CAPITULO V

CAPITULO V

5. METODOLOGIA PARA LA IMPLEMENTACION DE FIREWALLS DE SOFTWARE

5.1 PROTOCOLOS DE COMUNICACIÓN

Un protocolo de red es una especificación detallada de las "reglas" que deben seguir los diferentes programas que emplea una red de comunicaciones para intercambiar información. Para que un protocolo de red sea útil, su especificación debe ser pública y debe ser aceptada por una parte significativa de la industria, es el caso de TCP/IP, que desde hace más de 20 años es el protocolo de red de mayor uso en el mundo y el "motor" sobre el que está construido Internet.

Los protocolos de red suelen especificarse mediante "capas" superpuestas de funcionalidad, el objetivo de esta segmentación es que sea posible (por razones de cambio tecnológico, por ejemplo), sustituir una capa por otra equivalente, sin necesidad de sustituir la totalidad del hardware y el software que manejan las comunicaciones. Cada una de las capas que define un protocolo tiene que ver con un determinado "nivel" de funcionalidad, y precisamente por ello, se denominan "niveles". Los niveles más bajos tienen que ver con el hardware, los superiores son responsabilidad únicamente de los programas que intercambian información, y los niveles centrales constituyen el "núcleo" del protocolo y están implementados, normalmente en el Sistema Operativo o alguna librería estándar.

5.1.1 El Conjunto de Protocolos TCP/IP

En 1973, la Agencia de Proyectos de Investigación Avanzada para la Defensa (DARPA), de los Estados Unidos, inició un programa para la investigación de tecnologías que permitieran la transmisión de paquetes de información entre redes de diferentes tipos y características. El proyecto tenía por objetivo la interconexión de redes, por lo que se le denominó "Internetting", y a la familia de redes de computadoras que surgió de esta investigación se le denominó "Internet". Los protocolos desarrollados se denominaron el Conjunto de Protocolos TCP/IP, que surgieron de dos conjuntos previamente desarrollados; los Protocolos de Control de Transmisión (Transmition Control Protocol) y el Protocolo Internet (Internet Protocol).

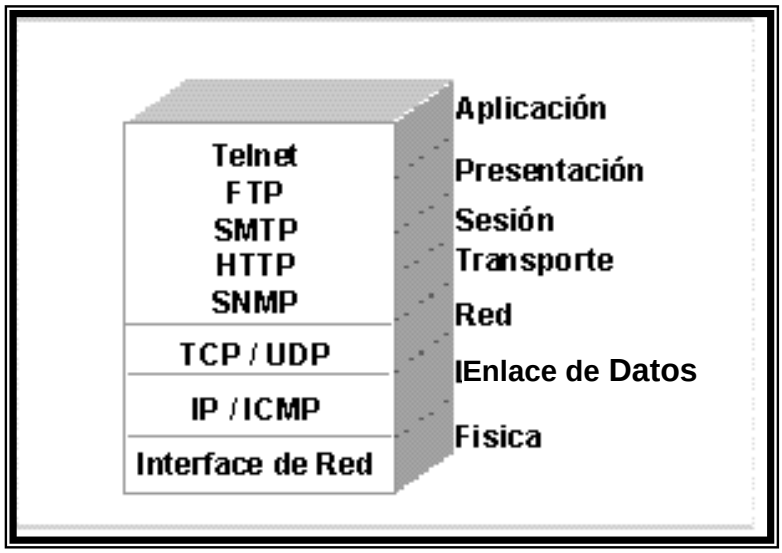


Fig. 5.01 Conjunto de Protocolos TCP/IP

Como muestra la figura 5.01 la comunicación de red se explica como un modelo de capas, donde la comunicación se realiza entre capas adyacentes sobre un equipo individual y entre capas paralelas sobre equipos que se comunican. El programa que se ejecuta (por ejemplo, un navegador Web) se encuentra en la parte superior, en la capa de aplicación, comunicándose con otro programa sobre otro equipo (por ejemplo, un servidor web).

Para que la aplicación cliente del navegador web envíe una petición de una página Web a la aplicación del servidor Web, tiene que usar llamadas de biblioteca y de sistema para obtener la información del navegador web y encapsularla en un mensaje apropiado para poder transportarlo entre dos programas de red. Estos mensajes son segmentos TCP o datagramas UDP de la capa de transporte; para conseguir estos mensajes, la capa de aplicación llama a la capa de transporte para que ofrezca este servicio. Los mensajes de la capa de transporte saben como entregar mensajes entre un programa de un equipo y un programa situado en el otro extremo de la red. Tanto el modelo OSI como el modelo TCP/IP llaman a esta capa la capa de transporte, aunque el modelo OSI divide esta capa en varias capas funcionales.

Para que los mensajes de la capa de transporte se entreguen entre los dos programas, es necesario enviar los mensajes entre los dos equipos, para ello, la capa de transporte hace uso de funciones del sistema operativo que toman el mensaje de transporte TCP o UDP y lo encapsulan en un datagrama de Internet adecuado para enviarlo al otro equipo, estos datagramas son paquetes IP. Los paquetes IP de Internet se envían entre los dos equipos a través de Internet. La capa Internet sabe como comunicarse con el equipo situado en el otro extremo de la red El modelo de referencia TCP/IP llama a esta capa la capa Internet.

Debajo de la capa de red está la capa de subred, de nuevo el paquete se encapsula en un encabezado de Ethernet. Desde el punto de vista de TCP/IP, la capa de subred es un conjunto de todo lo que sucede para conseguir que el paquete se entregue al siguiente equipo. Este conjunto incluye todo el direccionamiento y los detalles de entrega asociados con el enrutamiento de la trama entre los equipos, de un enrutador al siguiente, hasta que, por último se alcanza el equipo destino. Esta capa incluye la

traducción de la trama de red de una clase de red a otra a largo del camino. La mayoría de las redes actuales son redes Ethernet, pero también existen redes ATM, FDDI, Token Ring, etc., es decir, cualquier tecnología de red que se esté usando para transportar tramas entre dos equipos. Este grupo incluye el hardware, los cables físicos que conectan dos equipos, las señales y el cambio de voltaje, que representan los bits individuales de una trama y la información de control necesaria para crear una trama a partir de un byte individual.

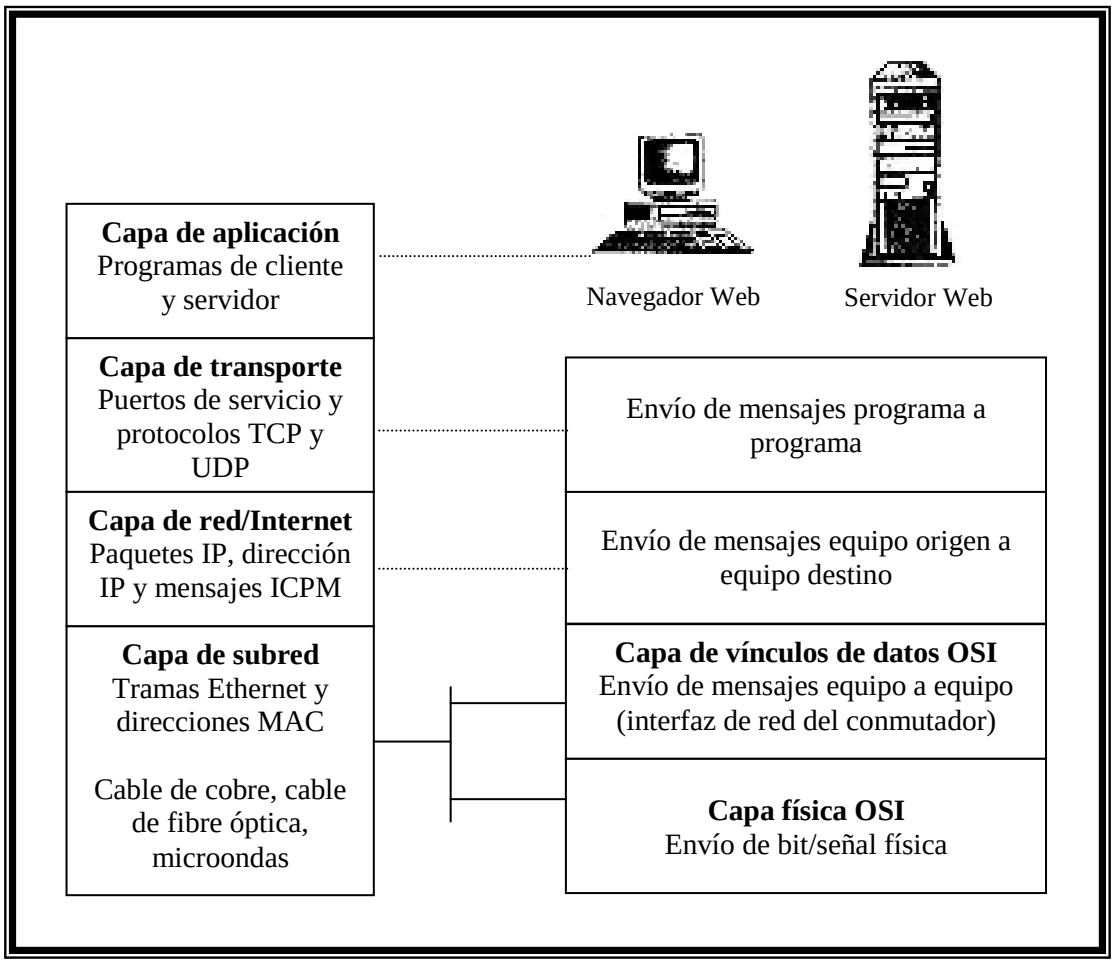


Fig. 5.02 Modelo de referencia TCP/IP

La idea general, como se muestra en la figura 5.02 es que la capa de aplicación representa la comunicación entre dos programas. La capa de transporte representa cómo se realiza la comunicación entre los dos programas. Los programas se identifican por números llamados puertos de servicio. La capa de red representa como se transporta esta comunicación entre los dos equipos

terminales. Los equipos, o sus tarjetas de interfaz de red individuales se identifican por números llamados direcciones IP. La capa de subred representa como se transporta la comunicación entre cada equipo individual a lo largo de la trayectoria. En una red Ethernet, esta interfaces de red del equipo se identifican por número llamados direcciones Ethernet o también conocidas como direcciones MAC de hardware impresas en la tarjeta de red.

5.1.2 Protocolo de Control de Transmisión (TCP)

El Protocolo de Control de Transmisión proporciona un número considerable de servicios a la capa IP y a las capas superiores. Aún de mayor importancia, proporciona a las capas superiores un protocolo orientado a conexión, que permite a una aplicación asegurarse de que un datagrama enviado sobre una red se recibió totalmente, en este papel TCP opera como un protocolo de validación de mensajes proporcionando comunicación confiable. Si un datagrama se corrompe o se pierde, por lo general es TCP (y no las aplicaciones de capas superiores) el que maneja la transmisión.

