Sistemas de cortafuegos

PID_00191697

Guillermo Navarro Arribas

Universitat Oberta de Catalunya

Índice

Introducción

Hoy en día, uno de los componentes más importantes que nos encontramos en la seguridad en redes informáticas son los sistemas de cortafuegos. Su uso se ha extendido a todo tipo de redes e incluso a ordenadores personales. Un cortafuegos proporciona un mecanismo de control de acceso a la red, mediante el filtrado de paquetes, muy importante. El objetivo principal de un cortafuegos es servir de barrera entre un entorno protegido y un entorno hostil. El entorno protegido suele ser una red interna a una organización (generalmente privada) o un conjunto de equipos informáticos, y el entorno hostil suelen ser redes públicas, como Internet.

Cheswick, Bellovin y Rubin, en su influyente libro *Firewalls and Internet Security. Repelling the Wily Hacker*, listan una serie de máximas sobre la seguridad informática que reproducimos a continuación:

"No existe la seguridad absoluta".

"La seguridad es siempre una cuestión económica".

"Mantén todas tus defensas al mismo nivel".

"Un atacante no atraviesa la seguridad, la rodea".

"Pon tus defensas en capas".

"Es mala idea confiar en la"<seguridad por oscuridad ">".

"Mantenlo simple".

"No des a una persona o programa más privilegios que aquellos estrictamente necesarios".

"La programación es difícil".

"La seguridad debería ser una parte integral de los diseños originales".

"Si no ejecutas un programa, no importa si tiene fallos de seguridad".

"Un programa o protocolo es inseguro hasta que se demuestre lo contrario".

"Una cadena es tan fuerte como su enlace más débil".

"La seguridad conlleva un compromiso con la comodidad".

"No subestimes el valor de tus recursos".

Creemos que son una buena introducción a la seguridad informática y más concretamente al tema que tratamos en este módulo. Muchas de estas frases se consideran principios fundamentales de la seguridad informática, en general, y del diseño y configuración de sistemas cortafuegos en particular.

En este módulo presentamos una introducción a los sistemas de cortafuegos. Nuestra intención es ofrecer un material inicial que os sirva para iniciaros en el tema y que os permita poder adentraros por su cuenta en el complejo mundo

de los cortafuegos. Este módulo asume que tenéis conocimientos básicos sobre el funcionamiento de redes de ordenadores y ciertas nociones sobre seguridad en redes. De manera más precisa, y aunque sin necesidad de conocimientos exhaustivos, sí consideramos que estáis familiarizados con la familia de protocolos TCP/IP.

En este módulo repasamos qué es un sistema cortafuegos. Veremos qué tipos hay y revisaremos las arquitecturas más utilizadas hoy en día. Dedicamos también un apartado introductorio al filtrado de paquetes, con ejemplos de configuraciones típicas. Finalmente repasamos algunos aspectos relacionados con los cortafuegos que os pueden ser de interés.

Objetivos

Los objetivos que se persiguen con el estudio de los materiales de este módulo son los siguientes:

1. Conocer qué es un sistema de cortafuegos y cómo se puede utilizar para proporcionar seguridad a una red informática.

2. Conocer y entender estrategias y arquitecturas diferentes de cortafuegos para la protección de redes de ordenadores.

3. Comprender las políticas de seguridad y el funcionamiento del filtrado de paquetes en sistemas de cortafuegos.

1. Sistemas de cortafuegos

Los **sistemas de cortafuegos**[1] son componentes hardware y/o software que controlan el tráfico de entrada y salida a una red. De esta manera, proporcionan un mecanismo de control de acceso sobre la capa de red, que permite, por ejemplo, separar nuestra red (donde los equipos que intervienen son de confianza) de los equipos situados en redes del exterior (potencialmente hostiles).

(1)En inglés, los sistemas de cortafuegos reciben el nombre de *firewall*.

Un **cortafuegos** es un sistema de red encargado de separar redes informáticas, efectuando un control del tráfico que transcurre entre ellas. Este control consiste, en última instancia, en permitir o denegar el paso de la comunicación de una red a otra mediante el control de los protocolos de red.

Un cortafuegos sirve de barrera en una red, puede bloquear el tráfico de entrada o salida, prevenir accesos no autorizados y, en general, permite implementar la política de seguridad del sistema. En este contexto, el concepto de política de seguridad es importante.

Una **política de seguridad** es el conjunto de reglas y prácticas que definen y regulan los servicios de seguridad de una organización o sistema con el propósito de proteger sus recursos críticos y sensibles. En otras palabras, es la declaración de lo que está permitido y lo que no está permitido hacer.

La política de seguridad es la base de la seguridad de un sistema. En ella se detallan los servicios de seguridad del sistema, se determina qué se puede o no hacer con los recursos del sistema, y quién lo puede hacer, y generalmente se especifica cómo se implementan dichos servicios. La implementación concreta de una política de seguridad se lleva a cabo mediante **mecanismos de seguridad**. La política no tiene por qué ser una declaración formal, a veces se trata de simples directrices sobre la seguridad del sistema en lenguaje informal.

En este sentido, un cortafuegos es un mecanismo de seguridad que permite implementar aquellas reglas de la política de seguridad relativas al control de acceso a nivel de red.

A la hora de instalar y configurar un sistema cortafuegos en una red, debemos tener presente lo siguiente:

1) Todo el tráfico que sale o entra a la red ha de pasar por el cortafuegos. Esto se puede conseguir bloqueando físicamente todo el acceso al interior de la red a través del sistema.

2) Solo el tráfico autorizado, definido en las políticas de seguridad locales del sistema, podrá traspasar el bloqueo.

3) El propio cortafuegos debe estar protegido contra posibles ataques o intrusiones.

Los cortafuegos como los conocemos en la actualidad aparecieron a finales de los ochenta desarrollados por las empresas DEC y AT&T, y en 1991 apareció el primer cortafuegos comercial, el DEC SEAL. Actualmente los cortafuegos son un elemento muy importante no solo en dispositivos de red, sino incluso en ordenadores personales. Existen distintas tecnologías para la implementación de cortafuegos y, sobre todo, existen muchas arquitecturas o formas de configurar cortafuegos en una red. En este módulo veremos algunas de las más relevantes. Es importante señalar que nos centraremos en el uso de cortafuegos en redes TCP/IP, a pesar de que el uso de cortafuegos no exclusivo de estos protocolos concretos.

Lecturas complementarias

El siguiente artículo (disponible en línea) comenta los orígenes de los sistemas cortafuegos:

F. Avolio (1999). "Firewalls and Internet Security, the Second Hundred (Internet) Years". *The Internet Protocol Journal* (vol. 2, núm. 2).

Así mismo, es importante tener siempre presente qué se quiere obtener con un sistema cortafuegos. El uso más común es poder establecer un control sobre el tráfico de entrada y salida de una red a otra. Generalmente, se trata de proteger la red interna de una organización frente a un red hostil, como Internet. En la red interna podemos encontrar desde equipos de ordenadores personales, impresoras, dispositivos móviles (*smartphones*, etc.), o servidores. La presencia de servidores, si ofrecen servicios a la red externa, puede requerir un trato especial en los sistemas de cortafuegos. Nos referimos por ejemplo a servidores web, de correo electrónico, o de compartición de ficheros que la organización quiere ofrecer a Internet. Estos servidores generalmente no son tratados igual que los ordenadores de carácter personal a la hora de diseñar su protección.

Un tipo de equipo importante que se tiene en cuenta en el diseño de sistemas de cortafuegos son los llamados equipos bastión[2].

(2)En inglés, *bastion hosts*.

Un **equipo bastión** es un sistema informático que ha sido fuertemente protegido para soportar ataques desde un lugar hostil (en este caso, Internet) y que suele actuar como punto de contacto entre el interior y el exterior de una red.

El equipo bastión está continuamente expuesto a ataques, por lo que es necesario que esté altamente protegido. Suele proporcionar servicios que, por ejemplo, la organización quiere hacer disponibles al exterior (web, correo electrónico, DNS, etc.), o proporcionar servicios de red críticos (enrutamiento, cortafuegos, etc.).

Aunque, como veremos en los siguientes apartados, los cortafuegos proporcionan muchas medidas de seguridad, hay que tener en cuenta que no son una solución definitiva ni única al problema de la seguridad en redes. Existen muchas amenazas que no se pueden cubrir con sistemas cortafuegos. En este sentido, un aspecto muy importante, como iremos viendo, es que resulta difícil protegerse ante un atacante interno mediante un cortafuegos. El propio cortafuegos, como todo sistema informático, puede presentar vulnerabilidades de día-cero, o ser vulnerable a *malware* y virus del sistema operativo donde se ejecuta. Por otra parte, los sistemas cortafuegos pueden tener cierta mala prensa entre los usuarios de la red, ya que muchas veces estos lo ven como contrapartida a su comodidad o facilidad de uso de servicios de red.

2. Tipos de cortafuegos

Tres de las tecnologías más utilizadas a la hora de construir sistemas cortafuegos son las siguientes:

- Encaminadores con filtrado de paquetes.
- Pasarelas a nivel de circuito.
- Pasarelas a nivel de aplicación.

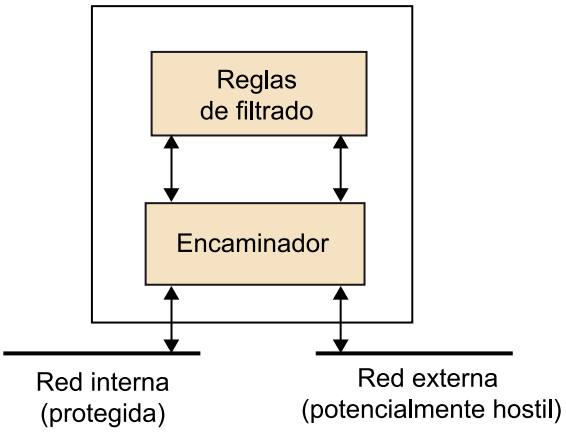
Estos tipos de cortafuegos a veces reciben el nombre de *cortafuegos de primera*, *cortafuegos de segunda* y *cortafuegos de tercera generación* respectivamente, debido al orden en el que han aparecido a lo largo de la historia. De manera muy genérica, podemos decir que su diferencia radica en el nivel al que realizan el filtrado, es decir, a la capa de red sobre la que actúan. Asimismo, hoy en día existen sistemas cortafuegos que pueden combinar el filtrado de paquetes a múltiples capas. Este tipo de filtrado multicapa recibe el nombre de *stateful multi layer inspection*.

2.1. Encaminadores con filtrado de paquetes

Un encaminador con filtrado de paquetes[3] (figura 1) es un dispositivo que encamina el tráfico TCP/IP (encaminador o *router* de TCP/IP) según unas reglas de filtrado que deciden qué paquetes se encaminan a través de él y cuáles se descartan.

[3]En inglés, reciben el nombre de *packet filtering firewall*, y el encaminador que realiza este filtrado suele llamarse *screening router*.

Figura 1. Encaminador con filtrado de paquetes



Las **reglas de filtrado** se encargan de determinar si a un paquete le está permitido pasar de la parte interna de la red a la parte externa, y viceversa.

Estas reglas de filtrado utilizan información presente en los paquetes de red que atraviesan el cortafuegos. Pueden aceptar o denegar paquetes fijándose en las cabeceras de los protocolos (por ejemplo, IP, UDP, TCP, etc.), como:

- Direcciones de origen y de destino.
- Tipos de protocolo e indicadores (*flags*) especiales.
- Puertos de origen y de destino o tipos de mensaje (según el protocolo).
- Contenido de los paquetes.
- Tamaño del paquete.

