4 BEST PRACTICE PER PROTEGGERSI DAGLI ATTACCHI ALLA SUPPLY CHAIN DEL SOFTWARE

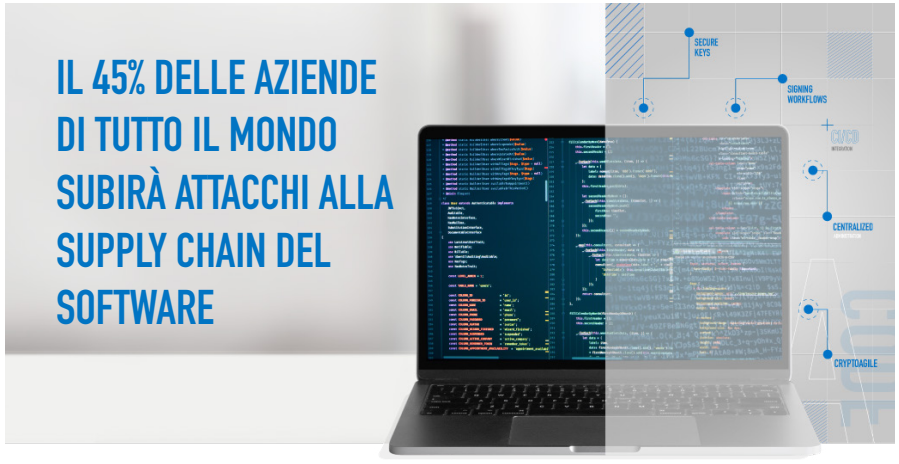
digicert®

Integrare firma sicura del codice, rilevamento di minacce e malware e generazione di SBOM

Gli attacchi alla supply chain del software sono sempre più frequenti. Secondo un recente report di Gartner, il 45% delle aziende di tutto il mondo subirà attacchi alle proprie supply chain del software entro il 2025, il triplo rispetto al 2021.

Questi attacchi coinvolgono aziende sia grandi che piccole. Oggi tutti sono ormai a conoscenza degli attacchi alle supply chain di SolarWinds, CircleCI e 3CX. Più di recente, Micro-Star International (MSI) ha subito un attacco che ha compromesso le chiavi private di firma del codice usate per il suo BIOS e le chiavi usate per il BootGuard di Intel.

Anche se le misure di sicurezza di base come Zero Trust assicurano alle aziende una prima linea di difesa, sono necessarie ulteriori azioni per proteggere la supply chain del software e il ciclo di vita del suo sviluppo.



A CHI SI RIVOLGE QUESTO DOCUMENTO:

Professionisti delle aree Security, DevSecOps e software engineering, responsabili o con un interesse per la protezione dell'infrastruttura di sviluppo software della loro azienda da violazioni, attacchi e altre vulnerabilità.

SFIDE

I criminali informatici usano varie strategie per attaccare le supply chain del software. Possono mirare alla password compromessa di uno sviluppatore software per introdursi in un sistema di compilazione del software, rubare o usare impropriamente chiavi private di firma del codice non protette, inserire malware a monte nei più diffusi pacchetti di software open-source o adottare un insieme di tutte queste strategie. L'attacco Sunburst a SolarWinds nel 2019 ha evidenziato che le strategie di attacco sono sempre più sofisticate e complesse e richiedono metodi più sofisticati per essere contrastate, come illustrato nella Figura 1.

Figura 1: Gli attacchi alla supply chain possono avvenire in ogni fase

Gli attacchi possono verificarsi in ogni fase del ciclo di vita dello sviluppo software. Gli attuali strumenti di sicurezza sono in grado di rilevare specifici tipi di attacco ma non tutti. Serve quindi un approccio "shift up" con una visione dall'alto, per introdurre misure di sicurezza in tutto il ciclo di vita dello sviluppo software.

Altre sfide sono di tipo organizzativo e culturale: i team di sviluppo software sono restii all'uso di strumenti di sicurezza perché trovano che siano poco pratici, interrompano le pipeline di compilazione CI/CD automatizzate o rallentino il processo di rilascio del software.

