6

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES
CUAUTITLAN

SISTEMAS DE INFORMACION.- "LOS FIREWALLS COMO POLITICA DE CALIDAD PARA LOS SISTEMAS DE INFORMACION."

TRABAJO DE SEMINARIO

QUE PARA OBTENER EL TITULO DE:

LICENCIADO EN INFORMATICA

P R E S E N T A :

CESAR DAVID CASTRO ARRIAGA

ASESOR: ING. MIGUEL ALVAREZ PASAYE.

CUAUTITLAN IZCALLI, EDO. DE MEXICO. 2001.

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN
UNIDAD DE LA ADMINISTRACION ESCOLAR
DEPARTAMENTO DE EXAMENES PROFESIONALES

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

U N A M
FACULTAD DE ESTUDIOS SUPERIORES-CUAUTITLAN

DEPARTAMENTO DE EXAMENES PROFESIONALES

DR. JUAN ANTONIO MONTARAZ CRESPO
DIRECTOR DE LA FES CUAUTITLAN
P R E S E N T E

ATN. Q. Ma. del Carmen Garcia Mijares
Jefe del Departamento de Exámenes
Profesionales de la FES Cuautitlán

Con base en el art. 51 del Reglamento de Exámenes Profesionales de la FES-Cuautitlán nos permitimos comunicar a usted que revisamos el Trabajo de Seminario
Sistemas de Información. - "Los Firewalls como política de
calidad para los Sistemas de Información."

que presenta el pasante: César David Castro Arriaga.

con número de cuenta: 9651064 - 5 para obtener el título de

Licenciado en Informática.

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el EXÁMEN PROFESIONAL correspondiente, otorgamos nuestro VISTO BUENO

A T E N T A M E N T E
"POR MI RAZA HABLARA EL ESPIRITU"
Cuautitlán Izcalli, Méx. a 11 de Septiembre del 2001.

MODULO	PROFESOR	FIRMA
I	M.C.C. Araceli Nivon Zaghi.	
II	M.C.C. Valentín Roldan Vázquez.	
III	Ing. Miguel Alvarez Pasaye.	

ÍNDICE

CAPÍTULO IV.

FIGURAS.

TABLAS.

AGRADECIMIENTOS.

Porque quien no es agradecido en esta vida sé esta cerrando las puertas que algún día estuvieron abiertas.

Antes que a nadie quisiera que éste sea un pequeño homenaje a Dios, por haberme dado la dicha, de poder realizar mis estudios en general, sin olvidar jamás que siempre estaré en deuda con Él por los dos más grandes favores de mi vida, la salud de mi madre y de mi sobrina. Dentro de éste rubro quisiera dar gracias a toda la obra Salesiana, que en realidad fueron los artífices de mi educación Cristiana, así como a la Virgen María Auxiliadora de los Cristianos por el llamado a formar parte de la congregación y ser el pilar de todos mis éxitos y fracasos.

En segundo término quisiera agradecer a mis padres porque sin ser vituperio, creo que Dios los pondría como el claro ejemplo de los que debe ser una Familia, cada uno en su papel ya sea de esposo-padre y esposa-madre, esto no es más que el producto de un esfuerzo compartido con ustedes, y que como lo he intentado hasta el día de hoy, es el darles siempre un pedazo de alegría, que es la que quiere siempre San Juan Bosco.

No quisiera que el orden que coloco en estas líneas signifique la relevancia que ocupa cada una de las personas que llegaré a nombrar en este trabajo, de antemano cada quien sabe el lugar que ocupa en mi vida. Quedaré infinitamente agradecido con todos(as) ustedes.

A la persona con quien estoy tratando de ser un solo ente, pese a todas las adversidades de toda índole que se nos han presentado y que se nos presentarán, sabes que esa persona eres tu amor, Alejandra Olivares. A mi hermano de sangre, Francisco Castro, para que siga los pasos de mi Padre en ésta nueva aventura que esta iniciando, así como a mis hermanos Guillermo Cummings, Sergio Cortes y Roman Saavedra, porque sin portar la misma sangre me han demostrado su amor en todo momento, así como a sus familias. No pueden pasar desapercibidas las familias Arriaga Colín y Castro Leyva, por ser una parte fundamental en mi vida.

Nunca olvidaré todos los momentos y ayuda, que me brindaron Patricia Baltasar, Alejandra Bernal, Karina Soto, Esther Luevano, Lourdes Lizcano y Elvira. Así como a las Instituciones, por haberme dado la oportunidad de poderme desarrollar en el ámbito académico y laboral, IFE Distrito No. 7, Computación Administrativa y de Diseño, Via Net Works México y a la benemérita Institución la UNAM.

Es una falta de respeto el no mencionar a tanta gente, que de alguna manera u otra ha contribuido, a mí formación tanto académica, civil, moral y espiritual, que reciban de mi primero que nada un reconocimiento y en segundo lugar una disculpa por la omisión.

SEMINARIO DE TITULACIÓN

SISTEMAS DE INFORMACIÓN.

TEMA:

Los *Firewalls* como política de calidad para los Sistemas de Información.

OBJETIVO GENERAL:

Llegar a entender a los *Firewalls* como una política de calidad, debido a que se debe de comprender este término como una inversión dentro de las empresas.

OBJETIVOS PARTICULARES:

- Desarrollar y elaborar un plan efectivo para la implementación de un *Firewall*, y así tener poder llegar a cumplir como una política de calidad dentro de los sistemas de información.

- Explicar en que consiste la herramienta de la seguridad informática denominada *Firewall*, así como sus diferentes arquitecturas existentes.

- Describir a los sistemas *Proxys*, como una alternativa viable, para complementar a un *Firewall*.

- Llegar a conformar las políticas de seguridad más adecuadas sin importar el tipo de *Firewall* a implementar.

INTRODUCCIÓN.

Desde la aparición de los primeros Sistemas de Información en los años 60`s, los riesgos han existido para los mismos, habiéndose desarrollado diversos mecanismos para poder combatir las diferentes amenazas que se han presentado y que los involucran. Debido a que las pérdidas por parte de las compañías año con año se han ido incrementando de manera geométrica y parece ser que la tendencia actual no es un tanto halagadora, debido a que los diversos tipos de ataques ya sean físicos o lógicos de la información producida a los Sistemas de Información se han incrementado en un 498 % desde el año de 1991, por lo que ahora más que nunca los directivos y consejos involucrados en la toma de decisiones, han empezado a tener una visión más analítica y concienzuda en cuanto a considerar a la información como el segundo activo más importante dentro de las empresas después claro está del recurso humano.

Debemos hoy en día definir a la Seguridad Informática como un recurso de calidad dentro de los Sistemas de Información, además de considerarla una medida de prevención y no como muy usualmente sucede en la mayoría de las empresas nacionales como medida correctiva.

Es de vital importancia considerar que la Seguridad Informática en estos momentos tiene que depender básicamente de tres aspectos fundamentales que son: los usuarios, las disciplina y la tecnología, aspectos que se trataran en el desarrollo de este trabajo.

Actualmente las empresas podríamos catalogarlas por el grado de importancia en cuanto el valor de la información que generan sus sistemas, además que esta ayuda en gran medida a los directivos a la toma de decisiones, pero bien no debemos olvidar que hay medios por los cuales pasa todo el proceso de generación de la información ya sean físicos y/o lógicos, internos y/o externos; y es ahí donde empieza a introducirse e inmiscuirse a la Seguridad Informática debiendo empezar a tomar a los *Firewalls* como una política de calidad, y así poder tener bases para que nuestros Sistemas de Información sean un poco más íntegros y confiables. Por lo cual no es inimaginable que hoy en día con una globalización impuesta, se pueda considerar a los *Firewalls* como un método de supervivencia para las empresas, pues de ellos dependerá en gran parte el futuro de las mismas.

Por tal motivo en el transcurso de este documento, nos iremos adentrando en los diferentes tipos de *Firewalls* existentes, claro esta desde sus antecedentes, y así poder estar sino un paso delante de las diferentes amenazas existentes, si a la par de ellas. Es por eso que este trabajo esta desarrollado y pensado, para que cualquier persona inmersa dentro del ámbito informático, y que pueda tener las bases suficientes para poder llegar a plantear una solución de un sistema *Firewall* para los Sistemas de Información dentro de cualquier empresa. Tenemos que empezar diciendo que este documento esta divido en 4 Capítulos, el primero de ellos denominado "Introducción al protocolo *TCP/IP* y la conectividad" contendrá desde una pequeña descripción del modelo *OSI*, pasando por la estructura tanto de *IP* como de *TCP*, el direccionamiento y como es el establecimiento de una conexión por *TCP*, además de la terminología utilizada en la conectividad. Como segundo capítulo entraremos de lleno a revisar y analizar, las definiciones de un

Firewalls, los diferentes tipos de diseño que existen, los *Routers* de selección, así como las listas de acceso, el modelo de la filtración de paquetes, las reglas que se deben seguir para la filtración de paquetes y se hará una pequeña descripción de sólo algunos servicios soportados por la gran mayoría de los *Firewalls* existentes en el mercado. Dentro del tercer capítulo de este trabajo tocaremos puntos muy interesantes tales como los sistemas *Proxys* actualmente muy difundidos, que son, como trabajan, la aplicación de los mismos y las limitaciones que presentan éstos. Como cuarto y último capítulo de este trabajo tendremos a las políticas de seguridad, desde ¿qué es una política de calidad?, pasando por la conformación de las mismas, los riesgos que existentes actualmente, determinación de responsabilidades, los planes de acción y por último un poco acerca de la certificación más difundida y muy poco conocida dentro del ámbito de la seguridad informática.

Cabe recordar que todos los términos técnicos, se explicarán y describirán concienzudamente, en la parte final de éste trabajo, en el Glosario de términos técnicos.

CAPÍTULO I

1. INTRODUCCIÓN AL PROTOCOLO *TCP/IP* Y LA CONECTIVIDAD .

Para poder llegar a comprender y entender la manera en que se maneja la conectividad de cualquier tipo de red es necesario tener muy en cuenta la manera en que se realiza la conexión entre éstas, y que elementos las componen, por tal motivo dentro de éste capítulo denominado Introducción al protocolo *TCP/IP* y la conectividad, tocaremos puntos, que si bien son de dominio público son indispensables, para sentar bases de conocimiento referente al tema central que son los *Firewalls*. No se debe olvidar que dichos puntos no se tratan a profundidad debido a que solamente se toman como una referencia necesaria. Dentro de los aspectos a destacar se describirá el modelo *OSI* brevemente, así como la estructura de los protocolos *TCP* e *IP*, además de explicar y mostrar gráficamente el establecimiento de una conexión *TCP*, esto sin olvidar que debido a la conectividad de los equipos de cómputo el ataque a los Sistemas de Información está muy ala alcance de las personas ya sea de forma voluntaria o involuntaria, por lo que con esto se dará por sentado las bases para poder empezar a hablar y profundizar nuestro tema central los *Firewalls*.

1.1. El modelo de referencia *OSI*.

OSI (Sistema abierto de interconexión), describe cómo se transfiere la información desde una aplicación de software en una computadora a través del medio de transmisión hasta una aplicación de software en otra computadora. Compuesto por siete capas desarrollado por *ISO* (Organización Internacional de Estándares) en el año de 1984. A cada una de estas siete capas se le asigna una tarea específica y se pueden implementar de manera independiente, y estas son las siguientes:

- Capa 7. Aplicación.
- Capa 6. Presentación.
- Capa 5. Sesión.
- Capa 4. Transporte.
- Capa 3. Red.
- Capa 2. Enlace de Datos.
- Capa 1. Física.

1.2. Interacción entre las capas del modelo *OSI*.

Por lo general una capa determinada del modelo *OSI* se comunica con otras tres capas *OSI*: la capa ubicada sobre ella, la capa ubicada directamente debajo de ella y su capa equivalente en otro sistema de computadoras en red. Como se muestra en la siguiente figura:

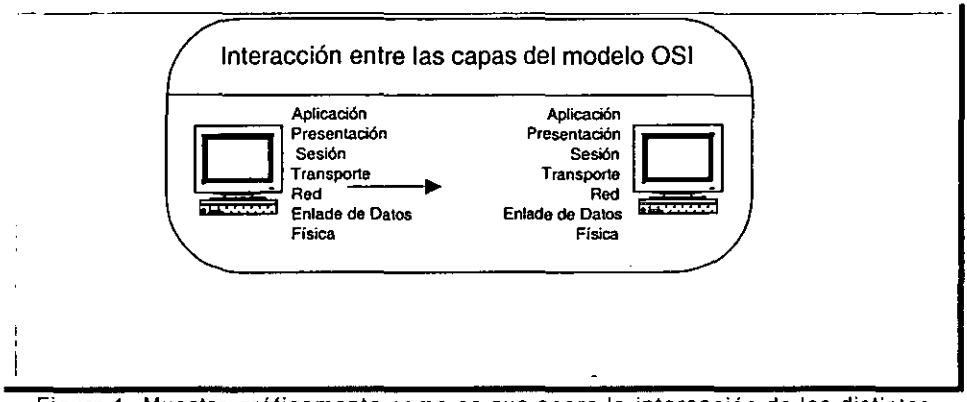

Figura 1. Muestra gráficamente como es que opera la interacción de las distintas capas del modelo *OSI*.

1.2.1. Capa Física.

Esta capa define las especificaciones eléctricas, mecánicas de procedimientos y funcionales para activar, mantener y desactivar el enlace físico entre sistemas de redes de comunicaciones. Las especificaciones de la capa física definen características como nivel de voltaje, temporización de cambios de voltaje, velocidades de transferencia de información, distancias máximas de transmisión y conectores físicos. Algunos ejemplos en el nivel físico son los conectores como *RS232C* y *V.35* y estándares para el uso de cable en redes *LAN*, como *IEEE 802.3*.

1.2.2. Capa de Enlace de Datos.

Proporciona el transito confiable de datos a través del enlace de red. Definen diferentes características de red y protocolo, incluyendo el direccionamiento físico, la topología de red, la notificación de error, la secuencia de tramas y el control de flujo. Algunos ejemplos de esta capa tenemos a los protocolos *Ethernet*, *FDDI* y *ATM*.

1.2.3. Capa de Red.

Esta capa proporciona el ruteo y funciones relacionadas que permiten a múltiples enlaces de datos combinarse en una red. Los protocolos de la capa de red son de hecho los protocolos de ruteo. Algunos ejemplos de esta capa tenemos a los protocolos *IP* e *IPX*.

1.2.4. Capa de Transporte.

Implementa servicios confiables de datos entre redes, transparentes a las capas superiores. Entre las funciones habituales de la capa de transporte se cuentan el control de flujo, el multiplexaje, la administración de circuitos virtuales y la verificación y recuperación de errores. Algunos ejemplos de esta capa tenemos a los protocolos *TCP*, *UDP* y *SPX*.

1.2.5. Capa de Sesión.

Establece, administra y finaliza las sesiones de comunicación entre las entidades de la capa de presentación. Las sesiones de comunicación constan de solicitudes y respuestas de servicio que se presenta entre aplicaciones ubicadas en diferentes dispositivos de red. Algunos ejemplos de esta capa tenemos a un *ASCII* o a un *JPG*

1.2.6. Capa de Presentación.

Brinda una gama de funciones de codificación y conversión que se aplican a los datos de aplicación. Estas funciones aseguran que la información enviada desde la capa de aplicación de un sistema sea legible por la capa de aplicación de otro sistema. Algunos ejemplos de esta capa tenemos a un *Telnet* o a *http*.

1.3. Formatos de Información.

Los datos y la información de control que se transmiten a través de las redes pueden tomar varias formas. Trama, paquete, *datagrama*, segmento, mensaje, celda y unidad de datos, pertenecen a los formatos comunes de información.

Un *datagrama* es una unidad de información cuyo origen y destino son entidades de la capa de enlace de datos. Una trama está compuesta por el encabezado de la capa de enlace de datos (y, posiblemente, un finalizador) y los datos de la capa superior. Un paquete es una unidad de información cuyo origen y destino son entidades de la capa red. Un paquete se compone de un encabezado de la capa de red (y, posiblemente, un finalizador) y datos de la capa superior. El término *datagrama*, por lo general, se refiere a una unidad de información cuyo origen y destino son entidades de la capa de red que utilizan servicios de red no orientados a la conexión. El término segmento, en general, se refiere a una unidad de información cuyo origen y destino son entidades de la capa de transporte. Un mensaje es una unidad de información cuyas entidades origen y destino están sobre la capa de red (muy frecuentemente, en la capa de aplicación). Una celda es una unidad de información de tamaño fijo cuyo origen y destino son las entidades de la capa de enlace de datos. Las celdas se utilizan en entornos

conmutados, como son las redes *ATM*. Una celda se compone de un encabezado e información útil. Una unidad de datos es un término genérico que se refiere a varias unidades de información.