Cada paquete que llegue al dispositivo será comparado con las reglas de filtrado, comenzando por el principio de la lista de reglas hasta que se encuentre la primera coincidencia. Si existe alguna coincidencia, la acción indicada en la regla se activará.

Por el contrario, si no es posible ninguna coincidencia, será consultada la "política por defecto" para saber qué acción hay que tomar (dejar pasar el paquete, descartarlo, redireccionarlo, etc.). Si se trata, por ejemplo, de una política de denegación por defecto, en el caso de no existir ninguna coincidencia con el paquete, este será descartado.

Una política de denegación por defecto suele ser más costosa de mantener, ya que será necesario que el administrador indique explícitamente todos los servicios que deben permanecer abiertos (los demás, por defecto, serán denegados en su totalidad).

Políticas por defecto

Una política de denegación por defecto también recibe el nombre de *deny all*, o *closed policy*, mientras que la política de aceptación por defecto también se denomina *allow all*, u *open policy*.

En cambio, una política de aceptación por defecto es más sencilla de administrar, pero incrementa el riesgo de recibir ataques contra la red, ya que requiere que el administrador indique explícitamente qué paquetes es necesario descartar (los demás, por defecto, serán aceptados en su totalidad).

En la mayoría de las ocasiones se opta por una política de denegación por defecto como medida de seguridad. Esta estrategia a veces recibe el nombre de *principio de fallo seguro* o *fail-safe*.

Un sistema cortafuegos cumple el principio *fail-safe* si ante un evento no previsto, por ejemplo, un paquete relativo a un nuevo servicio, este es rechazado.

2.2. Pasarelas a nivel de circuito

Las pasarelas a nivel de circuito no encaminan paquetes a nivel de enlace de red, sino que actúan como retransmisores (o *relays*) a nivel de transporte. Estos dispositivos pueden retener paquetes de red hasta obtener información suficiente sobre el estado de la comunicación y tomar la decisión de permitir o no la conexión o el envío de datos. Por ello se dice que realizan un filtrado con estado o *stateful packet inspection*.

Ved también

Sobre el filtrado con estado podéis ver el subapartado 4.2 de este módulo.

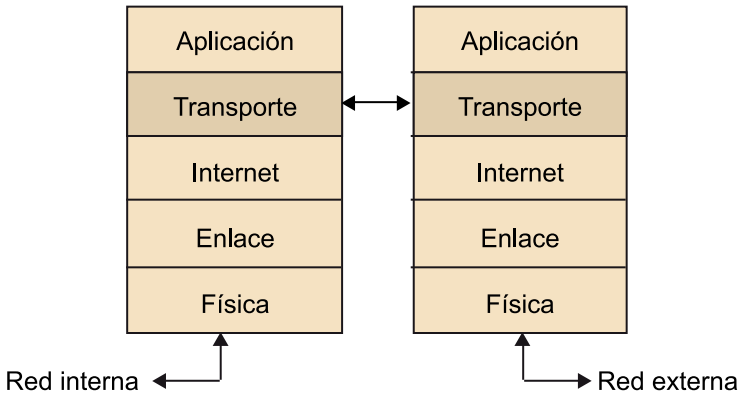
Una **pasarela a nivel de circuito** es un dispositivo que hace de pasarela a nivel de capa de transporte entre dos extremos. Establece una conexión con cada uno retransmitiendo los datos entre las dos conexiones.

Una pasarela a nivel de circuito actúa en la capa de transporte de TCP/IP, o capa 4 del modelo OSI, como se muestra en la figura 2. No se inspecciona el contenido de los paquetes a nivel de aplicación, lo que hace que sean sistemas más eficientes que las pasarelas a nivel de aplicación, aunque no tanto como los encaminadores con filtrado de paquetes.

Ved también

Las pasarelas a nivel de aplicación se estudian en el subapartado 2.3 de este módulo. Los encaminadores con filtrado de paquetes se estudian en el subpartado 2.1.

Figura 2. Pasarela a nivel de circuito



Este tipo de pasarelas se suele utilizar para crear conexiones entre redes aisladas. Las conexiones se pueden establecer de manera automática para determinados servicios TCP o utilizando protocolos específicos. Como ejemplo del segundo caso nos encontramos con el protocolo SOCKS. SOCKS (*SOCKet Secure*), permite el uso de TCP y UDP, y consta de un cliente y un servidor. El servidor SOCKS se ejecuta en la pasarela y el cliente en los *hosts* internos de la red. Así, los clientes de protocolos de aplicación suelen incorporar soporte para SOCKS.

En el caso de TCP, por ejemplo, el cliente establece una conexión con el servidor SOCKS situado en el cortafuegos (pasarela a nivel de circuito en este caso). El servidor autentica al cliente, evalúa la petición de conexión y, si esta se permite, establece una conexión con el servidor externo haciendo de *relay*

SOCKS

La versión 5 de SOCKS está definida en el RFC 1928 y es actualmente considerada como un estándar *de facto* para la implementación de pasarelas a nivel de circuito.

(retransmisor) entre el cliente y el servidor a nivel de TCP. De esta manera, las pasarelas a nivel de circuito también ocultan a nivel de IP los clientes frente al servidor externo.

2.3. Pasarelas a nivel de aplicación

Una pasarela a nivel de aplicación, conocida también como servidor *proxy* (*intermediario*, en inglés), actúa como retransmisor (o *relay*) a nivel de aplicación. Los usuarios de la red contactarán con el servidor intermediario, que a su vez estará ofreciendo un servicio *proxy* asociado a una o más aplicaciones determinadas.

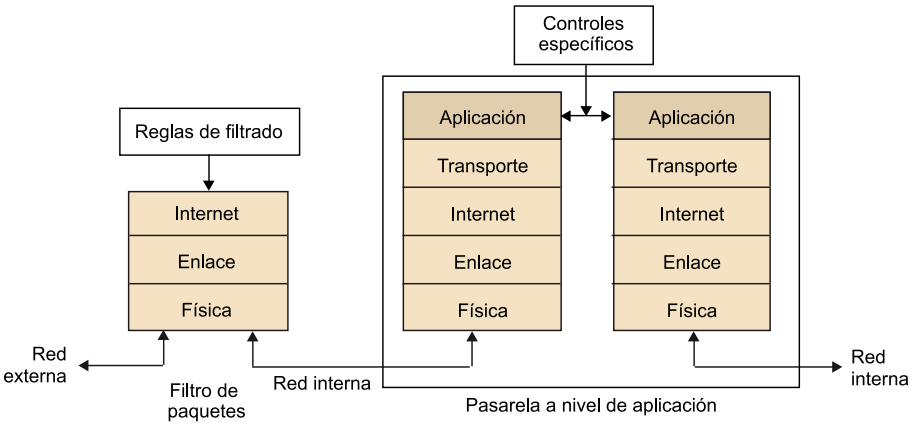
> El servicio *proxy* se encargará de realizar las conexiones solicitadas con el exterior y, cuando reciba una respuesta, se encargará de retransmitirla al equipo que había iniciado la conexión. Así, el servicio *proxy* ejecutado en la pasarela aplicará las normas para decidir si se acepta o se rechaza una petición de conexión.

Del mismo modo que las pasarelas a nivel de circuito, separa completamente el interior del exterior de la red. Pero al contrario que las primeras, lo hace a nivel de la capa de aplicación, ofreciendo únicamente un conjunto de servicios a nivel de aplicación. Esto permite la autenticación de los usuarios que realizan peticiones de conexión y el análisis de conexiones a nivel de aplicación.

Las pasarelas ofrecen mayor seguridad respecto a los filtros de paquetes, con lo que presentan un rango de posibilidades muy elevado. Por el contrario, la penalización introducida por estos dispositivos es mucho mayor. Las pasarelas a nivel de aplicación necesitan realizar una inspección de paquetes detallada, que en inglés suele recibir el nombre de *deep packet inspection*. En el caso de una gran carga de tráfico en la red, el rendimiento puede llegar a reducirse drásticamente. Una manera de mejorar el rendimiento es mediante los sistemas *porxy cache*, que mantienen una copia local de datos recibidos.

En la práctica, las pasarelas y los dispositivos de red con filtrado de paquetes son complementarios. Estos dos sistemas se pueden combinar, proporcionando más seguridad y flexibilidad que si se utilizara solamente uno, como se muestra en la figura 3.

Figura 3. Filtrado de paquetes y pasarela a nivel de aplicación



El uso de las pasarelas proporciona varios beneficios. De entrada, una pasarela podría permitir el acceso únicamente a aquellos servicios para los que hay un servidor *proxy* habilitado. Así, si una pasarela contiene servicios intermediarios únicamente para los servicios HTTP y DNS, entonces solo HTTP y DNS estarán permitidos en la red interna. El resto de los servicios serían completamente rechazados.

Otro beneficio del uso de pasarelas a nivel de aplicación es que el protocolo de aplicación también se puede filtrar, con lo que se prohíbe así el uso de distintos subservicios dentro de un mismo servicio permitido. Por ejemplo, mediante una pasarela que filtrara conexiones FTP, sería posible prohibir únicamente el uso del comando PUT de FTP, dejando habilitado el resto de los comandos. Esta característica no sería posible haciendo uso únicamente de filtros de paquetes.

Aun obteniendo más control global sobre los servicios vigilados, las pasarelas también presentan algunas problemáticas. Uno de los primeros inconvenientes que hay que destacar es la necesidad de tener que configurar un servidor *proxy* para cada servicio de la red que se debe vigilar (HTTP, DNS, SSH, FTP, etc.). Además, en el caso de protocolos cliente-servidor, como, por ejemplo, FTP, pueden llegar a ser necesarios algunos pasos adicionales para conectar el punto final de la comunicación.

3. Arquitecturas de cortafuegos

Un aspecto muy importante a la hora de diseñar un sistema de cortafuegos para proteger una red es decidir la estrategia o arquitectura del sistema. Hay que decidir, aparte del tipo de cortafuegos que se usa, dónde se sitúa el cortafuegos en la red y cómo este proporciona la seguridad perimetral de la red.

En este aspecto existen numerosas arquitecturas y topologías de red, así como muchas estrategias. Por lo general, cómo y dónde se instala un cortafuegos dependerá mucho del tipo de red que queremos proteger y del tipo de protección que queremos aportar. Actualmente no existe una clasificación consensuada pero sí algunas estrategias comunes que revisaremos a continuación.

Para facilitar la explicación, dividiremos las arquitecturas en dos tipos:

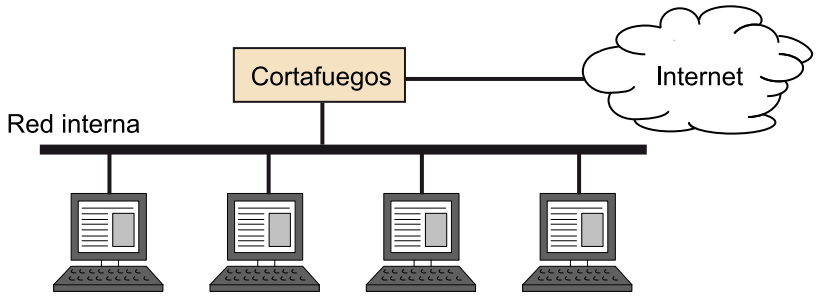
- Arquitecturas de un solo punto, que también son conocidas en inglés como *single-box*.
- Arquitecturas de red perimetral, *screened subnet*, o subred filtrada.

Por lo general, el objetivo a la hora de implantar un sistema de cortafuegos es proteger una red interna del exterior, que generalmente será Internet. Sin embargo, existen otros casos, como el uso de cortafuegos para separar partes de una misma red interna, como por ejemplo las estaciones de trabajo de un laboratorio de pruebas.

