

UNIVERSIDAD DEL AZUAY

FACULTAD DE CIENCIA Y TECNOLOGIA

ESCUELA DE INGENIERIA ELECTRONICA

RED PRIVADA VIRTUAL BAJO LINUX

**TRABAJO DE GRADUACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO ELECTRONICO**

**AUTOR:
CRISTIAN SEGUNDO TELLO VALLADARES**

**DIRECTOR:
LIC. LEOPOLDO CARLOS VASQUEZ RODRIGUEZ**

**CUENCA - ECUADOR
2009**

DEDICATORIA

“Cada hombre puede mejorar su vida mejorando su actitud”.

La culminación de una de mis metas en la vida, deseo dedicarle de una manera muy especial a Dios por darme salud, consistencia y dedicación, a mis Padres y mi esposa quienes con su apoyo incondicional me impulsaron a terminar mi carrera profesional.

AGRADECIMIENTO

“Para ser exitoso no tienes que hacer cosas extraordinarias. Haz cosas ordinarias, extraordinariamente bien.”

Un agradecimiento muy especial a mi amada esposa por el apoyo que me brindó cuando más decaído me sentía, su ánimo y valor fueron mi impulso y dedicación para concluir esta carrera profesional

A mí querida madre y su lucha constante en el día a día, me enseñaron que las adversidades solo se las sobrepasa afrontándolas, enseñanzas que le ayudan a uno a crecer como profesional y principalmente como ser humano.

Agradezco a mis profesores de la Universidad del Azuay por haberme brindado sus conocimientos para un mejor desarrollo intelectual.

A mis directores y tutores de Tesis por enriquecerme con sus conocimientos y compartir sus enseñanzas para la culminación de esta carrera.

RESUMEN

Antiguamente las empresas adoptaron computadoras personales autónomas, los inconvenientes en el intercambio de información les obligó a cambiar de esta red lenta a redes de área local LAN de alta velocidad hasta convertirse en redes extensas, uno de los tipos de enlaces WAN mas económicos son las Redes Privadas Virtuales o VPNs, La idea principal de nuestro proyecto es implementar una red VPN totalmente aplicable al campo empresarial teniendo en cuenta sus tipos de conexión y características de ser aplicadas bajo software de libre distribución como LINUX, características que han hechos de ellas muy útiles debido a su bajo costo de inversión.

ABSTRACT

Olderly companies adopted personal computers, disadvantages in the exchange of information forces them to change this slow network to local area networks LAN high-speed networks to become large, one of the types of WAN links are more economical Networks VPNs or Virtual Private, The main idea of our project is to implement a VPN totally applicable to business field taking into account their connection types and features to be implemented in open source software like Linux, features that have made them very useful because of its low investment cost.

INDICE

Dedicatoria.....	i
Agradecimiento.....	ii
Resumen.....	iii
Abstract.....	iv
INTRODUCCION.....	1

CAPITULO 1: EVOLUCION DEL NETWORKING

1.1 INTRODUCCION.....	2
1.2 REDES DE AREA LOCAL. (LAN).....	3
1.3 REDES WAN.....	3
1.4 REVISION DE LOS PRINCIPALES SERVICIOS WAN DE TELECOMUNICACIONES.....	4

CAPITULO 2: REDES PRIVADAS VIRTUALES

INTRODUCCION.....	6
2.1 QUE ES UNA VPN.....	6
2.1.2 ELEMENTOS DE UNA VPN.....	7
2.2 ESTRUCTURA DE LAS VPNS	9
2.3 TIPOS DE VPN.....	9
2.3.1 VPNS DE HARDWARE.....	10
2.3.2 VPNS DE SOFTWARE.....	10
2.4 DIAGRAMAS DE VPNS.....	10
2.4.1 DE CLIENTE A SERVIDOR.....	11
2.4.2 DE CLIENTE A RED INTERNA.....	11
2.4.3 DE RED INTERNA A RED INTERNA.....	11
2.5 PROTOCOLOS DE ENCRIPCIÓN UTILIZADOS EN LAS VPNS.....	12
2.5.1 IMPLEMENTACIONES DE CAPA 2 – ENLACE.....	12

2.5.2 IMPLEMENTACIONES DE CAPA 3 – RED.....	13
2.5.3 IMPLEMENTACIONES DE CAPA 7 – APLICACIÓN.....	13
2.6 FUNCIONAMIENTO SSL / TLS.....	13
2.6.1 AUTENTICACIÓN SSL / TLS.....	13
2.6.2 COMO ESTABLECER UNA CONEXIÓN SSL/TLS.....	13
2.7 VENTAJAS Y DESVENTAJAS DE LAS VPN.....	16
2.7.1 VENTAJAS.....	16
2.7.2 DESVENTAJAS DE LAS VPN.....	18

CAPITULO 3: CONTRUCCION DE UNA RED VPN BAJO LINUX

INTRODUCCION.....	19
3.1 DIRECCIONAMIENTO DE RED PARA MATRIZ, SUCURSAL Y CLIENTES O USUARIOS REMOTOS.....	19
3.2 DIRECCIONAMIENTO IP PARA VPN.....	24
3.3 INSTALACION, CONFIGURACION Y HABILITACION DE LOS SERVIDORES EN LINUX.....	25
3.3.1 MONTAJE DEL HARDWARE PARA NUESTRO SERVER VPN.....	25
3.3.2 ELECCIÓN E INSTALACIÓN DEL SOFTWARE PARA CONFIGURACIÓN DE NUESTRO SERVIDOR VPN.....	25
3.4 INSTALACION Y CONFIGURACION DE OPEN-VPN EN NUESTROS SERVIDORES LINUX.....	26
3.5. CREACION DE CERTIFICADOS.....	29
3.5.1 CREACIÓN DE CERTIFICADOS Y ARCHIVOS DE CONFIGURACIÓN OPENVPN EN MATRIZ_A.....	29
3.6 CREACION DE CLAVE SECRETA PARA CONEXIONES RED A RED.....	35
3.7 CREACION Y COMPILACION DE ARCHIVOS DE CONFIGURACION PARA CONEXIONES VPN.....	36
3.7.1 CONFIGURACION HOST_RED.....	36
3.7.1.1 ARCHIVO DE CONFIGURACIÓN EN MATRIZ_A.....	36
3.7.1.2 ARCHIVOS DE CONFIGURACIÓN PARA REMOTO1 Y REMOTO2...39	39
3.7.2 CONFIGURACIÓN RED_RED.....	41
3.7.2.1 ARCHIVO DE CONFIGURACIÓN EN MATRIZ_A PARA CONEXIÓN RED-RED, TUNEL_MATRIZ-A_SUCURSAL_B.CONF.....	41

3.7.2.2 ARCHIVO DE CONFIGURACIÓN EN SUCURSAL_B PARA CONEXIÓN RED-RED, TUNEL_MATRIZ-A_SUCURSAL_B.CONF.....	43
3.8 RESUMEN GENERAL.....	44
3.9 COMPROBACION DE LA RED VPN.....	45
4. CONCLUSIONES Y RECOMENDACIONES.....	46
5. BIBLIOGRAFIA.....	47

Tello Valladares Cristian Segundo
Trabajo de Graduación
Leopoldo Carlos Vázquez Rodríguez
Febrero del 2009

RED PRIVADA VIRTUAL BAJO LINUX

INTRODUCCION

Con la presentación de este trabajo de graduación pretendemos que el mismo sea una referencia teórico - práctico para la realización de una VPN (Red Privada Virtual) con Linux. El documento esta dividido en tres capítulos donde consideramos:

En el primer capitulo antecedentes previos al uso de redes de telecomunicaciones hasta llegar al uso de redes de alta velocidad locales y extensas, comparación de características entre diferentes tipos de enlaces para unir puntos remotos, concluyendo con la adopción de una VPN como método de enlace para la conexión de redes remotas.

Los capítulos dos y tres detallan las características principales de las Redes Privadas Virtuales y nos ofrece la información necesaria para configurar y ejecutar nuestra VPN, los ejemplos utilizados en este documento están basados en configuraciones de red que siguen en funcionamiento en la actualidad, si seguimos toda la teoría y recomendaciones planteadas deberíamos ser capaces de hacer que nuestra Red Privada Virtual funcione rápidamente.

CAPITULO 1

EVOLUCION DEL NETWORKING

1.1 INTRODUCCION.

La adopción de computadoras personales por parte de las empresas fue lenta al principio. El lanzamiento de Lotus 1-2-3 y otras aplicaciones diseñadas específicamente para uso empresarial impulsó el rápido crecimiento de la industria del computador personal.

Al principio, una empresa invertía en computadoras como dispositivos autónomos a los que a veces se conectaban impresoras. Cuando los empleados que no tenían impresoras conectadas a sus computadoras necesitaban imprimir documentos, tenían que copiar los archivos en disquetes, cargarlos en el computador de algún compañero que tuviera impresora, e imprimirlos desde allí. Esta “red” rudimentaria se llamaba “red a pie”

A medida que las empresas se desarrollaban, las desventajas de la “red a pie” se hicieron evidentes. Como consecuencia, las empresas invirtieron en Redes de area local o LAN. La LAN permitía que los usuarios que se encontraban dentro de un mismo departamento pudieran transferir rápidamente archivos a través de la red electrónica.

Las impresoras autónomas fueron reemplazadas por impresoras de red de alta velocidad, compartidas por todo el departamento. Sin embargo, en aquel momento la “red a pie” era normalmente la única manera posible de compartir archivos con los empleados de otro departamento, o que estuvieran conectados en otra LAN.

La expansión de las empresas implicó en muchos casos la apertura de nuevas oficinas regionales de ventas en todo el mundo. Cada oficina disponía de su propia LAN, su propio software y hardware, y su propio administrador de red. Cada departamento funcionaba de manera eficiente, pero siempre electrónicamente aislado de los demás departamentos. A menudo esto representaba una operación ineficiente

que afectaba a toda la empresa, y provocaba demoras en el acceso a la información que se debía compartir. Tres diferentes problemas hicieron que fuera necesaria la internetworking: la duplicación de equipos y recursos, la incapacidad de comunicarse con cualquier persona, en cualquier momento y lugar, y la falta de una administración de LAN. Estos problemas se transformaron en oportunidades para las empresas que desarrollaban soluciones de internetworking para las redes de área local y amplia.

1.2 REDES DE AREA LOCAL. (LAN)

Una red de área local (LAN: Local Área Network) es la interconexión de dispositivos de Cómputo que pueden comunicarse entre sí y compartir un grupo de recursos comunes, como impresoras, discos, etcétera, Normalmente, están limitadas en distancia (5 Km.) por lo que pueden abarcar desde un departamento hasta un edificio, o todo un campus universitario. En general, el hecho de trabajar dentro de una red de área local es sencillo y garantiza accesos seguros a quienes se encuentran interconectados a través de su alta velocidad. Las redes de área local son cada vez más útiles ya que ayudan a evitar el traslado de una persona de un lugar a otro y a diseñar economías de escala, debido a que se pueden compartir recursos entre todos los usuarios de la red.

Las redes de área local (LAN) se componen de computadores, tarjetas de interfaz de red, medios del networking, dispositivos de control del tráfico de red y dispositivos periféricos. Las LAN hacen posible que las empresas que utilizan tecnología informática compartan de forma eficiente elementos tales como archivos e impresoras, y permiten la comunicación, por ejemplo, a través del correo electrónico. Unen entre sí computadores servidores y computadores clientes, (servidores de aplicaciones, servidores de bases de datos, servidores Web, etc.) en resumen:

Las LAN están diseñadas para realizar lo siguiente:

- Operar dentro de un área geográfica limitada
- Permitir que varios usuarios accedan a medios de ancho de banda alto
- Proporcionar conectividad continua con los servicios locales
- Conectar dispositivos físicamente adyacentes

1.3 REDES WAN.

A medida que el uso de los computadores en las empresas aumentaba, pronto resultó obvio que incluso las LAN no eran suficientes. En un sistema de LAN, cada departamento, o empresa, era una especie de isla electrónica. Lo que se necesitaba era una forma de transferir información de manera eficiente y rápida de una empresa a otra.

La solución surgió con la creación de las redes de área amplia (WAN). Las WAN interconectaban las LAN, que a su vez proporcionaban acceso a los computadores o a los servidores de archivos ubicados en otros lugares. Como las WAN conectaban redes de usuarios dentro de un área geográfica extensa, permitieron que las empresas se comunicaran entre sí a través de grandes distancias. Como resultado de la interconexión de los computadores, impresoras y otros dispositivos en una WAN, las empresas pudieron comunicarse entre sí, compartir información y recursos, y tener acceso a Internet.

Algunas de las tecnologías comunes de las WAN son:

módems

RDSI (Red digital de servicios integrados)

DSL (Digital Subscriber Line)(Línea de suscripción digital)

Frame relay

ATM (Modo de transferencia asíncrona)

Series de portadoras T (EE.UU. y Canadá) y E (Europa y America Latina): T1, E1, T3, E3, etc.

SONET (Red óptica sincrónica)

1.4 REVISION DE LOS PRINCIPALES SERVICIOS WAN DE TELECOMUNICACIONES

Con el crecimiento de las empresas y a medida que el uso de la computadora se hizo más importante en el trabajo diario de estas empresas, surgió la necesidad de comunicar las diferentes redes locales para compartir recursos internos de la empresa, Para explicar otra de gran importancia en nuestro diseño recurriremos al siguiente ejemplo: Cuando se necesita enlazar las oficinas centrales con alguna sucursal u oficina remota se tiene las siguientes opciones:

1. A._MODEMS: Las desventajas es el costo de las llamadas y bajas velocidades en la conexión.

2. B._Línea Privada: Tendido de cable ya sea de cobre, fibra óptica o Montaje de Radios punto a punto, el costo es muy elevado, por ejemplo si necesito enlazar mi oficina central con una sucursal a varios Km. de distancia sin tener en cuenta el tiempo de solución que habría en caso de producirse algún daño en nuestra línea privada.
3. C._Líneas Dedicadas: Enlaces WAN que nos ofrece un ISP nos garantizan siempre un UPTIME de nuestro enlace y seguridad en la información que sea pero todo esto a costos elevados y que aumentan en función del ancho de banda que contratemos,
4. D._ Enlaces VPNs. La idea de implementar una VPN haría reducir notablemente los costos de comunicación dado que las llamadas telefónicas (en caso de usar dial-up módems) serian locales (al proveedor de Internet) teniendo en cuenta que el enlace de Internet necesario para la creación de estas vpns y además mucho mas económico que un enlace WAN DEDICADO, aumentando la utilidad en las empresas, por otro lado la posibilidad de que mis datos viajen encriptados y seguros, con una buena calidad y velocidad. Otro punto de importancia es la factibilidad con que el canal VPNs este disponible así los puntos remotos se encuentren en distintos puntos geográficos como por ejemplo en distintos países o continentes.

CAPITULO 2

REDES PRIVADAS VIRTUALES

INTRODUCCION

A medida que ha pasado el tiempo las compañías han querido que las redes LAN trasciendan más allá del ámbito de la oficina e incluyeran a los trabajadores y centros de información de otros edificios, ciudades, estados o incluso otros países, para conseguir esto usualmente tenían que invertir en hardware y servicios de telecomunicaciones costosos para crear redes amplias de servicio, WAN. De allí la idea de desarrollar nuevos servicios que mantengan las mismas características de seguridad pero que reduzcan los costos de comunicación aumentando así la utilidad de las empresas que las implementaran.

2.1 QUE ES UNA VPN.

Una VPN es una RED PRIVADA VIRTUAL (Virtual Private Network), conocida también como transporte o acceso a una red privada mediante el uso de una infraestructura pública como lo es el Internet. Para continuar con la descripción de la definición vamos a empezar definiendo los términos que abarcan la frase RED PRIVADA VIRTUAL. Así tenemos:

RED: una infraestructura a través de la cual las computadoras se comunican.

PRIVADA: utiliza criptografía para hacer que la información sea confidencial.

VIRTUAL (no necesita hardware independiente, utiliza como medio de enlace una red pública como el INTERNET).

