando su característica principal de privacía en los puntos finales respectivos.

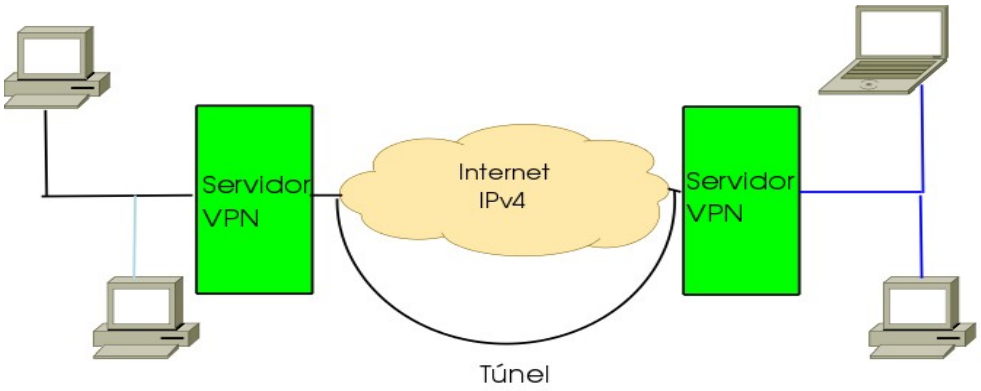


Imagen 1 “Túneles a través de la Internet”.

3.5 Conexiones

La conectividad que las VPNs puedan tener estará en función de qué tipo de políticas de seguridad se implementen y las respectivas herramientas que se utilicen para lograrlo. Será importante tomar en cuenta las ventajas y desventajas que cada opción pueda ofrecernos a fin de poder elegir aquella de acuerdo a nuestros requerimientos.

3.6 Seguridad en las VPN

Una VPN sin seguridad deja de ser privada, la cual es uno de los principales objetivos de las mismas. La seguridad en las TIs y en las VPNs se describe con tres aspectos:

1. *Privacidad (Confidencialidad):* los datos transmitidos sólo deberán estar disponibles para el receptor autorizado.

2. *Confiabilidad (Integridad)*: la información transmitida no deberá cambiar entre el receptor y el emisor.

3. *Disponibilidad*: la información trasferida deberá estar disponible cuando sea necesaria.

Todas estas metas deberán lograrse usando software confiable, hardware, IPS's y políticas de seguridad.

Una política de seguridad define las responsabilidades, procedimientos estandarizados, y los controles de daños además de los escenarios de recuperación que se prepararán para la peor situación posible.

Entendiendo que el daño máximo posible y el costo de la recuperación de la peor catástrofe posible pueden dar una idea de cuánto esfuerzo deberá gastarse en la seguridad.

La seguridad en las VPNs se logrará protegiendo el tráfico con modernos y fuertes métodos de cifrado, técnicas de autenticación segura y *firewalls* controlando el tráfico que se genera desde y hacia el túnel. Cifrar el tráfico no es suficiente, hay grandes diferencias en seguridad dependiendo del método que se implemente.

3.7 Criptografía

Usualmente se solía usar el cifrado de palabras claves o llaves como medio para garantizar el medio o cifrar datos. Si ambos lados usaban la misma llave para cifrar y descifrar la información se llama ***cifrado simétrico***. La llave de cifrado tiene que ser puesta en todas las máquinas que van a ser parte de la conexión VPN.

En el caso del cifrado simétrico y las llaves pre-establecidas, estas son estáticas por lo tanto pueden ser descifradas o adivinadas por ataques de fuerza bruta. Es sólo cuestión de tiempo para un atacante obtener la llave y leer, o en el peor de los casos escribir y destruir la información del sistema.

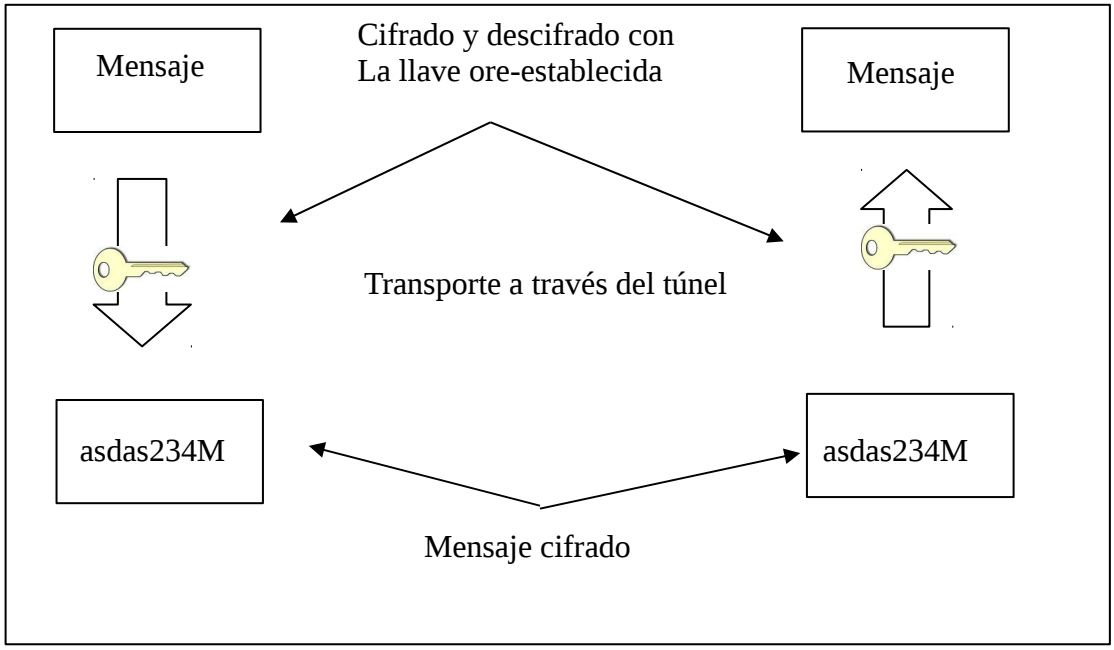


Imagen 2 “Gráfico del mecanismo de intercambio de llaves”.

En la imagen 2, esquematizamos un intercambio de llaves que se utilizan para conocer o sellar el contenido de un mensaje enviado/recibido para el caso del cifrado simétrico.
Así, protocolos como IPSec cambian las llaves en ciertos intervalos de tiempo, según se hayan configurado. Cada llave es válida sólo para cierto *periodo de tiempo*, llamado *tiempo de vida de la llave*. Una buena combinación del periodo de tiempo de la llave y la longitud de la misma, garantizarán que el atacante no pueda obtenerla cuando esta es aún válida.

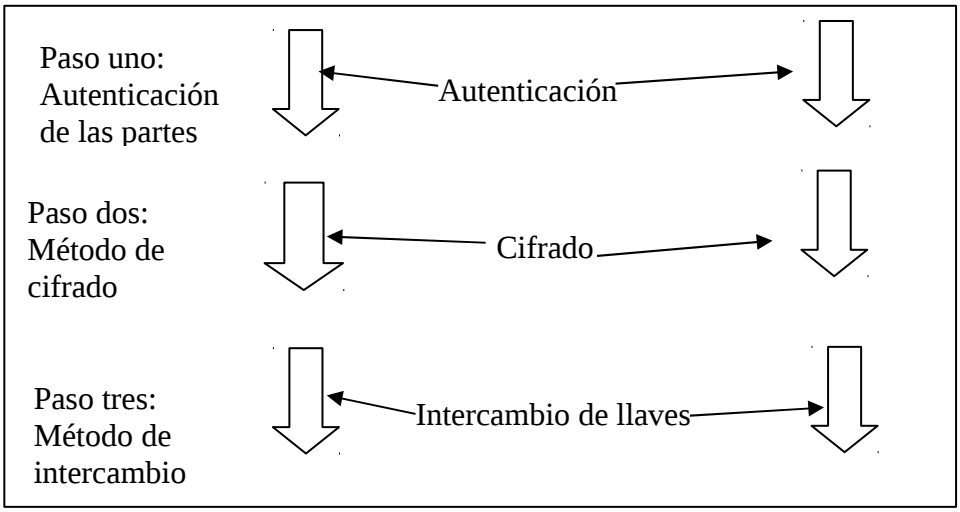


Imagen 3 “Pasos a seguir en un proceso de autenticación”.

En la imagen 3 vemos los pasos que se llevan a cabo durante la autenticación en un intercambio de cifrado simétrico.

3.8 Certificados y autenticación

Existen otros métodos para asegurar las comunicaciones entre los puntos involucrados, como es el uso de SSL/TSL. Estas capas usan el cifrado asimétrico, el cuál funciona de manera distinta que el cifrado simétrico. Para este caso, ambos puntos de comunicación tienen dos llaves cada uno: una pública y otra privada. La llave pública es la que se maneja sobre las comunicaciones, con la cual se cifra la información. Y sólo aquel que posea el otro par de las llaves, en este caso la llave privada, podrá descifrar los datos.

La autenticación de usuarios es un mecanismo implementado en las VPNs en el punto de acceso de las mismas, el cual es usado para garantizar que solo las personas que se autentican pueden acceder a la red y a sus recursos. Los esquemas que se pueden implementar individualmente o en combinación con otros incluyen los siguientes.

Identificación de usuario y clave de acceso (Login ID y password): este esquema usa la autenticación basada en la identificación del usuario y la clave de acceso basada en el sistema para verificar la identidad del usuario que accede al nodo VPN.

Clave de acceso secreta: en este esquema el usuario inicia la clave secreta seleccionando una palabra clave secreta y un número entero, n. Este número entero denota el número de veces que una función hash (actualmente MD4) será aplicada a la clave misma. El resultado es almacenado en el servidor correspondiente. Cuando los usuarios intenten acceder al sistema, el servidor llevará a cabo el procedimiento de autenticación. El software que el usuario usa para intentar la conexión solicitará la palabra clave, aplicará n-1 iteraciones de la función hash a la palabra clave, y se la enviará al servidor.

El servidor aplicará la función hash a esta respuesta, si el resultado obtenido es el mismo que el valor almacenado anteriormente, la autenticación fue exitosa. El usuario es entonces autorizado a ingresar al sistema.

RADIUS:

Es un protocolo de seguridad de Internet que está fuertemente basado en el modelo cliente/servidor, donde la máquina que ingresa a la red es el cliente y el servidor RADIUS, en el punto de acceso, autentica al cliente. Generalmente los servidores RADIUS autentican al usuario usando una lista de nombres de usuario y claves de acceso que mantienen internamente. RADIUS puede también actuar como un cliente para autenticar usuarios de los sistemas operativos, tales como UNIX, NT y NetWare. Adicionalmente, los servidores RADIUS pueden actuar como clientes para otros servidores RADIUS. Para asegurar aún más la información durante las transacciones entre los clientes y los servidores RADIUS esta puede ser cifrada usando mecanismos de autenticación, tales como el protocolo de autenticación de claves (*Password Authenticaction Protocol PAP*) y el protocolo de autenticación por aviso mutuo (*CHAP Challenge Handshake Authentication Protocol*)

Como este nombre lo sugiere, el esquema implementa la autenticación dual para verificar las credenciales del usuario. Combina el uso de un *token* y de una clave de acceso. Durante el proceso de autenticación, un dispositivo electrónico sirve como *token* y como identificador único, tales como el número personal de identificación (*PIN Personal Identification Number*) que es usado como la clave de acceso.

Control de acceso:

Después de que los usuarios se han autenticado exitosamente, estos ganan acceso a los recursos permitidos, servicios de red y aplicaciones localizadas en la misma. Esto puede ser un problema de seguridad porque el usuario, incluso el que ya esta autenticado, puede encontrarse con la información almacenada en varios dispositivos, sabiéndolo o no.

Los permisos de control de accesos son una parte integral del propio control. Los problemas de seguridad pueden ser manejados otorgando privilegios limitados a los usuarios. Por ejemplo, la información puede ser salvaguardada permitiendo a los usuarios no privilegiados sólo permisos de lectura de cierta información. Sólo los usuarios autorizados y el administrador deben de tener los privilegios para escribir, modificar o borrar información.

El control de accesos está basado en la identificación del usuario. Aunque otros parámetros, tales como la dirección IP de origen y la de destino, los puertos, y grupos, juegan un papel importante en el esquema tradicional de control de accesos.

Los mecanismos modernos y avanzados de control de accesos se basan en otros parámetros tales como el tiempo, día, aplicaciones, servicios, métodos de autenticación, URLs, y mecanismos de cifrado.

Cifrando Información

El cifrado de información o la criptografía es uno de los componentes más importantes de la seguridad de las VPNs y juega un papel primordial en la seguridad de la información durante su tránsito por las redes. Es el mecanismo de convertir la información a un formato ilegible, conocido como texto cifrado (*ciphertext*), así los intentos desautorizados de acceder a la información se pueden prevenir mientras la información es transmitida a través de un medio inseguro.

El cifrado de información previene algunas cuestiones como:

- Interceptación de la información y su lectura.
- Modificación de la información y su robo detectable.
- Fabricación de información.
- No-repudio de información.

Certificados Digitales

Un certificado digital es el equivalente electrónico de una identificación y es usado para identificar a una entidad única durante la transmisión. Además de establecer la identidad del dueño, los certificados digitales también eliminan las oportunidades de suplantaciones, reduciendo la oportunidad de la fabricación de información, y adicionalmente previenen efectivamente el rechazo de pertenencia de la información.

Un certificado digital consiste en información que ayuda a validar al emisor e incluye la siguiente información:

- El número de serie del certificado
- La fecha de finalización del certificado
- La firma digital del certificado de autorización (*CA*)
- La llave pública del propietario (*PKI*)

Durante la transacción, el emisor debe de enviar su certificado digital durante la transmisión con un mensaje cifrado para autenticarse a sí mismo. Como en el caso de las llaves públicas, la llave pública CA es ampliamente publicada y disponible a todo el mundo.

Sistema de distribución de certificados (*CDS Certificate Distribution System*)

El sistema de distribución de certificados es un repositorio para los usuarios y las organizaciones, adicionalmente un CDS genera y almacena pares de llaves, firma llaves públicas después de validarlas y almacena y remueve las llaves perdidas y caducas.

3.9 Aplicaciones específicas

El hardware VPN es básicamente para servidores VPN, clientes VPN y otros dispositivos de hardware, tales como enrutadores VPN y concentradores.

a) Servidores VPN

Generalmente, los servidores VPN son hardware dedicado corriendo software de servidores. Dependiendo de los requerimientos de la organización, puede haber uno o más servidores VPN. Como los servidores VPN deben de proveer servicio a los clientes remotos y locales, estos están siempre operativos y listos para las peticiones.

Las principales funciones de los servidores VPN incluyen las siguientes:

• Escuchar peticiones de conexión VPN.

• Negociar parámetros y requerimientos de conexión, tales como los mecanismos de cifrado y autenticación.

• Autenticación y autorización de clientes VPN.

• Aceptar información del cliente o la petición de reenvío de información del cliente.

• Actuar como el punto final del túnel VPN y la conexión. El otro punto de conexión se provee por las peticiones del usuario a la conexión VPN.

Los servidores VPN deben de poder soportar dos o más tarjetas de red (*NIC*). Una o más son usadas para conectarse con las organizaciones en la Intranet, mientras que las demás son usadas para conectarse a la Internet.

b) Clientes VPN

Los clientes VPN son máquinas locales o remotas que inicializan la conexión VPN a un servidor VPN y se introducen a la red remota después de haberse autenticado en el extremo de la misma. Después de un acceso exitoso pueden comunicarse mutuamente el servidor VPN y el cliente. Generalmente un cliente VPN es basado en software.

Aunque también puede ser un dispositivo de hardware dedicado. Un enrutador VPN basado en hardware con capacidades de conexión en demanda que se comunica con otro dispositivo VPN es un ejemplo de hardware dedicado. Con el incremento de una plantilla de trabajo móvil, muchos usuarios (clientes VPN) pueden tener perfiles de *roaming*. Estos usuarios pudieran haber usado una VPN para comunicarse con la Intranet del corporativo como por ejemplo:

•Usuarios móviles con laptops, palmtops, y notebooks los cuales usan redes públicas para conectarse con la Intranet de la organización accediendo a los correos y otros recursos de la Intranet.

• Administradores remotos los cuales usan las redes intermedias, tales como la Internet, para conectarse a una red remota para administrar, monitorear, diagnosticar, o configurar servicios y dispositivos.

c) Enrutadores VPN, Concentradores, y gateways

En el caso de la configuración de una VPN pequeña, el servidor VPN puede tomar una ruta para conectarse. Generalmente, un enrutador es el último extremo de una red privada a menos que esté detrás de un firewall. El papel de un enrutador VPN es hacer accesible las partes remotas de una Intranet. Por lo cual, los enrutadores son responsables de hallar posibles rutas hacia la red de destino y de escoger la ruta más corta del conjunto de rutas, como en el caso de las redes tradicionales.

Aunque los enrutadores tradicionales pueden ser usados en las VPNs, los expertos recomiendan usar enrutadores especialmente optimizados para las VPNs. Estos enrutadores, adicionalmente al enrutamiento, proveen seguridad, escalabilidad, y calidad de servicio (*QoS*) en la forma de redundancia en las rutas.

3.10 Administración

Para mantener a una VPN en óptimo estado de trabajo y con un rendimiento adecuado, deberán cuidarse algunos aspectos importantes. Debe recordarse que el desempeño de una VPN depende en gran medida del desempeño de los servidores VPN y de la infraestructura que se utilice. Se deberá revisar el desempeño de los servidores cuando menos una vez a la semana, para evitar cualquier imprevisto que baje el desempeño.

Es recomendable tener bitácoras detalladas de cada actividad relacionada con la VPN. Adicionalmente se deberán transferir dichas bitácoras a otra máquina para que en caso de que un intruso gane acceso este no pueda alterarlas.

Finalmente se deberá monitorear el desempeño de la red en general. Esto ayudará a identificar potenciales cuellos de botella y a identificar cuantos usuarios puede soportar su configuración de VPN antes de que los usuarios noten una degradación del rendimiento de la misma.

Los esquemas de cifrado, autenticación y de algoritmos pueden generar un considerable consumo de rendimiento, y se puede incrementar el propio de la red al analizar cuidadosamente las opciones a favor y en contra de los esquemas a utilizar, balanceando seguridad y rendimiento.

Los clientes son otro aspecto negativo de las VPNs. Será necesario controlar los clientes locales que estén en la intranet y avisar a los clientes remotos con perfiles de transferencia, de qué software para cliente VPN y qué sistema operativo usar para no afectar las transacciones a realizar.