3.1. Arquitecturas de un solo punto

Este tipo de arquitecturas es el más sencillo, ya que consiste en separar la red que se quiere proteger del exterior por un solo dispositivo cortafuegos (figura 4). Dependiendo de qué dispositivo haga de cortafuegos, podemos obtener diferentes soluciones.

Figura 4. Arquitectura *single-box*



En general, estas arquitecturas presentan un solo punto de configuración, lo que las hace más sencillas de implantar y administrar, aunque al mismo tiempo este punto se convierte en un punto crítico del sistema. Si un atacante consigue comprometer cualquiera de los servidores que se encuentre detrás de este punto único, las otras máquinas podrán ser atacadas sin ninguna restricción desde el equipo que acaba de ser comprometido.

3.1.1. Encaminador con filtrado de paquetes

Esta situación es posiblemente la más sencilla, donde un encaminador con filtrado de paquetes o *screening router* separa la red interna del exterior. Es decir, se realiza un filtrado de paquetes de red en un solo punto. Generalmente es una solución de bajo coste y su implantación es sencilla.

Ved también

El filtrado de paquetes se estudia en el subapartado 2.1 de este módulo.

Puede ser adecuada en situaciones donde se requiere eficiencia y se considera que los equipos de la red ya disponen de un nivel considerable de protección. Es decir, donde no se requiere un filtrado muy sofisticado.

3.1.2. Cortafuegos *dual-homed*

En este caso, el punto que separa la red interna de la externa es un equipo *dual-homed*. Un equipo *dual-homed* es un equipo con al menos dos interfaces de red, cada una asociada a una red, que puede actuar como encaminador entre las redes. Los sistemas de cortafuegos *dual-homed* utilizan este tipo de equipos, aunque no como simples encaminadores.

Una arquitectura de cortafuegos *dual-homed* se construye mediante el uso de un equipo *dual-homed* con la capacidad de encaminamiento desactivada. De esta manera, los paquetes IP de un extremo de la red (la parte hostil) no serán encaminados hacia la parte protegida, y viceversa, a no ser que se indique lo contrario.

Mediante esta arquitectura, los equipos de la red interna y de la red externa se pueden comunicar con el equipo *dual-homed*, pero no entre ellos. Los equipos de la red interna y externa no se pueden poner en comunicación directamente, sino que un servidor intermediario se encarga de realizar las conexiones en nombre de estas dos partes.

Los cortafuegos *dual-homed* proporcionan un nivel alto de control sobre el tráfico que entra y sale de la red, ya que los paquetes externos no entran en la red interna. El equipo *dual-homed* hace de pasarela (*proxy*).

Una de las principales desventajas de esta arquitectura es que los equipos *dual-homed* no son muy eficientes. Si el tráfico es elevado, pueden resultar ineficientes.

También es posible combinar el filtrado a nivel de paquetes en este tipo de arquitecturas, posibilitando así filtrado a varios niveles: Internet, transporte, o aplicación.

Ved también

Sobre el filtrado a nivel de Internet, de transporte o de aplicación podéis ver respectivamente los subapartados 2.1, 2.2 y 2.3 de este módulo.

Es muy común implementar la arquitectura *dual-homed* mediante un equipo bastión.

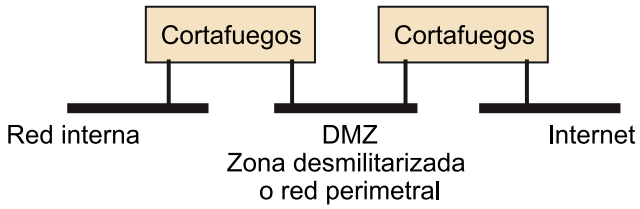
Ved también

El equipo bastión se estudia en el apartado 1 de este módulo.

3.2. Arquitecturas con redes perimetrales

Para añadir un nivel adicional de seguridad a las arquitecturas de un solo punto, encontramos las arquitecturas conocidas como *screened subnet*, *red filtrada* o *red perimetral*. En este caso la idea es añadir una **subred** entre la red interna y la exterior que actúe de barrera ante posibles ataques e intrusiones. Esta red perimetral suele recibir el nombre de **zona desmilitarizada** o **DMZ** (*demilitarized zone*).

Figura 5. DMZ o red perimetral



DMZ y red perimetral

No está clara la diferencia entre DMZ y red perimetral y en muchos casos se usan los dos términos indistintamente. En ocasiones, se considera la DMZ como el conjunto de redes perimetrales del sistema de cortafuegos. Es decir, una DMZ puede estar formada por una o más redes perimetrales.

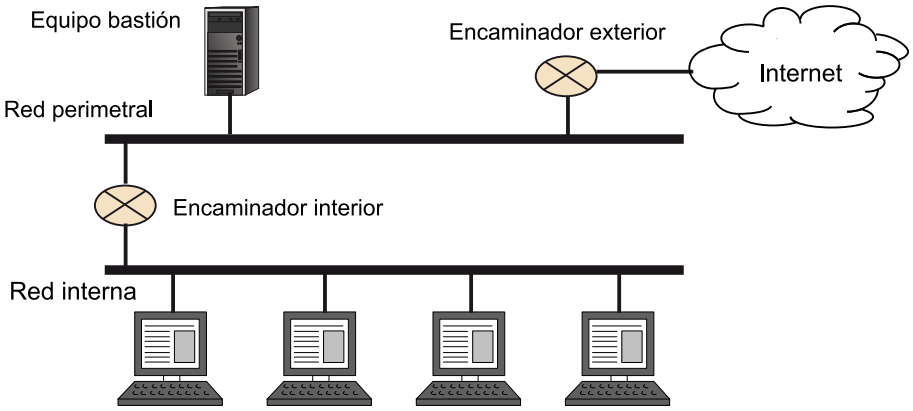
Esta red perimetral o DMZ generalmente alberga equipos bastión, lo que les proporciona un nivel de seguridad adicional, y una separación de la red interna. Estos equipos suelen tener servidores públicos que tienen que ser accesibles desde el exterior. En caso de que un atacante consiga burlar la seguridad del primer cortafuegos (o cortafuegos exterior) e introducirse en un servidor de la DMZ, no podrá atacar inmediatamente a equipos situados en la red interna, ya que están protegidos por el segundo cortafuegos (o cortafuegos interior).

En la figura 6 vemos con más detalle una arquitectura de red perimetral construida mediante encaminadores con filtrado de paquetes. La red perimetral de la figura contiene un equipo bastión, aunque podría haber varios. Aparte de proporcionar servicios al exterior, estos también se pueden destinar a tareas de filtrado a nivel de aplicación o circuito.

Ved también

Los encaminadores con filtrado de paquetes se estudian en el subapartado 2.1 de este módulo.

Figura 6. Ejemplo de arquitectura con red perimetral



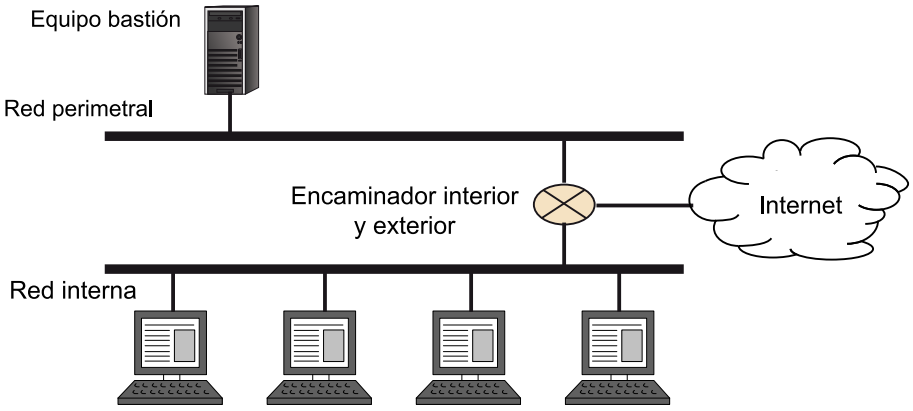
La función que desempeña cada uno de los dos encaminadores es diferente:

- **Encaminador interior** (o ***choke router***). Protege la red interna de Internet y de la red perimetral. Realiza la mayor parte del filtrado de salida y de entrada a la red interior, respecto al exterior. Asimismo controla el tráfico entre la red interior y el equipo bastión. Generalmente el tráfico entre la red interna y el equipo bastión es extremadamente limitado para evitar que el compromiso de un bastión comporte la posibilidad de atacar la red interna.

- **Encaminador exterior** (o ***access router***). Protege la red interna y la perimetral del exterior. Son menos restrictivos que los internos, y sus reglas de filtrado están especialmente pensadas para proteger el equipo bastión del exterior. En ocasiones, el encaminador externo puede estar controlado por una organización externa (por ejemplo, el proveedor de servicios de Internet).

En caso de disponer de un encaminador con filtrado de paquetes adecuado, se puede utilizar dicho encaminador para realizar las tareas de encaminador interior y exterior, como muestra la figura 7, con lo que se simplifica así el sistema.

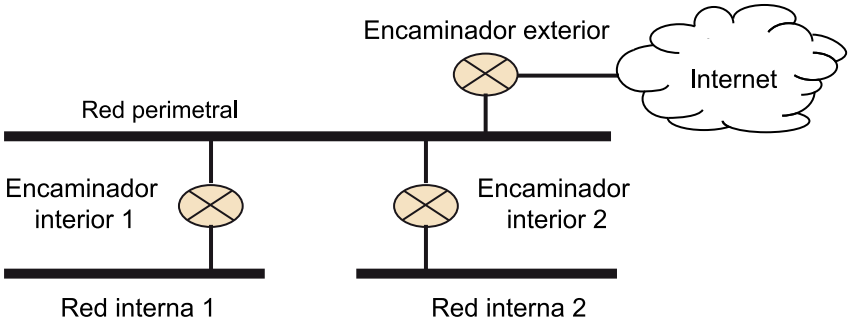
Figura 7. Ejemplo de arquitectura con red perimetral y un solo encaminador



En general, se pueden combinar en estas arquitecturas el uso de encaminadores con filtrado de paquetes, con pasarelas a nivel de aplicación o circuito según se considere oportuno. Por ejemplo, se puede sustituir el encaminador exterior por un equipo *dual-homed*, concretamente el equipo bastión de la red perimetral puede hacer de encaminador exterior con el propósito de simplificar y reducir los costes del sistema. Aunque el equipo bastión queda más expuesto, no se introducen más vulnerabilidades de manera significativa. Sin embargo, no se recomienda utilizar el equipo bastión como encaminador interno, ya que precisamente una función importante de este encaminador es proteger la red interna en caso de que el bastión sea comprometido.

En el caso de que tengamos dos redes internas independientes, estas pueden compartir la misma red perimetral o DMZ, como se muestra en la figura 8, mediante dos encaminadores internos.

Figura 8. Ejemplo de arquitectura con red perimetral y dos redes internas



Es importante remarcar que no se suele recomendar el uso de varios encaminadores internos para una sola red interna. Esto implica mayor carga administrativa y requiere una cuidadosa configuración de los encaminadores. Podría ser una solución aceptable si el objetivo es proporcionar redundancia; aun así, suele ser también desaconsejado, ya que generalmente se recomienda proporcionar redundancia también a nivel de red perimetral.

Ved también

La redundancia a nivel de red perimetral se trata en el subapartado 3.2.1 de este módulo.