Es decir una VPN es una red que extiende o transporta a través del encapsulado y cifrado de datos entre diferentes puntos remotos, estos datos encriptados viajan a través de infraestructuras públicas de transporte. Los paquetes de datos de la red privada viajan dentro del túnel definido en la red pública.

Este método permite enlazar dos o más redes simulando una única red privada permitiendo así la comunicación entre computadoras como si fuera punto a punto.

También un usuario remoto se puede conectar individualmente a una LAN utilizando una conexión VPN, y de esta manera utilizar aplicaciones, enviar datos, etc. de manera segura.

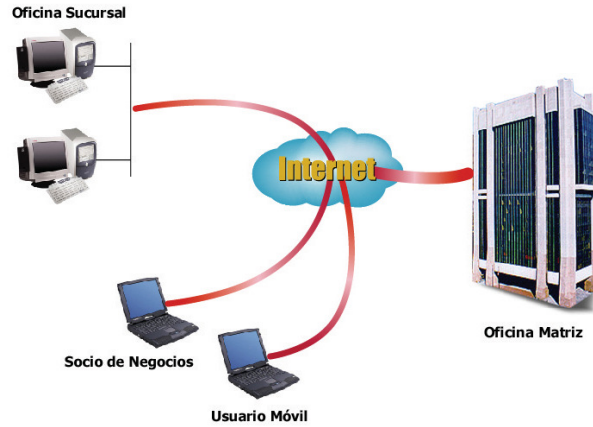


Fig. 2.1. Topología de Red VPN (a)

Fuente: www.gta.com/options/

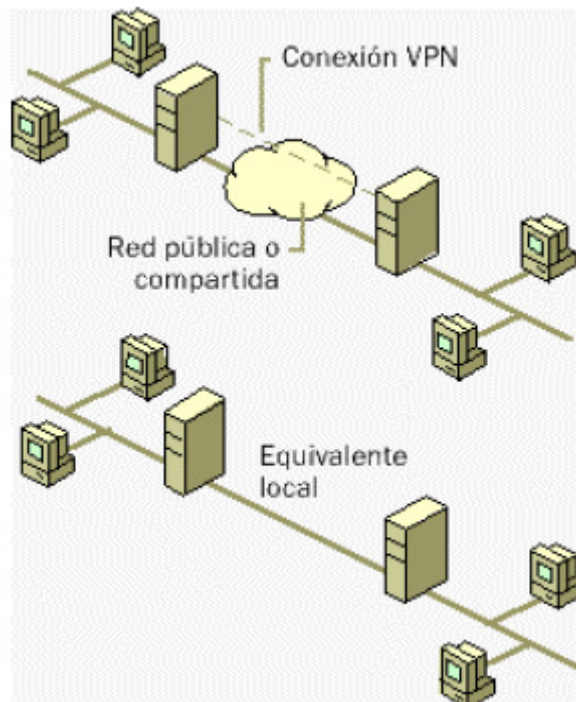


Fig. 2.1 Topología de Red VPN y su equivalente con la red LOCAL. (b)

Fuente: <http://www.scribd.com/doc/3500505/VPN>

2.1.2 ELEMENTOS DE UNA VPN

La forma de comunicación entre las partes de la red privada a través de la red pública se hace estableciendo túneles virtuales entre dos puntos para los cuales se negocian esquemas de encriptación y autenticación que aseguran la confidencialidad e integridad de los datos transmitidos utilizando la red pública, así tenemos:

a. Tecnología de Túneles.

La tecnología de túneles (“Tunneling”) es un modo de transferir datos en la que se encapsula un tipo de paquetes de datos dentro del paquete de datos de algún protocolo, no necesariamente diferente al del paquete original. Al llegar al destino, el paquete original es desempaquetado volviendo así a su estado original. En el traslado a través de Internet, los paquetes viajan encriptados.

b. Autenticación.

Estas técnicas aseguran a los participantes de la VPN que se están intercambiando información con el usuario o dispositivo correcto. Esta autenticación parecida a un sistema como nombre de usuario y contraseña, pero con necesidades mayores de aseguramiento de validación de identidades. La mayoría de los sistemas de autenticación usados en VPN están basados en un sistema de claves compartidas.

La autenticación es llevada a cabo generalmente al inicio de una sesión, y luego aleatoriamente durante el curso de la misma, para asegurar que no haya algún tercer participante que se haya intrometido en la conversación. La autenticación también puede ser usada para asegurar la integridad de los datos. Los datos son procesados con un algoritmo de hashing para derivar un valor incluido en el mensaje como checksum. Cualquier desviación en el checksum indica que los datos fueron corruptos en la transmisión o interceptados y modificados en el camino.

c. Encriptación.

Todas las VPNs tienen algún tipo de tecnología de encriptación, que esencialmente empaqueta los datos en un paquete seguro. La encriptación es considerada tan esencial como la autenticación, ya que protege los datos transportados de no poder ser vistos y entendidos en el viaje de un extremo a otro de la conexión. Existen dos tipos de técnicas de encriptación que se usan en las VPN: encriptación de clave secreta, o privada, y encriptación de clave pública.

En la encriptación de clave secreta, se utiliza una contraseña secreta conocida por todos los participantes que necesitan acceso a la información encriptada. Dicha contraseña se utiliza tanto para encriptar como para desencriptar la información. Este tipo de encriptación posee el problema que, como la contraseña es compartida por todos los participantes y debe mantenerse secreta, al ser revelada, debe ser cambiada y distribuida a los participantes, con lo cual se puede crear de esta manera algún problema de seguridad.

La encriptación de clave pública implica la utilización de dos claves, una pública y una secreta. La primera es enviada a los demás participantes. Al encriptar, se usa la clave privada propia y la clave pública del otro participante de la conversación. Al recibir la información, ésta es desencriptada usando su propia clave privada y la pública del generador de la información.

En las VPNs, la encriptación debe ser realizada en tiempo real. Por eso, los flujos encriptados a través de una red son encriptados utilizando encriptación de clave secreta con claves que son solamente buenas para sesiones de flujo.

2.2 ESTRUCTURA DE LAS VPNS.

Una VPN esta estructurada en:

Un servidor VPN. Ordenador que acepta conexiones VPN de clientes VPN.

Un cliente VPN. Un ordenador que inicia conexiones VPN a un servidor VPN. Puede ser un enrutador o un ordenador individual.

Un túnel, aquella porción de la conexión en que los datos están encapsulados. Los datos no tienen porque estar obligatoriamente cifrados.

Protocolos estándares de comunicación utilizados para gestionar el túnel y encapsular los datos privados, (tunneling protocolos).

Red de tránsito. Es la red pública o compartida a través de la que circulan los datos. Puede tratarse de Internet o de una intranet basada en IP privada



Fig 2.2 Estructura de VPN

Fuente: www.gta.com/options/

2.3 TIPOS DE VPN.

2.3.1 VPNS DE HARDWARE

Una VPN de hardware es una red privada virtual basada en un único dispositivo. Este dispositivo, que contiene un procesador dedicado y que además gestiona la autenticación y el cifrado.

VPNs de Hardware ofrecen una serie de ventajas sobre las de Software, además de un aumento de la seguridad, hardware VPN proporciona equilibrio de carga y la capacidad para manejar grandes cargas cliente. Administración a través de una interfaz de navegador Web.

La principal desventaja de un hardware VPN sobre un software VPN, es su alto costo, hardware VPN son una opción más realista para las grandes empresas que para las pequeñas empresas o sucursales. Varios proveedores ofrecen dispositivos que pueden funcionar como hardware VPN.

2.3.2 VPNS DE SOFTWARE.

Teniendo en cuenta los altos precios en las VPNs de hardware y la poca flexibilidad al tener que estar sujetos a un fabricante de tecnología nacen las VPNs de software libre para la implementación de VPNs usando distintos protocolos de encriptación. Las principales características son:

Multiplataforma: Tanto el servidor como el cliente pueden ejecutarse bajo GNU/Linux, Windows XP/2000 o superior, Mac OS/X, derivados de BSD, entre otros sistemas operativos. También existe un cliente para Pocket PC.

Gratuito: Puede obtenerse y utilizarse gratuitamente en VPNs de cualquier tamaño.

Flexibilidad: Puede ser utilizado en redes complejas sin mayor impacto en su configuración. Soporta adecuadamente el uso de direcciones IP dinámicas, NAT, *firewall* y hasta permite el establecimiento de las conexiones a través de *proxys HTTP*.

2.4 DIAGRAMAS DE VPNS

Hay varias posibilidades de conexiones VPN, esto será definido según los requerimientos de la organización, por eso es aconsejable hacer un buen relevamiento a fin de obtener datos como por ejemplo si lo que se desea enlazar son dos o mas redes, o si solo se conectaran usuarios remotos. Las posibilidades son:

2.4.1 DE CLIENTE A SERVIDOR.

Un usuario remoto que solo necesita servicios o aplicaciones que corren en el mismo servidor VPN.

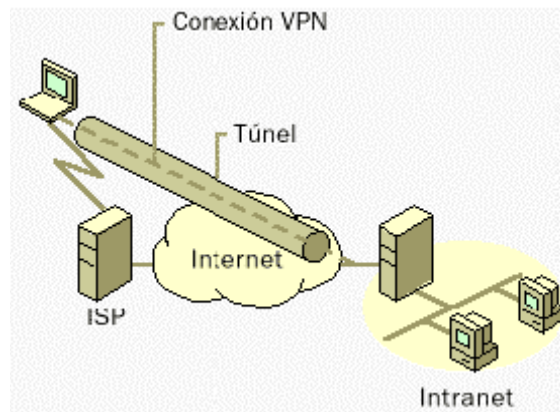


Fig 2.4.1 Diagrama VPN Cliente – Servidor

Fuente: www.gta.com/options/

2.4.2 DE CLIENTE A RED INTERNA.

Un usuario remoto que utilizara servicios o aplicaciones que se encuentran en uno o más equipos dentro de la red interna.



2.4.2 Diagrama VPN Cliente Red Interna

Fuente: <http://www.scribd.com/doc/3500505/VPN>

2.4.3 DE RED INTERNA A RED INTERNA.

Esta forma supone la posibilidad de unir dos intranets a través de dos enrutadores, el servidor VPN en una de las intranets y el cliente VPN en la otra.

Aquí entran en juego el mantenimiento de tablas de ruteo y enmascaramiento.

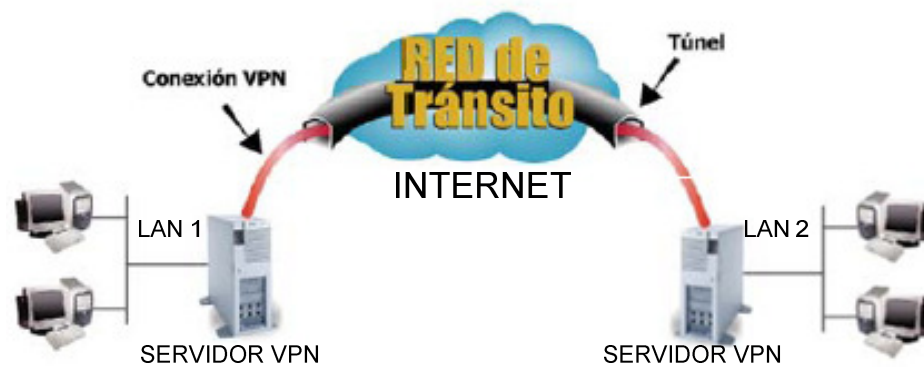


Fig 2.4.3 Diagrama VPN Red Interna – Red Interna

Fuente: www.gta.com/options/

2.5 PROTOCOLOS DE ENCRIPCIÓN UTILIZADOS EN LAS VPNS.

Las soluciones de VPN pueden ser implementadas a diferentes niveles del modelo OSI de red.

2.5.1 IMPLEMENTACIONES DE CAPA 2 - ENLACE

El encapsulamiento a este nivel ofrece ciertas ventajas ya que permite transferencias sobre protocolos no-IP, como por ejemplo IPX4 de Netware Systems. Teóricamente, las tecnologías implementadas en capa 2 pueden tunelizar cualquier tipo de paquetes y en la mayoría de los casos lo que se hace es establecer un dispositivo virtual PPP5 con el cual se establece la conexión con el otro lado del túnel.

Algunos ejemplos de estas tecnologías:

PPTP: Point to Point Tunneling Protocol. Desarrollado por Microsoft, es una extensión de PPP.

Su principal desventaja es que solo puede establecer un túnel por vez entre pares.

L2F: Layer 2 Forwarding. Desarrollado por la empresa Cisco principalmente, ofrece mejores posibilidades que PPTP principalmente en el uso de conexiones simultáneas.

L2TP: Layer 2 Tunneling Protocol. Usado por Cisco y otras fabricantes, se ha convertido en estándar de la industria y combina las ventajas de PPTP y L2F y además eliminando las desventajas. Dado que esta solución no ofrece mecanismos de seguridad, para su uso deberá ser combinada con otros mecanismos generalmente implementados en capa 3 del modelo OSI.

2.5.2 IMPLEMENTACIONES DE CAPA 3 - RED

IPsec es la tecnología más aceptada en este punto y fue desarrollada como un estándar de seguridad de Internet en capa 3. IPsec se puede utilizar para encapsular cualquier tráfico de capa 3. Su principal ventaja es que puede ser usado prácticamente en cualquier plataforma existiendo una gran variedad de soluciones tanto de software como de hardware.

Existen dos métodos principales usados por IPsec:

Modo Túnel. Todos los paquetes IP son encapsulados en un nuevo paquete y enviados a través del túnel siendo desempaquetados en el otro extremo y posteriormente dirigidos a su destinatario final. En este modo, se protegen las direcciones IP de emisor y receptor así como el resto de elementos de los paquetes.

Modo Transporte. Solo la carga útil (payload) de la sección de datos es cifrada y encapsulada. La sobrecarga entonces, es sensiblemente menor que en el caso anterior, pero se exponen los paquetes a posibles atacantes que podrán ver quien se está comunicando con quien.

2.5.3 IMPLEMENTACIONES DE CAPA 7 - APLICACIÓN

También es posible establecer túneles en la capa de aplicación y de hecho son ampliamente utilizados hoy en día siendo algunas aproximaciones soluciones como SSL6 y TLS7. El usuario accede a la VPN de la organización a través de un browser iniciando la conexión en un sitio web seguro (HTTPS-Secured website).

Además, existen otros productos como SSL-Explorer y otros que ofrecen una combinación de gran flexibilidad, seguridad fuerte y facilidad de configuración. La seguridad es lograda mediante cifrado del tráfico usando mecanismos SSL/TLS, los cuales han probado ser muy seguros y están siendo constantemente sometidos a mejoras.

2.6 FUNCIONAMIENTO SSL / TLS

2.6.1 Autenticación SSL / TLS

SSL/TLS tiene soporte interno para la autenticación de host. Esta autenticación se lleva a cabo de forma que se asegura de que un atacante no puede leer o manipular los datos que estamos transmitiendo y que los extremos son los que esperamos. Esta autenticación no es obligatoria, pero es una buena idea. Para poder autenticar un cliente o un servidor, deben tener dos cosas. La primera es una par de claves pública-privada, es decir, clave RSA o DSA. La segunda es un certificado, que es una versión firmada de la clave pública RSA o DSA. Este certificado es, esencialmente, una porción de datos que dice “Yo, el firmante, prometo que este par de claves pertenecen al propietario”. Los certificados utilizados por SSL/TLS son estructuras X509 y, para nuestros propósitos, no necesitamos saber que significa eso.

En una transacción SSL/TLS estándar, el cliente siempre autenticará el certificado del servidor, sin embargo, lo contrario puede no ser cierto. Por ejemplo, las transacciones TTPS rara vez requieren que el cliente (el navegador web) tenga un certificado para la autenticación.

Para asegurarnos de que estamos pidiendo el máximo nivel de seguridad, requeriremos la validación de certificados tanto del servidor como del cliente. Sin embargo, los certificados pueden firmarse de varias maneras y depende de nosotros determinar qué certificado queremos utilizar. Hay dos métodos principales de firmado de certificados: los certificados de terceros y los certificados autofirmados.

