# Iptables Firewall

Transcribed on July 30, 2025 at 4:00 PM by Minutes AI

Speaker 1 (00:01)

Bienvenidos a esta nueva sesión.

En esta sesión vamos a tratar el tema de iptables.

En esta primera parte veremos los conceptos básicos así como los comandos más utilizados.

Iptables es un cortafuego o firewall completo que funciona bajo sistemas operativos Linux, de hecho prácticamente viene de serie en cada distribución de Linux, es decir, podemos utilizarlo como un cortafuegos de red o para securizar un servidor, dependiendo del hardware que estemos utilizando.

Iptables es un cortafuego basado en reglas para controlar el tráfico de entrada y de salida.

Iptables se basa en tablas, las cuales gestionan cadenas que a su vez tienen unas reglas concretas.

Podemos encontrar tres tipos de cadenas que también se denominan chains.

Tenemos input, que es la cadena que controla las conexiones de entrada, lo que se llama el inbound.

Tenemos forward, esta cadena controla las conexiones entrantes que están dirigidas de forma local.

Y finalmente tenemos output, que es la cadena para la gestión del tráfico de salida.

Antes de continuar os quiero dar algunos consejos básicos para el hardening de un cortafuegos.

Sería por ejemplo, algunos lo conoceréis porque son genéricos, como realizar un backup de la configuración.

Si nos centramos en iptables tenemos que también indicar la versión y hacer también un backup de la configuración.

El orden de las reglas es jerárquico, por lo tanto al principio tenemos que colocar las reglas más específicas y las más genéricas.

Al final tenemos que utilizar listas blancas o whitelist para que cualquier grupo de usuarios pueda acceder, pero claro, por ejemplo el equipo de IT, gente que esté autorizada a gestionar el cortafuego, lo podríamos hacer a través de whitelist.

También tenemos que establecer una política por defecto y esta va a ser drop, que es parar todo el tráfico.

También tenemos que estudiar cada caso de regla y abrir solamente aquellos puertos que sean necesarios.

También tenemos que limitar el número de conexiones, esto es muy importante porque es la mejor solución para evitar ataques de denegación de servicio distribuido o dos.

Y también tenemos que hacer un mantenimiento de las reglas que estamos creando, no podemos dejar las que estamos haciendo porque irá aumentando el tamaño del fichero de configuración y realmente no nos harán falta e incluso puede ser un problema potencial de seguridad en el futuro, así que tenemos que borrar aquellas reglas que no utilicemos.

Vamos a ver ahora desde un punto de vista práctico los conceptos básicos que nos van a permitir comprender cómo administrar el tráfico de red de manera efectiva veremos algunos ejercicios básicos que nos van a ayudar a demostrar los diferentes aspectos y funcionalidades que tiene iptables.

La primera de ellas será mostrar las reglas actuales de iptables.

Para ello hacemos este comando iptables l que nos da esta salida.

Este comando lo que hace es listar todas las reglas actuales en todas las input, forward y output.

La salida muestra las políticas predeterminadas para cada cadena y cualquier regla específica aplicada.

En este caso vemos que no hay ninguna.

Bien, vamos a ver qué IP tengo en esta máquina.

Aquí la vemos, tenemos la IP 10.211.55.5 será la máquina Ubuntu, será la primera VM.

Bien, pues aquí estoy en la otra máquina y hago lo mismo y vemos la IP que tenemos, que tenemos la 17.

10.

211.

55.

17 pues bien, entre estas dos máquinas vamos a ir jugando, vamos a ir viendo las diferentes opciones que podemos hacer para bloquear tráfico y hacer muchas operaciones basándonos en iptables.

Bien, vamos a comprobar que hay conectividad entre ambas máquinas.

Si yo hiciera un ping a la 211.

555 que es la otra máquina, que es la unidad, vemos que hay respuesta, con lo cual hay conectividad entre ambas.