3.11 Direccionamiento y enrutamiento

Otros puntos importantes en el diseño e implementación de las VPNs son el direccionamiento y el enrutamiento, esto es garantizar que las direcciones IP que se necesiten asignar a dispositivos VPN están bien planeadas y correctamente asignadas. -
También es necesario asegurarse que el esquema de enrutamiento no sólo tendrá conectividad IP con la dirección asignada, sino que será capaz de adaptarse a cambios en el esquema de direccionamiento futuro. Además deberán ser tomadas las medidas adecuadas para garantizar que colegas de negocios externos y clientes remotos son capaces de conectarse a la VPN sin problemas. Algunos de los temas comunes son:

1. Si se usan líneas dedicadas para conectarse a un ISP, en los dispositivos VPN deberían tenerse direcciones estáticas. Pero del otro lado, si usa una conexión telefónica para conectarse a un proveedor de conexión, deberían usarse direcciones dinámicas, especialmente clientes que usen dispositivos móviles o viejos teleconmutadores. El problema de usar direcciones dinámicas es que se generan problemas de seguridad, ya que un atacante puede pasar por un usuario plenamente autorizado.

No importa si usa un servidor VPN o múltiples servidores, *todos* deberán tener direcciones IP estáticas. Si se usan direcciones dinámicas con los servidores, los clientes no podrán localizar el servidor incluso localmente.

Las colisiones de direcciones IP saltarán a la vista si se intenta mezclar dos redes privadas, como en el caso de la fusión de dos entidades. Cambiar el esquema de direccionamiento consume mucho tiempo generalmente en su lugar se deberá considerar usar un esquema de direccionamiento dinámico, como el que provee el protocolo de direccionamiento dinámico (a) *DHCP*.

Si no se tienen suficientes direcciones IP globales-únicas, el mejor esquema de aprovechamiento de la red corporativa será usar direcciones privadas en la red interna y usar *NAT* en la red externa para realizar la conexión global. Este esquema le ayudará a garantizar que no ocurrirán conflictos cuando se establezcan conexiones internas con el exterior.

3.12 Ventajas y desventajas

Las VPNs ofrecen muchos beneficios. En la siguiente lista se mencionan algunos de ellos:

- *Reducción de costos de implementación*: las VPNs son considerablemente menos costosas que las soluciones tradicionales, las cuales están basadas en líneas alquiladas, Frame Relay, ATM o ISDN. Esto es porque VPN elimina la necesidad de conexiones a larga distancia, remplazándola con conexiones a un portador (carrier) local o ISP.

- *Reducción de costos por administración y manejo*: al reducir los costos de comunicaciones a larga distancia, las VPNs también bajan los costos de las redes amplias (***WAN***) considerablemente. Además, una organización puede reducir los costos totales de la operación si sus dispositivos de red VPN son manejados por el ISP. La razón para esta afirmación es que la organización no necesitará contratar personal altamente entrenado y calificado para el mantenimiento de la VPN si ella misma la maneja.

- *Incrementa la conexión*: las VPNs emplean la estructura de la Internet para la interconectividad de partes remotas de redes distintas. Así como la Internet es mundialmente accesible, incluso las oficinas más lejanas, los usuarios móviles y los agentes viajeros podrán conectarse a la red interna (*Intranet*) corporativa.

- *Seguridad en las transacciones*: las VPNs usan las tecnologías de túneles para transmitir datos a través de las redes públicas 'inseguras'. Además las VPNs usan medidas de seguridad en extenso, tales como cifrados, autenticación y autorización para garantizar la seguridad, confiabilidad e integridad de los datos transmitidos. Como resultado una VPN ofrece un alto grado de seguridad en las transacciones.

Servicio	De acceso remoto	Punto a punto	Punto– multipunto
Provee protección entre el cliente y el gateway local	No	No disponible	No disponible
Provee protección entre los punto finales de la VPN	Sí	Sí	Sí
Provee protección entre el gateway remoto y el servidor remoto (a través del gateway)	No	No	No disponible
Transparente a los usuarios	Sí	No	No
Transparente a los usuarios del sistema	Sí	No	No
Transparente a los servidores	Sí	Sí	No

Tabla 3. “Comparativa de distintos tipos de VPN y algunas de sus características generales.”

- *Uso eficiente del ancho de banda*: En el caso de las conexiones a Internet basadas en líneas alquiladas, el ancho de banda es desperdiciado enteramente cuando no existe una conexión activa. Por otro lado, las VPNs crean túneles lógicos cuando son requeridas. Como resultado, el ancho de banda es usado únicamente cuando hay una conexión activa. Por lo tanto hay menos oportunidades de un desperdicio del ancho de banda.

- Alta escalabilidad: como las VPNs están basadas en las conexiones a Internet, permiten a la red interna (Intranet) corporativa evolucionar y crecer, cuando y como el negocio cambie, con el mínimo de equipo extra o expansiones. Esto hace de las redes internas, altamente escalables y adaptables para futuros crecimientos, sin poner demasiada tensión en la infraestructura de red de la organización. A pesar de las numerosas ventajas que ofrecen las VPNs, algunas desventajas están asociadas a su uso, lo que ha provocado escepticismo entre los usuarios para su adopción. Las desventajas incluyen algunas de las siguientes:

- *Alta dependencia de la conexión a Internet*: el desempeño de las redes virtuales privadas es altamente dependiente del desempeño de la Internet. En cambio, en las líneas de alquiler garantizan el ancho de banda que está especificado en un contrato entre el ISP y la organización.

• *Ausencia de soporte para protocolos legados*: las VPNs del presente están basadas enteramente en el IP. Sin embargo, muchas organizaciones continúan usando mainframes además de otros dispositivos y protocolos anteriores en sus operaciones diarias. Como resultado, las VPNs son incompatibles con dispositivos y protocolos previos. Este problema puede ser resuelto, ampliamente, con el uso de mecanismos de túneles. Pero empaquetar SNA y otros protocolos no-IP, puede disminuir considerablemente el desempeño de la red entera.

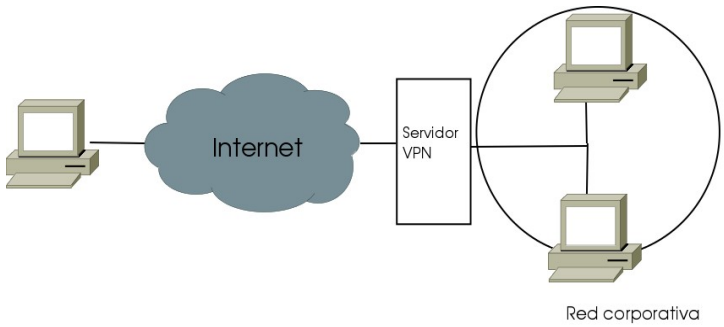


Imagen 4 “Operación de una VPN por un cliente remoto”

La imagen 4 muestra cómo un cliente remoro se conecta al servidor VPN de una red corporativa haciendo uso de la infraestructura de la Internet.

Capítulo 4

Implementación de OpenVPN

OpenVPN es un software de código abierto que ha desmostrado tener un robusto diseño y un desarrollo continuo tanto por el núcleo principal de desarrolladores como por la comunidad formada alrededor de esta solución. Es lo suficientemente flexible para que se adapte a las necesidades de desarrollo de este trabajo. El siguiente capítulo muestra el trabajo llevado a cabo, su descripción detallada y los resultados obtenidos de dicho esfuerzo.

Capítulo 4 Implementación de OpenVPN

4.1 Objetivos y metas con OpenVPN

El principal objetivo de este trabajo es lograr la difusión, entre la comunidad universitaria, tanto la academica-estudiantil como el público en general, del protocolo de nueva generación. Así como fomentar la adopción de los mecanismos de transición necesarios actualmente para llevar a cabo la transición entre la anterior y la siguiente versión del protocolo de internet. Esto ayudará a generar las buenas prácticas entre los encargados del mantenimiento y soporte a las redes así como el publico en general.

4.2 Descripción de funciones

La implementación de esta VPN, proveerá al público usuario de las siguientes funciones:

1. Autenticación y verificación de credenciales validadas por el servidor.

2. Administración sobre el periodo convenido sobre el servicio.(sujeto a las políticas de seguridad acordadas previamente)

3. Acceso a la red virtual privada.

4. Soporte para Ipv6.

5. Limitados privilegios al usuario.

4.3 Estructura de operación y control

Se contará con un equipo que ofrecerá el servicio de conexión a la vpn, el cuál realizará estas funciones de acuerdo a la demanda que se genere y a la disposición posible. Este se encontrará alojado en las instalaciones de la Dirección General de Cómputo Académico (DGSCA) de acuerdo a los lineamientos acordados de disponibilidad y requerimientos de la dirección de conformidad a su conveniencia.

La parte del control estará definida por las necesidades y disponibilidades del Laboratorio de Tecnologías Emergentes (NetLab), así como de los responsables a cargo del área de IPv6. De acuerdo a estas circunstancias será como se maneje y definan las políticas de acceso, disponibilidad así como también los mecanismos de control que sean necesarios para el buen manejo y las buenas prácticas que se pudieran desarrollar con dicho servicio.

4.4 Instalación de OpenVPN y soporte para IPv6

Para la instalación de OpenVPN se procedió a descargar el paquete que contenga el código fuente correspondiente a dicho software. En este caso al momento de documentar esta parte, se encontraba en su versión 2.1.1. En la página oficial del proyecto encontraremos la liga correspondiente a las descargas tanto del código fuente como de los binarios ejecutables para la plataforma Windows http://www.openvpn.net/index.php/open-source/downloads.html.

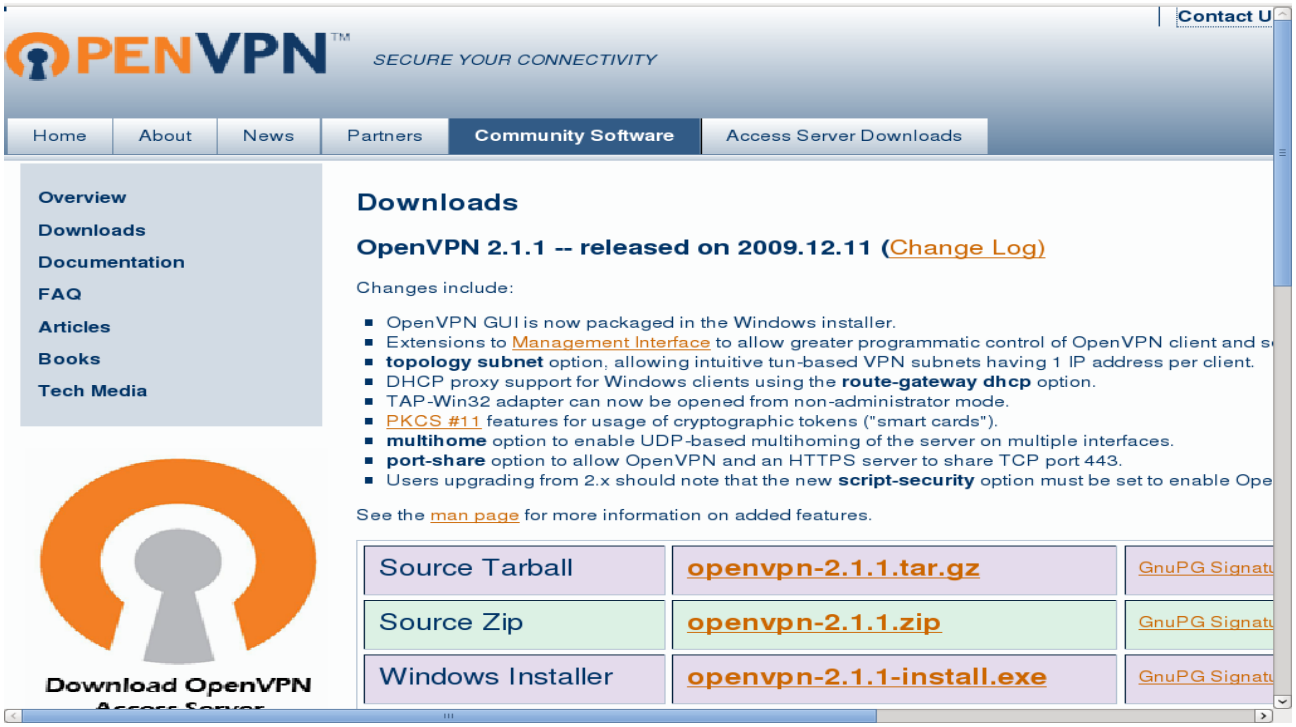


Imagen 5. "Sitio web oficial del proyecto OpenVPN"

Se procedió a descomprimir el paquete a fin de generar el directorio en donde se encontraban los distintos archivos del código fuente que eran necesarios para generar esta versión del software. El comando que se requiere para esto es el siguiente:

netlab@netlab:~/Downloads$ tar xvfz openvpn-2.1.1.tar.gz

de cuya respuesta obtenemos el resultado mostrado a continuación.

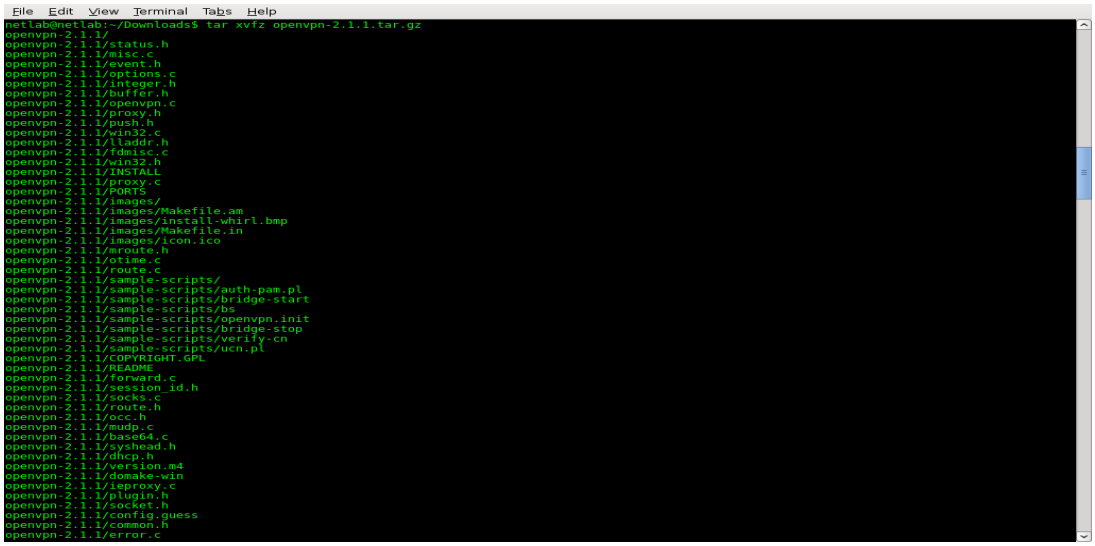


Imagen 6 “Captura de pantalla al descomprimir el paquete OpenVPN”

Esta acción nos creó automáticamente un directorio con todos los archivos necesarios para proceder a generar este software. En este directorio, que se llama openvpn-2.1.1, encontramos los siguientes archivos de configuración del código fuente.

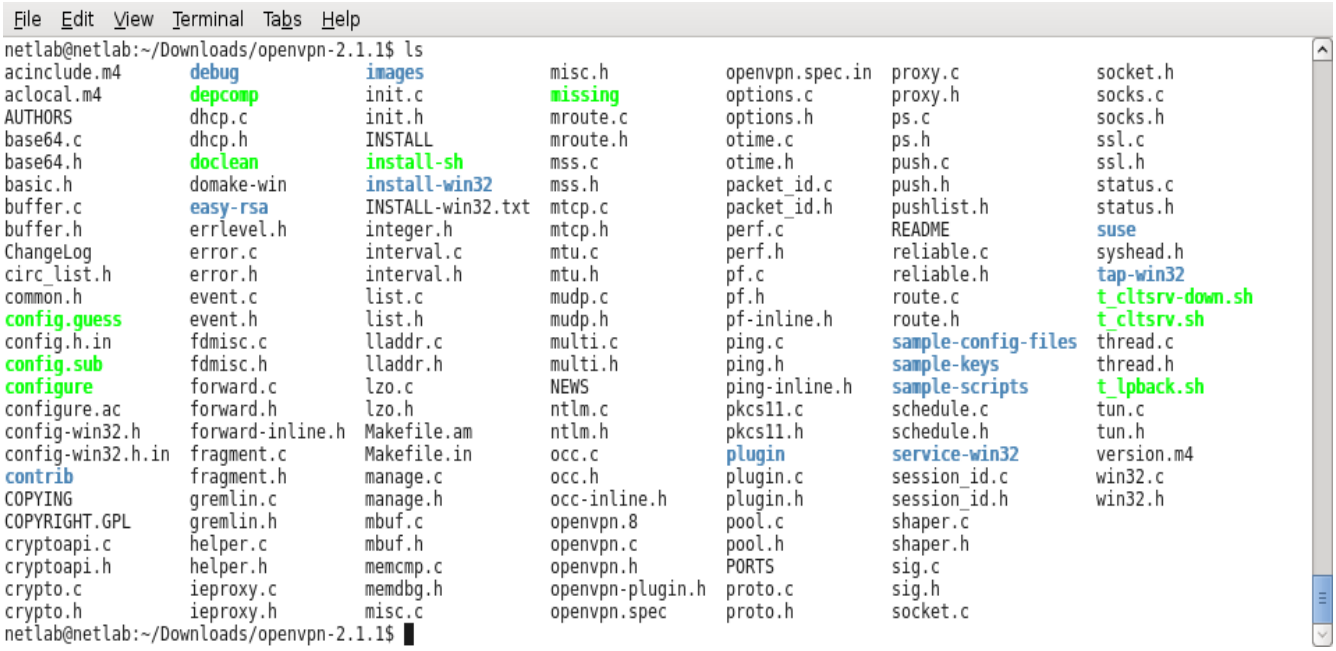


Imagen 7 “Listado de los paquetes del código fuente de OpenVPN”

Como se aprecia claramente, existen distintos tipos de archivos que corresponden a las diferentes funciones que se necesitan para crear el paquete. En este momento, se puede elegir crear el paquete tal cual se tiene, es decir seguir las instrucciones que se hallarán comúnmente en el archivo llamado INSTALL. Las cuales son:

- escribir en la línea de comandos *./configure* lo que debe entenderse como punto diagonal y teclear configure. Seguido de presionar el botón de Enter.
- Después teclear *make*. Seguido de presionar el botón Enter.
- Finalmente tecleamos *make install* . Y también presionamos el botón Enter.

Esto nos llevaría a obtener el paquete openvpn-2.1.1 tal cual se ofrece en la página web oficial del proyecto, pero para nuestros propósitos es necesario cursar otras vías. Una de ellas es la utilización de parches de software que los desarrolladores ofrecen a fin de mejorar o cambiar el comportamiento del software. Uno de ellos es ofrecido por Juan Jóse Ciarlante, el cual es posible descargarlo de la siguiente dirección http://github.com/jjo/openvpn-ipv6/downloads y otro se ofrece por parte de Bernard Schmidt y Gert Döring en http://www.greenie.net/ipv6. Obtendremos el siguiente paquete openvpn-2.1.1-20100307.tar.gz, el cual es el parche que requerimos aplicar al código fuente original para poder agregarle capacidades IPv6 al software principal. Lo hacemos con la siguiente instrucción :

patch <openvpn-2.1.1-20100307.patch

Esta acción nos dará como resultado tener un código que ya tiene algunas de las características que necesitamos. Modifica una serie de archivos para poder agregar las capacidades adicionales en el código fuente del software que se generó.