1.4. Protocolos.

Un protocolo es un conjunto formal de reglas y convenciones que gobierna el modo en que las computadoras intercambian información por un medio de transmisión de red.

1.4.1. Historia del *TCP/IP*.

Surge en el año de 1969 con el apoyo de *DARPA*, proyecto que se le conocía como *ARPANET*. Básicamente, esta red proporcionaba conectividad en ancho de banda alto entre los principales sitios de cómputo gubernamentales, educativos y de laboratorios de investigación. *ARPANET* proporcionaba a esos usuarios la capacidad de transferir correo electrónico y archivos de un sitio a otro, mientras que *DARPA* brinda el apoyo financiero para la manutención del proyecto. Durante el año posterior al nacimiento de *ARPANET* esta consistía básicamente de interconexiones de línea rentada de punto a punto. *DARPA* también empezó a presionar para que se investigaran formas alternas de vínculos de comunicación, como vía satélite y radio. Fue en ese momento cuando se empezó a desarrollar una estructura para un conjunto común de tecnologías funcionales para red. El resultado fue *TCP/IP*. Dicha implementación se dirigió principalmente a la implementación BSD de Unix de la Universidad de California de Berkeley. Para el año de 1983 la gran mayoría de las computadoras conectadas a *ARPANET* estaban utilizando los nuevos protocolos *TCP/IP*. Debido a que *ARPANET* estaba limitado un grupo selecto de departamentos y agencias gubernamentales, la *NFS* creó la NFSnet,

que también utilizaba los protocolos de *ARPANET*, ciertamente era una extensión de la anterior, que consistía de un *backbone* que conectaba a todos los centros de súper cómputo de los Estados Unidos. Gracias al enfoque adoptado en NSFnet se dispone de numerosas topologías de red y *TCP/IP* no se restringe a una sola. Esto significa que *TCP/IP* puede operar en *Token Ring*, *Ethernet* y otras topologías de bus, líneas rentadas de punto a punto y otras más. Sin embargo *TCP/IP* ha estado estrechamente ligado a *Ethernet*. Un aspecto importante de mencionar es que la documentación de los protocolos, se especifican en reportes técnicos llamados *RFC*, que se publican y, posteriormente, son revisados y analizados por la comunidad de Internet.

1.4.2. Protocolo Internet.

El *IP* es un protocolo de la capa de red – 3 que tiene información de direccionamiento e información de control que permite el ruteo de paquetes. El *IP* se encuentra documentado en el *RFC* 791 y es el protocolo principal de la capa de red en el conjunto de protocolos de Internet. Junto con el *TCP*, el protocolo *IP* representa el corazón de los protocolos de Internet. El *IP* tiene dos responsabilidades principales: ofrecer la entrega de datagramas basados en el mejor esfuerzo y sin conexión a través de una red; y ofrecer la fragmentación y el reensamblado de datagramas para soportar los enlaces de datos con tamaños diferentes de las *MTU*.

1.4.3. Formato de los paquetes *IP*.

Un paquete *IP* contiene varios tipos de información, como se muestra en la siguiente tabla:

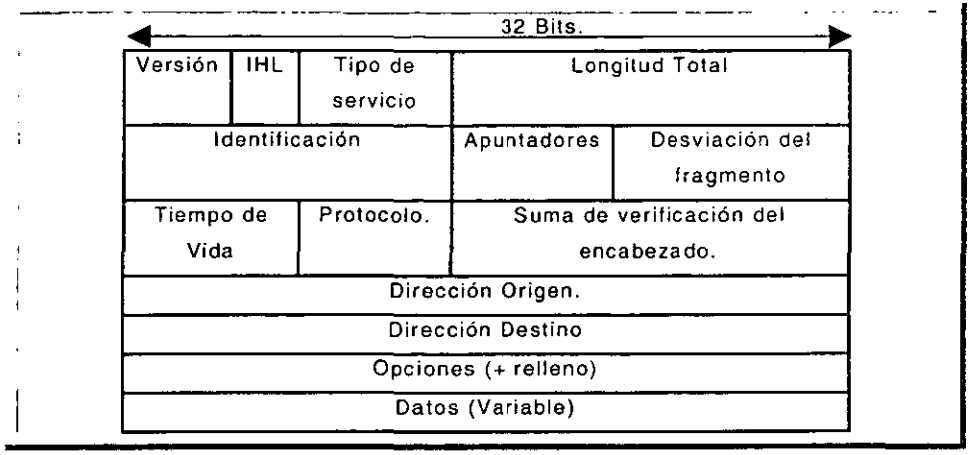

Tabla 1. Muestra los 14 campos de la composición del paquete *IP*.

- Versión: Indica la versión de *IP* actualmente en uso.
- IHL (Longitud del Campo *IP*): Indica la longitud del encabezado del *datagrama* en palabras de 32 bits.
- Tipo de Servicio: Especifica cómo desearía un protocolo de las capas superiores que se manejara un *datagrama* y les asigna diferentes niveles de acuerdo con su importancia.
- Longitud Total: Especifica la longitud, en bytes, del paquete *IP* total incluyendo los datos y el encabezado.
- Identificación: Consta de un número entero que identifica el *datagrama* actual. Este campo se utiliza para ayudar a reconstruir los fragmentos del *datagrama*.
- Apuntadores: Consta de un campo de 3 bits entre los cuales 2 bits de menor orden (los menos significativos) controlan la función de fragmentación. El bit de menor orden especifica si se puede fragmentar el paquete. El bit de en medio especifica si el paquete es el último fragmento en una serie de paquetes fragmentados. El tercer bit, o bit de orden mayor, no se usa.
- Desplazamiento del fragmento: Indica la posición de los datos del fragmento en relación con el comienzo de los datos en el

datagrama original, lo cual permite que el proceso *IP* del destino reconstruya adecuadamente el *datagrama* original.

- Tiempo de Vida: Conserva un contador que disminuye gradualmente hasta llegar a cero, donde se elimina. Esto evita que los paquetes circulen en ciclo de manera indefinida.
- Protocolo: Indica qué protocolo de las capas superiores recibe los paquetes entrantes una vez terminado el procesamiento *IP*.
- Suma de Verificación del Encabezado: Ayuda a asegurar la integridad del encabezado *IP*.
- Dirección Origen: Especifica el nodo emisor.
- Dirección Destino: Especifica el nodo receptor.
- Opciones: Permite que el protocolo *IP* soporte diferentes opciones como la seguridad.
- Datos: Contiene Información de las capas superiores.

1.4.4. Direccionamiento *IP*.

Igual que con cualquier otro protocolo de la capa de red, el esquema de direccionamiento de *IP* es fundamental en el proceso de ruteo de los datagramas *IP* a través de la red. Cada dirección *IP* tiene componentes específicos y sigue un formato básico. Estas direcciones *IP* pueden subdividirse y utilizarse para crear direcciones de subredes, como se analizará más adelante.

A cada *Host* en una red *TCP/IP* se le asigna una dirección lógica única de 32 bits que se subdivide en dos partes principales: el número de red y el número de *Host*. El número de red identifica una red y debe ser asignado por el *NIC*, si la red es parte de Internet. El número de *Host* identifica a un *Host* en la red y es asignado por el administrador de la red local.

1.4.5. Formato de Dirección *IP*.

La dirección *IP* de 32 bits se agrupa en 8 bits a un mismo tiempo, separados por puntos y representados en formato decimal. Cada bit en el octeto tiene un peso binario (128,64,32,16,8,4,2,1). El valor mínimo de un octeto es 0, y el valor máximo de un octeto es de 255. A continuación se muestra el formato básico de una dirección *IP*.

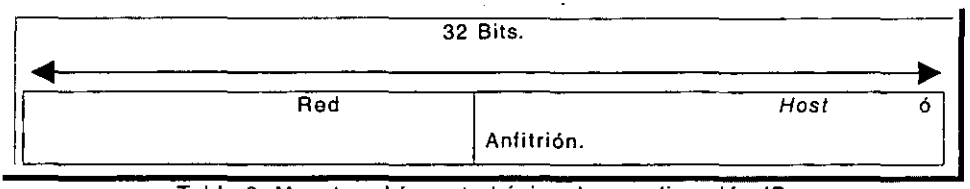

Tabla 2. Muestra el formato básico de una dirección *IP*.

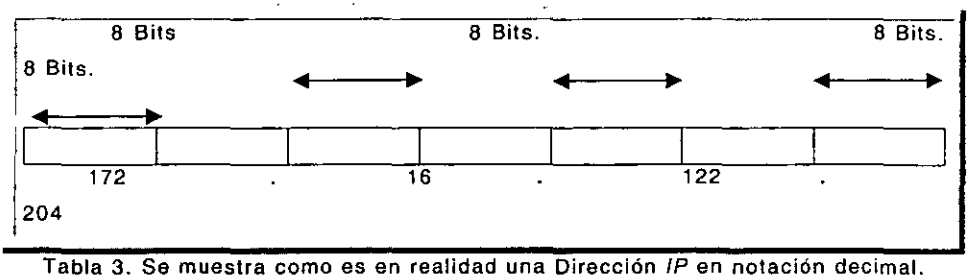

Tabla 3. Se muestra como es en realidad una Dirección *IP* en notación decimal.

1.4.6. Clases de Direcciones en *IP*.

El direccionamiento *IP* soporta cinco diferentes clases de direcciones: A,B,C,D y E. Solamente las clases A,B y C están disponibles para el uso comercial. Los bits de más a la izquierda indican de que clase de red se trata.

Clase de Dirección *IP*	Formato	Propósito.	Bits de orden superior	Rango de Direcciones	No. De Bits del *Host*/de Red.	Máximo de *Host*
A	R.H.H.H	Organizaciones muy grandes	0	1.0.0.0 a 126.0.0.0	7/24	16777214 2 = 2 24 - 2
B	R.R.H.H.	Organizaciones medianas.	1,0	128.1.0.0 a 191.254.0.0	14/16	65543 = 2 16 - 2
C	R.R.R.H	Organizaciones pequeñas.	1,1,0	192.0.1.0 a 223.255.254.0	22/8	245 = 2 8 - 2
D	N/A	Grupos de Multidifusión.	1,1,1,0	224.0.0.0 a 239.255.255.255	N/A	N/A
E	N/A	Experimental.	1,1,1,1	240.0.0.0 a 254.255.255.255	N/A	N/A

Tabla 4. Información de referencia respecto a las cinco clases de direcciones *IP*.

La clase de dirección se puede determinar fácilmente al examinar el primer octeto de la dirección y mapear ese valor con un rango de clases en la tabla anterior. En una dirección *IP* 192.168.1.1, por ejemplo, el primer octeto es 192, como 192 esta entre 192.0.1.0 a 223.255.254.0 es una dirección de clase C.

1.4.7. Direccionamiento de la subred *IP*.

Las redes *IP* se pueden dividir en redes pequeñas llamadas subredes. Las subredes representan varias ventajas para el administrador de red, entre ellas: una mayor flexibilidad, un uso más eficiente de las direcciones de red y la capacidad de manejar tráfico de difusión.

Las subredes están bajo una administración local. Como tales, el mundo exterior ve una organización como una sola red y no tiene un conocimiento detallado de la estructura interna de la organización.

Una determinada subdirección de red puede subdividirse en muchas subredes. Por ejemplo, 172.16.1.0, 172.162.2.0 y 172.16.3.0 son subredes dentro de la red 171.16.0.0.

1.4.8. Mascara de Subred *IP*

Una dirección de subred se crea pidiendo bits prestados del campo *Host* y designándolos como un campo de subred. El número de bits prestados varía y está especificado por la máscara de subred. Aquí mostramos cómo se piden prestados los bits del campo de dirección del *Host* para crear el campo de dirección de la subred.

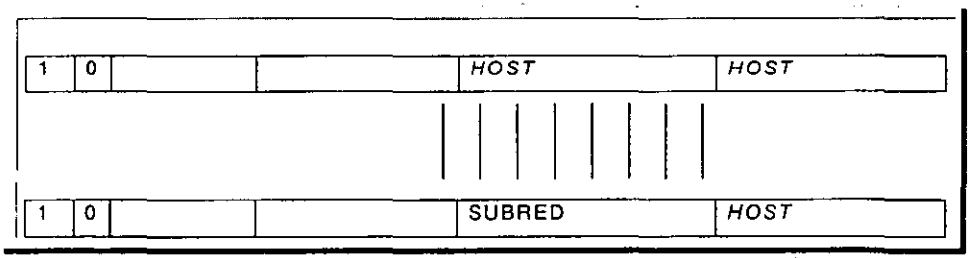

Tabla 5. En esta tabla podemos visualizar, que para poder crear el campo de dirección de la subred se piden bits prestados del campo de dirección *host*.

Las mascaras de subred utilizan el mismo formato y técnica de representación que las direcciones *IP*. Sin embargo, la máscara de subred tiene 1 binarios en todos los bits, los cuales especifican los campos de red y subred y 0 binarios en todos los bits que especifican el campo *Host*.

Los bits de la máscara de subred deben provenir de los bits de orden superior del campo del *Host* , como se muestra en la Tabla 6.

128	64	32	16	8	4	2	1		
1	0	0	0	0	0	0	0	=	128
1	1	0	0	0	0	0	0	=	192
1	1	1	0	0	0	0	0	=	224
1	1	1	1	0	0	0	0	=	240
1	1	1	1	1	0	0	0	=	248
1	1	1	1	1	1	0	0	=	252
1	1	1	1	1	1	1	0	=	254
1	1	1	1	1	1	1	1	=	255

Tabla 6. Nos da la visualización gráfica de que los bits de la máscara de subred provienen de los bits de mayor orden del campo *host*.

1.4.9. Uso de las máscaras de subred para determinar el número de red.

Un *Router* desempeña un proceso de activación para determinar la dirección de red. Primero, obtiene la dirección destino de *IP* del paquete entrante y recupera la máscara de la subred interna. Después realiza una operación lógica AND para obtener el número de la red. Esto hace que se elimine la porción del *Host* de la dirección destino de *IP*, mientras el número de la red de destino se conserva intacto.

Luego, el *Router* ve el número de la red destino y lo relaciona con una interfase de salida. Por último, direcciona la trama hacia la dirección *IP* destino.

Dirección *IP* destino	171.16.1.2	Red	Subred	*Host*
			00000001	00000010
Máscara de la subred	255.255.255.0		11111111	00000000
			00000001 AND	00000000 AND
			1	0

Tabla 7. En ésta tabla podemos observar que al aplicar una operación AND lógica entre la dirección *IP* de destino y la máscara de la subred se obtiene el número de subred.

1.4.10. Protocolo *TCP*

Este protocolo permite la transmisión confiable de datos en una ambiente *IP*. El protocolo *TCP* corresponde a la capa de Transporte – 4 del modelo de referencia *OSI*. Entre los servicios que ofrece *TCP* están la transferencia de datos en ráfagas, confiabilidad, control de flujo eficiente, operación full-duplex y multiplexaje.

Con el servicio de transferencia de datos por ráfagas, el protocolo *TCP* entrega una ráfaga no estructurada de bytes identificada por una secuencia de números. Este servicio beneficia a las aplicaciones, ya que éstas no tienen que fragmentar los datos en bloques antes de entregarlos a *TCP*. *TCP* agrupa los bytes en segmentos y los pasa al protocolo *IP* para su entrega.

El protocolo *TCP* ofrece la función de confiabilidad al permitir una entrega de paquetes confiable, de extremo a extremo, orientado a la conexión a través de una interred. Realiza esto colocando los bytes en secuencia con un número de confirmación de envío que indica al destino el próximo byte que el origen espera recibir. Los bytes que no

se confirman en un periodo específico se transmiten de nuevo. El mecanismo de confiabilidad de *TCP* permite que los dispositivos puedan lidiar con paquetes mal leídos, duplicados, retrasados o perdidos. Un mecanismo de expiración de tiempo permite a los dispositivos detectar paquetes perdidos y solicitar su retransmisión.