Ambas se ven Bien, pues vamos a ver ahora cómo podemos bloquear todo el tráfico entre ambas máquinas virtuales, que no haya ningún tipo de conectividad entre ambas.

Para eso utilizaremos estos dos comandos.

Estoy en la máquina 2, haremos sudo ip tables y le decimos guión usamos la chain input s ahora pondremos la dirección ip de la máquina 1.

Estamos en la 2, la máquina 1 hemos visto que es la 10.

211.55.5 damos guión j y ponemos el comando drop.

Ahora vemos que es cada uno de ellos.

Bien, ponemos la contraseña sería primero y hacemos otros ip tables, esta vez también le ponemos guión a, pero ahora ponemos output la otra cadena Y ahora haremos otra vez la ip y haremos lo mismo j qué es lo que hemos hecho ahora, porque en estos dos comandos lo que hace es bloquear todo el tráfico en entre la máquina 1 y la máquina 2 y también el tráfico saliente, de ahí input y output.

Lo que hace a es que añade la regla al final de la cadena y s lo que hace es especificar la dirección ip de origen y cuando veáis el guión d es la dirección ip de destino.

Finalmente el guión j es cuando se define la acción y en este caso es drop que es para el, que es detener.

Bien, pues en este momento no tendría que haber ningún tipo de comunicación entre ambas máquinas, si yo hiciera otro pin ahora a la 10.

211.

55.5 veis que ya operación no permitida, quiere decir que no hay conectividad, el cortafuegos ip tables está bloqueando la conexión, por ejemplo vamos a permitir el tráfico ssh, algo bastante habitual y que hemos comentado antes en el bastión host y todo eso, y lo que es la gestión de los dispositivos de red, siempre hacerlo por ssh.

Bien, pues para habilitar el acceso ssh desde la máquina 1 haremos el siguiente comando sudo ip tables como antes a de file chain pues le ponemos la input que es la de entrada y le decimos p y le decimos tcp y ahora deport, después os explico al final de cada comando os explicaré cada una de las acciones que hace.

Y ahora ponemos guión s, aquí ponemos la dirección y la dirección ip 10.211.555 que es la máquina 1, que es la remota que tengo en mi pequeña red j lo que hace es asignar la operación, en este caso aceptar, si os acordáis antes hicimos un drop que era para, ahora vamos a abrir, aceptar.

Bien, con esto la entrada ya la tenemos asociada a ssh.

Bien, vamos a hacerlo también con la otra chain que es el output, ahora cuento cada una, guión D mismo IP 10211.55.5 y en este caso lo mismo ACCEPT.

Bien, estos dos comandos lo que van a hacer es permitir el tráfico ssh, que es el puerto 22, como podéis ver en la definición de port y s port, y esto lo que hace que está permitiendo el tráfico desde y hacia la máquina 1, la máquina 1 ahora ya está autorizada en este equipo, que antes no había conectividad con nadie, esta vez ya se le está aceptando que desde fuera se conecte la ip 10.211.55.5 pero sólo al puerto ssh.

El guión p lo que hace es especificar el protocolo, en este caso tcp dport es el puerto de destino y S port es el puerto de origen, al final es el mismo, el 22 es SSH.

Bien, pues voy a la máquina 1, estamos en la 10.211.35.5 que es la máquina que acabo de autorizar a que conecte con ssh en remoto.

Bien, pues ahora deberíamos de poder conectar con la máquina remota, si ponemos ssh y ponemos el usuario, la dirección IP, Bien, no conectamos, quería que vierais esto, la importancia del orden en las reglas.

Antes hemos bloqueado todo el tráfico y después hemos habilitado el ssh, por este motivo no podemos conectarnos ahora.

En este caso lo que tenemos que hacer primero es habilitar el ssh y después bloquear todo para que la última regla que se aplica es la de bloquear todo, que será la principal, pero ya heredé la anterior que es la DSH.