Certificados de Terceros

El tipo de certificado mas utilizado es aquel en el que el administrador tiene su clave publica firmada por una tercera parte de confianza. A estas terceras partes las llamamos CA (Certificate Authorities, Autoridades de Certificados).

Nuestro navegador web contiene una gran lista de CA públicas. Cuando conectamos con un sitio web, se comprueba el certificado presentado por el servidor; si su clave pública está firmada por una de estas CA de confianza, se permite la conexión. Si el certificado no esta correctamente firmado por una de estas CA, normalmente se presenta un cuadro de diálogo de aviso y se nos pregunta si queremos continuar.

Normalmente, estas CA son grandes empresas (como Thawte o Equifax) que verifican que la empresa que pide el certificado tiene derecho al certificado pedido. Esto se lleva a cabo fuera de banda a través de varios métodos, como la verificación de la información de nuestra empresa en bases de datos o la verificación de nuestros números de teléfono y direcciones, la propiedad de nuestro nombre de dominio y otras cosas por las que cobran mucho dinero. Después de determinar que estamos legitimados, firmarán nuestra clave pública y nos devolverán el certificado.

La obtención de certificados de terceros requiere tiempo (generalmente una o dos semanas) y rellenar mucha información. También requiere dinero. Un certificado de sitio web estándar cuesta unos 350 dólares.

Para nuestros propósitos, no hay razón por la que no podamos utilizar los certificados autofirmados y ahorrarnos el problema y el coste de los de terceros.

Certificados Autofirmados

Cualquier clave pública-privada puede crear un certificado. No hay nada especial en las claves de una CA excepto el hecho de que están preinstaladas en nuestro navegador web.

Un método común de evitar la obtención de un certificado “oficial” de una de las CA es firmar nuestra propia clave. A esto lo llamamos certificado autofirmado.

En nuestra configuración VPN, crearemos un par de claves pública-privada para nuestro cliente y para nuestro servidor y generaremos certificados autofirmados para cada uno de ellos. El cliente autenticará el servidor comprobando una copia local del certificado del servidor, y viceversa.

Si fuéramos a utilizar un certificado firmado por un tercero, instalaríamos una copia local de la clave de la CA en cada uno de los hosts en lugar de la copia del certificado auto firmado. Al verificar la integridad del certificado autofirmado al instalarlo, realmente no hay diferencias de seguridad entre la utilización de un certificado autofirmado o unos de terceros, y elegiremos la ruta mas barata y rápida.

2.6.2 Como establecer una conexión SSL/TLS

Todas las conexiones SSL/TLS comienzan con un intercambio de señales que permite que los dos equipos se comuniquen de forma segura. En una nutshell, el intercambio de señales SSL/TLS tendría esta apariencia:

El cliente inicia la solicitud de conexión con un paquete “hello 1”. El servidor responde con un “hello 1”. Estos mensajes de saludo establecen la versión del protocolo SSL/TLS, el grupo de cifrados que soportan y un bit de datos aleatorios. Después, el servidor envía al cliente su certificado y el cliente lo verifica. Si lo desea, el servidor solicitará un certificado al cliente y lo verificará.

Entonces, los dos equipos utilizan changecipher spec (un componente de SSL independiente del proceso de intercambio de señales) para finalizar la negociación de los cifrados que se utilizarán en esta sesión y para llegar a un acuerdo sobre la clave a usar. Esta clave esta protegida de ojos curiosos al estar cifrada con la clave publica del igual.

Al haber verificado la autenticidad del extremo remoto, haber llegado a un acuerdo sobre el algoritmo de cifrado y haber comunicado de forma segura la clave a utilizar, la negociación SSL/TLS está completa y cada extremo indica el éxito. En este momento, se envían los datos reales de la comunicación encapsulados en la conexión SSL/TLS.

Esto ha sido un resumen breve de la configuración SSL/TLS. Si queremos los detalles de SSL/TLS, podemos leer la RFC-2246 y la especificación de SSL que podemos encontrar en <http://home.netscape.com/eng/ssl3/>. Lo creamos o no, todo el intercambio de señales tiene lugar en solo cuatro paquetes.

Así, al encapsular un protocolo (por ejemplo, HTTP) en SSL/TLS, la conexión tiene esta apariencia:

El cliente conecta con el servidor en el puerto 443 (HTTPS)

El cliente y el servidor negocian y activan SSL/TLS.

HTTP se envía a través de la conexión cifrada.

2.7 VENTAJAS Y DESVENTAJAS DE LAS VPN

2.7.1 VENTAJAS

Cómo reducen gastos.

Con estas redes podemos reducir gastos de varios tipos: costos de telecomunicaciones por el mantenimiento de muchas líneas de acceso costos en la administración del equipo de acceso remoto Las compañías suelen tener contratadas dos tipos de líneas de acceso: unas de alta velocidad de acceso a Internet y otras del tipo Frame Relay o ISDL.

Con las redes VPN sólo necesitamos un tipo de líneas ya que podremos utilizar una red pública IP para transportar todo tipo de datos. Ahorro de gastos operativos. Se permite tener un acceso a una red vía una VPN de manera que la compañía no tiene que preocuparse del mantenimiento y problemas de administración de un banco de módems y servidores de acceso remoto.

Cómo aumenta la seguridad.

Un VPN permite crear un perímetro de seguridad de operación. Incorpora routers y firewalls como base, y por encima utiliza mecanismos de seguridad como son:

Encriptación de datos. Se utilizan varias técnicas: DES, 3DES, RSA

Compresión de datos

Autenticación. El servidor VPN autentica al cliente para asegurarse que tienen los permisos necesarios. Si además el cliente autentica al servidor se protege contra la suplantación de servidores.

Administración distribuida de claves.

Tunneling (tunelado) para establecer las conexiones punto a punto

Acceso desde el exterior controlado por ser acceso remoto a un servidor seguro

Los protocolos empleados en estas redes son: PPTP (tuneleo Punto- Punto), IPsec (Protocolo de Internet de Seguridad) , L2TP (Protocolo de tuneleo de Capa 2), GRE

y SSH (Secure Shell) como recomendado si empleamos la administración distribuida de llaves.

También se utiliza el certificado digital para autenticar servidores, sitios remotos, empleados, socios y clientes, de forma que se garantice que sólo accedan a la organización usuarios autorizados y que cada uno sólo acceda a la información para la que tiene autorización.

Cómo mejoran las comunicaciones.

Las VPN se abren paso a través de la red pública IP o por redes compartidas IP creando una conexión que emula las propiedades de un enlace punto a punto privado. Para el usuario es como si realizase una conexión dentro de una LAN (red de área local).

Para conseguir esto se emplean técnicas de tunelado en las que se crea un túnel que conecte a ambos extremos y por los que se transmite la información. Los datos se encapsulan con una cabecera que contenga la información para su encaminamiento a través de los túneles previa encriptación de los datos. Cuando los datos salen a la red IP su seguridad está garantizada ya que sin la clave de desencriptación no se puede conocer su contenido.

2.7.2 DESVENTAJAS DE LAS VPN

Se deben establecer correctamente las políticas de seguridad y de acceso.

Mayor carga en el cliente VPN porque debe encapsular los paquetes de datos y encriptarlos, esto produce una cierta lentitud en las conexiones.

No se garantiza disponibilidad (NO Internet NO VPN)

Una VPN se considera segura, pero no hay que olvidar que la información sigue viajando por Internet (no seguro y expuestos a ataques)

CAPITULO 3

CONTRUCCION DE UNA RED VPN BAJO LINUX

INTRODUCCION.

Una vez revisado todos los aspectos teóricos en los capítulos anteriores estamos en capacidad en armar un enlace VPN el mismo que servirá para enlazar ya sea redes remotas o usuarios móviles contra sus respectivos servidores de aplicaciones por los accesos o canales de datos públicos como lo es el Internet.

Para empezar la configuración de nuestra VPN vamos a establecer a manera de ejemplo los elementos de la misma, un punto matriz de la empresa que se denominara “A” una sucursal de esa matriz que será “B” y además usuarios móviles que para este ejemplo serán “C”. El departamento informático y de telecomunicaciones de la empresa luego de haber hecho una extensivo análisis en base a las necesidades de la empresa opta que la mejor opción para unir la sucursal “B” contra la matriz es una VPN, teniendo en cuenta que además de la sucursal existirán los usuarios móviles o remotos “C” que consultaran en la Base de Datos información comercial en un servidor de la matriz.

Teniendo un poco más definido los parámetros de crecimiento y acceso de la empresa nos queda definir los segmentos de red que se manejaran en cada uno de los puntos a enlazar.

3.1 DIRECCIONAMIENTO DE RED PARA MATRIZ , SUCURSAL Y CLIENTES O USUARIOS REMOTOS.

“A” utilizara la red privada 192.168.2.0 /24 y la IP publica provista por un proveedor de Internet 100.100.100.1/24. Ver figuras. A-1, A-2, A-3, A-4 que indica la forma de configuración de tarjetas de red en servidor MATRIZ_A, la misma configuración se utilizaría en el servidor servidor SUCURSAL_B. Ver figuras A-5, A-6, A-7. Finalmente host remotos con el direccionamiento que se indica posteriormente.

“B” utilizará la red privada 192.168.3.0/24 y la IP pública provista por un proveedor de Internet 100.100.100.2/24. “C” utilizará las IPs públicas que para nuestro trabajo será 100.100.100.3/24 100.100.100.4/24.

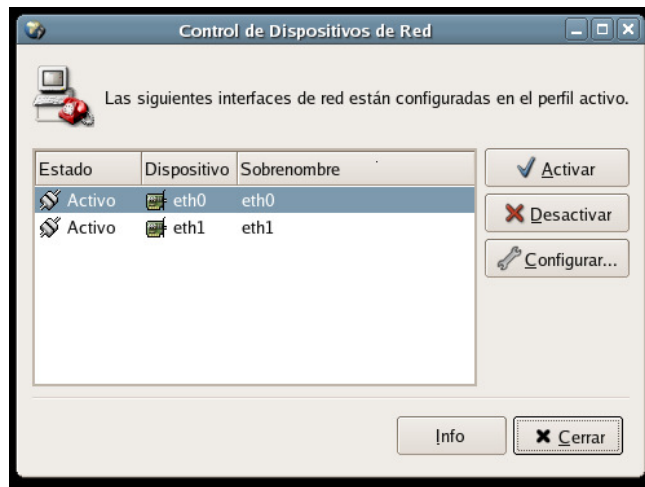


Fig A-1 Control de Dispositivos de Red en el servidor Linux para “A” y “B”

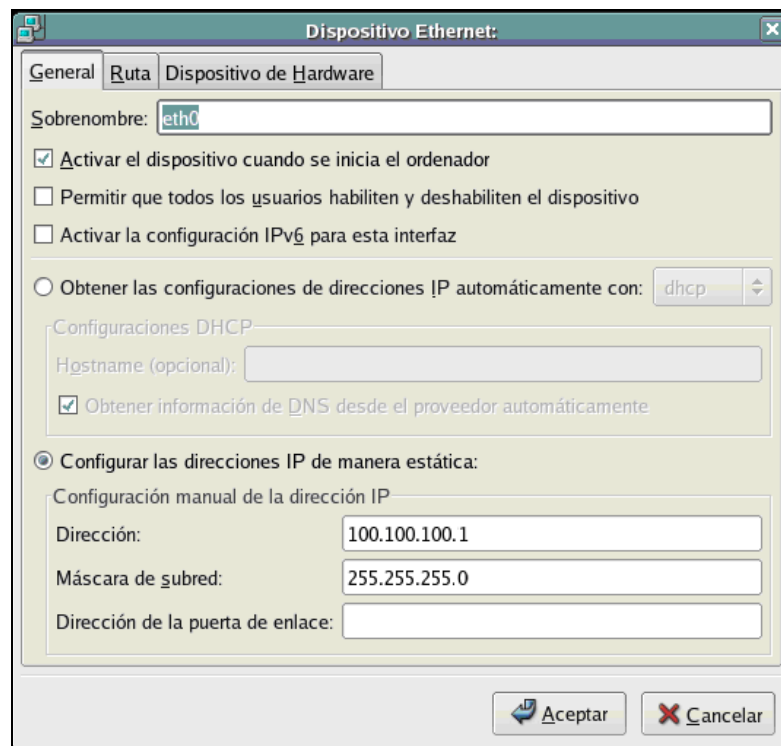


Fig A-2 Configuración de IPs dentro de Tarjeta ETH0 en “A”

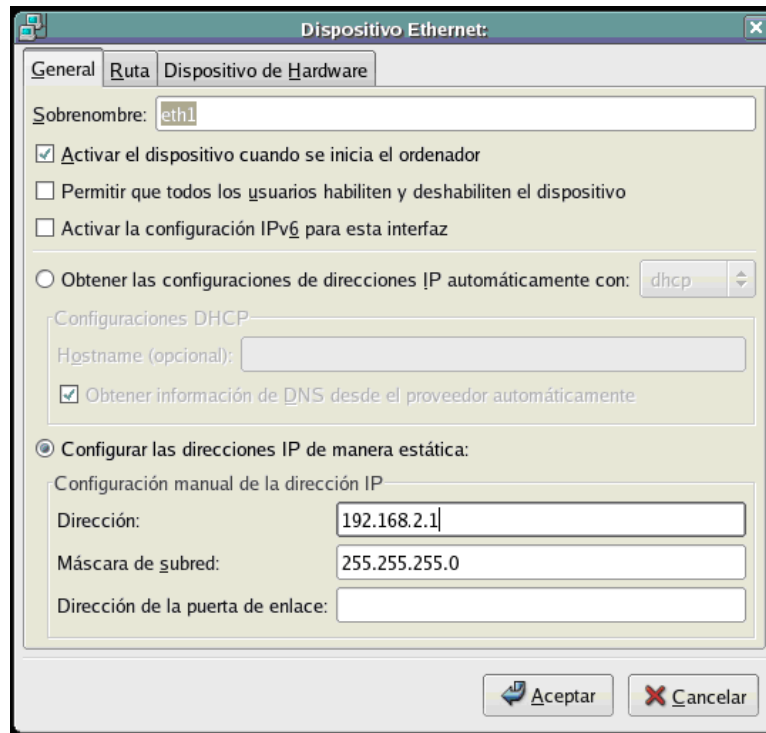


Fig A-3 Configuración de IPs dentro de Tarjeta ETH1 en “A”

```

Archivo Editar Ver Terminal Solapas Ayuda
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:13:8F:CC:43:43
          inet addr:100.100.100.1  Bcast:100.100.100.255  Mask:255.255.255.0
          inet6 addr: fe80::213:8fff:fecc:4343/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:114 errors:0 dropped:0 overruns:0 frame:0
          TX packets:30 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:16954 (16.5 KiB)  TX bytes:2716 (2.6 KiB)
          Interrupt:11 Base address:0xb000

eth1      Link encap:Ethernet  HWaddr 00:02:44:85:FA:86
          inet addr:192.168.2.1  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::202:44ff:fe85:fa86/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:103 errors:0 dropped:0 overruns:0 frame:0
          TX packets:21 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15976 (15.6 KiB)  TX bytes:1857 (1.8 KiB)
          Interrupt:11 Base address:0x6c00

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1412 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1412 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3623324 (3.4 MiB)  TX bytes:3623324 (3.4 MiB)

```

Fig A-4 Resumen de Direccionamiento IP en servidor “A”

