Técnica

# El desvío de los cortafuegos de red

Oliver Karow

Grado de dificultad

**Se suele tener la idea de que un cortafuegos protege por completo a las redes del acceso no autorizado. Sin embargo, los cortafuegos también tienen debilidades y es posible desviarlos, ya sea mediante un fallo en la configuración o a través de las debilidades del producto. Vamos a observar cómo un intruso puede acceder a un sistema por medio de la desviación de un cortafuegos.**

En la actualidad, una de las necesidades más importantes de las infraestructuras IT es la de proteger a una red de los ataques y del acceso involuntario de las redes no fiables como Internet. Esta es la zona en la que los cortafuegos comienzan a funcionar. La tarea primaria de un cortafuegos es la de separar las redes, y decidir si se les permite a los paquetes pasar de una red a otra.

Hay diferentes tipos de cortafuegos que tienen diferentes enfoques en cuanto al cumplimiento de la tarea primaria. Los dos tipos más comunes son: los filtros de paquetes y los cortafuegos de la capa de aplicación (ver Recuadro *Manual del cortafuegos básico*).

Independientemente del tipo que sea, un cortafuegos necesita de alguna base para decidir si un paquete será enviado a su destino o no. Esta es básicamente la política de un cortafuegos en forma de listas de acceso o de reglas de filtrado. Vamos a ver las posibilidades del desvío de tales políticas mediante el abuso de reglas de filtrado defectuosas, debilidades en los protocolos comunes, y las limitaciones de los diferentes tipos de cortafuegos.

## La detección de cortafuegos

Antes de que un sistema ubicado detrás de un cortafuegos pueda ser atacado, primero el intruso debe determinar si hay un cortafuegos en ese sitio. Esto no siempre es tan evidente como parece, pues quienes mantienen los cortafuegos utilizan a menudo trucos que impiden la detección de los mismos. Sin embargo, ya que un cortafuegos puede interferir en los resultados de un ataque, es importante ser conscientes de su existencia. Primero, veamos algunas técnicas que se emplean en la detección de un cortafuegos.

### En este artículo aprenderás...

- cómo funcionan los cortafuegos,
- cómo pueden ser detactados,
- cómo puede ser desviado un cortafuegos aprovechando las configuraciones erróneas o las debilidades de los productos de cortafuegos.

### Lo que deberías saber...

- debes estar al corriente sobre el TCP/IPv4,
- debes conocer el modelo de referencia ISO/OSI.

## Manual del cortafuegos básico

Un cortafuegos es en general un sistema con interfaces múltiples, sujeto a diferentes redes, que tiene un mecanismo de filtrado para que permita o bloquee el tráfico entre las redes. Los cortafuegos pueden ser clasificados según la capa TCP/IP utilizada para el análisis y envío de los paquetes:

### Los filtros de paquetes

El filtrado de paquetes analiza los paquetes en la Red (3) y Transporta (4) las capas del modelo ISO/OSI. Eso significa que un filtro de paquetes utiliza principalmente el siguiente criterio para acometer su decisión de filtrado:

- los protocolos (ICMP, OSPF, AH, ESP, etc.),
- la dirección IP de origen,
- la dirección IP de destino,
- el puerto de origen,
- el puerto de destino,
- los indicadores TCP (SYN, ACK, RST, FIN, etc.).

### Los filtros de paquetes dinámicos/estáticos

Un filtro de paquetes estático está al tanto de cada conexión y almacena esta información en tablas de estado internas, para ampliar las capacidades de un paquete de filtros sencillo. Cuando un paquete saliente pasa por el filtro de paquetes (e inicia una conexión), los puertos que se corresponden y las direcciones IP para los paquetes de respuesta se abren durante la conexión y luego se cierran.

Además, algunos filtros de paquetes estáticos son capaces de abrir puertos dinámicamente si se ha negociado un nuevo puerto o dirección IP entre cliente y servidor en una conexión permitida. Algunos servicios como el Oracle y el Portmapper lo hacen.

### Los cortafuegos a niveles de la aplicación

Los cortafuegos a nivel de aplicación son capaces de analizar los paquetes hasta la capa de aplicación del modelo ISO/OSI. Además de poseer las características de un filtro de paquetes estático/dinámico, también son capaces de inspeccionar la carga útil de un paquete. Mientras que un filtro de paquetes sólo puede tomar decisiones basadas en la información de la cabecera del paquete, un cortafuegos a nivel de la aplicación puede examinar la información específica de la aplicación. Por ejemplo, permite que este tipo de cortafuegos admita lo comunicación HTTP con el puerto 80/TCP en general, pero bloquea las consultas con ciertos comandos como el `CONNECT` o el `DELETE`.

Los cortafuegos a nivel de aplicación necesitan un servicio de proxy especial que se ejecute en cada protocolo que tiene que pasar a través de un cortafuegos. Ya que el servicio proxy no siempre está disponible, la mayoría de vendedores de cortafuegos emplean de manera adicional capacidades de filtrado de paquetes y servicios proxy genéricos, sin la habilidad de análisis del protocolo.

### Los cortafuegos híbridos y de capa 2

Muchos vendedores de cortafuegos están utilizando una tecnología híbrida para obtener lo mejor de cada tipo de cortafuegos. Eso significa que incluyen en sus productos el filtrado de paquetes estático así como las habilidades de la capa de la aplicación. También hay cortafuegos de capa 2 disponibles en el mercado. Estos no son tan populares como los de filtrado de paquetes y los de capa de la aplicación, y se usan principalmente a nivel de la interfaz, dependiendo del vendedor.

## El Traceroute

El Traceroute es un mecanismo utilizado para descubrir los routers que envían paquetes próximos a su destino. Si hay un cortafuegos colocado en el sitio podría responder a un paquete del traceroute.

Ya que el traceroute es en sí mismo una técnica muy antigua, la mayoría de los cortafuegos lo bloquean. Sin embargo, aún existen malos entendidos en cuanto a la funcionalidad del traceroute, lo que permite que los intrusos se abran paso a través de un sistema de cortafuegos.

El Listado 1 muestra los resultados de un traceroute, cuando es bloqueado por un cortafuegos. Como podemos apreciar, el traceroute funciona hasta que llega al sistema con la IP 10.4.4.254. Luego aparece algo en el sitio que bloquea los intentos de seguir una ruta.

Ahora vamos a intentar entender cómo funciona el seguimiento de rutas (ver también Figura 1). Para determinar la ruta de un paquete IP, se utiliza el campo TTL de la cabecera IP, de manera que este se reduce en uno cada vez que el paquete llega a un router. Si el router recibe un paquete IP con el valor de dos, este le restará uno, y si el resultado es mayor o igual a uno, será remitido al próximo router de acuerdo con la información del enrutado. Si un router recibe un paquete con el valor TTL de 1, lo restará, y ya que el valor resultante es de cero, no enviará el paquete al próximo router, y en su lugar enviará una notificación al remitente para informarle de que el paquete fue descartado por el camino.

