IPTables, el cortafuegos de Linux, forma parte de una extensión del kernel llamada NetFilter. Podríamos decir que NetFilter e IPTables forman un equipo fabuloso. Para explicarlo a lo NiPeGun: NetFilter Gómez Mancuernas es un portero de discoteca todo ciclado que anda todo loco esperando que le den órdenes para hacer su trabajo. Lo único que quiere el señor Gómez es analizar y filtrar gente y se deprime si no le dan instrucciones. IPTables, por otro lado, es el jefe de NetFilter Gómez. En realidad es el dueño de la discoteca. En definitiva, es el que le da las órdenes a NetFilter. Le dice al portero quien debe pasar y quien no. Incluso también le dice que cuando venga un latino, ni lo eche ni lo deje pasar, sino que directamente lo mande a otra discoteca que también es suya pero en la que ponen reggaeton. Y esto es como todo en la vida, el jefe le puede dar todas las instrucciones que quiera, pero el que realmente realiza el trabajo es el portero llamado NetFilter Gómez Mancuernas.

Hay muchas y variadas formas gráficas de que IPTables configure la «hoja de instrucciones» para darle a NetFilter. Incluso las hay completamente manejables con el ratón. Pero normalmente sólo te dejan configurar reglas predefinidas y no gozan de la flexibilidad habitual de la línea de comandos. Además, como siempre me gusta decir, cuando aprendes a interactuar con la linea de comandos te haces una idea más acertada de que es lo que ocurre con los cambios que estás haciendo. Por eso, siempre es preferible manejar IPTables desde la cli (o interfaz de línea de comandos, como prefieras). Además, en el caso de querer proteger un servidor sin entorno grafico, no te servirá de nada saber manejar IPTables sólo gráficamente.

Las reglas de IPTables se «meten» en principio ordenadas en tres tablas predefinidas. La tabla **filter**, la tabla **nat** y la tabla **mangle**. Y digo en principio, porque existe la posibilidad de crear otras tablas además de esas tres. Pero con esas tres normalmente se satisfacen el 99% de las necesidades. Las reglas las metes en esas tres tablas usando cadenas y conceptos. Si te acuerdas siempre de **TABLAS-CADENAS-CONCEPTOS** tendrás siempre presente en la memoria como crear reglas. Pero antes de entender las cadenas y los conceptos vamos a ver cuales son esas tres

TABLAS

- **FILTER**: Es la tabla donde pones las reglas para decidir si bloqueas un tipo de paquete o si, por el contratrio, le permites continuar su camino. Toooodos los paquetes pasan por esta tabla. Todos toditos. Tiene 3 cadenas predefinidas: INPUT, OUTPUT y FORWARD.
- **NAT**: Es la tabla donde metes las instrucciones para compartir una IP pública con muchos otros dispositivos. Normalmente configuras esta tabla cuando quieres usar tu distro como router. Los paquetes que pasan por esta tabla irán a donde le específiques en sus reglas. Tiene 4 cadenas predefinidas: INPUT, PREROUTING, POSTROUTING y OUTPUT
- **MANGLE**: Es la tabla donde configuras el QoS, las opciones de los paquetes y donde configuras además su marcaje para la posterior destrucción. Algo así como el matadero de vacas. Contiene absolutamente todas las cadenas predefinidas. INPUT, OUTPUT, FORWARD, PREROUTING y POSTROUTING.

Como decía antes, una vez que ya sabes en que tabla tienes que poner tu regla, tienes que aprender como funcionan sus respectivas