4.5 Pruebas de Interoperabilidad

Las pruebas que se llevaron a cabo exitosamente incluyen a los sistemas operativos tipo Unix. Estos son Debian 5 Lenny, Fedora 11 Leonidas y FreeBSD 8.0. En todos los casos estos funcionaron como servidores y clientes para ambas versiones del protocolo de internet, los sistemas Windows en sus versiones XP Professional Edition y Vista Home Edition fueron incapaces de lograr la conexión como cliente o servidor para IPv6, el soporte para IPv4 está garantizado. La Tabla 4 tiene el resumen de los resultados de las pruebas de conectividad para distintos sistemas operativos usados como clientes o servidores respectivamente.

Conectividad IPv6 VPN						
Cliente / Servidor	Windows Vista SP1	Windows XP SP3	FreeBSD 8	OpenBSD 4.6	Fedora 11	Debian 5
FreeBSD 8	Sin soporte	Sin soporte	----------	No funcionó	Sí	Sí
OpenBSD 4.6	Sin soporte	Sin soporte	No funcionó	----------	No funcionó	No funcionó
Fedora 11 Leonidas	Sin soporte	Sin soporte	Sí	No funcionó	----------	sí
Debian 5 Lenny	Sin soporte	Sin soporte	Sí	No funcionó	Sí	----------

Tabla 4.“Resumen de los resultados de la conectividad con IPv6 en la VPN. Modelo cliente-servidor punto a multipunto.”

4.6 Administración de fallas y cambios

A lo largo del desarrollo de este trabajo, se probaron distintos caminos entre los cuales destacan dos, el uso de scripts en ambos puntos de la conexión y el uso de código parchado a fin de generar el resultado deseado.

Con el primer camino, el uso de scripts, el proceso de implementación de este servicio se volvía engorroso para el administrador y los usuarios. Ambos tenían que tener conocimientos muy claros de lo que querían lograr y cómo hacerlo. Cualquier configuración incorrecta en alguno de los dos extremos de la conexión, cliente o servidor, y esta fallaba.

Para el caso del uso de parches aplicados al código original del proyecto, tenemos que decir que fue la opción viable para un servicio cómo el que se plantea implementar. Gracias al trabajo de estos desarrolladores alrededor del mundo fue que se tuvo un producto adecuado y viable. La administración del servicio es clara y simple de parte del servidor. El administrador responsable y enterado de lo que esta haciendo y tiene en sus manos, sabrá manejarlo. El usuario no tiene que tener grandes conocimientos de sistemas para poder hacer uso del servicio.

Se tenía planteado ofrecer un servicio multiplataforma que abarcara a las plataformas más usadas. Pero no fue posible tenerlo para el sistema MS Windows. La documentación que hay para este sistema esta fragmentada y en algunos casos es obsoleta o contradictoria, cómo la que se encuentra en línea en la página del proyecto OpenVPN. Las instrucciones de compilación que se encuentran en la página en línea pertenecen a versiones anteriores y las instrucciones que se pueden leer en el código fuente del programa no son del todo precisas.

Hay que añadir la gran cantidad de software adicional que es necesario para la generación de los ejecutables del software y el controlador que es necesario a fin de que funcione el programa en este sistema operativo. Específicamente el uso de bibliotecas de programación propietarias, llamadas DDK, que son parte del paquete de desarrollo de controladores para Windows (a) WDK. La versión que se solicita ya no esta disponible para su descarga en la página de Microsoft, lo cuál fue un impedimento para conseguir ambas partes del software, únicamente se tiene los ejecutables para el IPv4 no para IPv6, y no se pudo generar el controlador que soporte el IPv6.

Ventajas y desventajas de la VPN

- *Administración sencilla, centralizada, robusta y flexible.*
- *Bajo costo de implementación y mantenimiento.*
- *Alta escalabilidad y adaptación.*
- *Modular por diseño de origen y seguridad opr demanda.*
- *Archivos de configuración bien diseñados y bitacoras explícitas.*

- *Un solo puerto es usado por la VPN.*
- *Transparente a los usuarios.*
- *Protege los puntos finales de conexión.*

- *Requiere alta especialización por parte de los administradores del servicio.*
- *No es transparente a los servidores, requiere tener acceso autorizado por el firewall de los respectivos puntos de conexión, tanto punto a punto como punto a multipunto*
- *Ámplios conocimientos de los campos de redes, seguridad, administración y configuración.*
- *Constante actualización del conocimiento del producto y sus escenarios.*
- *En el producto libre la documentación está fragmentada, desarticulada y desactualizada.*
- *Sin la correcta administración y las políticas necesarias puede ser un boquete de seguridad.*

4.7 Generación de scripts y manual de conectividad

Los scripts se encontrarán en un archivo llamado *server.conf* o *client.conf* dado el caso, esto es por convención y para facilitar su ubicación a la vez de dar a entender su función de una manera clara y sencilla. Se ubicará preferentemente en los archivos de configuración que maneja el sistema, esto variará dependiendo de la elección del sistema operativo que sea el servidor, pero usualmente será algo como */etc/openvpn.*

El contenido de dicho archivo contiene las instrucciones necesarias para que el servidor provea las funcionalidades necesarias. Como se observa a continuación, esta es la configuración más básica par lograrlo:

```
port 1194
proto tcp
dev tun0
ca ca.crt
cert server.crt
key server.key
dh dh1024.pem
server-ipv6 10.8.0.0 255.255.255.0 2001:1218:1:6:42f:dd3a:b9b7:c0e9/64
ifconfig-pool-persist ipp.txt
comp-lzo
user nobody
group nobody
max-clients 10
keepalive 10 120
persist-key
persist-tun
status status-servidor.log
log-append bitacora-del-servidor.log
verb 9
reneg-sec 60
```

Cuandro 7 “Contenido del archivo server.conf”

En esta configuración, que se muestra en el cuadro 6, estamos habilitando los parámetros necesarios para que nuestro servidor pueda ofrecer conexión. Cada una de las opciones que se tienen se explican a continuación:

port 1194: nos habilita el puerto 1194, mismo que fue definido por la IANA como el estandar para esta software. Es posible utilizar cualquier otro, de acuerdo a las necesidades del administrador o las políticas que se hayan definido previamente.

proto tcp: habilita el protocolo a usar, que en este caso es tcp, pero existe otra opción que es udp. Dependerá de los requerimientos posteriores del servicio, el usar uno u otro.

dev tunX : define la interfaz a usar como medio de conexión, en este caso corresponde a la interfaz tun. También se puede usar la interfaz *tapX*. Los requerimientos del servicio determinarán cuál de estas es la más adecuada para esta implementación.

ca ca.crt: solicita el certificado de autenticación que el servidor manejará para firmar los certificados de los clientes a fin de llevar un control de los usuarios que hagan uso del servidor.

cert server.crt : maneja el certificado propio del servidor, al hacer el manejo de autenticación este es necesario para diferenciarlo de los clientes.

key server.key: controla la llave privada del servidor. A través del manejo de las llaves es cómo se autentica a los clientes del servidor.

dh dh1024.pem: define cual será el parámetro de seguridad a manejar. La cantidad de bits que manejará el algoritmo diffie-hellman para verificar la autenticidad de los certificados y las llaves.

server-ipv6: esta sentencia indica claramente que se trata del servidor y que además estamos manejando la parte IPv6. Se tienen que definir ambas direcciones IPv4 e IPv6, con sus respectivas máscaras de red.

Ifconfig-pool-persist: esta instrucción maneja el archivo de texto ipp.txt, en el cuál se almacenarán las direcciones asignadas a los clientes que manden peticiones de conexión al servidor.

comp-lzo: define la compresión que se utilizará en la conexión entre el cliente y el servidor.

u*ser nobody*: en particular esta opción sólo aplica a los sistemas tipo Unix verdaderos. Disminuye los privilegios del servicio ya que no permite usuarios en el grupo de usuarios de este servicio.

group nobody: se usa junto con la anterior instrucción para administrar de manera más segura el servicio.

max clients: limita la cantidad de clientes que serán usuarios del servicio.

keepalive: es una directiva auxiliar para simplificar la expresión de –**ping** y –**ping-restart** en las configuraciones de modo servidor.

persist-tun: evita cerrar/reabrir el dispositivo TUN/TAP o ejecutar/cerrar scripts a través de **SIGUSR1** o reinicios de –**ping-restart**.

persist-key: no relee archivos de llaves a través de –**ping-restart**.

log-append: crea la bitácora con los mensajes de conexión, intercambio de llaves, autenticación, etc. Muy útil si se necesita depurar o encontrar errores.

status: genera un archivo con el estado de la conexión.

verb: indica la generación de mensajes explícitos acerca de los estados de los procesos involucrados en la conexión.

reneg-sec: fija el tiempo de renegociación durante la conexión, según las necesidades. El default es 60 segundos.

El contenido de dicho archivo para los clientes contiene las instrucciones necesarias para que el servidor provea las funcionalidades necesarias. Como se observa a continuación, esta es la configuración más básica par lograrlo:

```
port 1194
proto tcp
dev tun0
ca ca.crt
cert client.crt
key client.key
comp-lzo
keepalive 10 120
persist-key
persist-tun
status status-client.log
log-append bitacora-del-cliente.log
verb 9
reneg-sec 60
```

Cuadro 8 “Cuadro del archivo client.conf”

En esta configuración, que se muestra en el cuadro 7, estamos habilitando los parámetros necesarios para que el cliente pueda obtener conexión del servidor VPN. Cada una de las opciones que se tienen se explican a continuación:

port 1194: nos habilita el puerto 1194, mismo que fue definido por la IANA como el estándar para este software. Es posible utilizar cualquier otro, de acuerdo a las necesidades del administrador o las políticas que se hayan definido previamente.

proto tcp: habilita el protocolo a usar, que en este caso es tcp, pero existe otra opción que es udp. Dependerá de los requerimientos posteriores del servicio, el usar uno u otro.

dev tunX : define la interfaz a usar como medio de conexión, en este caso corresponde a la interfaz tun. También se puede usar la interfaz *tapX*. Los requerimientos del servicio determinarán cuál de estas es la más adecuada para esta implementación.

ca ca.crt: solicita el certificado de autenticación que el servidor manejará para firmar los certificados de los clientes a fin de llevar un control de los usuarios que hagan uso del servidor.

cert client.crt : maneja el certificado propio del cliente, al hacer el manejo de autenticación este es necesario para diferenciarse del servidor y autenticarse ante el.

key client.key: controla la llave privada del cliente. A través del manejo de las llaves es cómo se autentica el cliente ante el servidor.

comp-lzo: define la compresión que se utilizará en la conexión entre el cliente y el servidor.

keepalive: es una directiva auxiliar para simplificar la expresión de –**ping** y –**ping-restart** en las configuraciones de modo servidor.

persist-tun: evita cerrar/reabrir el dispositivo TUN/TAP o ejecutar/cerrar scripts a través de **SIGUSR1** o reinicios de –**ping-restart**.

persist-key: no relee archivos de llaves a través de –**ping-restart**.

log-append : crea la bitácora con los mensajes de conexión, intercambio de llaves, autenticación, etc. Muy útil si se necesita depurar o encontrar errores.

status: genera un archivo con el estado de la conexión.

verb: indica la generación de mensajes explícitos acerca de los estados de los procesos involucrados en la conexión.

r*eneg-sec* : fija el tiempo de renegociación en el proceso de conexión, según las necesidades. El default es 60 segundos.

4.8 Realimentación a la propuesta

A fin de mejorar la propuesta se buscó cambiar su diseño de la implementación original. Al principio de la misma, se tenía planeado realizarla a través de una serie de scripts del lado del servidor y del cliente. A lo largo de los más dos años que este trabajo duró, aparecieron desarrolladores independientes del núcleo original de desarrollo del OpenVPN, que comenzaron a ofrecer sus parches al código fuente original a fin de que éste pudiera ofrecer características IPv6, que la comunidad de usuarios había estado solicitando. Con el uso de estos parches se logró el objetivo principal del planteamiento de la tesis, el cual es un servicio de una VPN con soporte para IPv6.

Conclusiones

Este trabajo sirvió para conocer las capacidades del software OpenVPN cuando este trabaja con soporte IPv6 mediante la realización de una serie de pruebas sin embargo, no fue posible profundizar sus posibilidades o limitaciones por falta de recursos para esto y falta de conocimiento en cuanto a su código. Se logró el conocimiento necesario y suficiente para lograr las comunicaciones en los sistemas tipo Unix, para el caso de sistemas tipo MS Windows se requiere software que no está a nuestro alcance, como bibliotecas de desarrollo de Windows, que son necesarias para poder crear ejecutables según las especificaciones de OpenVPN en sus últimas versiones.

También se pudieron probar ampliamente las capacidades de los distintos sistemas operativos que se utilizaron, tanto de las familias GNU/Linux y BSD's como las distintas versiones de MS Windows para evaluar la implementación de OpenVPN sobre cada uno de ellos. En el caso de los linuxes, esta plataforma demostró ser la más apta para OpenVPN al grado de ser la única que serviría completamente al propósito del presente trabajo.

En cuanto a los integrantes de la familia BSD, existen algunos problemas. Tanto para OpenBSD 4.X como para FreeBSD 6 y 7.X el sistema todavía adolece de una completa implementación del IPv6. Señalaremos particularmente que estos sistemas operativos no exhibe al mundo exterior una dirección global autoconfigurada IPv6, que si se verifica cuando se usan otros sistemas como GNU/Linux y MS Windows.

La verificación de la conectividad se realizó desde la conexión en IPv4 hasta la conexión en IPv6, verificando que esta fuera posible en los modos que son de la competencia del trabajo, modo cliente-cliente (o punto-a-punto) y cliente-servidor (o punto-a-multipunto). Y se verifico para las plataformas utilizadas para este propósito. Se puede afirmar que los sistemas probados tienen conectividad en IPv4 cuando el soporte para este es completo, mencionando que los sistemas windows no tienen soporte para la conexión cliente-cliente, tanto en IPv4 como en IPv6. Para el modelo cliente-servidor en IPv4, la versión de Windows Vista no soportó este tipo de conexión.

Para el caso del IPv6, cuando se utilizaron direcciones configuradas manualmente, sólo los sistemas tipo Unix pudieron completar la conexión con el otro extremo de la conexión. Las diferentes versiones del sistemas operativos Windows, tanto en sus versión XP Professional Edition como en Vista Home Edition, así como el sistema operativo OpenBSD, no contaron con soporte para esta plataforma por la ausencia de soporte en el mismo sistema operativo para las herramientas necesarias y la diferencia de implementaciones con respecto a los otros sistemas, a fin de permitir una manipulación de las interfaces lógicas del sistema con sus respectivas contrapartes de dispositivos físicos en ese sistema.

En las direcciones automáticas, las que se obtienen parchando el código, estas funcionaron nuevamente en los sistemas tipo Unix, con excepción del sistema OpenBSD y las versiones de Windows. Tanto para el modo p-a-p como p-a-mp.

Anexo

Anexo

Índice del Anexo

Índice de figuras y tablas del anexo

Índice de cuadros del anexo

Introducción

Las pruebas que se realizaron en el Laboratorio de Tecnologías Emergentes de Redes (NETLab) de la Universidad Nacional Autónoma de México (UNAM), están proyectadas para poder tener alternativas para la migración a IPv6 en la RedUNAM. Las VPNs han existido durante algún tiempo y también el IPv6, así como las redes privadas, virtuales o no. Como el IPv6 ha alcanzado un grado de madurez que las hace usables para los objetivos de NETLab, surge la necesidad de realizar pruebas con ambas tecnologías a fin de saber los alcances de las mismas. Se eligió el software OpenVPN por ser robusto, su tipo de licencia y capacidades de crecimiento. En virtud de su estado actual de desarrollo que es muy prometedor y estable.

Mencionaremos un problema que OpenVPN viene acarreando desde sus comienzos en las plataformas Windows y es la ausencia de soporte, tanto del software OpenVPN como del sistema operativo en cuestión, para la interfaz TUN. Esta interfaz es la que se maneja para la mayoría de los sistemas tipo Unix, pero cuyo soporte por parte de Microsoft es deficiente para su sistema operativo, y sin embargo existe la interfaz TAP. La diferencia entre este par de interfaces es que TUN esta diseñado como un dispositivo virtual para la conexión punto a punto, mientras que TAP esta diseñado como un dispositivo virtual para la conexión ethernet. La interfaz tun sólo soporta frames IP y tap sólo soporta frames ethernet. En la página en línea de OpenVPN, http://openvpn.net/index.php/open-source/documentation/install.html?start=1, refiere que la interfaz TAP cuenta con soporte limitado para la interfaz TUN. Pero esto no tuvo resultados satisfactorios cuando se usó con los sistemas Microsoft XP y Vista.

OpenVPN es una solución de conectividad en software, del tipo SSL VPN, para redes virtuales que tiene como cualidad ser escalable via web, es independiente del hardware. Soporta servicios VPN’s seguros y escalables a través de la Internet. Puede trabajar con las soluciones empresariales existentes y habilita la interacción en tiempo real de aplicaciones colaborativas. Es capaz de implementar modos básicos de conexión, en capa 2 o capa 3, con lo que se obtienen túneles capaces de enviar información en protocolos distintos al IP. Provee soporte para conexiones del tipo proxy. Funciona a través del proxy y puede ser configurado para ejecutar como un servicio TCP o UDP y además como servidor (simplemente esperando conexiones entrantes) o como cliente (iniciando conexiones). Solo un puerto en el firewall debe ser abierto para permitir conexiones, dado que desde OpenVPN 2.0 se permiten múltiples conexiones en el mismo puerto TCP o UDP. Las interfaces virtuales (tun0, tun1, etc.) permiten la implementación de reglas de firewall muy específicas. Alta flexibilidad y posibilidades de extensión mediante scripting. OpenVPN ofrece numerosos puntos para ejecutar scripts individuales durante su arranque. Soporte transparente para IPs dinámicas. Se elimina la necesidad de usar direcciones IP estáticas en ambos lados del túnel.

OpenVPN
No es un estándar y no es compatible con IPsec.
Disponible mayoritariamente como software, en casi todos los sistemas operativos existentes, y con algunas implementaciones en hardware.
Tecnología nueva y aun en crecimiento, suficientemente probada.
Sin interfaces gráficas profesionales, aunque ya existen algunos proyectos prometedores
Tecnología sencilla, no necesita modificaciones en la pila del protocolo IP
Usa las interfaces de red establecidas y paquetes estandarizados, no necesita modificar el kernel
Se ejecuta en el espacio del usuario y puede ser configurada para tener permisos elevados
Usa tecnologías de cifrado estandarizadas, no hay diferentes implementaciones o versiones entre varios proveedores.
Diseño fuertemente modular y facilidad de configuración, buena estructuración.
Curva de aprendizaje reducida, fácil de aprender y sencillo para principiantes
Solo Utiliza un puerto del firewall, que puede ser asignado de acuerdo a las necesidades. Por definición de la IANA es el puerto *1194*
Trabaja con servidores de nombres dinámicos como DynDNS o No-IP con reconexiones rápidas y transparentes.
Uso de SSL/TLS como estándar de criptografía.
Control de tráfico (Traffic shaping)
Velocidad (más de 20 Mbps en máquinas de 1Ghz), dependiendo de la conexión.
Alta compatibilidad con firewall y proxies
Ningún problema con NAT (ambos lados puede ser redes NATeadas)
Posibilidades para usuarios remotos (road warriors), dependiendo de las configuraciones y políticas.