I team di sicurezza hanno tutta un'altra serie di sfide. Devono fornire indicazioni sulla sicurezza e far rispettare la conformità a molti team di software sparsi nell'azienda e distribuiti geograficamente. Inoltre, questi team utilizzano spesso linguaggi di programmazione e ambienti diversi, distribuiscono il software su piattaforme diverse e utilizzano infrastrutture di compilazione del software e processi software diversi. Questa eterogeneità può rendere molto difficile la definizione, l'applicazione e il monitoraggio delle policy di sicurezza del software in tutta l'azienda.

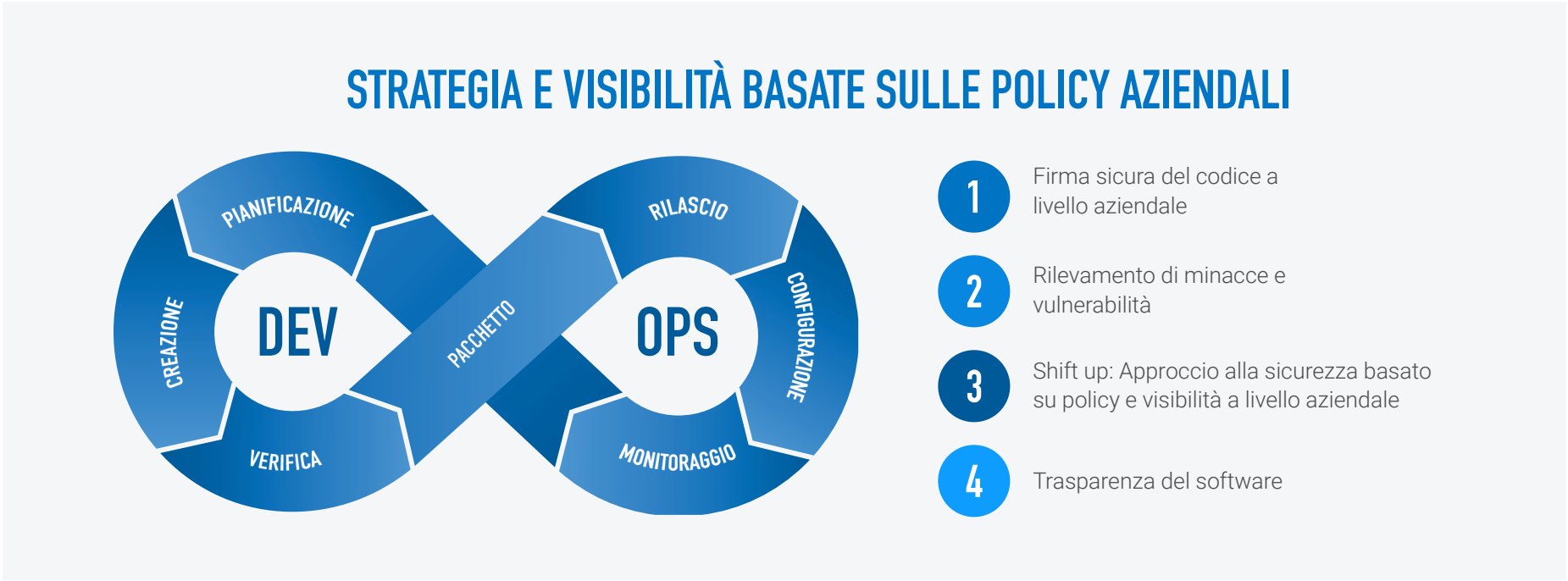
Infine, un altro tipo di sfida è quella della conformità alle varie normative governative e di settore nate per contrastare gli attacchi alla supply chain del software, come quelle che richiedono la trasparenza del software.

BEST PRACTICE PER PROTEGGERE LE SUPPLY CHAIN DEL SOFTWARE

Diverse normative e raccomandazioni del settore specificano non solo le best practice, ma anche i requisiti che le aziende devono seguire per garantire integrità alla supply chain del software. Ad esempio, l'ordine esecutivo degli Stati Uniti per migliorare la cybersecurity (Executive Order on Improving the Nation's Cybersecurity) e la legge sulla resilienza informatica dell'UE (Cyber Resilience Act) richiedono la generazione di distinte base (SBOM) per dimostrare la trasparenza del software. Il documento Security Considerations for Code Signing del NIST illustra i passi da seguire per una firma sicura del codice.