El traceroute comienza su trabajo enviando el primer paquete con un TTL igual a 1. Este obtiene una notificación de expiración del ICMP TTL del primer router. Entonces aumenta el TTL a 2 para pasar el primer router y conseguir una notificación similar desde el próximo router en curso. Esto sucede de manera continua hasta que se alcanza el objetivo. Como cada router envía una notificación, el traceroute puede construir una lista de routers (si no está configurado de otra manera).

También es importante saber que hay dos maneras de hacer uso del traceroute. La primera utiliza los paquetes ICMP *de consulta echo* (por ejemplo, las ejecuciones por parte de Windows del traceroute), y los otros paquetes UDP (por ejemplo la mayoría de las ejecuciones *NIX). Ambas utilizan la técnica del TTL. Por tanto, es importante para un administrador de cortafuegos filtrar ambas ejecuciones del traceroute.

**Listado 1.** *El Traceroute bloqueado por un cortafuegos*

```
# traceroute www.dummycompany.de
traceroute to www.dummycompany.de (10.10.10.10), 30 hops max, 40 byte packets
 1  10.255.255.254                    0.373 ms  0.203 ms  0.215 ms
(...)
10  router.company1.de (10.1.1.254)  88.080 ms  88.319 ms  87.921 ms
11  router.company2.de (10.2.2.254)  87.881 ms  89.541 ms  88.081 ms
12  router.company3.de (10.3.3.254)  86.749 ms  86.919 ms  86.734 ms
13  router.company4.de (10.4.4.254)  87.216 ms  87.312 ms  87.307 ms
14  * * *
```

**Listado 2.** *La utilización de la técnica TCP de traceroute con hping2*

```
# hping2 -T -t 1 -S -p 80 www.dummycompany.de
HPING www.dummycompany.de (eth0 10.10.10.10 ): S set, ←
  40 headers + 0 data bytes
hop=1 TTL 0 during transit from ip=10.255.255.254 name=UNKNOWN
hop=1 hoprtt=12.4 ms
(...)
hop=10 TTL 0 during transit from ip=10.1.1.254 name=router.company1.de
hop=11 TTL 0 during transit from ip=10.2.2.254 name=router.company2.de
hop=12 TTL 0 during transit from ip=10.3.3.254 name=router.company3.de
hop=13 TTL 0 during transit from ip=10.4.4.254 name=router.company4.de
hop=14 TTL 0 during transit from ip=10.5.5.254 name=UNKNOWN
len=46 ip=10.10.10.10 flags=SA DF seq=15 ttl=107 id=12852 win=29200 rtt=95.6 ms
len=46 ip=10.10.10.10 flags=R DF seq=15 ttl=107 id=12856 win=0 rtt=194.6 ms
```

**Listado 3.** *El envío de un paquete hacia un puerto cerrado*

```
# hping2 -S -p 99 -c 1 www.dontexist.com
HPING www.dontexist.com (eth0 192.168.10.10): S set, ←
  40 headers + 0 data bytes
ICMP Packet filtered from ip=192.168.9.254
```

**Listado 4.** *La observación del tráfico de red*

```
# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
12:59:18.778417 IP 172.16.1.1.1866 > 192.168.10.10.99: ←
  S 1958445360:1958445360(0) win 512
12:59:18.786914 IP 192.168.9.254 > 172.16.1.1 icmp 36: ←
  host 192.168.10.10 unreachable - admin prohibited filter
```

**Listado 5.** *La comparación de los valores del TTL*

```
# hping2 -S -p 80 -c 1 www.randomname.com
HPING www.randomname.com (eth0 192.100.100.10): ←
  S set, 40 headers + 0 data bytes
len=46 ip=192.100.100.10 flags=SA DF seq=0 ttl=55 id=0 win=5840 rtt=7.6 ms
# hping2 -S -p 99 -c 1 www.randomname.com

HPING www.randomname.com (eth0 192.100.100.10): ←
  S set, 40 headers + 0 data bytes
len=46 ip=192.100.100.10 flags=RA DF seq=0 ttl=56 id=0 win=0 rtt=7.6 ms
```

## El traceroute TCP

Ya que sabemos que el campo TTL es parte de la cabecera IP y los filtros comunes del traceroute sólo bloquean a los paquetes UDP e ICMP, podemos intentar evitar el filtro utilizando simplemente un paquete TCP en vez de un UDP o ICMP. Intentemos otra vez seguir la ruta hacia nuestro servidor de destino. Esta vez emplearemos la herramienta *hping2*, que nos permite enviar paquetes hábiles (ver Listado 2). Según podemos apreciar, hemos detectado otro salto. Mientras que el comando del *traceroute* fue bloqueado después del décimotercer router, *hping2* nos proporciona un resultado adicional.

## El análisis de los paquetes de respuesta

Para sondear la existencia de un cortafuegos, es bueno comparar los paquetes de respuesta de los puertos abiertos con aquellos de los puertos cerrados. Observemos algunos trucos que pueden demostrar la existencia de un cortafuegos.

Primero, utilicemos *hping2* para enviar un paquete a nuestro destino, hacia un puerto del que sabemos o asumimos que está abierto (ver Listado 3). Al mismo tiempo apreciemos el tráfico de la red utilizando el tcpdump (ver Listado 4). Podemos ver un mensaje de *destino inalcanzable* en forma de mensaje de *filtro de administración prohibido* desde la 192.168.9.254. Este mensaje indica que el acceso al puerto 99/TCP de nuestro sistema de destino es filtrado a través de una lista de acceso en el router. Ya que este indicador es muy evidente con respecto a la existencia de un cortafuegos, observemos otra técnica basada en el análisis de los valores del TTL.

## Las diferencias TTL

Cada vez que un paquete IP pasa por un dispositivo router, su TTL se reduce en uno. Así que si tenemos un servidor protegido por un cortafuegos de red instalado en un sistema determinado, podría ser posible que los paquetes que se originen desde el servidor tengan un TTL

diferente que los paquetes que se originen desde un cortafuegos.

El reto ahora es obtener un paquete de respuesta de ambos, del servidor y del sistema de cortafuegos potencialmente existente, y comparar los valores TTL de ambos paquetes. Si hay diferencia en el valor entonces es posible que haya un cortafuegos en el sitio.

Para forzar ambos sistemas a que respondan, podemos enviar un paquete a un puerto abierto y uno al puerto cerrado de nuestro sistema de destino, según lo cual el 80/TCP está abierto y el 99/TCP está cerrado (ver Listado 5). Como podemos apreciar, hay una diferencia entre los valores TTL (la diferencia de uno). Esto indica que existe un sistema de cortafuegos en el sitio que protege al servidor de destino.

## Determinar el tipo de cortafuegos

Las técnicas anteriores ayudan a demostrar la existencia de un cortafuegos. Si podemos identificar la direccion IP del cortafuegos, existen algunos trucos extra que ayudan a reunir información adicional, como el producto del cortafuegos o el sistema operativo en uso.