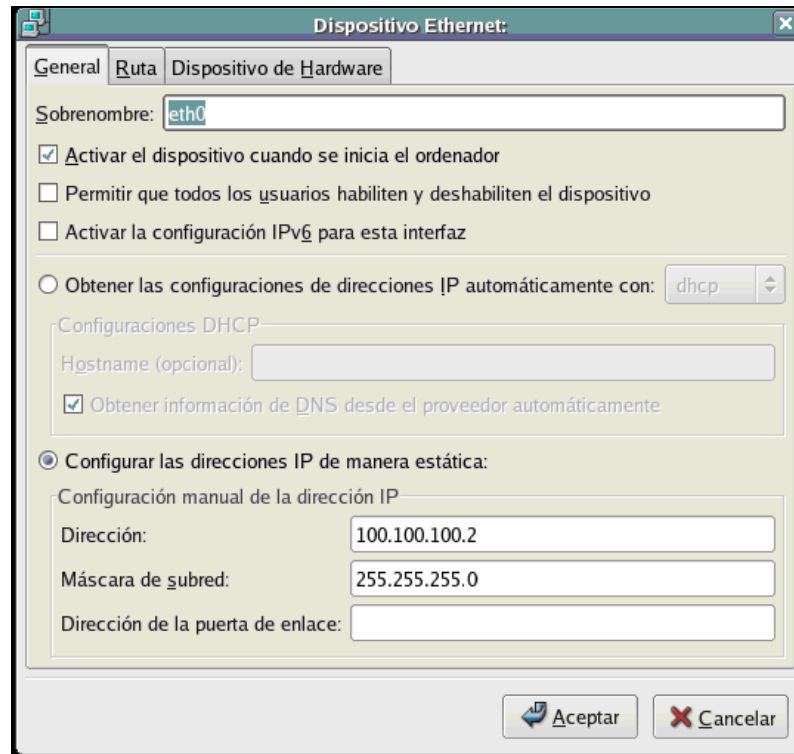


Fig A-5 Configuración de IPs dentro de Tarjeta ETH0 en “B”

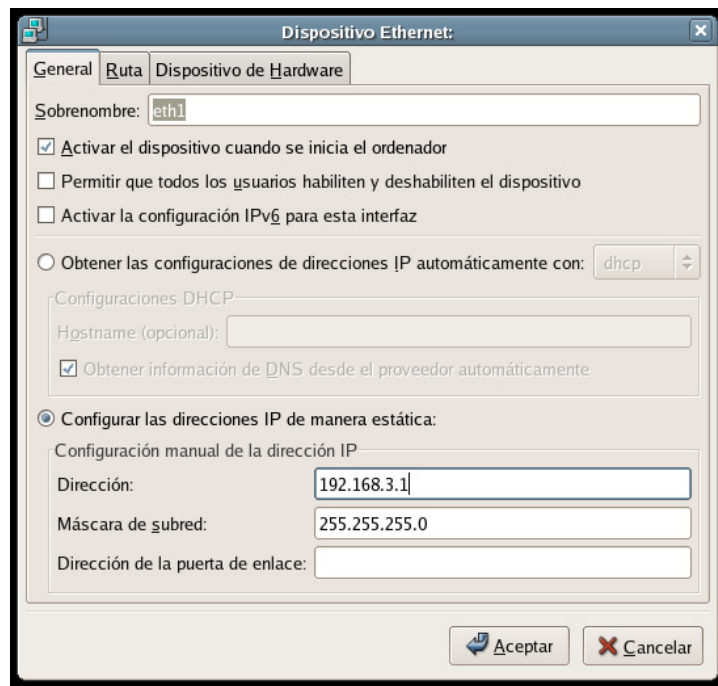


Fig A-6 Configuración de IPs dentro de Tarjeta ETH1 en “B”

```

Archivo Editar Ver Terminal Solapas Ayuda
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet HWaddr 00:13:8F:CC:43:43
          inet addr:100.100.100.2  Bcast:100.100.100.255  Mask:255.255.255.0
          inet6 addr: fe80::213:8fff:fecc:4343/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:114 errors:0 dropped:0 overruns:0 frame:0
          TX packets:30 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:16954 (16.5 KiB)  TX bytes:2716 (2.6 KiB)
          Interrupt:11 Base address:0xb000

eth1      Link encap:Ethernet HWaddr 00:02:44:85:FA:86
          inet addr:192.168.3.1  Bcast:192.168.3.255  Mask:255.255.255.0
          inet6 addr: fe80::202:44ff:fe85:fa86/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:103 errors:0 dropped:0 overruns:0 frame:0
          TX packets:21 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15976 (15.6 KiB)  TX bytes:1857 (1.8 KiB)
          Interrupt:11 Base address:0x6c00

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1412 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1412 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3623324 (3.4 MiB)  TX bytes:3623324 (3.4 MiB)

```

Fig A-7 Resumen de Direccionamiento IP en servidor "B"

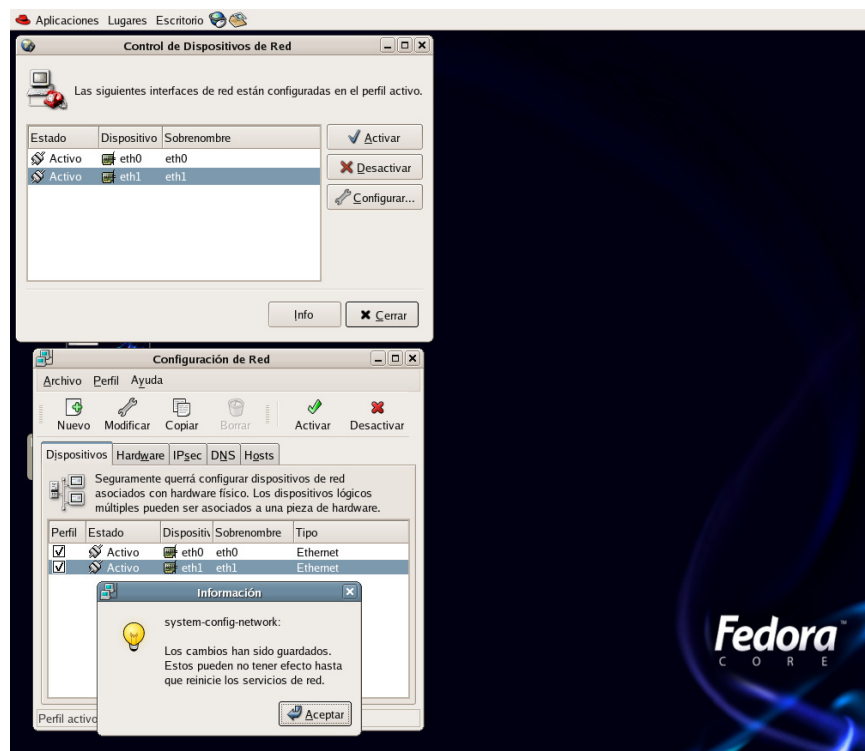


Fig A-8 Servicios de Red Configurados tanto para servidor "A" y "B"

3.2 DIRECCIONAMIENTO IP PARA VPN.

Las IPS publicas o Gateways de la VPN entre A y B sean establecidos con 100.100.100.1/24 y 100.100.100.2/24 respectivamente , las subinterfaces WAN o interfaces virtuales en los Gateways VPNs para unir el canal VPN red-red definido por MATRIZ_A con SUCURSAL_B serán en “A”10.10.10.1/30 y 10.10.10.2/30 en “B” respectivamente. Por ultimo queda por definir la interfaz virtual que se generara también en MATRIZ_A para las conexiones remotas “C” la misma que se ha definido con 192.169.1.1/24.

Resumiendo todo detallado anteriormente tenemos. Ver figura 3.1

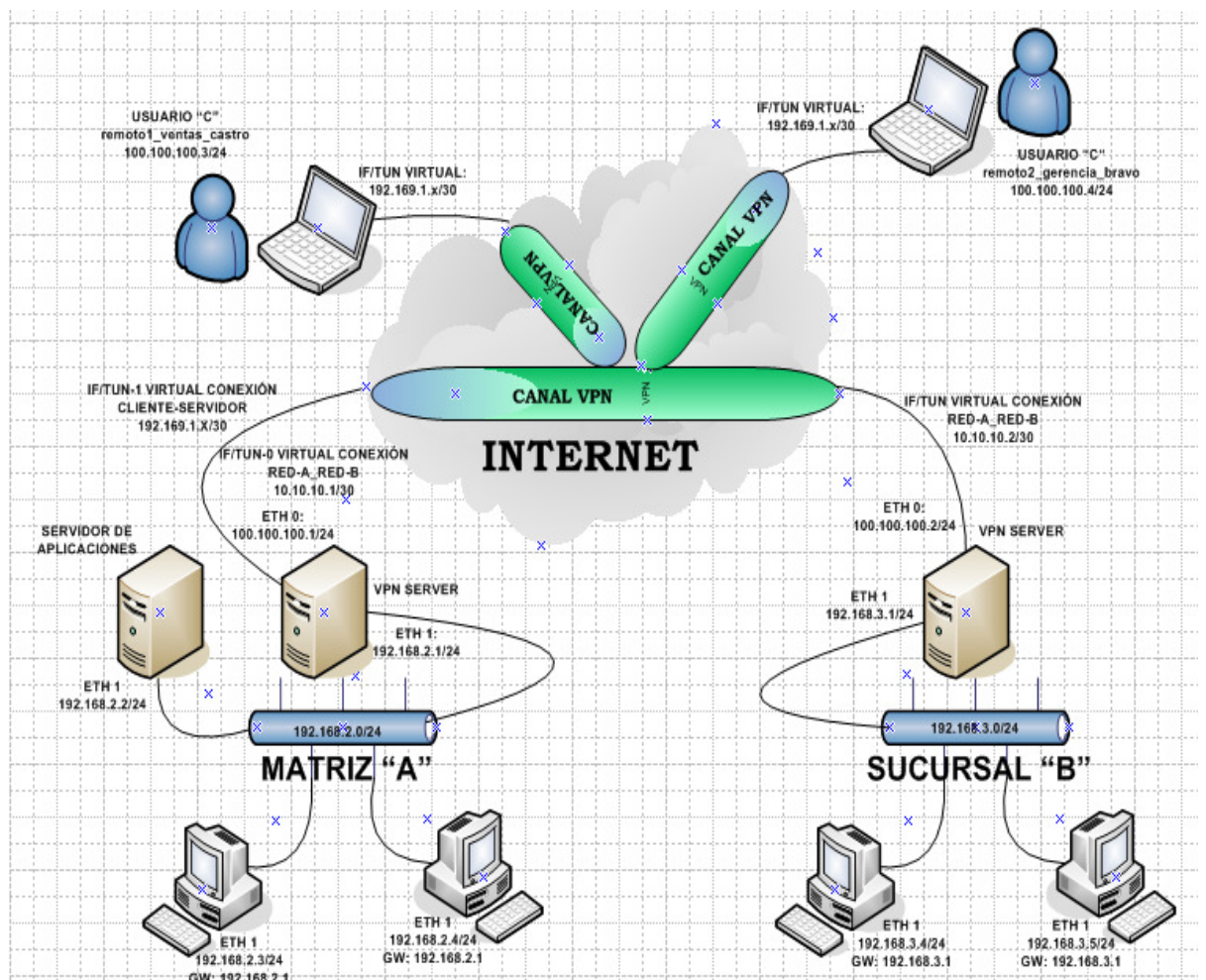


Fig 3.1 Diagrama de Red para conexión VPN entre MATRIZ-SUCURSAL y usuarios remotos.

3.3 INSTALACION, CONFIGURACION Y HABILITACION DE LOS SERVIDORES EN LINUX

Nuestro tema de Tesis trata netamente el enlace de los puntos “A” y “B” y además los usuarios móviles al servidor de datos en “A” claramente definiendo los diagramas de VPNs involucrados y anteriormente explicados como son el tipo RED-RED al unir la red de “A” con la red de “B”, el tipo HOST RED y HOST HOST al usarlo para enlazar los usuarios móviles en primera instancia contra el servidor de datos y luego para compartir recursos o archivos alojados en la red de “A”. y red “B”

Tomando en cuenta todas las recomendaciones y aclaradas en el marco teórico vamos a definir la continuación del diseño en:

3.3.1._Montaje del Hardware para nuestros Servers VPN

3.3.2._Elección, Instalación y Configuración del software para configuración de nuestro Server VPN.

3.3.1._Montaje del Hardware para nuestro Server VPN

Nuestro Server VPN para esta caso será una PC normal con características de velocidad y proceso de datos recomendadas, Procesador Pentium III, Memoria 512k DD 40Gb.

Un sistema operativo sobre el cual se configurara la VPN para nuestro diseño utilizaremos distribución FEDORA CORE 4 de la plataforma LINUX, dicha PC tendrá que tener instalada 2 tarjetas de RED, la 1 tarjeta servirá para la de conexión al Internet y la otra hacia las red Interna. Resumiendo tenemos un Server VPN en “A” uno en “B” para la conexión RED-RED en total 2, hay que tener en cuenta que el mismo Server en “A” será también el mismo que aceptara conexiones remotas de los usuarios móviles “C”

3.3.2 Elección e Instalación del software para configuración de nuestro Servidor VPN.

Para realizar este caso utilizaremos una aplicación freeware de fácil acceso y configuración. El software es OPEN VPN versión 2.0.1. OpenVPN, es una solución de conectividad basada en software: SSL (Secure Sockets Layer) creado por James Yonan en el año 2001 y que ha estado siendo mejorado desde entonces.

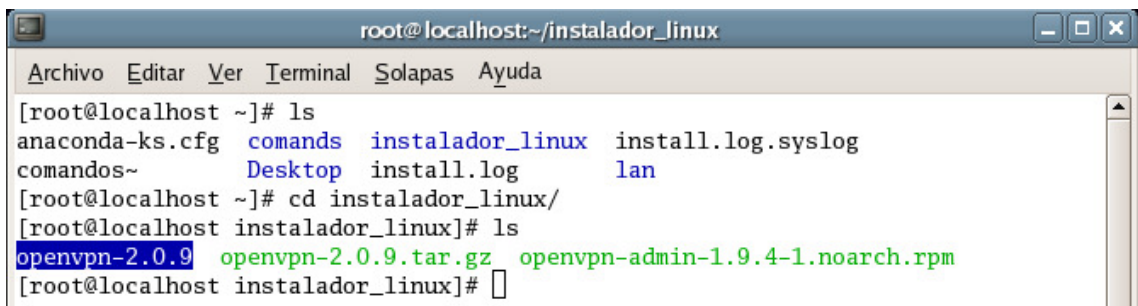
Ninguna otra solución ofrece una mezcla semejante de seguridad a nivel empresarial, seguridad y riqueza de características.

Es una solución multiplataforma que ha simplificado mucho la configuración de VPN's dejando atrás los tiempos de otras soluciones difíciles de configurar como IPsec y haciéndola más accesible para las personas.