Tabla 1 “Características generales de OpenVPN”

La tabla 1 nos muestra a detalle las principales características del OpenVPN. En cuanto a la técnicas conocidas cómo *bridging* y *routing* se pueden usar y de hecho se usan con OpenVPN, dependerá del caso que se esté manejando. El bridging es la técnica para crear una red de área amplia, virtual que se ejecute bajo una subred. Esencialmente es combinar una interfaz Ethernet (física) con una o más interfaces virtuales y ponerlas bajo la cobertura de una solo interfaz compartiendo una solo dirección IP. Las ventajas de usarlo son las siguientes:

4) Se anuncia a través de la VPN – esto permite al software que depende del broadcast como NetBIOS de windows que funciones el compartir archivos y la navegación entre vecinos .
5) No necesita configuraciones de router.
6) Trabaja con cualquier protocol que pueda funcionar sobre ethernet, incluyendo IPv4, IPv6, Netware IPX, AppleTalk, etc.

7) Solución fácil de configurar para los usuarios remotos (road warriors).

Una de sus desventajas sería:

4. Menor eficiencia que en el enrutamiento, y no se escala bien.

Las ventajas del enrutamiento son:

F) Eficiencia y alta escalabilidad.
G) Permite una mayor despersonalización del MTU para mayor eficiencia.

Y algunas desventajas:

- Los clients deben usar servidores WINS (tales como samba) para permitir la navegación a través de la VPN.
- Los Enrutadores deberán ser configurados para dar servicio a cada subred.
- El software que dependa del broadcasts no "podra" ver las máquinas del otro lado de la VPN.
- Trabaja con IPv4 en general, y IPv6 en casos donde las interfaces tun en ambos lados de la conexión lo soporten explicitamente.

Las versiones disponibles son desde la 1.6.0 hasta la versión actual, 2.1.1. Las versiones probadas fueron 2.0.6, 2.0.7, 2.1_rc1 hasta la versión 2.1_rc13, y finalmente la versión 2.1.1. El soporte para IPv6 se empezó a dar en la versión 2.0.x por parte del desarrollador argentino Juan José Ciarlante en el modo P2P, y para el modo servidor se dio en la versión 2.1.x por parte de los desarrolladores alemanes Gert Doering y Bernhard Schmidt.

Objetivo

Informar acerca de las pruebas que se han realizado sobre los escenarios de configuración básica de equipos, configuración de túneles, y funcionamiento.

Recursos

Hardware

- 4 PCs.
- 4 Tarjetas de red Ethernet.
- Cables para conexión RJ45.

Software

- Sistemas operativos tipo Linux [Fedora 11 Leónidas, Debian 5 Lenny].
- Sistemas operativos tipo BSD [OpenBSD 4.2 y 4.6, FreeBSD 7.1 y 8]
- Sistemas operativos tipo Windows [XP SP3 y Vista Home Edition]

Resumen de Pruebas

Las pruebas que se llevaron a cabo fueron:

- Verificación del soporte IPv6
- Configuración manual de túneles en equipos del mismo segmento.
- Configuración manual de túneles en equipos de diferentes segmentos.

Desarrollo y resultados

Verificación del soporte IPv6

Esta prueba consistió en verificar el soporte IPv6 en los sistemas operativos con que se cuenta.

Para verificar el soporte IPv6 en Windows XP se utilizó una ventana de comandos y se teclea el siguiente comando:

Ping ::1

Si el sistema operativo cuenta con soporte IPv6 se mostrara un resultado como el que se muestra:

```
Microsoft Windows [Versión 5.1.]
Copyright (c) 2006 Microsoft Corporation.  Reservados todos los derechos.
C:\> ping ::1
Haciendo ping a ::1 desde ::1 con 32 bytes de datos:
Respuesta desde ::1: tiempo<1m
Respuesta desde ::1: tiempo<1m
Estadísticas de ping para ::1:
```

```
    Paquetes: enviados = 2, recibidos = 2, perdidos = 0    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Cuadro 1. "Respuesta positiva del loopback para IPv6 en un sistema MS Windows"

De lo contrario:

```
C:\>ping ::1
La solicitud de ping no pudo encontrar el host ::1. Compruebe el nombre y vuelva a
intentarlo.
```

Cuadro 2. "Respuesta negativa del loopback para IPv6 en un sistema MS Windows"

FreeBSD y OpenBSD

Los sistemas *BSD en sus últimas versiones no tienen soporte IPv6 habilitado por defecto, este se tiene que habilitar en los respectivos archivos de configuración para poder dar de alta en el sistema este soporte.

En el caso de *OpenBSD* se tiene que editar el archivo *sysctl.conf*, donde hay que descomentar las líneas que dicen y asignar los valores pertinentes.

net.inet6.ip6.forwarding=0 # 1=Permit forwarding (routing) of IPv6 Packets
net.inet6.ip6.accept_rtadv=1 # 1=Permit IPv6 autoconf (forwarding must be 0)

A fin de que el sistema en cuestión pueda autoconfigurarse en el segmento en el que se encuentra. Cualquier otra configuración requerirá modificaciones adicionales y el uso del archivo de configuración *rc.conf* posiblemente.

Figura 1. "Archivo de configuración rc.conf de OpenBSD."

Para el caso de *FreeBSD* también se tiene que editar el archivo *rc.conf* ,cuyas variables por defecto se encuentran en */etc/default/rc.conf.* En el rc.conf por defecto encontraremos la sección llamada *### IPv6 options ###.* En esta parte vemos las variables que necesitamos copiar al */etc/rc.conf* para modificarlas y no alterar el archivo por defecto. Las variables son :

ipv6_enable="" # Set to YES to set up for IPv6
ipv6_network_interfaces="" # List of network interfaces (or "auto")

Estas son las variables que hay que modificar para que el sistema se autoconfigure.
Y para verificar que ya se tiene habilitado el IPv6 en una terminal se introduce el comando:

```
$ping6 ::1
PING6(56=40+8+8 bytes) ::1 --> ::1
16 bytes from ::1, icmp_seq=0 hlim=64 time=0.381 ms
16 bytes from ::1, icmp_seq=1 hlim=64 time=0.443 ms
16 bytes from ::1, icmp_seq=2 hlim=64 time=0.495 ms
--- ::1 ping6 statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 0.381/0.440/0.495/0.047 ms
```

Cuadro 3. "Respuesta positiva del loopback para IPv6 en un sistema FreeBSD"

Linux

En los sistemas Linux el soporte IPv6 esta habilitado por defecto, así que en una terminal se teclea:

```
$ping6 ::1
PING ::1(::1) 56 data bytes
64 bytes from ::1: icmp_seq=1 ttl=64 time=0.057 ms
64 bytes from ::1: icmp_seq=2 ttl=64 time=0.067 ms
64 bytes from ::1: icmp_seq=3 ttl=64 time=0.063 ms
--- ::1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.057/0.064/0.071/0.009 ms
```

Cuadro 4. "Respuesta positiva del loopback para IPv6 en un sistema GNU/Linux"

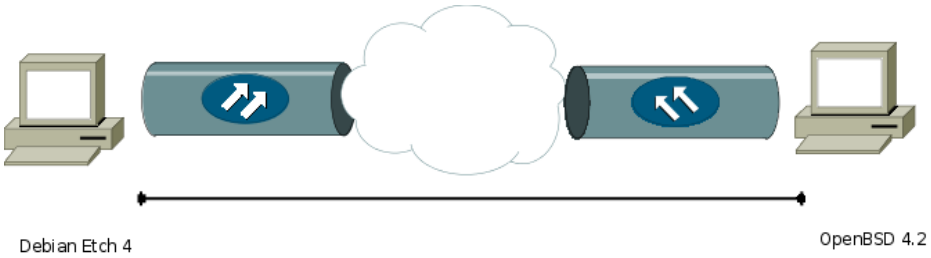


Figura 2. "Túnel manual entre dos equipos."

En esta parte se confirmo que todos los equipos y los sistemas operativos tuvieran conectividad y soporte para IPv6.

Configuración manual de túneles en equipos del mismo segmento, modelo cliente-cliente.

En equipos que cuenten con sistemas operativos GNU/Linux se pueden configurar túneles manuales de la siguiente manera:

GNU/Linux
En este sistema se creó una interfaz para el túnel y se configuró el final del mismo con los comandos:

```
# openvpn --remote 132.248.108.239 1194 --dev tun --ifconfig 10.4.0.2 10.4.0.1 --verb 9
```

Cuadro 5. "sentencia de inicio para openvpn en un sistema GNU/Linux en la línea de comandos"

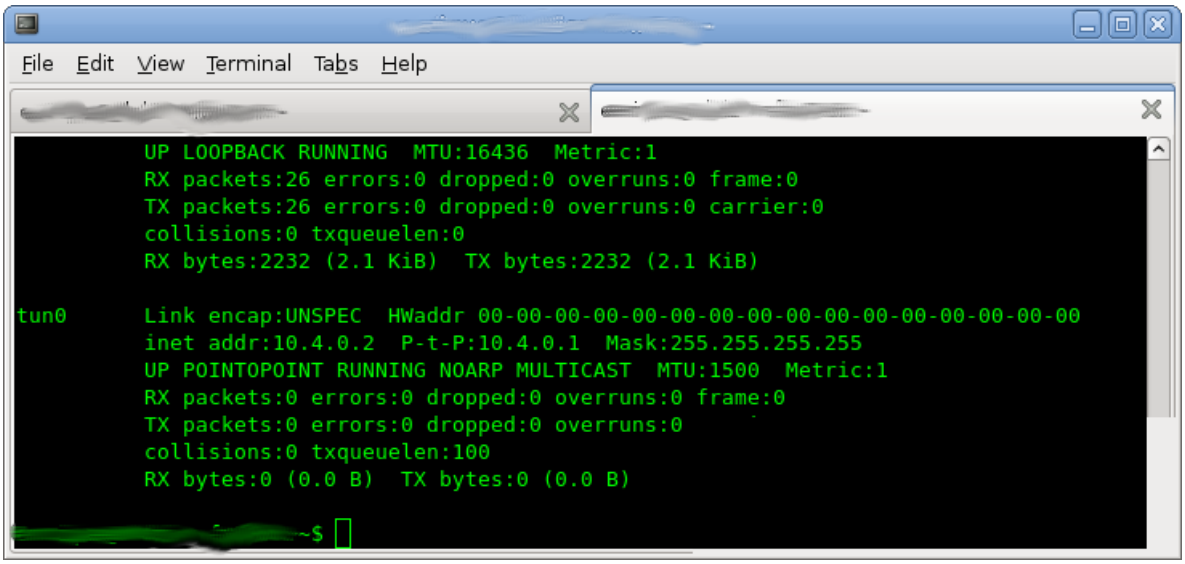


Figura 3. "Interfaz tun0 configurada ."

Para el sistema OpenBSD se creó una interfaz para el túnel y se configuró el final del mismo con los comandos:

```
# openvpn --remote 132.248.108.234 1194 --dev tun0 --ifconfig 10.4.0.1 10.4.0.2 --verb 9
```

Cuadro 6. "sentencia de inicio para OpenVPN en un sistema OpenBSD en la línea de comandos"

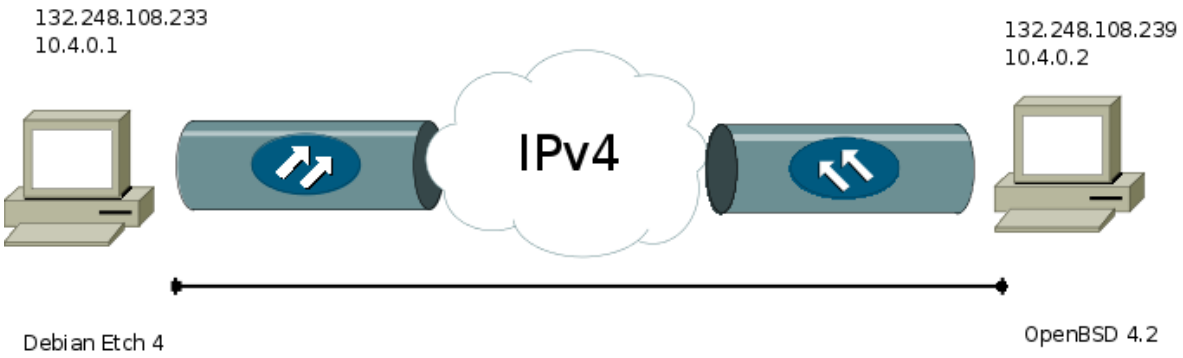


Figura 4." Túnel manual entre dos segmentos."

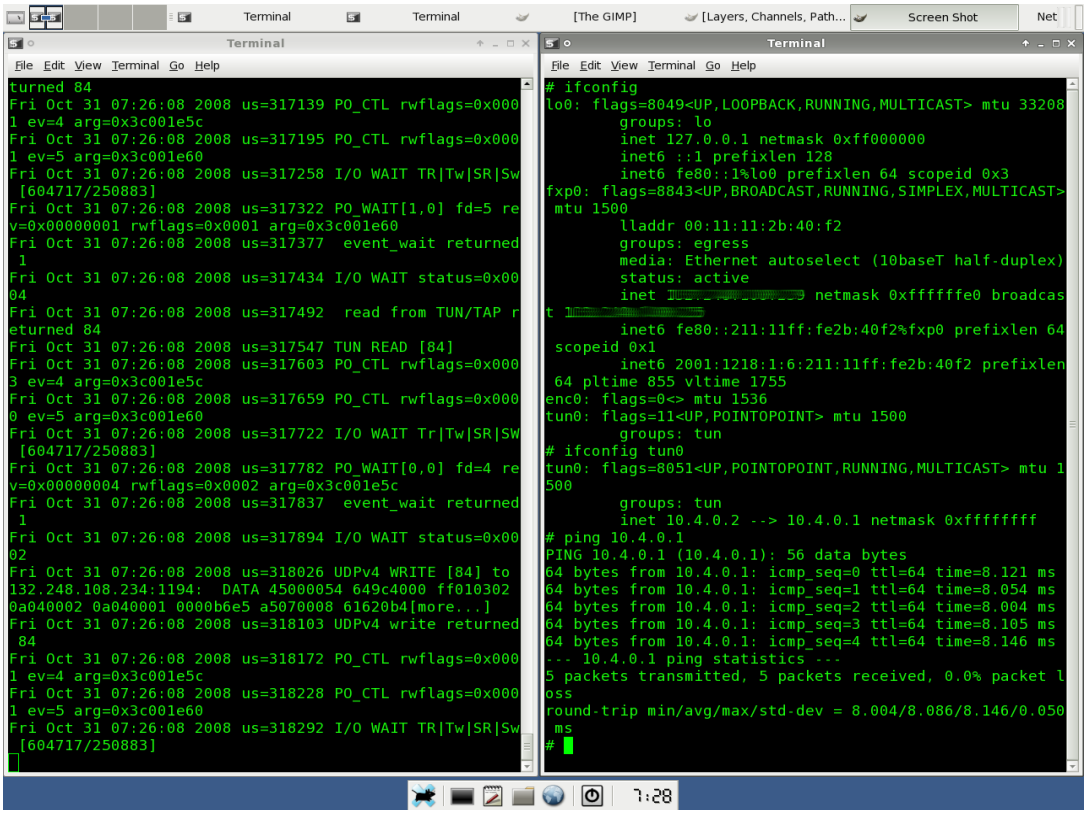


Figura 5. "Túnel manual entre OpenBSD y Debian."

Por último se realizaron pruebas de conectividad entre los hosts:

OpenBSD:

```
$ ping 10.4.0.1
64 bytes from 10.4.0.1: icmp_seq=1 ttl=64 time=0.104 ms
64 bytes from 10.4.0.1: icmp_seq=2 ttl=64 time=0.088 ms
64 bytes from 10.4.0.1: icmp_seq=3 ttl=64 time=0.087 ms
64 bytes from 10.4.0.1: icmp_seq=4 ttl=64 time=0.091 ms
--- 10.4.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.087/0.092/0.104/0.011 ms
```

Cuadro 7. "Respuesta positiva de la dirección IP del primer cliente desde un sistema OpenBSD"

Debian:

```
[netlab@Netlab-3 ~]$ ping 10.4.0.2
64 bytes from 132.248.59.73: icmp_seq=1 ttl=64 time=0.105 ms
64 bytes from 132.248.59.73: icmp_seq=2 ttl=64 time=0.087 ms
64 bytes from 132.248.59.73: icmp_seq=3 ttl=64 time=0.099 ms
64 bytes from 132.248.59.73: icmp_seq=4 ttl=64 time=0.096 ms
--- 10.4.0.2 ping statistics ---
4 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.087/0.098/0.105/0.009 ms
```

Cuadro 8. "Respuesta positiva de la dirección IP de segundo cliente desde un sistema GNU/Linux"

En esta prueba verificamos que existiera conectividad y respuesta de cliente a cliente para la VPN sobre IPv4.

Configuración de un túnel con seguridad de llave estática, modelo cliente-cliente.

Para esta prueba se utilizaron los mismos equipos, con las mismas configuraciones anteriores con el agregado de la generación de la llave estática la cual creamos con el siguiente comando :

openvpn --genkey --secret key

Este comando construyó un archivo llave aleatorio (en formato **ASCII**). Se copió la llave a **OpenBSD** con un medio seguro como el programa/comando scp(1)

Se procedió a ejecutar el siguiente conjunto de instrucciones para recrear el túnel y se configuró de la misma manera además de agregar el uso de la llave pública:

En GNU/Linux:

```
#  openvpn --remote 132.248.108.239 1194 --dev tun --ifconfig 10.4.0.1 10.4.0.2 --verb 9 --secret key
```

Cuadro 9. "sentencia de inicio para OpenVPN en un sistema GNU/Linux en la línea de comandos"

Para OpenBSD

Se creó una interfaz para el túnel y se configuró el final del mismo con los comandos, agregando el uso de la llave estática:

```
# openvpn --remote 132.248.108.234 1194 --dev tun0 --ifconfig 10.4.0.2 10.4.0.1 --verb 9 --secret key
```

Cuadro 10. "sentencia de inicio para OpenVPN en un sistema OpenBSD en la línea de comandos"

Se verificó que el túnel estuviera trabajando

En GNU/Linux

```
[netlab@Netlab-3 ~]$ ping 10.4.0.2
64 bytes from 132.248.59.73: icmp_seq=1 ttl=64 time=0.105 ms
64 bytes from 132.248.59.73: icmp_seq=2 ttl=64 time=0.087 ms
64 bytes from 132.248.59.73: icmp_seq=3 ttl=64 time=0.099 ms
64 bytes from 132.248.59.73: icmp_seq=4 ttl=64 time=0.096 ms
--- 10.4.0.2 ping statistics ---
4 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.087/0.098/0.105/0.009 ms
```

Cuadro 11. "Respuesta positiva de la dirección IP del cliente desde el servidor un sistema GNU/Linux"

Y en OpenBSD