Questo documento delinea quattro best practice che le aziende possono adottare per affrontare tali sfide, con un focus su come DigiCert può essere d'aiuto. L'aspetto chiave di queste best practice è che si attuano lungo tutto il ciclo di vita dello sviluppo software, e non in una singola fase dell'SDLC (vedi Figura 2).

Figura 2: Attendibilità del software: sicurezza sul lato sinistro o destro dello sviluppo... E DALL'ALTO



Per combattere gli attacchi alla supply chain del software mirati a vari aspetti dell'SDLC serve un approccio multiforme, integrato e di nuova generazione.

N. 1: Firma sicura del codice in tutta l'azienda

La firma del codice è una misura di sicurezza di base in uso da oltre 30 anni. Il sistema si basa sulla crittografia per firmare digitalmente un pacchetto software (eseguibile, libreria, applicazione, ecc.) e dimostrare che il software è autentico (es., il software firmato da DigiCert proviene effettivamente da DigiCert) e non è stato manomesso da terzi. Molti sistemi operativi come iOS, macOS o Windows, controllano la firma digitale del software prima di installare ed eseguire il pacchetto.

Questo metodo è così efficace che gli hacker hanno preso di mira i sistemi di firma del codice per rubare, compromettere o usare impropriamente i certificati di firma del codice e le chiavi private delle aziende.

Il primo passo per proteggere la firma del codice è accertarsi di aver conservato in modo sicuro tutte le chiavi private di firma del codice. Un recente requisito del Certificate Authority/Browser Forum richiede l'archiviazione di chiavi di firma private pubblicamente attendibili in un dispositivo FIPS 140 Level 2, Common Criteria EAL 4+ come un modulo di sicurezza hardware (HSM).

Se è vero che un sistema di archiviazione sicuro è il primo passo necessario, non è tuttavia sufficiente per rendere sicura la firma del codice in tutta l'azienda. Il motivo è che un nome utente o una password compromessi potrebbero consentire a un criminale informatico di accedere all'archivio sicuro.

È consigliabile quindi l'uso di un sistema di firma del codice di livello enterprise che offra una serie di misure di sicurezza aggiuntive, tra cui:

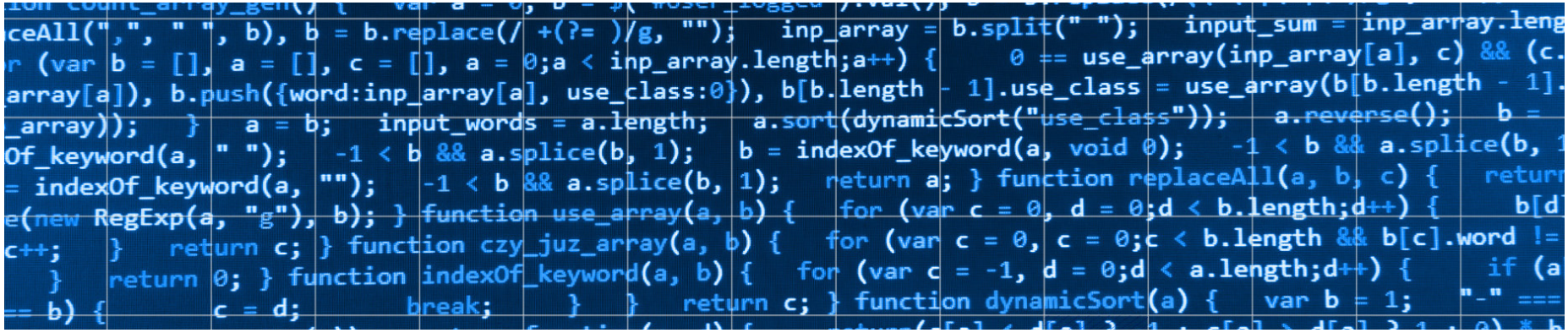
- Automazione della gestione dei certificati di firma del codice, incluse l'emissione e la revoca.
- Un processo di approvazione che specifica chi può accedere a una chiave di firma del codice ma anche chi deve approvare l'accesso e in quali circostanze è possibile accedere alla chiave (es., l'ora del giorno, la macchina di build, ecc.)
- Un sistema basato sui ruoli che specifica quali utenti possono accedere a determinati certificati e chiavi di firma del codice per le operazioni di firma, quali sono autorizzati ad approvare le operazioni di firma e quali sono autorizzati a monitorare.
- Mantenimento di un registro inconfutabile di tutte le attività di firma del codice cui fare riferimento in caso di certificato compromesso.
- Un unico sistema di firma del codice che supporta un gran numero di operazioni di firma del codice e molte piattaforme e strumenti diversi di distribuzione e sviluppo del software.
- Un sistema di misure di controllo basato su policy che devono essere soddisfatte prima che il software possa essere rilasciato.
- Firma di artefatti software intermedi lungo l'intero SDLC, incluso il codice sorgente prima del commit, di script e ricette di compilazione, dei file dei container e di altre infrastrutture software.
- Visibilità di livello aziendale, definizione e applicazione di policy da una postazione centralizzata, ovunque si trovino i vari team software.

