CUATRO PRÁCTICAS RECOMENDADAS PARA PROTEGERSE DE LOS ATAQUES A LA CADENA DE SUMINISTRO DE SOFTWARE

digicert®

Integración de firma de código segura, detección de amenazas y de malware, y generación de SBOM

Los ataques a la cadena de suministro de software son cada vez más frecuentes. Un informe reciente de Gartner afirma que, de aquí a 2025, el 45 % de las empresas de todo el mundo habrán sufrido un ataque a su cadena de suministro de software, una cifra que triplicaría la de 2021.

Los ataques afectan a empresas de todos los tamaños. A estas alturas, casi todo el mundo ha oído hablar de los ataques a las cadenas de suministro de SolarWinds, CircleCI y 3CX. Más recientemente, Micro-Star International (MSI) sufrió un ataque que dejó al descubierto las claves de firma de código privadas que utilizan para su BIOS y las que utilizan para el BootGuard de Intel.

Las medidas de seguridad básicas, como los principios Zero Trust, proporcionan a las empresas una primera línea de defensa, pero no son suficientes para proteger la cadena de suministro de software y el ciclo de desarrollo de una empresa.

A QUIÉN VA DIRIGIDO:

A los profesionales de la seguridad, de DevSecOps y de ingeniería de software encargados de proteger la infraestructura de desarrollo de software de una empresa para evitar brechas, ataques y otras vulnerabilidades.

RETOS

Los ciberdelincuentes utilizan tácticas diversas cuando atacan una cadena de suministro de software. Pueden tratar de hacerse con la contraseña de un desarrollador para introducirse en el sistema de creación del software, robar o hacer un uso indebido de claves privadas de firma de código desprotegidas, insertar malware en la fuente de paquetes de software de código abierto habituales o utilizar una combinación de todas estas tácticas. Como demostró el ataque de Sunburst a SolarWinds en 2019, las tácticas de ataque se están volviendo bastante sofisticadas y complejas, por lo que para combatir estos ataques se necesitan métodos más sofisticados, tal como se muestra en la figura 1.

Figura 1: Los ataques a la cadena de suministro pueden producirse en cualquier fase

Los ataques pueden producirse durante cualquier fase del ciclo de desarrollo de software. Las herramientas de seguridad independientes actuales tal vez detecten determinados tipos de ataques, pero pasarán por alto otros tipos. Por lo tanto, las medidas de seguridad deben implementarse al principio del ciclo de desarrollo de software (lo que se conoce como estrategia «shift up»).

También cabe señalar otros retos organizativos y culturales: los equipos de desarrollo de software pueden mostrarse reticentes a adoptar herramientas de seguridad porque pueden ser complicadas, interrumpir los ciclos de CI/CD automatizados o ralentizar el proceso de publicación del software.

Los equipos de seguridad se enfrentan distintos desafíos. Necesitan ofrecer soluciones de seguridad y garantizar el cumplimiento normativo con muchos equipos de software repartidos por toda la organización, a menudo distribuidos geográficamente. Además, normalmente estos equipos utilizan distintos entornos y lenguajes de programación, implementan software en diversas plataformas y utilizan infraestructuras de creación de software y procesos de software distintos. Esta heterogeneidad dificulta enormemente la tarea de definir, aplicar y supervisar la política de seguridad del software en toda la empresa.

Por último, cabe añadir un nuevo obstáculo: la obligación de cumplir con las diversas normativas gubernamentales y sectoriales elaboradas para hacer frente a los ataques a la cadena de suministro de software, como por ejemplo las que exigen la transparencia del software.

PRÁCTICAS RECOMENDADAS PARA PROTEGER LAS CADENAS DE SUMINISTRO DE SOFTWARE

Existen varias normativas y recomendaciones del sector que especifican no solo las prácticas recomendadas, sino también los requisitos que deben cumplir las empresas para garantizar la integridad de la cadena de suministro de software. Por ejemplo, la orden ejecutiva de EE. UU. relativa a la Mejora de la Ciberseguridad de la Nación y la Ley de Ciberresiliencia de la Unión Europea obligan a generar listas de materiales de software para demostrar la transparencia del software. En el informe «Security Considerations for Code Signing» sobre la seguridad para la firma de código elaborado por el NIST se describen los pasos necesarios para obtener una firma de código segura.

Este resumen de la solución recoge cuatro prácticas recomendadas que pueden aplicar las empresas para hacer frente a estos desafíos, además de todo lo que ofrece DigiCert. La ventaja fundamental de estas prácticas es que engloban todo el ciclo de desarrollo de software, en lugar de centrarse en una única etapa del SDLC (véase la figura 2).

