SEGURIDAD Y ALTA DISPONIBILIDAD
UD4

2º ASIR

04/02/2012

VICEN MORALES

INDICE .- UD 4: Instalación y configuración de cortafuegos

•Cortafuegos:

- Concepto . Utilización de cortafuegos.
- Historia de los cortafuegos.
- Funciones principales de un cortafuegos: Filtrado de paquetes de datos, filtrado por
aplicación, Reglas de filtrado y registros de sucesos de un cortafuegos.
- Listas de control de acceso (ACL).
- Ventajas y Limitaciones de los cortafuegos.
- Políticas de cortafuegos.
- Tipos de cortafuegos.
- Clasificación por ubicación.
- Clasificación por tecnología.
- Arquitectura de cortafuegos.
- Pruebas de funcionamiento. Sondeo.

•Cortafuegos software y hardware:

- Cortafuegos software integrados en los sistemas operativos.
- Cortafuegos software libres y propietarios.
- Distribuciones libres para implementar cortafuegos en máquinas dedicadas.
- Cortafuegos hardware. Gestión Unificada de Amenazas “Firewall UTM” (Unified
Threat Management).

UD 4: Instalación y configuración de cortafuegos

•Cortafuegos:

- Concepto . Utilización de cortafuegos.

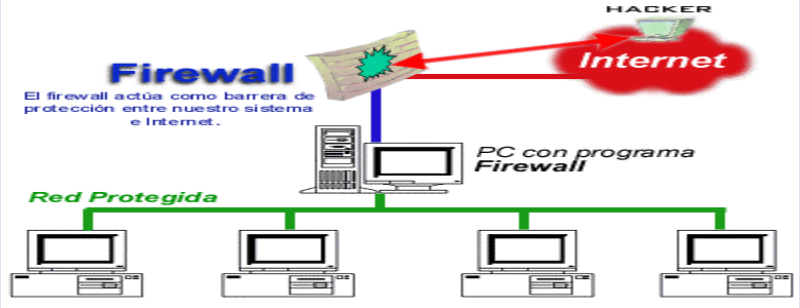
Es un sistema o grupo de sistemas que impone una política de seguridad entre laorganización de red privada y el Internet. Es un mecanismo para restringir acceso entre la Internet y la red corporativa interna. Típicamente se instala un firewall en un punto estratégico donde una red (o redes) se conectan a la Internet.

Un buen Firewall para Internet puede ayudarle a impedir que extraños accedan a su PC desde Internet. Los Firewalls pueden ser de dos tipos, de software o de hardware, y proporcionan una frontera de protección que ayuda a mantener fuera a los invasores no deseados de Internet.

La existencia de un firewall en un sitio Internet reduce considerablemente las probabilidades de ataques externos a los sistemas corporativos y redes internas, además puede servir para evitar que los propios usuarios internos comprometan la seguridad de la red al enviar información peligrosa (como passwords no encriptados o datos sensitivos para la organización) hacia el mundo externo.

Si el Firewall "observa" alguna actividad sospechosa: que alguien de fuera esté intentando acceder a nuestro Pc o que algún programa espía trate de enviar información sin consentimiento, el Firewall nos advertirá con una alarma en el sistema.

Para entender el funcionamiento de este sistema, debes saber que el ordenador dispone de varias puertas de salida y entrada cuando se conecta a Internet. Éstas se llaman puertos y cada servicio que utilizas se sirve de un puerto diferente: Los navegadores de internet necesitan el puerto 80, los programas FTP el 21, etc... En general tenemos todos los puertos abiertos.



- Historia de los cortafuegos.

El término "firewall / fireblock" significaba originalmente una pared para confinar un incendio o riesgo potencial de incendio en un edificio. Más adelante se usa para referirse a las estructuras similares, como la hoja de metal que separa el compartimiento del motor de un vehículo o una aeronave de la cabina. La tecnología de los cortafuegos surgió a finales de 1980, cuando Internet era una tecnología bastante nueva en cuanto a su uso global y la conectividad. Los predecesores de los cortafuegos para la seguridad de la red fueron los routers utilizados a finales de 1980, que mantenían a las redes separadas unas de otras. La visión de Internet como una comunidad relativamente pequeña de usuarios con máquinas compatibles, que valoraba la predisposición para el intercambio y la colaboración, terminó con una serie de importantes violaciones de seguridad de Internet que se produjo a finales de los 80:

Clifford Stoll, que descubrió la forma de manipular el sistema de espionaje alemán.

Bill Cheswick, cuando en 1992 instaló una cárcel simple electrónica para observar a un atacante.

En 1988, un empleado del Centro de Investigación Ames de la NASA, en California, envió una nota por correo electrónico a sus colegas que decía:

"Estamos bajo el ataque de un virus de Internet! Ha llegado a Berkeley, UC San Diego, Lawrence Livermore, Stanford y la NASA Ames."

El Gusano Morris, que se extendió a través de múltiples vulnerabilidades en las máquinas de la época. Aunque no era malicioso, el gusano Morris fue el primer ataque a gran escala sobre la seguridad en Internet; la red no esperaba ni estaba preparada para hacer frente a su ataque.

Primera generación – cortafuegos de red: filtrado de paquetes

El primer documento publicado para la tecnología firewall data de 1988, cuando el equipo de ingenieros Digital Equipment Corporation (DEC) desarrolló los sistemas de filtro conocidos como cortafuegos de filtrado de paquetes. Este sistema, bastante básico, fue la primera generación de lo que se convertiría en una característica más técnica y evolucionada de la seguridad de Internet. En AT&T Bell, Bill Cheswick y Steve Bellovin, continuaban sus investigaciones en el filtrado de paquetes y desarrollaron un modelo de trabajo para su propia empresa, con base en su arquitectura original de la primera generación.

El filtrado de paquetes actúa mediante la inspección de los paquetes (que representan la unidad básica de transferencia de datos entre ordenadores en Internet). Si un paquete coincide con el conjunto de reglas del filtro, el paquete se reducirá (descarte silencioso) o será rechazado (desprendiéndose de él y enviando una respuesta de error al emisor). Este tipo de filtrado de paquetes no presta atención a si el paquete es parte de una secuencia existente de tráfico. En su lugar, se filtra cada paquete basándose únicamente en la información contenida en el paquete en sí (por lo general utiliza una combinación del emisor del paquete y la dirección de destino, su protocolo, y, en el tráfico TCP y UDP, el número de puerto). Los protocolos TCP y UDP comprenden la mayor parte de comunicación a través de Internet, utilizando por convención puertos bien conocidos para determinados tipos de tráfico, por lo que un filtro de paquetes puede distinguir entre ambos tipos de tráfico (ya sean navegación web, impresión remota, envío y recepción de correo electrónico, transferencia de archivos…); a menos que las máquinas a cada lado del filtro de paquetes son a la vez utilizando los mismos puertos no estándar.

El filtrado de paquetes llevado a cabo por un cortafuegos actúa en las tres primeras capas del modelo de referencia OSI, lo que significa que todo el trabajo lo realiza entre la red y las capas físicas. Cuando el emisor origina un paquete y es filtrado por el cortafuegos, éste último comprueba las reglas de filtrado de paquetes que lleva configuradas, aceptando o rechazando el paquete en consecuencia. Cuando el paquete pasa a través de cortafuegos, éste filtra el paquete mediante un protocolo y un número de puerto base (GSS). Por ejemplo, si existe una norma en el cortafuegos para bloquear el acceso telnet, bloqueará el protocolo IP para el número de puerto 23.

Segunda generación – cortafuegos de estado

Durante 1989 y 1990, tres colegas de los laboratorios AT&T Bell, Dave Presetto, Janardan Sharma, y Nigam Kshitij, desarrollaron la tercera generación de servidores de seguridad. Esta tercera generación cortafuegos tiene en cuenta además la colocación de cada paquete individual dentro de una serie de paquetes. Esta tecnología se conoce generalmente como la inspección de estado de paquetes, ya que mantiene registros de todas las conexiones que pasan por los cortafuegos, siendo capaz de determinar si un paquete indica el inicio de una nueva conexión, es parte de una conexión existente, o es un paquete erróneo. Este tipo de cortafuegos pueden ayudar a prevenir ataques contra conexiones en curso o ciertos ataques de denegación de servicio.

Tercera generación - cortafuegos de aplicación

Son aquellos que actúan sobre la capa de aplicación del modelo OSI. La clave de un cortafuegos de aplicación es que puede entender ciertas aplicaciones y protocolos (por

ejemplo: protocolo de transferencia de ficheros, DNS o navegación web), y permite detectar si un protocolo no deseado se coló a través de un puerto no estándar o si se está abusando de un protocolo de forma perjudicial.

Un cortafuegos de aplicación es mucho más seguro y fiable cuando se compara con un cortafuegos de filtrado de paquetes, ya que repercute en las siete capas del modelo de referencia OSI. En esencia es similar a un cortafuegos de filtrado de paquetes, con la diferencia de que también podemos filtrar el contenido del paquete. El mejor ejemplo de cortafuegos de aplicación es ISA (Internet Security and Acceleration).

Un cortafuegos de aplicación puede filtrar protocolos de capas superiores tales como FTP, TELNET, DNS, DHCP, HTTP, TCP, UDP y TFTP (GSS). Por ejemplo, si una organización quiere bloquear toda la información relacionada con una palabra en concreto, puede habilitarse el filtrado de contenido para bloquear esa palabra en particular. No obstante, los cortafuegos de aplicación resultan más lentos que los de estado.

Acontecimientos posteriores

En 1992, Bob Braden y DeSchon Annette, de la Universidad del Sur de California (USC), dan forma al concepto de cortafuegos. Su producto, conocido como "Visas", fue el primer sistema con una interfaz gráfica con colores e iconos, fácilmente implementable y compatible con sistemas operativos como Windows de Microsoft o MacOS de Apple. En 1994, una compañía israelí llamada Check Point Software Technologies lo patentó como software denominándolo FireWall-1.

La funcionalidad existente de inspección profunda de paquetes en los actuales cortafuegos puede ser compartida por los sistemas de prevención de intrusiones (IPS).

Actualmente, el Grupo de Trabajo de Comunicación Middlebox de la Internet Engineering Task Force (IETF) está trabajando en la estandarización de protocolos para la gestión de cortafuegos.

Otro de los ejes de desarrollo consiste en integrar la identidad de los usuarios dentro del conjunto de reglas del cortafuegos. Algunos cortafuegos proporcionan características tales como unir a las identidades de usuario con las direcciones IP o MAC. Otros, como el cortafuegos NuFW, proporcionan características de identificación real solicitando la firma del usuario para cada conexión.

- Funciones principales de un cortafuegos: Filtrado de paquetes de datos, filtrado por aplicación, Reglas de filtrado y registros de sucesos de un cortafuegos.

Una de las funciones más importantes de un firewall es el filtrado o control de acceso de toda la información que sea recibida en los distintos puntos de acceso a la red interna o a los sistemas finales, que son administrados por aquél.

El filtrado de datos permite controlar la transferencia segura de datos basado principalmente en: la dirección de donde provienen los datos, la dirección de destino de los datos y los protocolos de transporte y aplicación utilizados.

Esta función puede ser implementada en diferentes niveles de la arquitectura de red, con lo cual se logran diferentes niveles de granularidad, es decir, qué tan minucioso es el control de seguridad efectuado. Sobre la base del nivel donde se efectúe el filtrado, la función se implementará en diferentes dispositivos[1]. Los niveles mencionados son tres[2]: *filtrado de paquetes*, *control de acceso de conexiones* y *filtrado de datos de aplicación*.

Filtrado de paquetes (a nivel de red)

Los filtros de paquetes operan al más bajo nivel de abstracción en el cual, los datos son transmitidos en paquetes y analizados como tales. En la familia de protocolos TCP/IP, los filtros son aplicados al nivel de transporte (TCP, UDP) y al nivel de red (IP) (ver Figura 6).

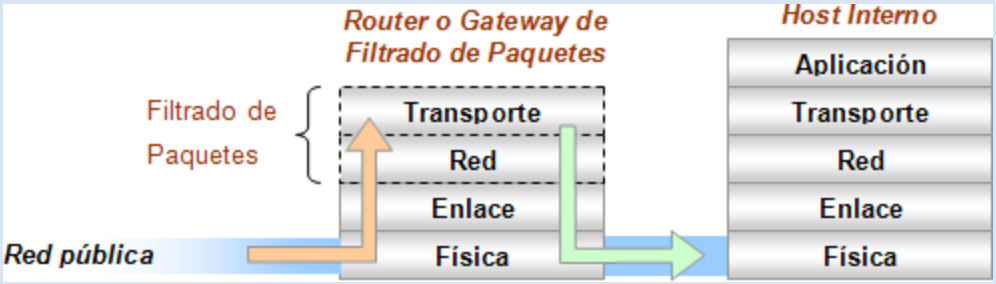


Figura 6 - Filtrado de paquetes en un Router o Gateway

Este mecanismo es implementado por lo general en los sistemas intermedios (*gateways* o *routers*) que conectan la red interna con la red pública. Cada paquete que ingresa a la red es interceptado y analizado por la función de filtrado, implementada por un filtro de paquetes en estos dispositivos intermedios. Suelen ser llamados *Router de Filtrado de Paquetes* o *Gateways de Filtrado de Paquetes*.

El filtro rechaza o reenvía los paquetes al destinatario original, según reglas especificadas en Listas de Control de Acceso (ACL), que son almacenadas en el router o gateway, basadas en los datos de los encabezados de los paquetes TCP e IP. Básicamente los datos analizados son las direcciones IP y puertos TCP de origen y destino de los paquetes.

Un filtro de paquetes no mantiene información de contexto para los paquetes que sean parte de una conexión; todos los paquetes son tratados de forma independiente, sin ser relacionados con ningún otro.