3.4 INSTALACION Y CONFIGURACION DE OPEN-VPN EN NUESTROS SERVIDORES LINUX.

Antes de empezar la instalación procedemos a descargar el software desde la web del autor en:

<http://openvpn.net/index.php/downloads.html> con extensión `.tar.gz` y lo guardamos en una carpeta el disco duro, dicho archivo es un paquete que hay que descomprimirlo, una vez realizado eso abrimos una terminal ingresamos al directorio donde guardamos el archivo y listamos con `ls` (comando para listar archivos en linux) para ver si esta generado la carpeta `openvpn-2-0.9` que es en donde se encuentran todos los archivos de compilación. Ver figura 3.2



```

root@localhost:~/instalador_linux
Archivo  Editar  Ver     Terminal  Solapas  Ayuda
[root@localhost ~]# ls
anaconda-ks.cfg  comands  instalador_linux  install.log.syslog
comandos~       Desktop  install.log       lan
[root@localhost ~]# cd instalador_linux/
[root@localhost instalador_linux]# ls
openvpn-2.0.9  openvpn-2.0.9.tar.gz  openvpn-admin-1.9.4-1.noarch.rpm
[root@localhost instalador_linux]#

```

Fig 3.2

Empezamos la configuración moviendo la carpeta descomprimida **openvpn-2.0.9** al directorio **etc**, el comando a utilizar **cd /etc** y luego listamos con **ls** para verificar que el directorio `openvpn-2.0.9` se haya cambiado. Ver Fig 3.3

```
[root@localhost etc]# ls
4Suite                cron.daily            gnome-vfs-2.0         inittab              madev.d
a2ps.cfg              cron.deny             gnome-vfs-mime-magic inputrc              man.config
a2ps-site.cfg        cron.hourly          gnopernicus-1.0     iproute2            mgetty+sendfax
acpi                  cron.monthly         gpm-root.conf       isdn                 mime.types
adjtime              crontab              gre.d                issue                minicom.users
alchemist             cron.weekly          group                issue.net            modprobe.conf
aliases              csh.cshrc            grub.conf            java                 modprobe.conf~
aliases.db           csh.login            gshadow              jwhois.conf         modprobe.conf.dist
alsa                  cups                  gshadow-             kermit               motd
alternatives          dbus-1               gshadow-             krb5.conf            mtab
anacrontab           default              gssapi_mech.conf    krb.conf             mtools.conf
asound.conf          dev.d                gtk                  krb.realms           Muttrc
asound.state         DIR_COLORS           gtk-2.0              ldap.conf            Muttrc.local
at.deny              DIR_COLORS.xterm    hal                  ld.so.cache          named.conf
auditd.conf          dumpdates            host.conf            ld.so.conf           netplug
audit.rules          environment          hosts                ld.so.conf.d         netplug.d
auto.master          esd.conf             hosts.allow          lftp.conf            nscd.conf
auto.misc            exports              hosts.deny           libuser.conf         nsswitch.conf
auto.net             fb.modes             hotplug              localtime            ntp
auto.smb             fdprm                hotplug.d            log.d                 ntp.conf
bashrc               fedora-release      howl                 login.defs            openldap
blkid.tab            gnome-vfs-2.0       httpd                logrotate.conf       openvpn-2.0.9
```

Fig 3.3

Ingresamos al directorio openvpn-2.0.9 con el comando cd y listamos con ls

Ver Fig 3.4(a) y 3.4(b)

```
cipe                  gaim                 lmrc                  mail                  passwd
cpuspeed.conf        gconf                init.d                mailcap               passwd-
cron.d                ghostscript          initlog.conf          mail.rc               passwd.OLD
[root@localhost etc]# cd openvpn-2.0.9/
```

Fig 3.4(a)

```
[root@localhost openvpn-2.0.9]# ls
acinclude.m4          COPYING              forward.o            list.c                mroute.c              occ-inline.h          ping.h
aclocal.m4            COPYRIGHT.GPL        fragment.c          list.h                mroute.h              occ.o                 ping-inline.h
AUTHORS               cryptoapi.c          fragment.h          list.o                mroute.o              openvpn                ping.o
base64.c              cryptoapi.h          fragment.o          lzo.c                 mss.c                 openvpn.8             plugin
base64.h              crypto.c             gentoo              lzo.h                 mss.h                 openvpn.c             plugin.c
base64.o              crypto.h             gremlin.c           lzo.o                 mss.o                 openvpn.h             plugin.h
basic.h               crypto.o             gremlin.h           Makefile              mtcp.c                openvpn.o             plugin.o
buffer.c              debug                gremlin.o           Makefile.am           mtcp.h                openvpn-plugin.h     pool.c
buffer.h              depcomp              helper.c            Makefile.in           mtcp.o                openvpn.spec          pool.h
buffer.o              doclean              helper.h            makefile.w32          mtu.c                  openvpn.spec.in       pool.o
ChangeLog             easy-rsa             helper.o            manage.c               mtu.h                  openvpn-status.log    PORTS
circ_list.h           errlevel.h           init.c              manage.h               mtu.o                  options.c              proto.c
common.h              error.c              init.h              management             mudp.c                 options.h              proto.h
config.guess          error.h              init.o              management             mudp.h                  options.o              proto.o
config.h              error.o              INSTALL             manage.o               mudp.o                  otime.c                proxy.c
config.h.in           event.c              install-sh          mbuf.c                multi.c                 otime.h                proxy.h
config.log            event.h              install-win32       mbuf.h                multi.h                 otime.o                proxy.o
config.status         event.o              INSTALL-win32.txt  mbuf.o                multi.o                 packet_id.c            push.c
config.sub            fdmisc.c             interval.c          memcmp.c               NEWS                    packet_id.h            push.h
configure             fdmisc.h             interval.h          memdbg.h               ntlm.c                  packet_id.o            push.o
configure.ac          fdmisc.o             interval.o          misc.c                  ntlm.h                  perf.c                 README
config-win32.h        forward.c             ipp.txt            misc.h                  ntlm.o                  perf.h                 reliable.c
config-win32.h.in     forward.h             key.txt             misc.o                  occ.c                    perf.o                 reliable.h
contrib               forward-inline.h     key.txt             missing                 occ.h                    ping.c                 reliable.o
```

Fig 3.4(b)

Empezamos la compilación del software ejecutando los scripts: configure, make y make install, utilizamos estos comandos para la compilación ya que nuestro software original era con extensión .tar y son compatibles con la

distribución de LINUX que tenemos. Luego reiniciamos nuestro servidor.

Ver Fig 3.5(a)

```
[root@localhost openvpn-2.0.9]# ls
acinclude.m4          COPYING             forward.o           list.c              mroute.c           occ-inline.h       ping.h
aclocal.m4           COPYRIGHT.GPL      fragment.c         list.h              mroute.h           occ.o              ping-inline.h
AUTHORS              cryptoapi.c        fragment.h         list.o              mroute.o           openvpn            ping.o
base64.c             cryptoapi.h        fragment.o         lzo.c               mss.c              openvpn.8          plugin
base64.h            crypto.c           gentoo             lzo.h               mss.h              openvpn.c          plugin.c
base64.o            crypto.h           gremlin.c         lzo.o               mss.o              openvpn.h          plugin.h
basic.h             crypto.o           gremlin.h         Makefile            mtcp.c             openvpn.o          plugin.o
buffer.c            debug             gremlin.o         Makefile.am        mtcp.h             openvpn-plugin.h  pool.c
buffer.h            depcomp           helper.c           Makefile.in        mtcp.o             openvpn.spec       pool.h
buffer.o            doclean           helper.h           management          mtu.c              openvpn.spec.in   pool.o
Changelog           easy-rsa          helper.o           manage.c            mtu.h              openvpn-status.log PORTS
circ_list.h         errlevel.h        init.c            manage.h            mtu.o              options.c          proto.c
common.h            error.c           init.h            manage.o            mudp.c             options.h          proto.h
config.guess        error.h           init.o            management          mudp.h             options.o          proto.o
config.h            error.o           INSTALL           manage.o            mudp.o             otime.c           proxy.c
config.h.in         event.c           install-sh        mbuf.c             multi.c            otime.h           proxy.h
config.log          event.h           install-win32     mbuf.h             multi.h            otime.o           proxy.o
config.status       event.o           INSTALL-win32.txt mbuf.o             multi.o            packet_id.c       push.c
config.sub          fdmisc.c          integer.h         memcmp.c            NEWS               packet_id.h       push.h
configure           fdmisc.h          interval.c        memdbg.h            ntlm.c             packet_id.o       push.o
configure.ac        fdmisc.o          interval.h        misc.c              ntlm.h             perf.c             README
config-win32.h      forward.c          interval.o        misc.h              ntlm.o             perf.h             reliable.c
config-win32.h.in  forward.h          ipp.txt          misc.o              occ.c               perf.o             reliable.h
contrib            forward-inline.h  key.txt          missing             occ.h               ping.c             reliable.o
[root@localhost openvpn-2.0.9]# ./configure
```

Fig 3.5(a)

Una vez reiniciado entramos nuevamente al directorio openvpn-2.0.9 y observamos que se hayan generado los ejecutables openvpn, el mismo que nos permitirá arrancar la aplicación OPENVPN y generar los tuneles. Ver figura 3.5(b)

```
[root@localhost openvpn-2.0.9]# ls
acinclude.m4          COPYING             forward.o           list.c              mroute.c           occ-inline.h       ping.h
aclocal.m4           COPYRIGHT.GPL      fragment.c         list.h              mroute.h           occ.o              ping-inline.h
AUTHORS              cryptoapi.c        fragment.h         list.o              mroute.o           openvpn            ping.o
base64.c             cryptoapi.h        fragment.o         lzo.c               mss.c              openvpn.8          plugin
base64.h            crypto.c           gentoo             lzo.h               mss.h              openvpn.c          plugin.c
base64.o            crypto.h           gremlin.c         lzo.o               mss.o              openvpn.h          plugin.h
basic.h             crypto.o           gremlin.h         Makefile            mtcp.c             openvpn.o          plugin.o
buffer.c            debug             gremlin.o         Makefile.am        mtcp.h             openvpn-plugin.h  pool.c
buffer.h            depcomp           helper.c           Makefile.in        mtcp.o             openvpn.spec       pool.h
buffer.o            doclean           helper.h           management          mtu.c              openvpn.spec.in   pool.o
Changelog           easy-rsa          helper.o           manage.c            mtu.h              openvpn-status.log PORTS
circ_list.h         errlevel.h        init.c            manage.h            mtu.o              options.c          proto.c
common.h            error.c           init.h            manage.o            mudp.c             options.h          proto.h
config.guess        error.h           init.o            management          mudp.h             options.o          proto.o
config.h            error.o           INSTALL           manage.o            mudp.o             otime.c           proxy.c
config.h.in         event.c           install-sh        mbuf.c             multi.c            otime.h           proxy.h
config.log          event.h           install-win32     mbuf.h             multi.h            otime.o           proxy.o
config.status       event.o           INSTALL-win32.txt mbuf.o             multi.o            packet_id.c       push.c
config.sub          fdmisc.c          integer.h         memcmp.c            NEWS               packet_id.h       push.h
configure           fdmisc.h          interval.c        memdbg.h            ntlm.c             packet_id.o       push.o
configure.ac        fdmisc.o          interval.h        misc.c              ntlm.h             perf.c             README
config-win32.h      forward.c          interval.o        misc.h              ntlm.o             perf.h             reliable.c
config-win32.h.in  forward.h          ipp.txt          misc.o              occ.c               perf.o             reliable.h
-----
contrib            forward-inline.h  key.txt          missing             occ.h               ping.c             reliable.o
[root@localhost openvpn-2.0.9]# ./configure
```

Fig 3.5(b)

Hasta aquí termina lo que sería la descarga, instalación y configuración de nuestro software que nos servirá para realizar conjuntamente con los archivos de configuración que los veremos mas adelante las VPNS que nos planteamos en nuestro tema de tesis, cabe indicar que este procedimiento que hemos realizado es el mismo que tenemos que realizar en nuestros servidores para MATRIZ_A,

SUCURSAL_B y en cada uno de los usuarios remotos “C” como son remoto1_ventas_castro y remoto2_gerencia_bravo.

3.5. CREACION DE CERTIFICADOS.

Conforme a lo explicado en la parte teórica esta VPN es una VPN de capa 4 que utiliza el protocolo SSL/TLS para la encriptación de información dicha información es comparada a través de intercambio de claves conocidas en este caso para SSL/TLS como certificados.

Los certificados en nuestro caso los tenemos que crear en nuestro servidor MATRIZ_A que es la matriz de la empresa que además de enlazar todo la red de la sucursal “B” permite o acepta las conexiones de los usuarios remotos “C”. Ver fig 3.1

3.5.1 Creación de certificados y archivos de configuración OPENVPN en MATRIZ_A

Confirmado lo del punto anterior procedemos a crear los certificados de autenticación los cuales iniciaran y mantendrán el cifrado de datos dentro del túnel el momento que se establezca la comunicación. Para ello entramos al directorio **easy-rsa**. Ver Fig 3.6

```
[root@localhost etc]# cd openvpn-2.0.9/
[root@localhost openvpn-2.0.9]# ls
acinclude.m4          COPYING              forward.o            list.c               mroute.c             occ-inline.h         ping.h
aclocal.m4            COPYRIGHT.GPL        fragment.c           list.h               mroute.h             occ.o                ping-inline.h
AUTHORS               cryptoapi.c          fragment.h           list.o               mroute.o             openvpn              plugin.o
base64.c              cryptoapi.h          fragment.o           lzo.c                mss.c                openvpn.8            plugin.c
base64.h              crypto.c              gentoo               lzo.h                mss.h                openvpn.c            plugin.h
base64.o              crypto.h              gremlin.c           lzo.o                mss.o                openvpn.h            plugin.h
basic.h               crypto.o              gremlin.h           Makefile              mtcp.c               openvpn.o            plugin.o
buffer.c              debug                gremlin.o            Makefile.am           mtcp.h               openvpn-plugin.h     pool.c
buffer.h              depcomp              helper.c             Makefile.in           mtcp.o               openvpn.spec         pool.h
buffer.o              doclean              helper.h             makefile.w32          mtu.c                openvpn.spec.in     pool.o
ChangeLog             easy-rsa             helper.o             makefile.w32-vc      mtu.h                openvpn-status.log   PORTS
circ_list.h           errlevel.h           init.c               manage.c              mtu.o                options.c            proto.c
common.h              error.c              init.h               manage.h              mudp.c               options.h            proto.h
config.guess          error.h              init.o               management            mudp.h               options.o            proto.o
config.h              error.o              INSTALL              manage.o              mudp.o               otime.c              proxy.c
config.h.in           event.c              install-sh           mbuf.c                multi.c               otime.h              proxy.h
config.log            event.h              install-win32        mbuf.h                multi.h               otime.o              proxy.o
config.status         event.o              INSTALL-win32.txt   mbuf.o                NEWS                  packet_id.c          push.c
config.sub            fdmisc.c             integer.h            memcmp.c              ntlm.c               packet_id.h          push.h
configure             fdmisc.h             interval.c           memdbg.h              ntlm.o               packet_id.o          push.o
configure.ac          fdmisc.o             interval.h           misc.c                 ntlm.h               perf.c               README
config-win32.h        forward.c             interval.o           misc.h                 ntlm.o               perf.h               reliable.c
config-win32.h.in    forward.h             ipp.txt             misc.o                 occ.c                 perf.o               reliable.h
contrib              forward-inline.h     key.txt             missing                occ.h                 ping.c               reliable.o
[root@localhost openvpn-2.0.9]# cd easy-rsa/
```

Fig 3.6

Dentro del directorio easy-rsa están todos los scripts que nos permitirán realizar los certificados explicados en los puntos anteriores. Ver Fig 3.7

```
[root@localhost easy-rsa]# ls
2.0 build-dh build-key build-key-pkcs12 build-req clean-all list-crl openssl.cnf revoke-cert sign-req Windows
build-ca build-inter build-key-pass build-key-server build-req-pass keys make-crl README revoke-full vars
[root@localhost easy-rsa]#
```

Fig 3.7

Empezamos con la configuración de las variables, para esto tenemos un archivo que se llama vars, el mismo lo vamos a editar con el editor vi, así tenemos: Ver Fig 3.8

```
[root@localhost openvpn-2.0.9]# cd easy-rsa/
[root@localhost easy-rsa]# ls
2.0 build-dh build-key build-key-pkcs12 build-req clean-all list-crl openssl.cnf revoke-cert sign-req Windows
build-ca build-inter build-key-pass build-key-server build-req-pass keys make-crl README revoke-full vars
[root@localhost easy-rsa]# vi vars
```

Fig 3.8

Dentro del archivo vars, lo que tenemos que hacer es registrar el directorio export D donde se generaran las claves y certificados, además llenamos los parámetros de:

Export KEY_COUNTRY=, Export KEY_PROVINCE=, Export KEY_CITY=,
Export KEY_ORG=, Export KEY_EMAIL=

No dejaremos ninguno de estos espacios en blanco porque no correrá la aplicación el momento que guardemos los cambios con el comando wq dentro del archivo. Ver Fig 3.9

```
Archivo Editar Ver Terminal Solapas Ayuda
# easy-rsa parameter settings
# NOTE: If you installed from an RPM,
# don't edit this file in place in
# /usr/share/openvpn/easy-rsa --
# instead, you should copy the whole
# easy-rsa directory to another location
# (such as /etc/openvpn) so that your
# edits will not be wiped out by a future
# OpenVPN package upgrade.
# This variable should point to
# the top level of the easy-rsa
# tree.
export D='/etc/openvpn-2.0.9/easy-rsa'
# This variable should point to
# the openssl.cnf file included
# with easy-rsa.
export KEY_CONFIG=$D/openssl.cnf
# Edit this variable to point to
# your soon-to-be-created key
# directory.
# WARNING: clean-all will do
# a rm -rf on this directory
# so make sure you define
# it correctly!
export KEY_DIR=$D/keys
# Issue rm -rf warning
echo NOTE: when you run ./clean-all, I will be doing a rm -rf on $KEY_DIR
# Increase this to 2048 if you
# are paranoid. This will slow
# down TLS negotiation performance
# as well as the one-time DH parms
# generation process.
export KEY_SIZE=1024
# These are the default values for fields
# which will be placed in the certificate.
# Don't leave any of these fields blank.
export KEY_COUNTRY=EC
export KEY_PROVINCE=AZ
export KEY_CITY=CCA
export KEY_ORG=MATRIZ_A
export KEY_EMAIL=wimer882@cue.satnet.net
-
-
:wq
```

