Engineer

secreto operativo

Credencial que necesita un programa, como clave o contraseña

Qué es

Un secreto operativo es un valor que permite al software o a quien lo opera actuar sobre un sistema y que concede el mismo poder a cualquier otra persona que lo lea. Lo son la contraseña de conexión a una base de datos, un token de API, la clave privada de un par de claves SSH y la frase de contraseña que cifra una copia de seguridad. En el uso cotidiano, el término se acorta a secretos, y el trabajo de crearlos, almacenarlos, rotarlos y destruirlos se denomina gestión de secretos.

Dos cosas los distinguen del código que los rodea. La primera es quién los lee. La contraseña de un usuario está en la memoria de una persona y se escribe cuando se solicita; un secreto operativo lo lee un programa, un script de despliegue o una tarea programada, y nadie necesita recordarlo. Por eso la hoja de referencia sobre gestión de secretos de OWASP les asigna un ciclo de vida propio: se generan con los mínimos privilegios necesarios, se rotan según un calendario, se revocan si pueden haberse filtrado y se les fija una caducidad cuando la plataforma lo permite, con vidas útiles que van de minutos a años. Las contraseñas de usuario quedan fuera de esa rotación periódica. La segunda es la separación entre un secreto y la configuración que apunta a él. Un nombre de servidor o la ruta de un repositorio de copias de seguridad indica dónde está algo. El secreto es la parte que permite entrar a quien lo tiene y cuyas copias deben contabilizarse.

Los secretos también se anidan. El almacén que los guarda necesita su propio secreto, y la hoja de referencia señala que ese secreto principal, como las credenciales raíz de un proveedor de nube, normalmente debe guardarse en otro lugar.

Utilice el concepto en una decisión

Aplique «¿Qué es un secreto operativo?» 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

Un secreto funciona para quien lo presenta. Un token no deja constancia de quién debía tenerlo, así que cada copia amplía el conjunto de personas y programas capaces de utilizarlo. Un repositorio de código fuente es el peor destino: cada clon contiene todo el historial, por lo que un valor incorporado una sola vez viaja a cada portátil, ejecutor de compilaciones y espejo que descargue los cambios, y permanece en ese historial aunque se borre la línea que lo contenía.

Las normas convierten esto en requisitos. El Estándar de Verificación de Seguridad de Aplicaciones de OWASP pide una solución de gestión de secretos, como una bóveda de claves, para crear y almacenar los secretos del backend, controlar el acceso a ellos y destruirlos, y establece que no deben incluirse en el código fuente de la aplicación ni en los artefactos de compilación. Exige aplicar el mínimo privilegio al acceso y, en su nivel más alto, que caduquen y se roten. NIST SP 800-53 añade el refuerzo de control IA-5(7): los autenticadores estáticos sin cifrar no deben incrustarse en aplicaciones ni en otros almacenamientos estáticos. Sus indicaciones también cierran una laguna: un valor utilizado exactamente como está almacenado cuenta como no cifrado, con independencia del archivo donde se encuentre.

Las contrapartidas son reales. Un almacén central es otro sistema que debe estar disponible cuando más se necesita, y la hoja de referencia recuerda que, durante un incidente, se esperan credenciales nuevas con rapidez. Compartir un secreto entre varios servicios ahorra trabajo y dificulta atribuir una filtración, porque después no puede saberse cuál de sus titulares lo perdió. La manipulación manual aumenta con cada secreto; por eso la hoja de referencia favorece la rotación automática y advierte de que reiniciar una aplicación deja operativa una credencial robada. Solo revocarla termina con ello.

Cómo se aplica en la práctica

Este sitio declara su regla en la página sobre cómo trabaja la empresa de forma segura: los secretos quedan fuera del repositorio. Las credenciales se guardan en un almacén de secretos o en la capa de despliegue, fuera del código fuente y de todo lo que produce una compilación. Un análisis automático se ejecuta en cada commit y lo detiene cuando detecta un secreto.

Las notas de ingeniería del repositorio detallan cómo se hace. El analizador es gitleaks: examina el árbol de trabajo en cada commit y vuelve a hacerlo en la comprobación completa de calidad. Un hallazgo bloquea el proceso y no debe eludirse mediante una lista de excepciones. Se excluyen la carpeta de dependencias y los árboles de salida generados porque son copias reconstruidas desde cero. Las notas también registran un fallo menos evidente: cuando desaparece un archivo durante el análisis, gitleaks imprime una línea de error, pero termina igualmente con éxito; por eso la tarea que lo ejecuta falla ante cualquier línea de error. El propio sitio es estático, no contiene secretos de aplicación y se despliega mediante SSH con claves que conserva la propietaria.

