Curso de Iptables

Por: Maikel Stinga Ruiz
Daniel Fernández Garrido

Índice

Configurar un firewall en Linux con iptables

Muchas son hoy en día las personas que se conectan, de una manera u otra, a Internet. Desde empresas que operan en la red hasta personas en sus casas que pasan un rato divertido navegando por sus páginas preferidas. Pero pocas de estas personas entienden realmente las consecuencias que tiene el abrir sus sistemas informáticos a Internet, unas consecuencias que no sólo son de carácter benigno e incluso beneficioso. El bien que obtenemos de Internet tiene un precio: Internet no es un lugar seguro.

Al igual que en cualquier sociedad, en Internet existen buenas intenciones, ayudas, compañerismo... pero también existen mentes perversas y llenas de maldad. En Internet existen personas decididas a hacer daño, pocas, pero es un hecho que existen, y debemos protegernos de sus acciones, por insignificantes que pensemos que somos. Es común entre los navegantes más o menos habituales de Internet, que nunca han tenido, o mejor dicho, creen que nunca han tenido un problema de seguridad en sus sistemas, el pensar que no es probable que lleguen jamás a recibir uno de estos ataques por el simple hecho de no poseer nada de interés, de no ser nadie importante. Esto es, claramente, falso.

Cualquiera puede ser presa de un ataque en la Red, cualquiera, por insignificante que se pueda pensar que uno es. Es

precisamente esa sensación de sentirse a salvo la que hace que sea este tipo de gente el que tome, por lo general, las menores precauciones, y por ello, al mismo tiempo, que se conviertan en la presa más apetecible para aquéllos que simplemente desean hacer daño, por el placer de hacerlo. Como ejemplo, valgan los sorprendentes datos recogidos por mí mismo como usuario de un proveedor de servicios Internet (ISP) común en España, detectando los intentos de atacar el puerto TCP 80 (servidor web) de mi ordenador mientras estaba conectado a Internet, puerto que había dejado abierto intencionadamente (aunque, naturalmente, protegiendo mi servidor web) para guardar un log de los ataques que se intentaban llevar a cabo. Los datos son los siguientes:Fechas: del 19 de Septiembre al 21 de Noviembre de 2001
Promedio de horas de conexión diarias: 1,5 horas.
Intentos de ataque al puerto 80/tcp: 87
Lo cual nos da una idea del peligro que corre un usuario cualquiera de Internet que no tome las precauciones mínimas, teniendo en cuenta que soy alguien tan insignificante como cualquier otro en la Red y que, de no ser porque deseaba hacer ese estudio, posiblemente no hubiese podido detectar dichos ataques, y por ello seguiría considerándome seguro.
Como dato, el 100% de los 87 ataques eran destinados a servidores Microsoft Internet Information Server o Microsoft Personal Web Server (afortunadamente yo tengo Apache), y se trataba de intentos de ejecución de scripts malignos, de intentos de ejecución de cgi's peligrosos y de explotar algún tipo de buffer overflow en parámetros de algunos scripts de estos servidores.
Por tanto, una vez visto que el peligro existe, es la hora de hablar de qué es un firewall.
Un firewall es, por lo general, un software (puede ser también un equipo hardware dedicado) a través del cual nos conectamos a una red como Internet, y que sirve como filtro

sobre el tráfico que por él pasa, en ambas direcciones, y que en un momento dado puede rechazar cierto tráfico en alguna de las direcciones.
Eso quiere decir que, mediante un firewall, podemos detectar el tráfico no deseado hacia nuestros sistemas, y en general, los posibles ataques de que seamos objeto. De esta manera podremos aislar nuestros equipos del exterior, permitiendo nuestro uso de Internet de manera absolutamente normal pero minimizando en lo posible la probabilidad de padecer las consecuencias de un ataque.
Así pues, ante la pregunta ¿Necesito un firewall? queda ya suficientemente patente que la respuesta es, sin lugar a ninguna duda, sí.
Este artículo cubrirá la configuración de un típico firewall doméstico, que permita conectarse a Internet de una manera segura y cerrar los puertos TCP y UDP que nos puedan causar problemas.

