Engineer

autenticación

probar que una solicitud viene de quien afirma enviarla

Qué es

La autenticación es la comprobación que hace un sistema antes de aceptar una identidad declarada: alguien afirma que es una determinada cuenta y el sistema pide pruebas que solo el titular de esa cuenta debería poder producir. Los ingenieros suelen abreviarla authn, y las interfaces llaman a ese mismo momento iniciar sesión o acceder a la cuenta.

La identificación es la declaración en sí, por ejemplo escribir un nombre de usuario. La autenticación es la prueba que respalda esa declaración. La autorización es una decisión distinta, tomada después, sobre lo que puede hacer la identidad ya conocida. Incluso HTTP desdibuja la frontera: RFC 9110 llama «Unauthorized» al código de estado 401, pero lo define como una solicitud que carece de credenciales de autenticación válidas y obliga al servidor a enviar una cabecera WWW-Authenticate que explique cómo aportarlas. El código para una identidad conocida a la que aun así se deniega lo solicitado es 403 Forbidden.

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Bearer realm="api"
```

Las pruebas se dividen en clases, habitualmente llamadas factores. Algo que usted sabe es una contraseña o un PIN. Algo que tiene es un teléfono, una llave física de seguridad o un archivo de clave privada. Algo que es puede ser una huella dactilar o un rostro. La autenticación multifactor combina dos clases distintas, de modo que a quien robe una todavía le falte la otra. Dos contraseñas siguen siendo un solo factor.

Buena parte de la autenticación no implica a ninguna persona. Los servicios se presentan tokens API entre sí, y SSH muestra con claridad la versión entre máquinas: RFC 4252 establece «publickey» como el único método que toda implementación debe admitir, y el cliente demuestra que posee una clave privada firmando con ella, por lo que la clave nunca atraviesa la red. El inicio de sesión único traslada la comprobación a otro lugar. Con OpenID Connect, un proveedor de identidad autentica al usuario y entrega a la aplicación un token ID firmado, de modo que la aplicación nunca ve una contraseña. Las passkeys, construidas sobre Web Authentication API del W3C, mantienen en el dispositivo una clave privada limitada a un solo sitio web y firman el desafío del servidor junto con el origen con el que realmente habla el navegador, de forma que un dominio que imita al legítimo no recibe nada que pueda reutilizar.

Utilice el concepto en una decisión

Aplique «¿Qué es la autenticación?» 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

Todos los demás controles presuponen que este funcionó. Los permisos, los registros de auditoría, los límites de frecuencia y la facturación se vinculan a una identidad; si la comprobación que estableció esa identidad es débil, todos se aplican fielmente a la parte equivocada. Una línea de registro que dice que un administrador borró una tabla identifica la cuenta que lo hizo, y solo una buena autenticación permite convertir ese dato en quién lo hizo.

La decisión práctica es cuán sólida debe ser la comprobación en cada puerta, y NIST SP 800-63B ofrece la escala más clara para decidirlo. Su primer nivel de garantía, AAL1, acepta un solo factor. AAL2 exige dos factores y que el verificador ofrezca al menos una opción resistente al phishing. AAL3 exige un autenticador resistente al phishing cuya clave privada no se pueda exportar. La guía es tajante respecto a una opción popular: los códigos que se introducen manualmente, tanto si llegan por mensaje de texto como si se muestran en una app, no son resistentes al phishing, porque nada vincula los dígitos introducidos con el sitio que los solicitó.

La revisión 4 del mismo documento, que sustituyó a la revisión 3 el 1 de agosto de 2025, también resolvió varios debates sobre contraseñas. Una contraseña usada como único factor debe tener al menos 15 caracteres; una usada únicamente dentro de un inicio de sesión multifactor, al menos 8. Los verificadores no pueden imponer otras reglas de composición ni exigir cambios periódicos, pero deben forzar un cambio cuando haya pruebas de que la contraseña se ha visto comprometida. Las contraseñas nuevas se contrastan con una lista de contraseñas comunes y filtradas, se explica al usuario por qué se rechazó su elección y la forma almacenada lleva sal y hash mediante un esquema con factor de coste. Los intentos fallidos consecutivos contra un autenticador se limitan a 100. Para las sesiones de AAL2, recomienda volver a pedir autenticación al menos cada 24 horas y después de una hora sin actividad.

Nuestra recomendación se desprende de lo anterior. Cuando los usuarios ya tengan una cuenta en un proveedor de identidad competente, deje que ese proveedor haga la comprobación y acepte su respuesta firmada, porque una contraseña que la aplicación nunca almacena es una contraseña que nunca puede filtrar. Cuando la aplicación deba conservar contraseñas, siga literalmente 800-63B en vez de hacerlo de memoria y proteja cualquier función administrativa con un método resistente al phishing.

Cómo se aplica en la práctica