### Las identificación digital del TCP

Haremos uso del hecho de que cada pila IP de un sistema operativo tiene patrones únicos que hacen posible determinar la versión y el tipo de sistema operativo que está en uso. Ya que la mayoría de las aplicaciones de cortafuegos influyen en la conducta de la pila IP, también es posible determinar frecuentemente el tipo y la versión del cortafuegos instalado. La herramienta a elegir sería *nmap* con su capacidad de detección de OS integrada (ver Listado 6). Sólo escaneamos tres puertos y fuimos capaces de determinar lo que es probablemente un cortafuegos Checkpoint Firewall-1 NG ejecutado en un sistema operativo Solaris.

Vamos a observar otro cortafuegos (ver Listado 7), esta vez es un Cortafuegos de Symantec Enterpri-

**Listado 6.** *La Identificación Digital del OS y del cortafuegos con nmap*

```
# nmap -sS -F -n -O  -p 80,99,443 192.168.190.1
Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) ↩
  at 2005-10-09 17:23 CEST
Interesting ports on 192.168.190.1:
PORT    STATE SERVICE
80/tcp  open  http
99/tcp closed metagram
443/tcp open  https
Device type: firewall|broadband router|general purpose
Running: Checkpoint Solaris 8, Belkin embedded, Sun Solaris 8
OS details: Checkpoint Firewall-1 NG on Sun Solaris 8, ↩
  Belkin DSL/Cable Router, Sun Solaris 8, Sun Trusted Solaris 8
```

**Listado 7.** *La huella digital de un Cortafuegos de Symantec Enterprise*

```
Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) ↩
  at 2005-10-10 13:43 CEST
Interesting ports on 192.168.99.1:
(The 1193 ports scanned but not shown below are in state: closed)
PORT     STATE    SERVICE
21/tcp   open     ftp
22/tcp   open     ssh
23/tcp   open     telnet
25/tcp   open     smtp
53/tcp   open     domain
80/tcp   open     http
119/tcp  open     nntp
139/tcp  open     netbios-ssn
443/tcp  open     https
481/tcp  open     dvs
512/tcp  open     exec
513/tcp  open     login
514/tcp  open     shell
554/tcp  open     rtsp
1720/tcp open     H.323/Q.931
2456/tcp open     unknown
5631/tcp open     pcanywheredata
7070/tcp open     realserver
No exact OS matches for host (If you know what OS is running ↩
  on it, see http://www.insecure.org/cgi-bin/nmap-submit.cgi).
```

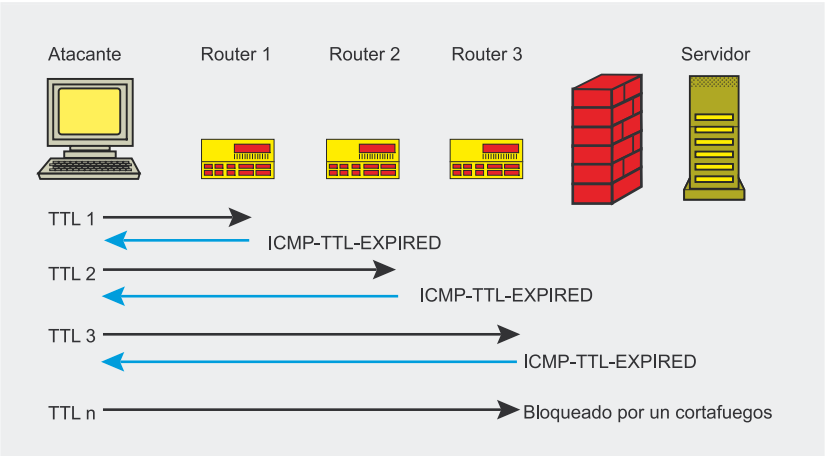


**Figura 1.** *Cómo funciona el traceroute*

**Listado 8.** *La comprobación de anuncios*

```
# netcat www.raptorfirewall.nix 80
> HEAD / HTTP/1.0
< HTTP/1.1 503 Service Unavailable
< MIME-Version: 1.0
< Server: Simple, Secure Web Server 1.1
< Date: Fri, 17 Sep 2004 19:08:35 GMT
< Connection: close
< Content-Type: text/html
< <HTML>
< <HEAD><TITLE>Firewall Error: Service Unavailable</TITLE></HEAD>
```

**Listado 9.** *Una exploración normal vs una exploración en el puerto de origen*

```
# nmap -sS -p 1-65535 192.168.0.1
Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) ↩
  at 2005-10-09 17:01 CEST
Interesting ports on 192.168.0.1:
(The 1658 ports scanned but not shown below are in state: closed)
PORT    STATE SERVICE
80/tcp open  http
Nmap run completed -- 1 IP address (1 host up) scanned in 6.607 seconds

# nmap -sS -g 80 -p 1024-65535 192.168.0.1
Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) ↩
  at 2005-10-09 17:01 CEST
Interesting ports on 192.168.0.1:
(The 1657 ports scanned but not shown below are in state: closed)
PORT    STATE SERVICE
80/tcp open  http
6000/tcp open X11
Nmap run completed -- 1 IP address (1 host up) scanned in 6.607 seconds
```

**Tabla 1.** *Los puertos abiertos que pueden ayudar a determinar el tipo de cortafuegos*

| Producto del Cortafuegos | Número del puerto | Objetivo |
|---|---|---|
| Cortafuegos de Symantec Enterprise | 888/TCP | OOB-Daemon |
| Cortafuegos de Symantec Enterprise | 2456/TCP | Administración basada en la Web |
| Checkpoint FW1-NG | 256/TCP | Administración |
| Checkpoint FW1-NG | 257/TCP | FW1_log |
| Checkpoint FW1-NG | 18181/TCP | Protocolo OPSEC del Contenido Vectorial |
| Checkpoint FW1-NG | 18190/TCP | Interfaz de administración |

se. Como podemos ver, *nmap* no fue capaz de determinar el sistema operativo y el producto del cortafuegos, pero el que tenga muchos puertos abiertos indica que este podría ser un cortafuegos basado en proxy y no hay muchos vendedores de esos productos.

Por tanto, además de la detección de huellas con *nmap*, también merece la pena echarle un vistazo a los puertos comunes de los diferentes productos de cortafuegos de los diferentes vendedores. Por ejemplo, el Cortafuegos de Symantec Enterprise (SEF) tiene dos puertos típicos, que son el 2456/TCP para la administración basada en la web y el 888/TCP para la Autenticación Fuera de Banda. La comparación de los resultados de la búsqueda con la Tabla 1 nos lleva un paso adelante en la definición del producto del cortafuegos. Por cierto, una capa de aplicación bien configurada como la SEF no tendrá tantos puertos abiertos en el cortafuegos externo. Un Cortafuegos Checkpoint 1 también tiene los típicos puertos abiertos que indican el producto del cortafuegos. Estos son, por ejemplo, los puertos administrativos en el rango 256–264/TCP y 18180–18265/TCP.

También vale la pena saber que *nmap* no es la única herramienta que puede ser utilizada para detectar un cortafuegos. Otras herramientas tales como *xprobe* y *p0f* también pueden ser utilizadas en la detección de un cortafuegos. Se puede encontrar más información útil en el Artículo *OS fingerprinting – ¿cómo no dejarnos reconocer?* publicado en la edición del 4/2004 de *hakin9*.