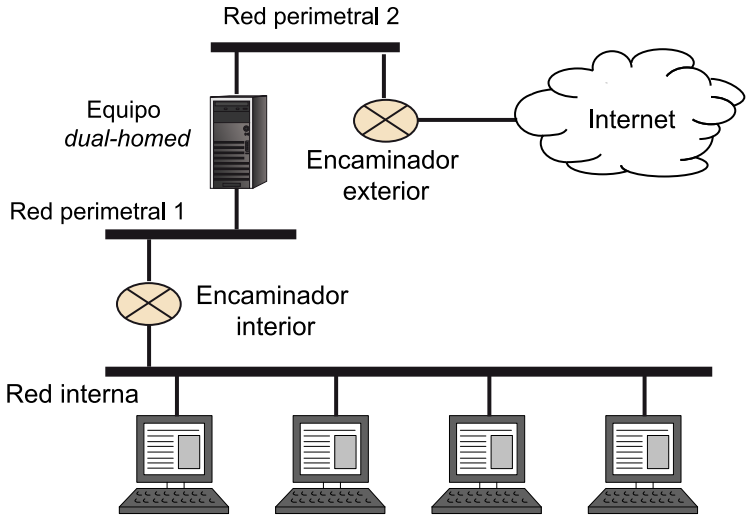
Por otra parte, el uso de varios encaminadores externos puede tener sentido en caso de querer introducir redundancia (no implica tantos problemas como el caso anterior de varios encaminadores internos), o simplemente porque se quiere dar salida o entrada a dos redes exteriores diferentes.

3.2.1. Múltiples redes perimetrales

Aunque el esquema mostrado en la figura 6 es bastante común, existen arquitecturas basadas en redes perimetrales más sofisticadas. Un ejemplo de esto es el uso de un equipo *dual-homed* que divide la red perimetral o DMZ como se muestra en la figura 9. Este tipo de arquitectura, también conocida como *split-screened subnet* o *belt-and-suspenders firewall*, busca proporcionar mayor segu-

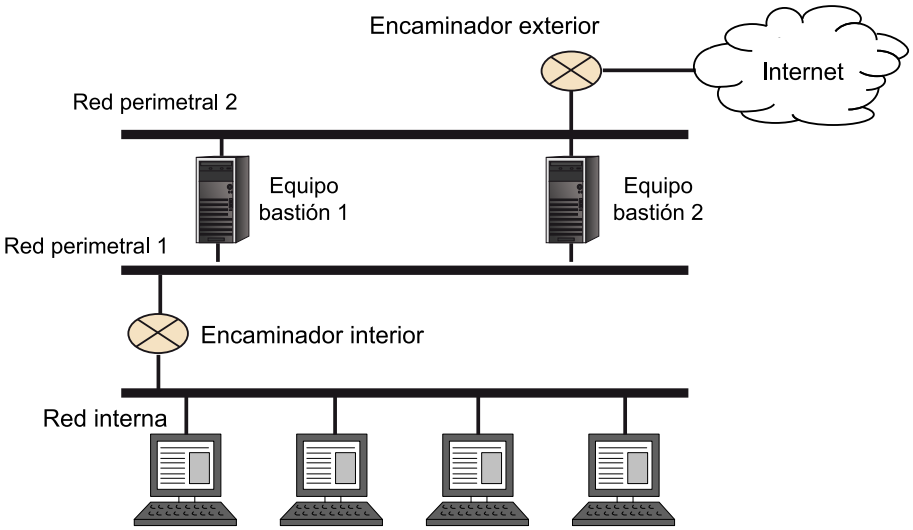
ridad y capacidad de defensa. Los encaminadores protegen el equipo *dual-homed* que realiza funciones de *proxy*. Proporciona un nivel de seguridad muy alto pero requiere una configuración más detallada y compleja.

Figura 9. Ejemplo de arquitectura de red perimetral dividida



Un uso típico de este esquema es el de poder controlar el acceso administrativo a los servicios alojados en equipos bastión de la DMZ. Para ello, se separa el tráfico administrativo (originado en la red interna) del tráfico externo como se muestra en la figura 10.

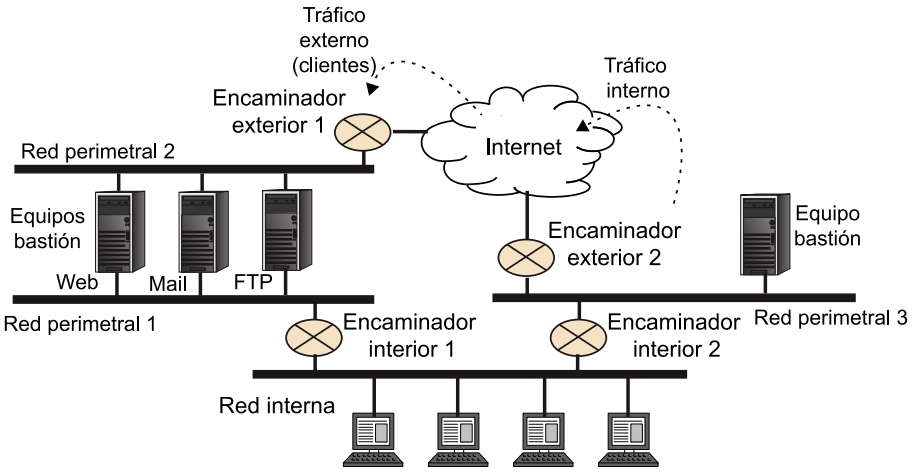
Figura 10. Ejemplo de arquitectura de red perimetral dividida con equipos bastión



También existen arquitecturas donde las diferentes redes perimetrales son totalmente independientes. Es decir, múltiples encaminadores externos e internos que separan por completo las redes perimetrales. Este tipo de arquitecturas se suele aplicar para proporcionar redundancia a través de dos accesos diferentes a Internet. También se puede usar como medida de privacidad separando tráfico entre una red perimetral u otra en función de su grado de confidencialidad. O para separar el tráfico de entrada a servidores de la organización del tráfico de salida de la propia organización. Este tipo de arquitecturas más

elaboradas suele proporcionar un mayor nivel de protección pero tienen el inconveniente de que su administración es mucho más compleja. En la figura 11 vemos un ejemplo de arquitectura compleja con múltiples redes perimetrales.

Figura 11. Ejemplo de arquitectura compleja de redes perimetrales



Basado en: Zwicky, E. D.; Cooper, S.; Chapman, D. B. (2000). *Building Internet Firewalls*. 2nd ed. O'Reilly Media.

4. Mecanismos y reglas de filtrado

Uno de los aspectos más importantes de un sistema cortafuegos es cómo se consigue implementar la política del sistema u organización mediante reglas concretas de filtrado. La administración y configuración de cortafuegos es a veces compleja y considerada como un arte en sí mismo. En este apartado mostraremos técnicas de filtrado genéricas. Hay que tener en cuenta que cada sistema de cortafuegos utiliza unas reglas diferentes, con formato diferente y que pueden ser aplicadas de maneras diferentes.

Consideramos el filtrado de paquetes como un mecanismo de seguridad en redes que tiene el objetivo de controlar el flujo de datos de entrada y salida a una red.

Un buen mecanismo de filtrado de paquetes puede permitir un control muy detallado y sofisticado sobre los datos que se transmiten por la red. Sin embargo, es importante ser conscientes del nivel de dificultad. Hay tareas mucho más sencillas que otras, es decir, que pueden ser realizadas de manera más eficiente y con un coste más bajo que otras. Por ejemplo, una regla que necesite información sobre el funcionamiento de un protocolo concreto será más compleja que una que no necesite dicha información. Asimismo, si necesitamos inspeccionar datos propios de la aplicación, estamos aumentando la complejidad. Estos dos últimos casos suelen ser realizados por pasarelas, mientras que los encaminadores con filtrado de paquetes realizan un filtrado más sencillo, y por tanto muy eficiente.

4.1. Filtrado de paquetes básico

El filtrado básico de paquetes se efectúa a partir de información disponible en las cabeceras del protocolo IP y protocolos de transporte, y se trata de manera independiente en cada paquete. Esta información suele ser relativamente reducida:

- Dirección IP de origen y destino.
- Puerto de origen y destino de la capa de transporte.
- Protocolo: este campo hace referencia al campo *Protocol* de la cabecera del datagrama IPv4 o *Next Header* en el caso de IPv6.

Next Header

En IPv6 *Next Header* identifica la extensión de cabecera siguiente. En caso de que no haya extensiones o se trate de la última extensión, el campo indica el protocolo encapsulado por el datagrama, utilizando el mismo identificador que IPv4.

Este tipo de filtrado permite especificar reglas del tipo: "Permitir todo el tráfico de salida destinado al puerto 80 (HTTP)", pero no permite expresar reglas del tipo: "Permitir todo el tráfico HTTP únicamente si no se está utilizando para descargar archivos de MS Word".

Cada regla de filtrado tiene asociada una *acción* o *target*, que determina qué se hace con el paquete que cumple con la regla. Las principales acciones que se pueden realizar con el paquete son: *aceptar* (*ACCEPT*) el paquete y, por tanto, permitir su paso por el cortafuegos, o bien *rechazar* (*DENY*) el paquete y descartarlo. Pueden existir acciones adicionales, como por ejemplo la acción de *LOG*, que hace que el cortafuegos guarde un log sobre el paquete, o la que se discute en el siguiente subapartado. Acciones de tipo *ACCEPT* o *DENY* provocan que se realice la acción sobre el paquete y se acaba el proceso de filtrado; sin embargo, acciones como *LOG* generan la acción pero por lo general se siguen procesando reglas de filtrado.

4.1.1. Cómo se rechaza un paquete

En el caso de rechazar un paquete, existe la posibilidad de generar un mensaje de error ICMP para notificar al origen que se ha descartado dicho paquete. Generalmente se trata de mensajes ICMP tipo 3 (*Destination Unreachable*) con los códigos que se muestran en la tabla 1. Es común llamar *DROP* a la acción de descartar un paquete y *REJECT* cuando se descarta y se genera el ICMP de error para diferenciarlas.

Tabla 1. Códigos de error ICMP que suelen enviar loscortafuegos al descartar un paquete

Tipo	Código	Descripción
3	0	*Destination network unreachable*
3	1	*Destination host unreachable*
3	9	*Network administratively prohibited*
3	10	*Host administratively prohibited*

Los códigos 9 y 10 fueron especialmente añadidos a la especificación de ICMP para ser utilizados por sistemas de filtrado. Aun así, muchos de estos sistemas siguen utilizando los códigos 0 y 1, que fueron inicialmente pensados con otros propósitos.

Generar o no un mensaje de error ICMP cuando se descarta un paquete tiene ventajas e inconvenientes:

- El hecho de enviar el mensaje ICMP de error hace que el origen pueda cerrar la conexión de manera inmediata, sin necesidad de esperar a ningún timeout, y sin intentar ninguna retransmisión.

- El mensaje ICMP concreto que se envía puede tener diferentes interpretaciones por parte de los equipos de origen que lo reciban.

- La generación de estos mensajes puede suponer una penalización en el rendimiento del cortafuegos.

- El hecho de enviar estos mensajes puede permitir a atacantes potenciales obtener información sobre el filtrado de paquetes.

Generalmente, se considera más seguro no enviar este tipo de mensajes de error, pero puede haber casos en los que sea conveniente hacerlo. Algunos sistemas de cortafuegos pueden incorporar mecanismos adicionales. Por ejemplo, permitir cerrar conexiones TCP de manera inmediata respondiendo con un *reset* de TCP.

4.1.2. Organización de las reglas de filtrado

El orden en el que se miran las reglas de filtrado es muy importante, ya que es el mecanismo principal para la resolución de conflictos. Es decir, si hay dos reglas contradictorias, la primera que se consulte será la que se ejecutará.

Por lo general, es el administrador quien decide el orden de las reglas, que suele ser el orden en el que se introducen. Sin embargo, algunos cortafuegos pueden modificar este orden para mejorar su eficiencia.