La principal *ventaja* del filtrado de paquetes es que permite proveer, en un único sitio o punto, una protección particular para la red entera. Además es transparente a los usuarios de la red ya que no requiere configuración alguna en los sistemas que interconecta ni realizar tareas especiales de transmisión u otro tipo, lo que ofrece una estructura flexible en el sentido de que puede ser modificada o re-estructurada sin necesidad de modificar el resto de la red.

Aunque existen algunas *desventajas* como posibles problemas de desempeño que pueden ser clasificados en tres categorías [Peri]: número de campos examinados, posición del campo en el paquete, demora del proceso. Esto afecta el espacio necesario para almacenar las reglas, el número de comparaciones y la complejidad del algoritmo.

El *núcleo* de un filtro de paquetes consiste de un *lenguaje de descripción* que permite expresar reglas de políticas de seguridad orientadas a paquetes. Las reglas definidas hacen referencia a entidades, es decir que identifica dispositivos o sistemas por medio de sus direcciones IP. La sintáxis de estos lenguaje no está estandarizada por lo que diferentes productos permiten expresar las reglas de diferentes formas.

Por ejemplo la siguiente regla (*Figura 7*) retransmitirá a la red interna todo paquete que provenga del socket *170.210.122.45:20*, destinado al socket *130.15.214.23:1032*.

Nombre	Dirección de origen y puerto	Dirección de destino y puerto	Acción
R1	170.210.122.45:20	130.15.214.23:1032	Permitir

Figura 7 - Ejemplo de regla de seguridad

Para ser efectivo, un filtro debe permitir expresar reglas utilizando comodines y rangos de valores para referirse a más de un host o dispositivo. Por ejemplo “permitir los paquetes destinados al host *130.15.214.23* pero sólo entre los puertos *1000* y *1050* (*130.15.214:1000..1050*)”; y “negar todos aquellos paquetes provenientes de la red *170.210.122.0* (*170.210.122.*:**)”.

También debe considerarse el espacio necesario para almacenar las reglas ya que existen múltiples caminos que pueden tomar los paquetes hasta llegar al dispositivo filtrador. De aquí surge, también, la necesidad de algoritmos de búsqueda de reglas. Todos estos aspectos afectan el desempeño del filtro afectando también el desempeño de la red.

Implementación

La función de filtrado de paquetes puede implementarse en varios sitios de la red interna. La forma más directa y simple es utilizar un *router* que la soporte.

Un router tendrá dos interfaces, una que conecte a la red externa y la otra a la red interna. Los filtros pueden aplicarse en una de las dos interfaces, o en ambas. Además puede aplicarse al tráfico de entrada como al de salida, o a ambos. Estas características varían con los diferentes routers. Tales consideraciones reflejan diferentes políticas más o menos flexibles, con más o menos puntos de control. Una buena política a respetar es que si un paquete ha de ser rechazado, que sea cuanto antes. Otra posible opción es efectuar el filtrado independientemente de la interfaz de red.

Una alternativa es utilizar *filtros basados en hosts*, tales como *screend* de Digital Equipment Corporation disponible para algunos sistemas operativos; *ipfilterd* de SGI Systems y Karlbridge. Otra herramienta es IPTables/NetFilter, la cuarta generación de filtros de paquetes para Linux. La primer generación fue ipfw, creada para BSD UNIX y portada para Linux. Uno de los objetivos de NetFilter es proveer una infraestructura de filtrado de paquetes dedicada que los usuarios y desarrolladores pudieran instalar como agregado en el kernel de Linux

Filtrado de paquetes con NAT

Es posible efectuar el filtrado de paquetes junto con la Traducción de Direcciones de Red sin causar dificultades a ninguna de las dos funciones. La función de filtrado de paquetes se diseña ignorando por completo cualquier traducción de direcciones que se lleve a cabo ya que ésta última se realiza entre la entrada / salida de datos en el borde de la red y el filtrado de paquetes. Las direcciones captadas por el filtro serán las direcciones origen y destino reales.

Control de Acceso de Conexiones

Este mecanismo controla y retransmite conexiones TCP manteniendo registro del estado de todos los paquetes que agrupan tal conexión, de forma que solo aquellos hosts externos confiables puedan establecer conexiones con aquellos dispositivos habilitados a ofrecer un servicio a tales usuarios. De la misma forma es posible restringir las conexiones originadas en la red interna con destino a ciertos sitios de la red externa. Esta función es realizada por un *proceso proxy* instalado en un gateway

que interconecta la red interna con la red pública. Estos dispositivos son llamados *gateways a nivel de circuitos*. (*ver Figura 8*)

Una alternativa a mantener el contexto de cada paquete es utilizar tablas dinámicas basadas en las banderas SYN/ACK del encabezado de los paquetes TCP. En esta forma, la tabla de reglas se genera a medida que un host interno solicita una conexión con un sitio externo por lo que el gateway asume la política de reenviar solo aquellos paquetes entrantes que pertenezcan a conexiones iniciadas desde el interior y rechazar aquellas iniciadas en el exterior (similar a la estrategia lograda con NAT dinámico).

Mediante el uso del proxy, los sistemas internos no podrán establecer conexiones directas con el exterior sino por intermedio del proxy; quien solicite una conexión, se conectará a un puerto TCP del gateway, luego el proxy determinará si la conexión es permitida o no, basado en un conjunto de reglas de acceso que utilizan información del encabezado del paquete TCP, luego (si la conexión fue aceptada) el gateway crea una conexión al dispositivo interno final. En este caso, el gateway retransmitirá todos los paquetes involucrados en la conexión.

Estos gateways pueden implementar algunos mecanismos de control de acceso tales como autenticación e intercambio de mensajes de protocolo entre cliente y proxy para establecer ciertos parámetros del circuito.

El control de acceso de conexiones no es del todo transparente ya que los usuarios deben ser configurados para dirigir todas sus solicitudes al dispositivo que implemente esta función.

La ventaja del mecanismo de filtrado a nivel de circuitos es que provee servicios para un amplio rango de protocolos aunque requiere software especial en el cliente, lo que lleva al problema de que la seguridad basada en hosts no es escalable (con una arquitectura de seguridad perimetral). A medida que crece la red, la administración de la seguridad de los clientes se hace más compleja por lo que demora más tiempo llevarla a cabo y propensa al error; esto si no se efectúa un control central e implementado de forma distribuida.

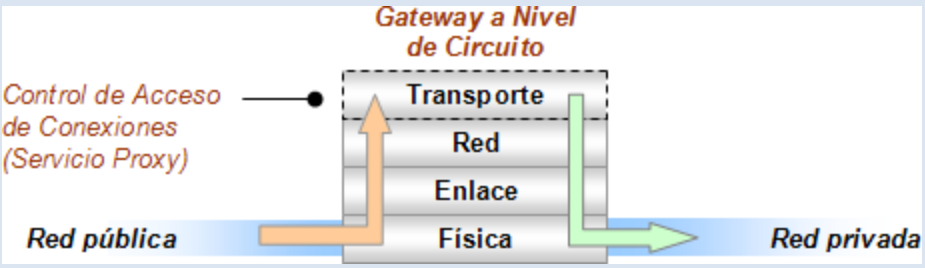


Figura 8 - Control de Acceso de Conexiones en un Gateway

Filtrado de Datos de Aplicación

Este mecanismo interpreta los datos encapsulados en los paquetes correspondientes a protocolos de aplicación particulares para determinar si deben o no deben ser procesados por la aplicación correspondiente, ya que pueden contener datos que afecten el buen funcionamiento de las mismas. La función de seguridad ofrecida por este mecanismo es mucho más segura que las anteriores (*ver Figura 9*)

Son implementados por servicios proxies instalados en gateways, llamados *gateways a nivel de aplicación*. Proveen una barrera de seguridad entre los usuarios internos y la red pública. Los usuarios de la red interna se conectan al filtro de datos de aplicación, quien funciona como intermediario entre diferentes servicios de la red externa y el usuario interno.

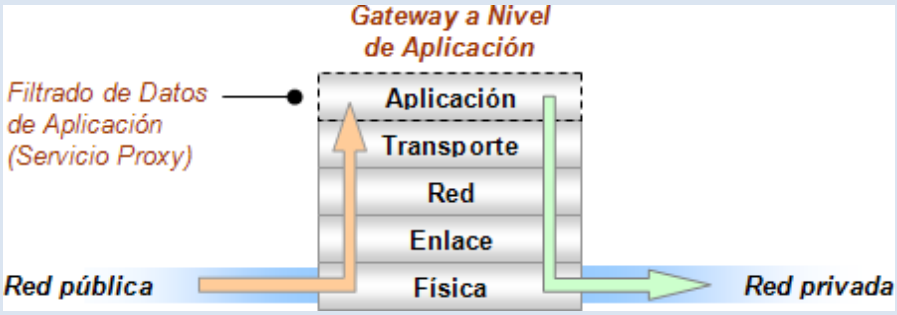


Figura 9 - Filtrado de Datos de Aplicación en un Gateway

Son implementaciones de propósito especial que intentan ofrecer servicios de seguridad a las aplicaciones que procesen tales datos. Son específicos de la aplicación, es decir que se necesita un proceso proxy para cada aplicación. Esto presenta una desventaja de implementación. Aunque solo algunos programas o protocolos de aplicación necesitan ser analizados (por Ej. FTP y protocolos de correo electrónico, ICMP) ya que otros no presentan peligros de seguridad. El correo electrónico puede ser dirigido a través de estos dispositivos, sin importar que tecnología se utilice en el resto del firewall. También hay que tener en cuenta que el tipo de filtrado usado depende de las necesidades locales. Un sitio con muchos usuarios de PC debería analizar los archivos que reciba por posibles virus.

Además presentan otra ventaja, que en algunos ambientes es bastante crítica: el registro de todo el tráfico de entrada y salida es simple implementar.

- Listas de control de acceso (ACL).

Una lista de control de acceso o ACL (del inglés, access control list) es un concepto de seguridad informática usado para fomentar la separación de privilegios. Es una forma de determinar los permisos de acceso apropiados a un determinado objeto, dependiendo de ciertos aspectos del proceso que hace el pedido.

Las ACL permiten controlar el flujo del tráfico en equipos de redes, tales como enrutadores y conmutadores. Su principal objetivo es filtrar tráfico, permitiendo o denegando el tráfico de red de acuerdo a alguna condición. Sin embargo, también tienen usos adicionales, como por ejemplo, distinguir "tráfico interesante" (tráfico suficientemente importante como para activar o mantener una conexión) en RDSI.

Paso 1	Definir la ACL con el siguiente comando: `Router(config)#access-list access-list-number {permit | deny} {test-conditions}` Una sentencia global identifica la ACL. Específicamente, el intervalo 1-99 se reserva para IP estándar. Este número se refiere al tipo de ACL. En la versión 11.2 o posterior de Cisco IOS, las ACL también pueden usar un nombre ACL, como educación_grupo, en lugar de un número El término permit o deny (permitir o denegar) de la sentencia ACL global indica cuántos paquetes que cumplan con las condiciones de prueba maneja el software Cisco IOS. Permit generalmente significa que el paquete puede usar una o más interfaces que se especifican posteriormente. El (Los) último(s) término(s) especifican las condiciones de prueba que utiliza la sentencia ACL.
	A continuación, es necesario aplicar las ACL en una interfaz mediante el comando access-group, como se muestra en el ejemplo.

En redes informáticas

En redes informáticas, ACL se refiere a una lista de reglas que detallan puertos de servicio o nombres de dominios (de redes) que están disponibles en un terminal u otro dispositivo de capa de red, cada uno de ellos con una lista de terminales y/o redes que tienen permiso para usar el servicio. Tanto servidores individuales como enrutadores pueden tener ACL de redes. Las listas de control de acceso pueden configurarse

generalmente para controlar tráfico entrante y saliente y en este contexto son similares a un cortafuegos.

Existen dos tipos de listas de control de acceso:

Listas estándar, donde solo tenemos que especificar una dirección de origen;

Listas extendidas, en cuya síntaxis aparece el protocolo y una dirección de origen y de destino.

Las redes empresariales necesitan seguridad para asegurarse de que solo los usuarios autorizados accedan a los recursos de red.

Las herramientas de filtrado, como las listas de control acceso, son un componente importante de la seguridad de red empresarial.

Las ACL permiten y rechazan tipos específicos de informacion entrante y trafico saliente.

Los ingenieros y los técnicos de red planifican, configuran y verifican las ACL en los routers y otros dispositivos de red.

En este capítulo describiremos los siguientes puntos:

Describir el filtrado de tráfico.

Explicaremos como las listas de control de acceso (ACL) pueden filtrar el tráfico en las interfaces de Reuter.

Para los que tienen dudas sobre el filtrado de trafico aqui les doy una idea:

Filtrado de Tráfico:

La seguridad dentro de una red empresarial es sumamente importante. Es esencial impedir el acceso de usuarios no autorizados y proteger la red de diversos ataques, como los ataques DoS. Los usuarios no autorizados pueden modificar, destruir o robar datos confidenciales de los servidores. Los ataques DoS impiden el acceso de los usuarios válidos. Estas dos situaciones hacen perder tiempo y dinero a las empresas.

Mediante el filtrado de tráfico, los administradores controlan el tráfico de varios

segmentos de la red. El filtrado es el proceso de analizar los contenidos de un paquete para determinar si debe ser permitido o bloqueado.