### La comprobación de anuncios

Para asegurarnos del producto de cortafuegos con el que vamos a trabajar, se puede utilizar una técnica de comprobación de anuncios para recibir más información. Por ejemplo, la salida de un demonio HTTP de un Cortafuegos de Symantec Enterprise puede ser identificada sencillamente por el Servidor: la cadena *Simple, Secure Web Server 1.1* (ver Listado 8).

Sin embargo, la salida misma del anuncio no es muy fiable, porque la mayoría de los demonios la pueden cambiar con facilidad. Pero, junto con la identificación de TCP y los números de puertos abiertos, es un buen indicador para determinar el producto del cortafuegos.

## La desviación de los cortafuegos

Cuando un intruso determina la existencia de un cortafuegos y su tipo, tiene varias posibilidades de desviar un cortafuegos. Vamos a observar métodos tales como el abuso de las listas de acceso mal configuradas,

## Los modos FTP activo y pasivo

El *File Transfer Protocol* (*Protocolo de transferencia de Archivos*) utiliza dos canales para la comunicación entre un cliente y un servidor. El canal de comandos se utiliza para enviar órdenes al servidor y respuestas al cliente. Si se transfieren datos o comunicación adicional, se establece el canal de datos. Los datos se transfieren, por ejemplo, si un archivo es descargado o bajado, pero también si se consulta una lista del directorio. Para establecer el canal de datos, el FTP admite dos modos, el FTP activo y pasivo. La diferencia entre los modos radica en quién establece el canal de datos.

En el caso de un FTP activo, el servidor FTP se conecta al cliente FTP. Por lo tanto, el cliente FTP le dice al servidor, a través del comando `PORT`, qué dirección IP y puerto se abrirá a la escucha para que acepte conexiones del servidor FTP. En el caso del FTP pasivo, el cliente FTP se conecta con el servidor FTP. Por ello el servidor FTP tiene que comunicarle al cliente FTP a qué direccion IP y puerto se puede conectar para establecer el canal de datos.

Para entrar en el modo pasivo, el cliente tiene que enviar un comando `PASV`. El servidor envía como una respuesta la información del socket al cliente en el formato `IP,IP,IP,IP,Hbyte,Lbyte` según lo cual el *Hbyte* y el *Lbyte* son los puertos para conectarse, y la dirección IP está separada por comas en lugar de puntos. Ver también Figuras 2 y 3.

conocidas como debilidades de protocolo y los fallos en los productos del cortafuegos, que pueden conducir al acceso no autorizado en un sistema protegido por cortafuegos.

### Los ataques del puerto de origen

Vamos a comenzar con los filtros de paquetes sencillos. Ellos toman sus decisiones analizando la cabecera IP o la TCP/UDP de cada paquete, mirando con frecuencia en la IP de origen, la IP de destino, el puerto de origen y el de destino de cada paquete, para decidir si enviarlo o bloquearlo.

Para crear una regla sencilla de acceso que permita a los usuarios de una red (interno) navegar por los sitios web de Internet (externo), necesitamos una regla para los paquetes de salida (la solicitud HTTP) y una regla para los de entrada (la respuesta del servidor web). Para crear una regla apropiada tenemos que saber que, por defecto, un servidor web basado en HTTP está escuchando en el 80/TCP y el puerto de origen elegido por el cliente HTTP (el buscador web) no es predecible, pero es generalmente mayor que 1024. La Tabla 2 muestra una lista de acceso mínima en un caso semejante.

A primera vista, este conjunto de reglas podría no parecer perjudicial. La regla 1 permite las consultas HTTP salientes y la regla 2 permite los paquetes de respuesta. La tercera regla está ahí para bloquear todo el tráfico restante, y por tanto se le llama la regla de limpieza. Sin embargo, una observación detallada de la regla 2 nos muestra que un paquete originado desde Internet (externo) y destinado a la red interna (interno) con un puerto de origen 80 y un puerto de destino mayor que 1024 pasa por el filtro de paquetes.

A este se le llama *puerto mayor* o un *ataque del puerto de origen*, pues el ataque está basado en el hecho de que un atacante sólo necesita modificar su cliente para utilizar un puerto reconocido como el 80/TCP como puerto de origen, para ser capaz de atacar servicios detrás de un cortafuegos que escucha en los puertos mayores. Algunos servicios interesantes de escucha en los puertos TCP mayores son el XWindow (6000–6063/TCP), el Windows Terminal Server (3389/TCP), y muchos puertos de aplicaciones web como el

**Tabla 2.** *La lista mínima de acceso para el tráfico HTTP*

| Nº | IP de Origen | IP de Destino | Puerto de Origen | Puerto de Destino | Acción | Descripción |
|---|---|---|---|---|---|---|
| 1 | Interno | Externo | >1024/TCP | 80/TCP | Permitir | Permitir la consulta HTTP que sale desde el cliente |
| 2 | Externo | Interno | 80/TCP | >1024/TCP | Permitir | Permitir la respuesta del servidor a una consulta HTTP |
| 3 | Cualquiera | Cualquiera | Cualquiera | Cualquiera | Denegar | Regla de Limpieza |

**Tabla 3.** *Un set de reglas de tráfico HTTP para un cortafuegos estático*

| Nº | IP de Origen | IP de Destino | Puerto de Origen | Puerto de Destino | Indicador ACK requerido | Acción | Descripción |
|---|---|---|---|---|---|---|---|
| 1 | Interno | Externo | >1024/TCP | 80/TCP | No | Permitir | Permitir la consulta HTTP que sale desde el cliente |
| 2 | Externo | Interno | 80/TCP | >1024/TCP | Sí | Permitir | Permitir la respuesta del servidor a una consulta HTTP |
| 3 | Cualquiera | Cualquiera | Cualquiera | Cualquiera | – | Denegar | Regla de limpieza |