Bien, volviendo a la máquina 2, vamos a ver ahora cómo están las reglas aplicadas.

Hacemos l, podemos ver que tenemos primero el drop en el que todo se bloquea y después se aceptan las conexiones, este puede ser el problema, vamos a hacerlo al revés, vamos a hacer antes, vamos a limpiar todo ahora y habilitar sh y después bloquear, porque la última de las reglas es la que se pone la primera dentro de todo lo que es la jerarquía.

Bien, para eliminar todas las reglas utilizaremos este comando sudo i petables f esto resetea, digamos que limpia todas las cadenas, todas las chains, input, output y forward.

Hacemos esto y miramos otra vez el listado, vemos que ya no tenemos ninguna regla aplicada, así que ahora vamos a hacerlo todo al contrario, ahora haremos primero la habilitación de sch, haremos m table a input p era tcp y la ip que queremos habilitar en este caso sería la 102.11.555 J AC y ahora hacemos prácticamente lo mismo, solo que esta vez, acordaros, hay que cambiar aquí por guión d s, aquí sería guión d en vez de port sería s port, la ip es la misma ac total, vale, bien, todo es igual y aquí output, Vale, iptables a Con esta opción lo que hacemos es lo que ya vimos antes también la salida, si hacemos el iptables l como antes, vemos que ya hemos añadido ambos del ssh y están habilitados.

Bien, pues ahora sí vamos a bloquear todo el tráfico, hacemos sudo iptable a acordaros que era input la primera, guión s, bloqueamos el tráfico de la máquina última, estamos bloqueando la máquina remota que es la 10.211.555 y ahora igual, sólo que esta vez utilizaremos el output guión d, que destino ahora destino.

Bien, si hacemos otro listado de las diferentes reglas, aquí la tenemos.

Primero está el SSH y después está el drop, aunque el drop será el primero que se va a bloquear, pero ya hemos habilitado el SSH.

En definitiva, el último de los comandos que tenemos que poner es el que lo bloquea todo.

A partir de ahí, antes hay que ir definiendo lo que vamos a ir abriendo y al final decimos, vale, bloqueamos todo.

Un poco al contrario de lo que hemos hablado desde la jerarquía, pero realmente tiene sentido.

Primero vamos a ver qué es lo que necesitamos y después cerramos todas las conexiones hacia esta máquina o hacia lo que sea.

Esto puede variar según qué tipo de cortafuegos utilices, pero siempre piensa que es algo jerárquico, que siempre tiene que haber un bloqueo genérico de todo y a partir de ahí ir abriendo aunque sea en un orden o en otro, pero hay que hacer un bloqueo global.

Bien, si ahora vuelvo a la máquina anterior, intento hacer otra vez la conexión SSH, que era la 17.

¿Esta vez sí nos deja, veis?

Ya no deja conectar, ya nos pediría al usuario y tal.

Con esto ya haríamos la conexión.

Bueno, si hacemos así, lo que hacemos es aceptar el certificado y ya solamente habría que poner la contraseña de la máquina remota.

Fijaros que sólo está habilitado el SSH.

Si yo hiciera un ping a la 10.211.55.17 No está funcionando.

¿Por qué?

Por el drop.

El drop lo ha bloqueado todo.

El drop genérico que hemos hecho al final lo ha bloqueado todo y sólo tenemos abierta la conexión SSH.

Es la única que nos va a permitir conectar el SSH.

Bien, ya que estamos hablando de SSH, podríamos incluso limitar aún más algún tipo de parámetro o facto de SSH.

Por ejemplo, una cosa que podríamos hacer es limitar la tasa de conexiones de SSH, es decir, limitar los intentos de conexión.

Por ejemplo, podemos poner que sean menos de cuatro en un minuto.

Así tenemos todavía un control más granular y más específico de las conexiones hacia nuestro servidor.

Bien, vamos a ver cómo se hace.