En muchos casos las reglas se agrupan por tipos o tablas, y existe un orden predefinido entre los tipos de reglas. Por ejemplo, el cortafuegos *Windows Firewall with Advanced Security* agrupa las reglas en los 6 tipos que se muestran en la tabla 2.

Tabla 2. Tipos de reglas de filtrado de Window Firewall with Advanced Security

Orden	Tipo	Descripción
1	*Windows Service Hardening*	Impiden que los servicios establezcan conexiones para las que no fueron diseñados.
2	*Windows Service Hardening*	Definen la autenticación con IPSec.
3	*Authenticated bypass rules*	Permiten que cierto tráfico se salte restricciones (reglas) que lo bloquee si ha sido autenticado con IPSec.
4	*Block rules*	Reglas que bloquean tráfico.
5	*Allow rules*	Reglas que permiten tráfico.
6	*Default rules*	Acciones por defecto.

Windows Firewall

Windows Firewall es el cortafuegos que incorporan los sistemas operativos Microsoft Windows (Windows 7, Windows Server 2008, Windows Server 2008 R2, Windows Vista). Anteriormente recibía el nombre de ICS (*Internet Connection Firewall*).

Este cortafuegos, además de ofrecer un filtrado de paquetes como el que vemos en este módulo, permite gestionar IPSec. Como vemos, todas las reglas relacionadas con IPSec tienen precedencia sobre el resto. Es interesante ver en este caso cómo se separan la reglas que bloquean tráfico de las que lo permiten. Esto quiere decir que el tráfico que coincida con dos reglas contradictorias (una que lo permite y otra que lo rechaza) será rechazado. El hecho que de las reglas que rechazan tráfico tengan precedencia se puede ver como una medida de seguridad: en caso de duda, no se permite el tráfico en cuestión.

Otro ejemplo de sistema de cortafuegos muy utilizado hoy en día es Netfilter/iptables, que forma parte del kernel de Linux. *Iptables* permite configurar el cortafuegos desde la línea de comandos. Netfilter/iptables utiliza la tabla *filter* para realizar filtrado de paquetes, donde las reglas se organizan en cadenas. Existen cadenas predefinidas y el administrador puede definir cadenas nuevas si lo considera conveniente. Las cadenas predefinidas para el filtrado de paquetes son:

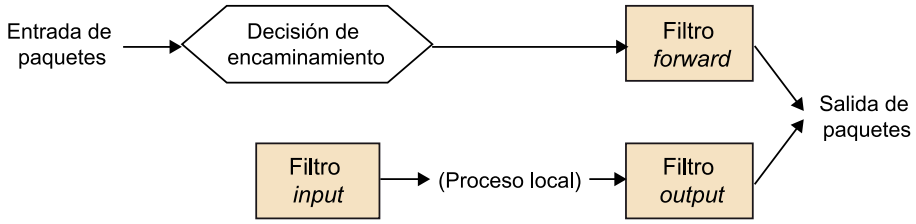
Nota

Netfilter/iptables dispone de más tablas para hacer otras tareas como NAT.

- *INPUT*: afecta a paquetes destinados al sistema local.
- *FORWARD*: paquetes que pasan a través del dispositivo (son encaminados por el sistema).
- *OUTPUT*: para paquetes generados por el propio sistema.

El orden de estas cadenas, bastante intuitivo, viene determinado por la decisión de encaminamiento del paquete (si el destino es el sistema local o no), como se muestra en la figura 12.

Figura 12. Cadenas de la tabla de filtrado de Netfilter/iptables



Cada cadena predefinida tiene una acción por defecto que especifica el administrador. Dentro de cada cadena el orden en el que se revisan las reglas es el que determina el administrador del sistema al introducir las reglas. De aquí viene el nombre de cadena, ya que una cadena no es más que un conjunto secuencial de reglas.

4.2. Filtrado dinámico

El filtrado dinámico o *stateful filtering* utiliza información sobre el estado de la conexión o sesión a la hora de filtrar paquetes. Permite asociar paquetes a sesiones concretas y estados de protocolos. En este sentido, el cortafuegos tiene que seguir el estado de las transacciones de paquetes y su comportamiento puede variar en función del tráfico que va pasando.

Ejemplo de uso de filtrado dinámico

Este tipo de filtrado permite establecer reglas como: "Permitir la entrada de mensajes UDP únicamente si se reciben como respuesta a una petición UDP originada en la red interna". Por ejemplo, la tabla 3 muestra tres paquetes UDP que podrían pasar por el cortafuegos. Consideramos que nuestra red interna es la 230.0.113.0/24. Siguiendo la regla descrita anteriormente, los paquetes 1 y 2 serían aceptados, mientras que el paquete 3 no. Este paquete está intentando entrar en nuestra red y, al contrario que el paquete 2, no se corresponde a una respuesta de un paquete originado en nuestra red, y puede tratarse de un paquete falso.

Tabla 3. Ejemplo de paquetes UDP

1	IP origen	230.0.113.1	IP destino	192.0.2.1
	P. origen	43321	P. destino	7
2	IP origen	192.0.2.1	IP destino	230.0.113.1
	P. origen	7	P. destino	43321
3	IP origen	192.0.2.1	IP destino	230.0.113.1
	P. origen	7	P. destino	34511

Hay que tener en cuenta que este tipo de filtrado no es perfecto y puede ser vulnerable a ataques de *IP spoofing*, donde se falsea la dirección IP y puerto de origen para que parezca que el paquete es la respuesta esperada.

El principal problema de este tipo de filtrado es su eficiencia. El cortafuegos necesita recursos tanto de memoria como de procesamiento para mantener el estado del tráfico que ve. Esto no solo supone una carga importante, sino que abre la posibilidad de recibir ataques de denegación de servicio.

El ejemplo anterior es muy simplista, y actualmente existen sistemas de cortafuegos dinámicos muy elaborados que permiten realizar muchas comprobaciones a todos los niveles de la pila de protocolos red y datos de aplicación.

Por ejemplo, se pueden utilizar técnicas de **comprobación de protocolo**. Un paquete UDP destinado al puerto 53 se suele considerar como una petición a un servidor DNS, pero podría ser un intento de enmascarar un paquete malicioso. La comprobación de protocolos permite al cortafuegos inspeccionar el paquete y ver si realmente se trata de una petición de DNS. Es decir, comprueba que el datagrama UDP contenga un mensaje de petición DNS con la cabecera e información correspondiente.

Mecanismos más avanzados permiten por ejemplo inspeccionar los datos del protocolo de aplicación. No es raro hoy en día disponer de pasarelas a nivel de aplicación que pueden filtrar contenidos de páginas web o conexiones FTP según el nombre de usuario.

En algunas ocasiones, la calificación de filtrado dinámico o con estado no está clara. Por ejemplo, algunos cortafuegos permiten establecer reglas de filtrado de tráfico TCP en función de los *flags* activos en el segmento TCP, como por ejemplo SYN o ACK. Esto permite distinguir si el paquete es el inicio de conexión TCP (no tiene el ACK activo) o forma parte de una conexión ya existente (tiene activo el ACK). Sin embargo, este tipo de comprobación se realiza a partir de cada segmento TCP de manera independiente. Es decir, el cortafuegos no necesita mantener el estado de la conexión TCP. Por este motivo, este tipo de filtrado no suele considerarse como filtrado con estado o dinámico.

4.3. Ejemplos de filtrado

A continuación veremos dos ejemplos[4] de reglas de filtrado para dos escenarios diferentes. El primer ejemplo considera una arquitectura de un solo punto, mientras que el segundo corresponde a una arquitectura de red perimetral. En ambos casos de trata de filtrado básico sin estado. El filtrado con estado no es tan genérico, y depende mucho del producto concreto que lo implementa.

(4)Estos ejemplos están lejanamente inspirados en la obra de Zwicky y otros (2000).

Ved también

La arquitectura de un solo punto se estudia en el subapartado 3.1.1 de este módulo, mientras que la arquitectura de red perimetral se estudia en el subapartado 3.2.

4.3.1. Ejemplo 1: cortafuegos de un solo punto

Como primer ejemplo podemos pensar en la red que se muestra en la figura 13. En ella se determina la siguiente política:

1) Se permite que los usuarios de la red interna puedan acceder a cualquier servicio TCP de Internet.

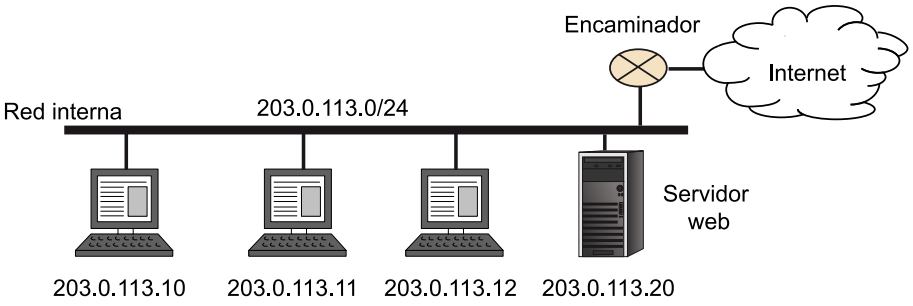
2) Se permite a los usuarios de la red conexiones UDP al exterior pero solo para realizar peticiones DNS.

3) Se permite tráfico ICMP de salida (originado en la red interna).

4) Desde el exterior solo se puede acceder al servidor web, no al resto de los equipos.

5) El resto de tráfico por defecto no se permite.

Figura 13. Ejemplo de filtrado



En este caso contamos con un encaminador con filtrado de paquetes. La información que puede obtener dicho encaminador para determinar las reglas de filtrado es: direcciones IP de origen y destino, puertos de origen y destino, y protocolo.

Si consideramos un filtrado de paquetes básico, podemos añadir las reglas que se muestran en la tabla 4.

Tabla 4. Reglas de ejemplo para el escenario de la figura 13

Regla	Acción	IP origen	P. origen	IP destino	P. destino	Protocolo
1	*ACCEPT*	203.0.113.0/24	> 1023	*	53	UDP
2	*ACCEPT*	*	53	203.0.113.0/24	> 1023	UDP
3	*ACCEPT*	203.0.113.0/24	> 1023	*	*	TCP
4	*ACCEPT*	*	*	203.0.113.0/24	> 1023	TCP
5	*ACCEPT*	203.0.113.0/24	-	*	-	ICMP
6	*ACCEPT*	*	-	203.0.113.0/24	-	ZICMP
7	ACCEPT	*	> 1023	203.0.113.20	80	TCP
8	*ACCEPT*	203.0.113.20	80	*	> 1023	TCP
9	*DENY*	*	*	*	*	*

Al tratarse de una política cerrada o *deny all*, hay que rechazar todo el tráfico por defecto. Esto se consigue mediante la regla 9, que rechaza todo y es la última. El resto de las reglas son las siguientes: pt

- **Reglas 1 y 2**: se permite la salida de tráfico UDP destinado al puerto UDP 53, que corresponde a DNS. La regla 2 es necesaria para permitir la entrada de la respuesta a la petición de DNS. El puerto de origen de la regla 1 se restringe a puertos > 1023. Es una buena idea restringir al máximo posible todas las reglas para evitar acciones no deseadas fruto de situaciones imprevistas.

- **Reglas 3 y 4**: permiten la salida de tráfico TCP desde puertos no privilegiados a cualquier parte de Internet. Asimismo, se permite la entrada de

tráfico TCP a esos mismos puertos no privilegiados. Aquí se ha optado por restringir el puerto de origen a un puerto de cliente.