El protocolo *TCP* ofrece un control de flujo eficiente, lo cual significa que cuando se envían confirmaciones de regreso al origen, el proceso *TCP* de recepción indica el número de secuencia más grande que puede recibir sin saturar sus dispositivos de almacenamiento internos.

La operación dúplex total significa que los procesos de *TCP* se pueden enviar y recibir al mismo tiempo. Y por último, el multiplexaje de *TCP* significa que es posible multiplexar varias conversaciones de las capas superiores de manera simultanea a través de una sola conexión.

1.4.11. Establecimiento de la conexión.

Para utilizar un servicio de transporte confiable, los Hosts *TCP* deben establecer una sesión orientada a la conexión entre sí. La conexión se establece por medio de un mecanismo de saludo en tres direcciones.

Un saludo en tres direcciones sincroniza ambos extremos de una conexión permitiendo que ambos lados convengan en cuanto a los números de secuencia iniciales. Este mecanismo también garantiza que ambos lados estén listos para transmitir datos y saber cada uno que el otro lado también está listo para transmitir. Esto es necesario para que los paquetes no se transmitan o retransmitan durante el establecimiento de la sesión o después de que la sesión haya terminado.

Cada *Host* selecciona de manera un número aleatoria un número de secuencia que se utiliza para rastrear los bytes dentro de la ráfaga que está enviando y recibiendo. Posteriormente, el saludo en tres direcciones procede de la manera siguiente:

El primer *Host* (A) inicia una sesión enviando un paquete con el número se secuencia inicial (X) y el bit SYN activado para indicar una solicitud de conexión. El segundo *Host* (B) recibe el SYN, graba el número de secuencia (X), y responde confirmando el SYN con un (ACK= X+1). El *Host* (B) incluye su propio número de secuencia inicial (SEQ=Y). Un ACK= 20 significa que el *Host* ha recibido los bytes 0 al 19 y espera 20 a continuación. A esta técnica se le llama confirmación hacia delante. El *Host* (A), posteriormente, confirma todos los bytes que el *Host* (B) envió con una confirmación adelante que indica el siguiente byte que el *Host* (A) espera recibir (ACK = Y + 1). Sólo en este momento puede comenzar la transferencia de datos.

A continuación se visualiza de manera gráfica lo antes mencionado:

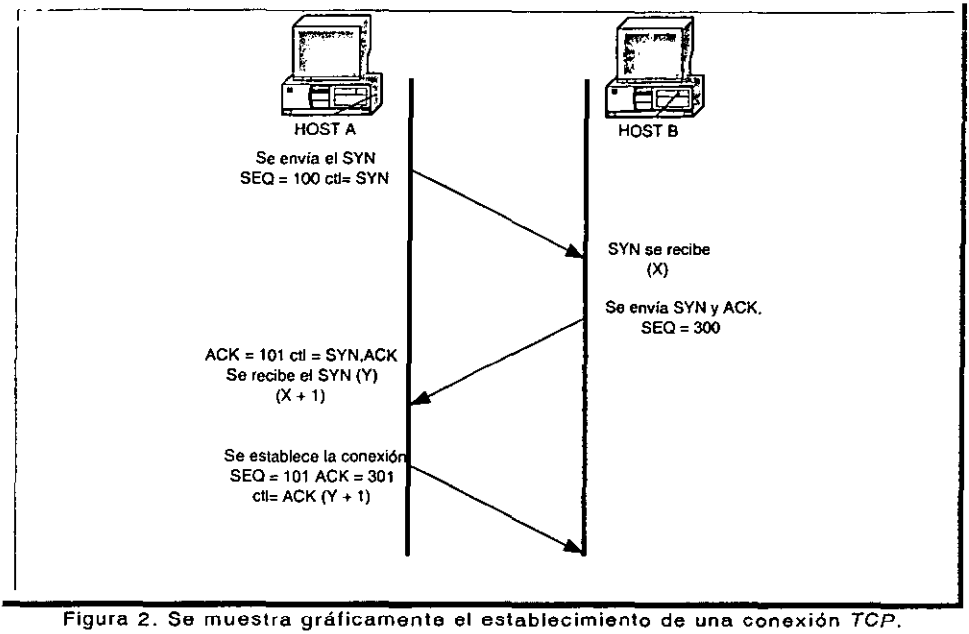

Figura 2. Se muestra gráficamente el establecimiento de una conexión *TCP*.

1.4.12. Formato de Paquete *TCP*.

Este es en forma gráfica el paquete *TCP*:

32 Bits.

Puerto Origen			Puerto Destino.
Número de secuencia.			
Número de confirmación.			
Desplazamiento de los datos.	Reservado.	Marcadores.	Ventana.
Suma de verificación.			Apuntador urgente.
Opciones (+ relleno).			
Datos (Variable.			

Tabla 8. Se puede observar los doce campos que componen un paquete *TCP*.

- Puerto Origen y Puerto Destino: Identifican los puntos en que los procesos de origen y destino de las capas superiores reciben los servicios *TCP*.
- Número de Secuencia: En general, especifica el número que se asigna al primer byte de datos del mensaje actual. En la fase del establecimiento de la conexión, este campo también puede utilizarse para identificar un número de secuencia inicial que será utilizado en una transmisión futura.
- Número de confirmación: Contiene el número de secuencia del siguiente byte de datos que el emisor del paquete espera recibir.
- Desplazamiento de datos: Indica el número de palabras de 32 bits en el encabezado de *TCP*.
- Reservado: Permanece reservado para su uso en un futuro.
- Apuntadores: Transportan una gran variedad de información de control, incluyendo los bits de SYN y ACK utilizados para el establecimiento de la conexión y el bit FIN que se utiliza para la terminación de la conexión.
- Ventana: Especifica el tamaño de la ventana del receptor del emisor, esto es el espacio de almacenamiento disponible para los datos entrantes.
- Suma de verificación: Indica si el encabezado se daño durante su viaje.
- Apuntador urgente: Apunta hacia el primer byte de datos urgente en el paquete.
- Opciones: Especifica las diferentes opciones de *TCP*.
- Datos: Contiene información de las capas superiores.

CAPÍTULO II

2. TIPOS Y FUNCIONAMIENTO DE UN *FIREWALL*.

Una vez que sentamos bases de conectividad, procederemos a profundizar un poco en una de las alternativas más difundidas y utilizas hoy en día en seguridad informática como lo son los *Firewalls* o Pared de Fuego, para uso de este trabajo se utilizará el primer término debido a su contracción y popularidad del término a nivel mundial. Será interesante poder describir los dos principales tipos de *Firewalls* que el primero es a nivel de red y el segundo a nivel de aplicación. Se analizará las dos caras de la moneda, que son los beneficios y limitaciones de un *Firewall*, se explicara brevemente algunos de los servicios soportados por la gran mayoría de los *Firewalls* hoy en día. Y punto a destacar serán los *Routers* de selección, que no son nada más un simple y llano *Router*, sino una alternativa muy apropiada y eficaz para que nuestra zona militarizada se vaya convirtiendo verdaderamente en una política de calidad. Por último entraremos a explicar en que consiste el modelo de filtración de paquetes, que es el modelo más difundido dentro de los fabricantes de *Firewalls*, así como el uso de las listas de acceso tanto simples como entendidas. Bajo este panorama estamos facultados para poder ir armando de una manera muy sencilla nuestro sistema *Firewall* que más tarde se convertirá en política de calidad.

2.1. Definición.

Un Firewall podemos definirlo de varias maneras tales, como un separador, un limitador y hasta un analizador. Su implementación física varía de un sitio a otro, esto se analizará un poco más adelante. Entonces un *Firewall* lo podemos definir como un conjunto de componentes de hardware y software que establecen una política de control de acceso entre dos redes.

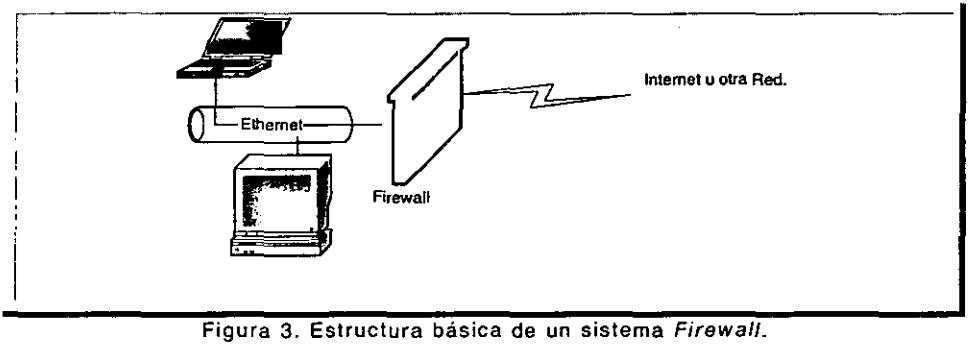

Figura 3. Estructura básica de un sistema *Firewall*.

Conceptualmente, hay dos tipos de *Firewalls* básicamente:

- Nivel de Red.
- Nivel de Aplicación.

No hay tantas diferencias entre los dos tipos como se podría pensar. Además las últimas tecnologías no aportan claridad para distinguirlas hasta el punto que no está claro cual es mejor y cual es peor.

2.1.1. *Firewalls* a Nivel de Red.

Generalmente los *Firewalls* a nivel de red, toman las decisiones de filtrar basándose en la fuente, dirección de destino y puertos, todo ello en paquetes individuales *IP*. Un simple *Router* tradicional entre otras funciones es un Firewall a nivel de red., particularmente desde el momento que no puede tomar decisiones sofisticadas en relación con quien está hablando un paquete ahora o desde donde está llegando en este momento. Los modernos *Firewalls* a nivel de red, se han sofisticado ampliamente, y ahora mantienen información interna sobre el estado de las conexiones que están pasando a través de ellas, los contenidos de algunos datagramas y más cosas. Un aspecto importante que mencionar de los *Firewalls* a nivel de red es que ellos enrutan el tráfico directamente a través de ellas, de forma que un usuario cualesquiera necesita tener un bloqueo válido de dirección *IP* asignado. Los *Firewalls* a nivel de red tienden a ser más veloces y más transparentes a los usuarios.

2.1.2. *Firewalls* a Nivel de Aplicación.

Son generalmente, Hosts que corren bajo *servidores Proxy*, que no permiten el tráfico directo entre redes y que realizan verificación de nombres de usuarios y auditan el tráfico que pasa a través de ellas. Los *Firewalls* a nivel de aplicación se pueden usar como traductoras de direcciones de red, desde que el tráfico entra por un extremo hasta que sale por el otro. Los modernos *Firewalls* a nivel de aplicación son bastantes transparentes, además de que tienden a proporcionar mayor detalle en los informes auditados e implementan modelos de conservación de seguridad., y ésta es básicamente la diferencia entre un *Firewall* de nivel de red y uno de aplicación.

2.1.3. Beneficios de un *Firewall*.

Los *Firewalls* administran los accesos posibles de una red exterior por ejemplo Internet a una red interna o privada. Sin un *Firewall*, cada unos de los servicios dentro de una red, tales como cualquier Sistema de Información se exponen a ataques de cualquier tipo de *servidor* externo a nuestra red. Esto significa que la seguridad en una red interna o privada depende de las políticas de seguridad que nuestra red cuente.

El *Firewall* permite al administrador de la red definir nuestro punto central, manteniendo al margen o fuera de la red a los usuarios no autorizados, prohibiendo potencialmente la entrada o salida los servicios de red, y proporcionar protección para varios tipos de ataques posibles. Uno de los beneficios clave de un *Firewall* es que ayuda a simplificar los trabajos de administración una vez que se consolida la seguridad en el sistema *Firewall*, además de que también se usa cuando la información en los servidores es muy delicada.

El *Firewall* ofrece un punto donde la seguridad puede ser monitoreada y si aparece alguna actividad sospechosa, esta generara una alarma y/o una acción ante la posibilidad de que ocurra un ataque, o suceda algún problema de tránsito de los datos.

Un *Firewall* también es hoy en día un lugar lógico para desplegar un *NAT* esto puede ayudar solucionando de alguna manera el espacio de direccionamiento.

2.1.4. Limitaciones de un *Firewall*.

Un *Firewall* no puede protegerse contra aquellos ataques que se efectúen fuera de su punto de operación.

Un *Firewall* no puede protegerse de las amenazas a que esta sometido por traidores o usuarios internos y además inconscientes. Un *Firewall* no puede prohibir que los usuarios traidores o inconscientes copien datos de configuración y extraigan esta información fuera de red.

Un *Firewall* no puede proteger contra ataques de la suplantación de usuarios y poder utilizar la red de manera temporal, aunque hay otros sistemas y mecanismos para ello.

La gran mayoría de los *Firewalls* actuales no puede protegerse contra los ataques posibles a la red interna por virus informáticos a través de archivos y software indeseables. Es por eso que es muy recomendable el instalar un *servidor Proxy*, como un excelente medio de prohibición de conexiones directas por usuarios externos y reduce el índice de amenazas posibles por los ataques con transferencias de datos, más adelante se aborda con mucho mayor detalle todo lo relativo a los *Servidores Proxy*.

2.1.5. ¿ Cuál es el mejor tipo de *Firewall* ?

Todos los tipos de *Firewall* que se han descrito en este trabajo que son los *Firewalls* de filtrado de paquetes y también los *Firewalls* a nivel de aplicación, son abundantes en el mercado, lo que da entender que no existe un *Firewall* óptimo. El *Firewall* más adecuado a un entorno determinado depende de diversos factores, como los conocimientos de los administradores, los tipos de servicios que se

pretenden soportar, el presupuesto y sobre todo las necesidades de la organización. Antes de decidir cuál es el *Firewall* que mejor se adapta a las necesidades propias, deben evaluarse las amenazas a las que está sujeta nuestra organización. Es necesario asimismo, desarrollar una política de seguridad, que más adelante en este trabajo se detallara y se comentará a profundidad este tema.

En el momento de determinar cuál es el *Firewall* más adecuado, tal vez la consideración más importante deba basarse en los tipos de servicio que se pretende soportar a través del mismo. Por dar algún ejemplo, quizá se desee soportar *FTP*, *Telnet*, correo electrónico y *HTTP*. Cada servicio posee su propio conjunto de puntos débiles y protecciones disponibles. Independientemente del *Firewall* que se adquiera, éste debe incluir las características más adecuadas que proporcionen la suficiente protección a los servicios que se pretende soportar.

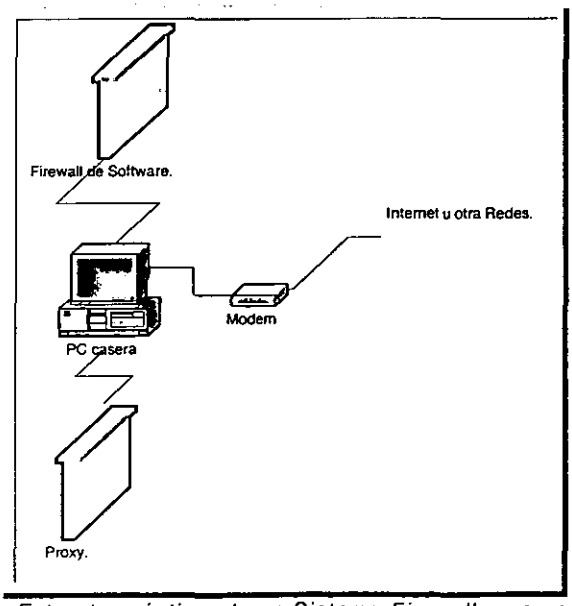

Figura 4. Estructura óptima de un Sistema *Firewall* para un casero.