TCP maneja el flujo de datagramas provenientes de las capas superiores, así como los datagramas de llegada provenientes de la capa IP, tiene que se asegurarse de que las prioridades y la seguridad son respetadas. TCP debe ser capaz de manejar la terminación de una aplicación en una capa superior, que estaba esperando la llegada de datagramas así como fallas en capas inferiores. TCP también debe mantener una tabla de estado de todos los flujos de datos hacia dentro y afuera de la capa TCP. El aislamiento de estos servicios en una capa por separado permite que las aplicaciones se diseñen sin preocuparse del control de flujo de la confiabilidad del mensaje. Sin la capa TCP, cada

aplicación tendría que implementar estos servicios por sí misma, lo que resultaría en un desperdicio de recursos.

TCP reside en la capa de transporte, colocado encima de IP, pero debajo de capas superiores y sus aplicaciones, como se muestra en la figura 5.03, TCP sólo reside en dispositivos que realmente procesen datagramas, asegurándose de que el datagrama vaya de la fuente hacia las máquinas destino, no reside en un dispositivo que simplemente enrute datagramas, por lo que en una compuerta no hay capa TCP. Esto tiene sentido, porque en una compuerta el datagrama no tiene ninguna necesidad de ir más arriba que la capa IP.

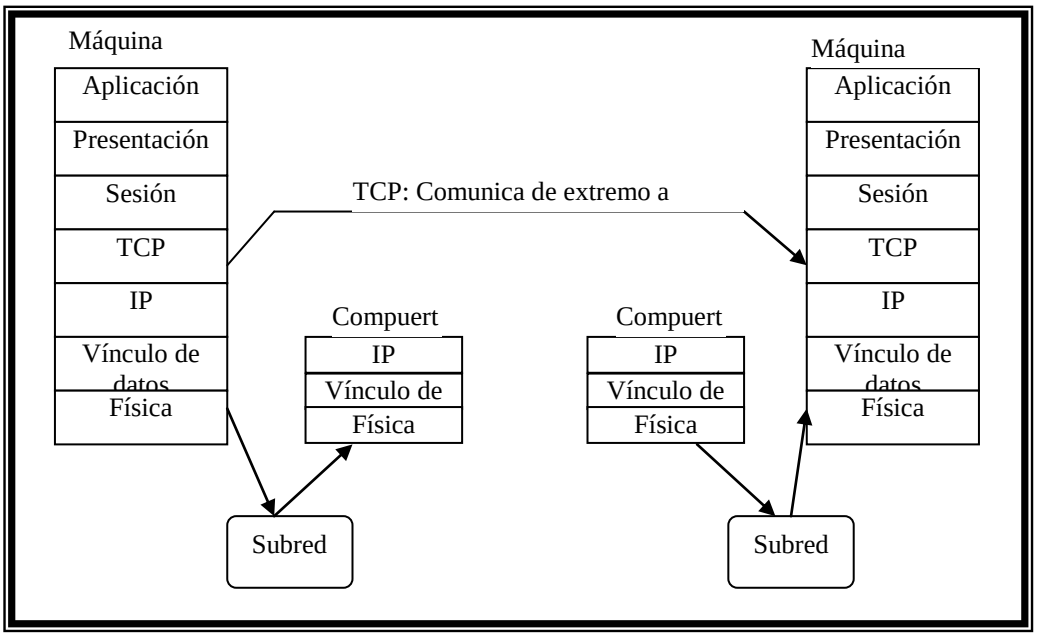


Fig. 5.03 Comunicación TCP/IP

Debido a que TCP es un protocolo orientado a conexión responsable de asegurar la transferencia de un datagrama desde la máquina fuente a la máquina destino, TCP debe recibir mensajes de comunicación de la máquina destino para acusar el recibo del datagrama. Por lo general se utiliza el término circuito virtual para referirse al saludo existente entre dos máquinas terminales, la mayor parte de los cuales son simples mensajes de

acuse de recibo (ya se confirmación de recibo o un código de fallo) y números de secuencia del datagrama.

5.1.2.1 Unidades de datos del protocolo TCP

Debido a que TCP debe comunicarse con IP en la capa de abajo utilizando un método definido por IP y con las aplicaciones de la capa superior mediante el uso de primitivas TCP-ULP. TCP también se debe comunicar con otras implementaciones TCP a través de las redes, para esto, utiliza unidades de datos de protocolo (PDUs) que en lenguaje TCP se denomina segmentos.

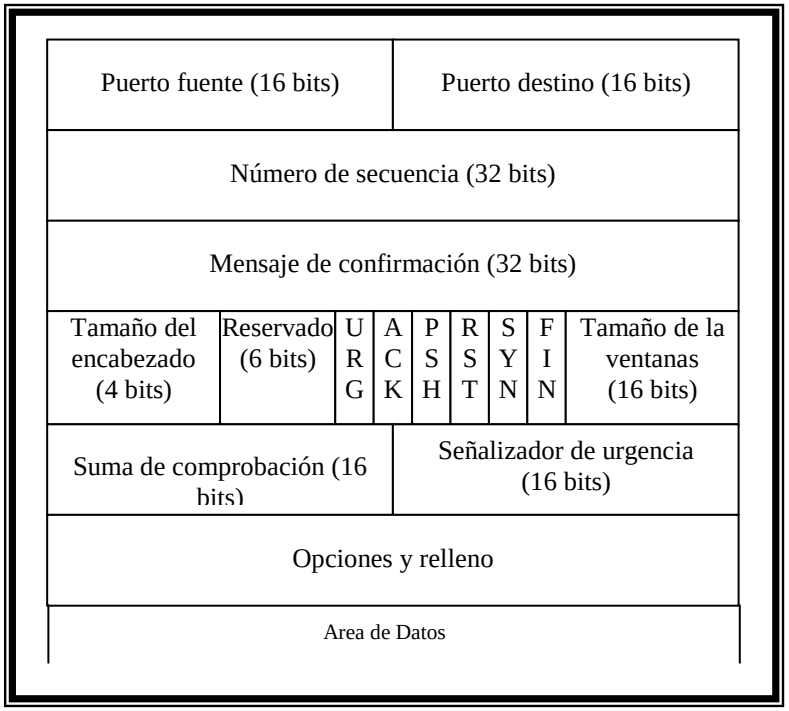


Fig. 5.04 Estructura del segmento TCP

El diseño de una PDU de TCP, comúnmente conocida como encabezado es similar al indicado en la figura5.04 y cada uno de sus campos indica el estado del encabezado.

El significado de los diferentes campos es:

Puerto fuente.- Un campo de 16 bits que identifica el puerto de la aplicación emisora.

Puerto destino.- Un campo de 16 bits que identifica el puerto de la aplicación destino.

Número de secuencia.- Identifica el primer byte de datos en el área de datos del segmento TCP.

Número de confirmación.- Identifica el siguiente byte de datos que el emisor espera recibir del flujo de información.

Longitud del encabezado.- Especifica la longitud del encabezado TCP en palabras de 32 bits.

Reservado.- Un campo de 6 bits reservado para uso futuro, los 6 bits se deben establecer en 0.

Bandera URG.- Le indica al módulo TCP receptor que el campo Indicador de urgencia está señalando datos urgentes.

Bandera ACK.- Le indica al módulo TCP receptor que el campo de Número de confirmación contiene un número de confirmación válido.

Bandera PSH.- Le indica al módulo TCP receptor que debe enviar los datos de inmediato a la aplicación destino.

Bandera RST.- Solicita al módulo TCP receptor que restablezca la conexión TCP.

Bandera SYN.- Indica al módulo receptor TCP que se deben sincronizar los números de secuencia, esta bandera se utiliza al establecerse una conexión.

Bandera FIN.- Indica al módulo TCP receptor que el emisor no tiene más datos por enviar, este es el equivalente a un marcador de fin de transmisión.

Tamaño de la ventana.- Le indica al módulo TCP receptor el número de bytes que el emisor está dispuesto a aceptar.

Suma de comprobación.- Ayuda al módulo TCP receptor a detectar la corrupción de datos.

Indicador de urgencia.- Apunta hacia el último byte de datos urgentes en el área de datos TCP.

Opciones.- Se utiliza para definir las opciones de TCP, cada opción consiste de un número de opción (1 byte), el número de bytes en la opción y los valores de la opción. Sólo son tres opciones y están actualmente definidas para TCP.

0 Lista de fin de opción
1 No operación
2 Tamaño máximo de segmento

Relleno.- Relleno par asegurar que el encabezado se completó a un múltiplo de 32 bits.

A continuación de la PDU o del encabezado aparecen los datos. El campo opciones tiene una función útil: especificar el tamaño máximo de búffer que una implementación TCP receptora puede aceptar.

Debido a que TCP utiliza áreas de datos de longitud variable, es posible que la máquina emisora genere un segmento que resulte más largo de lo que puede manejar el software receptor.

El campo suma de comprobación calcula la suma con base en todo el tamaño del segmento, incluyendo el seudoencabezado de 96 bits que se fija durante el cálculo al encabezado TCP. El seudoencabezado contiene la dirección fuente, la dirección destino, el identificador de protocolo y la longitud del segmento. Estos son los parámetro que se pasan a IP al transferir la instrucción de envío, y también son los que IP lee cuando se intenta una entrega.

5.1.2.2 Establecimiento de conexiones

Una conexión se puede establecer entre dos máquinas únicamente si hay una conexión entre dos sockets, si ambas máquinas están de acuerdo en la conexión, y si ambas máquinas tiene recursos TCP adecuados para darle servicio a la conexión, si alguna de estas condiciones no se puede cumplir, la conexión no se efectuará. La aceptación de las conexiones se puede originar desde una aplicación o desde una rutina de administración de sistema.

Cuando se establece una conexión, se le dan ciertas propiedades que se mantienen válidas hasta que se cierra la conexión, típicamente estas consistirán en un valor de precedencia y un valor de seguridad, estos parámetros se negocian entre las dos aplicaciones, cuando la conexión está en proceso de establecerse.

En la mayor parte de los casos, dos aplicaciones esperan una conexión, por lo que emiten solicitudes de apertura activas y pasiva. El la figura 5.05 se muestra un diagrama de flujo para una apertura TCP, el proceso empieza con el TCP de la máquina A recibiendo una solicitud de conexión de su ULP, el cual envía una primitiva de apertura activa a la máquina B. El segmento que se construye tendrá la bandera SYN activa y un número de secuencia asignado; el diagrama muestra lo anterior con la

notación SYN SEC 50 indicando que la bandera SYN está activa y el número de secuencia es 50.