¿Qué es iptables?

iptables es la herramienta que nos permite configurar las reglas del sistema de filtrado de paquetes del kernel de Linux, desde su versión 2.4 (en 2.2 era ipchains). Con esta herramienta, podremos crearnos un firewall adaptado a nuestras necesidades. Su funcionamiento es simple: a iptables se le proporcionan unas reglas, especificando cada una de ellas unas determinadas características que debe cumplir un paquete. Además, se especifica para esa regla una acción o target. Las reglas tienen un orden, y cuando se recibe o se envía un paquete, las reglas se recorren en orden hasta que las condiciones que pide una de ellas se cumplen en el paquete, y la regla se activa realizando sobre el paquete la acción que le haya sido especificada. Estas acciones se plasman en los

que se denominan targets, que indican lo que se debe hacer con el paquete. Los más usados son bastante explícitos: ACCEPT, DROP y REJECT, pero también hay otros que nos permiten funcionalidades añadidas y algunas veces interesantes: LOG, MIRROR...En cuanto a los paquetes, el total del sistema de filtrado de paquetes del kernel se divide en tres tablas, cada una con varias chains a las que puede pertenecer un paquete, de la siguiente manera.

- - filter: Tabla por defecto, para los paquetes que se refieran a nuestra máquina
- - INPUT: Paquetes recibidos para nuestro sistema
- - FORWARD: Paquetes enrutados a través de nuestro sistema
- - OUTPUT: Paquetes generados en nuestro sistema y que son enviados
- - nat: Tabla referida a los paquetes enrutados en un sistema con Masquerading
- - PREROUTING: Para alterar los paquetes según entren
- - OUTPUT: Para alterar paquetes generados localmente antes de enrutar
- - POSTROUTING: Para alterar los paquetes cuando están a punto para salir
- - mangle: Alteraciones más especiales de paquetes
- - PREROUTING: Para alterar los paquetes entrantes antes de enrutar
- - OUTPUT: Para alterar los paquetes generados localmente antes de enrutar

Dado que el soporte para el firewall está integrado en el kernel de Linux (Netfilter), para poder usar iptables tendremos que asegurarnos de que nuestro núcleo admite el uso de iptables y que añadimos a la configuración del núcleo todos aquellos targets que vayamos a necesitar (aunque siempre es bueno tener los más posibles).

Características del firewall a crear

Para crear nuestro sencillo firewall doméstico, tendremos primero que preguntarnos qué es lo que deseamos que haga.
Lo más usual, en un equipo que se usa para conexiones a Internet de manera normal (no es servidor de nada, etc...)
es que deseemos de nuestro firewall lo siguiente:

- Que nos permita realizar conexiones TCP hacia afuera de nuestra máquina (si no, no podríamos hacer casi nada).
- Que no permita realizar conexiones TCP desde afuera hacia nuestra máquina, para evitar que alguien intente conectarse a nuestros servidores web, ftp, telnet, X...
- Que permita el tráfico de paquetes TCP (paquetes que no establezcan conexiones) en ambas direcciones, pues necesitamos tráfico bidireccional de paquetes al usar casi cualquier cosa en Internet.
- Que Prohíba el tráfico UDP desde afuera de nuestra máquina, a excepción del necesario para las respuestas por parte de nuestros servidores DNS, que provendrán de su puerto UDP 53.
- En caso de tener una intranet, que no aplique estas restricciones al tráfico proveniente de y enviado hacia la intranet, ya que en esta red interna probablemente sí nos interese poder acceder remotamente a nuestra máquina.