```
$ ping 10.04.0.1
64 bytes from 10.4.0.1: icmp_seq=1 ttl=64 time=0.104 ms
64 bytes from 10.4.0.1: icmp_seq=2 ttl=64 time=0.088 ms
64 bytes from 10.4.0.1: icmp_seq=3 ttl=64 time=0.087 ms
64 bytes from 10.4.0.1: icmp_seq=4 ttl=64 time=0.091 ms
--- 10.4.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.087/0.092/0.104/0.011 ms
```

Cuadro 12. "Respuesta positiva de la dirección IP del servidor desde el cliente en un sistema OpenBSD"

En esta prueba verificamos que existiera conectividad y respuesta de cliente a cliente para la VPN sobre IPv4 con seguridad habilitada parcialmente.

Configuración de un túnel con seguridad TLS, modelo cliente-servidor**.**

Para esta prueba se utilizó el sistema GNU/Linux como servidor TLS y OpenBSD como el cliente. Debe mencionarse que la designación cliente-servidor solo tiene significado en el subsistema TLS, no tiene ninguna influencia en el modelo de comunicación punto a punto que utiliza OpenVPN.

En esta prueba se tuvieron que construir llaves por separado para el cliente y para el servidor, en nuestro caso GNU/Linux y OpenBSD. También se construyó un archivo de parámetros Diffie-Hellman.

Para OpenBSD se ejecutó:

```
# openvpn --remote 132.248.108.234 1194 --dev tun0 --ifconfig 10.4.0.2 10.4.0.1 --tls-client --ca
/usr/local/share/examples/openvpn/sample-keys/tmp-ca.crt –cert
/usr/local/share/examples/openvpn/sample-keys/client.crt --key
/usr/local/share/examples/openvpn/sample-keys/client.key --reneg-sec 60 --verb 9
```

Cuadro 13. "sentencia de inicio del cliente para OpenVPN en un sistema OpenBSD en la línea de comandos"

Para GNU/Linux se procede a ejecutar el siguiente conjunto de instrucciones :

```
# openvpn --remote 132.248.108.239 1194 --dev tun --ifconfig 10.4.0.1 10.4.0.2 --tls-server --dh
/usr/tools/openvpn-2.0.9/sample-keys/dh1024.pem --ca /usr/tools/openvpn-2.0.9/sample-keys/tmp-
ca.crt --cert /usr/tools/openvpn-2.0.9/sample-keys/server.crt --key /usr/tools/openvpn-
2.0.9/sample-keys/server.key --reneg-sec 60 --verb 9
```

Cuadro 14. "sentencia de inicio del servidor para OpenVPN en un sistema GNU/Linux en la línea de comandos"

Se verificó que el túnel estuviera trabajando en GNU/Linux

```
[netlab@Netlab-3 ~]$ ping 10.4.0.2
64 bytes from 132.248.59.73: icmp_seq=1 ttl=64 time=0.105 ms
64 bytes from 132.248.59.73: icmp_seq=2 ttl=64 time=0.087 ms
64 bytes from 132.248.59.73: icmp_seq=3 ttl=64 time=0.099 ms
64 bytes from 132.248.59.73: icmp_seq=4 ttl=64 time=0.096 ms
--- 10.4.0.2 ping statistics ---
4 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.087/0.098/0.105/0.009 ms
```

Cuadro 15. " Respuesta positiva a la IP del cliente desde el servidor en GNU/Linux "

Y en OpenBSD

```
$ ping 10.04.0.1
64 bytes from 10.4.0.1: icmp_seq=1 ttl=64 time=0.104 ms
64 bytes from 10.4.0.1: icmp_seq=2 ttl=64 time=0.088 ms
64 bytes from 10.4.0.1: icmp_seq=3 ttl=64 time=0.087 ms
64 bytes from 10.4.0.1: icmp_seq=4 ttl=64 time=0.091 ms
--- 10.4.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.087/0.092/0.104/0.011 ms
```

Cuadro 16. " Respuesta positiva a la IP del servidor desde el cliente en OpenBSD "

En esta prueba verificamos que existiera conectividad y respuesta de cliente a servidor y viceversa para la VPN sobre IPv4 con seguridad habilitada en el modelo cliente-servidor.

Configuración de un túnel con seguridad TLS, modelo cliente-servidor para el caso GNU/Linux-MS Windows.

Para poder realizar la comunicación entre cliente-servidor de un sistema MS Windows© se utilizó la siguiente configuración. Para el servidor GNU/Linux

```
$openvpn --port 1194 --proto tcp --dev tun --ca /usr/local/share/examples/openvpn/sample-keys/tmp-ca.crt --cert /usr/local/share/examples/openvpn/sample-keys/client.crt
--key /usr/local/share/examples/openvpn/sample-keys/client.key  --dh /usr/tools/openvpn-2.0.9/sample-keys/dh1024.pem --server 10.8.0.0 255.255.255.0 --ifconfig-pool-persist fipp.txt
--keepalive 10 120 –comp-lzo --user nobody --group nobody –persist-key –persist-tun --status openvpn-status.log --verb 9
```

Cuadro 17. "Sentencia de inicio del servidor para OpenVPN en un sistema GNU/Linux en la línea de comandos."

La cual es una variación de la solución propuesta en el foro comunitario de Gentoo (http://en.gentoo-wiki.com/wiki/HOWTO_OpenVPN_Linux_Server_Windows_Client)

Del lado del cliente MS Windows© se utilizó

```
C:\ARCHIV~1\OpenVPN\sample-config>openvpn client.ovpn
```

Cuadro 18. "inicio del cliente OpenVPN y su archivo de configuración desde el promt de MS Windows."

El cual contiene:

```
client
dev tun
proto tcp-client
remote 132.248.108.234 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca tmp-ca.crt
cert client.crt
key client.key
comp-lzo
verb 9
```

Cuadro 19. "Contenido del archivo de configuración para MS Windows XP "

Se verificó que estuviera trabajando el túnel:

En GNU/Linux

```
[netlab@Netlab-3 ~]$ ping 10.8.0.6
64 bytes from 132.248.108.234: icmp_seq=1 ttl=64 time=0.105 ms
64 bytes from 132.248.108.234: icmp_seq=2 ttl=64 time=0.087 ms
64 bytes from 132.248.108.234: icmp_seq=3 ttl=64 time=0.099 ms
64 bytes from 132.248.108.234: icmp_seq=4 ttl=64 time=0.096 ms
--- 10.8.0.6 ping statistics ---
4 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.087/0.098/0.105/0.009 ms
```

Cuadro 20. "Respuesta positiva de la dirección IP del cliente desde el servidor GNU/Linux "

Y en MS Windows©

```
$ ping 10.4.0.1
64 bytes from 132.248.108.239: icmp_seq=1 ttl=64 time=0.104 ms
64 bytes from 132.248.108.239: icmp_seq=2 ttl=64 time=0.088 ms
64 bytes from 132.248.108.239: icmp_seq=3 ttl=64 time=0.087 ms
64 bytes from 132.248.108.239: icmp_seq=4 ttl=64 time=0.091 ms
--- 10.8.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.087/0.092/0.104/0.011 ms
```

Cuadro 21. "Respuesta positiva de la dirección IP del servidor desde el cliente MS Windows"

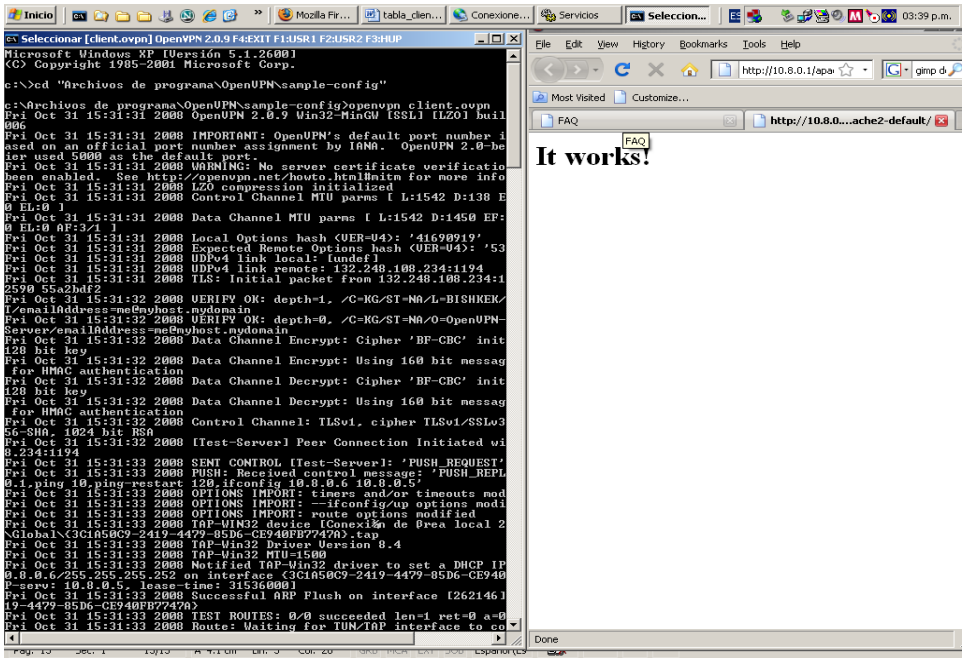


Figura 6. "Túnel entre un cliente MS Windows y un servidor Linux"

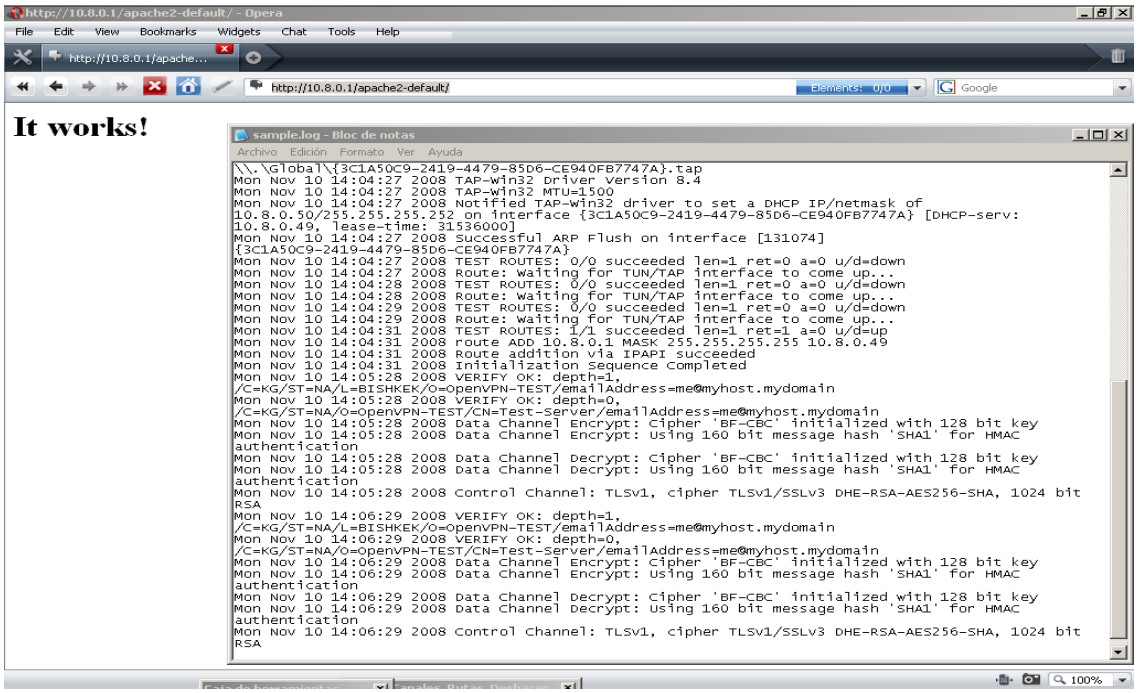


Figura 7. "Túnel entre un cliente MS Windows y servidor Linux bis."

En esta prueba verificamos que existiera conectividad y respuesta de cliente a servidor para la VPN sobre IPv4, con seguridad habilitada en el modelo cliente-servidor.

Configuración de un túnel con seguridad TLS, modelo cliente-servidor para el caso GNU/Linux-Ms Windows/OpenBSD.

Para esta configuración se utilizó la misma configuración que en el caso anterior para el sistema MS Windows© y la misma configuración para el sistema OpenBSD.

Ambos lograron la conexión.

En esta prueba verificamos que existiera conectividad y respuesta de cliente a servidor para la VPN sobre IPv4, con seguridad habilitada en el modelo cliente-servidor. Señalaremos que los sistemas que servían como clientes obtenían la misma dirección ip por causa de la configuración.

Generación de un ejecutable para el sistema MS Windows©

En esta etapa del proceso se tenía planeado generar un ejecutable nativo para todos los sistemas MS Windows© que se adecuara a nuestras necesidades, sin embargo sólo se pudo compilar y generar a partir del código fuente de la versión 2_rc7 del OpenVPN en donde todavía era posible compilarlo con herramientas propietarias de Microsoft como el Visual Studio express C, pero después de esa versión en el desarrollo de OpenVPN se decidió que cambiaría la forma de generar ejecutables para cualquier plataforma, dejando así lo anterior inservible e innecesario.

Configuración de un túnel con seguridad TLS, modelo cliente-servidor para el caso GNU/Linux-OpenBSD a partir del parche que ofrecía Juan José Cirlante.

Para esta prueba se utilizo el parche de Juan José Cirlante, el cual no implementa la capacidad de comunicación en IPv6 para OpenVPN, esta ya viene con el mismo, sino que la expande a más posibilidades.

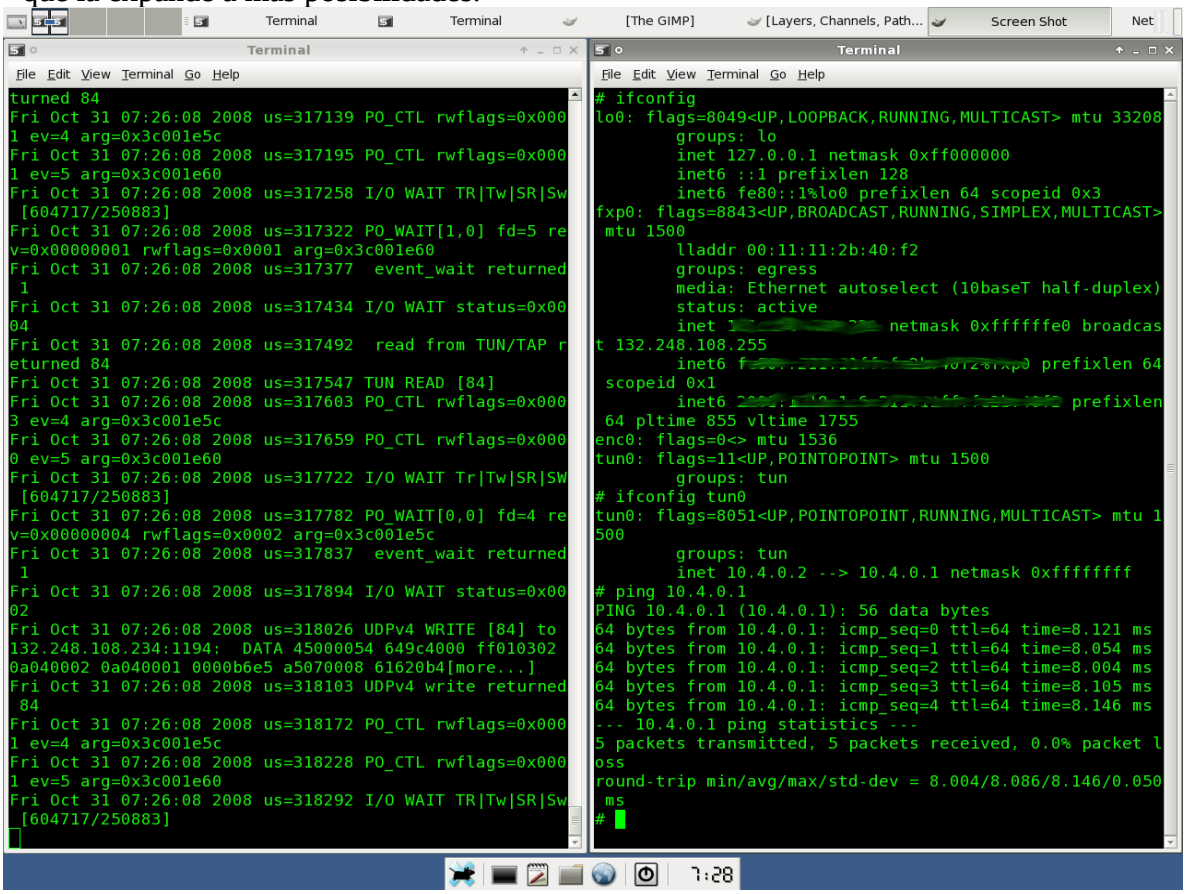


Figura 8. "Túnel con seguridad TLS, modelo cliente-servidor para el caso GNU/Linux-OpenBSD a partir del código parchado bis."

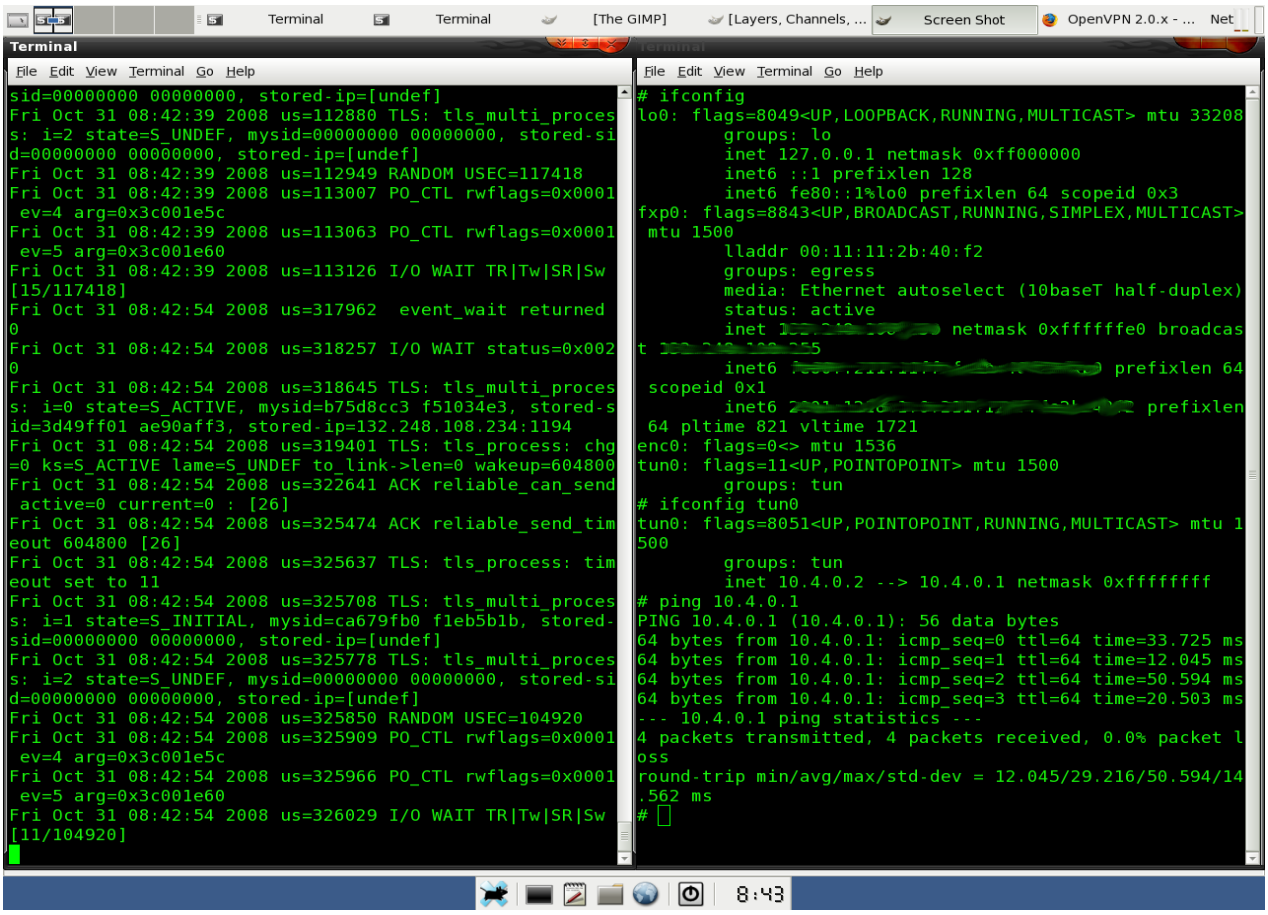


Figura 9. "Túnel con seguridad "

Se procedió a ejecutar el siguiente conjunto de instrucciones para recrear el túnel y se configuró de la misma manera, además de agregar el uso de la llave pública. Para OpenBSD 4.2

Cuadro 22. "Sentencia para ejecutar OpenVPN en un cliente OpenBSD con dirección local IPv6"