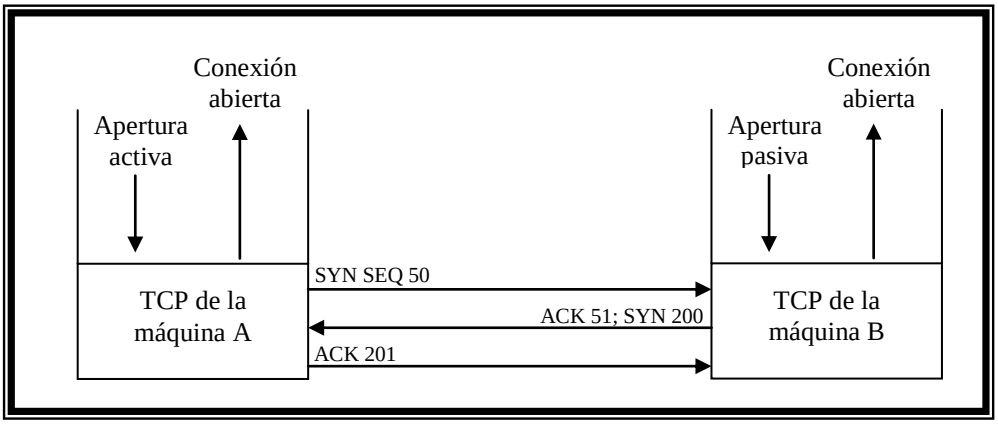


Fig. 5.05 Apertura de una conexión TCP

La aplicación de la máquina B habrá emitido una instrucción de apertura pasiva su TCP. Cuando recibe el segmento SYN SEQ 50, el TCP de la máquina B regresará un acuse de recibo a la máquina A, con el número de secuencia 51, la máquina B también definirá un número Unitial Send Sequence propio. El diagrama muestra este mensaje como ACK 51; SYN 200 indicando que el mensaje es un acuse de recibo con el número de secuencia 51, que tiene la bandera SYN activa y tiene un ISS de 200.

Al recibirlo, la máquina A regresa su propio mensaje de acuse de recibo, con el número de secuencia establecido en 201, este es el ACK 201 del diagrama, entonces después de abrir y acusar de recibo la conexión, la máquina A y la máquina B envían mensajes de apertura de conexión a través de ULP a las aplicaciones solicitantes.

No es necesario que la máquina remota tenga una instrucción de apertura pasiva, como se mencionó anteriormente, en ese caso la máquina emisora proporcionará tanto los números de socket emisor y receptor, así como los valores de precedencia, seguridad y tiempo terminado. Es común que dos aplicaciones soliciten

simultáneamente una apertura activa, esto se resuelve fácilmente, aunque involucra un poco más de tráfico de red.

5.1.2.3 Transferencia de datos

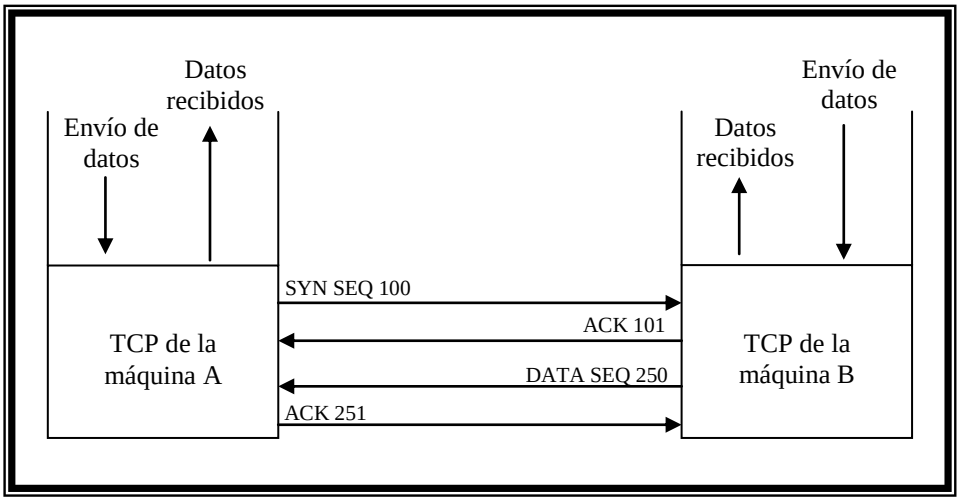


Fig. 5.06 Transferencia de datos

La transferencia de información es sencilla, el TCP de la máquina A encapsula cada bloque de datos que recibe proveniente de ULP y lo envía a la máquina B con un número de secuencia creciente. Una vez que la máquina B recibe el mensaje, confirma la recepción con un acuse de recibo de segmento que incrementa el número de secuencia siguiente.

El servicio de transporte de datos TCP de hecho incluye seis subservicios distintos:

Dúplex completo.- Habilita ambos extremos de una conexión para transmitir en cualquier momento, incluso simultáneamente.

Sincronización.- El uso de temporizadores asegura que los datos se transmiten dentro de un tiempo razonable.

En orden.- Los datos que envíe una aplicación se recibirán en el mismo orden en el otro extremo. Esto ocurre a pesar del hecho de

que los datagramas se pueden recibir en desorden a través de IP, ya que TCP reensambla un mensaje en el orden correcto antes de pasarlo a las capas superiores.

Etiquetado.- Todas las conexiones tienen una precedencia y un valor de seguridad preaceptados.

Flujo controlado.- TCP puede regular el flujo de la información mediante el uso de búffers y límites de ventana.

Corrección de errores.- Las sumas de verificación aseguran que los datos estén libres de errores.

5.1.2.4 Cierre de conexiones

Para cerrar una conexión, uno de los TCP's recibe una primitiva de cierre proveniente del ULP y emite un mensaje. En la figura 5.07, el TCP de la máquina A envía la solicitud de cierre de la conexión a la máquina B con el siguiente número de secuencia. La máquina B entonces regresará un acuse de recibo de la solicitud y su siguiente número de secuencia. A continuación, la máquina B envía el mensaje de cierre a través de su ULP a la aplicación y espera que ésta acuse recibo de cierre. Este paso no es estrictamente necesario; TCP puede cerrar la conexión sin la aprobación de la aplicación, pero un sistema con buen comportamiento informará a la aplicación del cambio de estado.

Después de recibir de la aplicación la información para cerrar la conexión, el TCP de la máquina B regresa un segmento a la máquina con la bandera FIN activa; finalmente la máquina A acusa recibo del cierre y la conexión termina.

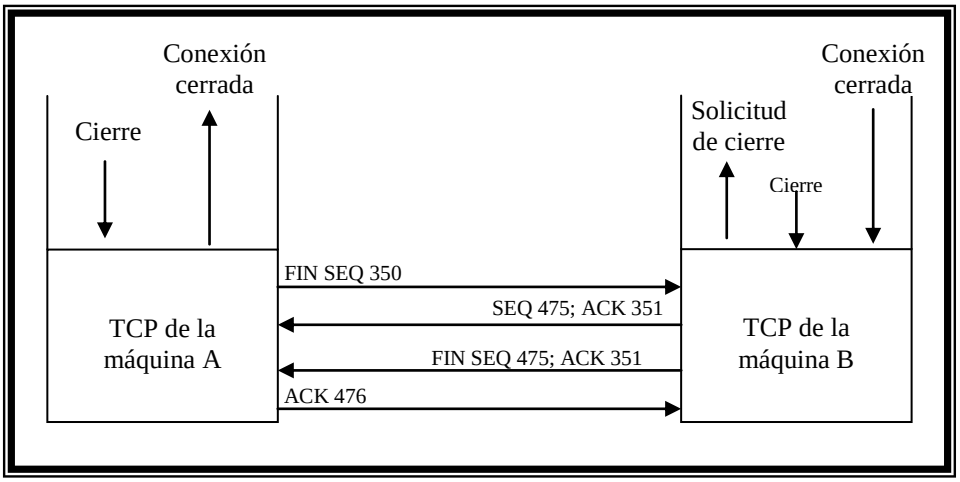


Fig. 5.07 Cierre de una conexión

Puede ocurrir una terminación abrupta de una conexión cuando uno de los extremos cierra el socket, esto se puede efectuar sin aviso a la otra máquina y sin preocuparse de que haya información en tránsito entre ambas. Aparte de cierres inesperados causados por malos funcionamientos o por pérdida de energía, una terminación abrupta la puede iniciar uno de los usuarios, una aplicación o una rutina de vigilancia de sistema, que juzgue que la conexión merece terminar, quizá el otro extremo de la conexión se de cuenta de la terminación abrupta hasta que intente enviar un mensaje y el temporizador expire.

Para llevar control de todas las conexiones, TCP utiliza una tabla de conexión; cada una de las cuales tiene una entrada en la tabla, que muestra la información sobre la conexión de un extremo a otro.

El significado de cada columna de la tabla de conexión es:

Estado.- El estado de la conexión: cerrada, en cierre, en escucha, en espera, etc.

Dirección local.- La dirección IP de la conexión, cuando se encuentra en estado de escucha, ésta se fijará en 0.0.0.0.

Puerto local.- El número de puerto local.

Dirección remota.- La dirección IP remota.

Puerto remoto.- El número de puerto de la conexión remota.

5.1.3 Protocolo de Datagrama de Usuario (UDP)

TCP es un protocolo basado en conexión pero hay ocasiones que se requiere un protocolo sin conexión, y entonces se utiliza UDP, este se emplea tanto con el Protocolo Trivial de Transferencia de Archivos (TFPT) como con el procedimiento de llamada remota (RPC). Las comunicaciones sin conexión no proporcionan confiabilidad, lo que significa que no hay indicación para el dispositivo emisor de que un mensaje se haya recibido correctamente. Los protocolos sin conexión tampoco ofrecen capacidades de recuperación de errores, mismos que se deben ignorar o compensar en capas superiores o inferiores.

UDP es mucho más sencillo que TCP, hace interfaz con IP sin preocuparse por mecanismos de control de flujo o de recuperación de errores, actuando simplemente como un emisor y receptor de datagramas.

