Engineer

Cortafuegos

reglas que deciden qué tráfico de red puede pasar

Qué es

Un cortafuegos es un filtro en una ruta de red que compara cada paquete o conexión con una lista de reglas y después lo deja pasar, lo descarta o responde rechazándolo. Puede ser software en la máquina protegida, un dispositivo entre dos redes o un servicio del proveedor de nube delante de una máquina virtual. Los administradores Linux suelen llamarlo también filtro de paquetes, y los proveedores venden cortafuegos con nombres como grupo de seguridad. En plural suelen formar capas: uno en el proveedor y otro en el host.

La guía del NIST SP 800-41 Rev. 1, de septiembre de 2009, los clasifica según la información del tráfico que examinan. Un filtro de paquetes mira direcciones, protocolo y números de puerto, paquete a paquete. Un cortafuegos con estado también recuerda conexiones, de modo que la respuesta a una solicitud del propio servidor puede entrar sin regla propia. Una pasarela de proxy de aplicación actúa como extremo de la conexión y lee el protocolo que esta transporta. Un cortafuegos basado en host se ejecuta en el servidor o portátil que protege y sabe qué programa posee un socket.

En un servidor Linux, el filtrado se hace en netfilter, el marco del kernel. nftables, disponible desde el kernel 3.13, sustituye a las antiguas herramientas iptables. Interfaces como ufw escriben reglas para quienes prefieren expresar intenciones en lugar de cadenas. Sea cual sea la herramienta que las escriba, el modelo es el mismo: una lista ordenada que se lee desde arriba y donde decide la primera regla que coincide.

A menudo se confunde el cortafuegos con herramientas vecinas. Un cortafuegos de aplicaciones web lee solicitudes HTTP y bloquea patrones de ataque; uno de red nunca profundiza tanto. Una herramienta de bloqueo de intrusos lee registros y añade reglas temporales contra direcciones que se comportan mal,

por lo que depende de un cortafuegos subyacente. Una VPN decide quién está en la red. Cada uno responde a una pregunta distinta y ninguno sustituye la lista básica de reglas.

Utilice el concepto en una decisión

Aplique «¿Qué es un cortafuegos?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Cuentas, contraseñas y permisos organizados para que cada persona tenga exactamente el acceso que necesita y lo pierda el mismo día que deja la empresa.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. 1.000 reglas RBAC para servidores Linux — 1,000 reglas de acceso escritas para cerrar una brecha de seguridad

1.000 reglas RBAC para servidores Linux Póngase en contacto →

Por qué importa

Cada servicio que escucha en una dirección pública es una puerta, y el cortafuegos es el inventario de las que existen. La decisión real del equipo tiene tres partes: dónde está, qué permite de forma predeterminada y quién puede cambiarlo.

El cortafuegos del proveedor detiene el tráfico antes de que llegue a la máquina, sigue funcionando si alguien obtiene root en el host y no consume CPU del servidor. Sus reglas viven en una consola o API fuera de la configuración del servidor, así que divergen si no se escriben como código. El del host viaja con la máquina, puede limitar la frecuencia del tráfico de un puerto concreto y se puede probar desde ella misma, pero una cuenta root comprometida puede reescribirlo, igual que el software que gestiona sus propias reglas. Usar ambos es habitual y razonable, al precio de sincronizar dos listas: cada puerto abierto o cerrado supone un cambio en dos lugares.

La segunda parte es el valor predeterminado. El NIST recomienda denegar todo tráfico no permitido expresamente, y casi todos coinciden para el entrante. El filtrado saliente es donde pesa la contrapartida: limita lo que puede alcanzar un servicio comprometido, pero rompe las actualizaciones de paquetes, la renovación de certificados y la monitorización hasta que cada destino tiene su regla.

La tercera parte es la responsabilidad. Un cortafuegos que deniega de forma predeterminada solo puede estar bien configurado si la lista de puertos declarados es correcta. El resultado útil es esa lista, conservada en el control de versiones y contrastada con las reglas activas. El software es gratuito en toda distribución Linux mayoritaria; lo que cuesta dinero es la atención necesaria para mantener la declaración ajustada a la realidad. Un cortafuegos es además una segunda línea de defensa. Un servicio ligado a la dirección de bucle local no es accesible desde fuera, digan lo que digan las reglas. Eso da una garantía mayor que una regla de denegación delante de un servicio que escucha en todas las interfaces.