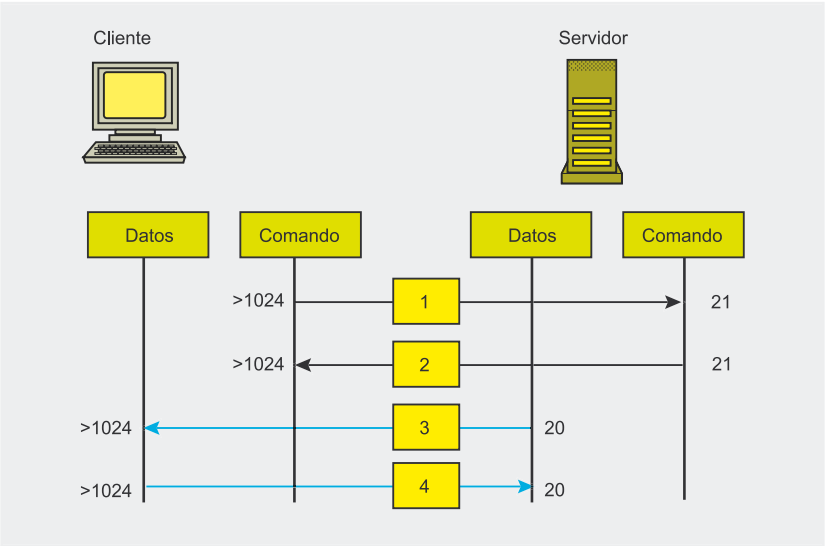


**Figura 2.** *Cómo funciona el FTP activo*

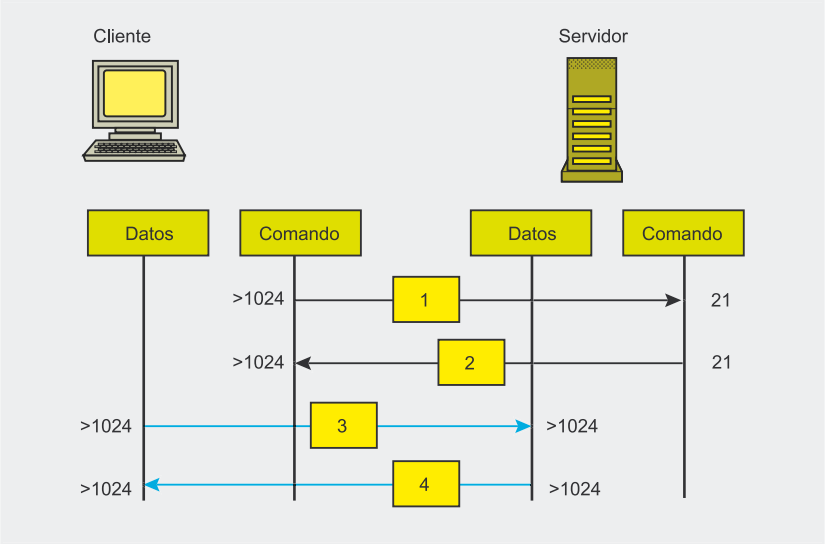


**Figura 3.** *Cómo funciona el FTP pasivo*

Jakarta Tomcat (8080/TCP) y el Bea Weblogic (7001/TCP).

Para comprobar si nuestro cortafuegos es vulnerable a este ataque, podemos utilizar *nmap* con la opción `-g`, diciéndole que utilice un puerto de origen definido. El Listado 9 muestra la diferencia entre la exploración normal y la del puerto de origen.

Como podemos observar, hay otro puerto abierto (el 6000/TCP) que fue detectado utilizando una exploración sencilla en el puerto de origen. No obstante, el intruso no es capaz de conectar con este puerto, al menos que el puerto de origen del cliente se sitúe en el 80/TCP.

El método más simple para establecer una conexión del puerto de origen es utilizar la FPipe del Foundstone. La FPipe es una herramienta de Windows, pero también funciona con Linux a través del Wine. Se ejecuta con las siguientes opciones:

```
> FPipe -l 100 -s 80 ↩
  -r 6000 192.168.0.1
```

que abrirán una escucha en el puerto local 100. Todos los paquetes enviados a este puerto alcanzarán el puerto de origen 80 y serán reenviados a la 192.168.0.1:6000.

Si ponemos a prueba un cortafuegos contra los ataques al puerto de origen, podríamos tener en cuenta la ejecución de nuestras pruebas con los puertos de origen 53 (DNS), 20 (FTP) y 88 (Kerberos), ya que por la naturaleza de estos protocolos algunos cortafuegos tienen escasas reglas de filtrado para ellos. Como ejemplo podemos citar una vez más al Checkpoint FW1, que hasta la versión 4.1 tuvo las llamadas *Reglas Implícitas*, que permiten el tráfico DNS de cualquier parte a cualquier parte.

La utilización por parte de Microsoft del Filtro IPSec que puede ser configurado como un cortafuegos local tiene una vulnerabilidad similar. Hay una regla de cortafuegos incrustada e invisible, que permite todo el tráfico entrante con el puerto de origen 88 (Kerberos). Para prevenir este ataque es necesario hacer cambios en el registro.

**Tabla 4.** *Un set de reglas para permitir el FTP activo*

| Nº | IP de Origen | IP de Destino | Puerto de Origen | Puerto de Destino | Indicador ACK requerido | Acción | Descripción |
|---|---|---|---|---|---|---|---|
| 1 | Interno | Externo | >1024/TCP | 21/TCP | No | Permitir | Canal de Comandos |
| 2 | Externo | Interno | 21/TCP | >1024/TCP | Sí | Permitir | Canal de Comandos |
| 3 | Externo | Interno | 20/TCP | >1024/TCP | No | Permitir | Canal de datos |
| 4 | Interno | Externo | >1024/TCP | 20/TCP | Sí | Permitir | Canal de datos |
| 5 | Cualquiera | Cualquiera | Cualquiera | Cualquiera | – | Terminar | Regla de limpieza |

## Los Cortafuegos Estáticos

Para impedir que un atacante establezca conexiones con los sistemas internos, estimulando así una respuesta a una consulta previa, es importante que un cortafuegos diferencie entre un paquete de respuesta y un paquete pensado para establecer una nueva sesión. Por tanto, el cortafuegos puede examinar los diferentes indicadores dentro de una cabecera TCP. Como cada nueva sesión TCP/IP comienza con un conjunto de indicadores SYN y todos los paquetes siguientes tienen como mínimo un conjunto de indicadores ACK, hay un atributo distinto para el cortafuegos. Además, la tabla de estado interna ayuda a seguir de cerca las sesiones, especialmente para la comunicación basada en UDP.

Como podemos observar en la Tabla 3, la respuesta de un servidor HTTP sólo será trasmitida si la cabecera TCP tiene el conjunto de indicadores ACK. En este caso, el ataque al puerto de origen ya no funcionará y el intruso tiene que buscar otras técnicas.

## El abuso de los FTP activos

Uno de los protocolos más utilizados en la comunicación de Internet es el *File Transfer Protocol* (FTP). Hay dos maneras diferentes en las que puede trabajar un FTP, de modo activo y pasivo (ver Recuadro *Los modos FTP activo y pasivo*). La diferencia principal entre los dos es la manera en que se establece la comunicación. Mientras está en modo activo, el cliente FTP establece el canal de comandos, y el servidor establece el canal de datos. En modo pasivo, ambos canales son establecidos por el cliente FTP.

El ataque contra el FTP activo es otro tipo de ataque del puerto de origen. En este caso, sin embargo, el FTP activo fuerza al cortafuegos a que permita los paquetes entrantes con un conjunto de indicadores SYN para el canal de datos (ver Tabla 4 para un ejemplo del set de reglas).

**Listado 10.** *La comunicación del FTP pasivo*

```
# nc ftp.hakin9.org 21
< 220-Welcome to hakin9.org.
> user anonymous
< 331 Please specify the password.
> pass secret
< 230 Login successful.
> pasv
< 227 Entering Passive Mode (192,168,200,23,230,242)
```

**Listado 11.** *La apertura de un puerto mediante el abuso del FTP pasivo*

```
# nc ftp.hakin9.org 21
< 220-Welcome to hakin9.org.
> user anonymous
< 331 Please specify the password.
> pass secret
< 230 Login successful.
> AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA227 ↩
  Entering Passive Mode (192,168,200,23,0,2)
< 500 command not understood: ↩
  'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA227 ↩
  Entering Passive Mode (192,168,200,23,0,22)'
```

Eso significa que incluso si el cortafuegos está examinando el indicador SYN, esto no impide que el intruso establezca una conexión desde el puerto de origen 20 hacia cualquier servicio mayor que 1024.