- **Reglas 5 y 6**: permiten la salida de tráfico ICMP y la entrada. Se ha tenido que incluir la regla 6, que permite la entrada de paquetes ICMP, ya que, por ejemplo, si permitimos a los usuarios de la red enviar un *ICMP echo request* (*ping*), no tiene sentido que no permitamos la contestación (*echo replay*).

- **Reglas 7 y 8**: permiten conexiones exteriores hacia el servidor web por el puerto TCP 80 y su respuesta.

Se ha tenido que permitir todo el tráfico de entrada ICMP, lo que contradice la política de seguridad. Este es un caso donde la información básica que hemos usado para definir las reglas no es suficiente para implementar la política. Sería necesario introducir más detalles que nos permitan discernir qué casos concretos de mensajes ICMP dejamos que entren en la red (mensajes de respuesta a peticiones iniciadas en la red interna). La mayoría de los cortafuegos actuales permiten detallar el tipo de mensajes ICMP en la regla.

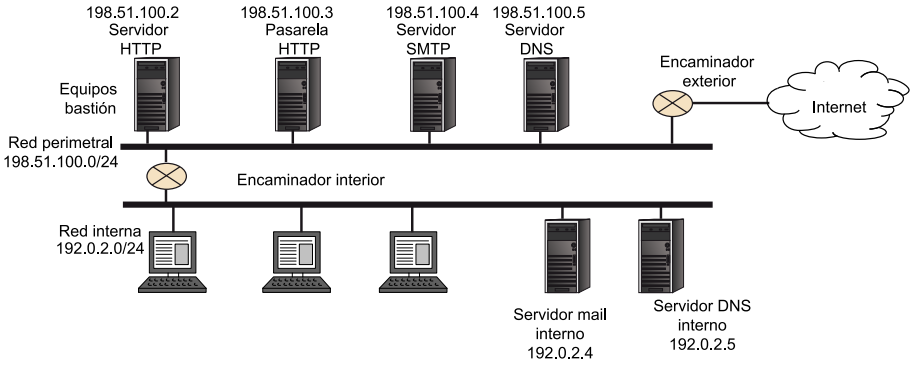
Otra observación importante sobre el ejemplo es que siempre se ha intentado restringir al máximo la aplicabilidad de las reglas. Por ejemplo, se restringe el uso de puertos *well known* (<1024), siempre que es posible. Es una práctica muy común y recomendada en el diseño de las reglas de cortafuegos seguir el principio de mínimo privilegio.

El **principio de mínimo privilegio** (o *least privilege principle*) determina que cada acción de sistema se realice con el conjunto de privilegios mínimo posible y estrictamente necesarios para llevar a cabo dicha acción.

4.3.2. Ejemplo 2: red perimetral

En este caso consideramos un ejemplo de red perimetral como el de la figura 14, la red perimetral (198.51.100.0/24) cuenta con 4 equipos bastión cada uno dedicado a un servicio concreto. Asimismo, la red interna (192.0.2.0/24) tiene un servidor de DNS y otro de correo electrónico para uso interno.

Figura 14. Ejemplo de filtrado con red perimetral



Las reglas utilizadas por los encaminadores contienen más información que en el ejemplo anterior. Esta información adicional es la dirección (orientación) del paquete, que puede ser entrada o salida. Hasta ahora hemos considerado que las direcciones IP de origen y destino nos proporcionaban dicha información. Sin embargo, esto no tiene por qué ser cierto, ya que las direcciones IP de un datagrama no tienen por qué ser legítimas. La mayoría de los cortafuegos permiten hacer esta distinción de manera explícita o indicando la interfaz de red por la que se recibe el paquete. Además, introducimos una condición relativa a segmentos TCP que indica si el segmento tiene el *flag* ACK activo o no.

Ved también

El *flag* ACK es estudia en el subapartado 4.2 de este módulo.

A continuación describimos la política del sistema:

1) Web. El equipo bastión 198.51.100.3 hace de pasarela de aplicación de los protocolos HTTP y HTTPS para los equipos de la red interna.

2) Servidor web. El equipo bastión 198.51.100.2 alberga el servidor web de la organización, que ofrece exclusivamente servicios por HTTP (puerto 80 de TCP) al exterior.

3) SSH. Se permite acceso por SSH a cualquier equipo de la red interna desde Internet o la red perimetral, así como acceso desde la red interna a cualquier equipo de Internet o red perimetral. En relación con la red perimetral, se permite el acceso desde Internet a dicha red para la administración remota de los equipos bastión.

Aunque SSH es un servicio importante y generalmente considerado seguro, es importante tener en cuenta que al permitir el tráfico SSH en ambas direcciones, estamos depositando una confianza importante en los usuarios. Permitir el establecimiento de conexiones SSH desde el exterior puede introducir vulnerabilidades sobre todo en presencia de usuarios maliciosos. Estos usuarios pueden ejecutar otros servicios en el puerto 22 o realizar *port forwarding* o túneles sobre SSH.

Ved también

Los túneles se estudian en el subapartado 5.1 de este módulo.

Una opción más segura es permitir únicamente conexiones SSH a la red interna desde la red perimetral (no desde todo Internet), y concretamente desde un equipo bastión que haría de pasarela SSH. Los usuarios deben entonces establecer primero una conexión con el equipo bastión, y desde este al equipo deseado de la red interna.

4) SMTP. Tenemos un servidor SMTP en el equipo bastión 198.51.100.4 de la red perimetral, que puede recibir tráfico SMTP de entrada y salida desde y hacia Internet. Así mismo, este servidor SMTP hará lo propio hacia un servidor SMTP de la red interna, el 192.0.2.4.

Esta es una configuración muy típica para servidores SMTP; los usuarios de la red interna utilizan el servidor interno, pero se limita el acceso de este desde el exterior mediante un equipo bastión. Este equipo bastión está haciendo en cierta manera de pasarela SMTP y evita que posibles atacantes puedan acceder directamente al servidor interno que utilizan los usuarios. La redirección del correo electrónico externo hacia el equipo bastión se suele hacer mediante registros MX en el DNS.

5) DNS. Hay un servidor interno de DNS (192.0.2.5) que utilizan los equipos de la red interna. Este servidor a su vez realiza las peticiones y recibe respuestas de un servidor DNS situado en un equipo bastión de la red perimetral (198.51.100.5), que a su vez utilizará servidores externos de Internet. Cualquier petición de DNS externa será recibida por el servidor en el equipo bastión y no podrá acceder directamente a la red interna. Asimismo, se permitirán las transferencias de zona (mecanismo para la replicación de bases de datos DNS) entre estos dos servidores.

Lecturas complementarias

Podéis encontrar información detallada sobre el funcionamiento de los protocolos de DNS en los RFC 1034 y 1035, y sobre la transferencia de zonas, en el RFC 5936.

El tráfico DNS suele tener cierta complejidad y conviene entender bien cómo funcionan los protocolos DNS. De modo breve recordemos que los mensajes de petición y respuesta pueden ir sobre TCP o UDP, el puerto del servidor es el 25. La petición y respuesta en caso de ser entre servidores se hace por UDP en el puerto 25 (ambos utilizan el mismo puerto).

A continuación procedemos a mostrar una posible configuración de reglas para los encaminadores interno y externo. Aparte de reglas relativas a cada protocolo y servicio, los dos encaminadores incorporan la reglas por defecto al final que rechazan cualquier paquete. En este caso, es necesario especificar una regla por defecto para entrada y otra para salida.

El hecho de utilizar la orientación del paquete nos permite también introducir reglas para poder desechar directamente paquetes con información errónea, evitando así algún posible ataque de *IP spoofing*. Por ejemplo, el encaminador interno no debería dejar entrar tráfico cuya dirección de origen sea de la pro-

pia red interna (regla X_1 del encaminador interno). De la misma manera, el encaminador externo no debería dejar entrar desde el exterior paquetes cuya dirección de origen sea de la red interna o perimetral (reglas X_1,X_2 del externo).

En la tabla 5 vemos las reglas correspondientes al encaminador interno.

Tabla 5. Reglas para el encaminador interno

Regla	Acción	Dir.	IP origen	P. origen	IP destino	P. destino	Prot.	ACK
X_1	*DENY*	In	192.0.2.0/24	*	*	*	*	*
H_1	*ACCEPT*	Out	192.0.2.0/24	> 1023	198.51.100.3	80	TCP	*
H_2	*ACCEPT*	In	198.51.100.3	80	192.0.2.0/24	> 1023	TCP	*
S_1	*ACCEPT*	Out	192.0.2.0/24	*	*	22	TCP	*
S_2	*ACCEPT*	In	*	22	192.0.2.0/24	*	TCP	Si
S_3	*ACCEPT*	In	*	*	192.0.2.0/24	22	TCP	*
S_4	*ACCEPT*	Out	192.0.2.0/24	22	*	*	TCP	Si
M_1	*ACCEPT*	Out	192.0.2.4	> 1023	198.51.100.4	25	TCP	*
M_2	*ACCEPT*	In	198.51.100.4	25	192.0.2.4	> 1023	TCP	Si
M_3	*ACCEPT*	In	198.51.100.4	> 1023	192.0.2.4	25	TCP	*
M_4	*ACCEPT*	Out	192.0.2.4	25	198.51.100.4	> 1023	TCP	Si
D_1	*ACCEPT*	Out	192.0.2.5	53	198.51.100.5	53	UDP	-
D_2	*ACCEPT*	In	198.51.100.5	53	192.0.2.5	53	UDP	-
D_3	*ACCEPT*	Out	192.0.2.5	> 1023	198.51.100.5	53	TCP	*
D_4	*ACCEPT*	In	198.51.100.5	53	192.0.2.5	> 1023	TCP	Si
D_5	*ACCEPT*	In	198.51.100.5	> 1023	192.0.2.5	53	TCP	*
D_6	*ACCEPT*	Out	192.0.2.5	53	198.51.100.5	> 1023	TCP	Sí
F_1	*DENY*	Out	*	*	*	*	*	*
F_2	*DENY*	In	*	*	*	*	*	*

A continuación se detallan estas reglas:

- H_1,H_2: permiten la salida de tráfico HTTP y HTTPS de la red interna hacia el equipo bastión que hace de pasarela de aplicación para estos protocolos y permite la recepción de su respuesta. Es importante señalar que los clientes necesitan configurar esta pasarela en sus navegadores, y que no es necesario habilitar el tráfico para el puerto 443 (HTTPS), ya que dicha conexión la realiza la pasarela, no el cliente (la conexión entre el cliente y la pasarela siempre es por el puerto 80).

- S_1,S_2: permiten establecer conexiones desde la red interna a servidores SSH del exterior (incluida la red perimetral)

- S_3,S_4: permiten establecer conexiones SSH desde el exterior a servidores situados en la red interna.

- M_1,M_2: permiten la salida del correo electrónico desde el servidor de correo interno hacia el servidor del equipo bastión en la red perimetral.

- M_3,M_4: permiten la entrada de correo desde el servidor de la red perimetral al servidor interno.

- D_1: permite peticiones y respuestas DNS por UDP desde el servidor interno al situado en el equipo bastión.

- D_2: igual que la anterior, pero desde el equipo bastión hacia el servidor interno.

- D_3,D_4: permiten peticiones DNS por TCP desde el servidor interno al del equipo bastión. Estas reglas también permiten la transferencia de zona desde el servidor del equipo bastión (primario) al de la red interna (secundario).

- D_5,D_6: equivalentes a las anteriores, pero intercambiando los servidores.

De la misma manera, la tabla 6 muestra las reglas del encaminador externo.