N. 2: Rilevamento completo delle minacce

La firma del codice può prevenire la manomissione dopo il rilascio del software, ma cosa succede se il software viene manomesso prima di essere firmato e rilasciato? Ad esempio, un cattivo attore che accede a un sistema di compilazione violato potrebbe inserire un malware nel repository del codice sorgente di un'azienda. Oppure uno sviluppatore di software potrebbe scaricare un pacchetto open-source che contiene un malware nascosto. O ancora potrebbe essere scoperta una vulnerabilità nascosta non rilevata in precedenza.

Molti strumenti di sicurezza oggi disponibili sul mercato rimediano ad alcuni di questi problemi, come gli strumenti di analisi statica del software (SAST). Tuttavia, questi strumenti spesso operano su un singolo aspetto (come le vulnerabilità del software open-source).

È invece necessario un metodo completo che scansioni i file binari del software finale alla ricerca di minacce, vulnerabilità, presenza di segreti incorporati e malware. Questa scansione va attuata indipendentemente dalla fonte originale dei componenti del software: codice open-source, librerie commerciali di terze parti o codice proprietario scritto dall'azienda.



N. 3: Trasparenza del software

Come detto prima, le normative governative e industriali richiedono sempre più spesso agli editori di software di rivelare il contenuto del loro software, proprio come un'etichetta nutrizionale indica gli ingredienti degli alimenti. Le Software Bills of Materials (SBOM) sono distinte che catalogano tutti i componenti di un software, a prescindere dalla fonte del componente (es., open-source, libreria di terze parti, ecc.). Come l'etichetta nutrizionale di un alimento, le SBOM non solo indicano un elenco di "ingredienti" software, ma possono anche essere utilizzate per valutare la qualità di tali ingredienti, come le versioni utilizzate, le vulnerabilità note, ecc.

Anche se non richiesto ai fini della conformità, le organizzazioni dovrebbero generare SBOM in modo da:

- **Valutare e prioritizzare la superficie di attacco del proprio software**: cercare ad esempio versioni vulnerabili o prese di mira di librerie o componenti, controllare le dipendenze del software e verificare se mancano patch di sicurezza obbligatorie.
- **Cercare componenti obsoleti**, come mitigazioni mancanti e pratiche di firma del codice non sicure.
- **Adottare misure di difesa proattive** utilizzando la struttura del software per supportare i team di rilevamento e risposta agli incidenti e per la modellazione delle minacce.

N. 4: Shift Up: controllo e visibilità su tutta l'azienda e l'SDLC

Molte misure di sicurezza riguardano/proteggono una parte specifica dell'SDLC. Riguardo al momento in cui integrare la sicurezza abbiamo sentito parlare degli approcci shift-left e shift right nel corso del processo DevOps, ma concentrarsi solo su uno o l'altro aspetto è come guardare gli alberi di una foresta singolarmente. Gli attacchi sempre più complessi di oggi richiedono alle aziende un cambiamento che consenta di guardare tutto dall'alto:

- Visione sull'intero processo SDLC, guardando al software nel suo complesso.
- Visibilità su policy di sicurezza, misurazioni e risultati in tutta l'azienda.
- Capacità di definire e applicare le policy di sicurezza nell'intera azienda.
- Esecuzione di analisi delle minacce e delle vulnerabilità sugli eseguibili finali del software appena prima che vengano firmati e rilasciati.