Mantener los secretos fuera del repositorio los traslada a un lugar al que no llegan sus comprobaciones, y una ejecución previa de comprobación de secretos cubre esa brecha. Los secretos acababan de separarse por servidor y el código era correcto, pero la bóveda cifrada del operador, guardada en su propio equipo y fuera del alcance del control de calidad obligatorio, aún contenía el antiguo valor de un único servidor. La primera ejecución de convergencia llegó al servidor en funcionamiento, ejecutó ocho tareas y se negó a continuar en la novena, después de haber escrito ya la configuración de swap en producción. La respuesta fue un play que no contacta con ningún servidor, no recopila datos del sistema y ejecuta contra la bóveda los mismos dos archivos de tareas que la convergencia real, en aproximadamente un segundo, antes de que el aprovisionamiento haga nada más. Reutilizar el código en vez de volver a formular la regla evita que la comprobación y la convergencia diverjan. Un secreto incorrecto se traduce ahora en un rechazo tras dos segundos en un portátil.

La separación en sí es el paso a un valor por servidor para cada secreto operativo. La plataforma tenía un servidor y estaba a punto de tener dos. Varios secretos se habían definido como valores únicos, entre ellos el repositorio de copias de seguridad, su frase de contraseña y un interruptor de alertas. Cada uno se convirtió en un mapa con claves basadas en el nombre de servidor del inventario, y junto a la configuración quedaron registrados dos motivos. Un interruptor de

hombre muerto compartido por dos servidores permanece en verde mientras cualquiera de ellos siga notificando su presencia, por lo que el segundo oculta la caída del primero. Los repositorios restic separados pueden tener claves maestras distintas aunque utilicen la misma frase de contraseña. Pero quien obtenga esa frase y pueda leer los datos cifrados y los archivos de claves de ambos repositorios puede desbloquear los dos; compartirla vincula la exposición de ambos repositorios en lugar de aislar los servidores. El servidor existente figura en los cuatro mapas, con el repositorio y ambos interruptores en blanco: la integración existe y el operador aún debe aportar el valor.

Dónde surgen los problemas

Un secreto acaba en un commit. Un analizador detecta un valor en un diff o alguien descubre un token en un archivo enviado al repositorio remoto porque estaba en una configuración o en datos de prueba al crear el commit. La hoja de referencia de OWASP establece el orden de reparación: primero revocar el valor expuesto, después rotarlo y solo entonces eliminarlo del código o de los registros. Reescribir el historial para retirar el commit cambia todos los posteriores y rompe sus enlaces, así que necesita planificación. La detección debe hacerse antes del commit, en un hook de pre-commit o en el editor, donde un hallazgo no cuesta nada.

El análisis da por buenos archivos que nunca leyó. Una ejecución termina en verde mientras parte del árbol queda sin examinar: un archivo desapareció durante el recorrido, una exclusión abarcó demasiado o el secreto no tiene una forma reconocible. Las reglas de patrones detectan tokens con formato fijo, pero no una contraseña inventada por una persona; por eso la hoja de referencia propone utilizar más de una herramienta. Trate cualquier error del analizador como un fallo y limite las exclusiones a copias que una compilación vuelva a crear.

Un secreto sirve a muchos titulares. Dos servidores, dos servicios o todo un equipo utilizan el mismo valor. Cuando se filtra, los registros no pueden decir qué copia se perdió, y sustituirlo exige actuar sobre todos sus titulares a la vez. SP 800-53 pide cambiar los autenticadores de las cuentas de grupo o de rol cuando cambian sus miembros; un secreto por titular convierte esa tarea en una sola sustitución, no en una operación coordinada.

El secreto se distribuye con la compilación o llega a un registro. Una imagen de contenedor, un script empaquetado o una línea de depuración saca el valor del lugar donde debía permanecer. ASVS menciona los artefactos de compilación

junto al código fuente, y la hoja de referencia pide mantener los secretos fuera de los registros y enmascararlos cuando una línea deba hacer referencia a uno. Analice también lo que produce la compilación, además del repositorio.

Lecturas adicionales

- Hoja de referencia de OWASP sobre gestión de secretos — el ciclo de vida de un secreto, su detección antes del commit y el orden de reparación tras una filtración.
- Estándar de Verificación de Seguridad de Aplicaciones de OWASP — los requisitos de gestión de secretos del grupo 13.3: bóveda, mínimo privilegio y rotación.
- NIST SP 800-53, revisión 5 — la gestión de autenticadores en IA-5, con IA-5(7) contra los autenticadores estáticos incrustados.
- Gitleaks — el analizador que este sitio ejecuta sobre su árbol de trabajo en cada commit.

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 secreto operativo?» 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/secret/