Tabla 6. Reglas para el encaminador externo

Regla	Acción	Dir.	IP origen	P. origen	IP destino	P. destino	Prot.	
X_1	*DENY*	In	192.0.2.0/24	*	*	*	*	*
X_2	*DENY*	In	168.51.100.0/24	*	*	*	*	*
H_1	*ACCEPT*	Out	198.51.100.3	> 1023	*	*	TCP	*
H_2	*ACCEPT*	In	*	*	198.51.100.3	> 1023	TCP	Si
H_3	*ACCEPT*	In	*	> 1023	198.51.100.2	80	TCP	*
H_4	*ACCEPT*	Out	198.51.100.2	80	*	> 1023	TCP	Si
S_1	*ACCEPT*	Out	192.0.2.0/24	*	*	22	TCP	*
S_2	*ACCEPT*	In	*	22	192.0.2.0/24	*	TCP	Si
S_3	*ACCEPT*	In	*	*	192.0.2.0/24	22	TCP	*
S_4	*ACCEPT*	Out	192.0.2.0/24	22	*	*	TCP	Si
S_5	*ACCEPT*	In	*	*	198.51.100/24	22	TCP	*
S_6	*ACCEPT*	Out	198.51.100/24	22	*	*	TCP	Si

Regla	Acción	Dir.	IP origen	P. origen	IP destino	P. destino	Prot.	
M_1	*ACCEPT*	Out	198.51.100.4	> 1023	*	25	TCP	*
M_2	*ACCEPT*	In	*	25	198.51.100.4	> 1023	TCP	Si
M_3	*ACCEPT*	In	*	> 1023	198.51.100.4	25	TCP	*
M_4	*ACCEPT*	Out	198.51.100.4	25	*	> 1023	TCP	Si
D_1	*ACCEPT*	Out	198.51.100.5	53	*	53	UDP	-
D_2	*ACCEPT*	In	*	53	198.51.100.5	53	UDP	-
D_3	*ACCEPT*	In	*	*	198.51.100.5	53	UDP	-
D_4	*ACCEPT*	Out	198.51.100.5	53	*	*	UDP	-
D_5	*ACCEPT*	Out	198.51.100.5	> 1023	*	53	TCP	*
D_6	*ACCEPT*	In	*	53	198.51.100.5	> 1023	TCP	Sí
D_7	*ACCEPT*	In	*	> 1023	198.51.100.5	53	TCP	*
D_8	*ACCEPT*	Out	198.51.100.5	53	*	>1023	TCP	Sí
F_1	*DENY*	Out	*	*	*	*	*	*
F_2	*DENY*	In	*	*	*	*	*	*

A continuación se comentan y justifican:

- H_1, H_2: permiten la salida y posterior entrada de tráfico TCP del equipo bastión que hace de pasarela HTTP, hacia cualquier servidor de Internet en cualquier puerto. Esto permite conexiones a servidores web en puertos 80 (HTTP) y 443 (HTTPS) y a puertos no estándar. Son unas reglas muy laxas en relación con la restricción del puerto. Este tipo de laxitud es relativamente normal en un encaminador externo, como se discutió en el subapartado 3.2. Como medida adicional, solo se permite el establecimiento de conexiones hacia el exterior, ya que la regla de entrada requiere que el segmento tenga el *flag* ACK activo, es decir, se descartará si se trata de un inicio de sesión (primer segmento SYN del establecimiento de conexión TCP).

- H_3, H_4: permiten la entrada de tráfico TCP al equipo bastión que hace de servidor web. Solo se permite el acceso al puerto 80, y solo se permite el establecimiento de conexión desde fuera hacia dentro (ACK).

- S_1, S_2, S_3,S_4: equivalentes a las respectivas reglas en el encaminador interno.

- S_5, S_6: permiten conexiones SSH desde el exterior a la red perimetral para la administración remota de los equipos bastión.

- M_1, M_2: conexiones SMTP desde el equipo bastión hacia Internet.

- M_3, M_4: conexiones SMTP desde Internet hacia el equipo bastión.

- D_1: peticiones y respuestas UDP desde el servidor DNS del equipo bastión a servidores de Internet.

- D_2: peticiones y respuestas UDP desde servidores de Internet al servidor del equipo bastión.

- D_3, D_4: permite a clientes DNS de Internet preguntar al servidor del equipo bastión y recibir sus respuestas. En este caso, la regla D_3 hace que la regla D_2 sea redundante; se ha incluido la D_2 a modo ilustrativo (aun así no es mala práctica permitir esta redundancia con el propósito de facilitar la administración y lectura de las reglas de filtrado).

- D_5, D_6: peticiones del servidor del equipo bastión a servidores de Internet (y sus respectivas respuestas). También permite transferencia de zona.

- D_7, D_8: equivalentes a las anteriores, pero intercambiando los servidores.

5. Más allá del filtrado de paquetes

Hoy en día los sistemas de cortafuegos realizan bastantes tareas que van más lejos del puro filtrado de paquetes. Algunos ejemplos de ello son la creación de redes privadas virtuales o NAT. También incorporan técnicas adicionales de seguridad, como *port knocking*. En este apartado repasamos de manera rápida estos conceptos.

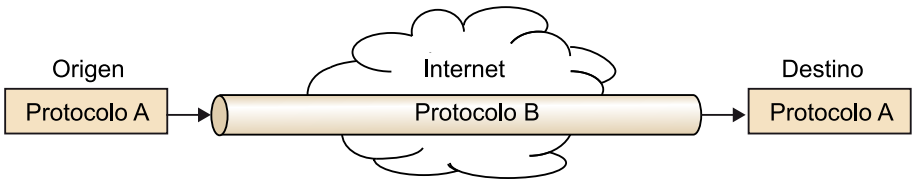
5.1. Túneles y redes privadas virtuales

En muchas ocasiones que se opta por el uso de sistemas cortafuegos muy restrictivos nos encontramos con la necesidad de establecer conexiones seguras entre puntos concretos de Internet que puedan saltarse las reglas del cortafuegos. Aquí entra en juego el concepto de *túnel*.

Un **túnel** es la encapsulación de un protocolo en otro, y se puede ver como un cable virtual que enlaza dos puntos de Internet.

Un túnel puede permitir saltarse la reglas de un cortafuegos, lo que puede tener aplicaciones ventajosas en algunos casos. En el origen se encapsula un protocolo en otro, que puede atravesar cualquier red. En el destino se puede desencapsular el protocolo original (figura 15).

Figura 15. Túnel



En principio, existen muchos tipos de túneles que pueden tener como propósito saltarse las reglas de un cortafuegos. En el ejemplo del subapartado 4.3.2 teníamos un sistema de cortafuegos con red perimetral en el que se permite conexión SSH directa entre la red interna y la exterior. Para ello se permitían conexión TCP al puerto 22 o hacia el puerto 22. Sin embargo, el tráfico HTTP tenía que pasar por la pasarela de apliación situada en la red perimetral.

Ante esta situación, un usuario malintencionado podría establecer un túnel TCP por el puerto 22 entre su equipo de la red interna y un equipo exterior. Es decir, encapsular en TCP cualquier protocolo y dirigirlo al puerto 22. Si lo hace de manera adecuada, podría por ejemplo acceder a servidores externos HTTP desde el interior sin necesidad de utilizar la pasarela HTTP, saltándose así la política del sistema y el filtrado que realizaría la pasarela. En este caso

podría incluso encapsular HTTP en SSH estableciendo un túnel SSH. No solo es muy sencillo hacerlo, ya que la mayoría de las implementaciones actuales de SSH incorporan esta funcionalidad directamente, sino que además el tráfico iría cifrado. El protocolo encapsulado no sería detectable por cualquier tipo de filtrado por el que pasase el túnel.

Actualmente existen numerosos tipos de túneles y posibilidades de encapsulamiento, sobre DNS, HTTP, FTP, etc. Este tipo de túneles puede ser usado con el propósito de violar la política implementada por un cortafuegos. Por este motivo se considera que es muy difícil proteger una red de sus usuarios internos mediante sistemas cortafuegos. Por lo general, un cortafuegos nos permite proteger una red ante una amenaza exterior, pero no de una interna.

Lectura complementaria

Como curiosidad, el RFC 1149 define el encapsulamiento de IP en palomas mensajeras.

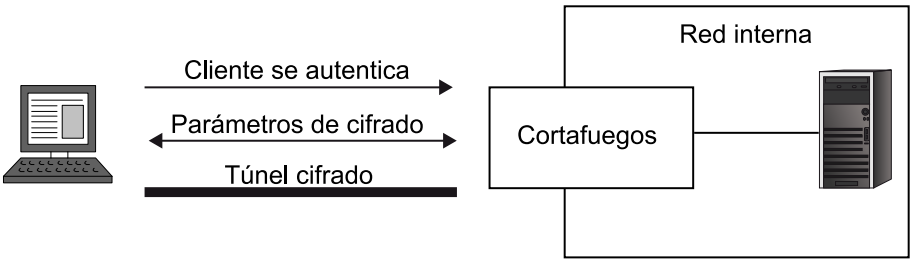
A pesar de los posibles usos maliciosos, los túneles pueden tener aplicaciones positivas. Se pueden utilizar para unir dos redes separadas por una red hostil, como Internet. Cada red está fuertemente protegida con un sistema cortafuegos y el túnel permite unirlas como si existiese un enlace físico entre ellas. En este caso, es conveniente que el tráfico del túnel esté cifrado. Este tipo de túneles suele recibir el nombre de red privada virtual.

Una red privada virtual (VPN, *Virtual Private Network*) es una red privada que interconecta puntos o redes remotas a través de redes públicas como Internet.

Una VPN utiliza túneles cifrados y mecanismos de autenticación para garantizar la seguridad de la red virtual. Los usuarios de la VPN tienen la sensación de estar en una red privada cuando en realidad están separados por una red pública. Existen muchos tipos de VPN, que pueden utilizar muchos tipos de protocolos para establecer el túnel a diferentes niveles de red. Es común el uso de protocolos genéricos como IPSec o SSL/TL por si solos o en combinación con protocolos más específicos, como *Layer 2 Tunneling Protocol* (L2TP), que encapsula IP sobre protocolos que soporten entrega punto-a-punto, como IP, ATM o Frame Relay.

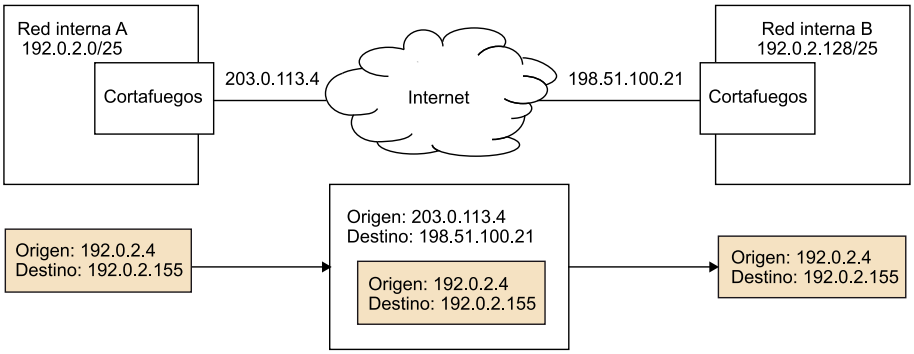
Aparte del tipo de túnel utilizado, una VPN puede proporcionar varios servicios. Generalmente, una VPN puede ser configurada, por ejemplo, para permitir que empleados de la empresa accedan a la red corporativa desde casa, a través de Internet. El usuario se autentica y se establece un túnel cifrado (tras establecer o negociar los parámetros necesarios) entre el cliente que tiene el usuario y la red interna de la empresa (figura 16).

Figura 16. VPN para la conexión de un cliente



Otra posibilidad también común es poder unir dos redes separadas geográficamente a través de un túnel por una red pública. En este caso, el objetivo es que se tenga la percepción de que las dos redes son una sola. Este tipo de VPN suele recibir el nombre de *branch office*. La figura 17 muestra un ejemplo muy simplificado donde los datagramas IP se encapsulan para pasar por Internet.