```
#  openvpn --remote fe80::2c0:4fff:fead:dcd2 1194 --dev tun0 –proto udp6 --ifconfig 10.4.0.2 10.4.0.1 –tls-client –dh /usr/local/share/examples/openvpn/sample-keys/dh1024.pem –ca /usr/local/share/examples/openvpn/sample-keys/tmp-ca.crt –cert /usr/local/share/examples/openvpn/sample-keys/client.crt –key /usr/local/share/examples/openvpn/sample-keys/client.key –reneg 60 -- verb 9
```

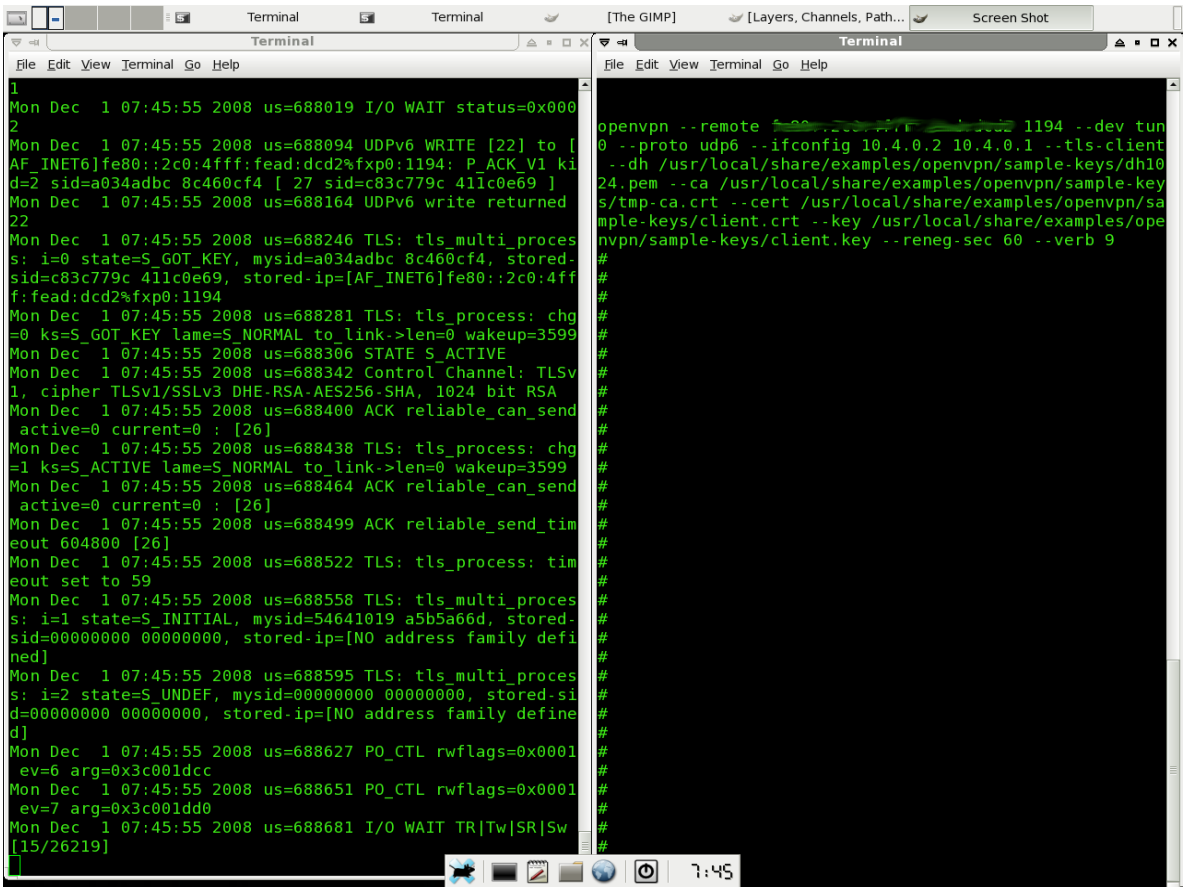


Figura 10. "Túnel con seguridad bis."

Para OpenBSD 4.2

```
#  openvpn --remote 2001:1218:1:6:2c0:4fff:fead:dcd2 1194 --dev tun0 –proto udp6
--ifconfig 10.4.0.2 10.4.0.1 –tls-client –dh /usr/local/share/examples/openvpn/sample-
keys/dh1024.pem –ca /usr/local/share/examples/openvpn/sample-keys/tmp-ca.crt –cert
/usr/local/share/examples/openvpn/sample-keys/client.crt –key
/usr/local/share/examples/openvpn/sample-keys/client.key –reneg 60 -- verb 9
```

Cuadro 23. "Sentencia para ejecutar OpenVPN en un cliente OpenBSD con dirección global IPv6"

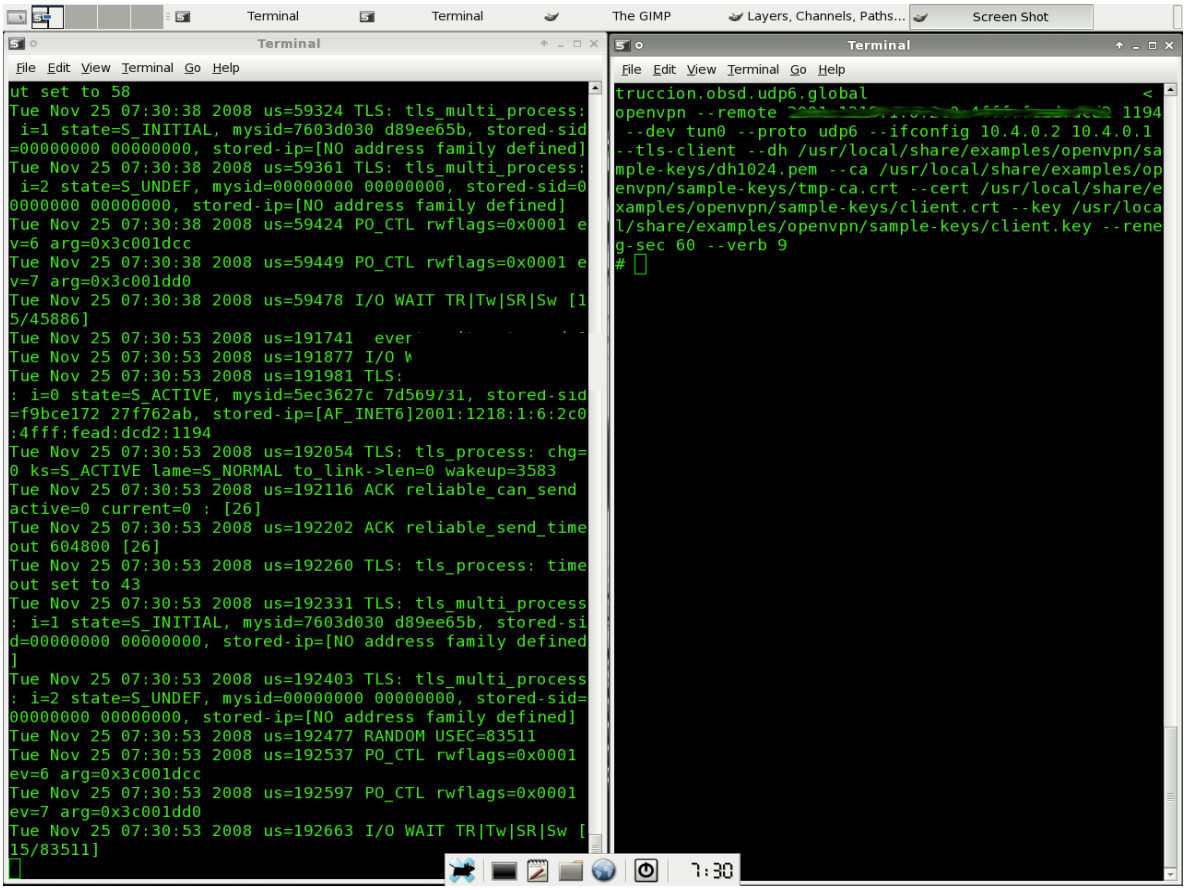


Figura 11. "Túnel con seguridad -tres"

En este caso se logro la conectividad y respuesta positiva desde cliente al servidor y viceversa, con seguridad habilitada en el modelo cliente servidor. En esta prueba no se incluyen clientes MS Windows porque no existe soporte para que estos convivan con clientes Unix.

Configuración de un túnel con seguridad TLS, modelo cliente-servidor para el caso con sistemas GNU/Linux en equipos de diferente segmento.

Para el caso de las pruebas que se realizaron entre equipos en diferentes segmentos utilizamos los mismos scripts del caso **8**. La única variación fue la dirección IP. Pero no se puedo lograr la conexión, entre ambos puntos , la respuesta observada fue el 'congelamiento' de la conexión que se era observada desde el servidor. Se presume que la infraestructura no es la adecuada para realizar estas pruebas.

10.Configuración de un túnel modelo cliente-cliente para el caso con sistemas MS Windows .

Para esta serie de pruebas se utilizaron las versiones XP profesional y Vista Home Edition. En ningún caso se logró la conexión con el otro extremo.

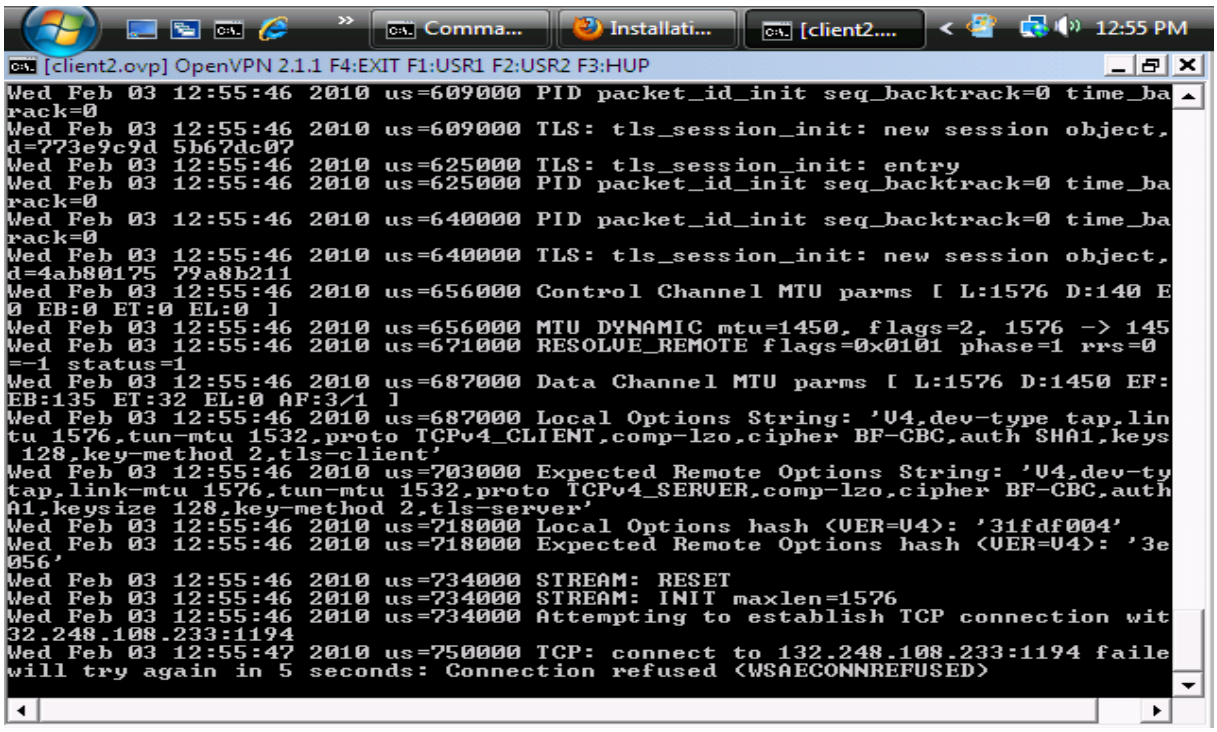


Figura 12. *"Conexión fallida Windows Vista Home Edition"*

11.Configuración de un túnel modelo cliente-cliente con seguridad de llave estática para el caso con sistemas MS Windows .

En estas pruebas los sistemas fallaron al intentar la conexión, como en el caso anterior.

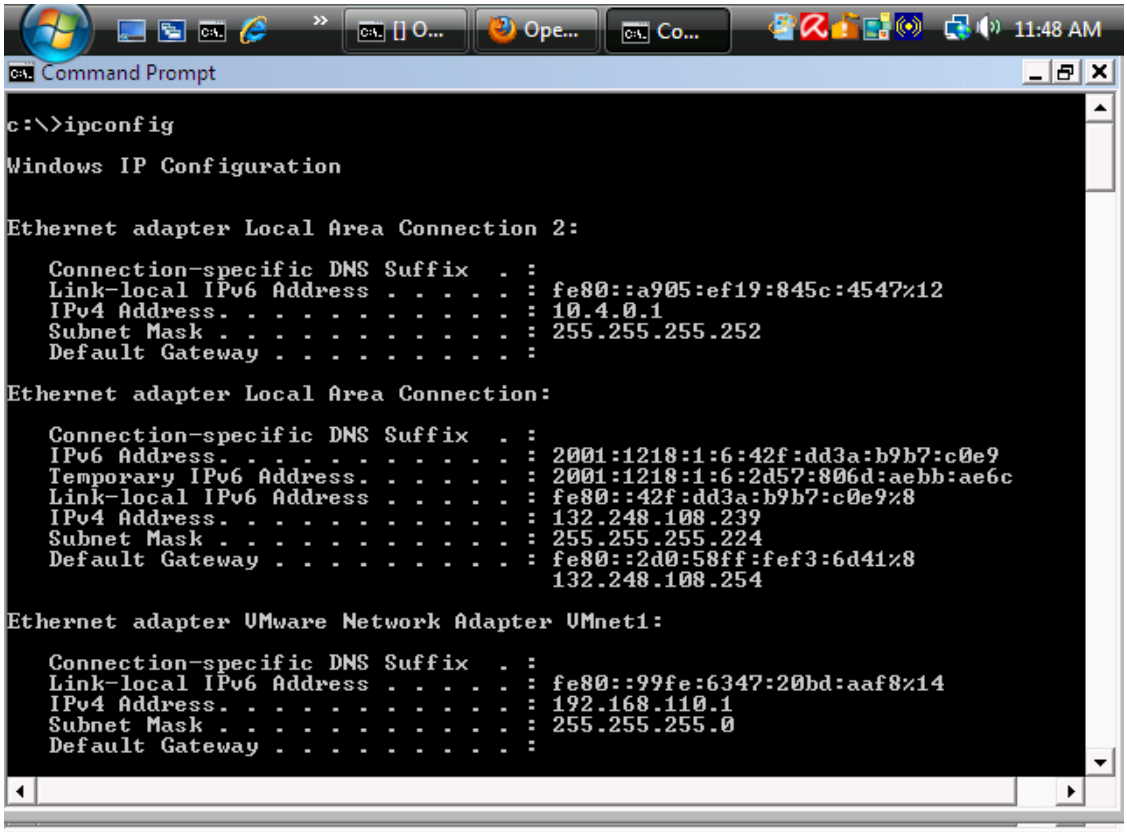


Figura 13. *"Configuración de un túnel IPv4"*

12. Configuración de un túnel modelo cliente-servidor con seguridad completa basada en TLS para el caso con sistemas MS Windows .

No fue posible lograr la conexión para los sistemas en prueba. Se hicieron pruebas de monitoreo de red en aislamiento para dichas pruebas pero sin embargo no se encontró una razón para la falla en estas pruebas con estos sistemas operativos.

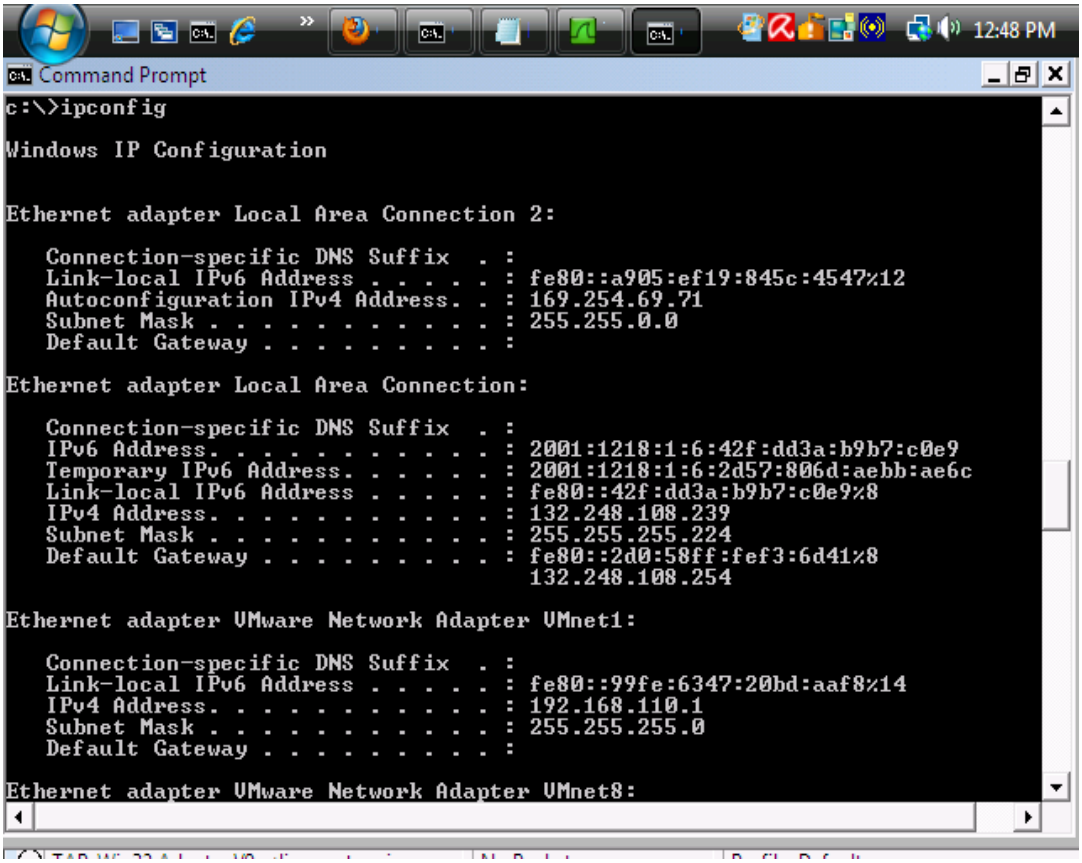


Figura 14. *"Configuración fallida de un túnel IPv4"*

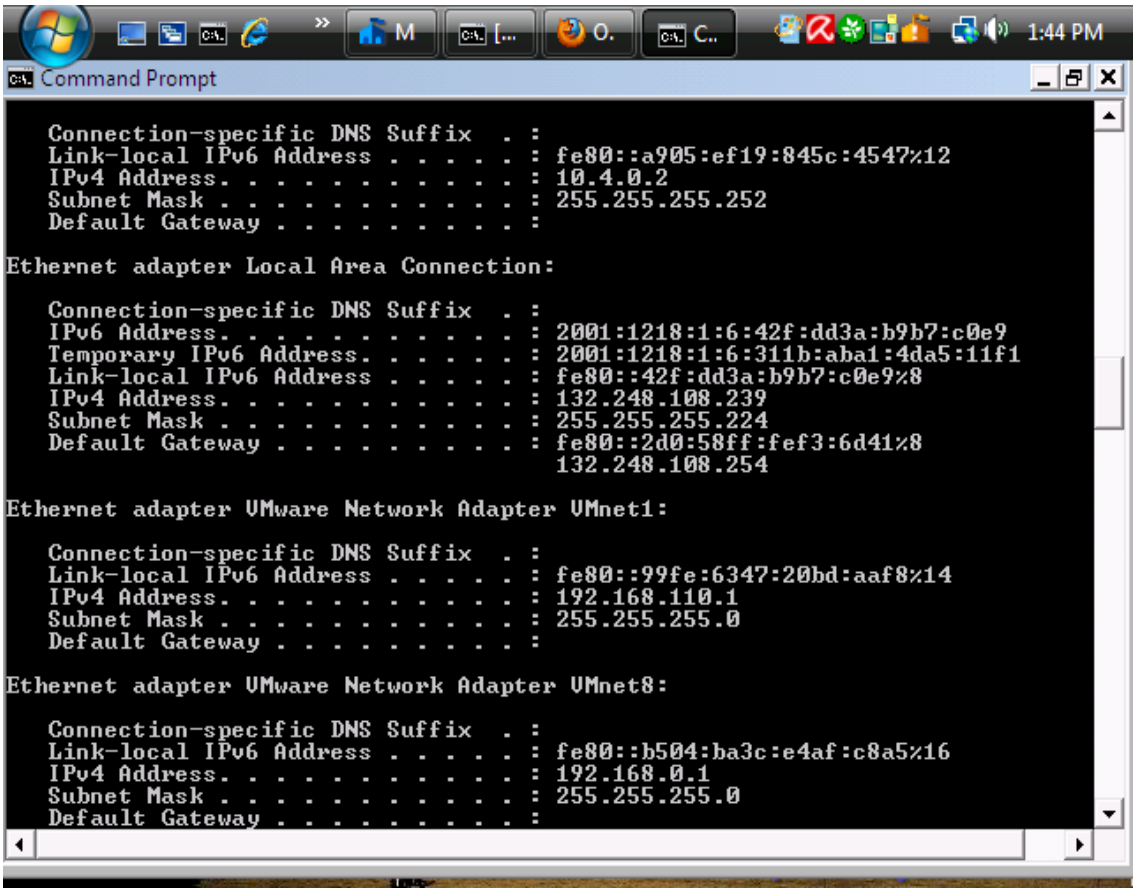


Figura 15. "Interfaces de red configuradas en Windows Vista Home Edition"

13.Configuración de un túnel con seguridad TLS, modelo cliente-servidor a partir del parche que ofrecían Bernhard Schmidt y Gert Döring.

Para este caso, se uso el parche que ofrecen los desarrolladores Bernhard Schmidt y Gert Döring. Este parche ofrece capacidades punto a multipunto con la interfaz tun o tap, punto a punto con interfaz tun y tap. Para instalarlo se procede de la siguiente manera:

1) Se descarga el código fuente de la página oficial del proyecto OpenVPN. Se obtendrá un paquete comprimido, puede ser en formato zip o tar.gz, llamado entonces openvpn N.tar.gz o puede ser openvpn N.zip. La N corresponderá al número de la versión en curso. Al momento de escribir este reporte es la versión openvpn-2.1.1

2) Se descomprime el paquete del código fuente obtenido. Dependerá del formato de compresión escogido para descargarlo. En el caso de un archivo zip y estando en un sistema tipo Unix, utilícese el comando unzip, para un *tarball* o sea un tar.gz, utilícese el comando tar -xfz . En caso de estar en un sistema tipo MS Windows , herramientas como WinZip o WinRar son suficientes para el propósito.

3) Una vez obtenido el directorio generado por el archivo descomprimido, descarge el parche de la página de los desarrolladores. http://www.greenie.net/ipv6/openvpn-2.1-ipv6-20100307-1.patch.gz . Al momento de escribir este reporte el parche es para la versión del punto *1*.

4) Para aplicar el parche del código fuente, descomprima el archivo con el comando gunzip. Después utilice el comando patch, de la manera siguiente.

$ patch < *archivo-de-.parche.patch*

A continuación verá como al código fuente original se le aplican ciertos cambios que como resultado nos darán lo necesario para nuestros propositos.

5) Una vez que ya se tiene el código fuente modificado se procede a compilarlo. Hay que tener en el sistema instaladas las bibliotecas de programación siguientes.

- Propenso
- LZO
- pkcs-helper

Estas son necesarias a fin de que el binario generado sea capaz de proveer el uso de la capa de seguridad SSL, la compresión lzo y el manejo de los certificados de seguridad y autenticaciones pkcs-helper.

Esta es la secuencia de comandos necesaria para generar los binarios nativos a partir del código fuente parchado.

./configure

make

make install

6) Una vez terminada exitosamente la compilación, tenemos instalado en nuestro sistema, los binarios o archivos ejecutables , dado el caso, con lo que podemos realizar nuestros propósitos.

Para levantar el servicio de la VPN con soporte para IPv6, utilizaremos la siguiente