Uso básico de iptables

Para crear nuestro firewall, necesitaremos ejecutar algunos comandos básicos sobre iptables, como:Para crear una
nueva regla al final de las ya existentes en una chain determinada:

```
$ /sbin/iptables -A [chain] [especificación_de_la_regla] [opciones]
```

Para insertar una regla en una posición determinada de la lista de reglas de una chain determinada:

```
$ /sbin/iptables -I [chain] [posición] [especificación_de_la_regla] [opciones]
```

Para borrar una regla en una posición determinada de la lista de reglas de una chain determinada:

```
$ /sbin/iptables -D [chain] [posición]
```

Para todas las reglas de una chain determinada:

```
$ /sbin/iptables -F [chain]
```

Para listar las reglas de una chain determinada:

```
$ /sbin/iptables -L [chain]
```

La especificación de reglas se hace con los siguientes parámetros (especificando aquellos que se necesite):

- -p [protocolo]: Protocolo al que pertenece el paquete.
- -s [origen]: dirección de origen del paquete, puede ser un nombre de host, una dirección IP normal, o una dirección de red (con máscara, de forma dirección/máscara).
- -d [destino]: Al igual que el anterior, puede ser un nombre de host, dirección de red o dirección IP singular.
- -i [interfaz-entrada]: Especificación del interfaz por el que se recibe el paquete.
- -o [interfaz-salida]: Interfaz por el que se va a enviar el paquete.
- [!] -f: Especifica que la regla se refiere al segundo y siguientes fragmentos de un paquete fragmentado. Si se antepone
!, se refiere sólo al primer paquete, o a los paquetes no fragmentados.

Y además, uno que nos permitirá elegir qué haremos con el paquete:
- -j [target]: Nos permite elegir el target al que se debe enviar ese paquete, esto es, la acción a llevar a cabo con él.
Algunas de las opciones que se permiten en los comandos de arriba son:
- -v: Modo verboso, útil sobre todo con iptables -L.
- -n: las direcciones IP y números de puertos se mostrarán numéricamente (sin resolver nombres).
- --line-numbers: Muestra los número de regla de cada regla, de manera que sea más fácil identificarlas para realizar operaciones de inserción, borrado...

Creación del firewall

Para crear nuestro firewall, iremos introduciendo una a una las reglas que necesitamos:Primera regla: permitiremos cualquier tráfico que provenga de nuestro interfaz de loopback (lo), para ello insertaremos en el chain INPUT (que se encarga de los paquetes que llegan con destino a nuestra máquina), de la tabla filter la siguiente regla:

```
$ /sbin/iptables -A INPUT -i lo -j ACCEPT
```

Atención: es importante aquí respetar las mayúsculas, pues los nombres del chain y del target son INPUT y ACCEPT, no input o accept. Segunda regla: si disponemos de intranet, permitiremos todo el tráfico que provenga de nuestro interfaz de red interna. Por ejemplo, imaginando que tuviésemos una ethernet en el interfaz eth0, haríamos:

```
$ /sbin/iptables -A INPUT -i eth0 -j ACCEPT
```

El hecho de que omitamos la dirección de origen, de destino... implica que nos referimos a todas. Tercera regla:
impediremos el paso de cualquier paquete TCP proviniente del exterior que intente establecer una conexión con nuestro
equipo. Estos paquetes se reconocen por tener el flag SYN asertado y los flags ACK y FIN desasertados. Para decirle a
la regla que reconozca específicamente estos paquetes, usaremos una opción que se puede usar cuando el protocolo del
paquete es declarado como tcp, la opción --syn. De la siguiente manera:

```
 $ /sbin/iptables -A INPUT -p tcp --syn -j REJECT --reject-with icmp-port-unreachable
```

Y vemos también el uso de una opción del target REJECT, que nos permite elegir de qué manera debe ser rechazado el paquete. Posibles valores son icmp-net-unreachable, icmp-host-unreachable, icmp-port-unreachable, icmp-proto-
unreachable, icmp-net-prohibited y icmp-host-prohibited.Cuarta regla: Antes de declarar que deseamos prohibir cualquier tráfico UDP hacia nuestra máquina, y dado que las reglas se recorren en orden hasta que una de ellas se activa con el paquete, tendremos que añadir ahora una regla que nos permita recibir las respuestas de nuestro/s servidor/es DNS cuando nuestro sistema les realice alguna consulta. Estas respuestas, vía UDP, saldrán del puerto 53 del servidor DNS. La regla, pues, será:

```
 $ /sbin/iptables -A INPUT -p udp --source-port 53 -j ACCEPT
```