Para comprobar si un cortafuegos es vulnerable a este tipo de ataques, podemos utilizar *nmap* como hicimos en el último ejemplo, pero esta vez con la opción `-g 20` en vez de la `-g 80`. Para establecer una conexión con un servicio de puerto mayor, la FPipe se puede utilizar otra vez para modificar el puerto de origen.

## El abuso del FTP pasivo

La mayoría de los servidores FTP actualmente soportan el modo pasivo, pero por desgracia muchos clientes FTP no lo soportan (como el cliente FTP predeterminado de Microsoft). Sin embargo, incluso la utilización del FTP pasivo podría no ser suficiente para proteger un sistema contra el acceso indeseado a los sistemas internos. Observemos la comunicación FTP en el modo pasivo. Para mayor legibilidad utilizaremos un netcat para establecer la conexión (ver Listado 10).

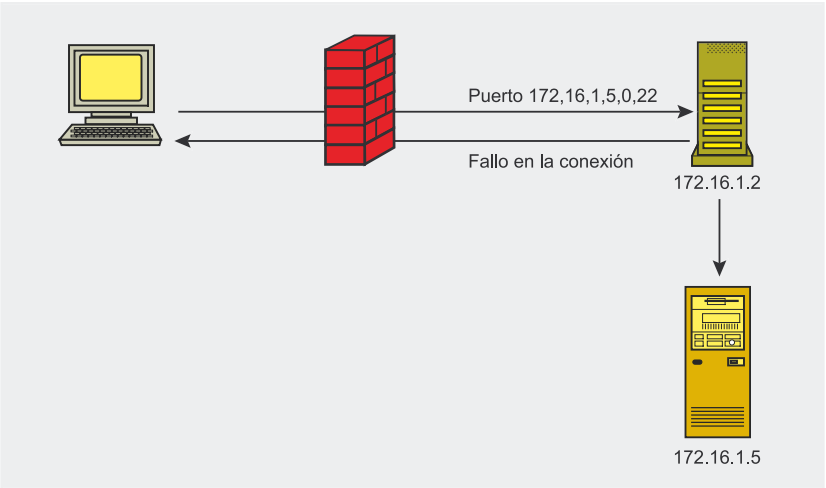


**Figura 4.** *Cómo funciona la exploración de rebote del FTP*

Las primeras seis líneas constituyen una comunicación FTP estándar para conectarse y logarse en un servidor FTP. En la séptima línea se le comunica al servidor FTP que utilice el modo pasivo para transferir datos. Como respuesta, el servidor FTP le indica al cliente (en la línea ocho) qué dirección IP y puerto se abre para aceptar una conexión que establezca el canal de datos.

Si hay un cortafuegos delante del servidor FTP, el cortafuegos no sabe qué puerto será escogido por el servidor FTP para el canal de datos. Por tanto, tiene dos opciones de personalización de su set de reglas para que permita la comunicación:

- La primera opción es abrirle al servidor FTP todos los puertos mayores para las conexiones entrantes. Esta opción no es buena, especialmente si uno tiene varios servidores FTP en la red, pues no es segura en absoluto.
- La segunda opción para el cortafuegos es analizar la comunicación entre el cliente FTP y el servidor. Si el cortafuegos ve un comando con formato `227 Entering Passive Mode (IP,IP,IP,IP,Hbyte,Lbyte)` dentro del canal de comandos enviados desde el servidor hacia el cliente, esta abrirá una regla temporal que permita una conexión entrante en la IP y puerto definidos en el mensaje.

Con semejante configuración uno puede engañar a un cortafuegos para que abra el puerto que uno escoja. Ya que los parámetros están en un formato `IP,IP,IP,IP,Hbyte,Lbyte`, enviado al cliente desde un servidor FTP dentro de un canal de control FTP, el intruso puede intentar forzar al servidor FTP para que envíe un mensaje hábil. Esto se puede llevar a cabo provocando un mensaje de error que contenga una cadena pasiva.

Si un comando no existente se le envía a un servidor FTP, en algunos casos se devolverá un mensaje de error que contenga el comando enviado, por ejemplo *command not understood AAAAAAAAAAA227 Entering Passive Mode 1,2,3,4,0,22*. Si comprobamos que el tamaño del mensaje de error es demasiado grande para un paquete IP, el paquete inexistente estará separado, mientras que la cadena del comando pasivo estará en el próximo paquete, entonces ya podríamos abrir un puerto adicional en el cortafuegos.

Si el cortafuegos lee el primer paquete que contiene los caracteres `A`, sencillamente lo pasará. Pero si lee la cadena `227 Entering Passive Mode (192,168,200,23,0,22)` creará una regla de permiso temporal para que el cliente FTP se conecte al puerto 22 del servidor 192.168.200.23. También se utiliza un mecanismo similar de creación de reglas de filtrado dinámicas para otros protocolos como el sqlnet de Oracle.

## La exploración de rebote del FTP

La exploración de rebote del FTP (ver Figura 4) utiliza funciones del FTP activo para explorar sistemas detrás de un cortafuegos. Dentro del FTP activo el servidor FTP establece un canal de datos mediante la conexión con un puerto abierto del cliente FTP. Debido a que el servidor no sabe en qué puerto el cliente está a la escucha del canal de datos, el cliente tiene que proporcionarle esta información al servidor dentro del canal de comandos.

Esto se hace a través del comando `PORT`. La sintáxis del `PORT` es `PORT IP,IP,IP,IP,Hbyte,Lbyte` como la `PORT 192,168,100,10,0,123`, que son similares a la del comando `PASV`. Con esta información, el servidor es capaz de establecer una conexión con la 192.168.100.10:123, si los datos deben ser transferidos.

Por definición, no hay restricciones para que la dirección IP sea la del cliente. Al contrario, es posible utilizar cualquier otra dirección IP en algunos servidores FTP. Después de emitir una orden como la `dir`, el servidor intenta conectarse a la IP definida como: puerto. Dependiendo del estado del puerto (abierto o cerrado), el servidor le devolverá al cliente un código de estado de éxito o de error. Analizar el código de

## La fragmentación

Cada sistema operativo intenta llevar el tamaño del paquete IP al máximo de tamaño de un cuadro de la tecnología fundamental de la capa 2. Para la Ethernet, la talla máxima es de 1518 bytes. A esta se le llama la *Unidad Máxima de Transferencia* (*Maximum Transfer Unit* o MTU). Debido a que la estructura misma de la Ethernet necesita 18 bytes para su información de cabecera, el espacio disponible para el paquete IP es de 1500 bytes.

