IPTables DMZ

Transcribed on July 30, 2025 at 5:01 PM by Minutes AI

Speaker 1 (00:04)

Bienvenidos a esta nueva sesión, en esta sesión vamos a tratar el tema de iptables pero un poco más avanzado y también haremos un ejercicio de cómo crear o cómo implementar una DMZ usando iptables.

Bien, vamos a hacer ahora unos ejercicios un poco más complejos y comenzaremos con uno de ellos en el cual vamos a utilizar una máquina como gateway de la otra, es decir, mi máquina número uno que es la 10.211.555, esta máquina tiene acceso a Internet y va a actuar como gateway para la máquina 2, bien, primero desde la máquina 1 que es en la que estoy ahora mismo, voy a utilizar o configurar NAT para permitir que la máquina 2 que tiene la IP 10.211.55.17 comparta la conexión a Internet de la máquina número 1, buscamos esta línea y la descomentamos, esta es la que nos va a permitir ese reenvío, almacenamos, aplicamos el cambio utilizando sudo sysgtl p El primer paso es preparar el servidor número 1, la máquina principal, pues hay que habilitar el enrutamiento, esto permite que la máquina 1 reenvíe paquetes entre interfaces, para ello tenemos que editar el fichero que vemos a continuación que es sudo el que está en etc sysctl conf escribo este comando nand y ahora ponemos a postrouting ahora os comento que es cada uno de ellos postrouting o aquí ponemos el nombre de la interfaz de red principal, en mi caso es think j masquerade Bien, ¿Qué es lo que hace este comando?

Este comando lo que va a hacer es ocultar las direcciones IP internas haciendo parecer que todo el tráfico sale desde la máquina 1, bien lo aplicamos, ponemos nuestra entrada de root y ya lo tenemos.

Bien, ahora hay que configurar la máquina 2 para que utilice la máquina 1 como gateway predeterminado, esto lo podemos hacer en la configuración de red y especificamos la IP de la máquina 1, que era la 10.211.555 como gateway, hay muchas formas de asignar la dirección IP, una por ejemplo aquí podéis ver un comando que lo haría directamente con eso lo configuraría.

Después habría que establecer el gateway predeterminado, esto lo podemos hacer con este comando default y vía 211.35 ya con esto asignaríamos la máquina que acaba en 5 que sería la 1 como el gateway principal.

También es interesante editar el fichero de configuración de la DNS, que os lo voy a poner porque también es interesante conocerlo, que este es el de SOL CO y aquí podríamos añadir abajo alguna más, por ejemplo, y aquí meter las de Google para probar.

Estamos probando, así que.

Y pondríamos la 8 8 4 4 y ya tenemos.

Pero ya os digo, hay muchas formas de hacer esto, yo directamente hacerlo, en mi caso es sencillo haciéndolo así, pero lo podías hacer desde incluso la interfaz gráfica de Ubuntu o como quieras.

Lo primero que vamos a hacer es un ping, por ejemplo a Google, vemos que está funcionando, pero estamos saliendo a través de la máquina 1, desde la 2 a través de la máquina número 1.

Vamos ahora otro ejercicio que está relacionado con el control de acceso y la seguridad y en concreto vamos a bloquear el ping.

Ping utiliza el protocolo ICMP y este es importante bloquearlo porque en la fase de descubrimiento de un posible atacante este protocolo podría dar mucha información.

Ya conocemos el comando nmap por ejemplo, y con nmap podríamos sacar muchísima información de los puertos utilizando ese comando.

Bien, pues vamos a bloquear los pin entrantes en la máquina número 1, pues hacemos sudo ip tables a input bueno, aquí ya lo conocemos todo, aquí se asigna el protocolo ICMP, ahora os comento esto.

Type echo request jdrop Vale, es sencillo, además es intuitivo, le decimos que con guión p que el protocolo ICMP lo vamos a bloquear con la última parte que es la guión jdrop, pero también estamos especificando qué tipo de respuesta queremos bloquear y es el eco request.

Ahí le decimos exactamente qué es lo que queremos hacer.

Bien, antes de aplicar esa regla de antes, vamos a ver que el pin funciona desde la máquina 2, tenemos un pin a la 102.11.45.5 vemos que es correcto, está funcionando y ahora sí aplicamos ese bloqueo de ICMP o del ping.