Es un poco más largo, hay que meter más información, pero ya veréis hasta qué punto de control nos puede dar.

Y petables con esto.

Imaginad que queremos hacer lo que acabo de decir, limitar los intento de conexión a SSH a menos de cuatro en un minuto, menos de cuatro en 1 minuto, entonces si hubiera seis conexiones dentro de ese minuto no le dejaría conectar, haría una especie de cola y dejaría solo conectar a menos de cuatro antes.

Recordar una cosa, hemos hecho el drop aquí, ¿Verdad?

Este drop hay que ponerlo al final.

Entonces vamos a ver cómo podemos eliminar ese drop, ya que afecta a toda la conectividad y poderos hacerlo en orden.

Primero habilitamos SSH, después ponemos la limitación que acabo de decir para el ssh y después ya es cuando bloqueamos todo.

Bien, eliminar las reglas específicas es bastante sencillo, es cambiar solamente el guión A por guión D.

Bien, pues para eliminarlo es solo poner guión D, por ejemplo pondríamos guión D en el output y D también en el input.

Con esto ya eliminaríamos las 2 cadenas.

Bien, para ver que lo hemos hecho correctamente hacemos un sudo irritable l y ya veremos que han desaparecido los drop que hemos utilizado antes.

Bien, aquí quiero actualizar que no hace falta eliminar las reglas, también las podemos ir moviendo en orden.

Por ejemplo, podemos utilizar el comando ip table de esta forma.

¿Qué es lo que haría esto?

Esto haría la regla que está en la posición 3 la pondría en la posición 1 y la regla que estaba en la posición 1 se desplazará 1 hacia abajo.

Por lo tanto podríamos haber utilizado esto para mover el drop que lo para todo y moverlo hacia abajo y dejarlo al final, no haría falta ir eliminando, pero bueno, quería que vierais cómo se eliminan reglas concretas dentro de esa jerarquía.

Ahora volveremos a utilizarlo, pero lo que vamos a hacer es lo que hemos comentado, vamos a añadir esas dos reglas que hemos dicho de ssh, si hago un sudor ip table guión a y le ponemos input 22, seguimos con ssh y ahora usamos guión m state new m set ahora os digo que es cada uno de los parámetros, no os preocupéis.

Haciendo esto tenemos el input y ahora hacemos iptables a input también va a poner ahora el contador p tcp deport 22 mstate state new y ahora lo que hacemos es m reset y ahora update second 60 hitcount jdr fijaros qué comando más largo, lo he hecho queriendo, quería buscar un comando complejo para que vierais la cantidad de definiciones que se puede hacer o hasta qué nivel de detalle se puede llegar a controlar las conexiones, en este caso en 60 segundos, 4 como mucho 3, depende como lo queramos poner.

Si hago esto, ya hemos añadido la regla.

Bien, si listamos las reglas con inspectables L ya podemos ver que la hemos añadido tanto en el input como en el output, entonces, bueno, en este caso sería solamente.

Ya podemos ver que la hemos añadido tanto en el input como en el output, entonces, bueno, en este caso sería solamente en el input, entonces porque estamos recibiendo llamada, no hay salida desde nuestra máquina hacia afuera, solo estamos cambiando reglas que están asociadas al input de la máquina a la entrada.

Bien, despejado todo, para explicaros un poco lo que hacen los dos comandos de forma breve, porque claro, aquí hay mucha documentación, hay muchos comandos y sería muy complejo explicar todo.

Bien, pues en la primera línea, esta de aquí, lo que estamos haciendo es digamos, agregar esa primera fase de la limitación que hemos hecho de las conexiones, por ejemplo, bueno ya sabéis lo que hace el input, lo que hace el guión p, lo que hace el guión ddport, que es el puerto destino, pero me centraré en los parámetros que no conocéis, como por ejemplo mstate y state new, esto lo que hace es utilizar un módulo que se llama state que especifica que la regla se aplicará solo a conexiones nuevas.