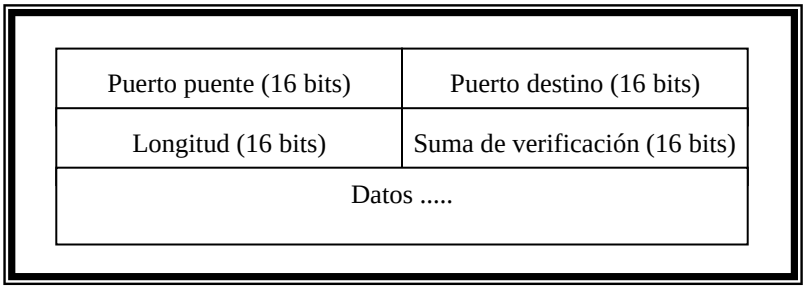


Fig. 5.08 Encabezado UDP

El encabezado del mensaje UDP es mucho más sencillo que el de TCP, en la figura 5.08 se muestra un diagrama, se puede añadir

relleno al datagrama para asegurar que el mensaje es un múltiplo de 16 bits.

El significado de los diferentes campos de un encabezado UDP es:

Puerto fuente.- Un campo opcional con el número de puerto, si no se especifica número de puerto, este campo se establece en 0.

Puerto destino.- El puerto de la máquina destino.

Longitud.- La longitud del datagrama, incluyendo al encabezado y los datos.

Suma de verificación.- El complemento a uno de 16 bits de la suma de complemento a uno del datagrama, incluyendo un seudoencabezado similar al TCP.

El campo de suma de verificación UDP es opcional; pero si no se utiliza, la suma de verificación no se aplica al segmento de datos porque la suma de verificación de IP se aplica únicamente al encabezado IP. Si no se utiliza suma de verificación, el campo se deberá establecer en 0.

5.2 ANÁLISIS DE LA POLÍTICA DE SEGURIDAD

La Política de Seguridad es un documento en el cual se detallan y especifican los aspectos de seguridad sobre los cuales trabajará la organización.

En el caso de la Universidad Técnica del Norte se ha desarrollo una política de seguridad cuya postura es "**la negación total de los servicios**", lo cual significa que únicamente los servicios que se habiliten a través del firewall estarán permitidos, caso contrario el servicio estará denegado.

En la Política también se abarcan otro tipo de medidas de seguridad como es el acceso de usuarios al servidor, privilegios sobre ciertos archivos, monitoreo del sistema, seguridad física del servidor y otros aspectos, los mismos que no serán cubiertos por el firewall, sino que al contrario se ha establecido en el servidor central aprovechando las bondades que brinda el Sistema Operativo instalado, en este caso Linux RedHat 7.1.

A continuación mencionan algunos aspectos considerados en la Política de Seguridad, pero que no son tratados por el firewall sino por configuraciones especiales realizadas en el servidor, entre estas se pueden mencionar:

- Limitaciones en el uso de recursos del sistema.
- Control de acceso de usuarios.
- Privilegios sobre archivos importantes del sistema.
- Bloqueos del comando "su" al root.
- Seguridad y optimización de la red.
- Herramientas de criptografía.
- Control de acceso remoto.
- Monitoreo del sistema.
- Configuración y optimización de servicios de red.
- Backup del sistema.

Estos aspectos mencionados anteriormente, forman parte de la Política de Seguridad debido a que mejoran el rendimiento de la red y permiten que los usuarios hagan uso de los servicios de internet con mayor seguridad de la que podrían tener al usar solamente un firewall.

Es necesario recordar que un firewall constituye una parte de la Política de Seguridad de una institución, éste por si solo no brindaría la seguridad que en la actualidad necesitan las organizaciones como para no ser víctimas de un ataque malintencionado, todos los beneficios que un firewall puede brindar, siempre deben ser reforzados por otras acciones que complementen y ayuden a poner en marcha la mejor Política de Seguridad posible para una institución.

Limitaciones en el uso de recursos del sistema

Es muy importante limitar y controlar la cantidad de recursos que cada usuario de la red puede utilizar, pues de esta manera se están evitando ataques que disminuyen el rendimiento el sistema y en ciertas ocasiones pueden dejarlo inhabilitado.

Control de acceso de usuarios

Controlar cuales usuarios tienen acceso al servidor es otro aspecto del cual nunca debe olvidarse el administrador de red, si desea que su sistema funcione adecuadamente y no sea víctima de ataques. En muchas ocasiones sucede que los problemas no se presentan por ataques remotos, sino por intrusiones de usuarios locales, por lo tanto es muy importante controlar cuales usuarios tienen permisos de conexión con el servidor y sobre todo que clase de actividades pueden realizar el momento que inician una sesión, ya sea local o remotamente.

Privilegios sobre archivos importantes del sistema

Debido a que en el servidor central se están ejecutándose varios servicios de red, existen en este los archivos de configuración de todos y cada uno de estos servicios. Estos archivos deben tener privilegios especiales, de tal manera que solamente el usuario

destinado a administrar o ejecutar un servicio sea el único que pueda modificarlos, caso contrario en cualquier momento se puede correr el riesgo de perder un servicio y generar un caos entre los usuarios de la red.

Bloqueos del comando "su" al root

Como es muy conocido el famoso comando su, permite cambiarse de usuario y en el momento que un usuario común logre cambiarse e ingresar como root puede causar daños irreparables en el sistema; por lo tanto se debe restringir al máximo que usuarios pueden hacer su al root.

Seguridad y optimización de la red

El correcto funcionamiento de la red proporciona la ventaja de brindar un mejor servicio a los usuarios, por lo tanto definir parámetros que optimicen el rendimiento de esta, es una buena medida para lograr el bienestar de los miembros de la institución.

La red al igual que cualquier otro servicio que pueda ejecutarse en el servidor, tiene archivos de configuración muy importantes, y la sola modificación de uno de estos puede causar la degradación total del sistema, por lo tanto estos deben tener todas las medidas de seguridad necesarias para no ser alterados o eliminados, ya sea por un error involuntario o voluntario.

Dentro de la seguridad de la red, también se pueden mencionar ciertos parámetros que permiten controlar accesos que en muchas ocasiones parecen normales pero que en realidad lo único que buscan es saturar el sistema y hecharlo a perder.

Herramientas de criptografía

En la actualidad es preferible que las contraseñas y cierta información confidencial no viaje en formatos entendibles para cualquier persona, lo más adecuado es usar herramientas que codifiquen este tipo de información y de esta manera se logre un poco más de seguridad en la información que se transmite a través de la red.

Control de acceso remoto

En muchas ocasiones es necesaria una administración remota del servidor, ya sea porque el administrador está en otro lugar o por cualquier otro motivo, pero en el caso de suscitarse una situación de este tipo, es conveniente disponer de una herramienta que garantice que la información que se transmite no será interceptada o descubierta por espías.

Además es sumamente importante establecer que un único usuario puede realizar acceso remoto y de ser preferible que lo realice desde una máquina específica, esto garantiza la seguridad del sistema y evita que personas mal intencionados ganen acceso con usuarios conocidos.

Monitoreo del sistema

Algo muy importante dentro de la tarea del administrador de red es saber que sucede todos los días con el servidor, es necesario tener un conocimiento pleno de que información pasó a través del equipo, quien hizo uso de algún servicio, a que usuario se le rechazó la conexión, que puertos pueden estar siendo víctimas de ataques, etc. en fin un sin número de actividades que deben ser registradas y por supuesto deben ser conocidas y analizadas de

una manera rápida, para de esta manera poder responder de forma apropiada.

Las herramientas de monitoreo ayudan al administrador de red a saber con exactitud que sucede en el servidor; sin embargo este siempre deberá revisar los logs del sistema para tener un conocimiento más claro de que es lo que está sucediendo en la red.

Configuración y optimización de servicios de red

Debido a que se desea brindar acceso a Internet a los usuarios de red interna, es necesario configurar varios servicios tales como: DNS, Mail, FTP, Telnet, IMAP, WWW, etc., todos y cada uno de estos tiene que ser habilitado de una manera que presten el servicio para el que fueron creado pero sin convertirse en puntos débiles y susceptibles de ser atacados.

Con esta finalidad se han configurado estos servicios aprovechando todas las características de seguridad que vienen incorporadas en estos paquetes y que pueden ser utilizadas para mejorar la seguridad del sistema.

Backup del sistema

Dentro de una red siempre es necesario realizar respaldos periódicos de la información importante y mucho más en el caso de tratarse de un servidor que está brindando una gran variedad de servicios. Los backups serán de mucha ayuda en el caso de que llegará a suceder algo en el equipo, pues simplemente se restaurarían estas copias y no se perdería tanto tiempo volviendo a configurar todos los servicios y las opciones de seguridad que cada uno de estos posee.

Para finalizar este análisis de la Política de Seguridad, cabe mencionar que esta no solamente puede implementarse a través del firewall, al contrario para lograr un funcionamiento correcto de la Política son necesarias muchas otras actividades complementarias, pero la más importante es la **colaboración del personal de la red interna,** estos constituyen el pilar fundamental para que la Política funcione adecuadamente, porque en muchas ocasiones se pueden crear verdadero muros impenetrables para usuarios externos, pero para los internos, el único muro que se puede crear es la conciencia y la buena voluntad para que la red siempre funcione adecuadamente.

5.3 ASPECTOS ECONÓMICOS PARA LA IMPLEMENTACIÓN DE FIREWALLS DE SOFTWARE

Para la implementación de un firewall de software, el aspecto económico realmente no es muy importante, pues estos no son demasiado costosos debido a que pueden ser desarrollados en herramientas comunes y fáciles de adquirir en el mercado.

Al desarrollar un firewall de software lo costoso puede ser el equipo en el que se va instalar el firewall, pues este debe ser un computador adecuado para de esta manera lograr un rendimiento apropiado del equipo, y no tener que disminuir su capacidad de funcionamiento por la herramienta de software desarrollada.

Para la implementación de un firewall de software se debe contar con personal preparado para el trabajo, las personas que integren el grupo que se dedica al desarrollo de un firewall de software tiene que tener una capacitación constante sobre los avances de la tecnología y los lenguajes de programación, pues en cualquier momento pueden aparecer lenguajes más sofisticados que

permitan obtener un software de grandes prestaciones y no muy complicado, sino al contrario de instrucciones fáciles y simples de comprender.

Un firewall de software también puede adquirirse, no es necesario desarrollar uno para la institución, estos no son demasiado costosos, en este caso los gastos estarían relacionados con la adquisición de software, la documentación y el soporte técnico que en algunos casos es necesario contratar para instalar herramientas un poco complicadas.

Otra opción es conseguir un firewall gratuito en el Internet, son también una buena opción pero la situación se complica en el hecho de que puede ser que no exista mucha información disponible para la administración del firewall y en ese caso lo único que se haría es perder tiempo sin conseguir los resultados apropiados y la pérdida de tiempo es realmente un gasto irrecuperable.

En la Universidad Técnica se ha instalado un firewall de software, completamente gratuito, debido a que este viene con el sistema operativo Linux RedHat 7.1. Este un software muy estable y recomendado, ya que Linux es un sistema operativo reconocido a nivel mundial y en la actualidad se encuentra ejecutándose en la mayoría de servidores internet del mundo.