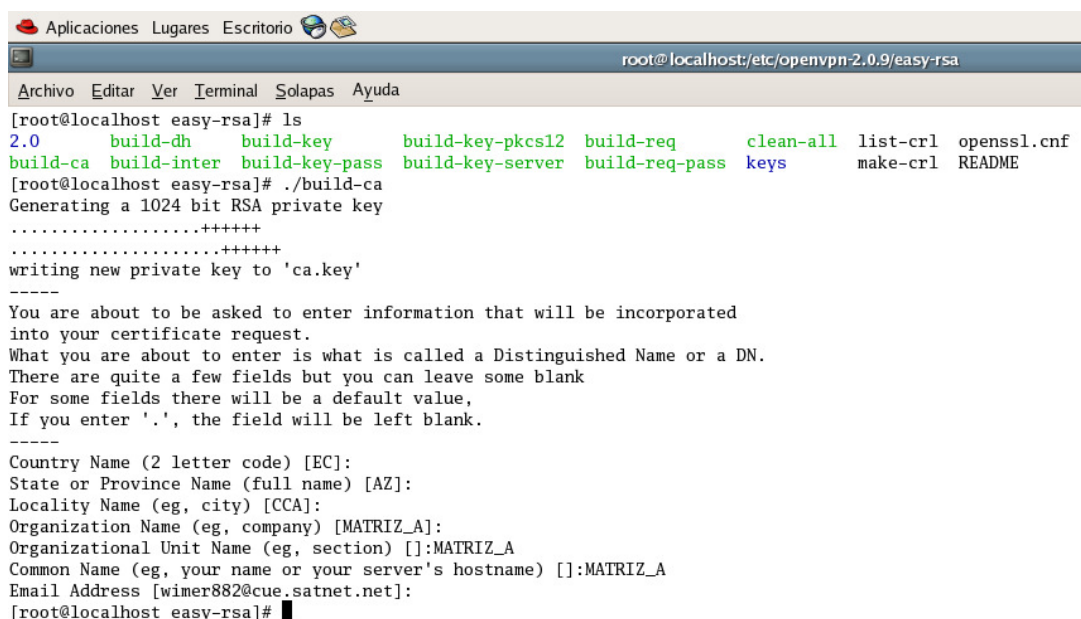
Fig 3.9

Luego de grabado los cambios compilaremos el archivo vars. Ver Fig 3.10

```
[root@localhost easy-rsa]# vi vars
[root@localhost easy-rsa]# . vars
NOTE: when you run ./clean-all, I will be doing a rm -rf on /etc/openssl-2.0.9/easy-rsa/keys
[root@localhost easy-rsa]#
```

Fig 3.10

Una vez configurado el archivo de variables empezaremos a crear los certificados, iniciando construyendo el certificado ca y llenando todos los parámetros. Este certificado ca tiene que ser copiado en todos los host a los que se les va a permitir las conexiones remotas en nuestro caso sería remoto1_ventas_castro y remoto2_gerencia_bravo ver fig 3.11

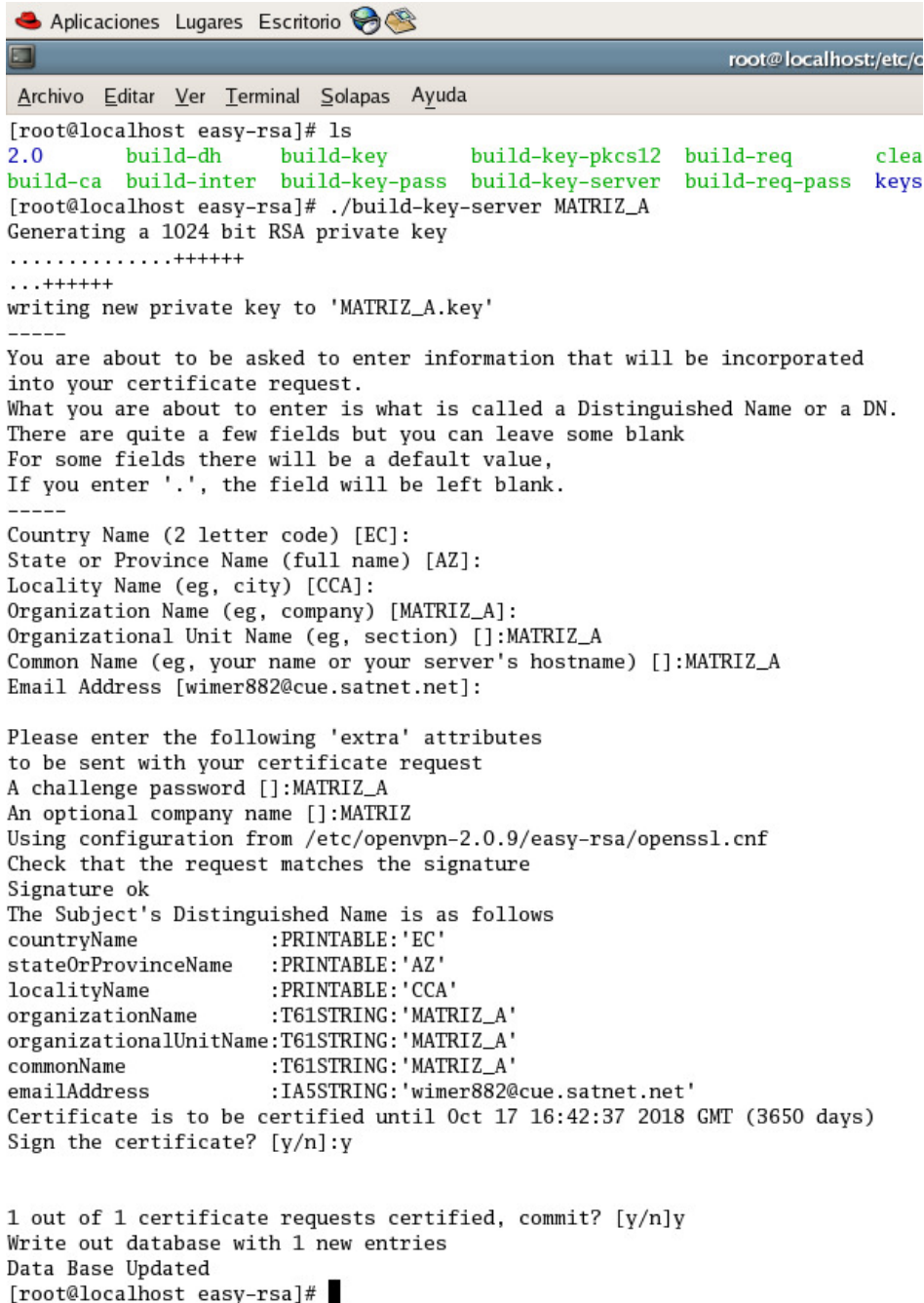


```
Aplicaciones Lugares Escritorio
root@localhost/etc/openssl-2.0.9/easy-rsa
Archivo Editar Ver Terminal Solapas Ayuda
[root@localhost easy-rsa]# ls
2.0      build-dh  build-key  build-key-pkcs12  build-req  clean-all  list-crl  openssl.cnf
build-ca build-inter build-key-pass build-key-server build-req-pass keys  make-crl  README
[root@localhost easy-rsa]# ./build-ca
Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [EC]:
State or Province Name (full name) [AZ]:
Locality Name (eg, city) [CCA]:
Organization Name (eg, company) [MATRIZ_A]:
Organizational Unit Name (eg, section) []:MATRIZ_A
Common Name (eg, your name or your server's hostname) []:MATRIZ_A
Email Address [wimer882@cue.satnet.net]:
[root@localhost easy-rsa]#
```

Fig 3.11

Creación del certificado clave del servidor en este caso para MATRIZ_A.

Ver Fig 3.12



```

Aplicaciones Lugares Escritorio
root@localhost:/etc/c
Archivo Editar Ver Terminal Solapas Ayuda
[root@localhost easy-rsa]# ls
2.0      build-dh      build-key      build-key-pkcs12  build-req      clea
build-ca  build-inter   build-key-pass  build-key-server  build-req-pass  keys
[root@localhost easy-rsa]# ./build-key-server MATRIZ_A
Generating a 1024 bit RSA private key
.....++++++
...++++++
writing new private key to 'MATRIZ_A.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [EC]:
State or Province Name (full name) [AZ]:
Locality Name (eg, city) [CCA]:
Organization Name (eg, company) [MATRIZ_A]:
Organizational Unit Name (eg, section) []:MATRIZ_A
Common Name (eg, your name or your server's hostname) []:MATRIZ_A
Email Address [wimer882@cue.satnet.net]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:MATRIZ_A
An optional company name []:MATRIZ
Using configuration from /etc/openssl-2.0.9/easy-rsa/openssl.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName          :PRINTABLE:'EC'
stateOrProvinceName  :PRINTABLE:'AZ'
localityName         :PRINTABLE:'CCA'
organizationName     :T61STRING:'MATRIZ_A'
organizationalUnitName:T61STRING:'MATRIZ_A'
commonName           :T61STRING:'MATRIZ_A'
emailAddress         :IA5STRING:'wimer882@cue.satnet.net'
Certificate is to be certified until Oct 17 16:42:37 2018 GMT (3650 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
[root@localhost easy-rsa]# █

```

Fig 3.12

Creación de certificados para usuarios remotos en este caso empezaremos con el que hemos denominado remoto1_ventas_castro como se puede apreciar también pertenece a la organización MATRIZ_A , cabe indicar que este certificado tiene que se copiado en el host remoto1 para que se puede logear contra el servidor MATRIZ_A Ver Figura 3.13

```

Archivo  Editar  Ver  Terminal  Solapas  Ayuda
[root@localhost easy-rsa]# ls
2.0      build-dh      build-key      build-key-pkcs12  build-req      clean-all  list-crl  openssl.cnf  re
build-ca  build-inter  build-key-pass  build-key-server  build-req-pass  keys       make-crl  README      re
[root@localhost easy-rsa]# ./build-key remoto1_ventas_castro
Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to 'remoto1_ventas_castro.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [EC]:
State or Province Name (full name) [AZ]:
Locality Name (eg, city) [CCA]:
Organization Name (eg, company) [MATRIZ_A]:
Organizational Unit Name (eg, section) []:MATRIZ_A
Common Name (eg, your name or your server's hostname) []:REMOTO1_VENTAS
Email Address [wimer882@cue.satnet.net]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:REMOTO1_VENTAS
An optional company name []:MATRIZ
Using configuration from /etc/openssl/easy-rsa/openssl.cnf
DEBUG[load_index]: unique_subject = "yes"
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName          :PRINTABLE:'EC'
stateOrProvinceName  :PRINTABLE:'AZ'
localityName         :PRINTABLE:'CCA'
organizationName     :T61STRING:'MATRIZ_A'
organizationalUnitName:T61STRING:'MATRIZ_A'
commonName           :T61STRING:'REMOTO1_VENTAS'
emailAddress         :IA5STRING:'wimer882@cue.satnet.net'
Certificate is to be certified until Oct 17 16:44:07 2018 GMT (3650 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
[root@localhost easy-rsa]# █

```

Fig 3.13

Creación de certificados para usuario2 que denominamos remoto2_gerencia_castro como se puede apreciar también pertenece a la organización MATRIZ_A . cabe indicar que este certificado tiene que se copiado en el host remoto2 para que se puede logear contra el servidor MATRIZ_A Ver Figura 3.14

```

Archivo  Editar  Ver  Terminal  Solapas  Ayuda
[root@localhost easy-rsa]# ls
2.0    build-dh    build-key    build-key-pkcs12  build-req    clea
build-ca  build-inter  build-key-pass  build-key-server  build-req-pass  keys
[root@localhost easy-rsa]# ./build-key remoto2_gerencia_bravo
Generating a 1024 bit RSA private key
.....+++++
...+++++
writing new private key to 'remoto2_gerencia_bravo.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [EC]:
State or Province Name (full name) [AZ]:
Locality Name (eg, city) [CCA]:
Organization Name (eg, company) [MATRIZ_A]:
Organizational Unit Name (eg, section) []:MATRIZ_A
Common Name (eg, your name or your server's hostname) []:REMOTO2_GERENCIA
Email Address [wimer882@cue.satnet.net]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:REMOTO2_GERENCIA
An optional company name []:MATRIZ
Using configuration from /etc/openssl.cnf
DEBUG[load_index]: unique_subject = "yes"
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName       :PRINTABLE:'EC'
stateOrProvinceName :PRINTABLE:'AZ'
localityName       :PRINTABLE:'CCA'
organizationName   :T61STRING:'MATRIZ_A'
organizationalUnitName:T61STRING:'MATRIZ_A'
commonName         :T61STRING:'REMOTO2_GERENCIA'
emailAddress       :IASSTRING:'wimer882@cue.satnet.net'
Certificate is to be certified until Oct 17 16:45:57 2018 GMT (3650 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]:y
Write out database with 1 new entries
Data Base Updated
[root@localhost easy-rsa]# █

```

Fig 3.14

Compilación de todos los certificados con build-dh. Ver fig 3.15

```

Archivo  Editar  Ver  Terminal  Solapas  Ayuda
[root@localhost easy-rsa]# ls
2.0    build-dh    build-key    build-key-pkcs12  build-req    clean-all  list-crl  openssl.cnf  revoke-cert  sign-req  Windows
build-ca  build-inter  build-key-pass  build-key-server  build-req-pass  keys      make-crl  README      revoke-full  vars
[root@localhost easy-rsa]# ./build-dh
Generating DH parameters, 1024 bit long safe prime, generator 2
This is going to take a long time
.....+++++
...+++++
[root@localhost easy-rsa]# █

```

Fig 3.15

Hasta ahora hemos concluido la construcciones de certificados tanto para el servidor MATRIZ_A como para los 2 usuarios remotos todos estos certificados se han creado automáticamente en el directorio /keys de /easy-rsa. Ver Fig 3.16 y 3.17

```

Aplicaciones Lugares Escritorio
root@localhost/etc/openssl-2.0.9/easy-rsa/keys
Archivo Editar Ver Terminal Solapas Ayuda
[root@localhost easy-rsa]# ls
2.0 build-dh build-key build-key-pkcs12 build-req clean-all list-crl openssl.cnf revoke-cert sign-req
build-ca build-inter build-key-pass build-key-server build-req-pass keys make-crl README revoke-full vars
[root@localhost easy-rsa]# cd keys/
[root@localhost keys]#
    
```

Fig 3.17

```

Archivo Editar Ver Terminal Solapas Ayuda
[root@localhost keys]# ls
01.pem ca.crt index.txt index.txt.old MATRIZ_A.key remoto1_ventas_castro.key remoto2_gerencia_bravo.key
02.pem ca.key index.txt.attr MATRIZ_A.crt remoto1_ventas_castro.crt remoto2_gerencia_bravo.crt serial
03.pem dh1024.pem index.txt.attr.old MATRIZ_A.csr remoto1_ventas_castro.csr remoto2_gerencia_bravo.csr serial.old
[root@localhost keys]#
    
```

Fig 3.18

Cada uno de estos certificados tiene que ser copiados en los hosts remotos dependiendo del tipo de conexión VPN que realizaremos.

3.6 CREACION DE CLAVE SECRETA PARA CONEXIONES RED A RED,

Para nuestro caso seria red MATRIZ_A contra SUCURSAL_B,

Tenemos que dirigirnos al directorio /etc/openssl-2.0.9 y generar una clave la misma que tendrá que ser copiada en el servidor SUCURSAL_B esta clave tendrá que ser la misma en los 2 servidores ya que a través de esta comenzara la validación para llegar posteriormente a secuencia de encriptación y enlace.

