Política empresarial de firma de código

Actualizado el 22 de mayo de 2025

1. Descripción general

Múltiples grupos de las empresas modernas publican software para uso interno y externo. Para ayudar a protegerse de la manipulación del software y confirmar la autenticidad de las versiones, las mejores prácticas establecen que los binarios de software deben firmarse con claves criptográficas. Sin embargo, para que la firma sea de confianza, los procesos de firma de código deben someterse a una supervisión y controles adecuados.

Esta política pretende ayudar a cualquier empresa que cree y distribuya software.

2. Objetivo

Esta política describe los requisitos y procedimientos necesarios para controlar los procesos de firma de código de software con el fin de garantizar que solo se firme y distribuya código autorizado y verificado, mejorando así la seguridad e integridad del software.

Esta política sirve para garantizar que la empresa sabe lo que publica y que el software publicado no introduce riesgos de seguridad inaceptables. También permite a los usuarios verificar la integridad de los archivos publicados para demostrar que no han sido manipulados y alertar a los usuarios si el software no ha sido creado por el editor cuya identidad se ha adjuntado al código.

Una política rigurosa de firma de código y un mantenimiento adecuado de los registros también pueden facilitar el trabajo de los auditores, tanto internos como externos.

3. Ámbito de aplicación

Esta política se aplica a todo el personal, los procesos y los sistemas implicados en el proceso de firma de código dentro de la empresa, incluidos desarrolladores internos y externos, firmantes, revisores y auditores. Se aplica al código desarrollado por la empresa para uso interno o venta externa y a los artefactos de software creados o utilizados en el SDLC y el proceso de CI/CD (por ejemplo, SBOM y scripts de compilación).

No se aplica al software de terceros que la empresa utilice para el desarrollo u otras actividades (por ejemplo, herramientas de productividad como Microsoft Office o herramientas de desarrollo como JFrog).

4. Acceso y privilegios

4.1 Autenticación

Los usuarios que accedan al entorno de desarrollo de software deben utilizar servicios o métodos de autenticación aprobados. No deben utilizar cuentas compartidas, credenciales almacenadas de forma insegura, cuentas locales ni otras formas de autenticación que infrinjan la política de autenticación de la empresa.

Se debe utilizar la autenticación multifactor o la autenticación de servidor a servidor para acceder y autorizar la firma según los requisitos básicos del CA/B Forum.

4.2 Definición de roles: control de acceso basado en roles (RBAC)

Según las mejores prácticas, la empresa debe establecer roles separados con privilegios definidos para las actividades de firma de código. Estos roles deben incluir, como mínimo, «Remitente del código», «Firmante» y «Responsable de seguridad», y ampliarse para incluir «Revisor», «Auditor» y «Administrador del sistema».

4.3 Separación de tareas

Los usuarios pueden tener múltiples roles en función de las prácticas comercialmente razonables. Sin embargo, ningún usuario puede tener el control exclusivo de un proceso, ya que esto introduce un punto único de fallo para la seguridad.

Limite el acceso al conjunto mínimo de privilegios para obtener un control de seguridad más estricto basado en el principio de seguridad del mínimo privilegio.

4.4 Restricción del acceso a los recursos básicos

Todos los accesos, pero especialmente el acceso a recursos privilegiados, como los servidores de compilación y la administración de HSM, deben asignarse en función de los roles definidos, y no de usuarios específicos (o grupos no relacionados como «Administrador» o «Directivo»).

Para tener un control más estricto de la seguridad, a los recursos básicos debe dárseles el mínimo conjunto de accesos y privilegios.

4.5 Acciones de alto riesgo

Ciertas operaciones, como la exportación o eliminación de pares de claves y la revocación de certificados, pueden suponer grandes riesgos para la seguridad o causar importantes interrupciones en los procesos si no se realizan correctamente. Este tipo de acciones puede justificar que se requiera más de una autorización para firmar, además de privilegios de firma muy limitados.

4.6 Riesgo de que el software comprometido ocasione daños

El riesgo de cada producto o proyecto de software debe evaluarse en función de los daños que podrían producirse en el supuesto de que el software contuviese código malicioso o

vulnerabilidades explotables que pudieran provocar un acceso no autorizado a los datos, la pérdida de funcionalidad del sistema o algún otro fallo (por ejemplo, software con acceso de bajo nivel al sistema, como el acceso de root en Linux).

Un software de riesgo suficientemente alto puede justificar que se requiera más de una autorización para firmar, además de privilegios de firma muy limitados.

5. Gestión de certificados y pares de claves

5.1 Entornos de software

En los entornos de producción y de no producción, deben utilizarse pares de claves y certificados de firma de código únicos. Los entornos de prueba a los que se puede acceder en remoto y que procesan datos confidenciales deben tratarse como entornos de producción.

5.2 Almacenamiento de certificados y pares de claves

Las claves de firma del código de producción deben generarse y almacenarse de forma segura, utilizando un módulo de seguridad de hardware que cumpla con el estándar FIPS 140-2 de Nivel 2 o el Common Criteria EAL 4+, y utilizarse de acuerdo con las políticas de gestión criptográfica y de certificados de la empresa.