Donde --source-port es una opción presente cuando el protocolo es udp (también cuando es tcp) y nos permite en este
caso especificar que la consulta provenga del puerto destinado al DNS.Quinta regla: Prohibimos ahora el resto del
tráfico UDP. La regla de por sí implica a todo el tráfico UDP, pero como un paquete sólo activará esta regla si no ha
activado la anterior, los paquetes UDP referentes a una

transacción con un servidor de nombres no se verán afectados.

```
$ /sbin/iptables -A INPUT -p udp -j REJECT --reject-with icmp-port-unreachable
```

Dado que los targets por defecto (denominados policy o política) en la tabla filter son ACCEPT, si un paquete no activa ninguna de las reglas, será aceptado, de manera que no tendremos que preocuparnos de, por ejemplo, los paquetes de tráfico normal de TCP, ya que estos serán aceptados al no activar regla alguna.Si ahora escribimos:

```
$ /sbin/iptables -L -v
```

Deberíamos obtener algo como:

Chain INPUT (policy ACCEPT 3444 packets, 1549K bytes)

pkts bytes target prot opt in out source destination
11312 3413K ACCEPT all -- lo any anywhere anywhere
0 0 ACCEPT all -- eth0 any anywhere anywhere
0 0 REJECT tcp -- any any anywhere anywhere tcp flags:SYN,RST,ACK/SYN reject-with icmp-port-unreachable
0 0 ACCEPT udp -- any any anywhere anywhere udp spt:domain
0 0 REJECT udp -- any any anywhere anywhere reject-with icmp-port-unreachable

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)

pkts bytes target prot opt in out source destination

Chain OUTPUT (policy ACCEPT 15046 packets, 4218K bytes)

pkts bytes target prot opt in out source destination

De modo que nuestro pequeño y básico firewall doméstico ya está configurado. Nuestro equipo es ahora muchísimo más seguro, puesto que los ataques a nuestro sistema requerirían ahora mucha más elaboración, tiempo y esfuerzo por parte del

atacante, de manera que nuestra condición de insignificantes ya empezará a ser importante como garante de seguridad.

Guardar y reusar nuestra configuración de iptables

Pero, si una vez realizadas estas configuraciones, apagásemos nuestro equipo, todo esto se perdería, y tendríamos que volver a realizar una a una las sentencias de configuración. Para evitar esto, iptables cuenta con dos programas auxiliares: iptables-save e iptables-restore, el primero de los cuales nos permite sacar por salida estándar el contenido de nuestras tablas IP, y el seguno nos permite, a partir de la salida generada por iptables-save, recuperar la configuración de las tablas. De manera que para volcar la configuración de nuestro firewall en un fichero ejecutaremos:

```
$ /sbin/iptables-save -c > [fichero]
```

Donde -c es una opción que nos permite guardar los contadores del número de paquetes que activaron cada regla.Y, cuando queramos, podremos recuperar la configuración del firewall con:

```
$ /sbin/iptables-restore -c < [fichero]
```

En cuyo caso -c tiene el mismo significado que con iptables-saveEstas llamadas a iptables-save e ipatbles-restore podrán ser incluidas en los scripts adecuados para que se lleven a cabo de manera automática en el arranque y el cierre del sistema.En caso de ser usuarios de Red Hat Linux, a partir de su versión 7.1, una vez configurado el firewall con iptables tal y como se ha descrito en este artículo, y una vez salvada la configuración con iptables-save en el archivo /etc/sysconfig/iptables, se pueden activar los scripts que arrancarán y cerrarán el firewall automáticamente al arrancar y apagar el equipo, mediante la Text Mode Setup Utility (/usr/sbin/setup), en la sección System Services.