En su recorrido por una red, un paquete IP puede pasar por un router que sólo es capaz de administrar paquetes más pequeños, debido a las limitaciones de su tecnología fundamental de la capa 2. Para poder enviar paquetes a través de semejante router, que tiene una MTU más pequeña que 1500 bytes, necesitamos dividir el datagrama IP en varios paquetes menores. A esto se le llama fragmentación.

En el extremo que recibe, el servidor de destino tiene que reunir todos los fragmentos IP y colocarlos en el orden correcto. A este proceso se le llama reconstrucción. El proceso de reconstrucción necesita de algunos datos para poder reunificar los paquetes en el orden correcto y no mezclar los fragmentos de diferentes conexiones destinadas al mismo servidor.

Hay dos campos principales dentro de la cabecera IPv4, elementales en el proceso de remontaje, que son la *Compensación de fragmentos* y *la Identificación* (ID). Cada fragmento del mismo datagrama tiene el mismo campo de ID. Esto permite que la pila IP reconozca todos los paquetes que pertenecen al mismo datagrama. Para colocar los paquetes en el orden apropiado se utiliza el campo de *Compensación de fragmentos*. El primer fragmento de un paquete tiene una compensación de cero. La compensación de cada fragmento siguiente aumenta en el valor de la longitud de la parte de los datos del fragmento. El bit *Más fragmentos* (MF) de la cabecera IP indica si le siguen más fragmentos o si el fragmento actual es el último.

estado permite que el atacante vea si los puertos están abiertos o cerrados. *nmap* admite la exploración de rebote FTP y puede utilizarse de la manera siguiente:

```
$ nmap -b \
  anonymous@myftpserver:21 \
  targetserver
```

## El rebote del proxy HTTP

Los cortafuegos de aplicación a menudo funcionan como proxies HTTP, ya sean trasparentes o no para el tráfico de filtrado HTTP. El problema de un proxy HTTP es que si no está bien configurado, puede permitir el acceso a los servidores internos.

La manera más fácil de probar si un cortafuegos es vulnerable al rebote de proxy es establecer la interfaz interna del cortafuegos como un proxy HTTP e intentar navegar en servidores web internos:

La configuración del proxy para lynx:

```
# http_proxy='http://myfirewall.de:8080'
# no_proxy='localhost'
# export http_proxy no_proxy
```

La navegación por sitios web internos:

```
# lynx 192.168.100.20
```

Un aspecto relevante sobre esta técnica es que incluso las direcciones IP privadas pueden estar disponibles desde el exterior, ya que el atacante sólo se conecta a la dirección IP oficial del cortafuegos y consulta al demonio HTTP para que se conecte al objetivo. Como el demonio HTTP también conoce las direcciones IP privadas internas, puede conectarse a ellas.

También merece la pena intentar obtener acceso a los diferentes puertos de los servidores internos:

```
# lynx 192.168.100.20:25
```

Sin embargo, algunos buscadores como el Mozilla Firefox están bloqueando por defecto esas consultas en el lado del cliente. Por tanto se recomienda comprobarlo con netcat o telnet.

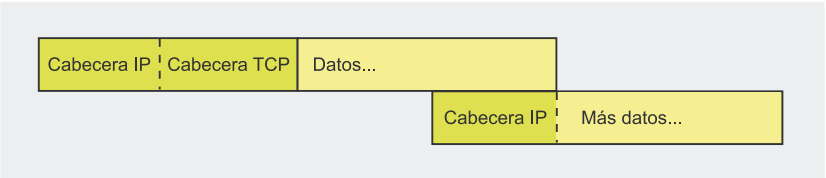


**Figura 5.** *El remontaje normal de los paquetes TCP*

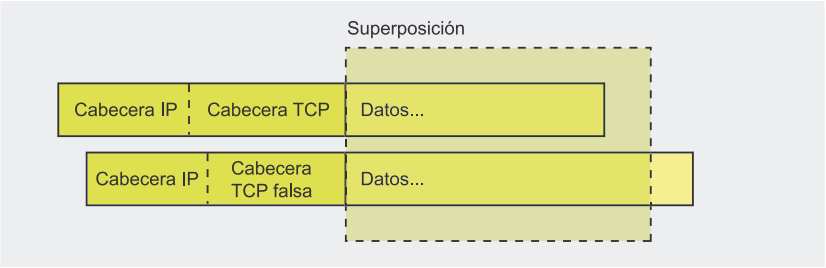


**Figura 6.** *El ataque de superposición de los fragmentos*

## El HTTP-Connect

La opción CONNECT de HTTP es normalmente utilizada como túnel para el tráfico SSL a través de un servidor proxy. El proxy, por tanto, sencillamente se abre para la sesión TCP entre el proxy y el servidor de destino y envía los datos del cliente. Por desgracia, algunos cortafuegos no comprueban la validez de las IPs y puertos de destino, y por lo tanto abren grandes agujeros que son utilizados por los atacantes.

Los cortafuegos deben der instalados de manera que los puertos administrativos sólo estén disponibles desde las interfaces internas. Esto debería impedir que los atacantes, por ejemplo, ejecuten exploits contra el demonio que se registra o que adivinen los logins y contraseñas del cortafuegos. La debilidad CONNECT permite a un atacante que establezca una conexión con la interfaz administrativa de las redes externas:

```
# nc firewall 8080
CONNECT 127.0.0.1:22 HTTP/1.0
SSH-1.99-OpenSSH_3.8p1
```

Mediante el uso del método CONNECT, un atacante también puede establecer conexiones a sistemas internos. Al igual que el ataque de rebote del proxy, este permite alcanzar los sistemas internos con direcciones IP privadas:

```
# nc firewall 8080
CONNECT 10.1.1.100:25 HTTP/1.0
220 mailserver ESMTP
```

Podemos ver claramente que la comprobación de las vulnerabilidades CONNECT en un cortafuegos es muy fácil. Se puede utilizar la misma técnica para obtener información sobre los rangos IP internos y explorar detrás de un firewall, de forma similar al ataque de la exploración de rebote FTP. El hecho de que en el pasado los productos líderes de cortafuegos, tales como el Checkpoint FW1 y el Astaro Secure Linux, fuesen vulnerables a los ataques de la HTTP-Connect es muy interesante.

## El ataque de superposición de fragmentos

El objetivo de un ataque de superposición de fragmentos es sobreescribir la información de la cabecera UDP o TCP después de que el cortafuegos haya tomado la decisión basada en el primer fragmento. Si la fragmentación ocurre dentro de la comunicación basada en TCP o UDP, sólo el primer fragmento IP contiene tal información como el puerto de destino de la cabecera TCP/UDP. Por ejemplo, si hay una regla de cortafuegos que permita las conexiones con el puerto 80/TCP

**Tabla 5.** *Algunos ejemplos de vulnerabilidades de cortafuegos*

| Producto | Vulnerabilidad |
|---|---|
| Checkpoint Secure Platform | Vulnerabilidad del Cruce de Reglas del Cortafuegos |
| Checkpoint VPN-1 | Exceso del Buffer ASN.1 |
| Checkpoint VPN-1 | Exceso del Buffer ISAKMP |
| Cisco IOS Firewall | Exceso del Buffer de Autenticación Proxy |
| Cisco Catalyst 6500/6700 | Vulnerabilidad del Cruce del FW Services Module ACL |

de un servidor web, pero rechace las conexiones al demonio de Secure Shell del mismo servidor en el puerto 22/TCP, se puede llevar a cabo un ataque de superposición de fragmentos.