El filtrado de paquetes puede ser simple o complejo, denegando o permitiendo el tráfico basado en:

Dirección IP de origen

Dirección IP de destino

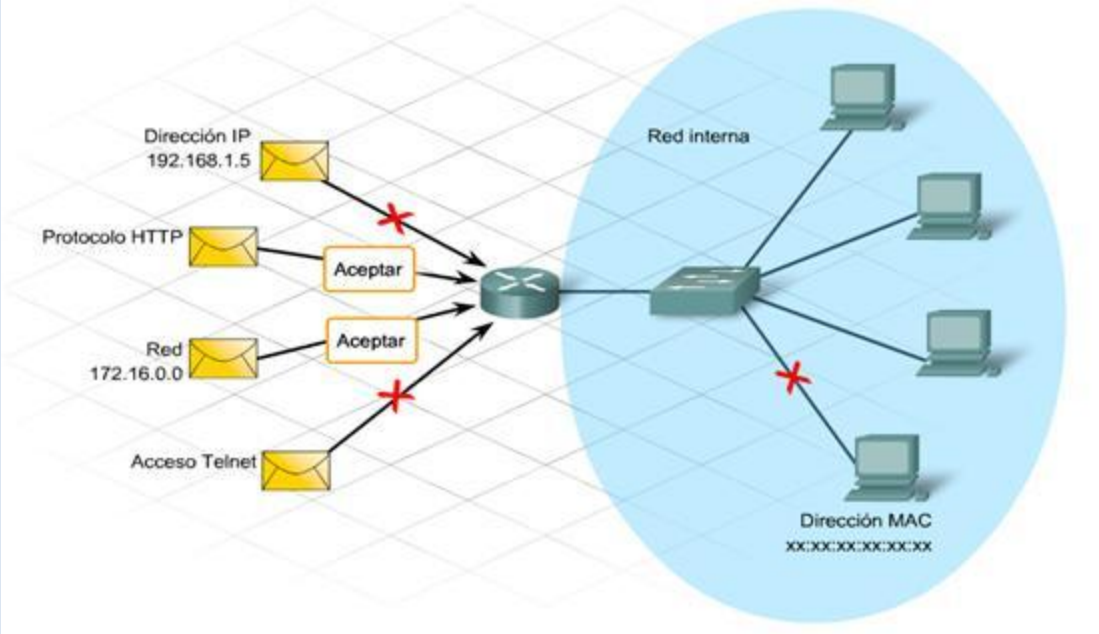
Direcciones MAC

Protocolos

Tipo de aplicación

El filtrado de paquetes se puede comparar con el filtrado de correo basura. Muchas aplicaciones de correo electrónico permiten a los usuarios ajustar la configuración para que los correos electrónicos enviados desde una dirección de origen particular se eliminen automáticamente. El filtrado de paquetes se puede utilizar de la misma forma mediante la configuración de un router para identificar el tráfico no deseado.

El filtrado de tráfico mejora el rendimiento de la red. Al denegar el tráfico no deseado o restringido cerca de su origen, éste no viajará a través de la red ni consumirá recursos valiosos.



Los dispositivos más utilizados para proporcionar filtrado de tráfico son:

Firewalls incorporados en routers integrados

Aplicaciones de seguridad dedicadas

Servidores

Algunos dispositivos sólo filtran el tráfico que se origina en la red interna. Los dispositivos de seguridad más sofisticados reconocen y filtran los tipos de ataques conocidos de fuentes externas.
Los routers empresariales reconocen el tráfico perjudicial e impiden que ingrese y dañe la red. Casi todos los routers filtran tráfico de acuerdo con las direcciones IP de origen y de destino de los paquetes. También filtran aplicaciones específicas y protocolos tales como IP, TCP, HTTP, FTP y Telnet.

Listas de Control de Acceso:

Uno de los métodos más comunes de filtrado de tráfico es el uso de listas de control de acceso (ACL). Las ACL pueden utilizarse para administrar y filtrar el tráfico que ingresa a una red, así como también el tráfico que sale de ella.

El tamaño de una ACL varía desde una sentencia que permite o deniega el tráfico de un origen, hasta cientos de sentencias que permiten o deniegan paquetes de varios orígenes. El uso principal de las ACL es identificar los tipos de paquetes que se deben aceptar o denegar.
Las ACL identifican el tráfico para varios usos, por ejemplo:

Especificar hosts internos para NAT

Identificar o clasificar el tráfico para funciones avanzadas tales como QoS y colas

Restringir el contenido de las actualizaciones de enrutamiento

Limitar el resultado de la depuración

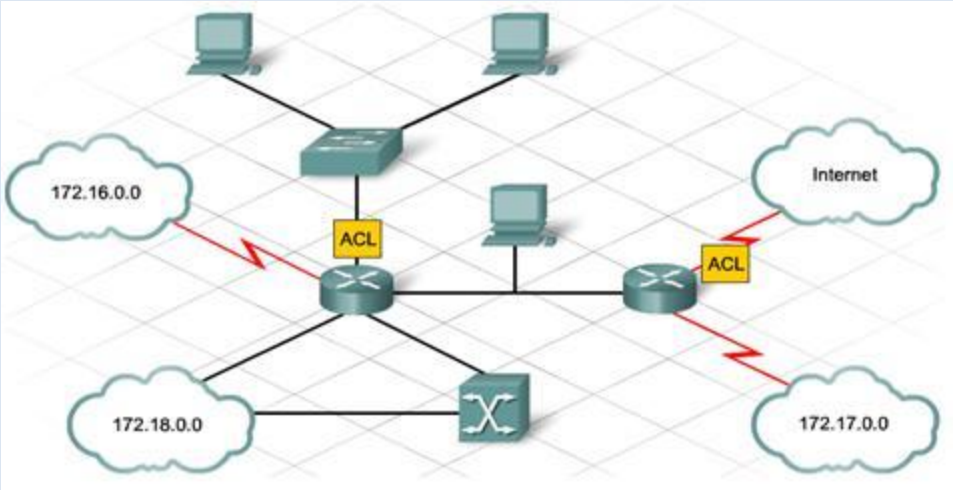
Controlar el acceso de terminales virtuales a los routers

El uso de las ACL puede provocar los siguientes problemas potenciales:

La carga adicional sobre el router para verificar todos los paquetes se traduce en menos tiempo para el envío de paquetes.

Las ACL con diseños defectuosos colocan una carga aún mayor sobre el router y podrían interrumpir el uso de la red.

Las ACL colocadas de forma incorrecta bloquean el tráfico que debe ser permitido y permiten el tráfico que debe ser bloqueado.



Tipos y Uso de ACL:

Al crear listas de control de acceso, el administrador de red tiene varias opciones. La complejidad de las pautas de diseño determina el tipo de ACL necesaria.

Hay tres clases de ACL:

1.- ACL estándar:
La ACL estándar es la más simple de las tres clases. Al crear una ACL IP estándar, las ACL filtran según la dirección IP de origen de un paquete. Las ACL estándar permiten o deniegan el acceso de acuerdo con la totalidad del protocolo, como IP. De esta manera, si un dispositivo host es denegado por una ACL estándar, se deniegan todos los servicios provenientes de ese host. Este tipo de ACL sirve para permitir el acceso de todos los servicios de un usuario específico, o LAN, a través de un router y, a la vez, denegar el acceso de otras direcciones IP. Las ACL estándar están identificadas por el número que se les ha asignado. Para las listas de acceso que permiten o deniegan el tráfico IP, el número de identificación puede variar entre 1 y 99 y entre 1300 y 1999.

2.- ACL extendidas:
Las ACL extendidas filtran no sólo según la dirección IP de origen, sino también según la dirección IP de destino, el protocolo y los números de puertos. Las ACL extendidas se utilizan más que las ACL estándar porque son más específicas y ofrecen un mayor control. El rango de números de las ACL extendidas va de 100 a 199 y de 2000 a 2699.

3.- ACL nombradas:
Las ACL nombradas (NACL, Named ACL) son ACL estándar o extendidas a las que se hace referencia mediante un nombre descriptivo en lugar de un número. Cuando se configuran ACL nombradas, el IOS del router utiliza un modo de subcomando de NACL.

Tipos de Listas de acceso IOS

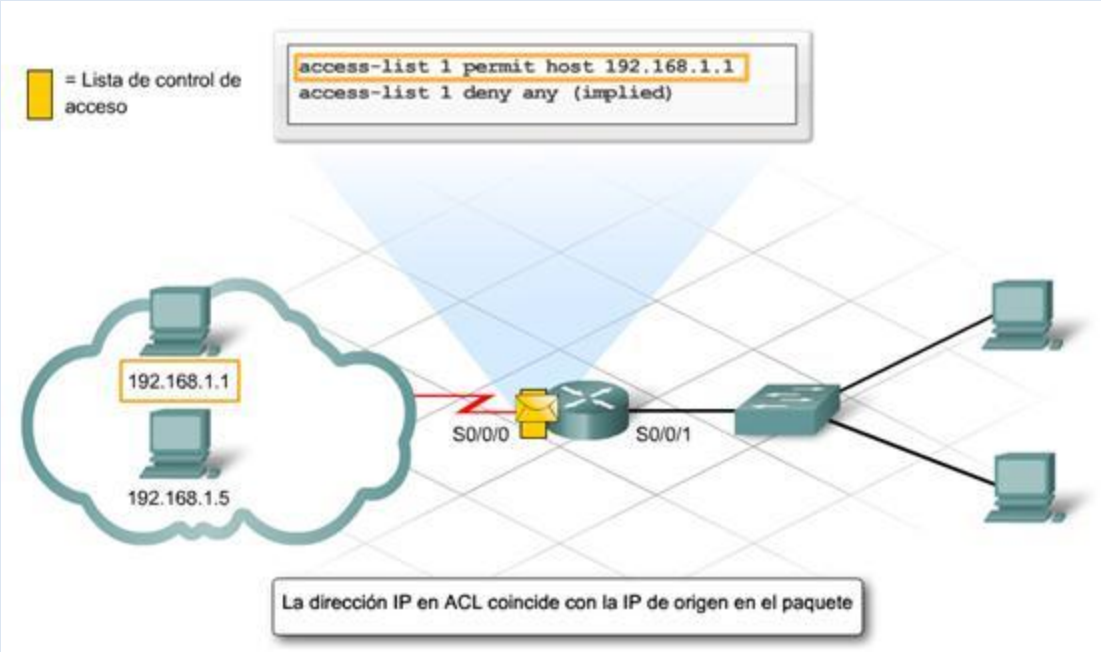
Tipo de ACL	Ejemplo de comando/sentencia ACL	Objetivo de sentencia
Estándar	`Router(config)#access-list 1 permit host 172.16.2.88`	• Permite una dirección IP específica
Extendida	`Router(config)#access-list 100 deny tcp 172.16.2.0 0.0.0.255 any eq telnet`	• Deniega el acceso desde la subred 172.16.2.0/24 a cualquier otro host, si están intentando utilizar telnet
Nombrado	`Router(config)#ip access-list standard permit-ip` `Router(config-ext-nacl)#permit host 192.168.5.47`	• Crea una lista de acceso estándar que se denomina permit-ip • Permite el acceso desde la dirección IP 192.168.5.47 • El primer comando configura al router en el modo de subcomando NACL

Procesamiento de ACL

Las listas de control de acceso consisten de una o más sentencias. Cada sentencia puede permitir o denegar el tráfico según parámetros específicos. El tráfico se compara con cada sentencia de la ACL en forma secuencial hasta encontrar una coincidencia o hasta que no haya más sentencias.

La última sentencia de una ACL es siempre una denegación implícita. Esta sentencia se inserta automáticamente al final de cada ACL, aunque no esté presente físicamente. La denegación implícita bloquea todo el tráfico. Esta función impide la entrada accidental de tráfico no deseado.

Después de crear una lista de control de acceso, aplíquela a una interfaz para que entre en vigencia. La ACL se aplica al tráfico entrante o saliente a través de la interfaz. Si un paquete coincide con una sentencia de permiso, se le permite entrar o salir del router. Si coincide con una sentencia de denegación, no puede seguir avanzando. Una ACL que no tiene al menos una sentencia de permiso bloquea todo el tráfico. Esto se debe a que al final de todas las ACL hay una denegación implícita. Por lo tanto, una ACL rechazará todo el tráfico que no está específicamente permitido.



El administrador aplica una ACL entrante o saliente a una interfaz de router. La dirección se considera entrante o saliente desde la perspectiva del router. El tráfico que ingresa a un interfaz será entrante y el tráfico que sale de ella será saliente.

Cuando un paquete llega a una interfaz, el router controla los siguientes parámetros:

¿Hay una ACL asociada con la interfaz?

¿La ACL es entrante o saliente?

¿El tráfico coincide con los criterios para permitir o para denegar?

Una ACL aplicada en dirección saliente a una interfaz no tiene efectos sobre el tráfico entrante en esa misma interfaz.

Cada interfaz de un router puede tener una ACL por dirección para cada protocolo de red. Respecto del protocolo IP, una interfaz puede tener una ACL entrante y una ACL saliente al mismo tiempo.

Las ACL aplicadas a una interfaz agregan latencia al tráfico. Incluso una ACL larga puede afectar el rendimiento del router.

- Ventajas y Limitaciones de los cortafuegos.

Ventajas de un cortafuegos

Bloquea el acceso a personas y/o aplicaciones no autorizadas a redes privadas.

Limitaciones de un cortafuegos

Las limitaciones se desprenden de la misma definición del cortafuego: filtro de tráfico. Cualquier tipo de ataque informático que use tráfico aceptado por el cortafuegos (por usar puertos TCP abiertos expresamente, por ejemplo) o que sencillamente no use la red, seguirá constituyendo una amenaza. La siguiente lista muestra algunos de estos riesgos:

Un cortafuegos no puede proteger contra aquellos ataques cuyo tráfico no pase a través de él.

El cortafuegos no puede proteger de las amenazas a las que está sometido por ataques internos o usuarios negligentes. El cortafuegos no puede prohibir a espías corporativos copiar datos sensibles en medios físicos de almacenamiento (discos, memorias, etc.) y sustraerlas del edificio.

El cortafuegos no puede proteger contra los ataques de ingeniería social.

El cortafuegos no puede proteger contra los ataques posibles a la red interna por virus informáticos a través de archivos y software. La solución real está en que la organización debe ser consciente en instalar software antivirus en cada máquina para protegerse de los virus que llegan por cualquier medio de almacenamiento u otra fuente.

El cortafuegos no protege de los fallos de seguridad de los servicios y protocolos cuyo tráfico esté permitido. Hay que configurar correctamente y cuidar la seguridad de los servicios que se publiquen en Internet.