El firewall instalado es un firewall de filtrado de paquetes, no necesita grandes requerimientos de hardware debido a que el sistema operativo en si no es muy complicado. Este firewall requiere tener cierto soporte en el kernel y un archivo de configuración en el cual se listan las reglas de filtrado de paquetes, las cuales constituyen un problema del software, pues son bastante complicadas y es necesario estudiarlas para comprender y optimizar el uso del firewall.

Finalmente se puede decir que un firewall de software no es costoso de implementar, existen varias posibilidades para instalar uno en la red privada, lo único que se debe hacer es un análisis de cual es la mejor opción y de acuerdo a ello decidirse por uno y empezar a complementar la seguridad de la red.

5.4 DETERMINACIÓN DE LOS COMPONENTES DE UN FIREWALL DE SOFTWARE

Los componentes necesarios para un firewall de software realmente no son muy complicados, pues básicamente se necesitaría el equipo en el cual se va a instalar el cortafuego y el software, que constituye el firewall en si mismo.

En el caso de desear construir el software, se necesitaría un equipo en el cual desarrollar el sistema, este no necesariamente tiene que ser un computador muy sofisticado simplemente tiene que ser un equipo que permita instalar la herramienta en la que se va a programar, compilar los programas fuente y realizar las pruebas para determinar si el firewall realiza las tareas para las cuales fue creado.

Finalmente se puede decir que en cuanto a componentes necesarios para la implementación de un firewall de software no existe ningún otro requerimiento, pues como se dijo anteriormente estos son bastante simples en cuanto a componentes y dependiendo de los requerimientos del software pueden fácilmente instalarse en equipos sin mucha tecnología actualizada.

5.5 ELECCION DEL TIPO DE FIREWALL A IMPLEMENTARSE

En la Universidad Técnica del Norte se han implementado dos firewall de software. El primero es un programa comercial que se distribuye gratuitamente con el sistema operativo Linux RedHat 7.1, y el segundo es una aplicación que se ha desarrollado mediante Java.

Anteriormente se ha hablado sobre la complementación que debe existir entre la todas las medidas de seguridad que puedan establecerse en el servidor central y el firewall que se desee tener en la organización. Para lograr este objetivo, se procedió a instalar un Servidor de Acceso a Internet con el sistema operativo Linux RedHat 7.1 y se aprovecharon todas las medidas de seguridad que este sistema operativo brinda para establecer la seguridad en los diferentes servicios, la misma que al final se complementa con el firewall que se instaló.

Linux es un sistema operativo ampliamente conocido en la actualidad y el cual es empleado por la mayoría de servidores internet que existen en el mundo; es bastante estable y debido a que es de código abierto se pueden conseguir nuevos parches y herramientas de seguridad gratuitas y de fácil administración.

Para obtener la mayor seguridad posible, se ha realizado un estudio detallado de todos los servicios que se pueden brindar, y de las medidas de seguridad que se deben tomar, para que los mismos se ejecuten en un ambiente seguro y confiable, no solamente para los usuarios internos sino también para los usuarios externos que acceden a los servicios que brinda esta red. Una vez que los servicios se han configurado de manera segura, es momento de instalar un firewall, el cual ayudará a mejorar la

seguridad, para esto se instaló una herramienta que viene incluida en el sistema operativo y que se llama IPTABLES.

Iptables es un firewall de filtrado de paquetes, trabaja con la versión de Kernel 2.4 del sistema operativo Linux RedHat 7.1. Se escogió instalar esta herramienta ya que no tiene ningún costo adicional y viene como parte del sistema operativo, además que se conoce que es muy estable y brinda un nivel de seguridad apropiado.

Esta es una excelente herramienta debido a que permite realizar un control de los accesos al servidor y del funcionamiento de los servicios que se ejecutan en este, pues es a través del firewall que se permiten o niegan los servicios que están disponibles en el servidor. No basta con tener un servicio instalado y configurado, es indispensable establecer los permisos necesarios para que el servicio este disponible a través del firewall, caso contrario no se podrá hacer uso del mismo, aunque éste se encuentre ejecutándose en el servidor.

Este tipo de control constituye una gran ayuda para el administrador, pues además de saber como se maneja la seguridad de cada servicio individualmente, el firewall le ayudará en caso de que existan errores en las configuraciones, pues éste bloqueará cualquier acceso y permitirá al administrador conocer si algo esta mal o si alguien está tratando de ingresar al servidor sin tener los privilegios adecuados.

El segundo firewall que se implementó, es un prototipo de firewall basado en proxies, es decir se creó un proxy que puede controlar los accesos de los clientes de la red interna hacia Internet.

En este firewall se adoptó la política "**lo que no está expresamente prohibido, está permitido**", se decidió por esta

postura para demostrar las dos formas de funcionamiento de un firewall.

A través de un firewall proxy, no solamente se puede controlar el acceso a Internet, sino que además se permite que varios usuarios tengan acceso a la red de redes a través de un solo equipo, pues en realidad únicamente el computador donde se ejecuta el servicio proxy es el que está enlazado a Internet, los demás computadores de la red se conectan al servidor proxy y por medio de éste al Internet.

Para finalizar, se debe mencionar que los firewalls elegidos para implementarse tienen cada uno sus ventajas y obviamente, como cualquier otra aplicación tiene inconvenientes, pero lo realmente importante es que entre los dos, se logre establecer la mayor seguridad posible, se pueda controlar los accesos y sobre todo se tenga un registro de las actividades del sistema.

5.6 IMPLEMENTACION DE UN PROTOTIPO DE FIREWALL DE SOFTWARE UTILIZANDO JAVA

Para la implementación de un prototipo de firewall de software se escogió Java, debido a la gran capacidad de éste para escribir programas que utilizan e interactúan con recursos de Internet y la World Wide Web.

Los ordenadores que se ejecutan en Internet se comunican unos con otros utilizando los protocolos TCP y UDP que son protocolos de 4 capas, cuando se escriben programas Java que se comunican a través de la red, se está programando en la capa de aplicación, típicamente no se necesita trabajar con las capas TCP y UDP, en su lugar se puede utilizar las clases del paquete java.net; las mismas que proporcionan comunicación de red independientemente del sistema.

Antes de proceder con el desarrollo de la aplicación , es necesario conocer algunos temas que se relacionan con las clases del paquete java.net y que son la base para el desarrollo del programa.

5.6.1 Puertos

Generalmente hablando, un ordenador tiene una solo conexión física con la red, todos los datos destinados a un ordenador particular llegan a través de la conexión, sin embargo, los datos podrían ser utilizados por diferentes aplicaciones ejecutándose en el ordenador. ¿Entonces cómo se sabe el ordenador a qué aplicación enviarle los datos?, a través del uso de los puertos.

Los datos transmitidos por internet están acompañados por una información de dirección que identifica el ordenador y el puerto a la que están destinados. El ordenador está identificado por su dirección de 32 bits, esta dirección se utiliza para enviar los datos al ordenador correcto en la red. Los puertos están identificados por un número de 16 bits, que TCP y UDP utilizan para enviar los datos a la aplicación correcta.

En aplicaciones basadas en la conexión , una aplicación establece una conexión con otra aplicación uniendo un socket a un número de puerto, esto tiene el efecto de registrar la aplicación con el sistema para recibir los datos destinados es ese puerto. Dos aplicaciones no pueden utilizar el mismo puerto: intentar acceder a un puerto que ya está utilizado, generará un error.

Los números de puertos tienen un rango de 0 a 65535, los puertos 0 - 1023 están reservados para servicios bien conocidos como HTTP, FTP y otros servicios del sistema.

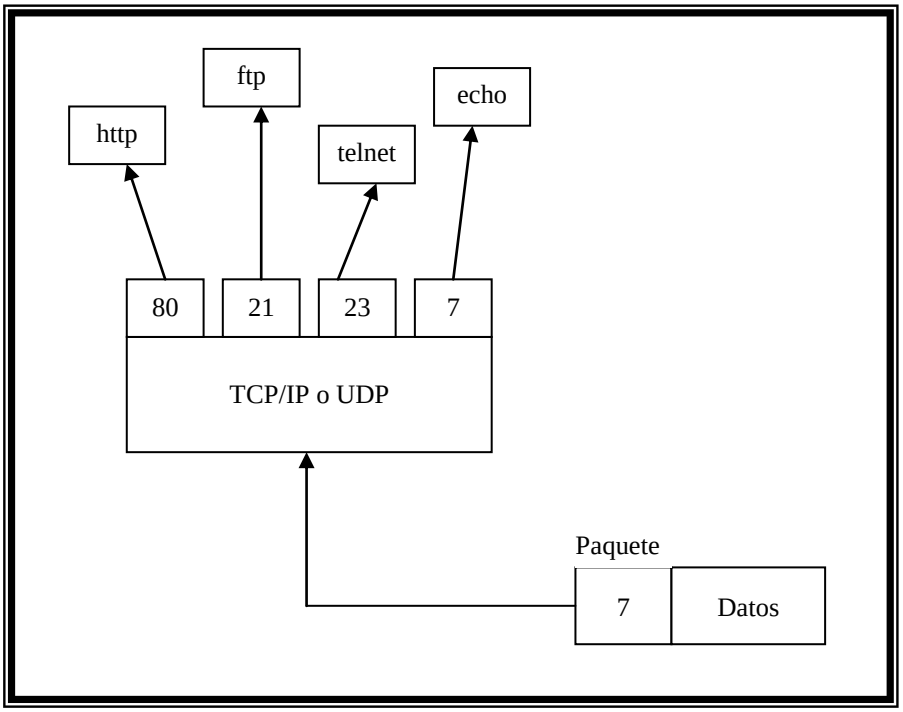


Fig. 5.09 Puertos

Por medio de las clases del paquete java.net, los programas Java pueden utilizar TCP o UDP para comunicarse a través de Internet. Las clases URL, URLConnection, Socket y SocketServer utilizan TCP, y son las clases que se emplearán para el desarrollo del prototipo de firewall de software.

5.6.2 Sockets

Una aplicación servidor normalmente escucha a un puerto específico esperando una petición de conexión de un cliente, cuando esta llega, el cliente y el servidor establecen una conexión dedicada sobre la cual poder comunicarse. Durante el proceso de conexión, el cliente es asignado a un número de puerto, y ata un socket a ella; el cliente habla al servidor escribiendo sobre el socket y obteniendo información del servidor cuando lee de él. Similarmente, el servidor obtiene un nuevo número de puerto local para poder continuar escuchando para petición de conexión del puerto original. El servidor también conecta un socket a este

puerto local y comunica con él mediante la lectura y escritura sobre él.