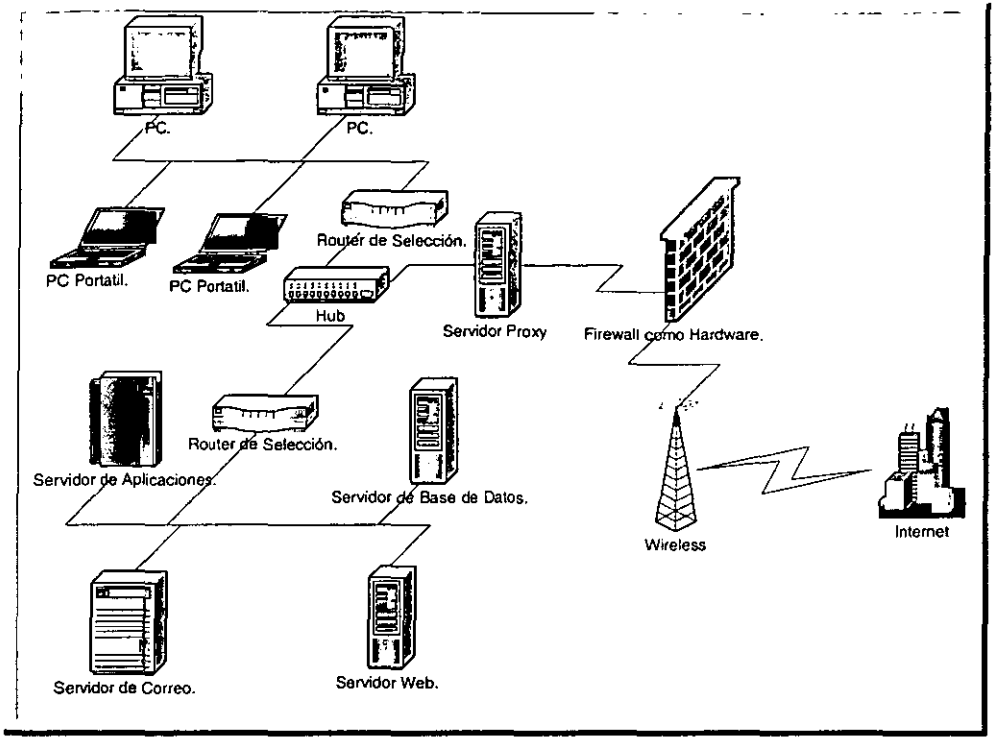

Figura 5. Posible estructura de un Sistema *Firewall* para un *ISP*.

2.1.6. *Firewalls* Basados en Redes.

Los tipos de *Firewalls* que se han mencionado hasta este momento en este trabajo, filtrado de paquetes y a nivel de aplicación se basan esencialmente en instalaciones físicas. En un *Firewall* de este tipo, todos los componentes protectores del mismo están situados en las instalaciones de la organización o compañía a proteger. En un futuro no muy lejano, de hecho algunas compañías ya lo tienen, los *Firewalls* podrán pasar de ser un producto a un servicio que ofrecerán tanto los *ISP* como los *ASP*. Dicho servicio se proporcionará mediante un *Firewall* basado en red.

Un *Firewall* basado en red está situado en la red del proveedor de servicios de acceso, al cual la organización estará conectada. No debe de perderse de vista que el *Firewall* basado en red deberá contemplar múltiples políticas de seguridad, para cada cliente e impedir que se mezclen con la de otros clientes.

Los *Firewalls* basados en red tendrán que superar dos obstáculos principales antes de ponerse en marcha comercialmente. El primero de ellos es la velocidad en pocas palabras en rendimiento. Debido a que el Firewall de red habrá de gestionar o administrar el tráfico de diversas redes, o también conocidas actualmente como Intranets, deberá de soportar un elevado rendimiento total de procesamiento. El segundo problema al que se enfrenta es impedir que cualquier *Router* situado entre el cliente y el *Firewall* pueda evitar el *Firewall* y permitir el acceso incontrolado a la red o Intranet del cliente.

Con toda probabilidad, los proveedores de servicios de acceso que ofrezcan el servicio de *Firewall* basados en red, conseguirán superar estos obstáculos y así suministrar el servicio sin ningún tipo de inconveniente. Es muy posible por la idiosincrasia mexicana que algunas organizaciones o compañías se sientan incómodas por tener en manos de extraños la protección de su red o Intranet. En la actualidad, sólo algunas de las organizaciones o compañías tienen su propio Firewall basado en instalaciones físicas, y esta tendencia parece que no habrá de desaparecer por completo.

2.1.7. Diferencias en la Administración.

Otra diferencia muy importante entre los diversos tipos de *Firewalls* en la actualidad es el método empleado para la administración de los mismos. Como se ha mencionado ya en este

trabajo, la protección que ofrece un *Firewall* puede llegar a ser contraproducente si no es configurado y/ administrado correctamente. Una interfaz gráfica y fácil de utilizar y con un número reducido de opciones de configuración reduce en gran medida la posibilidad de que se produzcan errores de administración, naturalmente un número reducido de opciones en la configuración también es una arma de dos filos ya que puede significar también menor flexibilidad de configuración.

Existen tres clases de interfaz del administrador de *Firewalls*:

- Administración basada en archivos de texto.
- Administración basada en menús de texto.
- Administración basada en GUI.

La interfaz basada en archivos de texto es la de uso más extendido y generalizada dentro de las organizaciones alrededor del mundo actualmente en lo que respecta a los *Routers* y a los *Firewalls* de instalaciones físicas. Este tipo de interfaces permite al administrador editar un archivo en específico donde puede introducir parámetros de configuración específicos. Se trata de le interfaz de elección para los administradores principalmente que trabajan sobre sistemas UNIX tradicionales, dado que ofrece una interfaz de control a bajo nivel con los mecanismos del *Firewall*. La desventaja de dicho control a bajo nivel es que resulta mucho más fácil a cometer errores, ya que, el editar un archivo, pueden producirse errores de escritura u otros errores técnicos que, en un sistema basado en menús es menos probables que ocurran.

La interfaz de administrador basada en menús de texto presenta un menú de texto valga la redundancia que reduce la posibilidad de producirse errores pero que proporciona menor capacidad de control

para el administrador. Sin embargo, la posibilidad de error no queda totalmente excluida, dado que el administrador no siempre puede ver el efecto de algunos cambios.

La interfaz gráfica de usuario o GUI, para administradores incorpora ventanas, botones, menús desplegables y pantallas de ayuda que facilitan el trabajo de configuración y administración. La mayoría de los proveedores de *Firewalls* ha adoptado por incluir esta interfaz en sus productos, puesto que tiende a ser más fácil de utilizar y no es susceptible a muchos errores que pueden producirse en los otros dos tipos de interfaz.

Algunos productos *Firewalls* ofrecen la posibilidad de realizar la administración centralmente, lo que permite configurar múltiples *Firewalls* desde una ubicación individual remota. Esta característica puede ser importante si existen múltiples *Firewalls* situados en diversas ubicaciones pero se dispone solamente de un especialista en *Firewalls* en una de esas ubicaciones. La administración centralizada permite a este especialista configurar cada uno de los *Firewalls* desde un sitio central.

2.1.8. Control Básico de Acceso.

La función primordial de un *Firewall* es controlar el acceso a la red en función de la dirección del *Host* y del servicio solicitado. Por ejemplo, puede utilizar el control básico de acceso para permitir al *Host* 192.168.1.1 acceder a la red mediante el servicio de *Telnet*.

Todos los *Firewalls* disponibles ofrecen un mecanismo para controlar el acceso. También ofrecen otras características para reforzar el control

de acceso, facilitar la administración del mismo y hacerlo más difícil de engañar.

Los *Firewalls* de filtrado de paquetes o de sesiones emplean reglas de acceso para definir el control de acceso desde y hacia la red. Las reglas de acceso pueden soportar cualquier servicio. Por ejemplo, para soportar el servicio de *Telnet*, una regla solamente tiene que autorizar el acceso al puerto 23. Asimismo, esta misma flexibilidad también puede hacer más compleja la administración de las reglas, ya que especificar un número de puerto incorrecto va dejar una puerta abierta para la entrada de un intruso.

Los *Firewalls* a nivel de aplicación, emplean dos tipos de listas de acceso, las basadas en los Hosts y las basadas en los servicios. Las primeras describen conjuntos de servicios autorizados para cada *Host* o red. Las segundas identifican los conjuntos de *Host* o redes que pueden utilizar cada uno de los servicios. Las listas de acceso solamente pueden soportar políticas de seguridad simples, además resulta más sencillo comprenderlas y configurarlas. Sin embargo, muchos administradores de *Firewalls* prefieren la flexibilidad de las reglas de acceso, en la medida en que éste disponible una interfaz de administrador adecuada. Más adelante en este mismo capítulo se explicaran más a detalle en que consisten las listas de acceso.

2.2. Tipos de Ataque.

Dentro de las varias opciones que tiene las personas indeseadas, para poder hacer daño a nuestros Sistemas de Información tenemos diferentes tipos como se muestran a continuación:

- Ataque DoS: Es el acrónimo de la expresión inglesa Deny of Service, que significa denegación del Servicio. Este tipo de ataques tiene por objeto inutilizar la Pc atacada, de forma que deje de responder o haya que reiniciarla. No se consigue penetrar dentro del sistema, pero se ocasionan muchísimos problemas, tales como saturar el ancho de banda, de forma que se disminuya mucho la velocidad de la red.
- Caballos de Troya: Se trata de programas que suelen regalar o enviar por correo como si fueran inofensivos, juegos o bromas. En realidad, una vez introducidos en las máquinas, permiten a un intruso tomar el control de los sistemas.
- Barrido de Puertos: Un barrido de puertos trata de identificar que puertos *TCP* o *UDP* están abiertos en nuestra Pc, para poder aprovechar ciertos servicios que dependen de ellos para entrar en los Sistemas. Existen en la actualidad una infinidad de herramientas de barrido de puertos accesibles en la Internet y ésta será una de las primeras cosas que compruebe un atacante.
- Detección de *Proxies*: Muchos de los atacantes buscan *Proxys* que puedan estar instalados delante de nuestras Pc. Un *Proxy* sirve, básicamente para compartir una conexión a Internet entre varios PC´s. El intruso o atacante desea encontrar *Proxys* mal configurados, para convertir sus ataques en anónimos o invisibles. Dado que hoy en día casi la mayoría de los servidores llevan un registro de las direcciones *IP* desde las que se accede a ellos, si un individuo malintencionado quisiera atacarlos sería fácil seguirle la pista a través de su dirección *IP*. Sin embargo, utilizando un *Proxy* ajeno enmascaría su verdadera *IP*, el ataque parecería que proviene de dicho *Proxy*, ocultando las huellas del verdadero agresor.
- Ataques Smurf/Fraggle: Bajo el nombre de los graciosos personajes de dibujos animados, el significado de un Smurf es

Pitufo, se esconde una técnica de ataque masivo a servidores utilizando para ello a usuarios inocentes que no están bien protegidos. Concretamente el ataque se basa en el envío de paquetes de difusión de máquinas de una subred. Estos paquetes llevan falseada la dirección de origen, apuntando en realidad a un servidor que desea atacar. Todas las máquinas de la subred responden a los paquetes falsos dirigiéndolos a la víctima, la cual se ve sobrecargada instantáneamente por el enorme efecto multiplicador debido a la difusión. La diferencia entre el ataque Smurf y el ataque Fraggle estriba en el tipo de paquete enviado.

2.3. Servicios Soportados.

Los servicios se refieren a los protocolos a nivel de aplicación que el *Firewall* reconoce y autoriza. Es muy importante que los *Firewall* nieguen cualquier servicio que no puedan reconocer. Los servicios se identifican mediante el número del puerto destino *TCP* o *UDP*. Estos servicios tienen números de puertos conocidos que son fijos como por ejemplo *Telnet* = Puerto 21.

A continuación se listan una serie de servicios que a mi consideración son los más básicos y soportados por la mayoría si no es que su totalidad de los *Firewalls* de la actualidad, además de colocar una breve descripción de cada uno de ellos.

2.3.1. *DNS* = Puerto 53 de *TCP*.

Por lo general el *DNS* no es un servicio que utilicen directamente los usuarios. Cuando un usuario solicita conectarse a un *Host*, la aplicación de red llama al *DNS* para averiguar la dirección *IP* asociada al *Host*. Si el *servidor DNS* local del usuario no tiene esta información,

la solicita a otros *servidores DNS*. Los *servidores DNS* comparten información. Es precisamente esta capacidad de la que debemos protegernos, debido a que no es deseable que ningún *servidor DNS* de Internet pueda actualizar los nombres de *servidor* de la red interna o de la Intranet. En una situación de este tipo, los intrusos o atacantes pueden redefinir la dirección de un *Host* externo a la red interna o Intranet con una dirección de confianza de la red también denominada no homologadas.

El *Firewall* puede permitir a los *servidores DNS* de la red propia el acceso a *servidores* de nombres del exterior e incluso actualizarlos con direcciones nuevas, negando a su vez a los externos poder actualizar los registros de los *servidores* de nuestra red.

Los *Firewalls* a nivel de aplicación desprovistos de la capacidad de un *Proxy* invisible, no necesitan soporte *DNS* porque realizan la consulta directamente a un *servidor* de nombres externo. Más adelante en el capítulo dedicado a los *Proxys* se aclararán todos los conceptos relacionados con este tema.

2.3.2. Finger = Puerto 79 de *TCP*.

El servicio de finger fue desarrollado para permitir a los usuarios de una red poder localizar a otros usuarios. Gracias a un finger es posible averiguar nombres de entrada en el sistema nombres de usuarios, y los nombres reales de los usuarios. Esta información es valiosa para un atacante o intruso potencial, por lo que el *Firewall* debe prohibir cualquier solicitud de finger procedente del exterior. Una alternativa para descartar las solicitudes de un finger procedentes del exterior es tener un *Proxy* de finger que muestre un mensaje de error como puede ser. "Esta red no soporta el servicio de finger".

A menudo los intrusos o atacantes emplean solicitudes finger a modo de sondeo. Por esta razón, algunos administradores instalan un *Servidor Proxy* de finger responde a una solicitud finger efectuándose a su vez una solicitud inversa a fin de obtener información referente del solicitante. Sin embargo, hay que tener cuidado con esta política porque la solicitud inversa puede ser atrapada por un *servidor* finger que haya sido diseñado también para efectuar una solicitud finger inversa. El resultado será un ciclo infinito de solicitudes finger entre ambos *Host*, lo cual bloqueará innecesariamente recursos de los dos sistemas.

2.3.3. *FTP* = 21 de *TCP*.

FTP, se trata del protocolo estándar para transferir archivos entre sistemas que soporta una autentificación sencilla de contraseñas.

Para cada archivo *FTP* o transferencia de información, se establece habitualmente una conexión de red a parte desde el *Host* de destino hacia el *Host* desde donde se origina la conexión *FTP*. El *Firewall* debe ser capaz de permitir la segunda conexión en el sentido contrario ya que si no, no se transferirán los datos. Por lo general, un puerto *TCP* de 20 identifica la conexión de datos.

2.3.4. Gopher = Puerto 70 de *TCP*.

El producto y servicio Gopher proporciona un sistema sencillo de menús textuales cuya función es ayudar a encontrar información en Internet.

Gopher es uno de los precursores del HTML, y los *servidores* y clientes Gopher plantean las mismas amenazas que los clientes y *servidores*

HTTP. Por ejemplo, es posible engañar a un cliente gopher para que ejecute órdenes no autorizadas en la computadora del usuario. Al igual que HTML, los *servidores* gopher pueden estar disponibles en otros puertos además del puerto predeterminado.

2.3.5. *ICMP*.

ICMP es un protocolo soportado por encima de *IP*, a nivel de *TCP* o *UDP*. *IP* lo utiliza para enviar mensajes de error o de prueba entre sistemas distintos. Un mensaje *ICMP* contiene campos de tipo y de código que indican un mensaje predefinido como "No se puede contactar la red" o "Acceso denegado para propósitos de administración.

La conocida aplicación *ping* emplea el protocolo *ICMP* para enviar mensajes de petición de eco QCMP tipo 8, código 0, para comprobar si es posible acceder a un *Host*. El *Host* destino responde con un mensaje de respuesta, de echo (*ICMP* tipo 0 código 0). El *Firewall* puede configurarse para permitir algunos mensajes *ICMP* y denegar otros. Por ejemplo, tal vez se desee impedir a los *Host* de Internet hacer un *ping* en la red propia para averiguar que Hosts pueden ser atacados. Por otra parte, quizá sea deseable que los mensajes de error regresen a los Hosts de la red.

2.3.6. *IRC* = Puerto 6667 de *TCP*.

La aplicación *IRC* ofrece la posibilidad de participar en conferencias con múltiples usuarios en un entorno de texto. Con una aplicación *IRC* cliente, un usuario puede ponerse en contacto con un *servidor IRC* y unirse a una conversación.

