Engineer

Endurecimiento

Eliminar lo que podría aprovechar un atacante

Qué es

El endurecimiento es la tarea de configurar un sistema para que exponga y permita solo lo que necesita para su función: menos servicios en ejecución, menos puertos a la escucha, menos privilegios para cada proceso y valores predeterminados más estrictos en todo lo que permanezca. El software es el mismo antes y después. Lo que cambia es cuánto puede alcanzar un intruso y hasta dónde le lleva un único punto de entrada. También se habla de endurecimiento del sistema, del servidor, del sistema operativo o de SSH, según la capa que se esté reforzando.

Una intervención típica de endurecimiento en un servidor Linux abarca varias capas a la vez. El acceso remoto acepta claves y rechaza contraseñas. La cuenta root no puede iniciar sesión directamente. Los paquetes que nadie usa se eliminan, en lugar de limitarse a detenerlos. Cada servicio de larga duración recibe su propio usuario sin privilegios, una vista de solo lectura del sistema de archivos y una lista de las llamadas al sistema que puede realizar. El cortafuegos admite los pocos puertos necesarios para los servicios del host. Ninguno de estos pasos es exótico: la disciplina consiste en hacerlos todos, en cada host, y mantenerlos vigentes.

Conviene situar el endurecimiento junto a las prácticas con las que suele confundirse. Aplicar parches corrige un defecto concreto del código cuando existe una solución. La monitorización y la detección de intrusiones descubren un ataque en curso. Un cortafuegos decide quién puede llegar a un puerto. El endurecimiento decide qué puede hacer el programa que hay detrás una vez alcanzado, y si ese puerto debería estar abierto. Las referencias publicadas, como los CIS Benchmarks, lo convierten en listas de comprobación por sistema operativo y producto: útiles como mínimo de partida, engañosas como meta final.

Utilice el concepto en una decisión

Aplique «¿Qué es el endurecimiento?» 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

La decisión real es dónde gastar un presupuesto de seguridad limitado, y el endurecimiento tiene la particularidad de proteger frente a vulnerabilidades que no se descubrirán hasta el año siguiente. Un parche protege frente a una vulnerabilidad conocida. Un servicio endurecido restringe lo que puede hacer un proceso comprometido. Por ejemplo, impedir que escriba en memoria ejecutable, abra sockets sin procesar o acceda a los directorios personales busca limitar el impacto de un fallo de ejecución remota de código. Los parches cierran agujeros; el endurecimiento reduce lo que vale cualquiera de ellos. Una operación seria necesita ambos, y el endurecimiento permite ganar tiempo entre la divulgación de un fallo y la llegada de su corrección.

El coste consiste sobre todo en tiempo de ingeniería y riesgo de roturas, más que en licencias. Cada restricción apuesta a que el servicio nunca necesita lo que se le retira, y algunas apuestas salen mal: una copia de seguridad que se ejecuta una vez por noche, una rotación de registros que envía una señal, un complemento que ejecuta un programa externo mediante una shell. Por eso el endurecimiento manual de un servidor en producción tiende a estancarse. Quien lo hace sufre una restricción que rompe algo a las tres de la mañana y deja de reforzar las restricciones. Si se expresa como código, con restricciones revisadas, versionadas y aplicadas de forma idéntica en cada reconstrucción, el mismo trabajo resulta barato de repetir y seguro de revertir.

Para quien contrata el trabajo, las preguntas prácticas son cuándo hacerlo y cómo saber que se hizo. La primera respuesta es durante la creación del host, como parte de la automatización que lo construye, porque una lista aplicada una sola vez a una máquina en marcha deja de reflejar la realidad en unos meses. La segunda es aportar pruebas del sistema en ejecución, y no del repositorio de configuración. Cada vez más, los auditores piden ambas: una configuración de referencia documentada y pruebas de que producción coincide con ella.

Cómo se aplica en la práctica

En el host de producción de esta empresa, SSH es la única vía de acceso interactivo y su configuración la escribe la automatización. En septiembre de 2026, un informe de acceso leyó los ajustes efectivos del demonio y encontró dos directivas previstas que habían sido anuladas: los valores predeterminados del sistema operativo se cargaban antes que el archivo complementario de la empresa y, para esos ajustes, prevalece el primer valor. Un prefijo de nombre de archivo que se ordena antes corrigió el orden. El cambio duradero fue la validación en tres etapas de 24 directivas SSH en cada convergencia: cada archivo candidato debe superar el análisis sintáctico antes de sustituir al archivo activo, la configuración ensamblada debe superar una comprobación tras instalarse y, después, se leen los ajustes efectivos del demonio y se comprueba su correspondencia con las 24 directivas. Se excluyeron deliberadamente otras dos directivas porque el demonio ya no las implementa.