Figura 2: Confianza de software: «shift left», «shift right» y… «shift up»

Para luchar contra los ataques a la cadena de suministro de software se necesita un enfoque integrado de última generación que sea versátil y que se centre en varios aspectos del SDLC.

1) Firma de código segura en toda la empresa

La firma de código es una medida de seguridad básica que lleva utilizándose más de 30 años. Se vale de la criptografía para firmar digitalmente un paquete de software (ejecutable, biblioteca, aplicación, etc.) para demostrar que el software es auténtico (por ejemplo, que el software firmado por DigiCert procede realmente de DigiCert) y no ha sido manipulado por terceros. Numerosos sistemas operativos, como iOS, macOS o Windows, comprueban la firma digital del software antes de instalar y ejecutar el paquete.

La firma de software es tan eficaz que los sistemas de firma de código se han convertido en el blanco de ataque de los ciberdelincuentes para robar, dejar al descubierto o hacer un uso indebido de los certificados de firma de código y las claves privadas de las empresas.

Un primer paso lógico para proteger la firma de código es garantizar que todas las claves de firma de código privadas se almacenen de forma segura. Un requisito reciente del Certificate Authority/Browser Forum obliga a almacenar las claves de firma de código privadas de confianza pública en un dispositivo FIPS 140 Nivel 2, Common Criteria EAL 4+, como por ejemplo un módulo de seguridad de hardware (HSM).

Aunque el almacenamiento seguro es un primer paso necesario, no es suficiente para proteger la firma de código en toda la empresa. La razón es que un nombre de usuario o contraseña en riesgo podrían permitir a un atacante acceder al almacenamiento seguro.

En su lugar, recomendamos utilizar un sistema de firma de código empresarial que ofrezca medidas de seguridad adicionales, como por ejemplo:

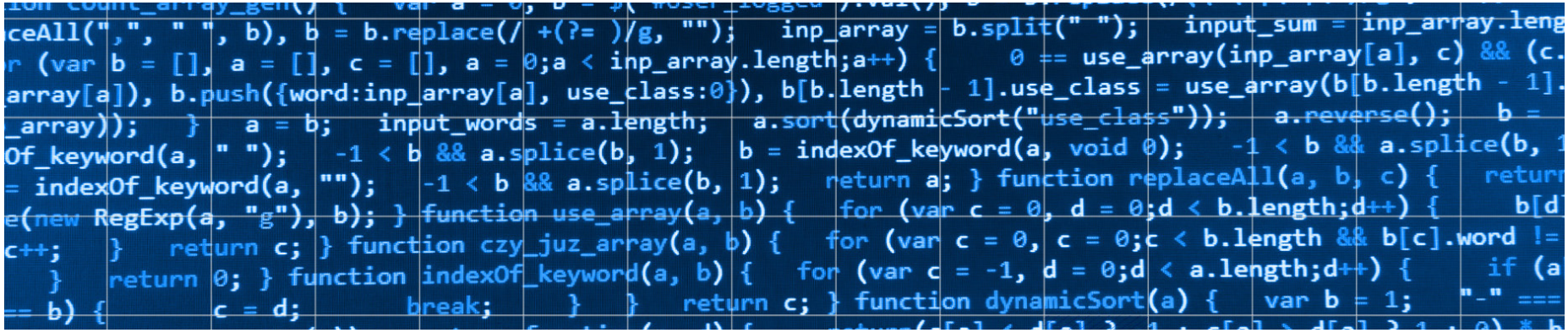
- Automatización de la gestión de certificados de firma de código, que incluya la emisión y la revocación.
- Un proceso de aprobación que especifique no solo quién puede acceder a una clave de firma de código, sino también la persona que debe aprobar el acceso y las circunstancias en las que se puede acceder a la clave (por ejemplo, la hora del día, la máquina de compilación, etc.).
- Un sistema basado en roles que especifique qué usuarios pueden acceder a determinados certificados de firma de código y claves para operaciones de firma y, de ellos, quién está autorizado a aprobar operaciones de firma y quién puede supervisarlas.
- Mantenimiento de un registro inequívoco de todas las actividades de firma de código para consultarlo en caso de que un certificado esté en riesgo.
- Un único sistema de firma de código que admita un gran volumen de operaciones de firma de código y muchas plataformas y herramientas diferentes de implementación y desarrollo de software.
- Un sistema basado en políticas con medidas de control de obligado cumplimiento para que el software pueda ser publicado.
- Firma de artefactos de software intermedios en todo el SDLC, incluido el código fuente antes de ser confirmado, los scripts y recetas de compilación, y los archivos de contenedores y otros archivos de infraestructura de software.
- Visibilidad en toda la empresa y definición y aplicación de políticas desde una ubicación centralizada, estén donde estén los distintos equipos de software.