```
openvpn --port 1194 --proto tcp --dev tun --ca /usr/share/doc/openvpn/examples/sample-keys/ca.crt --cert /usr/share/doc/openvpn/examples/sample-keys/server.crt --key /usr/share/doc/openvpn/examples/sample-keys/server.key --dh /usr/share/doc/openvpn/examples/sample-keys/dh1024.pem --server 10.8.0.0 255.255.255.0 --server-ipv6 2001:448:1:1::100/64 --ifconfig-pool-persist ipp.txt --keepalive 10 120 --comp-lzo --max-clients 10 -- persist-key --persist-tun --status /etc/openvpn/logs/openvpn-status.serverIPv6-patch-BS-GD-3.log --log-append /etc/openvpn/logs/openvpn.serverIPv6-pathc-BS-GD-2-2.log --verb 9
```

Cuadro 24. "Sentencia para ejecutar OpenVPN en un servidor con soporte para IPv6"

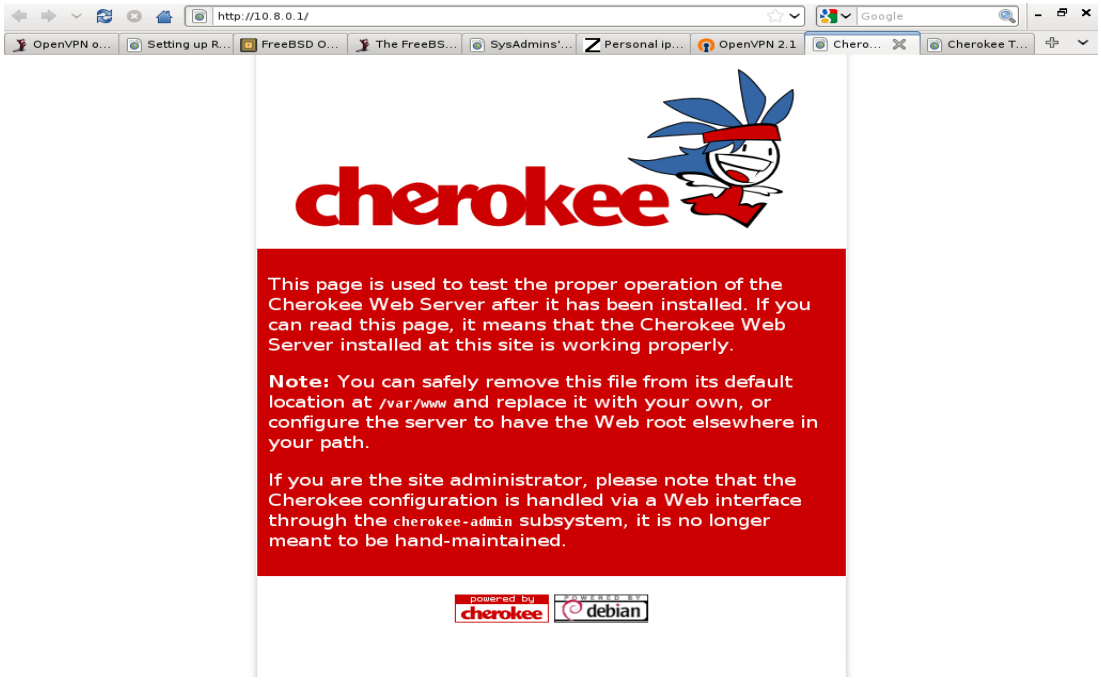


Figura 16 "Servidor Web Cherokee con dirección IPv4"

Con lo que se tiene un servidor VPN con soporte para IPv6. Verificamos que es capaz de transmitir tráfico con el servidor web Cherokee, tanto para IPv4 como para IPv6.

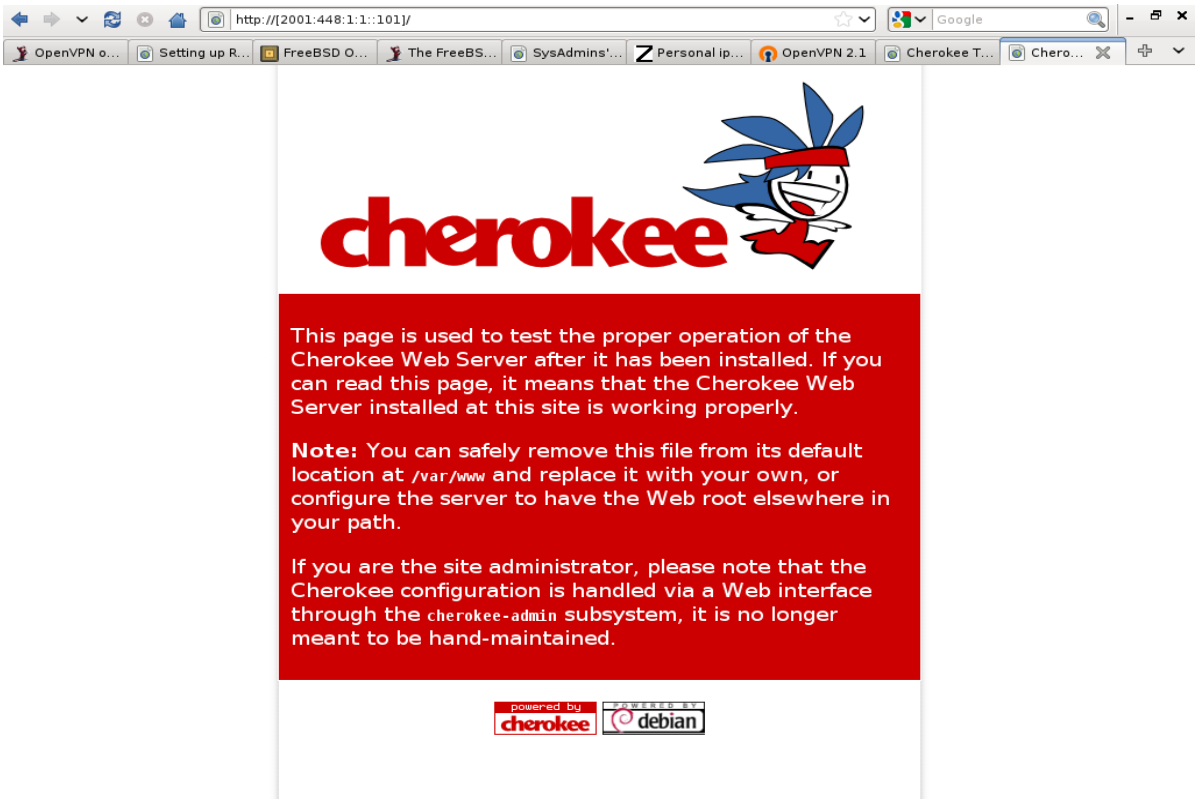


Figura 17 "Servidor Web Cherokee con dirección IPv6"

Se puede ver que este está funcionando en el sistema operativo FreeBSD.

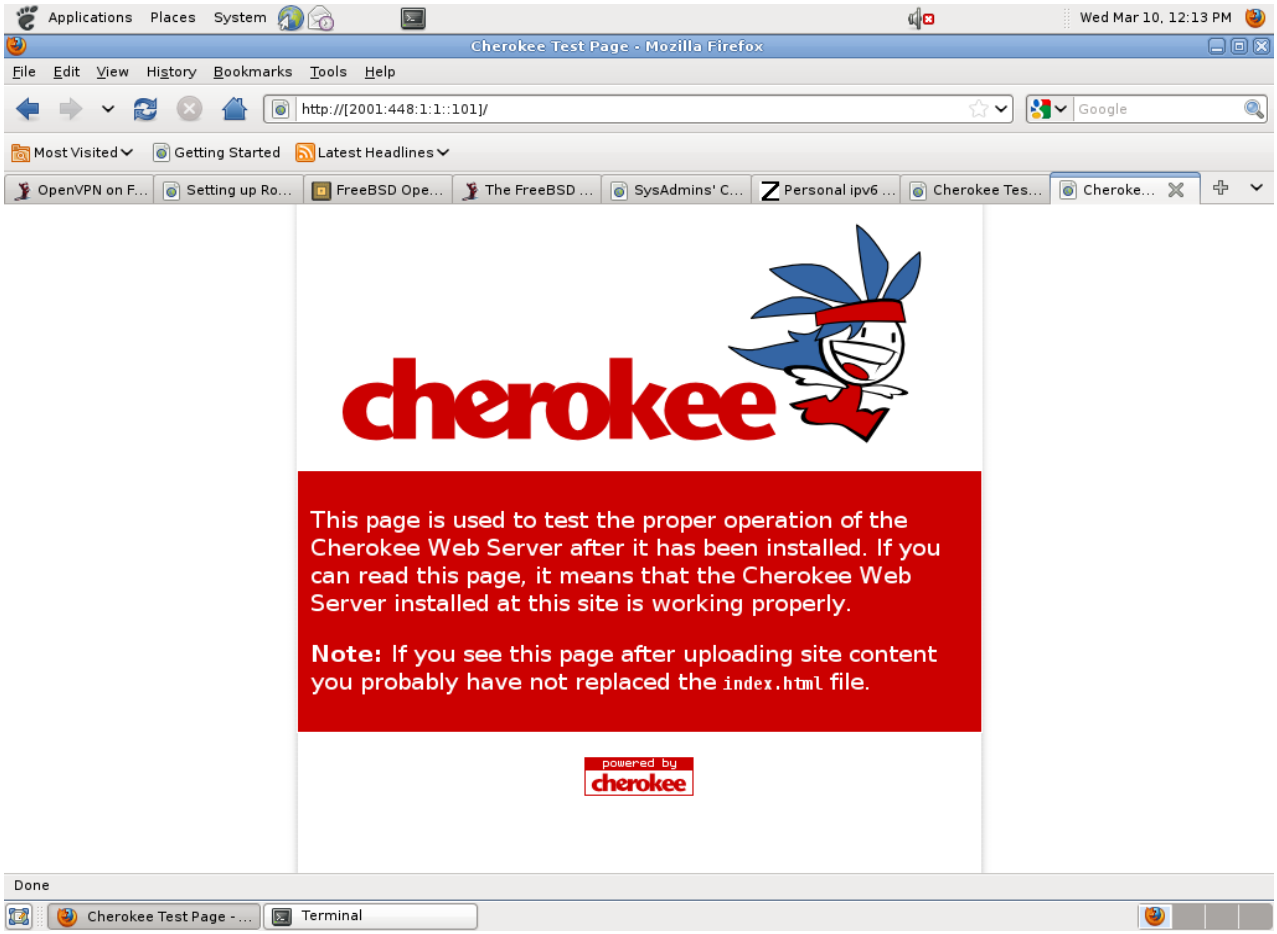


Figura 18 " Servidor Web Cherokee con dirección IPv6 en Fedora"

También funciono para el sistema Fedora Linux, por lo que se puede ver que este funciona en todos los sitemas Unix reales. No fue posible hacerlo funcionar en los sistemas MS Windows. En el caso de los clientes se utilizaron los siguientes archivos de configuración.