El atacante está fragmentando un datagrama IP (ver Recuadro *La fragmentación*) y asignando el 80 como puerto de destino dentro de la cabecera TCP. El fragmento llega al cortafuegos y se corresponde muy bien contra la regla *Permitir*. Ya que todos los fragmentos IP de un datagrama tienen la misma IP e ID, el cortafuegos pasa a todos los fragmentos que siguen con la misma ID y la misma IP de origen y de destino que el primer fragmento.

La compensación del primer datagrama es cero y el extremo de este fragmento está por ejemplo en el byte 128. La compensación del segundo fragmento ahora debe tener el valor que le sigue directamente al byte 128. Si esta es menor que 128, una parte del primer fragmento será sobreescrita. A esto se le llama compensación negativa. Si el atacante calcula la compensación del segundo fragmento de manera que sobreescriba el puerto de destino de la cabecera TCP dentro del primer fragmento, es posible cambiar el valor del puerto de 80 a 22 (ver Figuras 5 y 6).

Después del remontaje completo, ya sea en el cortafuegos o en el host de destino, se establece la conexión con el puerto 22/TCP en vez de con el puerto 80/TCP. El cortafuegos se ha desviado con éxito.

Hay muchas maneras de utilizar los cortafuegos de ataques de fragmentación. Veamos en el Recuadro *En la Red* un vínculo a un ataque interesante contra el Filtro IP de BSD.

## Los ataques de Túneles

Los atacantes pueden querer comunicarse a través de un cortafuegos, por ejemplo, tener un caballo de Troya o una puerta trasera instalados en un sistema interno, que se comuniquen con el sistema del intruso. Este envía un comando al troyano y quiere que le reenvíen los resultados de los comandos.

Si las reglas de filtrado en un cortafuegos sólo permiten protocolos comunes como HTTP, FTP y DNS para el tráfico saliente, el atacante tiene que utilizar uno de estos protocolos para la comunicación. Por desgracia para el atacante, algunos sistemas de cortafuegos modernos están haciendo una comprobación de sintáxis RFC para el tráfico de capas de aplicación. Por lo tanto, si la comunicación no es compatible con RFC, será bloqueada por el cortafuegos.

Los intrusos que saben de esto, están realizando ataques de túnel con herramientas que no violan las definiciones RFC, envuelven los datos en comandos de protocolo válido. Si los datos envueltos además estan codificados y encriptados utilizando caracteres 7-bit ASCII, la detección mediante un cortafuegos es prácticamente imposible.

Los túneles basados en HTTP y DNS son buenos ejemplos. Mientras que las herramientas para los túneles HTTP compatibles con RFC como la rwwwshell (ver Recuadro *En la Red*) son relativamente fáciles de utilizar, y por tanto estarán disponibles durante muchos años, los túneles basados en DNS que están bien hechos son un poco más difíciles.

Un túnel DNS que utiliza una técnica llamada namedropping (entre otras) se basa en el *Protocolo del Servidor de Transporte* (*Name Server Transport Protocol* o NSTX) y necesita una compatibilidad NSTX del servidor y del cliente DNS, por lo que el servidor tiene que ser autoritario para un dominio (ver Recuadro *En la Red*). Vamos a imaginar que el atacante tiene autorización para el dominio *baddomain.com* y posee un sistema comprometido dentro de


## Sobre el autor

Oliver Karow trabaja como Consultor Jefe de Seguridad para un comercio dedicado a esta rama. Actualmente su trabajo está centrado en los cortafuegos, la tecnología IDS/IPS, las auditorías de seguridad y los tests de penetración. Oliver también está etudiando Tecnología de la Información en una universidad alemana a distancia. Trabaja en la IT desde 1994, y a partir de 1999 se ha dedicado a la seguridad IT.


## En la Red

- *http://cert.uni-stuttgart.de/archive/bugtraq/2001/04/msg00121.html* – Thomas Lopatic, *Un ataque fragmentario contra IP Filter*,
- *http://www.ccc.de/congress/2004/fahrplan/files/221-firewallpiercing_21c3.pdf* – Maik Hensche & Frank Becker – *Firewall Piercing – Explotación creativa de protocolos válidos de Internet*,
- *http://www.thc.org/download.php?t=r&f=rwwwshell-2.0.pl.gz* – HTTP implementación de túnel, rwwwshell,
- *http://www.csnc.ch/static/services/research/dnstunnel.html* – DNS implementación de túnel.

una red protegida por un cortafuegos. El atacante quiere ser capaz de controlar remotamente el sistema desde fuera, de enviar comandos y recibir respuestas.

Si el cliente quiere transferir datos al servidor, este consulta un nombre de host hábil como el *b2xpdmVyIGthcm93.baddomain.com*, en el que *b2xpdmVyIGthcm93* son los datos codificados. Debido a que el servidor interno de nombres no es responsable de este dominio, enviará la consulta al servidor NSTX del atacante. El servidor de nombres del atacante ahora puede extraer y descodificar el nombre de host de la consulta.

Para poder reenviar datos al cliente, el servidor de nombres del atacante coloca los datos en registros de recurso TXT. Es un registro de texto libre que puede ser utilizado con diferentes propósitos, por ejemplo para dar a conocer claves PGP públicas. Por tanto, no es fácil para un cortafuegos distinguir entre un registro TXT válido y un mensaje oculto de un troyano.

Para más información acerca de los ataques en túnel se recomienda leer el artículo *Firewall Piercing* (ver Recuadro *En la Red*).

## Las vulnerabilidades de Cortafuegos

La seguridad de una red falla si lo hace la seguridad de un cortafuegos. Si el mismo cortafuegos es vulnerable a los ataques como el exceso del buffer, la desviación puede hacerse sin problemas, pues un atacante puede reconfigurar el cortafuegos para sus necesidades. En el caso de una vulnerabilidad que le de a un atacante una cubierta de comando remoto, todos los ataques contra los sistemas internos se originarán de la dirección IP del cortafuegos. Si no hay un entorno de varios niveles de cortafuegos en el lugar, no habrá más protección para la red disponible.

Desafortunadamente, las vulnerabilidades ejecutables de manera remota en los productos líderes de cortafuegos son descubiertas muy a menudo. Sólo tienes que observar en *http://www.securityfocus.com/* para tener una visón general de las vulnerabilidades existentes (ver Tabla 5).

## Conclusión

Hay muchas maneras de desviar un cortafuegos. Algunas de ellas son consecuencia de las pocas habilidades del producto, otras se deben a la mala configuración, o a vulnerabilidades del mismo producto. Sin embargo, el despliegue de varios niveles de la tecnología estática de cortafuegos, así como las auditorías regulares del entorno de los cortafuegos pueden ayudar a establecer una buena protección para las redes internas. ●