- Políticas de cortafuegos.

Hay dos políticas básicas en la configuración de un cortafuegos que cambian radicalmente la filosofía fundamental de la seguridad en la organización:

Política restrictiva: Se deniega todo el tráfico excepto el que está explícitamente permitido. El cortafuegos obstruye todo el tráfico y hay que habilitar expresamente el tráfico de los servicios que se necesiten. Esta aproximación es la que suelen utilizar la empresas y organismos gubernamentales.

Política permisiva: Se permite todo el tráfico excepto el que esté explícitamente denegado. Cada servicio potencialmente peligroso necesitará ser aislado básicamente caso por caso, mientras que el resto del tráfico no será filtrado. Esta aproximación la suelen utilizar universidades, centros de investigación y servicios públicos de acceso a internet.

La política restrictiva es la más segura, ya que es más difícil permitir por error tráfico potencialmente peligroso, mientras que en la política permisiva es posible que no se haya contemplado algún caso de tráfico peligroso y sea permitido por omisión.

- Tipos de cortafuegos.

Nivel de aplicación de pasarela

Aplica mecanismos de seguridad para aplicaciones específicas, tales como servidores FTP y Telnet. Esto es muy eficaz, pero puede imponer una degradación del rendimiento.

Circuito a nivel de pasarela

Aplica mecanismos de seguridad cuando una conexión TCP o UDP es establecida. Una vez que la conexión se ha hecho, los paquetes pueden fluir entre los anfitriones sin más control. Permite el establecimiento de una sesión que se origine desde una zona de mayor seguridad hacia una zona de menor seguridad.

Cortafuegos de capa de red o de filtrado de paquetes

Funciona a nivel de red (capa 3 del modelo OSI, capa 2 del stack de protocolos TCP/IP) como filtro de paquetes IP. A este nivel se pueden realizar filtros según los distintos campos de los paquetes IP: dirección IP origen, dirección IP destino. A menudo en este tipo de cortafuegos se permiten filtrados según campos de nivel de transporte (capa 3 TCP/IP, capa 4 Modelo OSI), como el puerto origen y destino, o a nivel de enlace de datos (no existe en TCP/IP, capa 2 Modelo OSI) como la dirección MAC.

Producto	Puntos
Comodo Personal Firewall 2.3.6.81	9350
Jetico Personal Firewall 2.0.0.16 beta	9125
ZoneAlarm PRO 6.5.737.000	8250
Trend Micro PC-cillin Internet Security 2007 15.00.1329	7500
Outpost Firewall PRO 4.0 (971.584.079)	6675
Lavasoft Personal Firewall 1.0.543.5722 (433)	6500
Kaspersky Internet Security 6.0.0.303	6350
BlackICE PC Protection 3.6.cpv	5750
Sunbelt Kerio Personal Firewall 4.3.268	4825
Look 'n' Stop 2.05p2	4675
Norton Personal Firewall 2006 9.1.0.33	4600
Safety.Net 3.61.0002	4000
Sygate Personal Firewall 5.6.2808	2350
McAfee Internet Security Suite 2006 8.0	2325
CA Personal Firewall 2007 3.0.0.196	1000
BitDefender Internet Security 10.108	750
F-Secure Internet Security 2007 7.01.128	750
Panda Antivirus + Firewall 2007 6.00.00	650
AVG Anti-Virus plus Firewall 7.5.431	500
Filseclab Personal Firewall 3.0.0.8686	500
Windows Firewall XP SP2	0

Cortafuegos de capa de aplicación

Trabaja en el nivel de aplicación (capa 7 del modelo OSI), de manera que los filtrados se pueden adaptar a características propias de los protocolos de este nivel. Por ejemplo, si se trata de tráfico HTTP, se pueden realizar filtrados según la URL a la que se está intentando acceder.

Un cortafuegos a nivel 7 de tráfico HTTP suele denominarse proxy, y permite que los computadores de una organización entren a Internet de una forma controlada. Un proxy oculta de manera eficaz las verdaderas direcciones de red.

Cortafuegos personal

Es un caso particular de cortafuegos que se instala como software en un computador, filtrando las comunicaciones entre dicho computador y el resto de la red. Se usa por tanto, a nivel personal.

- Clasificación por ubicación.

-Cortafuegos personales (para PC)

Es una novedad en el mercado, Protegen el PC controlando el snack IP e inspeccionando las aplicaciones más comunes, permiten filtros de entrada y salida, y son utilizables en PC en LAN, MODEM, ADSL…

-Cortafuegos para pequeñas oficinas (SOHO)

Small Office Home Office (SOHO)

Protegen a varios usuarios en pequeñas oficinas (típicamente entre 2 y 50), suelen ser pequeños equipos instalados antes del router, o incluso integrados, son muy utilizables para el acceso con ADSL.

-Equipos hardware específicos (Appliances)

Utilizados en oficinas medias y sucursales, fáciles de configurar, con funcionalidades básicas y gestionados centralizadamente, utilizan sistemas operativos propios del hardware en el que están implantados.

-Cortafuegos corporativos

El punto central de accesos a Internet de una empresa. Es el punto central donde se implanta la política de seguridad de la empresa, puede conectar múltiples redes. Software que se instala en grandes servidores con configuraciones tolerantes a fallos.

- Clasificación por tecnología.

-Filtros de paquetes:

Trabajan a nivel de res (IP), filtran paquetes Ip según sus cabeceras y basados en los siguientes criterios:

*Direcciones IP origen y destino

*Puertos TCP/UDP origen y destino

Un router es un cortafuegos básico a nivel de red

Pueden utilizar filtros estáticos o dinámicos

Ventajas: independencia de las aplicaciones, son muy rápidos y escalables

Desventajas: menor nivel de seguridad, no examinan el tráfico ni entienden el contexto

-Proxy de aplicación

Son gateways de aplicación (nivel aplicación), no permiten tráfico entre las dos redes, si no es mediante un Proxy a nivel de aplicación, existe una conexión entre el esterior y el cortafuegos y otra entre el cortafuegos y el interior.

Ventajas: alto nivel de seguridad, examinan información a nivel de aplicación, toman decisiones basadas en datos de cada aplicación

Desventajas: menor rendimiento y escalabilidad, rompe el modelo cliente/servidor (requiere dos conexiones), requiere implantar un Proxy por cada aplicación

-Inspección de estados (statefull inspection)

Realizan filtros en base a las cabeceras, el paquete se intercepta a nivel de red, pero extrae información de los datos para analizar en función de la aplicación, mantiene una tabla dinámica de estados con información para las decisiones de seguridad, no rompe el modelo cliente/servidor

Ventajas: alta seguridad, velocidad y escalabilidad, inspecciona los datos a nivel aplicación.

Desventajas: no se rompe la conexión totalmente.

-Híbridos

La mayoría de los productos comerciales actuales son así, incorporan mezcla de varias tecnologías, pueden definirse reglas de filtros para tráfico que requiere alta velocidad, y Proxy para alta seguridad, adaptativos (Proxy durante el establecimiento y filtro de paquetes durante la transferencia de datos)

- Arquitectura de cortafuegos.

Cortafuegos de filtrado de paquetes

Un *firewall* sencillo puede consistir en un dispositivo capaz de filtrar paquetes, un *choke*: se trata del modelo de cortafuegos más antiguo ([Sch97]), basado simplemente en aprovechar la capacidad de algunos *routers* - denominados *screening routers* - para hacer un enrutado selectivo, es decir, para bloquear o permitir el tránsito de paquetes mediante listas de control de acceso en función de ciertas características de las tramas, de forma que el *router* actue como pasarela de toda la red. Generalmente estas características para determinar el filtrado son las direcciones origen y destino, el protocolo, los puertos origen y destino (en el caso de TCP y UDP), el tipo de mensaje (en el caso de ICMP) y los interfaces de entrada y salida de la trama en el *router*.

En un cortafuegos de filtrado de paquetes los accesos desde la red interna al exterior que no están bloqueados son directos (no hay necesidad de utilizar *proxies*, como sucede en los cortafuegos basados en una máquina con dos tarjetas de red), por lo que esta arquitectura es la más simple de implementar (en muchos casos sobre *hardware* ya ubicado en la red) y la más utilizada en organizaciones que no precisan grandes niveles de seguridad - como las que vemos aquí -. No obstante, elegir un cortafuegos tan sencillo puede no ser recomendable en ciertas situaciones, o para organizaciones que requieren una mayor seguridad para su subred, ya que los simples *chokes* presentan más desventajas que beneficios para la red protegida. El principal problema es que no disponen de un sistema de monitorización sofisticado, por lo que muchas veces el administrador no puede determinar si el *router* está siendo atacado o si su seguridad ha sido comprometida. Además las reglas de filtrado pueden llegar a ser complejas de establecer, y por tanto es difícil comprobar su corrección: habitualmente sólo se comprueba a través de pruebas directas, con los problemas de seguridad que esto puede implicar.

Si a pesar de esto decidimos utilizar un *router* como filtro de paquetes, como en cualquier *firewall* es recomendable bloquear todos los servicios que no se utilicen desde el exterior (especialmente NIS, NFS, X-Window y TFTP), así como el acceso desde máquinas no confiables hacia nuestra subred; además, es también importante para nuestra seguridad bloquear los paquetes con encaminamiento en origen activado.

Dual-Homed Host

El segundo modelo de cortafuegos está formado por simples máquinas Unix equipadas con dos o más tarjetas de red y denominadas ([SH95]) anfitriones de dos bases (*dual-homed hosts*) o multibase (*multi-homed hosts*), y en las que una de las tarjetas se suele conectar a la red interna a proteger y la otra a la red externa a la organización. En esta configuración el *choke* y el bastión coinciden en el mismo equipo: la máquina Unix.

El sistema ha de ejecutar al menos un servidor *proxy* para cada uno de los servicios que deseemos pasar a través del cortafuegos, y también es necesario que el *IP Forwarding* esté deshabilitado en el equipo: aunque una máquina con dos tarjetas puede actuar como un *router*, para aislar el tráfico entre la red interna y la externa es necesario que el *choke* no enrute paquetes entre ellas. Así, los sistemas externos `verán' al *host* a través de una de las tarjetas y los internos a través de la otra, pero entre las dos partes no puede existir ningún tipo de tráfico que no pase por el cortafuegos: todo el intercambio de datos entre las redes se ha de realizar bien a través de servidores *proxy* situados en el *host* bastión o bien permitiendo a los usuarios conectar directamente al mismo. La segunda de estas aproximaciones es sin duda poco recomendable, ya que un usuario que consiga aumentar su nivel de privilegios en el sistema puede romper toda la protección del cortafuegos, por ejemplo reactivando el *IP Forwarding*); además - esto ya no relativo a la seguridad sino a la funcionalidad del sistema - suele ser incómodo para los usuarios tener que acceder a una máquina que haga de puente entre ellos e Internet. De esta forma, la ubicación de *proxies* es lo más recomendable, pero puede ser problemático el configurar cierto tipo de servicios o protocolos que no se diseñaron teniendo en cuenta la existencia de un *proxy* entre los dos extremos de una conexión.

Screened Host

Un paso más en términos de seguridad de los cortafuegos es la arquitectura *screened host* o *choke-gate*, que combina un *router* con un *host* bastión, y donde el principal nivel de seguridad proviene del filtrado de paquetes (es decir, el *router* es la primera y más importante línea de defensa). En la máquina bastión, único sistema accesible desde el exterior, se ejecutan los *proxies* de las aplicaciones, mientras que el *choke* se encarga de filtrar los paquetes que se puedan considerar peligrosos para la seguridad de la red interna, permitiendo únicamente la comunicación con un reducido número de servicios.

Pero, >dónde situar el sistema bastión, en la red interna o en el exterior del *router*? La

mayoría de autores recomiendan situar el *router* entre la red exterior y el *host* bastión, pero otros defienden justo lo contrario: situar el bastión en la red exterior no provoca aparentemente una degradación de la seguridad, y además ayuda al administrador a comprender la necesidad de un elevado nivel de fiabilidad en esta máquina, ya que está sujeta a ataques externos y no tiene por qué ser un *host* fiable; de cualquier forma, la `no degradación' de la seguridad mediante esta aproximación es más que discutible, ya que habitualmente es más fácil de proteger un *router* que una máquina con un operativo de propósito general, como Unix, que además por definición ha de ofrecer ciertos servicios: no tenemos más que fijarnos en el número de problemas de seguridad que afectan a por ejemplo a IOS (el sistema operativo de los *routers* Cisco), muy reducido frente a los que afectan a diferentes *flavours* de Unix. En todo caso, aparte de por estos matices, asumiremos la primera opción por considerarla mayoritaria entre los expertos en seguridad informática; así, cuando una máquina de la red interna desea comunicarse con el exterior existen dos posibilidades:

- El *choke* permite la salida de algunos servicios a todas o a parte de las máquinas internas a través de un simple filtrado de paquetes.
- El *choke* prohibe todo el tráfico entre máquinas de la red interna y el exterior, permitiendo sólo la salida de ciertos servicios que provienen de la máquina bastión y que han sido autorizados por la política de seguridad de la organización. Así, estamos obligando a los usuarios a que las conexiones con el exterior se realicen a través de los servidores *proxy* situados en el bastión.

La primera aproximación entraña un mayor nivel de complejidad a la hora de configurar las listas de control de acceso del *router*, mientras que si elegimos la segunda la dificultad está en configurar los servidores *proxy* (recordemos que no todas las aplicaciones soportan bien estos mecanismos) en el *host* bastión. Desde el punto de vista de la seguridad es más recomendable la segunda opción, ya que la probabilidad de dejar escapar tráfico no deseado es menor. Por supuesto, en función de la política de seguridad que definamos en nuestro entorno, se pueden combinar ambas aproximaciones, por ejemplo permitiendo el tráfico entre las máquinas internas y el exterior de ciertos protocolos difíciles de encaminar a través de un *proxy* o sencillamente que no entrañen mucho riesgo para nuestra seguridad (típicamente, NTP, DNS...), y obligando para el resto de servicios a utilizar el *host* bastión.