La principal amenaza asociada a este servicio no es inherente al protocolo, sino que representa más bien una amenaza de ingeniería social. Entre otras cosas , la Ingeniería Social es el acto que puede perpetrar un atacante para obligar a un usuario o administrador a que proporcione información de autenticación o a que reduzca controles de seguridad. Un usuario de Internet puede intentar convencer a un usuario interno para que modifique la configuración de su computadora a fin de proporcionar una característica determinada. Dicha modificación puede ser un método del que le servirá al usuario externo para penetrar en la computadora del usuario interno. Un *Proxy IRC* instalado es de gran utilidad para contrarrestar esta amenaza.

2.3.7. *E-mail* = Puerto 25 de *TCP*.

El correo electrónico es el servicio más utilizado en Internet. Permite a los usuarios enviar mensajes sin que sea necesario establecer una conexión directa entre el *Host* remitente y el *Host* destinatario. Un mensaje de correo puede recorrer un gran número de *Host* antes de llegar a su destino. El protocolo de correo estándar que se emplea es el *SMTP*.

2.3.8. Mbone = Protocolo *IP*.

Mbone se emplea para dirigir paquetes multitransmitidos a través de *Routers* que no soportan el direccionamiento de multitransmisión. Este es un mecanismo que sirve para retransmitir el mismo paquete *IP* hacia varios sitios de Internet. Se emplea en servicios de conferencia en tiempo real como radio en Internet y suministros de video.

Algunos *Routers* no soportan el direccionamiento de multitransmisiones. Mbone encapsula un paquete *IP* de transmisión única para que pueda

atravesar un *Router* que no soporta la multitransmisión. El número de protocolo indicado en el campo de protocolo *IP* es *IP* 4. Este paquete *IP* incluido en *IP* es recibido por otro *Router* de multitransmisión que suprime el paquete *IP* multitransmitido original y lo vuelve a enviar.

La amenaza que supone permitir la entrada de paquetes Mbone en la red es nuestro desconocimiento del protocolo y servicio del paquete *IP* interno. Un *Firewall* que soporte Mbone debería, como mínimo, examinar la dirección del paquete *IP* interno y asegurarse de que es un paquete multitransmitido. También sería útil poder realizar un filtrado según el protocolo y servicio del paquete *IP* interno.

2.3.9. Network News = Puerto 119 de *TCP*.

Networks News es otro servicio de uso bastante generalizado. Permite a los usuarios acceder a un newsgroups a fin de leer información o de participar en debates. Los newgroups constan de una serie de mensajes que tienen un tema común. Los usuarios pueden leer estos mensajes y agregar los suyos propios. Existe un newsgroup para prácticamente cualquier tema imaginable. Y el protocolo empleado es el NNTP.

2.3.10. *NFS* = Puerto 2049 de *UDP*.

NFS permite a los usuarios compartir sistemas de ficheros con otros usuarios. Este tipo de característica es un estándar de las redes de PC como Novell Netware o Microsoft Windows. El *NFS* estándar proporciona muy poca seguridad y, por eso, es vulnerable a los ataques. La mayoría de expertos en *Firewalls* recomiendan no permitir conexiones *NFS* a través de la red, aunque muchos administradores de las mismas o de Intranets no están dispuestos a prescindir de este

servicio. De todos modos, los *Firewalls* a nivel de aplicación no soportan este servicio y el filtrado de paquetes no elimina el riesgo que trae el mismo.

2.3.11. RPC Puerto 111 de *TCP* o *UDP*.

Las aplicaciones RPC emplean un asignador de puerto para obtener el número de puerto *TCP* o *UDP* actual de un servicio. Si bien se dice que las aplicaciones *servidor* utilizan números de puerto conocidos, esto no es siempre cierto. Algunas aplicaciones *servidor* pueden ejecutarse en cualquier número de puerto y dependen de la aplicación *servidor* de asignador de puerto para dirigir los clientes hacia el número de puerto apropiado. Esto dificulta enormemente la posibilidad de que el *Firewall* pueda realizar el filtrado o establecer un *Proxy* para dichas aplicaciones, puesto que el número de puerto de las últimas puede cambiar en cualquier momento.

2.3.12. rLogin = Puerto 513 de *TCP*.

El rLogin son utilizados para acceder de un sistema local a otro remoto. Pero no se recomienda su uso para acceder a / o desde Internet porque la mayoría de ellos no soportan funciones adecuadas para la autentificación de usuarios.

2.3.13. *Telnet* = Puerto 23 del *TCP*.

Telnet es el protocolo y aplicación estándar para la entrada en sistemas remotos. Proporciona una conexión entre dos sistemas basadas en carácter.

2.3.14. *WWW* = Puerto 80 y otros de *TCP*.

Probablemente *www* es la principal responsable del repentino Interés y expansión de Internet actualmente. El principal protocolo de servicio empleado por la web es el *http*, que permite a los usuarios transferir documentos desde un *servidor http*. Este protocolo está soportado por aplicaciones clientes gráficas conocidas como navegadores o exploradores. Son varios los problemas de seguridad que se han relacionado con el *http* y los *servidores* y navegadores asociados al mismo.

2.3.15. X11 Puerto 6000 y superiores de *TCP*.

X11 se refiere a la especificación correspondiente al entorno gráfico de usuario, de un uso generalizado en estaciones de trabajo UNÍS. La mayoría de *servidores* X soportan más de un puerto X6001, y es posible emplear números de puertos superiores a este último.

El protocolo X11 es un servicio muy potente que permite a una aplicación remota presentar gráficos y aceptar órdenes de un ratón en una estación de trabajo X con interfaz de Windows. Sin embargo, esta potencia se proporciona a expensas de un cierto riesgo para la seguridad. La aplicación remota puede tomar completamente el control de la pantalla, del teclado e incluso del ratón. Se tiene previsto establecer conexiones X11 desde Internet, el *Firewall* debe poder soportar este tipo de operaciones, además de filtrar las conexiones o comandos X11 no deseados.

2.4. Introducción a los *Routers* de Selección.

Actualmente la mayoría de los *Routers* comerciales tiene la capacidad de seleccionar paquetes con bases a una serie de criterios, como el tipo de protocolo, los campos de dirección de origen y dirección destino para un tipo particular de protocolo y los campos de control que son parte del protocolo. A estos *Routers* se les llama *Routers* de selección. Esto pueden proporcionar un mecanismo poderoso para controlar el tipo de tráfico de red, que puede existir en cualquier segmento de una red. Al controlar ese tipo de tráfico, los *Routers* de selección pueden controlar servicios que puede existir en un segmento de red, por lo tanto, pueden restringirse servicios que pueden poner en peligro la seguridad de la red.

Los *Routers* de selección pueden discriminar entre el tráfico de red con base en el tipo de protocolo y en los valores de los campos del protocolo en el paquete. A la capacidad del *Router* para discriminar entre paquetes y restringirlos en sus puertos con base en criterios específicos de protocolo se le denomina filtración de paquetes.

2.4.1. Filtración de Paquetes.

Los *Routers* de selección puedenutilizar la filtración de paquetes como medio para mejorar la seguridad de la red. La función de selección también puede ser desarrollada por muchos productos de *Firewall* comerciales. Como por ejemplo, los fabricantes de *Routers* como Cisco, Wellfleet, 3Com, Digital Newbridge y otros proporcionan *Routers* que permiten programarse para desarrollar funciones de filtración de paquetes.

2.4.2. Modelo Simple para la Filtración de Paquetes.

Por lo general, un filtrado de paquetes se coloca entre uno o más segmentos de red. Estos segmentos de red están clasificados como segmentos de red externos o internos. Los segmentos de red externos conectan su red con redes externas como Internet. Los segmentos de red internos se utilizan para conectar los Hosts de la empresa y otros recursos de la red.

2.4.3. Operación de Filtración de Paquetes.

Casi todos los dispositivos de filtración de paquetes actuales (*Routers* de selección o gateways de filtración de paquetes) funcionan de la siguiente manera, como se describe a continuación:

Los criterios de filtración de paquetes deben almacenarse para los puertos del dispositivo de filtración de paquetes. A los criterios de filtración de paquetes se les llama reglas de filtración de paquetes. Cuando un paquete llega al filtro, se analizan los encabezados del paquete. La mayoría de los dispositivos de filtración de paquetes examinan los campos solo encabezados de *IP*, *TCP* o *UDP*.

Las reglas de filtración de paquetes se almacenan en un orden específico. Cada regla se aplica al paquete en el orden en que la reglas de filtración de paquetes se almacena.

Si una regla bloquea la transmisión o la recepción de un paquete, este no es permitido.

Si una regla permite la transmisión o la recepción de un paquete, a dicho paquete se le permite proceder.

Si un paquete no satisface alguna regla, se le bloquea.

A continuación se muestra el diagrama de flujo para la operación de la filtración de paquetes:

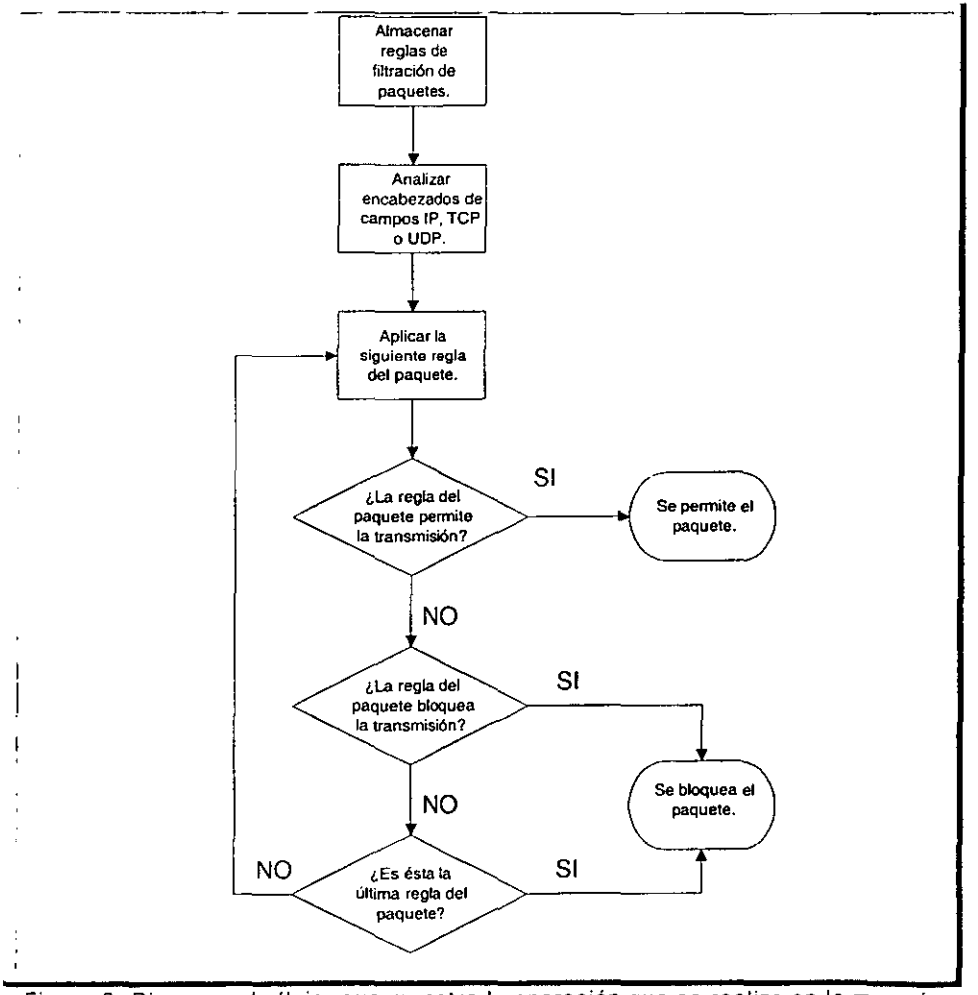

Figura 6. Diagrama de flujo, que muestra la operación que se realiza en la mayoría de los procesos de filtración de paquetes.

2.4.4. Diseño de la Filtración de Paquetes.

En muchas de las ocasiones se debe poner gráficamente la política de seguridad que se necesita implementar, posteriormente ésta debe traducirse en una regla formal de filtración de paquetes, a esta traducción de lo gráfico a una hoja de trabajo de filtración de paquetes se le llama Diseño de Filtración de paquetes.

2.4.5. Reglas de Filtración de Paquetes y Asociaciones Totales.

Los *Routers* de selección, en general, pueden filtrar con base en cualquier de los valores de campo que se encuentran en los encabezados del protocolo *TCP* o *IP*. Para la mayoría de las políticas de seguridad de redes pueden implementarse con *Routers* de selección, ya que solo se necesitan especificar los indicadores *TCP*, las opciones de *IP* y los valores de las direcciones origen y destino.

A continuación se muestra una hoja de trabajo que puede utilizarse para diseñar reglas de filtración de paquetes.

No. De regla de Filtración	Acción	*Host*/Red de Origen	Puerto de Origen	*Host* / Red Destino	Puerto de Destino	Opciones de indicadores de protocolo	Descripción

Tabla 9. Este es uno de los formatos que puede llegar a utilizarse para diseñar reglas de filtración de paquetes.

Si se revisan cada una de las filas de la hoja de trabajo, se observará que describe completamente la conexión de *TCP*. Formalmente, a una descripción completa de una conexión se le llama Asociación Completa. Cuando se diseñan reglas de filtración de paquetes es útil tener a

consideración las definiciones de asociación completa, media asociación y extremos.

2.4.6. Implementación de Reglas de Filtración.

Una vez que se han diseñado las reglas de filtración de paquetes y se han traducido a la hoja de trabajo correspondiente tienen que se implementadas en el *Router* de selección o bien el *Firewall*.

Cada Tipo de dispositivo de filtración de paquetes tiene su propio conjunto de reglas y de sintaxis para programar las reglas de filtración de paquetes. Por lo tanto, se debe estudiar la documentación del dispositivo y se debe aprender las peculiaridades de la sintaxis de las reglas de filtración de paquetes para ese dispositivo. El tipo de sintaxis que utilizaremos para este trabajo será con relación a los *Routers* de selección de Cisco, debido a que es el fabricante líder del mercado, con referencia a la filtración de paquetes para otros fabricantes son muy parecidas en cuanto a sus fundamentos pero sintácticamente diferentes.

2.4.7. Ventajas de la Filtración de Paquetes.

La mayoría de los *Firewalls* son diseñados únicamente utilizando filtración de paquetes .

El costo para implementar la filtración de paquetes no es cara, además de optimizar la operación del *Router* moderando el tráfico y definiendo menos filtros.

Finalmente, el filtrado de paquetes es transparente para los usuarios finales.

2.4.8. Desventajas del Filtrado de Paquetes.

Definir el filtrado de paquetes puede ser una tarea muy compleja porque el administrador de la red necesita tener un estudio detallado de varios servicios, como los formatos del encabezado de los paquetes y los valores específicos esperados a encontrarse en cada campo. Si las necesidades son muy complejas, se necesitará soporte adicional con lo cual el conjunto de reglas de filtrado pueden empezar a complicar y alargar el sistema haciendo más difícil su administración y comprensión.

Estas reglas serán menos fáciles de verificar para las correcciones de las reglas de filtrado después de ser configuradas en el *Router* de selección. Muy probablemente se puede dejar una localidad abierta sin probar su vulnerabilidad.

El filtrado de paquetes *IP* no puede ser capaz de proveer el suficiente control sobre el tráfico, es por eso que la implementación de un *servidor Proxy* que controla las capas más altas, es muy eficiente para cubrir ésta limitante.

2.4.9. Definición de Listas de Acceso.

Los *Routers* de selección definen listas de acceso como una colección secuencial de condiciones de permiso y negación que se aplica a direcciones de Internet. Estas condiciones de lista de acceso se utilizan para implementar las reglas de filtración de paquetes.

Cuando un *Router* de selección se programa con lista de acceso, prueba los paquetes contra cada una de las condiciones de la lista de acceso. La primera coincidencia determina si el *Router* acepta o

rechaza el paquete. Como el *Router* de selección se detiene a probar las condiciones en la lista de acceso después de la primera coincidencia, la orden de las condiciones es crucial. Si no hay condiciones que coincidan, el paquete es rechazado.

Los *Routers* de selección tienen dos tipos de listas de acceso:

- Listas de Acceso Estándar.
- Listas de Acceso Extendidas.

Las listas de acceso estándar tienen una sola dirección para las operaciones coincidentes, y las listas de acceso extendidas tienen dos direcciones con información opcional del tipo de protocolo para operaciones coincidentes. Para muchas operaciones prácticas de filtración, se necesitan tanto las listas de acceso estándar como las extendidas.