Ver Fig 3.18 y 3.19

```

crypto.h init.h misc.c openssl.8 PORTS shaper.h
crypto.o init.o misc.h openssl.c openssl.proto.c shaper.o
[root@localhost openssl-2.0.9]# openssl --genkey --secret key_MA_SB.txt
[root@localhost openssl-2.0.9]#
    
```

Fig 3.18 y Fig 3.19

```

[root@localhost openssl-2.0.9]# ls
acinclude.m4 debug INSTALL misc.h openssl.c
aclocal.m4 depcomp install-sh misc.o openssl.h
AUTHORS docclean install-win32 missing openssl.o
base64.c easy-rsa INSTALL-win32.txt mroute.c openssl-plugin.h
base64.h errlevel.h integer.h mroute.h openssl.spec
base64.o error.c interval.c mroute.o openssl.spec.in
basic.h error.h interval.h mss.c openssl-status.log
buffer.c error.o interval.o mss.h options.c
buffer.h event.c ip.txt mss.o options.h
buffer.o event.h key_MA_SB.txt mtcp.c options.o
ChangeLog event.o list.c mtcp.h otimer.c
circ_list.h fdmisc.c list.h mtcp.o otimer.h
common.h fdmisc.h list.o mtu.c otimer.o
config-guess fdmisc.o lzo.c ntu.h packet_id.c
config.h forward.c lzo.h ntu.o packet_id.h
config.h.in forward.h lzo.o nudp.c packet_id.o
config.log forward-inline.h Makefile nudp.h perf.c
config.status forward.o Makefile.am nudp.o perf.h
config.sub fragment.c Makefile.in multi.c perf.o
configure fragment.h makefile.w32 multi.h ping.c
configure.ac fragment.o makefile.w32-vc multi.o ping.h
config-win32.h gentoo manage.c NEWS ping-inline.h
config-win32.h.in gremlin.c manage.h ntlm.c ping.o
contrib gremlin.h management ntlm.h plugin
COPYING gremlin.o manage.o ntlm.o plugin.c
cryptoapi.c helper.c mbuf.c occ.c plugin.h
cryptoapi.h helper.h mbuf.o occ.h plugin.o
crypto.c init.c mbuf.c occ-inline.h pool.c
crypto.h init.h mbuf.o memcmp.c occ.o pool.h
crypto.o init.o mbuf.h mndbg.h openvpn pool.o
misc.c mndbg.h openvpn.8 PORTS
    
```

Hasta aquí tenemos todo el software compilado y listo para trabajar junto con los archivos de configuración que veremos mas adelante su aplicación uso estaría completo.

3.7 CREACION Y COMPILACION DE ARCHIVOS DE CONFIGURACION PARA CONEXIONES VPN.

Los archivos de configuración no son más que los scripts que se ejecutaran en el servidor y los dispositivos remotos para empezar las secuencias de validación, encriptación y enlace entre los tipos de VPNS que vayamos a realizar.

3.7.1 Configuración HOST_RED,

En nuestra red los tenemos definidos como remoto1_ventas_castro y remoto2_gerencia_bravo accediendo a archivos en el servidor MATRIZ_A y/o archivos en host que pertenezcan a la red MATRIZ_A.

3.7.1.1 Archivo de configuración en MATRIZ_A

Entramos en el directorio /etc/openvpn-2.0.9 y creamos con el editor vi un archivo al que hemos denominado server_conex_remotas.conf . Ver fig 3.20

```

[root@localhost openvpn-2.0.9]# ls
acinclude.m4      COPYING          forward.o        list.c           mroute.c        occ-inline.h    ping.h           route.c          socket.o
aclocal.m4       COPYRIGHT.GPL   fragment.c      list.h           mroute.h        occ.o           ping-inline.h   route.h          socks.c
AUTHORS          cryptoapi.c     fragment.h      list.o           mroute.o        openvpn         ping.o           route.o          socks.h
base64.c         cryptoapi.h     fragment.o      lzo.c           mss.c           openvpn.8       plugin           sample-config-files
base64.h         crypto.c        gentoo          lzo.h           mss.h           openvpn.c       plugin.c        sample-keys
base64.o         crypto.h        gremlin.c      lzo.o           mss.o           openvpn.h       plugin.h        sample-scripts
basic.h          crypto.o        gremlin.h      Makefile        mstp.c          openvpn.o       plugin.o        schedule.c
buffer.c         debug          gremlin.o      Makefile.am     mtcp.h          openvpn-plugin.h pool.c          schedule.h
buffer.h         depcomp        helper.c        Makefile.in     mtcp.o          openvpn-spec   pool.h          schedule.o
buffer.o         doclean        helper.h        makefile.w32   mtu.c           openvpn-spec.in pool.o          server_conex_remotas.conf
Changelog        easy-rsa       helper.o        makefile.w32-vc mtu.h           openvpn-status.log ports           server.conf
circ_list.h      errlevel.h     init.c          manage.c        mtu.o           options.c       proto.c         server_red_red.conf
common.h         error.c        init.h          manage.h        mudp.c          options.h       proto.h         service-win32
config_guess     error.h        init.o          management     mudp.h          options.o       proxy.c         session_id.c
config.h         error.o        INSTALL        manage.o        mudp.o          otime.c        proxy.h         session_id.h
config.h.in      event.c        install-sh     mbuf.c         multi.c         otime.h        proxy.o        session_id.o
config.log       event.h        install-win32 mbuf.h         multi.h         otime.o        push.c         shaper.c
config.status    event.o        INSTALL-win32.txt mbuf.o         multi.o         packet_id.c    push.h         shaper.h
config.sub       fdmisc.c      integer.h      memcmp.c       NEWS            packet_id.h    push.o         shaper.o
configfigure     fdmisc.h      interval.c     memdbg.h       ntlm.c          packet_id.o    push.o         sig.c
configure.ac     fdmisc.o      interval.o     misc.c         ntlm.h          perf.c         README         sig.h
config-win32.h   forward.c     interval.o     misc.h         ntlm.o          perf.h         reliable.c     sig.o
config-win32.h.in forward.h     ipp.txt       misc.o         occ.c            perf.o         reliable.h     socket.c
contrib          forward-inline.h key.txt       missing        occ.h            ping.c         reliable.o     socket.h

```

Fig 3.20

Este archivo .conf creado con un editor de texto en este caso vi contendra:

Archivo de configuración OPENVPN_configurado en MATRIZ_A para aceptar conexiones remotas.

local 100.100.100.1 (la direccion Ip Publica del servidor)
 port 1194 (Puerto UDP en el cual trabaja la aplicacion OPenVpn)
 proto udp (tipo de protocolo que su usara para la transmisión)
 dev tun (tipo de tunnel que se manejara en este caso routing)

(Directorios donde se encuentran creados los certificados autenticados.)

```
ca "/etc/openvpn-2.0.9/easy-rsa/keys/ca.crt"
cert "/etc/openvpn-2.0.9/easy-rsa/keys/MATRIZ_A.crt"
key "/etc/openvpn-2.0.9/easy-rsa/keys/MATRIZ_A.key"
dh "/etc/openvpn-2.0.9/easy-rsa/keys/dh1024.pem"
```

(El Servidor hace de DHCP entregando IPs del siguiente rango de RED)

```
server 192.169.1.0 255.255.255.0
ifconfig-pool-persist ipp.txt
push "route 192.168.3.0 255.255.255.0" (Ruta que permite acceder a host detras de
servidor OPENVPN configurado para aceptar clientes remotos)
keepalive 10 120 (control del canal )
user nobody
max-clients 100 (numero de conexiones remotas en este ejemplo esta 100)
persist-key (si se cae el dispositivo remoto tiene que volver a reabrirse la etapa de
intrecambio de claves)
persist-tun (no cerrar y reabrir el canal tun/tap)
status openvpn-status.log
verb 1 (sumario)
```

Grabamos el archivo `vi server_conex_remotas.conf` con el comando `wq`.

Se necesita activar el ruteo de paquetes.

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Ejecutamos el script de open VPN con el comando `openvpn --config` y el nombre del archivo en este caso `server_conex_remotas.conf`. Ver fig 3.21

```
[root@localhost openvpn-2.0.9]# ls
acinclude.m4      COPYING          forward.o        list.c           mroute.c        occ-inline.h     ping.h           route.c
acllocal.m4      COPYRIGHT.GPL    fragment.c       list.h           mroute.h        occ.o            ping-inline.h    route.h
AUTHORS          cryptoapi.c      fragment.h       list.o           mroute.o        openvpn          plugin.o         route.o
base64.c         cryptoapi.h      gentoo           lzo.c           mss.c           openvpn.8        plugin.h         sample-config-files
base64.h         crypto.c         gremlin.c       lzo.h           mss.h           openvpn.c        plugin.h         sample-keys
base64.o         crypto.h         gremlin.o       lzo.o           mss.o           openvpn.h        plugin.h         sample-scripts
basic.h          crypto.o         gremlin.h       Makefile         mtcp.c          openvpn.o        plugin.o         schedule.c
buffer.c         debug           gremlin.o       Makefile.am      mtcp.h          openvpn-plugin.h pool.c           schedule.h
buffer.h         depcomp        helper.c         Makefile.in      mtcp.o          openvpn.spec     pool.h           schedule.o
buffer.o         docclean       helper.h         makefile.w32     mtu.c           openvpn.spec.in pool.o           server_conex_remotas.conf
ChangeLog       easy-rsa       helper.o        makefile.w32-vc mtu.h           openvpn-status.log PORTS            server_conf
circ_list.h     errlevel.h     init.c          manage.c         mtu.o           options.c        proto.c          server_red_red.conf
common.h        error.c        init.h          manage.h         mudp.c          options.h        proto.h          service-win32
config.guess    error.h        init.o         management       mudp.h          options.o        proxy.c          session_id.c
config.h        event.c       INSTALL        manage.o         mudp.o          otime.c         proxy.h          session_id.h
config.h.in     event.o       install-sh      mbuf.c          multi.c         otime.h         proxy.o          session_id.o
config.log      event.h       install-win32  mbuf.o          multi.h         otime.o         proxy.h          shaper.c
config.status   fdmisc.c     INSTALL-win32.txt mbuf.o          multi.o         packet_id.c     push.c          shaper.h
config.sub      fdmisc.h     integer.h      memcmp.c        NEWS            packet_id.h     push.h          shaper.o
configure       fdmisc.o     interval.c     memdbg.h        ntlm.c         packet_id.o     push.o          sig.c
configure.ac    fdmisc.o     interval.h     misc.c          ntlm.h         perf.c          push.h          sig.h
config-win32.h  forward.c    interval.o     misc.h          ntlm.o         perf.h          reliable.c      sig.o
config-win32.h.in forward.h     app.txt       misc.o          occ.c          perf.o          reliable.h      socket.c
contrib        forward-inline.h key.txt        missing         occ.h          ping.c          reliable.o      socket.h
```

```
[root@localhost openvpn-2.0.9]# openvpn --config server_conex_remotas.conf
Sun Oct 12 20:30:24 2008 OpenVPN 2.0.9 1686-pc-linux [SSL] [EPOLL] built on Jun 28 2008
Sun Oct 12 20:30:24 2008 TUN/TAP device tun0 opened
Sun Oct 12 20:30:24 2008 /sbin/ifconfig tun0 192.169.1.1 pointopoint 192.169.1.2 mtu 1500
Sun Oct 12 20:30:24 2008 UID set to nobody
Sun Oct 12 20:30:24 2008 UDPv4 link local (bound): 100.100.100.1:1194
Sun Oct 12 20:30:24 2008 UDPv4 link remote: [undef]
Sun Oct 12 20:30:24 2008 Initialization Sequence Completed
```

Fig 3.21

La inicializacion de esta scripts ha sido exitosa si verificamos los interfaces podemos observar que se ha generado una interfaz virtual que conectara a todos los usuarios remotos en nuestro caso para `remoto1` y `remoto2`. Ver Fig 3.22

```
[root@localhost openvpn-2.0.9]# ifconfig
eth0      Link encap:Ethernet HWaddr 00:13:8F:CC:43:43
          inet addr:100.100.100.1 Bcast:100.100.100.255 Mask:255.255.255.0
          inet6 addr: fe80::213:8fff:fecc:4343/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:108 errors:0 dropped:0 overruns:0 frame:0
          TX packets:25 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:16360 (15.9 KiB) TX bytes:2266 (2.2 KiB)
          Interrupt:11 Base address:0xb000

eth1      Link encap:Ethernet HWaddr 00:02:44:85:FA:86
          inet addr:192.168.2.1 Bcast:192.168.2.255 Mask:255.255.255.0
          inet6 addr: fe80::202:44ff:fe85:fa86/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:102 errors:0 dropped:0 overruns:0 frame:0
          TX packets:21 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15916 (15.5 KiB) TX bytes:1857 (1.8 KiB)
          Interrupt:11 Base address:0x6c00

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:1412 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1412 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3623324 (3.4 MiB) TX bytes:3623324 (3.4 MiB)

tun0     Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:192.169.1.1 P-t-P:192.169.1.2 Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 b) TX bytes:40 (40.0 b)
```

Fig 3.22

3.7.1.2 Archivos de configuración para remoto1 y remoto2

Los mismos que tienen que ser realizados en los respectivos host remotos a través del editor vi con la extensión .conf y compilado a través del comando `openvpn --config "nombre del archivo.conf"`

Archivo para remoto1_ventas_castro.conf

Client (remoto1)

dev tun

port 1194

proto udp

remote 100.100.100.1

resolv-retry infinite

nobind

persist-key

persist-tun

ca "/etc/openvpn-2.0.9/easy-rsa/keys/ca.crt"(Archivo generado en servidor MATRIZ_A y copiado en remoto1_ventas_castro)

cert "/etc/openvpn-2.0.9/easy-rsa/keys/remoto1_ventas_castro.crt" (Archivo generado en servidor MATRIZ_A y copiado en remoto1_ventas_castro)

key "/etc/openvpn-2.0.9/easy-rsa/keys/remoto1_ventas_castro.key" (Archivo generado en servidor MATRIZ_A y copiado en remoto1_ventas_castro)

ns-cert-type server

cipher BF-CBC

route 192.168.3.0 255.255.255.0

verb 1

De igual forma al terminar el archivo `remoto1_ventas_castro.conf` hay que compilarlo con `openvpn --config remoto1_ventas_castro.conf` una vez inicializado también se genera una interfaz virtual con la que accederemos hacia el servidor MATRIZ_A.

Archivo de configuración para remoto2_gerencia_bravo.conf

Client (remoto2)

dev tun

port 1194

proto udp

remote 100.100.100.1

resolv-retry infinite

nobind

persist-key

persist-tun

ca "/etc/openvpn-2.0.9/easy-rsa/keys/ca.crt"(Archivo generado en servidor MATRIZ_A y copiado en remoto2_gerencia_bravo)

cert "/etc/openvpn-2.0.9/easy-rsa/keys/remoto2_gerencia_bravo.crt" "(Archivo generado en servidor MATRIZ_A y copiado en remoto2_gerencia_bravo)

key "/etc/openvpn-2.0.9/easy-rsa/keys/remoto2_gerenci_bravo.key" "(Archivo generado en servidor MATRIZ_A y copiado en remoto2_gerencia_bravo)

ns-cert-type server

cipher BF-CBC

route 192.168.3.0 255.255.255.0

verb 1

De igual forma al terminar el archivo remoto1_ventas_castro.conf hay que compilarlo con `openvpn --config remoto1_ventas_castro.conf` una vez inicializado tambien se generar una interfaz virtual con la que accederemos hacia el servidor MATRIA_A.

En conclusión el momento que realizamos una VPN para conectar usuario remotos debemos generar en primera instancia el archivo de configuracion en el servidor con los respectivos certificado y claves las claves dependiendo el usuario deben ser utilizadas en los respectivos host remotos atraves de los archivos de configuración .conf

3.7.2 Configuración RED-RED,

Estos archivos servirán para enlazar las respectivas redes LAN de la MATRIZ_A como de la SUCURSAL_B, como habíamos indicado anteriormente se iniciaba generando un archivo key.txt que para nuestro caso es key_MA_SB.txt copiando el mismo en el servidor remoto y concluyendo compilando los respectivos archivos de configuración .conf .