El cliente y el servidor deben ponerse de acuerdo sobre el protocolo, esto es, deben ponerse de acuerdo con el lenguaje para transferir la información de vuelta a través del socket.

El paquete java.net del entorno de desarrollo de Java proporciona una clase Socket que representa un final de una comunicación de dos vías entre un programa java y otro programa de la red. La clase Socket y ServerSocket implementa el lado del servidor de un enlace de dos vías y son las que básicamente se emplean para establecer la comunicación entre el cliente y servidor.

5.6.3 Manejo de comunicaciones en Java

Java ofrece varias capacidades integradas de trabajo de red que facilitan el desarrollo de aplicaciones basadas en Internet y la Web. Java no sólo puede especificar paralelismo mediante multihilado; también habilita a los programas para buscar información en todo el mundo y colaborar con programas que se ejecutan en otras computadoras internacionalmente, nacionalmente o sólo dentro de la organización.
Debido a este gran potencial que tiene Java para el desarrollo de aplicaciones de red y con el fin de que posteriormente se comprenda cual es el funcionamiento de servidor proxy desarrollado, es conveniente a indicar la manera como Java maneja la comunicación entre el cliente y el servidor.

5.6.3.1 Servicio orientado a conexión (Stream Socket)

El uso de este tipo de sockets permite a las aplicaciones cliente y servidor disponer de un stream que facilita una comunicación

libre de errores, esto va a ser muy útil siempre cuando se desea fiabilidad en la comunicación.

El comportamiento para usar este tipo de socket es diferente en el cliente y el servidor, cada uno de ellos utilizará unos métodos distintos. El esquema básico pasa por suponer que el servidor adoptará un papel pasivo y procederá a esperar conexiones de los posibles clientes. Mientras que los clientes serán los encargados de solicitar conexiones a los servidores de forma activa.

5.6.3.2 Creación del ServerSocket

En Java existen dos constructores para crear un ServerSocket:

```
public ServerSocket(int port);
public ServerSocket(int port, int count);
```

En el primer caso se crea un socket local al que se enlaza el puerto especificado y acepta hasta 50 peticiones en cola (pendientes) de conexión por parte de los clientes.

En el segundo caso se puede especificar el número máximo de peticiones de conexión que se pueden mantener en cola.
En cualquiera de los dos casos, un puerto 0 indica que se utilice cualquier puerto disponible.

En la mayoría de los casos no suele importar en qué puerto se halle el servidor, pero es fundamental que el puerto escogido sea conocido por el cliente, ya que, de no ser así no se podrá establecer la conexión.

En otras ocasiones se trata de protocolos normalizados que tendrán que atender en un puerto local específico. Así, por ejemplo, el servidor de SMTP (Simple Mail Protocol) para correo

electrónico escucha siempre en el puerto 25 o el de HTTP (HyperText Transfer Protocol), el del WWW, escucha siempre en el 80. Los primeros 1024 puertos (del 0 al 1.023) son de uso reservado para el sistema.

5.6.3.3 Espera de conexiones de clientes

Sobre un ServerSocket se puede realizar una espera de conexión por parte del cliente mediante el método accept(). Este método es bloqueante, el proceso espera a que se realice una conexión por parte del cliente para seguir su ejecución.

```
public Socket accept();
```

Una vez que se establece una conexión por el cliente, el método accept() devuelve un objeto de tipo Socket, a través del cual se establecerá la comunicación con el cliente.

Los objetos de tipo ServerSocket únicamente sirven para aceptar llamadas de clientes, no para establecer comunicaciones con los clientes. Es una especie de encargado de recibir llamadas (peticiones de conexión) que no es capaz de completarlas, tan sólo nos avisa cuando se han producido y proporciona la información necesaria para que podamos completarlas mediante la creación de un objeto Socket.

5.6.3.4 Operaciones en el cliente

Como ya se ha indicado, es el cliente el que iniciará activamente el proceso de conexión. Para poder conectar con algún servidor, el cliente necesita varias cosas:

1. Conocer la dirección IP dónde reside el servidor

2. Conocer el puerto en el que está esperando conexiones el servidor

Desde el punto de vista del cliente, será necesario realizar las peticiones a una dirección destino determinada en la que se encontrará esperando el servidor (como se indicó anteriormente, esto supone especificar la dirección IP más el número de puerto).

Existen cuatro constructores en Java para implementar un socket:

```
public Socket(InetAddress address, int port);
public Socket(InetAddress address, int port, boolean stream);
public Socket(String host, int port);
public Socket(String host, int port, boolean stream);
```

Para la creación de un Socket, hay que proporcionar la dirección (o nombre) del host y el puerto, del host remoto.

Servidor: 130.1.1.20	**Cliente: 130.1.1.14**
`ServerSocket(1500)` `accept()`	`Socket("130.1.1.20",1500)`
`Socket` "130.1.1.14",1024,1500	`Socket` "130.1.120",1500,1024

Fig. 5.10 Socketes en Java

5.6.3.5 Envío y recepción de datos a través de sockets

Como se representa en la figura 5.11, el servidor crea un socket (ServerSocket), le asigna una dirección y un puerto y acepta llamadas (accept). Tras el accept, el proceso queda bloqueado a la espera de recibir una llamada. Una vez se recibe una llamada

(Socket cliente con la dirección y puerto del servidor), el accept crea un nuevo socket, por lo que todo servidor orientado a conexión requerirá, al menos, dos sockets, uno para recibir conexiones y otro para procesarlas. Cuando un cliente desea comunicarse, crea su socket (socket), y establece una conexión al puerto establecido. Es únicamente en ese momento cuando existe la conexión y ésta durará hasta que se libere (close()).

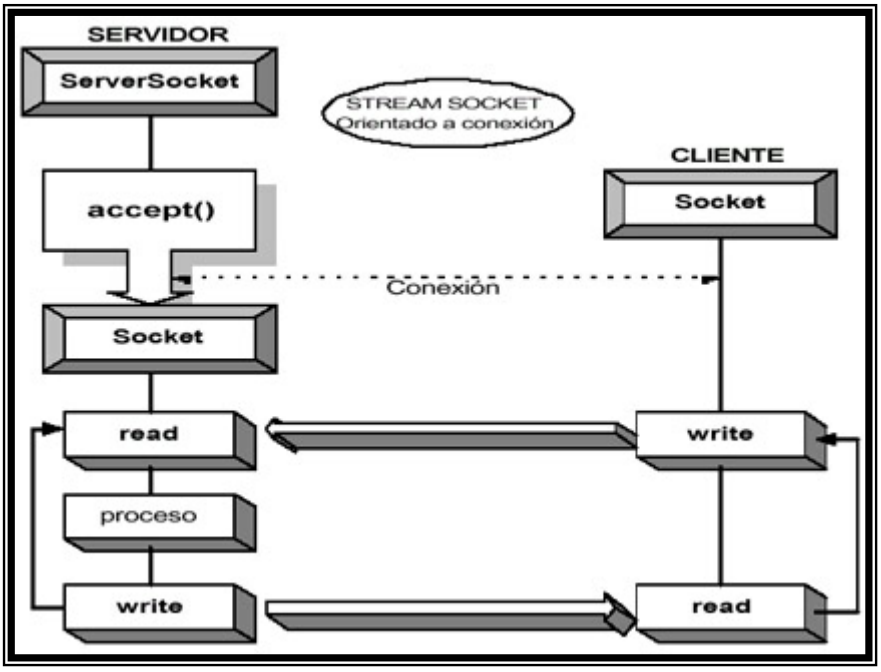


Fig. 5.11 Envío y recepción de datos

Los sockets tienen asociados un Stream de entrada (InputStream) y otro de salida (OutputStream) a través de los cuales se puede leer y escribir datos respectivamente.

La forma de obtener estos streams a partir del socket es la siguiente:

objetoDeTipoSocket.getInputStream() - Devuelve un objeto de tipo InputStream

objetoDeTipoSocket.getOutputStream()- Devuelve un objeto de tipo OutputStream

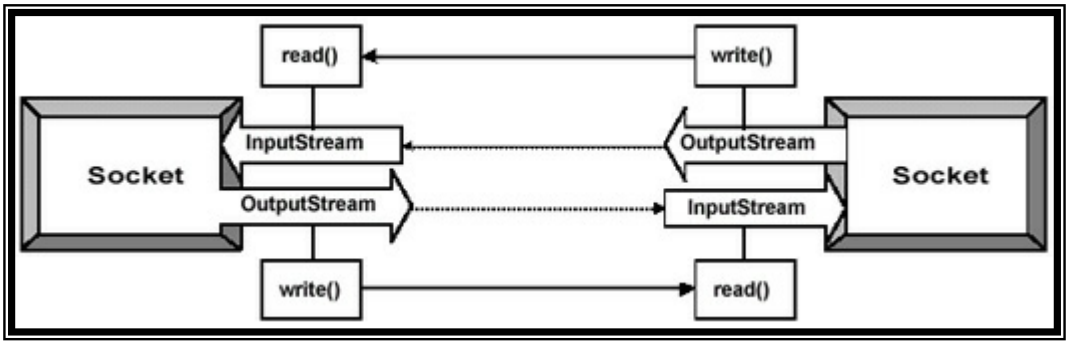


Fig. 5.12 Streams sobre sockets

Envío de datos

Para enviar datos, pueden utilizarse los OutputStream de los socket directamente si lo que se pretende enviar es un flujo de bytes sin buffer, o puede crearse un objeto de tipo stream de datos más evolucionado basado en el OutputStream que proporciona el socket.

Recepción de datos

Para recibir datos, al igual que para el envío, puede utilizarse el InputStream que proporciona el socket o definir un nuevo objeto de tipo stream más eficiente.

5.6.4 Prototipo de firewall de software

El prototipo de firewall desarrollado es un proxificador GET HTTP, es decir es un servidor que sirve peticiones de clientes que solicitan conexiones GET con Internet.

En primer lugar se debe iniciar el servidor, el mismo que estará escuchando en el puerto 8080, una vez inicializado el servidor es necesario establecer en los clientes, en este caso en los navegadores de Internet, que utilicen servidor proxy para conectarse con Internet, luego de esto los clientes podrán

comunicarse de manera transparente con cualquier sitio web que deseen.

El momento de iniciar el servidor, se presenta una ventana en la cual se indica que el servidor está ejecutándose y en espera de peticiones de los clientes, mientras no se detecte ninguna conexión de los clientes con el servidor proxy, este permanecerá en estado pasivo, es decir no realizará ninguna tarea mientras no sea necesario.

Para detener el servidor, se debe cerrar la ventana que se abrió cuando el servidor se inició, de esta forma se cierra la conexión con los clientes y esto nos podrán acceder a Internet, logrando de esta manera controlar el acceso de los usuarios de la red interna.