Cuando la empresa trasladó su código fuente a su propio servidor Git, la autenticación fue el primer punto de la lista de lo que implica operar una forja. La elegida da prioridad a SSH y carece de interfaz web administrativa, por lo que la entrada es una clave. Se rechaza tanto una sesión anónima como una que llegue sin clave, y además todos los repositorios declarados están marcados como privados. El punto de escucha que comprueba esas claves, en el puerto 23231, solo acepta conexiones de la interfaz loopback, de modo que cualquier inicio de sesión desde fuera pasa primero por un servidor de salto y la superficie declarada del cortafuegos conserva el mismo tamaño.

La misma preocupación adopta otra forma a escala de toda una organización. En un puesto anterior de ingeniera de infraestructura, el trabajo consistió en consolidar identidades y accesos en M365, 1Password, Red Hat SSO y OKTA SSO, donde distintas personas habían concedido permisos de forma improvisada a lo largo del tiempo. Se hizo cumplir el principio de mínimo privilegio y se estandarizaron las altas y las bajas para revocar el acceso con prontitud, en vez de

dejarlo vigente después de una marcha. El riesgo de acceso no autorizado se redujo drásticamente y el acceso pasó a ser auditable: el equipo pudo responder por fin quién podía acceder a un recurso y por qué.

La autenticación también tiene un rostro, y los usuarios la juzgan por sus mensajes de error. En el inicio de sesión de una aplicación contra un gran proveedor de correo, el mensaje decía que la contraseña era incorrecta. Era correcta: el proveedor exige una contraseña específica de aplicación para clientes de terceros y rechaza la contraseña de la cuenta, por mucho cuidado que se ponga al escribirla. El mensaje se reescribió para nombrar al proveedor, explicar que exige una contraseña específica de aplicación para otros clientes e indicar dónde crearla. Ahora una prueba provoca ese rechazo y comprueba que el mensaje contenga el nombre del proveedor y el tipo de credencial requerido, de modo que la redacción anterior no pueda volver sin que se detecte.

Dónde surgen los problemas

- **El mensaje culpa al usuario de una causa externa.** Una cuenta bloqueada, una contraseña caducada, un reloj desincronizado con el generador de códigos y un proveedor que exige otra credencial llegan todos como contraseña incorrecta, y el usuario vuelve a escribir algo que nunca funcionará. Nombre la causa sobre la que pueda actuar, manteniendo una respuesta idéntica para las cuentas que existen y las que no, para que el formulario no facilite a nadie una lista de nombres de usuario válidos.
- **Una página de phishing retransmite el código en tiempo real.** El usuario escribe una contraseña y un código de seis dígitos en una copia de la página real, que reenvía ambos al sitio auténtico dentro de los 30 segundos que RFC 6238 recomienda como intervalo temporal predeterminado. Los códigos de un solo uso frenan las contraseñas robadas y reutilizadas, pero hacen poco frente a una retransmisión en directo. Las claves vinculadas al origen, como las passkeys, firman para un solo sitio: por eso NIST las considera resistentes al phishing, a diferencia de los códigos introducidos manualmente.
- **Las credenciales sobreviven a quienes las tenían.** La baja cierra la cuenta en el directorio, mientras siguen funcionando una clave SSH en un archivo `authorized_keys`, un token de acceso personal y una contraseña de administrador compartida. Cada credencial necesita un responsable, un lugar donde quede registrada y una fecha para que alguien vuelva a revisarla.
- **La sesión sigue siendo válida mucho después del inicio de sesión.** Una cookie o un token que dura meses es una contraseña con una mecha larga. Cerrar sesión debe terminarla en el servidor, no limitarse a borrar la cookie en un navegador; un token autónomo que el servidor nunca vuelve a comprobar sigue siendo utilizable hasta que caduca, pase lo que pase con la cuenta.

- **La recuperación de cuenta es la entrada más débil.** Un inicio de sesión sólido junto a un enlace de restablecimiento enviado a una dirección de correo convierte el buzón en la credencial real. La recuperación merece el mismo examen que la puerta principal, porque el atacante elegirá la que sea más débil.

Lecturas adicionales

- NIST SP 800-63B-4 — el texto vigente: niveles de garantía, mínimo de 15 caracteres para contraseñas y qué se considera resistente al phishing.
- NIST SP 800-63B, revisión 3 — la guía de autenticación de 2017, conservada en línea y marcada como sustituida por la revisión 4 desde el 1 de agosto de 2025.
- RFC 9110, sección 11: autenticación HTTP — el desafío y la respuesta detrás de 401, WWW-Authenticate y la cabecera Authorization.
- RFC 4252: protocolo de autenticación SSH — inicio de sesión con clave pública, contraseña y basado en host, tal como los define SSH.
- RFC 6238: TOTP — cómo las apps autenticadoras derivan sus códigos breves de un secreto compartido y del reloj.
- Web Authentication, Level 3 — la API del W3C detrás de las passkeys y sus credenciales de clave pública limitadas al origen.

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 la autenticación?» 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/authentication/