Per fare tutto questo con efficacia e per essere accettati dai team di sviluppo software, gli strumenti di sicurezza devono integrarsi perfettamente nei processi software esistenti e non rallentare il processo di rilascio del software. Gli sviluppatori devono trovare i software facili, se non trasparenti.

DIGICERT SOFTWARE TRUST MANAGER

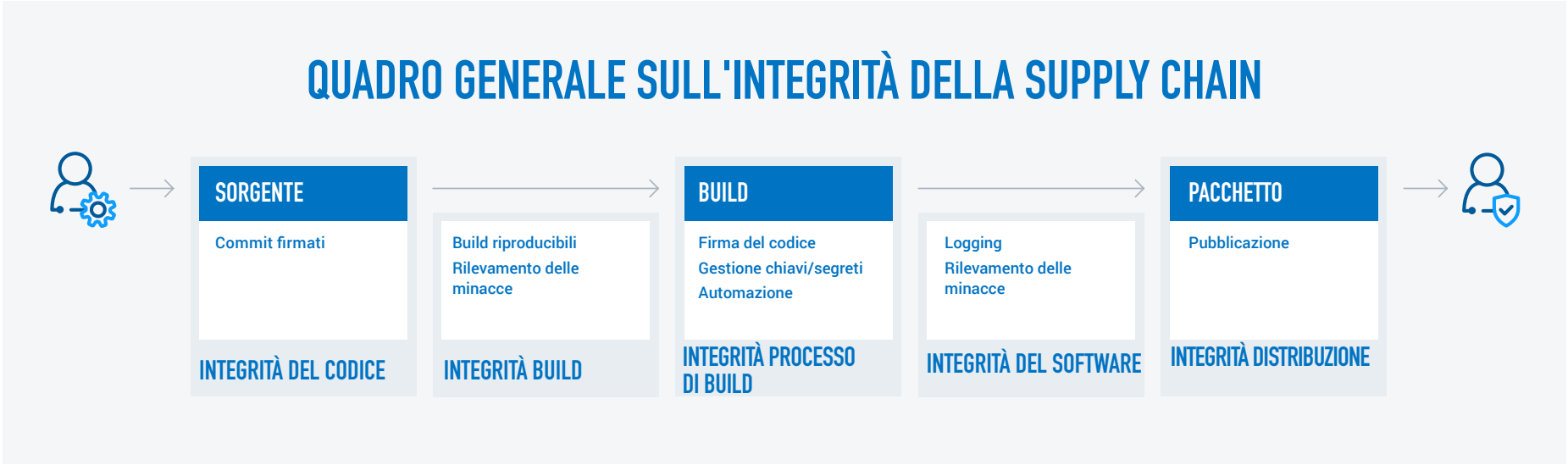
DigiCert® Software Trust Manager è una soluzione di fiducia digitale che protegge l'integrità del software lungo tutta la sua supply chain, riducendo il rischio di compromissione del codice, applicando le policy aziendali e di legge e fornendo controlli granulari su accessi e utilizzo delle chiavi per la firma del codice.

DigiCert Software Trust Manager riduce al minimo i rischi di violazioni della sicurezza e propagazione di malware nelle fasi di sviluppo, compilazione e rilascio del software, consentendo alle aziende di tutelarsi dagli attacchi alla supply chain del software e di rispettare le norme di legge.

DigiCert Software Trust Manager aiuta le grandi aziende a implementare facilmente queste quattro best practice.

Per ulteriori informazioni, invia un'email a pki_info@digicert.com.

Figura 3. Software Trust Manager



DigiCert Software Trust Manager protegge l'intero ciclo di vita dello sviluppo software tramite firma sicura del codice, rilevamento completo delle minacce, trasparenza del software e visibilità e controllo di livello aziendale.