CADENAS

- **PREROUTING:** Aunque estén en la table nat y en la mangle, en ambas se comportan diferente. Básicamente esta cadena se usa para los paquetes que llegan desde la red hasta el ordenador que está ejecutando iptables. tráfico entrante, justo antes de ingresar a la pila de red del kernel. Las reglas en esta cadena son procesadas antes de tomar cualquier decisión de ruteo respecto hacia dónde enviar el paquete
- **INPUT:** Todos los paquetes que tienen como destino al ordenador.tráfico entrante, luego de haber sido ruteado y destinado al sistema local.
- **FORWARD:** Todos los paquetes que no se originan en el ordenador pero que tampoco lo tienen como destino, sino que están de “paso” a través de él. Normalmente esta cadena se usa cuando usas el ordenador como router.ráfico entrante, luego de haber sido ruteado y destinado hacia otro host (reenviado).
- **OUTPUT:** Todos los paquetes que se originan en el ordenador.tráfico saliente originado en el sistema local, inmediatamente luego de haber ingresado a la pila de red del kernel.
- **POSTROUTING:** tráfico saliente originado en el sistema local o reenviado, luego de haber sido ruteado y justo antes de ser puesto en el cable.

CADENAS ORDENADAS POR ORDEN DE PASO POR LAS TABLAS

PREROUTING:

1. raw
2. mangle
3. nat (DNAT)

INPUT:

1. mangle
2. filter
3. security
4. nat (SNAT)

FORWARD:

1. mangle
2. filter
3. security

OUTPUT:

1. raw
2. mangle
3. nat (DNAT)
4. filter
5. security

POSTROUTING:

1. mangle
2. nat (SNAT)

las cadenas se evalúan en el siguiente orden:

Tráfico entrante destinado al sistema local:

1. PREROUTING
2. INPUT

Tráfico entrante destinado a otro sistema (paquetes que se deben reenviar):

1. PREROUTING
2. FORWARD
3. POSTROUTING

Tráfico originado desde el sistema local:

1. OUTPUT
2. POSTROUTING

El objetivo de las cadenas es poder controlar cuándo, a lo largo del flujo de un paquete a través del sistema y la pila de red, una regla es evaluada.

PROCESOS

- Direcciones IP
- Protocolos (tcp, udp, icmp)
- Puertos

Con todos los conceptos que se explicaron arriba se crean las

REGLAS

Una vez que creas una regla, ésta se agrega a la tabla que le hayas indicado y se pone en marcha inmediatamente. A partir del momento en el que la hayas creado todo el tráfico que pasa por el sistema empieza a filtrarse con esa regla. Pero claro, seguramente esa no será la única regla que tengas en las tablas. Las reglas se agregan una a una y se van ejecutando por turno empezando por la que esté más arriba. Si el paquete que está siendo procesado encaja con una regla específica, el paquete es procesado según la acción que marca la regla y ya no es procesado por las otras reglas en la cadena.

Si el paquete pasa por todas las reglas de la cadena y llega hasta abajo sin haber coincidido con ninguna regla, entonces netfilter toma la acción marcada por la

POLÍTICA POR DEFECTO

La política por defecto puede configurarse como aceptar (ACCEPT) o descartar (DROP) dependiendo de las necesidades de cada sistema. De hecho, eso es lo más importante a la hora de configurar el sistema. Si configuras una politica de descarte (DROP) descartas todos los paquetes que entran menos los que coincidan con alguna regla que hayas configurado para ser aceptados. Si configuras una politica de paso (ACCEPT), el sistema dejará entrar todos los paquetes menos los que específicamente marques como para que no pasen.
La política DROP es más segura, pero tardarás mucho más tiempo en configurarla porque, hasta que no hagas los cambios, los programas que usen la red no te funcionarán.
La política ACCEPT es más a prueba de fallos, porque funcionará todo. Pero no sabrás sobre un problema de seguridad que puedas tener en la red hasta que sea demasiado tarde

Para entenderlo a modo de resúmen: Cada uno de los paquetes de datos que quieran entrar, salir o pasar por nuestro sistema Linux es identificado, marcado, analizado, aceptado, rechazado, redirigido por NetFilter (siempre que se lo hayas mandado hacer). Y NetFilter cumplirá las órdenes al pie de la letra y hará con esos paquetes exactamente lo que le has ordenado. Por lo que, si eres bueno dando órdenes, tendrás con Linux el sistema más seguro que te hayas podido echar a la cara.