Vale, está aplicado, vamos a ver si ahora funciona el P.

Antes metimos la operación de pin y vemos ya que no está ofreciendo ningún tipo de respuesta.

Pues bien, este comando sencillo de iptables nos ofrece una gran seguridad por lo que he dicho, bloquea cualquier tipo de entrega a la información que se base en el protocolo ICMP.

Vamos a ver cómo podemos revertir ese comando utilizando lo que dijimos en el anterior vídeo de las posiciones de las reglas, para ver cómo están las reglas asignadas en el servidor, pero viendo los números de líneas que son importantes para poder hacer ese cambio de ubicación, usamos el comando sudo ip TABLES I input y le decimos line numbers Veis aquí veis ya todo lo que hemos puesto, de momento, como no tenía ninguna

Bien, pues con esto si podemos alistarlas, ya ha desaparecido, la hemos eliminado, pero no por concepto, sino por ubicación o posición dentro de la jerarquía de las reglas, y por supuesto si le hacemos otro pin, pues debería dar respuesta sin ningún problema.

Otro ejercicio interesante que podemos aplicar es directamente redirigir el tráfico que llega a un puerto y redirigirlo hacia otro.

Por ejemplo, todo el tráfico que llegue al puerto 80-80 en la máquina 1, lo podemos redireccionar al puerto 80 de la máquina 2.

¿Cómo se podría hacer esa opción?

Pues de nuevo IPTABLES guión T ahora os explicaré paso a paso todos los parámetros.

NAT a PREROUTING, este ya lo hemos utilizado anteriormente PCP Dport 8080 muchos de los parámetros ya os tienen que sonar juntas de NAT, son nuevo destination 10211 55 17 con el puerto 80.

Bien, os cuento paso a paso qué son estos parámetros.

Lo primero, este comando de IP tables configura una regla en la tabla NAT para realizar la escritura de direcciones en lo que está haciendo es modificando NAT para redireccionar ese puerto, como os He dicho del 80-80 al puerto 80.

El IPTables, bueno el comando prerouting lo que hace es que agrega la cadena PREROUTING a tabla NAT y esta cadena se aplica antes de que se realice el enrutamiento, y esto lo que hace es que permite modificar paquetes antes de que se procesen, esto es importante.

PTCP lo que hace es especificar el protocolo DPORT especifica la regla que se aplicará a los paquetes que tengan como puerto de destino el puerto 8080.

Después tenemos J de NAT, indica que los paquetes coinciden con esta regla, por lo tanto se aplicará una acción de reescritura de la dirección de red.

Destination native Finalmente vemos el guión guión to destination con la ip de destino y el puerto 80.

Esto especifica la nueva dirección de destino y el puerto al que se enviarán los paquetes.

En este caso se indica que se van a enviar a la dirección IP que veis ahí con el puerto 80.

Resumiendo, esta regla de IP tables va a redireccionar todo el tráfico TCP que llegue al puerto 8080 de la máquina hacia la dirección IP 10.

211.

55.

17 en el puerto 80.

Esto permite la dirección o la redirección de tráfico desde un puerto externo a un puerto interno específico en otra máquina de la red.

¿Y esto es exactamente lo que hace un NAT un R.

Bien, vamos a hacer ahora otro ejercicio bastante interesante, esta vez lo que vamos a crear es una DMZ, os acordáis?

Es una zona desmilitarizada.

Lo que veremos aquí es cómo utilizar IP tables para segmentar la red dirigiendo el tráfico específico a ciertas máquinas.

Para entenderlo mejor aquí tenéis este diagrama.

Aquí podemos ver que tenemos la conexión a Internet hacia la red externa, esta es la zona de Internet, este es el acceso al Internet global.

Tenemos el firewall o el gateway que sería la 10211.55.5.

Esto actúa como el punto de entrada para el tráfico dirigido a la DMZ, filtrando las solicitudes y permitiendo solo aquellas dirigidas al puerto 80 del servidor web.

Este gateway está dentro del área que no es DMZ, pero actúa como el controlador hacia la zona DMZ.

El siguiente elemento es el servidor web, este sí que está ubicado en la DMZ y representa el servidor web accesible desde la red externa, es lo único que se puede acceder.

Aquí veis un poco todo el bloqueo que hemos aplicado en genérico que bloquea SSH y cualquier otro servicio, solo habilita el que hemos indicado.