Figura 17. VPN para la conexión de dos redes corporativas



5.2. *Port knocking*

Port knocking (a veces traducido como *golpeo de puertos*) es una técnica que apareció en 2003 como un mecanismo de autenticación ante un cortafuegos.

Port knocking permite la comunicación auténtica de datos a través de un sistema cortafuegos con los puertos cerrados. O, dicho de otra manera, permite ofrecer un servicio por un cortafuegos que tiene los puertos cerrados por defecto.

Lectura recomendada

La primera publicación que hace referencia a la técnica del golpeo de puertos es la siguiente: M. Krzywinski (2003). "Port Knocking: Network Authentication Across Closed Ports". *SysAdmin Magazine* (núm. 12, pág. 12-17).

La idea consiste en ofrecer un servicio protegido por un cortafuegos que no permite el paso de ningún paquete, es decir, tiene todos los puertos cerrados. El cliente realiza una serie de intentos de conexión a una secuencia de puertos concreta. Si esta secuencia es la correcta, el sistema de cortafuegos abrirá el puerto que permitirá al cliente acceder al servicio protegido. Un intento de conexión puede ser, por ejemplo, el envío de un segmento de sincronización de TCP (SYN) con el propósito de iniciar una conexión. La secuencia de puertos es la clave (compartida por el cliente y servidor) que va a permitir al cliente autenticarse.

Los sistemas de *port knocking* pueden monitorizar los logs del sistema de cortafuegos, o simplemente capturar paquetes antes de que lleguen al cortafuegos, para detectar la secuencia de puertos. Existen también actualmente sistemas de cortafuegos, como Netfilter/iptables, que incorporan soporte para *port knocking*.

Knockd

Knockd es un servidor de *port knocking* que analiza el tráfico a nivel de capa de enlace para detectar secuencias de puertos. Estas secuencias se pueden hacer enviando paquetes TCP o UDP a los puertos correspondientes. *Knockd* está disponible para Linux, Window y OSX en zeroflux.org.

La versión básica de *port knocking*, y la más común, establece una secuencia de puertos compartida entre el servidor y el cliente como clave. Sin embargo, actualmente existen modalidades más sofisticas. Un caso más extremo es cifrar información sobre la conexión deseada: IP de origen, puerto, etc. con un cifrado simétrico. Esta información cifrada se puede dividir en bytes, cada byte se interpreta como un número decimal (de 0 a 255) y a cada número se le suma, por ejemplo 24000. Al final tenemos una secuencia de números en el rango de 24000 a 24255 que será la secuencia de puertos utilizada. El servidor recibe la secuencia y la puede descifrar obteniendo así la IP de origen[5], puerto y servicio que debe facilitar al cliente. En este caso, es necesario que tanto cliente como servidor compartan una clave simétrica, pero el método es mucho más seguro que el modo básico, ya que puede evitar ataques de repetición. En el ejemplo anterior estamos enviando un byte de información con cada paquete. Dado que el puerto son 16 bits, se podrían enviar un máximo de dos bytes.

(5)La dirección IP de los paquetes que hacen el port knocking suele ser falsa para evitar a posibles espías.

Existe una variante de *port knocking* conocida como *Single Packet Authorization* (SPA), que permite hacer la autenticación con un solo paquete. La idea es enviar un paquete que contenga como datos la información de autenticación cifrada. Esta variación consigue enviar la información en un solo paquete.

fwknop

fwknop (*FireWall KNock OPerator*) es un servidor disponible para Linux, BSD y OSX que permite implementar *port knocking* y SPA.

El uso de estas técnicas añade un nivel más de seguridad defensiva al sistema que ya estaba protegido por el sistema cortafuegos. El hecho de que el cortafuegos esté cerrado totalmente por defecto puede evitar muchos problemas derivados de vulnerabilidades de día-cero por ejemplo. En el campo de la seguridad informática se utiliza el principio de *defense in depth* (defensa en profundidad). Este viene a decir que la seguridad de un sistema se mejora superponiendo múltiples mecanismos defensivos.

5.3. Uso de NAT

NAT (*Network Address Translation*) permite usar en una red internamente un conjunto de direcciones y otro diferente de cara al exterior. Su uso más común es el de compartir una misma dirección IP pública entre varios dispositivos que internamente utilizan direcciones privadas.

El funcionamiento de los dispositivos de NAT tiene muchas similitudes con el de los sistemas cortafuegos, hasta el punto de que muchos sistemas de cortafuegos pueden realizar también NAT. Es el caso de Netfilter/iptables, por ejemplo.

El uso de NAT puede tener ventajas adicionales desde el punto de vista de la seguridad y los sistemas de cortafuegos:

- Al asignar direcciones IP privadas a dispositivos de la red, nos aseguramos de que estas deben hacer la comunicación con el exterior a través del dispositivo de NAT que puede estar haciendo también funciones de cortafuegos. Si un equipo interno intenta saltarse el cortafuegos, no podrá hacerlo con su dirección IP privada, ya que no se puede encaminar en Internet.

- Ayuda a limitar el tráfico de entrada. Dependiendo del tipo de NAT que se esté haciendo, este dejará pasar solo el tráfico de entrada que forme parte de una comunicación iniciada desde el interior.

- También puede ayudar a ocultar información sobre la red interna a posibles atacantes externos.

Estas son indicaciones generales, ya que dependiendo de qué tipo de NAT se utilice tendremos más ventajas o desventajas. Por ejemplo, se puede hacer NAT utilizando mapeo de puertos (de direcciones internas); este tipo de NAT puede, en algunos casos, interferir con el sistema de filtrado, ya que por ejemplo se modifica el puerto de origen del datagrama.

Resumen

Un sistema de cortafuegos permite establecer una barrera de seguridad entre redes informáticas. Mediante cortafuegos podemos proteger una red, o parte de ella, de entornos hostiles que sean una potencial fuente de ataque. Existen varias tecnologías y arquitecturas de sistemas de cortafuegos, y cada una tiene su aplicabilidad, presentando ventajas y desventajas.

Los cortafuegos controlan el flujo de red filtrando los paquetes que pasan por ella. Este filtrado puede tener varios niveles de complejidad y consecuentemente eficiencia. Desde un filtrado sencillo en el que solo se considera información de las cabeceras de los protocolos principales, hasta sofisticados sistemas de filtrado continuo de contenidos.

Actividades

1. Los códigos 9 y 10 fueron especialmente añadidos a la especificación de ICMP para ser utilizados por sistemas de filtrado. Aun así, muchos siguen utilizando los códigos 0 y 1 que fueron inicialmente pensados con otros propósitos. ¿Por qué creéis que se tuvieron que introducir los nuevos códigos? ¿Qué contraindicaciones puede tener el uso de los códigos 0 y 1?

2. Existe un sistema de filtrado de grandes dimensiones en Internet conocido como *the Great Firewall of China*. Buscad información sobre este sistema. ¿Qué es y qué propósito tiene? ¿Cómo consigue este sistema bloquear el tráfico? O, dicho de otra manera, ¿cómo rechaza paquetes? ¿Cómo se puede evitar?

Pista. Podéis consultar el siguiente artículo: R. Clayton; S. J. Murdoch; R. N. M. Watson (2006). "Ignoring the Great Firewall of China" (en línea). 6th Workshop on Privacy Enhancing Technologies.

3. En este módulo se comenta que hay sistemas de cortafuegos como Netfilter/iptables que incorporan *port knocking*. Definid el conjunto de reglas de iptables que necesitáis para definir un *port knocking* que abra el puerto 23 si se recibe la secuencia de puertos *port knocking*: 1000, 2314, 4132, 2222.

4. En este módulo se comenta que la modalidad de cifrado en *port knocking* puede evitar ataques de repetición. ¿En qué consisten estos ataques en el contexto de *port knoking*? ¿Hasta qué punto el cifrado evita estos ataques? ¿Es SPA también vulnerable a ataques de repetición?

Glosario

amenaza *f* Violación de la seguridad en potencia, que existe a partir de unas circunstancias, capacidad, acción o evento que pueda llegar a causar una infracción de la seguridad y/o causar algún daño en el sistema.

ataque *m* Agresión a la seguridad de un sistema fruto de un acto intencionado y deliberado que viola la política de seguridad de un sistema.

DNS, *Domain Name System* *m* Sistema de nombres jerárquico y distribuido que permite asociar nombres de dominio a direcciones IP.

dirección IP *f* Dirección utilizada por el protocolo IP.

equipo bastión (o *bastion host*) *m* Sistema informático que ha sido fuertemente protegido para soportar ataques desde un lugar hostil.

equipo *dual-homed* *m* Equipo con al menos dos interfaces de red, cada una asociada a una red, que puede actuar como encaminador entre las redes.

ICMP, *Internet Control Message Protocol* *m* Protocolo de control, principalmente para envío de mensajes de error, de TCP/IP.

IP, *Internet Protocol* *m* Protocolo para la interconexión de redes.

pasarela a nivel de circuito *f* Dispositivo que hace de pasarela a nivel de capa de transporte entre dos extremos. Establece una conexión con cada una retransmitiendo los datos entre las dos conexiones.

política de seguridad *f* Conjunto de reglas y prácticas que definen y regulan los servicios de seguridad de una organización o sistema con el propósito de proteger sus recursos críticos y sensibles. En otras palabras, es la declaración de lo que está permitido y lo que no está permitido hacer.

port knocking *f* Técnica que permite la comunicación auténtica de datos a través de un sistema cortafuegos con los puertos cerrados.

TCP, *Transmission Control Protocol* *m* Protocolo de transporte (extremo-a-extremo) de TCP/IP.

túnel *m* Encapsulación de un protocolo en otro. Se puede ver como un cable virtual que enlaza dos puntos de Internet.

UDP, *User Datagram Protocol* *m* Protocolo de transporte (extremo-a-extremo) de TCP/IP.

VPN, *Virtual Private Network* *f* Red privada que interconecta puntos o redes remotas a través de redes públicas como Internet.

vulnerabilidad de día-cero (*zero-day vulnerability*) *f* Vulnerabilidad de cuya existencia no se tiene conocimiento en el momento de ser explotada.

vulnerabilidad de seguridad *f* Fallo o debilidad en el diseño, la implementación, la operación o la gestión de un sistema, que puede ser explotado con el fin de violar la política de seguridad del sistema.

Bibliografía

Avolio, F. (1999). "Firewalls and Internet Security, the Second Hundred (Internet) Years".*The Internet Protocol Journal* (vol. 2, núm. 2)

Cheswick, W. R.; Bellovin, S. M.; Rubin, A. D. (2003). *Firewalls and Internet Security: Repelling the Wily Hacker, 2a ed*. Addison-Wesley Professional Computing

Fraser, B. (1997). *Site Security Handbook*. RFC 2196, IETF. The Internet Society.

Garcia-Alfaro, J. (2004). "Mecanismos de prevención". En: J. Herrera; J. Garcia-Alfaro; X. Perramon. *Seguridad en redes de computadores*. Fundació Universitat Oberta de Catalunya, 287 pág.

Microsoft (2010). "Windows Firewall with Advanced Security Getting Started Guide" (en línea). Microsoft TechNet Library.

Rash, M. (2007). *Linux Firewalls: Attack Detection and Response with iptables, psad, and fwsnort*. No Starch Press.

Shirey, R. (2000). *Internet Security Glossary* RFC 2828, IETF. The Internet Society

Zwicky, E. D.; Cooper, S.; Chapman, D. B. (2000). *Building Internet Firewalls 2a. ed.* O'Reilly Media.