Es por eso que dentro de la Seguridad Informática hay dos convenciones indispensables a considerar:

1. No todo lo específicamente permitido esta prohibido.
2. Ni todo lo específicamente prohibido esta permitido.

2.4.10. Uso de las Listas de acceso Estándar.

La sintaxis para la lista de acceso estándar es como se muestra a continuación:

acces-list número de la lista {permit|deny}dirección máscara-comodín
no acces-list lista

acces-list 1 permit 172.16.0.0 0.0.255.255 (Permite el acceso de *Host* a la red de clase B 172.16.0.0).

El número de la lista es un entero que va del 1 al 99 y que se utiliza para identificar una o más condiciones. Es posible asignar a cada regla su propia lista de acceso, pero es altamente ineficiente y propenso a errores. Cada lista de acceso está asociada con una interfaz en el *Router*, tal como una interfaz de red o de la consola. La Lista de Acceso 0 está predefinida; es la predeterminada para toda la interface y las únicas restricciones colocadas en la interfaz son las que soportará el sistema operativo del *Router*.

El uso de las palabras clave permit|deny corresponde a las palabras permitir y bloquear en las reglas de filtración de paquetes analizadas anteriormente. La dirección *Ip* de origen en el paquete se compara con el valor de dirección especificado en el comando acces-list. Si se utiliza la palabra clave permit, una coincidencia hace que el paquete sea aceptado. Si se utiliza la palabra clave deny, una coincidencia hace que el paquete sea rechazado.

La dirección y la máscara-comodín son valores de 32 bits y están escritos utilizando la notación de punto decimal. Máscara-comodín no debe confundirse con las máscaras de subred que se utilizan para dividir una asignación de número de red de *IP*. En la comparación, se ignoran los bits de dirección que corresponden a un 1 en máscara-comodín. Los bits de dirección correspondientes a 0 en máscara-comodín son utilizados en la comparación.

Es posible utilizar el comando no acces-list número de lista para eliminar toda la lista de acceso, pero debe utilizarse con precaución. Las listas de acceso tienen efecto inmediato. Si no se tiene la debida

precaución, puede bloquearse a sí mismo fuera del *Router*, con lo que hace imposible la configuración y la operación.

2.4.11. Uso de las Listas de Acceso Extendidas.

Las listas de acceso extendidas permiten filtrar el tráfico de interfaz con base en las direcciones *IP* de origen y destino y la información del protocolo.

La sintaxis para la lista de acceso extendida es como sigue:

access-list número de lista {permit|deny} protocolo origen máscara-origen destino máscara-destino [operador operando]

El número de lista que va de 100 a 199 y que se utiliza para identificar una o más condiciones permit|deny extendidas. Los números 100 a 199 están reservados para listas de acceso extendidas y se encuentran fuera del rango de los números 1 a 99 utilizados para las listas de acceso estándar.

Si se utiliza la palabra clave permit, una coincidencia con la condición hace que el paquete se acepte. Es la equivalente de la regla permitir utilizada en las reglas de diseño de filtración de paquetes. Si se utiliza la palabra clave deny, una coincidencia hace que se rechace el paquete. Es la equivalente de la regla bloquear utilizada en las reglas del diseño de filtración de paquetes. El resto de la lista extendida no se procesa después de que ocurre una coincidencia.

El protocolo puede representar cualquiera de los siguientes valores correspondientes a los protocolos *IP*, *TCP*, *UDP* e *ICMP*. Cabe recordar

que como *IP* encapsula paquetes *TCP*, *UDP* e *ICMP* puede utilizarse para cumplir cualquiera de esos protocolos.

Origen y Máscara-origen, son valores de 32 bits y están escritos utilizando la notación de punto decimal. Estos valores se utilizan para identificar la dirección *IP* origen. No debe confundirse máscara-origen con la máscara de subred que se utiliza para subdividir una asignación del número de red *IP*. Los bits de dirección correspondientes a 1 en la máscara-origen son ignorados en la comparación. Los bits de dirección correspondientes a 0 en la máscara-origen son utilizados en la comparación.

Destino-máscara-destino, son utilizados para hacer coincidir la dirección de *IP* del destino. También se escribe utilizando la notación de punto decimal, y máscara-destino se utiliza de la misma manera que la máscara-origen para las direcciones de origen.

El operador y operando se utilizan para comparar números de puerto, puntos de acceso a servicio o nombres de contacto. Estos valores son significativos para los protocolos *TCP* y *UDP*. Para los valores clave de protocolo *TCP* y *UDP*, el operador puede ser cualesquiera de los siguientes valores:

- lt (menor que).
- Eq (igual a).
- Gt (mayor que).
- Neq (no igual a).

El operando es una palabra clave o el valor decimal del puerto de destino para el protocolo especificado. También incluye un rango de

valores, permitiendo que la regla de lista de acceso tenga efecto sobre un rango de puertos.

A continuación se muestra un ejemplo de una línea de comandos para una regla que rechaza un paquete *TCP* que proviene del *Host* 192.168.1.1 a la red 172.16.0.0 con un puerto destino de 25 (*SMTP*).

access-list 101 deny *tcp* 192.168.1.1 0.0.0.0 172.16.0.0 0.0.255.255 eq 25

CÁPITULO III

3. TIPOS Y FUNCIONAMIENTO DE UN *PROXY*.

Como una alternativa válida y funcional a los sistemas tradicionales de *Firewalls*, de filtrado de paquetes tenemos a los denominados *Proxys*, ya que tienen una función si no del todo diferente a la de un *Firewall*, si son dentro de las empresas contemporáneas una opción más que se agrega a la larga fila de productos de seguridad informática. El motivo de la inclusión de este capítulo es por la necesidad de los administradores de redes el de tener un mayor control sobre el tráfico que circula en sus redes. Además no hay que olvidar que puede existir solamente un *Proxy* sin necesidad de tener un *Firewall* como tal, y esto a su vez si se encuentra administrado correctamente puede considerarse una magnifica política de calidad para los Sistemas de Información, así que no hay que perderlos de vista. Dentro del capítulo si bien no se profundiza porque no es la columna principal de éste trabajo, si se abordará lo que a mi juicio es lo más relevante que es básicamente el funcionamiento de los *Proxys*, así como también los tipos de conexiones existentes, así estamos en posibilidad una vez terminado este tema, el ir acrecentando nuestro panorama para poder crear nuestro sistema *Firewall* y convertirlo realmente en una política de calidad para nuestros Sistemas de Información.

3.1. Definición de un *Servidor Proxy*.

Los *servidores Proxys* son Hosts que ejecutan una aplicación para un protocolo o conjunto de protocolos en particular y que controlan el tráfico entre una red privada y otra red ya sea privada y/o publica como puede ser Internet. Además aquí no debemos olvidar que un *servidor Proxy* no requiere de un hardware especial , aunque sí de un software especial para la mayoría de los servicios.

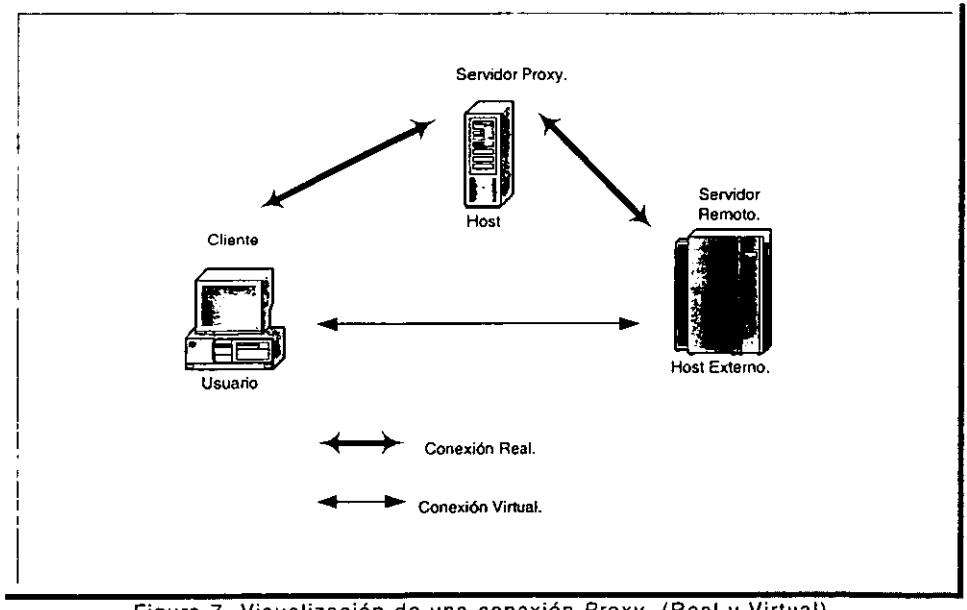

Figura 7. Visualización de una conexión *Proxy*. (Real y Virtual).

3.2. ¿ Porque utilizar un *Servidor Proxy* ?

Bueno básicamente por las siguientes cuatro razones:

- Ocultamiento de la Información: Los nombres de los sistemas internos no necesariamente necesitan ser conocidos por medio de

los *DNS* para los sistemas externos, desde un *Proxy* puede ser el *Host* la única cuyo nombre debe hacerse conocido afuera de los Sistemas.

- Robusta Autentificación y Registro: Los *Proxys* pueden autentificar el tráfico de la aplicación antes de que llegue a los diversos Hosts, por lo cual puede entonces ser registrado el tráfico más efectivamente que con el registro estándar del *Host*.
- Costo: Dentro de la Seguridad Informática este tipo de aplicaciones son en realidad muchísimo más accesibles de adquirir por su costo que cualquier otro tipo de dispositivo de seguridad.
- Menos complejos que las reglas de filtrado: Las reglas en el *Router* de filtrado de paquetes serán menos complejas que aquellas que serían si el *Router* necesitara el filtrar el tráfico de la aplicación y dirigir este a un número de sistemas específicos. El *Router* en este caso sólo necesita permitir el tráfico destinado para la aplicación *Proxy* y rechazar el resto.

3.3. Funcionamiento de un *Proxy*.

Ahora bien el funcionamiento de un *Servidor Proxy* se efectúa de la siguiente manera, el programa cliente del usuario interactúa con el *Servidor Proxy* en lugar de hacerlo directamente con el *servidor* remoto que este en otra red o bien en Internet. El *Servidor Proxy* evalúa el requerimiento del cliente y decide cuál pasar y cuál descartar. Si el requerimiento es aprobado, el *Servidor Proxy* habla con el *Servidor* Remoto en representación del cliente y procede efectuándose dichos requerimientos y devolviéndole la respuesta al cliente.

En lo que al usuario se refiere, interactuar con el *Servidor Proxy* es lo mismo que hacerlo directamente con el *servidor* remoto y en lo que a éste último se refiere, es interactuar con un usuario del *Host* donde se

ejecuta el *Servidor Proxy* el cual no sabe que el usuario está en realidad en un lugar distinto. De esta manera, el usuario cree estar tratando directamente con el *servidor* remoto en otra red al cual quiere realmente acceder.

Es además es estos Hosts donde generalmente se encuentran implementados los mecanismos a través de los cuales se accede al mundo exterior (la nube de Internet) y por ende también los *Firewalls* a nivel de aplicación. Dada la funcionalidad y servicios ofrecidos por los *Proxys*, son varios los temas concernientes a seguridad que pueden verse implementados directamente por ellos. En general son los mismos *Proxys* los utilizados para prevenir que el tráfico pase directamente entre las distintas redes, en lugar de los controladores de tráfico basados en *Routers* o switches; muchos *Proxys* realizan registros adicionales de información o soportan autentificaciones de usuarios y ya que los *Proxys* deben entender el protocolo que esté siendo utilizado, pueden a su vez implementar un mecanismo específico de seguridad para dicho protocolo.

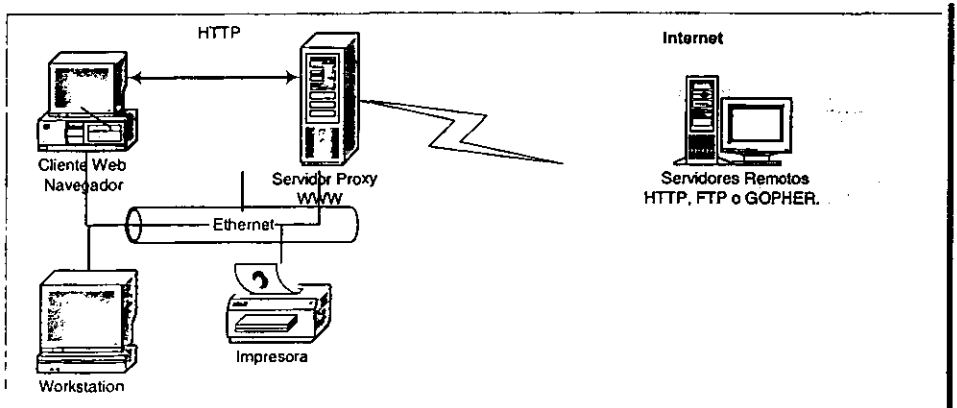

Figura 8. Visualización de un acceso seguro a otras redes o Internet a través de un *Servidor Proxy*.

3.3.1. Conexión Directa.

El primer método utilizado en un sistema de *Servidor Proxy* consiste en que el usuario debe conectarse al *Servidor Proxy* del *Firewall* en la mayoría de los casos utilizando la dirección de éste y el número de puerto de aquel. El *Servidor Proxy* solicita al usuario de la dirección de *Host* a la que desea conectarse.

Esta técnica obliga al usuario a conocer la dirección de su *Firewall*. Asimismo, precisa que el usuario introduzca dos direcciones para cada conexión: la del *Firewall* y la del destino deseado.

Por último, impide a las aplicaciones o programas de nuestros Sistemas de Información o de nuestra red, realizar una conexión por el usuario, dado que desconocen la forma especial con que deben comunicarse con el *Servidor Proxy*.

3.3.2. Cliente Modificado.

El siguiente método utilizado en una configuración de aplicaciones *Proxy* consiste en que la computadora del usuario disponga de aplicaciones cliente modificadas. El usuario ejecuta estas aplicaciones especiales para establecer una conexión a través del *Servidor* Poxy, y posteriormente salir. A los ojos del usuario, estas aplicaciones actúan de forma idéntica que las aplicaciones originales. El usuario proporciona la dirección o nombre del *Host* con el que desea realizar una conexión. La aplicación modificada obtiene la dirección del *Firewall* de un archivo de configuración local, establece la conexión con la aplicación *Proxy* que se encuentra instalada en el *Firewall* y le comunica la dirección proporcionada por el usuario.

Este método es eficaz e invisible para los usuarios. Sin embargo, la necesidad de disponer de una aplicación cliente modificada para cada servicio de red supone un inconveniente considerable. Estas aplicaciones clientes modificadas deben portarse y mantenerse localmente en el gran número de tipos de computadoras que se utilizan en ese segmento de red.

3.3.3. *Proxy* Invisible.

Recientemente se ha desarrollado un método de acceso a los *Proxys* conocido como de *Proxys* Invisibles. Con este sistema no es necesario modificar las aplicaciones cliente, y los usuarios no tiene que dirigir sus comunicaciones hacia un *Firewall* o incluso saber que existe alguno.

Mediante controles de direccionamientos básicos, todas las conexiones que se realizan con la red externa se hacen a través del *Firewall* son dirigidos automáticamente hacia una aplicación *Proxy* que esta a la espera de los mismos. Esta obtiene el destino verdadero examinando la dirección de destino correspondiente a la sesión. De este modo, el *Firewall* se disfraza eficazmente como el *Host* destino y las aplicaciones interceptoras. Una vez efectuada la conexión con el *Proxy* del *Firewall*, la aplicación piensa que esta conectada con el destino verdadero. Si se recibe autorización para ello, la aplicación *Proxy* realiza entonces las funciones estándar que son las propias, consistentes en establecer la segunda conexión con el destino real.

3.4. Puntos a considerar.

Aunque en general el software correspondiente apara la implementación de *Servidores Proxys* se encuentra ampliamente

disponible para los servicios más antiguos y simples como lo pueden ser el caso del *FTP* o del *Telnet*, no ocurre así para los servicios más nuevos o menos usados. Por este motivo, un sistema que necesite nuevos servicios, puede tener que ser ubicado fuera de la zona desmilitarizada abriendo una gran brecha dentro de la seguridad de nuestro Sistema.