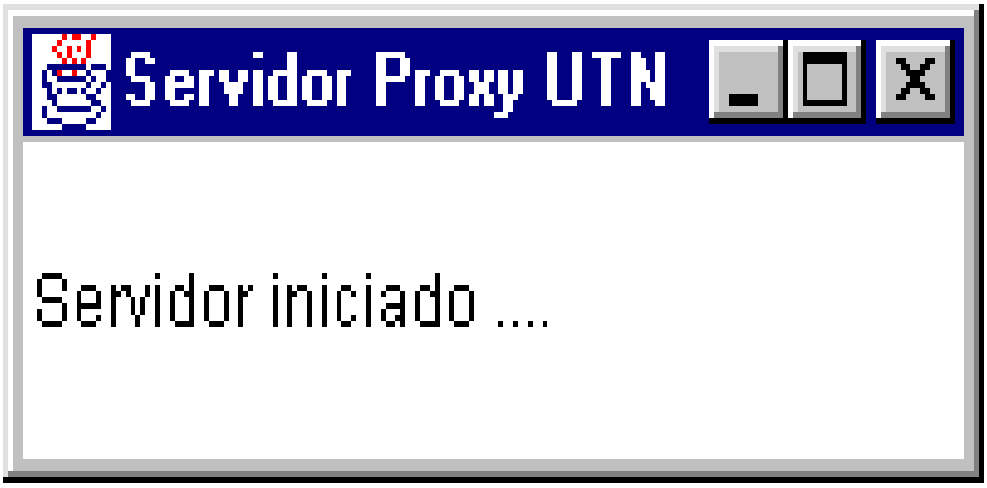


Fig. 5.13 Inicio del Servidor Proxy

El trabajo del servidor proxy consiste en bloquear accesos a Internet, los mismos que pueden ser:

- Accesos a dominios restringidos
- Accesos desde máquinas establecidas

Esto quiere decir que el servidor puede bloquear accesos a dominios internet como www.algunlugar.com y máquinas con direcciones IP establecidas. Para realizar esta tarea el servidor

utiliza dos archivos en los cuales el administrador añadirá nuevos dominios o direcciones IP a los cuales se restringir la conexión.

Para realizar el control de la dirección IP, existe un archivo llamado ip_deny, en el cual se encuentran todas las direcciones de los clientes de la red interna que tienen no permiso para conectarse con el servidor. Una vez que el cliente solicita una conexión, el primer paso es conocer la dirección IP de la máquina que se conecta y someterla a un proceso de análisis para determinar si esta existe o no en el archivo ip_deny, si la respuesta es positiva, se envía al cliente una notificación de que su acceso está restringido y se almacena en el archivo ip_deny_log el intento de acceso del cliente, para que de esta manera el administrador pueda conocer a los clientes restringidos que desean acceder a internet sin tener los privilegios adecuados.

Si la dirección IP del cliente no se encuentra en el archivo ip_deny, entonces se procede a analizar la URL solicitada; de igual manera que en el caso de las direcciones IP existe un archivo llamado dominio_deny, en el cual se especifican los dominios restringidos. Similarmente al de las direcciones IP, si la URL solicitado se encuentra en el archivo dominio_deny, se envía una notificación de que el dominio está restringido y se almacena en dominio_deny_log el intento de acceso.

Finalmente si la dirección IP y la URL no están restringidas, entonces el servidor procede a conectarse con la URL solicitada y entregará al cliente la respuesta que el sitio web devuelva al servidor proxy.

El administrador puede añadir, eliminar o revisar el contenido de los archivos ip_deny y dominio_deny a través de una herramienta administrativa desarrollada en Java, la misma que permite

manipular la información de estos archivos de una manera más óptima.

Las opciones del menú principal de la herramienta de administración del firewall son:

- Dominios
- Direcciones IP
- Logs
- Salir

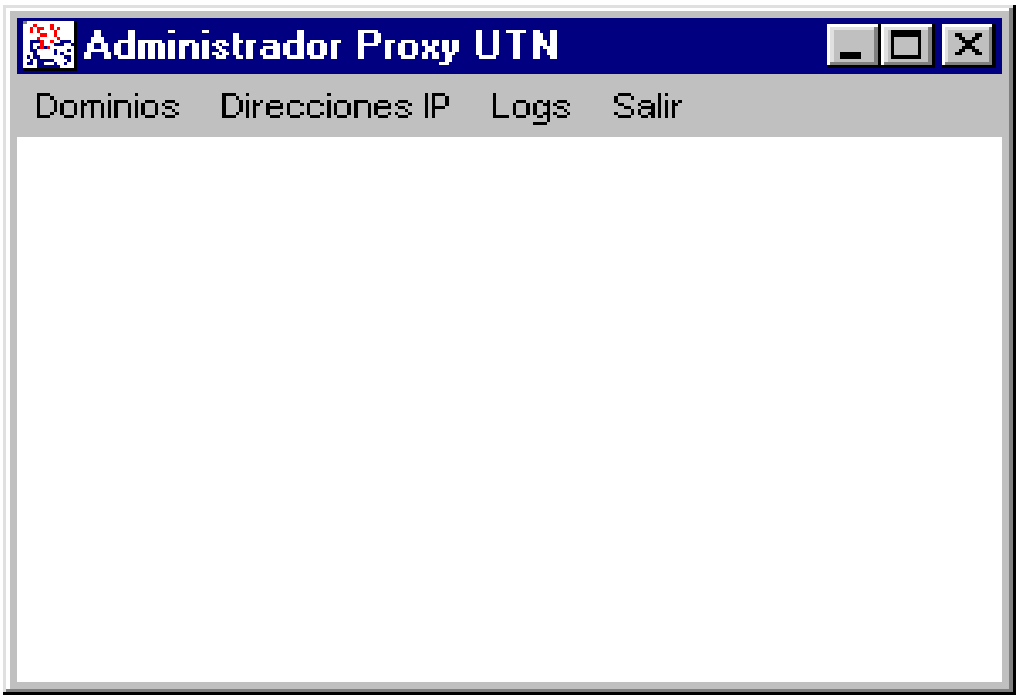


Fig. 5.14 Opciones de Administración

A través de la opción ***Dominios*** se puede:

- Añadir nuevos dominios a restringirse
- Eliminar dominios almacenados
- Revisar el contenido del archivo dominio_deny

Fig. 5.15 Opciones del menú Dominios

Agregar Dominios.- Esta opción permite agregar nuevos dominios a restringirse, los mismos que se almacenan en el archivo dominio_deny.

Fig. 5.16 Agregar dominios

Eliminar Dominios.- Esta opción permite eliminar dominios que se encuentran almacenados en el archivo dominio_deny.

Fig. 5.17 Eliminar dominios

Revisar.- La opción revisar permite visualizar el contenido del archivo dominio_deny.

Fig. 5.19 Revisar un archivo

Con la opción ***Direcciones IP***, se puede:

- Añadir nuevas direcciones IP.
- Eliminar direcciones IP almacenadas.
- Revisar el contenido del archivo ip_deny.

Las pantallas para realizar estas tareas son similares a las empleadas en la opción Dominios.

La opción ***Logs*** permite revisar los archivos de log que se generan cada vez que el servidor ejecuta alguna tarea, como se sabe una

parte importante del trabajo de una herramienta que controla el flujo de información entre dos redes, es almacenar que es lo que sucede en entre estas dos redes.

A través de la opción logs se puede:

- Revisar el archivo dominio_deny_log
- Vaciar el contenido del archivo dominio_deny_log
- Revisar el archivo ip_deny_log
- Vaciar el contenido del archivo ip_deny_log
- Revisar el archivo access_log
- Vaciar el contenido del archivo access_log

Fig. 5.20 Opciones de Logs

Revisar dominio.- Permite revisar el contenido del archivo dominio_deny_log, en el cual se almacenan los intentos de usuarios de la red interna de acceder a dominios que están restringidos.

Limpiar dominio.- Vacía el contenido del archivo dominio_deny_log, para así tener un archivo en blanco cuando este se llene con demasiada información.

Revisar dirección IP.- Permite revisar el contenido del archivo ip_deny_log, en el cual se almacenan los intentos de acceso por parte de usuarios que se encuentran restringidos para acceder a Internet.

Limpiar dirección IP.- Vacía el contenido del archivo ip_deny_log, para así tener un archivo en blanco cuando este se llene con demasiada información.

Revisar accesos.- Permite revisar el contenido del archivo access_log, el cual almacena las conexiones que el servidor realiza con servidores reales de Internet.

En este archivo se almacenan los siguientes datos, ya que se consideran son importantes para analizar la conexión solicitada:

- Fecha
- Hora
- Dirección IP del cliente
- Página Web solicitada

De los datos almacenados en el archivo access_log, los que más en cuenta debe tener el administrador son: la dirección IP del cliente y la página web solicita, debido que ciertos usuarios pueden estar accediendo a lugares no apropiados pero que todavía no están restringidos, ya que en muchas ocasiones el administrador puede no conocer ciertas direcciones que no

deberían ser permitidas, pero con la información almacenada en este archivo puede conseguirlas y bloquearlas inmediatamente.

El archivo access_log también puede ser revisado a través de una herramienta administrativa, logrando de esta manera facilitar el trabajo del administrador ya que le permite conocer que sitios web han sido visitados y por cuales clientes de la red interna, el contenido del archivo access_log es similar al indicado en la fig. 5.21.