No deben utilizarse las mismas claves de firma de código de producción para diferentes usuarios, a menos que los logs de actividad puedan registrar e identificar las acciones por usuarios únicos, incluidos los usuarios del servicio.

5.3 Inventario y ciclo de vida de los certificados

Lleve un inventario de todos los certificados de firma de código. Supervise las fechas de caducidad y genere certificados nuevos de forma proactiva.

Los nuevos certificados y claves pueden crearse de forma segura y ponerse en uso automáticamente con la mayoría de los sistemas de gestión del ciclo de vida de los certificados (CLM).

5.4 Rotación de claves

Rotar las claves mejora la seguridad, ya que reduce el impacto que podría tener la existencia de una clave comprometida. El proceso puede realizarse manualmente, pero es más fácil y rápido automatizarlo.

5.5 Cifrado de claves

Cifre las claves al nivel más alto posible en función de los métodos de descifrado que pueda utilizar el usuario final. Anime al receptor de las claves cifradas a actualizar sus opciones de descifrado, de modo que puedan utilizarse algoritmos de cifrado preparados para la PQC cuando

estén disponibles.

5.6 PKI pública vs. PKI privada

El software que se utiliza o al que se accede fuera de su empresa debe firmarse con una clave pública que pueda verificarse fácilmente con listas de PKI estándar.

El software que solo se vaya a utilizar y al que solo se vaya a acceder a través de una red corporativa puede utilizar una «PKI privada», en la que los certificados están encadenados a una raíz de confianza, pero es la empresa la que los emite a través de una autoridad de certificación intermediaria (ICA). Esta es una situación habitual en el caso del software que se desarrolla y se utiliza únicamente a nivel interno.

6. Proceso de firma de código

6.1 Revisión de seguridad

El código fuente del software debe superar una revisión de seguridad de acuerdo con la política de seguridad del ciclo de vida de desarrollo de software (SDLC).

Cada versión y cada uno de sus componentes asociados deben analizarse para detectar posibles indicadores maliciosos y vulnerabilidades conocidas. No firme ningún software que contenga malware confirmado o alguna vulnerabilidad explotable.

6.2 Procesos automatizados

Lo recomendado según las mejores prácticas es automatizar el proceso de firma de código como parte del SDLC y del ciclo de CI/CD. Utilice una autoridad de certificación (CA) aprobada para los certificados de firma de código. Lo ideal es que los desarrolladores utilicen una clave de firma única para firmar cualquier código que envíen. Las firmas individuales deben verificarse antes de firmar una versión.

6.3 Sellos de tiempo

Las versiones que sigan siendo válidas una vez que el certificado de firma de código haya caducado deben incluir un sello de tiempo de un servicio de sellado de tiempo aprobado para demostrar el momento en el que se realizó la firma. Asegúrese de que sus sellos de tiempo y su autoridad de sellado de tiempo (TSA) cumplan los nuevos requisitos de seguridad del CA/B Forum en vigor desde abril de 2025.

6.4 Firma de artefactos de la versión y listas de materiales de software (SBOM)

Los artefactos de la versión, incluida la rama de código del software, el SBOM, los scripts de compilación y los documentos de intercambio de explotabilidad de vulnerabilidades, deben firmarse y almacenarse para ayudar a detectar posibles manipulaciones de datos y archivos.

Los artefactos deben almacenarse de forma segura, y también podría ser necesario disponer de métodos seguros para compartirlos con los clientes u órganos de gobierno. Los SBOM, en particular, pueden formar parte de un paquete de propuesta comercial o de una presentación de producto.

6.5 Auditoría y registros

Mantenga un registro de auditoría exhaustivo que incluya pruebas de las revisiones de código, análisis de seguridad y acciones relativas a las claves de firma de código. Las actividades deben estar asociadas a usuarios y cuentas de servicio únicos. La firma y cualquier acción asociada a certificados, pares de claves, usuarios y plantillas deben figurar en los registros.

7. Agilidad criptográfica: preparación y respuesta ante el cambio

7.1 Preparación y respuesta ante incidentes y cambios inesperados

Defina y automatice tantos procedimientos como sea posible para documentar, revocar y reemitir rápidamente las claves comprometidas.

Defina y automatice procedimientos para volver a firmar e implementar el código y los artefactos de software que se habían firmado con la clave comprometida.

7.2 Preparación y respuesta ante los cambios previstos

Las normativas cambian constantemente. Sin embargo, suelen introducirse gradualmente o dar un margen de tiempo para realizar cambios en el sistema. Defina y automatice procedimientos para documentar y realizar los cambios progresivamente; por ejemplo, realizando actualizaciones al volver a emitir un certificado próximo a caducar.

8. Revisión y actualización

Esta política se revisará anualmente y se actualizará según sea necesario para garantizar su eficacia y su adecuación a las normas del sector y a los cambios que se produzcan en la empresa.

Haga clic aquí para descargar una copia editable de este documento.