Debido a que un *Servidor Proxy* tiene que entender un protocolo para determinar cómo trabajar, pueden ser necesarios diferentes *Proxys* por cada uno de estos protocolos. Juntar, instalar y configurar cada uno de dichos *servidores* puede ser una tarea bastante costosa, tanto en tiempo como en dinero.

En cuanto a soluciones de seguridad, la habilidad de un *Proxy* se basa prácticamente en determinar cuáles operaciones son seguras para un protocolo determinado. Lamentablemente, no todos los protocolos proveen una manera sencilla de hacer esto.

Con respecto a los clientes y procedimientos existentes, excepto por unos pocos servicios diseñados específicamente para *Proxys*, requieren que se les hagan modificaciones para poder utilizarlos. Si bien para algunos servicios basta con realizar cambios en la configuración de los *Servidores*, para la mayoría de ellos no es así y se requiere, además del software adecuado del lado del *Servidor*, la adopción, del lado del cliente, de uno de los siguientes puntos:

- Software para cliente específico: Cuando un usuario hace un requerimiento, el programa debe conocer cómo contactarse con el *Servidor Proxy* en lugar del *Servidor* Remoto y cómo indicarle de qué *Servidor* remoto se trata.

- Procedimientos de Usuario: El usuario usa programas estándar para comunicarse con el *Servidor Proxy* y le dice como conectarse con el *Servidor* Remoto, en lugar de conectarse directamente.

En este segundo punto, si bien no introduce demasiadas complicaciones en su implementación, brinda una solución muy poco transparente para el usuario. Por el contrario, el primer punto es muchísimo más simple pero trae entre otras cosas algunos problemas que requieren de un apartado el cuál no se discutirá en este trabajo. Entre otras cosas y sólo por mencionar algunos puntos el software del cliente se encuentra disponible sólo para plataformas específicas y si el usuario no dispusiese de dicha plataforma simplemente no podría utilizarlo. Más aún, el usuario podría contar con la plataforma pero no estar conforme con el desarrollo hecho para ella o bien querer utilizar alguna otra que no estuviese preparada para ser utilizada en conjunto con un *servidor Proxy*. En general, son muy pocos los programas clientes que viene con soporte para *Proxys* y si bien uno podría pensar en realizar las modificaciones necesarias sobre los mismos, este proceso requeriría disponer de los códigos fuente y de las herramientas adecuadas para compilarlos.

Existen ya en la actualidad clientes específicos para trabajar sobre el *WWW*, más conocidos como navegadores o exploradores (Netscape Navigator y/o Internet Explorer). La mayoría si no es que su totalidad de estos programas ya permiten la utilización de *Servidores Proxy*.

En lo que a estos navegadores o exploradores se refiere, las modificaciones necesarias para que trabajen con *Proxys* son en realidad mínimas y no hay necesidad de compilar versiones especiales con bibliotecas de *Firewalls*. El cliente común y corriente puede estar configurado para ser un cliente *Proxy*. De ésta forma, incorporando

seguridad dentro de los *Proxys*, éstos se convertirán a su vez en una especie de *Firewall*, evitando que tener que adaptar cada cliente existente para soporte de un producto en específico ó método de *Firewall* especial. Esto es de verdadera relevancia ya que en el caso de los navegadores o exploradores comerciales, ya que no se dispone de su código fuente para poder modificarlos.

Más aún, los usuarios no necesitan tener clientes *FTP*, Gopher, etc., separados y modificados especialmente para poder utilizar un *Servidor Proxy*, ya que éste último puede, en conjunto con un solo navegador o explorador, manejar todos éstos casos. El *Proxy* estandariza la apariencia de los listados *FTP* y Gopher entre los clientes, evitando así que cada uno de ellos tenga que efectuar el manejo individual de los mismos.

CAPÍTULO IV

4. POLÍTICAS DE SEGURIDAD.

Dentro de nuestro cuarto y último capítulo de éste trabajo abordaremos un tema trascendental como lo son las políticas de seguridad, para que éstas mismas se conviertan a su vez en políticas de calidad para nuestros *Firewalls*, entendiendo claro ésta a éstos últimos como un Sistema, y no como un simple software o hardware aislados. Tenemos primero que nada explicar detalladamente lo que es una política de seguridad y a su vez los elementos intrínsecos que la componen, además de que debemos empezar por hacer el planteamiento de que es lo que tenemos que salvaguardar, y también se detallará en un estudio breve pero conciso el análisis de riesgo de nuestro sistema, todo este estudio se trasladará a algunas tablas para que la organización se una parte intrínseca de política de calidad que deseamos. No pasemos por alto que la seguridad empieza en casa, así que tenemos que tomar muy en cuenta quienes tendrán acceso, a que tipo de recursos, así que nos daremos a la tarea de traspasarlo de igual manera a una tabla, donde se visualiza de mejor manera para el análisis y su estudio. Para terminar tocaremos brevemente que es lo que se tiene que hacer en dado caso de un ataque, y como punto final se hablará de la certificación más difundida dentro de la industria que es la de *ICSA*. Bajo este tenor, y teniendo ahora si las bases suficientes, el lector de éste trabajo esta en posibilidades de poder establecer a los *Firewalls* en su caso como una política de calidad para los Sistemas de Información.

4.1. Contenido de una Política de Seguridad.

En la sociedad de hoy en día, al acceso a cualquier clase de información se ve facilitado por diferentes tecnologías de conexión, con costos muy accesibles, que permiten interactuar local o remotamente con diferentes centros de cómputo. Sin embargo, este acceso instantáneo y gratuito a la información trae consigo muchas dificultades en cuanto a la seguridad y privacidad de la información de una organización. Es por eso que dentro de la misma organización se deben desarrollar políticas de seguridad para prevenir cualquier acceso no autorizado a nuestros Sistemas de Información.

Deberíamos empezar definiendo lo que significa una política de seguridad, que es elaborar procedimientos y planes que salvaguarden los recursos de la red contra ataques, pérdidas y daños a nuestros Sistemas de Información.

4.1.1. Explicativa y Comprensiva.

Es de vital importancia que una política de seguridad, para los Sistemas de información sea verdaderamente explicativa y compresible para todos y cada uno de los integrantes de la organización que utilicen los recursos de la red, así como de los Sistemas de Información. No hay que olvidar que una política debe especificar lo que debe hacerse, pero también lo que no, porque de lo contrario puede llegar a fracasar.

4.1.2. Responsabilidad.

Una política de seguridad estable por sí misma una expectativa y responsabilidad entre los autores y los usuarios de la misma; esto es básicamente para permitir cual es el papel de cada uno de los

integrantes del círculo de la política. De cualquier modo hay que tener ética profesional, para poder publicar una política de seguridad.

4.1.3. Lenguaje Común.

Dentro de éste punto es muy importante saber que no estamos inmersos dentro del ámbito jurídico, y es por eso que la mayoría de los usuarios de los Sistemas de Información de los recursos de red, se sienten muchísimo más libres cuando se ocupa un lenguaje común y entendible. Siempre debe escribirse de una manera que no sea muy técnica, porque de ser así se pierde el fondo de la misma, y los usuarios siempre estarán predispuestos a rechazarla. Pero tampoco hay que ser tan informal y parecer condescendiente o desordenado, hay que buscar el equilibrio.

4.1.4. Autoridad.

Escribir y Publicar la política de seguridad no es lo más importante sino llevarla a cabo y cumplirla, es por eso que cuando no se esta cumpliendo a cabalidad hay que poner en marcha un plan para solucionar esta situación, por tal situación debe haber una persona que sea capaz de hacerse responsable y con tal autoridad para que todas y cada una de las políticas de seguridad expuestas sean cumplidas. No debe existir por ningún motivo fugaz de autoridad, porque es muy riesgoso este tipo de incumplimientos. Además también debe existir quien será la persona que una vez que se infrinja determinada política de seguridad ejecutara la sanción que deberá imponerse al infractor.

4.1.5. Revisiones.

Es muy común dentro de todos los ámbitos de las organizaciones de nuestro país, se tenga muy poca memoria, y bueno el área informática no ésta exenta de esa pérdida de memoria, es por ello que las políticas de seguridad deben estar siempre vigentes, porque hoy en día la única constante es el mismo cambio, no debemos olvidar que también las políticas de seguridad pueden llegar a convertirse en obsoletas, así que las revisiones y el monitoreo de las mismas debe ser constante.

4.2. Planteamiento de una Política de Seguridad.

Es de vital importancia que el diseño de una política de seguridad este involucrada la gente más capacitada, no necesariamente la gente que administra la red, o que maneja el centro de cómputo. Tiene que involucrar a la mayoría de los sectores de la organización para que ésta no encuentre impedimentos ni obstáculos, como es una costumbre dentro de la cultura mexicana. Nunca una política de seguridad va a ser concebida en perjuicio de la organización, sino más bien todo lo contrario, pero para esto se debe desarrollar un planteamiento como puede ser una hoja de trabajo, como se muestra a continuación:

Recursos de la fuente.			Tipo de Usuario del que hay que proteger al recurso	Posibilidad de amenaza	Medidas que se deben implementar para proteger al recurso de red.
Número	Nombre	Importancia del Recurso.			

Tabla 10. Hoja de trabajo para el planteamiento de una política de seguridad.

- Número: Es el número de recurso de red para identificar al mismo.
- Nombre: Es la descripción en el lenguaje común de los recursos.

- Importancia: Esta se establece de acuerdo en escalas, que pueden ser numéricas o en expresiones.
- Tipo de Usuario: Los más comunes pueden ser los internos externo, pero hoy en día se definen como grupos de trabajo.
- Posibilidad de amenaza: Esta se establece de acuerdo en escalas, que pueden ser numéricas o en expresiones.
- Medidas a implementar: En software podemos mencionar los permisos que pueda llegar a tener nuestro sistema operativo y en cuestión hardware, se establecer el bloque o no del dispositivo solicitado.

4.3. Análisis de Riesgo.

Cuando se crea una política de seguridad, es importante comprender que la razón para crear una política es, en primer lugar, asegurar que los esfuerzos dedicados a la seguridad sean financiables. Esto en pocas palabras significa que hay que saber detectar cuáles son los recursos que valen la pena proteger, y para eso se hizo la hoja de trabajo de planteamiento de la seguridad. No hay que olvidar que el 71% de los ataques que sufrieron las organizaciones en el año 1991 hace ya una década provinieron de usuarios internos, esto nos lleva a preguntarnos básicamente que el análisis de riesgo implica determinar tres cuestiones:

- ¿ Qué se necesita proteger ?
- ¿ De qué se necesita protege ?
- ¿ Cómo debo protegerlo ?

Los riesgos deben clasificarse por nivel de importancia y gravedad de la pérdida, es por debe que en el análisis de riesgo hay que determinar los siguientes dos factores:

- Estimación del riesgo de perder el recurso (RR).
- Estimación de la importancia del recurso (IR).

Como ya se explico en la hoja de trabajo de planteamiento de seguridad se tomaran valores numéricos para cuantificar el riesgo del recurso (RR), y de igual modo a la importancia del recurso (IR). Con estos datos el riesgo evaluado del recurso(RER) será igual a la multiplicación de (RR) por (IR), como se muestra a continuación:

$$RER = RR * IR$$

Además de que también podemos ir registrando este tipo de cálculos en una hoja de trabajo denominada Análisis de riesgo, la cual sería de la siguiente manera:

Recursos de la Red		Riesgo de los Recursos de la Red (RR).	Peso (Importancia) del Recurso (IR)	Riesgo Evaluado (RER).
Número	Nombre			

Tabla 11. Hoja de trabajo denominada análisis de riesgo.

- Número: Es el número de recurso de red para identificar al mismo.
- Nombre: Es la descripción en el lenguaje común de los recursos.
- RR: Riesgo del recurso que se establece de acuerdo en escalas, que pueden ser numéricas o en expresiones.
- IR: Importancia del recurso se establece de acuerdo en escalas, que pueden ser numéricas o en expresiones.
- RER: Riesgo evaluado, como el producto de los valores de riesgo e importancia.

Además podríamos sacar cuál es el riesgo general de los recursos de nuestro *Backbone* y podríamos sacarlo de la siguiente manera:

$$RG = (RR1 * IR1 + RR2 * IR2 + ... + RRn * IRn) / (IR1 + IR\ 2 + ... + IRn)$$

4.4. Identificación de Recursos.

Al realizar el análisis de riesgo, se deben identificar todos y cada uno de los recursos que están expuestos al riesgo de sufrir alguna violación de seguridad informática. Dentro de la *RFC* 1244 se enlistan los siguientes recursos de red que se deben considerar al calcular las amenazas a la seguridad general.

- Hardware: Todos y cada uno de los dispositivos físicos colocados dentro de nuestro *backbone*, tales como procesadores, tarjetas, teclados, terminales, estaciones de trabajo, computadoras personales, impresoras, unidades de disco, *routers*, *firewalls*, *servidores*, etc.
- Software: Programas fuente, utilerías, sistemas operativos, programas de diagnóstico, de comunicaciones etc.
- Datos: Durante la ejecución de algún proceso, almacenamiento en línea, archivos fuera de línea, correos almacenados, respaldos, bases de datos, registros, logs, etc.
- Personas: Usuarios tanto internos como externos, y toda aquella persona involucrada o no para operar los diferentes sistemas de información de nuestra organización.
- Documentación: Sobre programas, hardware, sistemas, procedimientos administrativos, procedimientos técnicos, etc.
- Suministros: Papel, formularios, discos, cintas, medios magnéticos, etc.

4.5. Identificación de las Amenazas.

Una vez que se han identificado los recursos que requieren protección, se debe proceder ahora a identificar las amenazas a las que se está expuesto. Dentro de éste ámbito tenemos la posibilidad de determinar las pérdidas que puedan existir. Y bueno dentro de este trabajo sólo se examinan tres de ellas que son:

- Definición del Acceso no Autorizado.
- Riesgo de Revelación de Información.
- Negación del Servicio.

4.5.1. Definición del Acceso no Autorizado.

El acceso a los recursos de nuestros Sistemas de Información y de red, sólo deben estar permitidos a los usuarios autorizados, así que estamos en posibilidad de considerar que el uso de cualquier otro recurso dentro de los Sistemas de Información y de red sin previa autorización o permiso es un acceso no autorizado. Durante los últimos años la gravedad de ésta amenaza ha ido incrementándose a tal punto que en el año de 1991 el ataque a los Sistemas de Información bajo esta definición fue del 26%, que es el segundo más alto del total de los ataques. Existen varias formas de acceso, entre la más común es el uso de la cuenta de otro usuario para tener acceso a los recursos.

4.5.2. Riesgo de Revelación de Información.

La revelación de información, ya sea voluntaria, consciente o involuntaria e inconsciente, es una de las mayores amenazas que existen para los Sistemas de Información, es por eso que hoy en día en la mayoría de las organizaciones en México y en el mundo dentro de los

contratos de trabajo, una de las cláusulas de mayor peso, es aquella en donde la revelación de información es penada por las normas de la empresa así como por las leyes vigentes del país. Como por ejemplo podemos mencionar un proyecto de investigación que represente varios meses de trabajo puede darle a su competidor una ventaja injusta.

4.5.3. Negación del Servicio.

Las redes hoy en día vinculan servicios y recursos valioso, como Bases de Datos, dispositivos tales como unidades, cintas magnéticas, etc que son imprescindibles para las organizaciones. Hoy en día todos los usuarios de los Sistemas de Información dependen de los servicios que una red proporciona, y es indispensable para que los mismos puedan realizar su trabajo eficaz y eficientemente. Si no están disponibles estos servicios, hay una pérdida obvia de productividad. No hay que olvidar también que una mala planeación en la negación de servicios puede traer consigo más problemas de lo que debíamos evitar.

4.6. Autorizaciones de Acceso.

Dentro de las políticas de seguridad también debe estar especificado quién es la persona que está autorizando determinado permiso entre los más comunes tenemos s loa de lectura, escritura y ejecución, además que dichas personas por lo regular el administrador de usuarios deben saber que tipo de permiso debe contener cada uno de los usuarios dentro de su grupo de trabajo u organización. Muy a menudo si no se puede llegar a controlar adecuadamente a cada uno de los usuarios que tienen acceso a los Sistemas de Información o a los servicios de red, se estará aún más vulnerable de las amenazas tanto internas como externas, esto no quiere decir que entre más usuarios mayor amenaza hay, sino más bien se puede resumir en que si no hay

un estricto control de acceso, existirá una mayor amenaza y así contrariamente.