```
/usr/local/sbin/openvpn --client --dev tun0 --tun-ipv6 --proto tcp --remote 132.248.108.233 --port 1194 --resolv-retry infinit e --nobind --persist-tun --persist-key --ca /home/netlab/Downloads/openvpn-2.1.1-Berhard-Schmid-Gert-Doering/sample-keys/ca.cr t --cert /home/netlab/Downloads/openvpn-2.1.1-Berhard-Schmid-Gert-Doering/sample-keys/client1Debian.crt --key /home/netlab/Dow nloads/openvpn-2.1.1-Berhard-Schmid-Gert-Doering/sample-keys/client1Debian.key --comp-lzo --status /etc/openvpn/logs/openvpn-s tatus.clientIPv6-patch-BS-GD-deb5-2.log -log-append /etc/openvpn/logs/openvpn.clientIPv6-patch-BS-GD-deb5-2-1.log --verb 9
```

Cuadro 25 "Sentencia de ejecución del cliente Debian al conectarse al servidor Fedora VPN IPv6"

```
/usr/local/sbin/openvpn --client --dev tun0 --tun-ipv6 --proto tcp --remote 132.248.108.239
--port 1194 --resolv-retry infinit e --nobind --persist-tun --persist-key --ca
/home/netlab/Downloads/openvpn-2.1.1-Berhard-Schmid-Gert-Doering/sample-keys/ca.cr t
--cert /home/netlab/Downloads/openvpn-2.1.1-Berhard-Schmid-Gert-Doering/sample-
keys/client2Debian5.crt --key /home/netlab/Do wnloads/openvpn-2.1.1-Berhard-Schmid-
Gert-Doering/sample-keys/client2Debian5.key --comp-lzo --status /etc/openvpn/logs/openvpn
-status.clientIPv6-patch-BS-GD-deb5-FBSD-3.log
--log-append /etc/openvpn/logs/openvpn.clientIPv6-patch-BS-GD-deb5-FBSD-3-1.log
 --verb 9
```

Cuadro 26 "Sentencia de ejecución del cliente Debian al conectarse al servidor FreeBSD VPN IPv6"

14.Tablas de resultados

En resumen, de las pruebas realizadas entre los equipos, las comunicaciones que se realizaron entre ellos pueden verse en las siguientes tablas. La siguiente tabla muestra la conectividad entre los diversos sistemas que se probaron bajo el modelo cliente-cliente (punto a punto). Como se puede apreciar , únicamente el sistema Windows Vista fue incapaz de lograr la conectividad con los demás sistemas. En todas las versiones de Linux y en los BSD's , así como en Windows XP si existe la conexión.

Conectividad IPv4 VPN (modelo cliente-cliente)						
Sistemas Operativos	Windows XP SP3	FreeBSD 6/7.2	OpenBSD 4.2	Fedora 6/11	Debian 4/5	Windows Vista SP1
FreeBSD 6/7.2/8	Sí	----------	Sí	Sí	Sí	NO
OpenBSD 4.2	Sí	Sí	----------	Sí	Sí	NO
Fedora 6 /11	Sí	Sí	Sí	----------	Sí	NO
Debian 4/5	Sí	Sí	Sí	Sí	----------	NO
Windows XP SP3	----------	Sí	Sí	Sí	Sí	NO
Windows Vista SP1	NO	NO	NO	NO	NO	----------

Tabla 2."Resumen de los resultados de la conectividad con IPv4 en la VPN."

En la tabla que se obtuvo para el modelo cliente-servidor, se tiene que existe la conectividad en todas las versiones de Linux y en los BSD's, como servidores. En esta parte no es posible usar a los sistemas Windows como servidores dada la implementación que estos tienen para su interfaz de red. En cuanto la parte de los clientes, nuevamente todos los sistemas Linux y los BSD's, así como Windows XP SP3, son capaces de conectarse con los servidores, para la versión de Windows Vista esto no es posible.

Conectividad IPv4 VPN (modelo cliente-servidor)						
Cliente / Servidor	Windows XP SP3	Windows Vista SP1	FreeBSD 6/7.2/8	OpenBSD 4.X	Fedora 6/11	Debian 4/5
FreeBSD 6/7.2/8	Sí	NO	----------	Sí	Sí	Sí
OpenBSD 4.2	Sí	NO	Sí	----------	Sí	Sí
Fedora 6 /11	Sí	NO	Sí	Sí	---------	Sí
Debian 4/5	Sí	NO	Sí	Sí	Sí	----------

Tabla 3.“Resumen de los resultados de la conectividad con IPv4 en la VPN. Modelo cliente-servidor.”

En la siguiente tabla lo que muestran son los resultados de la conectividad que se lograron, al utilizar scripts de configuración en el lado del servidor a fin de que en los mismos se tuviera una conexión IPv6 . Todas las direcciones se configuraban manualmente tanto del lado del servidor como del cliente. En esta tabla se observa que los sistemas Windows, en sus versiones XP y Vista, además de OpenBSD no fueron capaces de soportar este tipo de configuración, en los primeros por la incapacidad de la plataforma para soportar estas implementaciones y en el ultimo por razones no específicas.

Conectividad IPv6 VPN (modelo cliente-servidor)(Direcciones Globales Manuales) OpenVPN v. X						
Cliente / Servidor	Windows XP SP3	Windows Vista SP1	FreeBSD 7.2	OpenBSD 4.5/4.6	Fedora 11	Debian 5
FreeBSD 7.2/8	Sin soporte	Sin soporte	----------	NO	Sí	Sí
OpenBSD 4.6	Sin soporte	Sin soporte	NO	----------	NO	NO
Fedora 11 Leonidas	Sin soporte	Sin soporte	Sí	Sí	----------	Sí
Debian 5 Lenny	Sin soporte	Sin soporte	Sí	Sí	Sí	----------

Tabla 4.“Resumen de los resultados de la conectividad con IPv6 en la VPN. Modelo cliente-servidor.”

En la siguiente tabla, se utilizó el parche que el desarrollador argentino Juan José Ciarlante ofrece para agregar capacidades IPv6 al software OpenVPN. Este parche ofrece la conectividad punto a punto y funcionó en los sistemas Linux (Debian y Fedora) y FreeBSD, para la parte de los sistemas Windows, esto no fue posible porque no esta bien soportado este desarrollo en Windows. En el sistema OpenBSD al intentar generar el paquete fallaba.

Conectividad IPv6 VPN (modelo cliente-servidor) con parche (Direcciones Automáticas)						
Cliente / Servidor	Windows Vista SP1	Windows XP SP3	FreeBSD 6/7.2/8	OpenBSD 4.6	Fedora 11	Debian 5
FreeBSD 8	No funcionó	No funcionó	----------	No funcionó	Sí	Sí
OpenBSD 4.6	No funcionó	No funcionó	No funcionó	----------	No funcionó	No funcionó
Fedora 11 Leonidas	No funcionó	No funcionó	Sí	No funcionó	----------	Sí
Debian 5 Lenny	No funcionó	No funcionó	Sí	No funcionó	Sí	----------

Tabla 5.“Resumen de los resultados de la conectividad con IPv6 en la VPN. Modelo cliente-servidor punto a punto. Compilado con el parche de Juan José Cirlante.”

Para la última tabla, es donde se pueden ver los resultados del trabajo de los desarrolladores alemanes Bernhard Schmidt y Gert Döring, quienes desarrollaron su propio parche para la conectividad IPv6. Dicho parche ofrece conectividad punto a multipunto con lo cual se alcanzo el objetivo buscado, el modelo cliente-servidor. Desafortunadamente tampoco funciona para los sistemas Windows, ni en OpenBSD. En los Linux (Debian o Fedora) y en FreeBSD no tiene ningún problema.

Conectividad IPv6 VPN con parche de Bernard Schmidt & Gert Döring(Direcciones Automáticas)						
Cliente / Servidor	Windows Vista SP1	Windows XP SP3	FreeBSD 8	OpenBSD 4.6	Fedora 11	Debian 5
FreeBSD 8	Sin soporte	Sin soporte	----------	No funcionó	Sí	Sí
OpenBSD 4.6	Sin soporte	Sin soporte	No funcionó	----------	No funcionó	No funcionó
Fedora 11 Leonidas	Sin soporte	Sin soporte	Sí	No funcionó	----------	sí
Debian 5 Lenny	Sin soporte	Sin soporte	Sí	No funcionó	Sí	----------

Tabla 6.“Resumen de los resultados de la conectividad con IPv6 en la VPN. Modelo cliente-servidor punto a multipunto. Compilado con el parche de Bernhard Shcmidt y Gert Döring.”

Glosario de términos

3DES (Triple **DES**):algoritmo de cifrado.

AES (**A**dvanced **E**ncription **S**tandard):Estandar de Cifrado Avanzado.

AH (**A**uthentication **H**eader):Encabezado de Autenticación ,IPSec.véase capítulo 2.

Blowfish algoritmo de cifrado

CHAP (**C**hallenge **H**andshake **A**uthentication **P**rotocol): Protocolo de Autenticación por Desafío Mutuo, método de autenticación remota o inalámbrica.

DES (**D**ata **E**ncription **S**tandar): algoritmo de cifrado.

DHCP (**D**ynamic **H**ost **C**onfiguration **P**rotocol):Protocolo de Configuración Dinámica de servidores, protocolo de red, que permite a los nodos obtener los parámetros de configuración automáticamente.

ESP (**E**ncapsulating **S**ecurity **P**ayload): Encabezado de Carga de Seguridad de Encapsulamiento,uno de los encabezados de cifrado para IPSec. véase capítulo 2.

GRE (**G**eneric **R**outing **E**ncapsulation): Encapsulamiento Genérico de Enrutamiento, es un protocolo de tuneleo desarrollado por Cisco que puede encapsular una amplia variedad de paquetes de protocolos de Capa de red dentro de túneles IP.

HMAC (**H**ashed **MAC**): algoritmo de autenticación.

IANA (**I**nternet **A**ssigned **N**umber **A**uthority):Agencia de Asignación de Números Internet. Fue el antiguo registro central de protocolos,puertos , números de protocolo y empresa asociada a ellos, opciones y códigos.

ICV (**I**ntegrity **C**heck **V**alue):IPSec, véase capítulo 2.

IDEA (**I**nternational **D**ata **E**ncryption **A**lgorithm):

IETF (**I**nternet **E**ngineering **T**ask **F**orce): Organización internacional que participa en el desarrollo de los estándares (protocolos , algoritmos, etc) de la Internet.

IKE (**I**nternet **K**ey **E**xchange): véase pág. X

IP (**I**nternet **P**rotocol): Protocolo de capa 3. Este protocolo es el de mayor uso e implementación en las redes existentes. Se le denomina IPv4 debido a la nomenclatura de la versión destinada a sustituirlo.

IPLS (**IP**-only **LAN S**ervices): redes VPLS simplificadas, como en aquellas , las interfaces LAN corren en modo promiscuo, y los frames se envían en base a sus direcciones físicas (**MAC**) de destino.

IPSec (**I**nternet **P**rotocol **Sec**urity): Protocolo de seguridad, *véase capítulo 2.*

ISAKMP (**I**nternet **S**ecurity **A**ssociation and **K**ey **M**anagement **P**rotocol): Protocolo de Manejo de Llaves y Asociación Segura de Internet.

ISO (**I**nternational **S**tandars **O**rganization): La Organización Internacional de Estándares, es una organización no gubernamental, encargada de producir normas internacionales con la finalidad de facilitar el intercambio de información y el comercio.

ISP (**I**nternet **S**ervice **P**rovider): Proveedor de Servicio de Internet.

LAN-2-LAN redes virtuales que conectan dos LANs.

L2F (**L**ayer **2 F**orwarding) : Protocolo de envío de datos a través de la Capa 2 el modelo OSI. Creado por CISCO para establecer túneles de tráfico desde los usuarios remotos hasta las oficinas corporativas.

L2TPv2 (**L**ayer **2 T**unneling **P**rotocol **v**ersion **2**): Protocolo de túneleo a través de Capa 2 del modelo OSI. Básicamente el protocolo L2TP cómo se definió en el RFC 2661.

L1VPN redes virtuales que se manejan a nivel de capa 1, generalmente mediante el uso de dispositivos físicos.

L2VPN redes virtuales privadas que se manejan a nivel de capa 2 , del modelo OSI.

L3VPN redes virtuales privadas que se manejan a nivel de capa 3 , del modelo OSI.

MAC (Media **A**cces **C**ontrol**)**:control de acceso al medio, identificador de 8 bits, o 6 bloques hexadecimales , que teóricamente corresponden de forma única a un dispositivo de red ethernet.

MD5 (**M**essage **D**igest version 5):algoritmo de autenticación.

MPLS (**M**ulti **P**rotocol **L**abel **S**witching):Conmutación multiprotocolo mediante Etiquetas, mecanismo de transporte de datos estándar creado por la IETF. Opera entre las capas de enlace de datos (capa 2)y la capa de red (capa 3) del modelo OSI.

NIC (**N**etwork **I**nterface **C**ard): tarjeta de red.

OSI (**O**pen **S**ystem **I**nterconneted): El modelo de Interconexión de Sistemas Abiertos fue propuesto por la ISO, describiendo como deberían conectarse las distintos equipos de computo y redes para poder interactuar entre sí.

PAP (**P**rotocol **A**uthentication **P**assword): Protocolo de autenticación de clave, autentica a un usuario contra un servidor de acceso.

PKI (**P**ublic **K**ey **I**nfrastructure): Infraestructura de Llave/clave Pública, combinación de hardware, software y políticas de seguridad que permiten la ejecución segura de operaciones de cifrado, firma digital , y no repudio de transacciones electrónicas.

PPoE (**P**oint to **P**oint **o**ver **E**thernet): Protocolo Punto a Punto sobre Ehernet, protocolo de red para la encapsulamiento PPP sobre una capa de Ethernet.

PPP (**P**oint to **P**oint **P**rotocol): Protocolo Punto a Punto, protocolo que se maneja a nivel de la Capa de Enlace estandarizado en el RFC 1661.

PPTP (**P**oint to **P**oint **T**unneling **P**rotocol): protocolo desarrollado por el PPTP Forum para implementar redes virtuales privadas.

RFC (**R**equest **F**or **C**omments): Documentos de especificaciones que se exponen públicamente para su discusión.

RIPEMD (**RACE I**ntegrity **P**rimitives **E**valuation **M**essage **D**igest):Algoritmo de Autenticación.

RADIUS (**R**emote **A**uthentication **D**ial **I**n **U**ser **S**ervice): es un protocolo de autenticación y autorización para aplicaciones de acceso a la red o movilidad IP.

SA (**S**ecurity **A**ssociations): Asociaciones de Seguridad de IPSec. véase pág. X

SHA-1(**S**ecure **H**ash **A**lgorithm):algoritmo de autenticación.

SONET (**S**ynchronous **O**ptical **N**etworks): estándar para el transporte de telecomunicaciones en redes de fibra óptica.

SP (**S**ecurity **P**olicies): Políticas de Seguridad de IPSec.

SPI (**S**ecurity **P**arameter **I**ndex): Índice de Parámetros de Seguridad, es el identificador de seguridad utilizado por IPSec.

SSL (**S**ecure **S**ocket **L**ayer): Protocolo de Capa de Conexión Segura, proporciona autenticación y privacidad de la información entre los extremos de una conexión a través de Internet mediante el uso de algoritmos de cifrado.

TCP (**T**ransmission **C**ontrol **P**rotocol): protocolo de transporte orientado a conexión, utilizado en internet para establecer comunicaciones confiables.

TTL (**T**ime **T**o **L**ive): campo de 8 bits dentro del encabezado IPv4.

UDP (**U**ser **D**atagram **P**rotocol):Protocolo de Datagrama a nivel de Usuario, es un protocolo de nivel de transporte basado en el intercambio de datagramas a través de la red sin necesidad de que se haya establecido con anterioridad una conexión.

VPLS (**V**irtual **P**rivate **LAN S**ervices):una manera de proveer comunicacionesss ethernet multipunto a multipunto sobre redes MPLS/IP.

VPN (**V**irtual **P**rivate **N**etwork):Red Privada Virtual, son aquellas redes que utilizan la infraestructura pública para crear una red virtual , que al cifrar el contenido se vuelve 'privada' con respecto al demás contenido.

VPWS (**V**irtual **P**rivate **W**ire **S**ervices):proveen conectividad punto a punto entre los sitios de los clientes, donde la red del proveedor de servicios emula un conjunto de cables entre los sitios de los clientes bajo un túnel MPLS.

VC-ATM (**V**irtual **C**ircuit- **A**synchronous **T**ransfer **M**ode): Circuito Virtual – Modo de Transferencia Asíncrona , sistema de comunicación que hace uso de más de un circuito real durante un periodo de tiempo, donde la conmutación es transparente para el usuario.

WAN (**W**ide **A**rea **N**etwork): uno de los tipos de red de computadoras que se extiende sobre un área geográfica amplia.

WiFi nombre comercial por el que se conoce al protocolo IEEE 802.11b de secuencia directa. Provee comunicaciones inalámbricas.

Bibliografía

Building and Integrating Virtual Private Networks with Openswan, Packt Publishing, 2006

Comparing, Designing, and Deploying VPNs, Cisco Press, 2006

Troubleshooting Virtual Private Networks, Cisco Press; 2Rev Ed edition, 2004

Wei Luo, Layer 2 VPN architectures, Cisco, c2005

Tan, Nam-Kee, Building VPNs : with IPSec and MPLS, McGraw-Hill, c2003

Peter H. Salus, Big book of IPv6 addressing RFCs, 2000

Regis Desmeules, Cisco self-study : implementing IPv6 networks (IPV6), 2003

Ciprian Popoviciu, Eric Levy-Abegnoli, Patrick Grossetete, Deploying IPv6 networks, 2006

Feilner, Markus OpenVPN Building and Integrating Virtual Private Networks, Packt Publishing 2006

Benedikt Stockebrand, IPv6 in practice : a Unixer's guide to the next generation Internet, Springer, 2006

Hagino, Jun-ichiro itojun, IPv6 network programming, Elsevier Digital, c2005

Hosner, Charlie OpenVPN and the SSL Revolution, SANS Institute, 2004

European Commission, IPv6 and Broadband (IPv6 Cluster), Information Society Technologies , 2002

Hagen Silvia, IPv6 Essentials, ed. O'Reilly Media Inc. EUA 2002

Iljitsch van Beijnum, Running IPv6, ed. Apress, E.U. A., 2006

Parenti Edgar Jr., Browne Brian, Knipp Eric, Configuring IPv6 for Cisco IOS, ed. Syngress, EUA, 2002

Welsh Matt, Kalle Matthias, Kaufman Lar, Running Linux, ed. O'Reilly, 3a ed., California EUA, 1999

Dunmore, Martin An IPv6 deployment guide, Javvin Technologies Inc. Distribution 2009

Artículos

G Shorrock and C Awdry, Concert IP Secure – a managed firewall and VPN Service

Páginas de internet

OpenVPN Proyecto OpenVPN http://www.openvpn.org

Mingw Proyecto Cross-Compiler MinGW http://www.mingw.org

Debian Distribución Linux http://www.debian.org

Fedora Distribución Linux http://www.fedora.org

MS Windows http://www.microsoft.asp

6BONE deployment of IPv6 http://www.6bone.net

6sos información y soporte para IPv6 http://www.6sos.org

BSD soporte de los BSD's para IPv6 http://www.freebsd.org http://www.openbsd.org

CUDI Corporación Universitaria para el desarrollo de internet http://www.cudi.edu.mx

IPv6 Ready Committeé http://www.ipv6ready.org/frames.html

IPv6 información sobre IPv6 http://www.ipv6.org

UNAM IPv6 en la Universidad Nacional Autónoma de México
http://www.ipv6.unam.mx

NetLab Laboratorio de Tecnologías Emergentes DGSCA http://www.netlab.unam.mx

IRC (Internet Relay Channel)

##openvpn	#mingw
#freebsd	#openbsd
#fedora	#lidsol
#debian	#debian-mx
#ipv6	#unix-mexico

Listas de correo

openvpn, openvpn-developers , openvpn-users, mingw, debian, fedora, freebsd, openbsd, debia-mx , fedora-es, lidsol