La arquitectura *screened host* puede parecer a primera vista más peligrosa que la basada en una simple máquina con varias interfaces de red; en primer lugar, tenemos no uno sino dos sistemas accesibles desde el exterior, por lo que ambos han de ser configurados con las máximas medidas de seguridad. Además, la mayor complejidad de diseño hace más fácil la presencia de errores que puedan desembocar en una

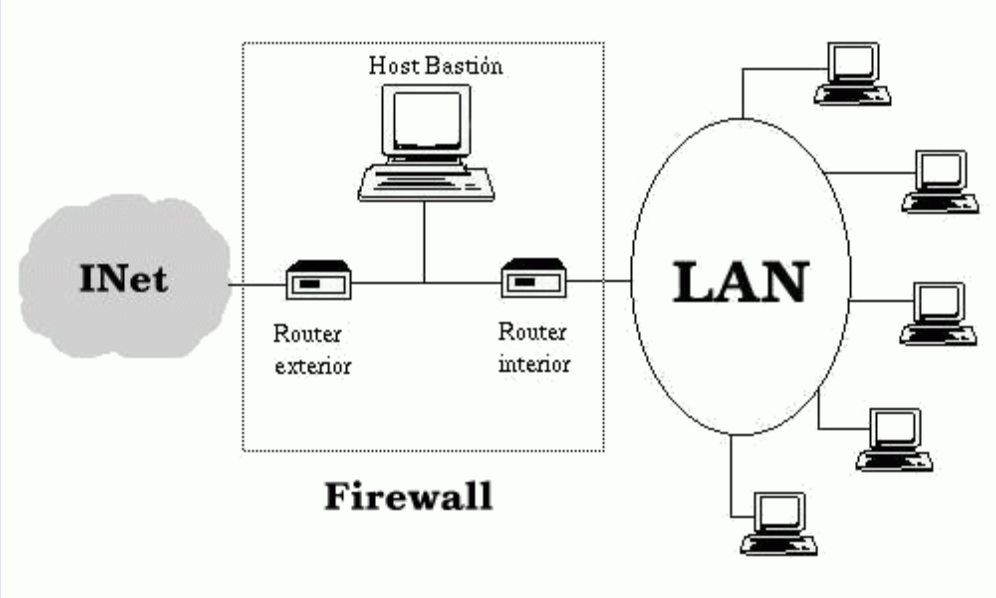
violación de la política implantada, mientras que con un *host* con dos tarjetas nos aseguramos de que únicamente aquellos servicios con un *proxy* configurado podrán generar tráfico entre la red externa y la interna (a no ser que por error activemos el *IP Forwarding*). Sin embargo, aunque estos problemas son reales, se solventan tomando las precauciones necesarias a la hora de diseñar e implantar el cortafuegos y definiendo una política de seguridad correcta. De cualquier forma, en la práctica esta arquitectura de cortafuegos está cada vez más en desuso debido a que presenta dos puntos únicos de fallo, el *choke* y el bastión: si un atacante consigue controlar cualquiera de ellos, tiene acceso a toda la red protegida; por tanto, es más popular, y recomendable, una arquitectura *screened subnet*, de la que vamos a hablar a continuación.

Screened Subnet (DMZ)

La arquitectura *Screened Subnet*, también conocida como red perimétrica o *De-Militarized Zone* (DMZ) es con diferencia la más utilizada e implantada hoy en día, ya que añade un nivel de seguridad en las arquitecturas de cortafuegos situando una subred (DMZ) entre las redes externa e interna, de forma que se consiguen reducir los efectos de un ataque exitoso al *host* bastión: como hemos venido comentando, en los modelos anteriores toda la seguridad se centraba en el bastión, de forma que si la seguridad del mismo se veía comprometida, la amenaza se extendía automáticamente al resto de la red. Como la máquina bastión es un objetivo interesante para muchos piratas, la arquitectura DMZ intenta aislarla en una red perimétrica de forma que un intruso que accede a esta máquina no consiga un acceso total a la subred protegida.

Screened subnet es la arquitectura más segura, pero también la más compleja; se utilizan dos *routers*, denominados exterior e interior, conectados ambos a la red perimétrica. En esta red perimétrica, que constituye el sistema cortafuegos, se incluye el *host* bastión y también se podrían incluir sistemas que requieran un acceso controlado, como baterías de módems o el servidor de correo, que serán los únicos elementos visibles desde fuera de nuestra red. El *router* exterior tiene como misión bloquear el tráfico no deseado en ambos sentidos (hacia la red perimétrica y hacia la red externa), mientras que el interior hace lo mismo pero con el tráfico entre la red interna y la perimétrica: así, un atacante habría de romper la seguridad de ambos *routers* para acceder a la red protegida; incluso es posible implementar una zona desmilitarizada con un único *router* que posea tres o más interfaces de red, pero en este caso si se compromete este único elemento se rompe toda nuestra seguridad, frente al caso general en que hay que comprometer ambos, tanto el externo como el interno. También podemos, si necesitamos mayores niveles niveles de seguridad,

definir varias redes perimétricas en serie, situando los servicios que requieran de menor fiabilidad en las redes más externas: así, el atacante habrá de saltar por todas y cada una de ellas para acceder a nuestros equipos; evidentemente, si en cada red perimétrica se siguen las mismas reglas de filtrado, niveles adicionales no proporcionan mayor seguridad.



Esta arquitectura de cortafuegos elimina los puntos únicos de fallo presentes en las anteriores: antes de llegar al bastión (por definición, el sistema más vulnerable) un atacante ha de saltarse las medidas de seguridad impuestas por el enrutador externo. Si lo consigue, como hemos aislado la máquina bastión en una subred estamos reduciendo el impacto de un atacante que logre controlarlo, ya que antes de llegar a la red interna ha de comprometer también al segundo *router*; en este caso extremo (si un pirata logra comprometer el segundo *router*), la arquitectura DMZ no es mejor que un *screened host*. Por supuesto, en cualquiera de los tres casos (compromiso del *router* externo, del *host* bastión, o del *router* interno) las actividades de un pirata pueden violar nuestra seguridad, pero de forma parcial: por ejemplo, simplemente accediendo al primer enrutador puede aislar toda nuestra organización del exterior, creando una negación de servicio importante, pero esto suele ser menos grave que si lograra acceso a la red protegida.

Aunque, como hemos dicho antes, la arquitectura DMZ es la que mayores niveles de seguridad puede proporcionar, no se trata de la panacea de los cortafuegos.

Evidentemente existen problemas relacionados con este modelo: por ejemplo, se puede utilizar el *firewall* para que los servicios fiables pasen directamente sin acceder al bastión, lo que puede dar lugar a un incumplimiento de la política de la organización. Un segundo problema, quizás más grave, es que la mayor parte de la seguridad reside en los *routers* utilizados; como hemos dicho antes las reglas de filtrado sobre estos elementos pueden ser complicadas de configurar y comprobar, lo que puede dar lugar a errores que abran importantes brechas de seguridad en nuestro sistema.

Otras arquitecturas

Algo que puede incrementar en gran medida nuestra seguridad y al mismo tiempo facilitar la administración de los cortafuegos es utilizar un bastión diferente para cada protocolo o servicio en lugar de uno sólo; sin embargo, esta arquitectura presenta el grave inconveniente de la cantidad de máquinas necesarias para implementar el *firewall*, lo que impide que muchas organizaciones la puedan adoptar. Una variante más barata consistiría en utilizar un único bastión pero servidores *proxy* diferentes para cada servicio ofertado.

Cada día es más habitual en todo tipo de organizaciones dividir su red en diferentes subredes; esto es especialmente aplicable en entornos de I+D o empresas medianas, donde con frecuencia se han de conectar campus o sucursales separadas geográficamente, edificios o laboratorios diferentes, etc. En esta situación es recomendable incrementar los niveles de seguridad de las zonas más comprometidas (por ejemplo, un servidor donde se almacenen expedientes o datos administrativos del personal) insertando cortafuegos internos entre estas zonas y el resto de la red. Aparte de incrementar la seguridad, *firewalls* internos son especialmente recomendables en zonas de la red desde la que no se permite *a priori* la conexión con Internet, como laboratorios de prácticas: un simple PC con Linux o FreeBSD que deniegue cualquier conexión con el exterior del campus va a ser suficiente para evitar que los usuarios se dediquen a conectar a páginas *web* o *chats* desde equipos no destinados a estos usos. Concretamente en el caso de redes de universidades sería muy interesante filtrar las conexiones a IRC o a MUDs, ya sea a nivel de aulas o laboratorios o a nivel de todo el campus, denegando en el *router* de salida de la red hacia INet cualquier tráfico a los puertos 6667, 8888 y similares; aunque realmente esto no evitaría que todos los usuarios siguieran jugando desde los equipos de la universidad - por ejemplo a través de un servidor que disponga de conexión en otros puertos -, sí conseguiría que la mayor parte de ellos dejara de hacerlo.

- Pruebas de funcionamiento. Sondeo.

Los expertos conocen docenas de técnicas de sondeo y eligen la más apropiada (o una combinación de éstas) para la tarea que están realizando. Los usuarios sin experiencia y los "script kiddies", sin embargo, intentan resolver cada problema con el sondeo SYN por omisión. Dado que Nmap es libre, la única barrera que existe para ser un experto en el sondeo de puertos es el conocimiento.

La mayoría de los distintos tipos de sondeo disponibles sólo los puede llevar a cabo un usuario privilegiado. Esto es debido a que envían y reciben paquetes en crudo, lo que hace necesario tener acceso como administrador (root) en la mayoría de los sistemas UNIX. En los entornos Windows es recomendable utilizar una cuenta de administrador, aunque Nmap algunas veces funciona para usuarios no privilegiados en aquellas plataformas donde ya se haya instalado WinPcap. La necesidad de privilegios como usuario administrador era una limitación importante cuando se empezó a distribuir Nmap en 1997, ya que muchos usuarios sólo tenían acceso a cuentas compartidas en sistemas como usuarios normales. Ahora, las cosas son muy distintas. Los ordenadores son más baratos, hay más personas que tienen acceso permanente a Internet, y los sistemas UNIX (incluyendo Linux y MAC OS X) son más comunes. También se dispone de una versión para Windows de Nmap, lo que permite que se ejecute en más escritorios. Por todas estas razones, cada vez es menos necesario ejecutar Nmap utilizando cuentas de sistema compartidas. Esto es bueno, porque las opciones que requieren de más privilegios hacen que Nmap sea más potente y flexible.

Aunque Nmap intenta generar resultados precisos, hay que tener en cuenta que estos resultados se basan en los paquetes que devuelve el sistema objetivo (o los cortafuegos que están delante de éstos). Estos sistemas pueden no ser fiables y enviar respuestas cuyo objetivo sea confundir a Nmap. Son aún más comunes los sistemas que no cumplen con los estándares RFC, que no responden como deberían a las sondas de Nmap. Son especialmente susceptibles a este problema los sondeos FIN, Null y Xmas. Hay algunos problemas específicos a algunos tipos de sondeos que se discuten en las entradas dedicadas a sondeos concretos.

Esta sección documenta las aproximadamente doce técnicas de sondeos de puertos que soporta Nmap. Sólo puede utilizarse un método en un momento concreto, salvo por el sondeo UDP (-sU) que puede combinarse con cualquiera de los sondeos TCP. Para que sea fácil de recordar, las opciones de los sondeos de puertos son del estilo -s<*C*>, donde <*C*> es una letra característica del nombre del sondeo, habitualmente la primera. La única excepción a esta regla es la opción obsoleta de sondeo FTP rebotado (-b). Nmap hace un sondeo SYN por omisión, aunque lo cambia a un sondeo Connect() si el usuario no tiene los suficientes privilegios para enviar paquetes en crudo (requiere acceso de administrador en UNIX) o si se especificaron objetivos IPv6. De los sondeos que se listan en esta sección los usuarios sin privilegios sólo pueden ejecutar los sondeos Connect() o de rebote FTP.

-sS (sondeo TCP SYN)

El sondeo SYN es el utilizado por omisión y el más popular por buenas razones. Puede realizarse rápidamente, sondeando miles de puertos por segundo en una red rápida en la que no existan cortafuegos. El sondeo SYN es relativamente sigiloso y poco molesto, ya que no llega a completar las conexiones TCP.

-sT (sondeo TCP connect())

El sondeo TCP Connect() es el sondeo TCP por omisión cuando no se puede utilizar el sondeo SYN. Esto sucede, por ejemplo, cuando el usuario no tiene privilegios para enviar paquetes en crudo o cuando se están sondeando redes IPv6. Nmap le pide al sistema operativo subyacente que establezcan una conexión con el sistema objetivo en el puerto indicado utilizando la llamada del sistema connect(), a diferencia de otros tipos de sondeo, que escriben los paquetes a bajo nivel

-sU (sondeos UDP)

Aunque la mayoría de los servicios más habituales en Internet utilizan el protocolo TCP, los servicios UDP también son muy comunes. Tres de los más comunes son los servicios DNS, SNMP, y DHCP (puertos registrados 53, 161/162, y 67/68 respectivamente). Dado que el sondeo UDP es generalmente más lento y más difícil que TCP, algunos auditores de seguridad ignoran estos puertos. Esto es un error, porque es muy frecuente encontrarse servicios UDP vulnerables y los atacantes no ignoran estos protocolos. Afortunadamente, Nmap puede utilizarse para hacer un inventario de puertos UDP.

El sondeo UDP se activa con la opción -sU. Puede combinarse con un tipo de sondeo TCP como el sondeo SYN (-sS) para comprobar ambos protocolos al mismo tiempo.