2) Detección completa de amenazas

Aunque la firma de código puede evitar la manipulación del software una vez publicado, ¿qué ocurre si el software se manipula antes de que se firme y se publique? Por ejemplo, un ciberdelincuente con acceso a un sistema de compilación desprotegido podría insertar malware en el repositorio de código fuente de la empresa. O un desarrollador de software podría descargar un paquete de código abierto que contenga malware oculto. O podría descubrirse una vulnerabilidad oculta no detectada previamente.

Varias herramientas de seguridad del mercado solucionan algunos de estos problemas, como las herramientas de análisis estático de software (SAST). Sin embargo, normalmente solo se centran en un único aspecto (como las vulnerabilidades del software de código abierto).

En su lugar, se necesita un método exhaustivo que analice los binarios de software finales en busca de amenazas, vulnerabilidades, secretos integrados y malware. Este análisis debe realizarse sea cual sea la fuente original de los componentes de software: código abierto, bibliotecas comerciales de terceros o código privado creado por la empresa.



3) Transparencia del software

Como ya se ha dicho, ahora las normativas gubernamentales e industriales exigen con más frecuencia a los distribuidores de software que revelen el contenido del software, del mismo modo que las etiquetas nutricionales muestran los ingredientes de los alimentos. Conocidos como listas de materiales de software (SBOM, por sus siglas en inglés), estos documentos catalogan todos los componentes de una determinada pieza de software sea cual sea la fuente del componente (código abierto, biblioteca de terceros, etc.). Al igual que las etiquetas de los alimentos, las SBOM no solo indican la lista de «ingredientes» del software, sino que además permiten evaluar la calidad de esos ingredientes, como las versiones utilizadas, las vulnerabilidades conocidas, etc.

Aunque no sea obligatorio a efectos de cumplimiento, las organizaciones deberían generar listas SBOM para:

- **Evaluar y priorizar la superficie de ataque del software**: busque versiones vulnerables o afectadas de bibliotecas o componentes, examine las dependencias de software y compruebe si faltan revisiones de seguridad obligatorias.
- **Busque componentes obsoletos,** como por ejemplo mitigaciones que falten y prácticas de firma de código inseguras.
- **Adopte medidas defensivas proactivas** utilizando la estructura de software para dar apoyo a los equipos de detección y respuesta a incidentes de amenazas y para el modelado de amenazas.

4) «Shift up»: control y visibilidad en toda la empresa y el SDLC

Hay muchas medidas de seguridad que se centran o que protegen una parte específica del SDLC. Hemos oído hablar de estrategias de seguridad «shift left» y «shift right» en procesos de DevOps, que es como fijarse en los árboles en lugar del bosque. La cuestión es que, a medida que los ataques se vuelven más complejos, las organizaciones necesitan ampliar la perspectiva y adoptar un enfoque «shift up»:

- Contemple todo el proceso del SDLC, considerando el software como un todo.
- Gane visibilidad sobre la política de seguridad, las medidas y los resultados en toda la empresa.
- Defina y aplique la política de seguridad en toda la empresa.
- Realice análisis de amenazas y vulnerabilidades en los ejecutables de software finales justo antes de que se firmen y publiquen.

Para poder hacer todo esto, y que los equipos de desarrollo de software adopten las herramientas de seguridad, estas deben integrarse perfectamente en los procesos de software existentes y no ralentizar el proceso de publicación del software. Tienen que ser fáciles, incluso transparentes, para que los desarrolladores de software las utilicen.

DIGICERT SOFTWARE TRUST MANAGER

DigiCert® Software Trust Manager es una solución de confianza digital que protege la integridad del software a lo largo de toda la cadena de suministro, reduciendo el riesgo de que el código se vea afectado, aplicando la política corporativa y reguladora, y ofreciendo controles detallados de acceso y de uso de claves para la firma de código.

DigiCert Software Trust Manager reduce los riesgos de que se produzcan brechas de seguridad y de que se propague el malware durante el desarrollo, la creación y la publicación del software, lo que permite a las empresas protegerse contra los ataques a la cadena de suministro de software y cumplir con las normativas gubernamentales.

DigiCert Software Trust Manager ayuda a las grandes empresas a implementar fácilmente estas cuatro prácticas recomendadas.

Póngase en contacto con pki_info@digicert.com para obtener más información.

Figura 3: Software Trust Manager

DigiCert Software Trust Manager protege todo el ciclo de desarrollo del software ofreciendo una firma de código segura, detección de amenazas completa, transparencia del software, y visibilidad y control de la empresa.