3.7.2.1 Archivo de Configuración en MATRIZ_A para conexión RED-RED, tunnel_MATRIZ-A_SUCURSAL_B.conf

```
remote 100.100.100.2
port 1195
dev tun
persist-tun # necesario al ejecutarse como "nobody".
ifconfig 10.10.10.1 10.10.10.2 # nodo local - nodo remoto.
ping 15
ping-restart 120
verb 3
secret /etc/openvpn-2.0.9/key_MA_SB.txt hay que crearla y copiarla al remoto.
persist-key # necesario al ejecutarse como "nobody".
persist-tun
route 192.168.3.0 255.255.255.0 # se ruta por aquí lo que vaya a la red
SUCURSAL_B.
status openvpn-status.log
```

Estas son las líneas de comando que estarían dentro del script, guardamos los cambios, activamos el ruteo de paquetes y lo ejecutamos. Ver fig 3.23

También hay que acotar que luego de ejecutar el script tunnel_MATRIZ-A_SUCURSAL-B se generará otra interfaz virtual tun1-00 que será el punto de conexión para toda la red de SUCURSAL_B. Ver fig 3.24

```
[root@localhost openvpn-2.0.9]# vi tunel_MATRIZ-A_SUCURSAL-B.conf
[root@localhost openvpn-2.0.9]# ls
acinclude.m4      debug          INSTALL       misc.h        openvpn.c     proto.c       shaper.o
aclocal.m4       depcomp       install-sh    misc.o        openvpn.h     proto.h       sig.c
AUTHORS          docclean      install-win32 missing       openvpn.o     proxy.c       sig.h
base64.c         easy-rsa      INSTALL-win32.txt mroute.c     openvpn-plugin.h proxy.o       socket.c
base64.h         errlevel.h   integer.h    mroute.h     openvpn.spec  proxy.h       socket.h
base64.o         error.c      interval.c   mroute.o     openvpn.spec.in proxy.o       socket.o
basic.h          error.h      interval.h   mss.c        openvpn-status.log push.c
buffer.c         error.o      interval.o   mss.h        options.c     push.h        socks.c
buffer.h         event.c      ipp.txt     mss.o        options.h     push.o        socks.h
buffer.o         event.h      key.txt     mtcp.c       options.o     README        socks.o
Changelog        event.o     list.c      mtcp.h       otime.c      reliable.c    ssl.c
circ_list.h     fdmisc.c    list.h      mtcp.o       otime.h      reliable.h    ssl.h
common.h        fdmisc.h    list.o      mtu.c        otime.o      reliable.o    ssl.o
config_guess    fdmisc.o    lzo.c      mtu.h        packet_id.c  route.c      stamp-h1
config.h         forward.c   lzo.h      mtu.o        packet_id.h  route.h      status.c
config.h.in     forward.h   lzo.o      mudp.c       packet_id.o  route.o      status.h
config.log      forward-inline.h Makefile    mudp.h       perf.c       sample-config-files status.o
config.status   forward.o   Makefile.am mudp.o       perf.h       sample-keys   suse
config.sub      fragment.c  Makefile.in multi.c      perf.o       sample-scripts syshead.h
configure       fragment.h  makefile.w32 multi.h      ping.c       schedule.c   tap-win32
configure.ac    fragment.o  makefile.w32-vc multi.o      ping.h       schedule.h   t_cltsrv.sh
config-win32.h  gentoo     manage.c   NEWS        ping-inline.h schedule.o    thread.c
config-win32.h.in gremlin.c  manage.h   ntlm.c      ping.o       server_conex_remotas.conf thread.h
contrib         gremlin.h  management ntlm.h      plugin       server.conf  thread.o
COPYING         gremlin.o  manage.o   ntlm.o      plugin.c     server_red_red.conf t_lpbck.sh
COPYRIGHT.GPL  helper.c   mbuf.c    occ.c        plugin.h     service-win32 tun.c
cryptoapi.c     helper.h   mbuf.h    occ.h        plugin.o     session_id.c  tune1
cryptoapi.h     helper.o   mbuf.o    occ-inline.h pool.c       session_id.h  tune.h
crypto.c        init.c    memcmp.c  occ.o        pool.h       session_id.o  tune.o
crypto.h        init.h    memdbg.h  openvpn     pool.o       shaper.c      win32.c
crypto.o        init.o    misc.c    openvpn.8   PORTS       shaper.h      win32.h
[root@localhost openvpn-2.0.9]# openvpn --config tunel_MATRIZ-A_SUCURSAL-B.conf
Sun Oct 12 20:57:39 2008 OpenVPN 2.0.9 i686-pc-linux [SSL] [EPOLL] built on Jun 28 2008
Sun Oct 12 20:57:39 2008 Static Encrypt: Cipher 'BF-CBC' initialized with 128 bit key
Sun Oct 12 20:57:39 2008 Static Encrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Sun Oct 12 20:57:39 2008 Static Decrypt: Cipher 'BF-CBC' initialized with 128 bit key
Sun Oct 12 20:57:39 2008 Static Decrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Sun Oct 12 20:57:39 2008 TUN/TAP device tun0 opened
Sun Oct 12 20:57:39 2008 /sbin/ifconfig tun0 10.10.10.1 pointopoint 10.10.10.2 mtu 1500
Sun Oct 12 20:57:39 2008 /sbin/route add -net 192.168.3.0 netmask 255.255.255.0 gw 10.10.10.2
Sun Oct 12 20:57:39 2008 Data Channel MTU parms [ L:1544 D:1450 EF:44 EB:4 ET:0 EL:0 ]
Sun Oct 12 20:57:39 2008 Local Options hash (VER=V4): 'cbd9413b'
Sun Oct 12 20:57:39 2008 Expected Remote Options hash (VER=V4): '216375b8'
Sun Oct 12 20:57:39 2008 UDPv4 link local (bound): [undef]:1195
Sun Oct 12 20:57:39 2008 UDPv4 link remote: 100.100.100.2:1195
Sun Oct 12 20:57:43 2008 Peer Connection Initiated with 100.100.100.2:1195
Sun Oct 12 20:57:43 2008 Initialization Sequence Completed
```

Fig 3.23

```
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet HWaddr 00:13:8F:CC:43:43
          inet addr:100.100.100.1 Bcast:100.100.100.255 Mask:255.255.255.0
          inet6 addr: fe80::213:8fff:fecc:4343/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:114 errors:0 dropped:0 overruns:0 frame:0
          TX packets:30 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:16954 (16.5 KiB) TX bytes:2716 (2.6 KiB)
          Interrupt:11 Base address:0xb000

eth1      Link encap:Ethernet HWaddr 00:02:44:85:FA:86
          inet addr:192.168.2.1 Bcast:192.168.2.255 Mask:255.255.255.0
          inet6 addr: fe80::202:44ff:fe85:fa86/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:103 errors:0 dropped:0 overruns:0 frame:0
          TX packets:21 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15976 (15.6 KiB) TX bytes:1857 (1.8 KiB)
          Interrupt:11 Base address:0x6c00

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:1412 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1412 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3623324 (3.4 MiB) TX bytes:3623324 (3.4 MiB)

tun0     Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
-00
          inet addr:192.169.1.1 P-t-P:192.169.1.2 Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 b) TX bytes:80 (80.0 b)

tun1     Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
-00
          inet addr:10.10.10.1 P-t-P:10.10.10.2 Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 b) TX bytes:40 (40.0 b)
```

Fig 3.24

3.7.2.2 Archivo de Configuración en SUCURSAL_B para conexión RED-RED, `tunel_MATRIZ-A_SUCURSAL_B.conf`

```

remote 100.100.100.1
port 1195
dev tun
persist-tun # necesario al ejecutarse como "nobody".
ifconfig 10.10.10.2 10.10.10.1 # nodo local - nodo remoto.
ping 15
ping-restart 120
verb 3
secret /etc/openvpn-2.0.9/key_MA_SB.txt hay que crearla y copiarla al remoto.
persist-key # necesario al ejecutarse como "nobody".
persist-tun
route 192.168.1.0 255.255.255.0 # se ruta por aquí lo que vaya a la red MATRIZ_A.
status openvpn-status.log

```

Estas son las líneas de comando que estarían dentro del script, guardamos los cambios, activamos el ruteo de paquetes (`echo 1 > /proc/sys/net/ipv4/ip_forward`) y lo ejecutamos. Ver fig 3.23

También hay que acotar que luego de ejecutar el script `tunel_MATRIZ-A_SUCURSAL-B` se generará otra interfaz virtual `tun0-00` que será el punto de conexión para toda la red de `MATRIZ_A`. Ver fig 3.24.

3.8 RESUMEN GENERAL.

Como resumen general se puede indicar que hemos realizado 3 enlaces VPN de todas las opciones posibles(host-host, host-red y red-red), definiendo las VPNS tenemos una conectan toda la red de MATRIZ_A (192.168.2.0/24) con la red de SUCURSAL_B (192.168.3.0/24) a través del archivo de configuración `tunel_MATRIZ-A_SUCURSAL-B.conf` uno en cada servidor, las otras 2 VPNS son para enlazar los puntos remotos atravez de los archivos de configuración `server_conex_remotas.conf` en la MATRIZ_A y `remoto1_ventas_castro.conf` y `remoto2_gerencia_bravo.conf` en los respectivos host remotos, es decir en el servidor MATRIZ_A se ejecuta simultáneamente 2 archivos de configuración que genera 2 interfaces virtuales definidas como `tun0-00` (para enlazar los host remotos) y `tun1-00` (para enlazar la red remota).

```

Archivo  Editar  Ver  Terminal  Solapas  Ayuda
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:13:8F:CC:43:43
          inet addr:100.100.100.1 Bcast:100.100.100.255 Mask:255.255.255.0
          inet6 addr: fe80::213:8fff:fecc:4343/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500 Metric:1
          RX packets:114 errors:0 dropped:0 overruns:0 frame:0
          TX packets:30 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:16954 (16.5 KiB) TX bytes:2716 (2.6 KiB)
          Interrupt:11 Base address:0xb000

eth1      Link encap:Ethernet  HWaddr 00:02:44:85:FA:86
          inet addr:192.168.2.1 Bcast:192.168.2.255 Mask:255.255.255.0
          inet6 addr: fe80::202:44ff:fe85:fa86/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500 Metric:1
          RX packets:103 errors:0 dropped:0 overruns:0 frame:0
          TX packets:21 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15976 (15.6 KiB) TX bytes:1857 (1.8 KiB)
          Interrupt:11 Base address:0x6c00

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436 Metric:1
          RX packets:1412 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1412 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3623324 (3.4 MiB) TX bytes:3623324 (3.4 MiB)

tun0-00   Link encap:UNSPEC   HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:192.169.1.1 P-t-P:192.169.1.2 Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 b) TX bytes:80 (80.0 b)

tun1-00   Link encap:UNSPEC   HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.10.10.1 P-t-P:10.10.10.2 Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 b) TX bytes:40 (40.0 b)

```

Explicado claramente como esta estructurada toda la red VPN tenemos todo listo para TRAFICAR por la misma.

3.9 COMPROBACION DE LA RED VPN.

Para nuestro caso y teniendo en cuenta que se trata de un trabajo de grado con ejemplo practico comprobaremos nuestra red a través de pruebas de icmp y transferencia de archivos vía FTP entre cada uno de los host involucrados en la red también podremos apreciar a través del comando tracert y traceroute los respectivos saltos que hace cada uno de los host al hacer las respectivas de peticiones hacia los equipos y servidores remotos. Ver ejemplo Fig 3.9.1

```

Sufijo de conexión específica DNS :
Dirección IP. . . . . : 100.100.100.5
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada :

Adaptador Ethernet Conexión de área local 7 :

Sufijo de conexión específica DNS :
Dirección IP. . . . . : 192.169.1.22
Máscara de subred . . . . . : 255.255.255.252
Puerta de enlace predeterminada :

C:\Documents and Settings\Administrador>tracert 192.168.3.2
Traza a 192.168.3.2 sobre caminos de 30 saltos como máximo.

 1    1 ms    <1 ms    <1 ms    192.169.1.1
 2    2 ms    1 ms     1 ms     10.10.10.2
 3    2 ms    1 ms     1 ms     192.168.3.2

Traza completa.
C:\Documents and Settings\Administrador>

```

Fig 3.9.1 Traceroute desde usuario remoto “C” conectado a la red interna a través de MATRIZ “A”, como habíamos explicado anteriormente la red a la que pertenecerá los usuarios remotos “C” es 192.169.1.0/24 en este caso la ip del usuario remoto entregado por el servidor VPN en MATRIZ_A es 192.169.1.22 que hace una prueba de conexión a el host 192.168.3.2/24 de la SUCURSAL_B. Como podemos ver en la figura el primer salto es el servidor VPN en “A” que acepta las conexiones remotas de los usuarios móviles (conexión host-host), este enruta el paquete IP hacia el servidor VPN de la SUCURSAL_B 10.10.10.2/30 (conexión host-red), finalmente llegando al host 192.168.3.2 que el destino.

4. CONCLUSIONES Y RECOMEDACIONES

A la terminación de este proyecto podemos concluir que hemos analizado la teoría necesaria sobre las Redes Privadas Virtuales y las características de software GNU Linux aplicado a estas redes, luego de agregar el hardware y software necesario hemos conseguido configurar y habilitar los dos servidores Linux para que nuestra red VPN en cada uno de los diagramas expuestos funcione adecuadamente, validando su correcto funcionamiento a través de pruebas de icmp, uso de aplicaciones como ssh y transferencia de archivos vía ftp y ftps entre los diferentes host que intervienen en la red. Además hemos podido ratificar que las VPN representan hoy en día una gran alternativa para los enlaces WAN y se ha vuelto un tema muy importante en las organizaciones ya que a mas de reducir significativamente los costos de comunicación, nos proveen de seguridad y confidencialidad en el intercambio de información.

Por otro lado, puesto que este documento trata de ciertos conceptos y términos familiarizados en configuraciones de redes y Linux nuestro objetivo principal esta en la realización de la Red Privada Virtual, se recomienda familiarizarnos con los debidos fundamentos antes de empezar a trabajar con este proyecto.

5. BIBLIOGRAFIA

5.1 Referencias Bibliograficas:

Vpns Illustrated

Snader, Jon (Addison Wesley)

Redes Cisco: Guía De Estudio Para La Certificación Ccna 640-801

ARIGANELLO, E. (Editorial Ra-ma)

Redes Locales, 4ª Edición.

RAYA, J.L.- RAYA, L. (Editorial Ra-ma)

Redes Privadas Con Linux

Oleg Kolesnikov-Brian hatch (Prentice Hall)

Redes De Comunicación

Alberto León-García; Indra Widjaja (McGRAW-

HILL/INTERAMERICANA DE ESPAÑA, S.A.U.)

Network Security Fundamentals

DeLaet, Gert; Schauwers, Gert (CISCO PRESS)

Linux

Kofler, Michael (Addison Wesley Verlag)

Transmisión De Datos Y Redes De Comunicaciones, 4ª Ed.

Forouzan Behrouz (McGRAW-HILL/INTERAMERICANA DE ESPAÑA,

S.A.U.)

Mpls Fundamentals

De Ghein, Luc (CISCO PRESS)

Firewall Linux Guia-Avanzada (Robert L. Zangler)-Prentice Hall.

5.2 Referencias Electrónicas:

Vpns de Software: <http://openvpn.net/>,

Tecnología para el transporte de redes: <http://www.tellabs.com/>

El inicio de las Redes de Telecomunicaciones: <http://www.cisco.com/>,

El cifrado de las Vpns: <http://www.seguridadysistemas.com>,

Características de Seguridad en las redes: <http://www.configurarequipos.com>

Creacion de Tuneles seguros con protocolos de Vpns Estándar :

<http://www.freeswan.org/>

Características de software para Vpns: <http://vpn.ugr.es>

Conceptos y definiciones del protocolo SSL/TLS

<http://en.wikipedia.org/wiki/>

Configuración de Vpns bajo Linux: <http://www.unixwiz.net/techtips/iguide-ipsec.htm>

How to configure Openswan:

<http://wiki.openswan.org/index.php/Openswan/Configure>

Vpns bajo Linux con IPSEC: <http://FreeS-WAN.com>