-sA (sondeo TCP ACK)

Este sondeo es distinto de otros que se han discutido hasta ahora en que no puede determinar puertos abiertos (o incluso abiertos|filtrados). Se utiliza para mapear reglas de cortafuegos, y para determinar si son cortafuegos con inspección de estados y qué puertos están filtrados.

La sonda de un sondeo ACK sólo tiene fijada la bandera ACK (a menos que utilice --scanflags). Cuando se sondean sistemas no filtrados los puertos abiertos y cerrados devolverán un paquete RST. Nmap marca el puerto como no filtrado, lo que significa que son alcanzables por el paquete ACK, pero no se puede determinar si están abiertos o cerrados. Los puertos que no responden o

que envían mensajes de error ICMP en respuesta (tipo 3, código 1, 2, 3, 9, 10, o 13), se marcan como filtrados.

-sO (sondeo de protocolo IP)

El sondeo de protocolo IP le permite determinar qué protocolos (TCP, ICMP, IGMP, etc.) soportan los sistemas objetivo. Esto no es, técnicamente, un sondeo de puertos, dado que cambia los números de protocolo IP en lugar de los números de puerto TCP ó UDP. Pero también se puede utilizar la opción -p para seleccionar los números de protocolo a analizar, los resultados se muestran en el formato de tabla utilizado para los puertos e incluso utiliza el mismo motor de sondeo que los métodos de sondeo de puertos reales. Es tan parecido a un sondeo de puertos que debe tratarse aquí.

-b <sistema de rebote ftp> (sondeo de rebote FTP)

Una funcionalidad interesante en el protocolo FTP (RFC 959) es la posibilidad de utilizar conexiones FTP de pasarela. Esta opción puede abusarse a muchos niveles así que muchos servidores han dejado de soportarla. Una de las formas de abusar de ésta es utilizar el servidor de FTP para hacer un sondeo de puertos a otro sistema. Simplemente hace falta decirle al servidor de FTP que envíe un fichero a cada puerto interesante del servidor objetivo cada vez. El mensaje de error devuelto indicará si el puerto está abierto o no. Esta es una buena manera de atravesar cortafuegos porque, habitualmente, los servidores de FTP de una organización están ubicados en un lugar en el que tienen más acceso a otros sistemas internos que el acceso que tiene un equipo en Internet. Nmap puede hacer sondeos con rebotes de FTP con la opción -b. Esta opción toma un argumento como: *<usuario>*:*<contraseña>*@*<servidor>*:*<puerto>*. *<Servidor>* es el nombre de la dirección IP del servidor FTP vulnerable. Al igual que con una URL normal, se puede omitir *<usuario>*:*<contraseña>*, en caso de que se deseen utilizar credenciales de acceso anónimo (usuario: anonymous contraseña:wwwuser@) También se puede omitir el número de puerto (y los dos puntos que lo preceden). Si se omiten se utilizará el puerto FTP estándar (21) en *<servidor>*.

•Cortafuegos software y hardware:

Cortafuegos de hardware

Los cortafuegos de hardware proporcionan una fuerte protección contra la mayoría de las formas de ataque que vienen del mundo exterior y se pueden comprar como producto independiente o en routers de banda ancha.

Desafortunadamente, luchando contra virus, gusanos y Troyanos, un cortafuegos de hardware puede ser menos eficaz que un cortafuegos de software, pues podría no detectar gusanos en emails. 

Los firewall de hardware se utilizan más en empresas y grandes corporaciones. Normalmente son dispositivos que se colocan entre el router y la conexión telefónica. Como ventajas, podemos destacar, que al ser independientes del PC, no es necesario configurarlos cada vez que reinstalamos el sistema operativo, y no consumen recursos del sistema.

Su mayor inconveniente es el mantenimiento, ya que son difíciles de actualizar y de configurar correctamente

Cortafuegos de software

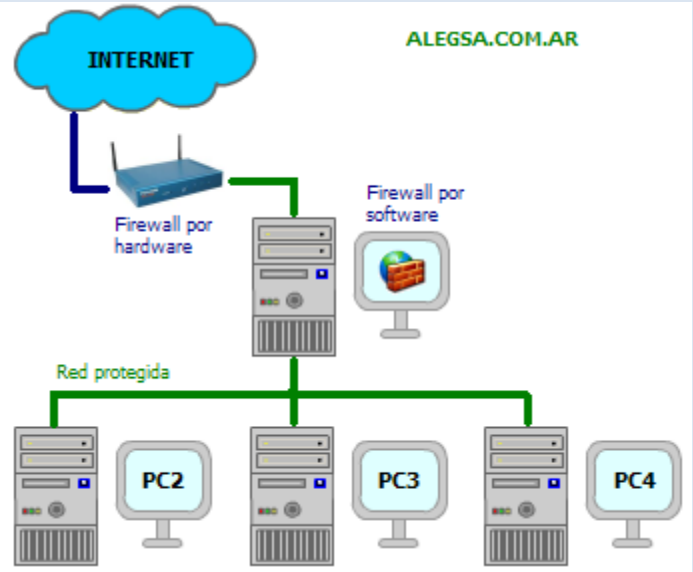
Para usuarios particulares, el cortafuegos más utilizado es un **cortafuego de software**. Un buen cortafuegos de software protegerá tu ordenador contra intentos de controlar o acceder a tu ordenador desde el exterior, y generalmente proporciona protección adicional contra los troyanos o gusanos de E-mail más comunes.

Estos programas son los más comunes en los hogares, ya que a parte de resultar mucho más económicos que el hardware, su instalación y actualización es más sencilla. Eso sí, presentan algunos problemas inherentes a su condición: consumen recursos del ordenador, algunas veces no se ejecutan correctamente o pueden ocasionar errores de compatibilidad con otro software instalado.

Actualmente, los sistemas operativos más modernos como Windows XP y Linux integran soluciones básicas de firewall, en algunos casos, como en el software libre, son muy potentes y flexibles, pero requieren una gran conocimiento en redes y puertos necesarios para las aplicaciones. Para no tener problemas, existen una serie de

herramientas externas que facilitan este trabajo de protección, como el **Kit Seguridad de Terra**, una solución que junta firewall y antivirus.

La desventaja de los cortafuegos de software es que protegen solamente al ordenador en el que están instalados y no protegen una red.



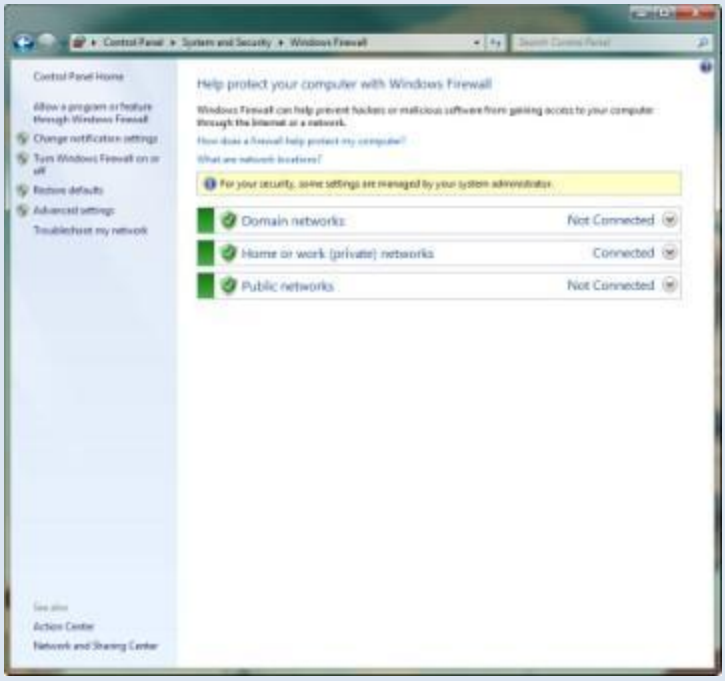
Arquitectura hardware: servidores estándar Windows o Unix con S:O: Seguros o reforzados, hardware específico

- Cortafuegos software integrados en los sistemas operativos.

Si utiliza Windows 7, Windows Vista o Windows XP Service Pack 2 (SP2), ya dispone de un firewall integrado y activado de forma predeterminada como parte de los beneficios que le brindan estos Sistemas Operativos.

Si utiliza Windows XP pero no lo tiene actualizado con el Service Pack 2, sigue teniendo acceso al Firewall de conexión a Internet (ICF) integrado en Windows XP, pero tendrá que activarlo.

Si desconoce lo que es el Service Pack 2 o no tiene una conexión de banda ancha a Internet para descargarlo y actualizar su PC, le agradará saber que para su comodidad hemos incorporado el Service Pack 2 de Windows XP en este DVD.

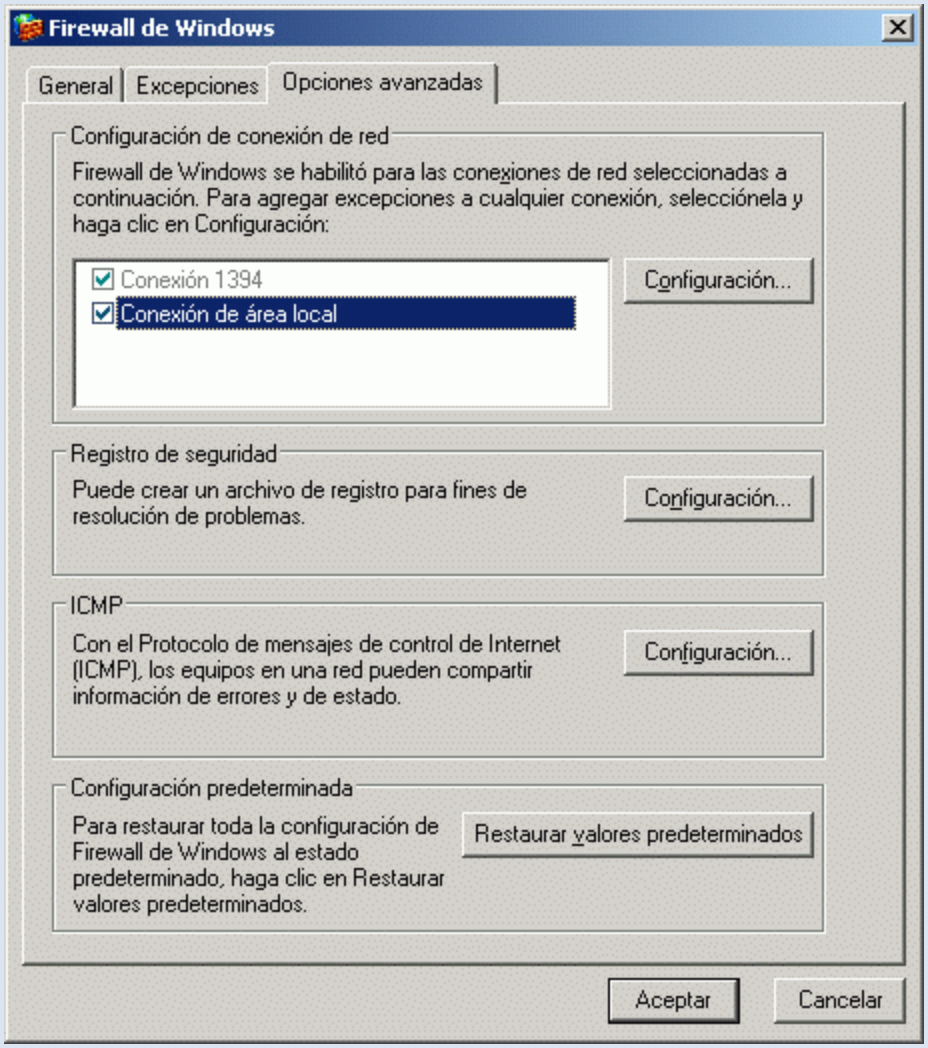


Abrir puertos cortafuegos Windows XP

Si el sistema operativo que utilizamos es Windows XP Profesional, no la versión HOME ya que carece de esta opción, proporciona un cortafuegos integrado.
Este cortafuegos funciona como cualquier otro cortafuegos del mercado.
Para que algunas aplicaciones funcionen correctamente, Emule, Juegos, Telnet, se necesita que tengan abierto ciertos puertos de entrada y de salidad y tanto TCP, UDP u otros.
Para abrir puertos en el cortafuegos integrado de windows realizar lo siguiente:

Primero estar seguro de que esta habilitado. Para esto ya existe una tutorial **Habilitar, Deshabilitar cortafuegos integrado**.