Bien, ahora lo siguiente que haremos es preparar digitables para nativos.

Bien, que esto ya lo hemos hecho antes, pero vamos a hacerlo de nuevo enfocando a este ejercicio.

Hacemos un sudo ip tables t nat a postrouting Ya sabemos lo que hacía el postrouting o más que la e, ahora os cuento lo que es cada uno.

Este comando biptable lo que hará será configurar una regla en la tabla NAT para realizar un enmascaramiento.

Bien, pues los comandos iptab no lo conocemos, el guión t nat también lo hemos visto que es utilizar la tabla NAT, el guión a postrouting también lo hemos visto, que lo que hace ya sabéis que se aplica después de que se realice el enrutamiento de los paquetes.

Después vemos el guión oeth, aquí se especifica la red de salida, que en mi caso es la eth, Esta regla se va a aplicar solo al tráfico saliente, o sea al tráfico que sale a través de la, es decir, al tráfico que sale a través de la interfaz de red ETH.

Y finalmente vemos el guión j masquerade aquí se indica que si los paquetes coinciden con esta regla se aplicará una opción de masquerading o de enmascaramiento.

El enmascaramiento es una técnica NAT que modifica la dirección IP de origen de los paquetes salientes para que parezca que provienen del enrutador o firewall que realiza el enmasquerade o el enmascaramiento en lugar de la IP real que tienen los sistemas de la red interna.

El siguiente paso será crear las reglas que hemos pensado antes para la DMZ, en este caso era que permita el acceso a un servidor web que está en el puerto 80, por ejemplo, pues lo que haríamos sería aquí sudo iptables a forward p tcp de port 80 el puerto de destino guión d la dirección IP de la máquina que está en la TMZ j acept Bien, pues este comando lo que va a hacer es permitir el tráfico HTTP hacia la máquina 10.211.55.17 que es mi servidor cliente digamos.

Y ahora viene un paso importante que es bloquear todo lo demás.

Con esto nos aseguramos que el tráfico no deseado hacia y desde la máquina 2, la máquina cliente, esté bloqueado.

Pues para ello hacemos un sudo ip tables a forward d 211 55 17 j drop y ahora igual para aplicarlo en ambos sentidos, solo que esta vez cambiaremos el d por la s Y ahora igual para aplicarlo en ambos sentidos, solo que esta vez cambiaremos el d por la s.

Bien, pues con esto ya tenemos ese bloqueo del tráfico no deseado y vamos a hacer una prueba.

Bien, para hacer la prueba lo que voy a hacer en la máquina 2, que es la máquina cliente que está detrás de la DMZ.

Voy a levantar un servicio en el puerto 80 para que podamos hacer todo el proceso y para ello utilizaré Python con este comando sencillo que lo que hace es levantar un proceso en el puerto 80 que lo que va a hacer simplemente es mostrar el contenido de los directorios en el que estoy ahora mismo.

Bien, si abro el navegador y pongo localhost 80 localhost 2 80 vemos el listado de los ficheros que tengo yo en esta carpeta.

Vemos el listado de los ficheros que tengo yo en esta carpeta.

Bien, para ver si todo está correcto podemos hacer una prueba muy sencilla desde la máquina 1 nos vamos al servidor, ponemos un curl y ponemos la dirección 10211 55 17 esto debería devolver el listado correcto, pues con lo cual esto está funcionando y también lo que no nos dejaría es hacer una conexión por ejemplo ssh debido al bloqueo que hemos puesto genérico de todo todas las conexiones que no sean las que están

Si yo ahora hiciera un ssh user y pusiera la dirección IP vemos que no rechaza la conexión, con lo cual está todo bloqueado excepto ese acceso al puerto 80 que hemos comentado antes.

Como conclusión podemos decir que iptables demuestra las capacidades avanzadas de control de red, ya que ofrece filtrado de paquetes muy granular y mecanismo de enrutamiento sofisticados mediante su conjunto de funciones que es muy completo y una gestión de reglas robustas.

Iptables es un pilar fundamental para la seguridad avanzada de redes y también para la optimización del rendimiento, pero tenemos que resaltar su gran adaptabilidad a cualquier tipo de arquitectura de red.

Llegamos al final de la sesión, os esperamos en el siguiente vídeo.