Es por eso que el gran reto al que se enfrenta cualquier administrador de accesos, es llegar a tener un equilibrio y conceder sólo los privilegios suficientes e indispensables para poder llegar a cumplir con las tareas asignadas eficaz y eficientemente.

Esto trae consigo un problema que es el llevar el control de cada uno de los usuarios de los recursos tanto de nuestros Sistemas de Información como de red, y es por ello que puede seguirse una manera formal de solicitudes, a ésta debe darle un seguimiento en su cadena de mando dentro de cada organización y así llegar hasta el administrador de cuentas, el cual deberá de documentar las restricciones de seguridad o de acceso a las que estará sujeto el usuario que hizo la petición. Una forma de hacer sencillo este proceso es mediante la siguiente hoja de trabajo, la cual nos permite tener una descripción clara y concisa de los permisos que se le han otorgado a un determinado usuario.

Recurso del Sistema o de Red.		Tipo de Acceso.	Permiso del Sistema Operativo o del Sistema de Información.
Número	Nombre		

Tabla 12. Hoja de trabajo, donde podemos obtener de manera rápida el tipo de acceso y el permiso que tiene cada uno de los usuarios.

- Numero: Es el número de red de identificación interna de cada recurso.
- Nombre: Es la descripción en un lenguaje común de los recursos.
- Tipo de Acceso: Puede usurase para describir como puede llegar a ser el acceso, si es local y / o remotamente,

- La guía de pruebas debe estar suministrada por el vendedor y será revisada por el equipo del laboratorio *ICSA* para su total exactitud.
- El sistema operativo será instalado en una máquina limpia y almacenado de acuerdo a las instrucciones del vendedor.
- El *Firewall* será instalado de diversas maneras (De red, de inicio, de cd, etc).
- El *Firewall* será rastreado, una vez instalado.
- El *Firewall* será configurado por servicios de *ICSA*, y cada uno de estos se revisará que este respondiendo adecuadamente.
- El *Firewall* será rastreado una vez configurado por los servicios de *ICSA*.
- Todas y cada una de las funciones ofrecidas por el *Firewall* serán valoradas y revisadas, en cumplimiento, y así poderlo certificar.

CONCLUSIONES

Una vez que hemos concluido éste trabajo, estamos en la posibilidad de sentar bases sólidas, para poder llegar a establecer el objetivo principal, tal como se había buscado, el poder entender a los *Firewalls* como una inversión y no como un gasto, como casi siempre ocurre en la mayoría de los casos, como se habrán dado cuenta en ningún momento se menciona un software o hardware supuestamente de *Firewall*, la principal causa es que no tengo ninguna intención discutir cuál tiene mayores ventajas a los demás, o bien contiene algunas desventajas con respecto a productos similares, lo que si busco es implementar que de acuerdo a las necesidades de cada empresa u hogar, se instale no solamente un *Firewall* sino todo un sistema que cubra las necesidades de la misma. Cuando se llegue a entender que cubriendo las necesidades de la empresa o las de un usuario de acceso telefónico, claro está previamente un estudio, como se explicaron en el capítulo IV, llegaremos a que los *Firewalls* sean una verdadera política de calidad, porque si bien debe considerarse los costos de determinado producto, lo más importante es el estudio de necesidades que se debe realizar, y si éste llegue a concebirse correctamente llegará a hacer una inversión y no un gasto. Si bien la columna vertebral la llevan los capítulos II y IV, por contener el peso en cuanto contenido, así que no hay que olvidar los *Proxys* son una herramienta complementaria a los sistemas *Firewalls* como los hemos llamado algunas líneas arriba, debido a que en muchas ocasiones las famosas necesidades de las que ya hemos descrito anteriormente, únicamente nos llevarían a instalar un *Proxy*, y no necesariamente hacer un gasto inútil en alguna otra herramienta innecesaria. Y por otro lado, si alguna persona interesada en éste tema no esta completamente empapado, será de vital importancia e

indispensable revisar el tan discutido capítulo I, para que cualquier otra bibliografía que analice no llegue a ser complicada. Esperando en verdad que sea de alguna utilidad este trabajo y aporte algún punto a su acervo.

GLOSARIO DE TÉRMINOS TÉCNICOS.

ACK: Reconocimiento – Acknowledgment. Notificación enviada de un dispositivo de red a otro, para confirmar que ha ocurrido algún evento.

ASCII: Código Estándar Americano para el Intercambio de Información. Es un código de 8 bits para la representación de caracteres 7 bits más 1 de paridad.

ARPANET: Red de la agencia de Proyectos de Investigación Avanzada. Importante red de conmutación de paquetes establecida en el año de 1969, la red *ARPANET* fue desarrollada en los años 70´s por BBN y financiada por ARPA y después por DARPA. Finalmente se convirtió en Internet . El término *ARPANET* fue càncelado oficialmente en 1990.

ASP: Proveedor de Servicios de Acceso - Access Service Provider.

ATM: Modo de Transferencia Asíncrono – Asyncronice Transfer Mode. Estándar internacional para conmutación de celdas, en el que se transportan varios tipos de servicios por medios de celdas de longitud fija de 53 bytes. Las celdas de longitud fija permiten que el procesamiento de celdas se haga en hardware, reduciéndose así los retardos de transmisión. *ATM* está diseñado para aprovechar al máximo medios de transmisión a alta velocidad como E3, SONET Y T3.

BACKBONE: Columna vertebral, término utilizado para describir la estructura e infraestructura de una red.

DARPA: Agencia de Proyectos de Investigación Avanzada de la Defensa - Defense Advance Research Projects Agency.

DATAGRAMA: Es la agrupación lógica de información que se envía como una unidad de la capa de red por un medio de transmisión, sin el establecimiento previo de un circuito virtual. Los datagramas *IP* son las unidades de información de mayor importancia en la Internet. Los términos trama, mensaje, paquete y segmento también se utilizan para describir agrupaciones lógicas de información en distintas capas del modelo de referencia *OSI* y en varios círculos de tecnología.

DNS: Sistema de Nombres de Dominio – Domain Name System. Sistema utilizado en Internet para traducir los nombres de los nodos de red en direcciones.

ETHERNET: Especificación de *LAN* de banda base, inventada por la Corporación Xerox desarrollada en conjunto por Xerox, Intel y Digital Equipment Corporation. Las redes *Ethernet* utilizan métodos de acceso CSMA/CD y corren sobre una gran variedad de tipos de cables a 10 Mbps. La red *Ethernet* es similar a los estándares de la serie *IEEE 802.3*.

E-MAIL: Correo electrónico. Aplicación de red muy popular, en la que se transmiten electrónicamente mensajes de correo entre usuarios terminales a través de varios tipos de reds, mediante varios protocolos de red.

FDDI: Interfase de Datos Distribuidas por Fibra. Estándar *LAN* definido por la ANSI X3T9.5 que especifica una red de *Token ring* a 100 Mbps que utiliza cable de fibra óptica, con distancias de

transmisión de hasta 2 Kms. El estandar *FDDI* utiliza una arquitectura de anillo doble para proporcionar redundancia.

FIREWALL: Elementos basados en Hardware, Software o en una combinación de ambos, que controla el flujo de datos que entra y sale de una red a otra.

FTP: Protocolo de Transferencia de Archivos – File Transfer Protocol. Protocolo de aplicación, parte de la pila de protocolos *TCP/IP*, que se utilizan para la transferencia de archivos entre nodos de la red. Esta definido en el *RFC* 959.

FULL-DUPLEX: Característica que permite transmitir datos de manera simultánea entre una estación emisora y una estación receptora.

HOST: Sistema de computación en una red. Es similar al término nodo excepto que el *host* por lo común implica un sistema de computadoras, en tanto que un nodo en general se aplica a cualquier sistema de red, incluyendo a los *servidores* de acceso y *ruteadores*.

HTTP: Protocolo de Transferencia de Hiper Texto – Hyper Text Transfer Protocol.

ICMP: Protocolo de Control de Mensajes de Internet – Internet Control Message Protocol. Protocolo de Internet de la capa de red que reporta errores y proporciona información relevante al procesamiento de paquetes *IP*. Esta docuemtado en el *RFC* 792.

ICSA: Asociación Internacional de Seguridad en Computadoras – International Computer Security Association. Localizada en Reston,

Va., es una organización reconocida mundialmente como la autoridad en servicios de convicción de seguridad fundada en el año de 1989.

IEEE 802.3: Protocolo IEEE *LAN* que especifica una aplicación de la capa física y de la subcapa MAC de la capa de enlace de datos. El *IEEE 802.3* utiliza el método de acceso CSMA/CD a una gran variedad de velocidades sobre diferentes medios medios de transmisión. Las extensiones del estándar *IEEE 802.3* especifican aplicaciones para Fast *Ethernet*. Las variaciones físicas de la especificación original *IEEE 802.3* incluyen 10B2, 10B5, 10BF, 10BT y 10B36. Las variaciones físicas de Fast *Ethernet* incluyen 100BT, 100BT4 y 100BX.

IP: Protocolo de Internet – Internet Protocol. Protocolo de la capa de red en la pila *TCP/IP* que ofrece un servicio de red sin conexión. El protocolo *IP* proporciona características de direccionamiento, especificación del tipo de servicio, fragmentación y reensamblado y seguridad. Esta documentado en el *RFC* 791.

IPX: Intercambio de Paquetes de Red. Protocolo de la capa de red – 3 de Netware, que se utiliza para transferir datos de los *servidores* a las estaciones de trabajo. *IPX* es similar a *IP* y a XNS.

IRC: Transmisión de conversación por Internet – Intenet Relay Chat.

ISO: Organización Internacional de Normas – International Standars Organization. Organizacxión internacional responsable de una amplia gama de estándares, incluyendo los pertinentes a las redes. El *ISO* desarrolló el modelo de referencia *OSI*.

ISP: Proveedor de Servicios de Internet – Internet Service Provider.

JPG: Formato de Interncambio de Archivos - File Interchange Format.

LAN: Red de Área Local – Local Area Network. Red de datos de alta velocidad y baja tasa de errores, que cubre un área geográfica relativamente pequeña. Las *LAN*`s conectan estaciones de trabajo, periféricos, terminales y otros dispositivos en un solo edificio u otra área geográfica limitada. Los estándares de las *LAN* especifican el cableado y la señalización en las capas física y de enlace de datos del modelo *OSI*. Las redes *Ethernet*, *FDDI* y *Token ring* son tecnologías *LAN* ampliamente utilizadas.

MTU: Unidad Máxima de Transmisión – Unit Maximum Transmision . Es el tamaño máximo de paquete en bytes que puede manejar una interfase en particular.

NAT: Traductor de Direcciones de Red – Network Address Translation. Proceso de traducción de dirección *IP* no homologadas a direcciones *IP* homologadas, o que son válidas para Internet y viceversa.

NFS: Fundación Nacional para las Ciencias – National Science Foundation.

NIC: Centro de Información de la Red – Network Information Center. Organización que da servicio a la comunidad de Internet mediante asistencia al usuario, documentación, entrenamiento y otros servicios.

OSI: Interconexión de Sistemas Abiertos – Open System Interconnections. Es el programa de estandarización internacional

creado por la *ISO* y la ITU-T para desarrollar estándares para las redes de datos que faciliten la interpolación de equipos fabricados por diferentes proveedores.

PING: Paquete explorador de Internet. Mensaje de eco de *ICMP* y su respuesta. Se suele utilizar para probar el alcance de un dispositivo de red.

PROXY: Agente – Representante. Es un *servidor* que administra el tráfico que se produce entre una red privada y una pública, como puede ser Internet.

RFC: Solicitud de Comentarios – Request for Comments. Es una serie de documentos que se utiliza como la manera principal para comunicar información al respecto a Internet. Algunos *RFC* están designados por el IAB como estándares de Internet. La mayoría de los *RFC* documentan especificaciones de protocolo, como Telnet y *FTP*. Los *RFC* están disponibles en línea en muy variadas fuentes.

RS232C: Interface popular de la capa física. Actualmente se le conoce como EIA/TIA – 232.

RUTEADOR O ROUTER: Dispositivo de la capa de red que utiliza una o más medidas para determinar la trayectoria óptima a lo largo de la cual deba direccionarse el tráfico de la red. Los *ruteadores* direccionan paquetes de una red a otra con base en la información de la capa de red. A veces se les llama compuerta.

SERVIDOR: Server. Es un nodo que provee de diversos servicios a los clientes o usuarios.

UDP: Protocolo de *Datagrama* de Usuario – User Datagram Protocol. Es el protocolo de la capa de transporte no orientado a la conexión, en la pila de protocolos de *TCP/IP*. *UDP* es un protocolo simple que intercambian datagramas sin reconocimientos o entregas garantizadas, y requiere por ello que el procesamiento de errores y la retransmisión sean manejados por otros protocolos. El protocolo *UDP* está definido en el *RFC* 768.

V.35: Estándar de la ITU-T que describe un protocolo síncrono de la capa física que se utiliza para las comunicaciones entre un dispositivo de acceso a la red y una red de paquetes. *V.35* se utiliza más comúnmente en Estados Unidos y en Europa y se recomienda a velocidades de hasta de 48 Kbps.

WWW: World Wide Web – Red Mundial. Es una red de gran tamaño de *servidores* de Internet que proveen hipertexto y otros servicios a terminales que corren aplicaciones de cliente como pueden ser los navegadores.

BIBLIOGRAFÍA

"Seguridad en la información en sistemas de cómputo."

Rodríguez, Luis Ángel
Ediciones Ventura, S.A. de C.V., México, 1995
353 pp.

"Seguridad Informática."
Nombela, Juan José
Editores Thomson, España, 1997
258 pp.

"Seguridad Informática. Técnicas Criptográficas."
Caballero, Gil Pino
Grupo Editorial Alfa Omega, España, 1996
137 pp.

"Seguridad en centros de cómputo, políticas y procedimientos."
Fine, Leonard H.
Editorial Trillas, U.S.A., 1998
130 pp.

"Seguridad en UNIX."

Mediavilla, Manuel
Grupo Editorial Alfa Omega, España, 1997
222 pp.

"Manual de seguridad para PC y redes locales."

Cobb, Stephen
Editorial McGrawHill, España, 1994
627 pp.

"Firewalls y la seguridad en Internet."

Siyan, Karanjit – Hare, Chris
Editorial Prentice Hall, México, 1996
631 pp.

"Firewalls and Internet Security."
Cheswick, William R. – Bellovin, Steven M.
Editorial Addison Wesley, U.S.A., 1994
306 pp.

"Redes Locales y TCP/IP."
Raya, José Luis
Grupo Editorial Alfa Omega, España, 1995
177 pp.

"Redes globales de información con Internet y TCP/IP." 3ra Ed.
Comer, Douglas E.
Editorial Prentice Hall, México, 1996
621 pp.

"TCP/IP Network Administration." 4ta Ed.
Hunt, Craig.
Editorial O´Reilly & Associates, U.S.A., 1994
472 pp.

"Internet Security Professional Reference."
Atkins, Derek. – Buis, Paul.
New Riders Publishing, Indianapolis, IN, 1996
908 pp.

"Instructor Guide Version 1.0."

Cisco System, Inc. USA, 2001
Interconnecting Cisco Network Devices.
852 pp.

REFERENCIA DE PÁGINAS EN LA INTERNET.

"El fenómeno de los Firewalls."
http://it.unex.es/sypi/trabajos/incidentes/jjst/ff1a.htm

"Seguridad en Internet."
http://www.paisvirtual.com/informatica/redes/seguinf/index.htm

"Gestión de Firewalls."
http://www.deepzone.org/editions/others/gestion.htm

"Firewalls."
http://www.geocities.com/siliconvalley/way/4651/seguridad/firewalls/

"Firewalls y Seguridad en Internet."
http://www.aebius.com/b_datos_doc/pages/firewalls1.htm

"Firewalls."
http://www-mat.upc.es/~jforne/firewalls.pdf

"Información general sobre Firewalls."
http://www.cyberangels.org/international/espanol/net-ed/firewalls.html

"Seguridad en Internet."
http://www.inei.gob.pe/cpi/bancopub/libfree/lib620/indice.htm

"ICSA"

http://www.icsa.net

"Firewalls"

http://www.firewall.com