Una vez dentro del cortafuegos se pincha en opciones avanzadas

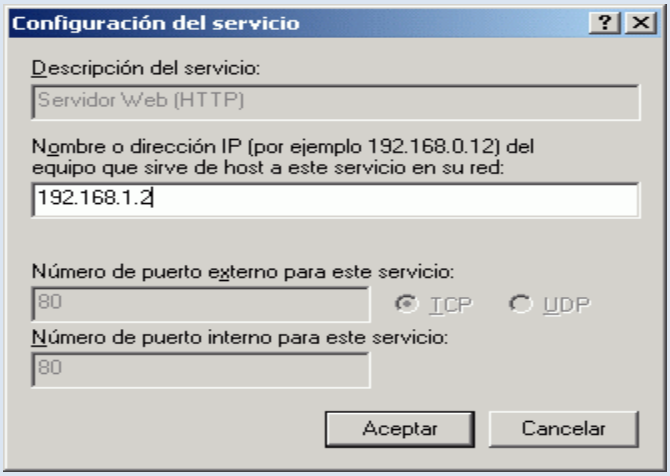


Una vez dentro del cortafuegos se pincha en opciones avanzadas y en la conexión que se esta utilizando para tener acceso a la red. Seguidamente se pincha en configuración y se abrirá una pantalla de configuración avanzada

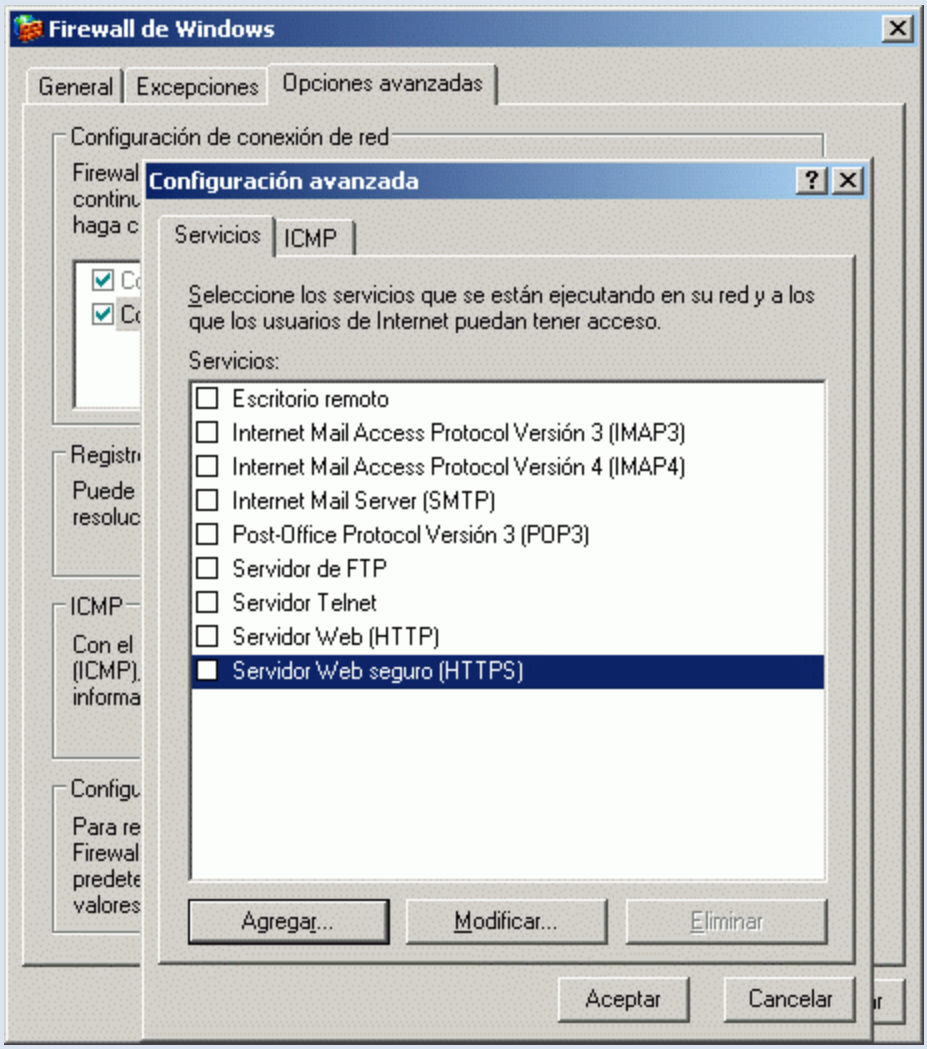
En esa pantalla aparecen posiblemente desactivados algunos servicios, como servidor web, telnet etc. Estos servicios como el servidor web necesitan estar "abiertos" al exterior para poder mostrar, en este caso del servidor web, las páginas web o cualquier servicio que se quiera ejecutar.

Si se quiere abrir los puertos del servidor web se pincha en Servidor Web - Por defecto tiene asignado el puerto 80.

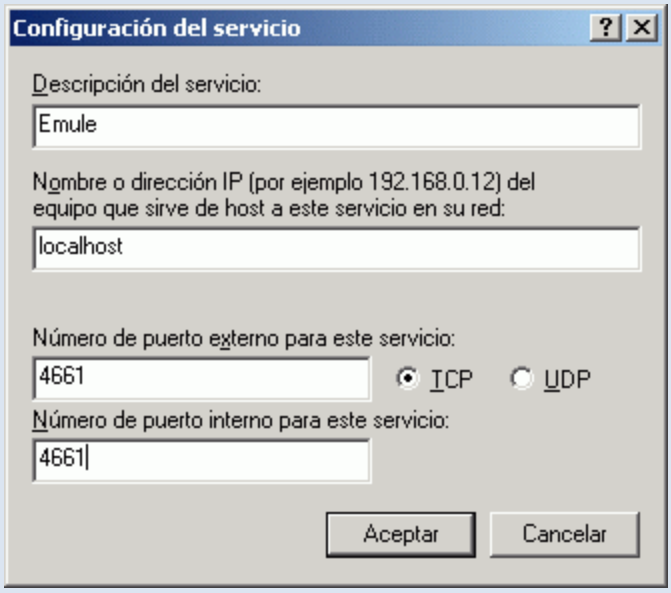
El nombre de la máquina se puede cambiar asignando la dirección IP de la máquina. Hay un tutorial sobre esto en **Sacar IP de la máquina desde cmd** , si no se esta seguro dejarlo como esta.



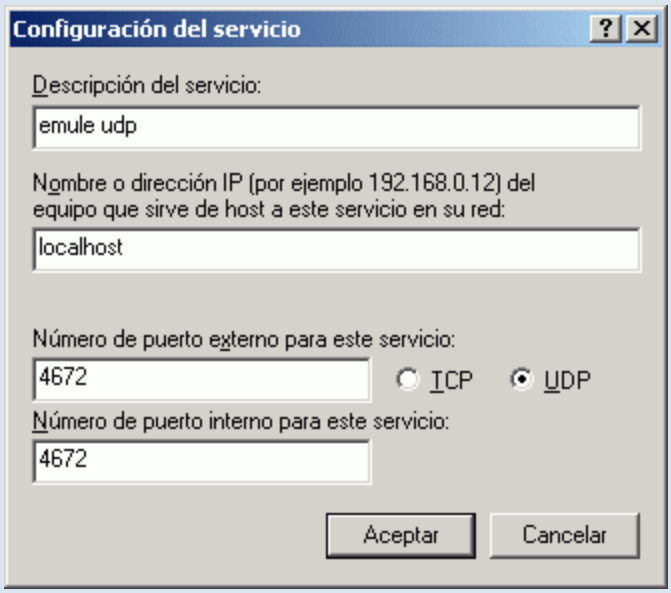
Para abrir puertos de un servicio que no este listado pinchar en agregar en la ventana de configuración avanzada.



Se va añadir la configuración para Emule. Se escribe el nombre de servicio, el nombre es indiferente, la dirección de la máquina, aquí aparece localhost, ya que localhost significa la propia máquina, se escribe el puerto de entrada y de salida 4661 en este caso TCP.



También hay que añadir la configuración de otro puerto en este caso UPD. Seleccionar UDP en vez de TCP y listos



Así se abren los puertos. La mayoría de las veces las aplicaciones especifican que puertos se deben abrir y que tipo así que lo único que se tiene que hacer es abrir esos puertos y la aplicación se encargará de usarlos o no. Nada más.

- Cortafuegos software libres y propietarios.

El Software Libre y la seguridad informática

¿Qué es el Software Libre?

Para entender la situación de este tipo de software con respecto a su uso en seguridad informática es imprescindible describir, en primer lugar, a qué se refiere este documento cuando hace referencia a "software libre".

El concepto de software libre es, en primera instancia, fácil de presentar, aún no existiendo una única descripción reconocida por todos de lo que es realmente este tipo de software. En general se entiende como software libre aquel programa o conjunto de ellos de los que el usuario puede disponer del código fuente, sin restricciones, y el cual puede modificar y redistribuir también sin restricciones. Estas libertades garantizadas al usuario del software (o a aquel que lo recibe) no son contrarias a los derechos legítimos del autor del programa, es decir, éste no tiene por qué perder sus derechos sobre el mismo. No se incluye, por tanto, en esta definición software en el "domino público" (aquél para en el que el autor ha cedido todos sus derechos).

Una descripción más completa de lo que podría considerarse software libre, es la dada por las directrices de software Libre de Debian, que constituyen la base de la definición de *Open Source* (Open Source Definition, www.opensource.org), aunque existen entre ellas ciertas diferencias. Entre las licencias más utilizadas para este tipo de software cabe destacar la licencia GNU GPL y la licencia BSD.

Ventajas del Software Libre en el mundo de la seguridad

Si se analiza la descripción realizada previamente de la definición de software libre se derivan una serie de ventajas principales de este tipo de software sobre el software propietario, algunas de las cuales son muy adecuadas para el mundo de la seguridad. A saber:

- Al disponer del código fuente de los programas en su totalidad, éste puede ser analizado por terceras personas ajenas a sus autores en busca de fallos de diseño o de implementación. Es decir, cualquiera con los conocimientos necesarios puede realizar una auditoría del código del program.
- La posibilidad de realizar modificaciones libremente al código fuente y distribuirlas permite que cualquiera pueda ofrecer mejoras sobre éste. Estas mejoras podrán ser nuevas funcionalidades que se incorporen al mismo o parches que corrijan problemas detectados anteriormente.
- Las características del software libre hacen que no sea lógico cargar costes sobre el software en sí (dado que se ha de distribuir sin cargo), lo que permite

que este tipo de software pueda ser utilizado por organizaciones y personas con menos recursos económicos. Esto se presenta como una ventaja cuando se compara con los precios de lo que cuesta el software de seguridad propietario hoy en día (licencias de cortafuegos, vpns, sistemas de detección de intrusos, etc.). El software libre pone en manos de cualquiera el tipo de tecnología que, hoy por hoy, sólo podían tener grandes corporaciones.

- De igual forma, la posibilidad de modificar libremente el software permite a las organizaciones que lo adapten a sus propias necesidades, pudiendo eliminar funcionalidades que no le sean de interés. En el mundo de la seguridad existe la máxima de "lo más sencillo es más seguro" por ello poder eliminar funciones innecesarias de las herramientas las puede convertir de forma inmediata en más seguras (porque no podrán ser utilizadas estas funcionalidades para subvertirlas).

Frente al análisis de fallos que puede sobrevenir en la realización del software (presentado anteriormente), el software libre protege a sus usuarios con una serie de mecanismos determinados. Entre estos:

- La posibilidad de una auditoría de código en las herramientas software reduce los riesgos de seguridad debido a la aparición de fallos desconocidos, a la introducción de funcionalidades no deseadas en el código o la incorrecta implementación de algoritmos públicos. Aunque no se pueda asegurar que el código esté carente de errores, si es posible garantizar que tantas posibilidades tiene de encontrar un fallo de programación en éste (que lleve implícito un riesgo de seguridad) un atacante externo como la organización lo utilice. Si bien no se puede asegurar que los mejores cerebros del mundo realicen la auditoría de código del software que una compañía utiliza, dicha compañía si tiene la posibilidad, en función de sus necesidades respecto a la seguridad, de realizar ella misma dicha auditoría de código o pagar a alguien para que la realice. Muchos de los proyectos de software libre, entre ellos el núcleo de Linux, el proyecto Apache, y la distribución OpenBSD realizan auditorías del código para asegurar su integridad, seguridad y ajuste a las especificaciones de funcionalidades requeridas.
- La posibilidad de corregir los programas y distribuir dichas correcciones permite que los programas evolucionen de una forma más abierta. En el mundo de la seguridad, un fallo en el sistema significa exponer a éste a una "ventana de vulnerabilidad" que tiene lugar desde la detección del fallo (por parte de sus usuarios legítimos o de terceras partes, hostiles incluso) a la aplicación de la medida correctiva, que pueda ser la instalación del parche adecuado que arregle el problema, pasando por la *generación* de dicho parche. El hecho de que la generación de dicho parche pueda realizarse por un número de personas (confiables) elevado, y no por un sólo fabricante, debe, en teoría, reducir este tiempo de exposición a dicha vulnerabilidad.

- El hecho de que exista una cierta independencia entre el software y su fabricante, o distribuidor original, permite que los usuarios de este software, en caso de pérdida de soporte, puedan realizar el mantenimiento de éste ellos mismos o subcontratarlo a una tercera empresa. Este hecho es, si cabe, de gran importancia en el mundo de la seguridad dado que la seguridad de una entidad no debe depender de la solvencia de terceras compañías a las que adquiere productos de seguridad y actualmente, sin embargo, es así. Debido a la gran variabilidad de riesgos potenciales contra los que un elemento de seguridad informática debe proteger, estos productos han de ser frecuentemente actualizados, muchas veces empujados por el descubrimiento de ataques antes desconocidos. Sin embargo, si una compañía depende de un producto de una tercera entidad y, de forma transitiva, de esta tercera entidad, la pérdida de soporte de este producto (por quiebra de la tercera entidad o abandono de una determinada línea de negocio) da lugar a que la compañía no esté adecuadamente asegurada contra los nuevos riesgos que puedan surgir. Las únicas opciones posibles serán mantener un sistema de seguridad que, con el tiempo, quedará obsoleto, o migrar a un sistema de seguridad nuevo (otro producto de otro fabricante) con sus consecuencias económicas y de impacto en servicios ya consolidados.

Las auditorías de código son, por tanto, posibles o no en determinados sistemas operativos en función de la publicidad dada a su código fuente. Sin embargo, no basta con decir qué se puede hacer una auditoría del código, es necesario considerar los resultados de dichas auditorías. Si bien Microsoft y Sun ofrecen el código fuente de su sistema operativo (el primero con más restricciones que el segundo), ninguno de los dos incorporará, necesariamente, los resultados de una auditoría de código sobre la base del sistema operativo realizado por terceras entidades. Los criterios para tomar dicha decisión no dependen de la auditoría en sí sino de la política de la propia compañía. Sin embargo, en la auditoría que se pueda realizar a sistemas operativos libres, como es el caso de GNU/Linux o BSD, la aplicación de los resultados o no se realiza mediante una discusión pública y es el propio resultado de la auditoría el que debe valer por si mismo para su introducción o no. No existen presiones comerciales de pérdida de imagen, ni el "time to market" ni ningún tipo de consideraciones que no sean las puramente técnicas. Este mismo hecho, la modificación inmediata del código y su distribución, es el que puede dar lugar a que, aun cuando Sun distribuya de forma pública el código de Solaris, se audite de forma más intensiva el código de GNU/Linux o BSD, ya que son las propias personas que realizan la auditoría las que pueden sugerir implementaciones de las modificaciones sugeridades que se podrán incorporarán rápidamente en el código auditado.