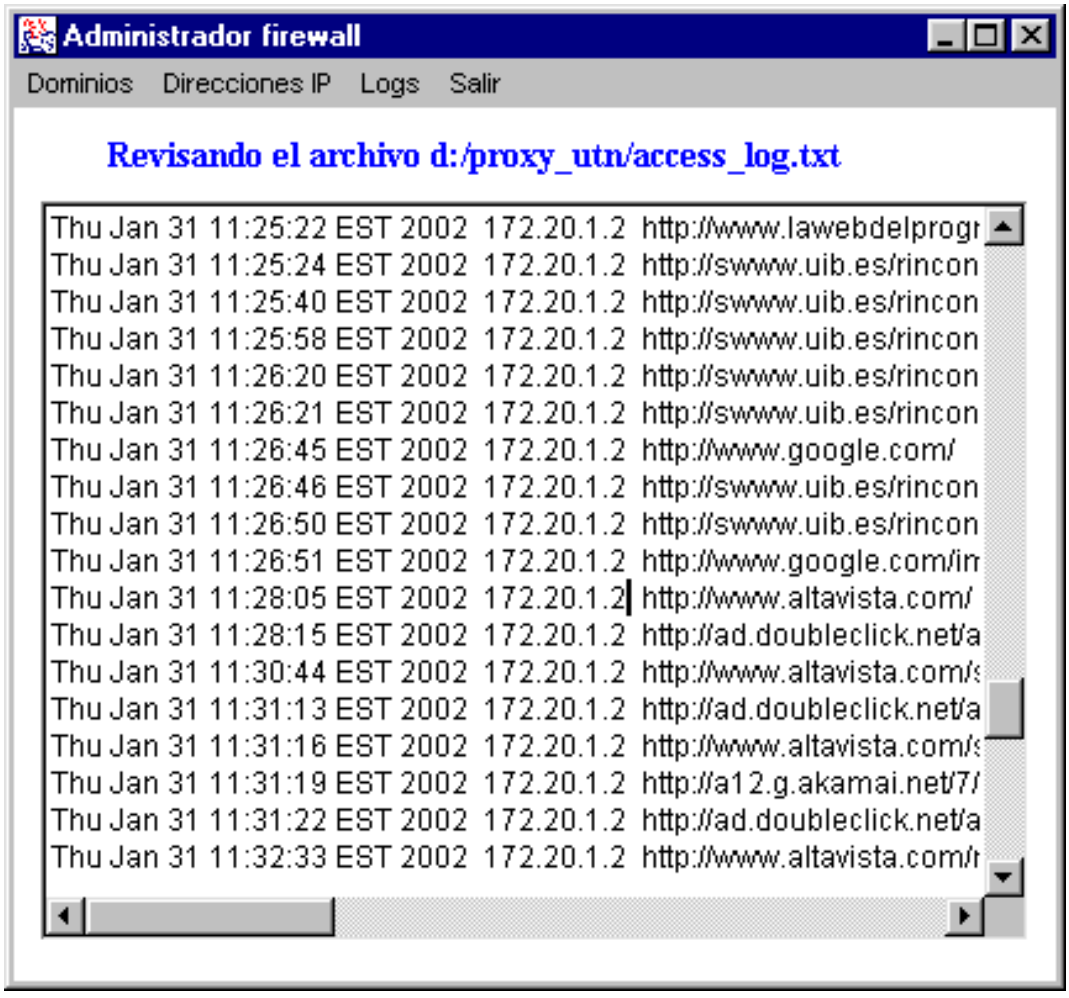


Fig. 5.20 Archivo access_log

Para finalizar se puede decir que el objetivo de firewall implementado es detener el tráfico no permitido y llevar un registro de la información que fluye a través del mismo. Las tareas de administración del firewall, como son el añadir nuevas

direcciones IP, dominios internet y revisar los archivos de log, son el complemento del firewall.

5.7 EVALUACION DEL FIREWALL IMPLEMENTADO

El firewall implementado con Java es un prototipo básico de firewall proxy, cuyo objetivo principal es controlar el acceso de clientes a internet y permitir que varios usuarios de la red interna tenga una conexión al mundo exterior, a través de un solo punto de comunicación.

Este firewall maneja únicamente solicitudes GET HTTP, pues él se encarga de comunicar a los clientes de la red interna con el exterior y esto se realiza a través del protocolo HTTP. Como se mencionó en la sección referente a los firewall proxy, estos deben implementarse para cada protocolo que se desee manejar, en este caso se escogió HTTP debido a que es el más utilizado en la actualidad.

Otros protocolo tales como Telnet y FTP también pueden implementarse, pero el inconveniente es que no son muy conocidos, pues únicamente los utilizan personas que conocen de informática y su uso no es muy común, para personas que no tienen muchos conocimientos de los servicios que brinda el Internet.

En el futuro si se desea se pueden añadir nuevos protocolos a esta aplicación, lo único que se debe hacer es la implementación del protocolo, para lo cual es necesario realizar un estudio de los puertos a través de los cuales trabaja el protocolo que se desea implementar. Además se debe conocer como es el flujo de información, es decir como el protocolo realiza la solicitud y como interpreta los datos que el servidor contactado envía, el momento que se logre conocer estos datos, la implementación no es muy

complicada, ya que Java es un lenguaje que brinda muchas facilidades y permite el desarrollo de aplicaciones óptimas.

El firewall implementado lleva un registro de las peticiones de conexión de los clientes de la red interna, esto muy bueno debido a que permite llevar un control de las solicitudes de los clientes.

Es conveniente registrar tanto los aciertos como las peticiones que son negadas, ya que de esta manera se puede conocer cuales usuarios que no tiene privilegios tratan de conectarse, pues si un cliente a pesar de saber que no tiene permiso de conexión externa, insiste constantemente en conectarse, es una señal para que el administrador tome decisiones apropiadas sobre este cliente, es necesario conocer con exactitud el porque de su persistencia para obtener conexión con Internet, para así evitarse problemas futuros.

El registro de intentos de acceso a dominios restringidos es muy importante, ya que permite conocer a los usuarios que tratan de utilizar los recursos de la red en tareas que no están acorde con las actividades que desempeñan y que por el contrario están desperdiciando tiempo en actividades no productivas.

En definitiva los registros de los archivos de log son necesarios para la toma de decisiones oportunas y para optimizar el uso de los recursos de la red, ya que permiten saber que hacen los clientes de la red interna y sobre todo ayudan al administrador en su trabajo de detectar posibles amenazas, que si no se detectan a tiempo pueden convertirse en serios problemas de seguridad.

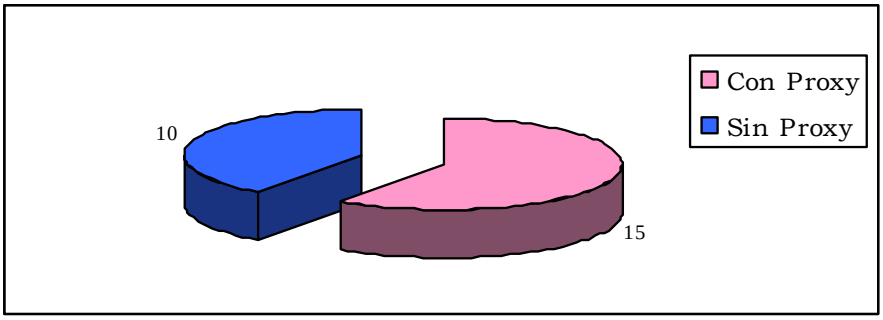
Para finalizar se presenta un cuadro de las características principales que tiene un firewall de software comercial y las características del prototipo desarrollado en Java.

Zone Alarm	**Proxy UTN**
Maneja las dos posturas de firewall, el usuario puede escoger la que desee.	Maneja la postura "lo que no está expresamente prohibido, está permitido".
No permite que varios usuarios de la red local accedan a Internet a través de él.	Varios usuarios de la red local, pueden acceder a Internet a través de él.
Envía las alertas de notificación de bloqueo, el momento que el usuario intenta acceder a Internet.	El usuario es notificado del bloqueo, en el mismo instante en que el intenta acceder a un recurso no permitido.
Las opciones de configuración de la aplicación, deben establecerse en cada equipo en el que se instala el programa.	Las opciones para el funcionamiento de la aplicación, únicamente deben establecerse en el equipo donde se tiene instalado el servidor.
Lleva archivos de log, de las alertas que se generan, cada vez que los usuarios intentan acceder a algún recurso que está bloqueado.	Se generan archivos de log, tanto de las conexiones establecidas con Internet, así como de las conexiones rechazadas.
Los archivos de log, pueden vaciarse cuando el administrador lo desee.	Cuando lo considere necesario, el administrador puede vaciar los archivos de log.
Realiza bloqueos a nivel de aplicaciones que desean conectarse a Internet.	Los bloqueo se realizan en función de la dirección IP que solicita la conexión o del dominio que se está solicitando.
La versión gratuita solamente brinda una parte básica de las funcionalidades que tiene la aplicación.	Actualmente solo brinda el servicio GET HTTP debido a que es un prototipo básico, pero como se dispone del

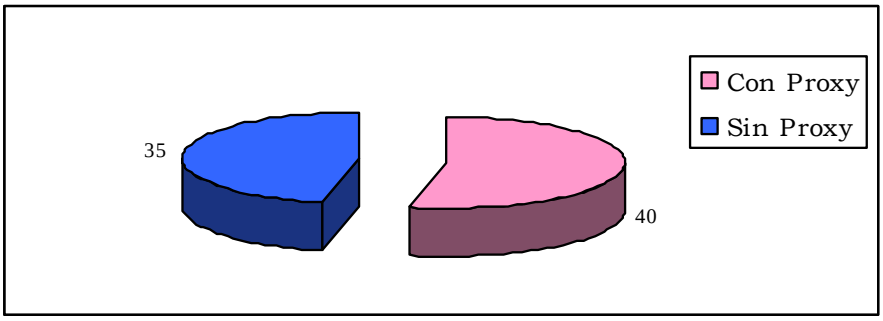
Si se desea obtener una versión completa se debe comprar el programa con las licencias a un costo de: Por una 39,95 USD Por dos 73,90 USD Por diez 315,00 USD Por cincuenta 1.458,00 USD Y la renovación cada año a un valor de: Por una 19,95 USD Por dos 39,90 USD Por diez 99,75 USD Por cincuenta 199,50 USD	código fuente puede irse incrementándole más opciones HTTP tales como: POST, CONNECT, etc. Además se pueden implementar nuevos protocolos como: FTP, Telnet, etc. Todo esto sin ningún costo adicional, únicamente se debería hacer un estudio de cómo es la comunicación de las nuevas opciones a implementarse y se desarrollaría la aplicación en Java, el cual es un lenguaje de programación que constantemente está desarrollándose y cada día brinda nuevas facilidades a los programadores.
El rendimiento en cuanto a la conexión no se ve afectada, ya que no sirve las páginas a los clientes, únicamente se limita a determinar si la aplicación tiene o no privilegios de conexión.. En caso de la aplicación no esta bloqueada, deja pasar el tráfico y es el Servidor Internet de la organización el que se conecta y entrega a los clientes las respuestas de las peticiones que realizó.	Al iniciar el proxy el rendimiento de la conexión se disminuye, debido a que es este servidor el que se conecta con Internet, recibe la respuesta del servidor real y la entrega al cliente que la solicitó. Una muestra de páginas solicitadas es: Con Proxy Sin Proxy Hotmail 15 seg. 10 seg. Ibm 40 seg. 35 seg. Javahispano 55 seg. 35 seg.

A continuación se presenta una representación gráfica, del tiempo que se demora el proxy UTN en entregar los datos que recupera desde Internet a los clientes de la red interna.

Tiempo en segundos al recuperar la página www.hotmail.com



Tiempo en segundos al recuperar la página www.ibm.com



Tiempo en segundos al recuperar la página www.javahispano.com