El mismo host ejecuta 13 archivos de unidad instalados por su propia automatización, que habían funcionado con los privilegios predeterminados del gestor de servicios. Cada plantilla de unidad recibió un techo de privilegios, un filtro de llamadas al sistema, familias de direcciones restringidas y memoria que puede ser escribible o ejecutable, pero nunca ambas cosas. Según la puntuación del propio supervisor, el aislamiento redujo la exposición de un servicio de interruptor de hombre muerto de 9,6 a 1,5, la de la forja Git accesible desde Internet de 8,3 a 1,5 y la de las dos unidades de copia de seguridad de 9,6 a 2,3 y 2,5. Al escribirlas se descubrió una directiva mal escrita en tres plantillas; el supervisor registra una clave desconocida y arranca la unidad de todos modos. Ahora cada unidad pasa por el verificador del propio supervisor durante la convergencia.

Una revisión de endurecimiento en agosto de 2026 preguntó si funcionaban las protecciones contra fuerza bruta de SSH y consultó el sistema en ejecución para averiguarlo. Las tres protecciones habían estado inactivas desde la creación del host: el mecanismo de bloqueo (jail) vigilaba el puerto 22 mientras el demonio escuchaba en el 1986, un límite de frecuencia del cortafuegos estaba debajo de una regla más amplia que coincidía antes y la acción de bloqueo invocaba un binario de filtrado de paquetes que el gestor de paquetes solo recomienda y que no estaba instalado. De esa revisión salieron dos reglas que siguen rigiendo el repositorio: un control de seguridad lleva una aserción, no un comentario, y un cortafuegos se verifica por la posición de sus reglas, no por su presencia.

Dónde surgen los problemas

El ajuste está en el archivo y el servicio lo ignora. Muchos demonios leen varios archivos y conservan el primer o último valor que encuentran para cada clave. Una directiva de endurecimiento en un archivo complementario puede perder frente a un valor del proveedor que se carga antes. Pregunte al programa qué hará realmente, en SSH con `sshd -T`, y compare la salida con los valores previstos en cada ejecución.

Una errata convierte un control en un comentario. Los gestores de servicios y muchos demonios registran una clave desconocida y continúan. La unidad arranca, el panel está en verde y la restricción no existe. Ejecute el verificador del propio analizador como parte del despliegue para que una directiva desconocida lo detenga.

```
# What the SSH daemon will do once every Include is resolved
sshd -T | grep -E '^(passwordauthentication|permitrootlogin) '
# How much a unit's sandbox still exposes, scored 0.0 to 10.0
systemd-analyze security --no-pager example.service
# Unknown directives and other unit-file errors
systemd-analyze verify /etc/systemd/system/example.service
```

Una protección vigila la puerta equivocada. Un puerto cambia y la regla de bloqueo sigue vigilando el anterior. Se añade un límite de frecuencia debajo de una regla amplia de autorización que ya coincidió. El control informa de que está sano porque está en ejecución, lo que no dice nada de si llega a activarse. Pruebe cada control por su efecto: actívelo deliberadamente desde fuera y confirme que se descarta el paquete.

El aislamiento rompe el servicio por la noche. Un filtro de llamadas al sistema o una ruta de solo lectura rompe una ruta de código poco frecuente, como la exportación semanal o el hook de copia de seguridad, y el fallo aparece horas después del cambio, durante la ejecución y no durante el análisis. Aplique restricciones por etapas, ejercite todas las tareas programadas tras cada una y vigile en el registro los procesos terminados por el filtro.

La puntuación de la lista de comprobación se convierte en el objetivo. Un host puede superar todas las líneas de una lista genérica mientras su única aplicación expuesta a Internet se ejecuta como root y puede escribir en su propio código.

Empiece por lo accesible desde la red y por lo que podría tocar cada proceso alcanzable; después use la lista de comprobación para detectar lo que esa revisión pasó por alto.

Lecturas adicionales

- Endurecimiento, en el glosario del NIST — La definición del NIST y la publicación de la que procede.
- sshd_config(5) — Todas las directivas del servidor SSH y la regla que establece que se usa el primer valor obtenido para una palabra clave.
- systemd.exec(5) — Directivas de aislamiento de servicios: filtros de llamadas al sistema, familias de direcciones y memoria que puede ser escribible o ejecutable, pero nunca ambas cosas.
- systemd-analyze(1) — La puntuación de exposición del subcomando `security` y el subcomando `verify`, que detecta directivas desconocidas.
- CIS Benchmarks — Configuraciones de referencia consensuadas para sistemas operativos, plataformas de nube y software de servidor.

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 el endurecimiento?» 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/hardening/