Después tenemos el guión m reset send, que esto realmente lo que hace utilizar modos reset, que lo que hace es establecer una marca de tiempo para la dirección IP, M reset set todo ese bloque lo que hace es justamente eso, crear una marca de tiempo, y esto marca la dirección IP como reciente en la tabla de conexiones recientes.

Después el segundo comando, bueno conocéis el guión a guión p, otra vez aparece el guión m state new, que se aplicará solo a conexiones nuevas, eso lo que significa ese comando.

También tenemos el mresent, pero esta vez pone update, porque lo que hace es utilizar el módulo resent para actualizar la información sobre la dirección IP de origen que ya estaba almacenada en la tabla reciente que hemos visto antes en el comando anterior.

Bien, pues después pondríamos 60, y este especifica el periodo de tiempo en segundos, en este caso 60, durante el cual se va a rastrear las conexiones.

Hitcount establece el número máximo de conexiones permitidas durante el periodo de tiempo que hemos especificado, que eran en este caso serie 4.

Y finalmente el guión jdrop, ya sabéis lo que es, pero en este caso lo que pasa es que si se supera el límite de conexiones permitidas, la regla envía la acción DROP, lo que significa que se van a descartar todos los paquetes de conexión posteriores de esa dirección, y esta es la clave para hacer toda esa franja de tiempo que hemos dicho dentro de un rango de tiempo que eran 60 segundos solamente permitir cuatro conexiones.

Resumiendo, la primera regla establece una marca de tiempo para la dirección IP de origen cuando se inicia una conexión nueva SSH entrante en el puerto 22, y la segunda regla verifica si una dirección IP ha intentado más de cuatro conexiones SSH en los últimos 60 segundos.

Si este límite se supera, los paquetes de conexión posteriores de esa dirección IP serán descartados, lo que ayuda a mitigar ataques de fuerza bruta contra el servicio SSH y también a mitigar ataques de denegación de servicio.

Bien, pues ya sólo nos quedaría ahora sí añadir el DROP que hemos quitado antes.

Bien, pues ya tenemos algunas reglas ya configuradas que son importantes, este fichero irá creciendo y petables irá cambiando, tenemos que ver la opción para poder guardar las reglas que estamos creando, esto es interesante ya que así nos aseguramos de su persistencia de la configuración cuando se reinicie el sistema pues se podrán restaurar, con lo cual es importante utilizar este comando una vez que ya tenemos una serie grande de diferentes reglas y no las queramos perder o incluso antes de hacer alguna operación que puede llevar algún bloqueo, alguna prida de alguna de las reglas anteriores.

Bien, almacenar las reglas es sencillo sudo ip tables y ahora ponemos save y simplemente ahora le decimos dónde queremos almacenar, le decimos donde lo queremos almacenar, en este caso sería etc lunes v Esta sería la forma de almacenarlas, si queremos restaurarlas pues sería en vez de poner shape pondríamos restore y esta vez cambiamos el orden del pi y ahora directamente ponemos la dirección de ansetables y con esto restauraríamos una configuración anterior.

Evitables es una herramienta muy utilizada debido a que está en cualquier distribución Linux como hemos contado antes, pero es que además es un nivel bastante alto de profesionalidad, ya que se puede aplicar prácticamente cualquier tipo de filtro utilizando iptables, de hecho encontraremos iptables en cantidad de dispositivos comerciales, por eso tener mucho conocimiento, tener un conocimiento amplio de iptables nos puede sacar de muchos problemas, además de ayudar muchísimo a la hora de securizar nuestra arquitectura, ya que iptables al filtrar paquetes, también intentar evitar accesos no autorizados, también pueden mitigar una gran variedad de amenazas que están asociadas a la red.

Y todo esto hace que iptable sea un elemento crítico en la arquitectura de una red.

Y por defecto un cortafuegos lo ves también.