Desventajas del software propietario

En primer lugar, es necesario aclarar que en este documento se entenderá como software propietario aquél que se distribuye en forma de binarios, sin código fuente,

por parte de una compañía que licencia dicho software para un uso concreto, con un coste determinado. No se van a realizar comparativas con la nebulosa intermedia de distintos tipos de software cuyas licencias se sitúan entre ambos extremos, por ejemplo: software que se distribuye el código fuente pero no se puede modificar, software que se distribuye con limitaciones para su uso comercial, etc.

Con respecto a la seguridad, las mismas garantías que ofrece el software libre en el mundo de la seguridad son problemas que se le pueden achacar al software propietario. Se puede hablar de las siguientes desventajas del software propietario para el usuario final:

- Posibilidad de que existan funcionalidades no deseadas en dicho software. Dependiendo de la programación realizada, algunas funcionalidades podrán ser activadas o desactivadas por el usuario, pero pueden existir también funcionalidades que no se puedan desactivar o que, incluso, no se encuentren documentadas. Llevándolo al extremo se podría hablar de "puertas traseras" abiertas por el fabricante del software que, después de todo, es un agente comercial y, por tanto, tiene sus propios intereses que pueden ser contrarios a los de la compañía que instala un software de seguridad específico.
- Desconocimiento del código por parte del usuario. Esto puede llevar a que el fabricante pueda llegar a tener una falsa sensación de seguridad por oscuridad, es decir, las vulnerabilidades de su producto no tienen por qué ser conocidas porque nadie tiene acceso a las "tripas" del mismo. De igual forma, esto puede llevar a que el fabricante no tenga interés en desarrollar el código de una forma adecuada porque, al fin y al cabo, el usuario no va a ver dicho código ni evaluar la calidad de su implementación.
- Necesidad de confiar totalmente en el fabricante. Esto es así por cuanto éste ha implementado los algoritmos de seguridad y el usuario no puede garantizar por sí mismo que su implementación ha sido correcta y que, por ejemplo, las propiedades matemáticas necesarias para que estos algoritmos funcionen correctamente se cumplan en todas las condiciones.
- Dependencia de una tercera entidad, ya que es el fabricante del producto el único que puede ofrecer nuevas versiones de éste en caso de fallo o incluir nuevas funcionalidades que puedan ser necesarias. Esto es una desventaja debido a que el usuario no puede transferir esta dependencia a otra entidad, en caso de que el fabricante original haya traicionado su confianza (demasiados errores en la implementación, demasiado tiempo en la generación de parches para arreglar problemas graves, etc..)

Cabe hacer notar que, algunos fabricantes de software, observando las ventajas del modelo *Open Source* ofrecen, con restricciones o sin ellas, copias del código fuente a terceras entidades interesadas. Tal es el caso, por ejemplo, de fabricantes de sistemas operativos como Sun Microsystems y Microsoft y de fabricantes de productos de seguridad como PGP (hasta febrero de 2001 con su suite de aplicaciones basadas en cifrado asimétrico) y NAI (con su cortafuegos Gauntlet).

Desventajas del software libre

Sin embargo, el uso de software libre no está exento de desventajas. Así se podrían enumerar las siguientes:

- la posibilidad de una generación más fácil de troyanos, dado que el código fuente también puede ser modificado con intenciones maliciosas. Si el troyano logra confundirse con la versión original puede haber problemas graves. La fuente del programa, en realidad, será el método de distribución de software, que, de no ser seguro, permitirá que un tercer agente lo manipule. La distribución de software se asegura añadiendo posibilidad de firmado de hashes de la información distribuida
- el método de generación de software libre suele seguir, en la mayoría de los casos, el modelo *bazar*, es decir, muchas personas trabajan sobre partes concretas e integrando sus cambios o personas desde el exterior contribuyen mejoras al proyecto global. Esto puede dar lugar a que se realice una mala gestión del código fuente del software por no seguir métodos formales de seguimiento, la consecuencia final es que falten piezas clave (que nadie ha contribuido) como es el caso de la documentación.
- Al no tener un respaldo directo, la evolución futura de los componentes software no está asegurada o se hace demasiado despacio.

En mayor o menor medida, algunas de estas desventajas están comenzado a soluciones. El caso la difusión de troyanos se limita mediante el uso de técnicas de firma digital para garantizar la inviolabilidad del código o binarios transmitidos. Es frecuente que algunos autores de software libre al distribuir el código indique también información (sumas MD5 firmadas) que permitan garantizar la integridad del código descargado. Asimismo, las distribuciones del sistema operativo, como Debian o RedHat, han incorporado a lo largo del año 2001 soluciones de firma digital para la distribución de código fuente y binario de forma que el usuario pueda garantizar la integridad del mismo tras una descarga.

De igual forma, los problemas de evolución futura empiezan a quedar resueltos con un cambio de paradigma por parte de las compañías de software. Se trata del cambio de un modelo de negocio en el software que pasa a enfocar el negocio orientado a el cobro de la realización de servicios en lugar del cobro a la utilización de productos. Ya se observan, en el mundo de software libre, compañías que contratan a personal cualificado para hacer mejoras sobre proyectos libres para cubrir sus propios intereses y ofrecen soporte de productos de software libre. Estas compañías, a diferencia de la orientación propietaria previamente presentada, siguen haciendo públicas las modificaciones realizadas al código fuente.

- Distribuciones libres para implementar cortafuegos en máquinas dedicadas.

Existen equipos diseñados específicamente para trabajar como cortafuegos. La ventaja fundamental de estos aparatos es que todos sus componentes han sido diseñados con los mismos requisitos de seguridad, al contrario de lo que ocurre con otras soluciones.

Algunos cortafuegos dedicados (o "cajas negras") disponen de circuitos que realizan algunas funciones que de otra forma se harían por software, acelerando enormemente las prestaciones. El cifrado en las redes privadas virtuales es lo que más se beneficia de esto; un router solamente puede hacerlo a velocidades moderadas, mientras algunos cortafuegos dedicados son capes de cifrar un flujo de datos a velocidades de hasta 100 Mbps.

En Linux:

- ClearOS: La distro que combina facilidad de uso con funcionalidad.
- IPCop: Distribución versátil y rápida. Altamente configurable.
- eBox Platform: Algo más que un simple software cortafuegos.
- Monowall: La más liviana de las propuestas de la entrada.
- PfSense: Si desea un servidor de seguridad integral y nada más, no busques más.
- Smoothwall Express: Probablemente la distribución firewall con la mayor reputación.
- Smoothwall Advanced: Y su versión de pago, con asistencia técnica y más opciones.

La ventaja fundamental de algunos de ellos, de cualquier manera, es la sencillez de configuración. Muchos problemas de seguridad se deben a errores provocados por equipos complicados o tediosos de configurar. Cuanto más sencillo resulte, más improbable es cometer errores.

Algunos modelos recientes, además, pueden incorporar un antivirus dentro de la misma unidad, ofreciendo una primera l'nea de defensa cuyo funcionamiento no se ve afectada por los propios virus; algunos de ellos desactivan el software antivirus de los equipos afectados.

- Cortafuegos hardware. Gestión Unificada de Amenazas "Firewall UTM" (Unified Threat Management).

Listado de firewalls hardware o cortafuegos hardware

- **AlphaShield**
  - **AlphaShield Home Edition**
  - **AlphaShield Professional Edition**
- **Clavister**
  - **Clavister Security Gateway 50 series**
  - **Clavister Security Gateway 3100 series**
  - **Clavister Security Gateway 4200 series**
  - **Clavister Security Gateway 4400 series**

Las organizaciones de todos los tamaños están encarando retos relacionados a la seguridad de la información. Las regulaciones gubernamentales, el alto costo de la pérdida de imagen pública cuando se da un ataque, y la creciente complejidad de los nuevos ciber-ataques, son sólo algunos de estos retos.

Las organizaciones actuales confían en entornos informáticos seguros y de alta disponibilidad para realizar para desarrollar sus negocios. Los firewalls y los UTM (Unified Threat Management) son componentes claves de una red segura y deben ser gestionados adecuadamente para asegurarse de que protegen sus activos de información crítica.

Los firewalls y UTMs deben ser configurados para permitir el tráfico "bueno" y para mantener el tráfico "malo" afuera. Los firewalls y UTMs deben ser actualizadas continuamente para apoyar a los requerimientos empresariales cambiantes, tales como:

- Nueva VPN de usuarios
- Cambios en la condición de empleado
- Nuevos socios
- Nuevas aplicaciones

La administración de Firewalls y UTMs es intensiva en recursos y requiere un alto nivel de conocimientos. Debido a la complejidad asociada a estas tareas, la mayoría de las violaciones son causadas por la incorrecta configuración de reglas y políticas de firewall.

Ximark UTM/Firewall Administrado es un servicio de administración de dispositivos de seguridad de perímetro, conocidos también como "Administradores de Amenazas Unificadas" que protegen a las organizaciones con herramientas integrales como: anti-spam, anti-virus, sistema de prevención de intrusos (IPS), filtro de contenido web, control de "peer-to-peer" (P2P) y control de chat, entre otros.

El registro de eventos, la administración de la configuración y los reportes centralizados son fundamentales de acuerdo a las mejores prácticas de seguridad modernas. Con Ximark UTM/Firewall Administrado las organizaciones pequeñas y medianas pueden cumplir con estas prácticas. Con nuestro modelo de seguridad "On-Demand" las empresas de todos los tamaños pueden beneficiarse de una solución centralizada de administración y monitoreo de sus dispositivos de seguridad perimetral de varios fabricantes líderes. No hay requerimientos adicionales de hardware, software o facilidades, lo cual provee un costo competitivo.

Funcionalidades del Servicio

- **Monitoreo**
  - Administración de registros (logs). Revise registros en tiempo real o históricamente. Para diagnóstico o análisis de seguridad.
  - Monitoreo de la disponibilidad del dispositivo. Sea notificado cuando el dispositivo presenta problemas de desempeño o conectividad.
- **Reportes**
  - Servicio "On-demand". Acceda al servicio desde cualquier ubicación vía Internet vía un browser.
  - Reportes pre-configurados. Vea reportes de actividades como los hosts más activos, servicios más usados, sitios más visitados y otra información útil para diagnóstico y control.
- **Administración**
  - Administración de la configuración. La ejecución de cambios programados se incluyen dentro Ximark UTM/Firewall Administrado.
  - Mantenimiento. Las tareas de mantenimiento como actualización de firmware y otras.
  - Actualización de servicios de protección. Los servicios de protección UTM como anti-spam, anti-virus, IPS y filtrado de contenido web se actualizan.
- **Mesa de Servicio**
  - Mesa de servicio vía Web. Puede abrir casos 24x7x365 días al año vía web y por teléfono. Como un sólo punto de contacto para todas sus necesidades de soporte, nuestros ingenieros que atienden vía nuestra plataforma web, tienen experiencia en soportar redes y ayudar a diagnosticar problemas y proveer soluciones. La mesa de servicio permite que Ximark responda tan rápido como sea posible o de acuerdo a los acuerdos de niveles de servicio (SLA) establecidos con el cliente. Para ellos es posible manejar escalamientos y alertas a los gerentes del

área de soporte. La mesa de servicio es basado en Web y en tecnologías de última generación. El sistema posee una base de datos conocimiento, calendario, manejo de SLAs, y otras funcionalidades que permiten brindar un servicio de soporte avanzado.

- **Niveles de Servicio**
  - Ximark UTM/Firewall Administrado ofrece niveles de acuerdo de servicio (SLA) para garantizar la disponibilidad y confiabilidad.

Funcionalidades de Dispositivos UTM

Ximark UTM/Firewall Administrado ofrece la administración y el monitoreo de dispositivos UTM de fábricas líderes de la industria. Estas plataformas proveen funcionalidades de protección unificada contra las principales amenazas que encaran los negocios hoy día. Las características de seguridad son comunes a los fabricantes que Ximark UTM/Firewall Administrado soporta son enunciados a continuación.

- Firewall. Es posible con controlar y bloquear el tráfico de entrada y salida a la red interna o a la zona desmilitarizada (DMZ). El tráfico a controlar se puede definir por servicio (SMTP, HTTP, etc.) y por horarios.
- Red Privada Virtual (VPN). Una VPN es un túnel seguro que se crea entre dos o más UTMs o entre usuarios que se mueven fuera de la empresa. Esta capacidad viene incluida dentro de las funcionalidades del UTM.
- Filtrado de contenido web. Se controla el acceso a sitios que puedan con contenido no deseado como sitios que contengan pornografía o contenido malicioso. Estos sitios se controlan por categorías que definen grupos de sitios según su contenido. Asimismo se establecen acceso a sitios personalizados, en caso de que no hayan sido clasificados en las categorías.
- Prevención contra intrusos (IPS). El IPS provee la capacidad de bloquear ataques contra vulnerabilidades conocidas de aplicaciones, sistemas operativos y hardware.
- Anti-Virus y AntiSpyware. Protección anti-virus que inspecciona el tráfico que entra y sale de la red de los protocolos más comunes como SMTP, POP3, HTTP, FTP y otros.
- Anti-Spam. Distingue comunicaciones por correo electrónico legítimas de aquellas que son spam, bloqueando este último en tiempo real.

Beneficios Claves

- Con Ximark UTM/Firewall Administrado los clientes obtienes los siguientes beneficios:
- Minimiza el riesgo del impacto que podrían tener las violaciones de seguridad.
- Les permite cumplir con regulaciones y certificaciones de la industria.
- Reduce la sobre-carga de administración logrando un mejor uso del recurso humano interno.
- Provee administración consolidada y “todo-en-uno” de la seguridad de la red.