Cómo se aplica en la práctica

En un host, un límite de frecuencia de conexiones SSH permaneció un año en las reglas sin recibir un solo paquete, porque una autorización más amplia situada encima coincidía primero. Las herramientas de estado mostraron el límite todo el año, con razón: existía. Las comprobaciones se reescribieron para leer la posición de las reglas y no su presencia: ahora comparan el orden de las reglas numeradas y de la cadena con la que filtra realmente el kernel frente a lo previsto. Cada puerto tiene una regla y cada regla indica un protocolo; se limita la frecuencia de conexiones SSH y no hay una regla de autorización aparte. Los cinco puertos públicos se declaran juntos y se comprueban como conjunto, de modo que uno abierto que falte en la declaración hace fallar el control. El informe de seguridad enumera las reglas en el orden en que las encuentran los paquetes y explica cómo leerlo.

El primer servidor se creó a mano y recibió el cortafuegos después, dejando un intervalo en el que la máquina nueva era accesible desde internet. Para un segundo servidor en otro proveedor, su propio play crea el cortafuegos antes que la máquina, así que la máquina queda expuesta a internet con las reglas ya presentes. Ese proveedor asocia cortafuegos mediante etiquetas, que se crean primero: una regla dirigida a una etiqueta necesita que exista. El cortafuegos de cada proveedor se configura directamente a través de la API REST de ese proveedor con una tarea HTTP sencilla. Esa tarea no hace nada en una simulación, por lo que los cambios se ensayan ejecutando el propio play del cortafuegos: crear uno en el proveedor no cuesta nada y se puede borrar después.

Los contenedores pueden saltarse un cortafuegos sin tocarlo. Docker escribe sus propias entradas en el filtro de paquetes del kernel. El tráfico de un puerto publicado para un contenedor coincide con esas reglas antes que con cualquiera

de las añadidas por el rol de endurecimiento del host, de modo que el contenedor puede ser accesible desde fuera independientemente de lo ordenado al cortafuegos. Los contenedores de esta plataforma se ejecutan bajo systemd con Podman y Quadlet, comparten la red del host y no publican puertos. No hay, por tanto, ningún puerto publicado de un contenedor que permita eludir las reglas.

Dónde surgen los problemas

Una regla aparece en el estado y nunca actúa. Las reglas se evalúan por orden y gana la primera coincidencia. Una autorización amplia encima de un límite o denegación más concretos los deja sin efecto por completo, aunque ambos sigan apareciendo en el listado. Lea las reglas como el kernel, numeradas y por orden, ponga las concretas antes que las generales y pruebe desde una máquina externa:

```
ufw status numbered
nft list ruleset
```

Un contenedor responde en un puerto que el cortafuegos deniega. Docker publica puertos redirigiendo tráfico en la tabla NAT del kernel antes de que ese tráfico llegue a las cadenas de filtrado de ufw. Su propia documentación dice que esto ignora de hecho la configuración del cortafuegos. Publique solo en la dirección de bucle local, anteponga un cortafuegos del proveedor o ejecute contenedores sin puertos publicados, como antes.

IPv4 está protegido e IPv6 completamente abierto. Las dos familias tienen reglas separadas en iptables y ufw solo escribe reglas IPv6 cuando IPv6 está activado en su archivo de valores predeterminados. Un host con dirección IPv6 pública y reglas solo para IPv4 expone cada puerto en el que escucha. Use una tabla nftables de la familia `inet`, que ve ambas, o audite los dos listados y escanee desde fuera la dirección IPv6 del host.

La sesión SSH se congela al activar el cortafuegos. La denegación predeterminada entró en vigor antes que la regla SSH, o el demonio SSH cambió de puerto mientras la regla sigue indicando 22. Añada primero la regla SSH, mantenga abierta la consola del proveedor en otra ventana y programe una reversión automática para unos minutos después que se cancele cuando entre una sesión nueva.

Una regla abre más de lo previsto. En ufw, indicar un puerto sin protocolo lo abre tanto para TCP como para UDP. Nombre el protocolo en cada regla y declare toda la superficie esperada para que cualquier añadido haga fallar una comprobación en lugar de esperar a una auditoría.

Lecturas adicionales

- Cortafuegos — Las definiciones de las publicaciones del NIST, cada una enlazada a su documento de origen.
- SP 800-41 Rev. 1: directrices sobre cortafuegos y su política — Tipos de cortafuegos, denegación predeterminada y gestión de reglas, del NIST.
- ufw(8) — La interfaz de cortafuegos de Ubuntu, incluido el orden de reglas y la regla `limit`.
- Wiki de nftables — La documentación del proyecto netfilter para el marco que sustituye a iptables.
- Filtrado de paquetes y cortafuegos — La explicación de Docker sobre cómo los puertos publicados rodean las reglas del host.

Glosario

Elija según sus restricciones reales

Describa la ruta de despliegue, los riesgos operativos y quiénes se encargan de las guardias. Podemos definir el alcance de un cambio con comprobaciones, una vía de recuperación y responsabilidades claras.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es un cortafuegos?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/firewall/