CRYPTONET
a DIGITAL360 company LABS

Application Security

è ora di un SOC applicativo per affrontarla efficacemente

White Paper

Abstract

Tutta la nostra quotidianità è pervasa dall'utilizzo di **applicazioni**, in particolare web e mobile, che ci consentono di gestire e trattare quantità crescenti di dati per le più disparate finalità. Occuparsi della loro sicurezza e conformità a standard o regolamenti è diventato un imperativo.

Questo white paper esplora le **criticità** che affliggono il loro ciclo di vita e il loro ciclo di sviluppo, perché in ogni fase è concreto il rischio di introdurre vulnerabilità. Non solo descrive i problemi, ma propone anche un **percorso per affrontarli**, in modo graduale e sostenibile, tenendo conto della maturità di un'organizzazione sull'Application Security.

Questo percorso si basa sugli interventi modulari di un **Security Operation Center** (SOC) **Applicativo**, ovvero un team specializzato per affrontare la sicurezza applicativa in ogni fase, con particolare focus al punto nevralgico della remediation. L'obiettivo è chiaro: un impegno costante perché la sicurezza sia considerata prioritaria e per garantirne un adeguato livello una volta che le nostre applicazioni o app sono portate in produzione.

Indice

1 Introduzione

Ci fermiamo mai a chiederci con quante applicazioni abbiamo a che fare ogni giorno? Che siano applicazioni web esposte su Internet, mobile app, applicativi desktop o di altra natura, è parte della nostra esperienza quotidiana interagire, per finalità lavorative o personali, con numerose applicazioni.
D'altronde queste sono lo strumento con cui accediamo, fruiamo, modifichiamo ed elaboriamo i dati che sono necessari per qualsiasi attività. Se pensiamo poi alla immediatezza e pervasività delle mobile app, usate da una enorme base di utenti, di diverse età e con differenti tassi di alfabetizzazione informatica, quanto appena affermato diventa ancora più evidente.

Il portfolio di applicazioni di qualsiasi organizzazione costituisce un asset strategico, per consentire di veicolare a utenti finali, dipendenti, collaboratori, fornitori e clienti le informazioni fondamentali per realizzare la propria attività.
Ciò è amplificato ulteriormente dal contesto di trasformazione e accelerazione digitale che spinge a ripensare qualsiasi tipologia di business con un'attenzione crescente a produrre, valutare, consumare e aggregare dati in quantità sempre maggiori, al fine di prendere decisioni sempre più informate (si pensi ad esempio al fenomeno IoT o a Industria 4.0).

La sicurezza di questo portfolio applicativo non può essere ignorata in quanto i dati, fondamentali per il business o critici per la loro sensibilità, nonché la proprietà intellettuale, sono costantemente esposti al rischio di furto, diffusione o alterazione da parte di attori malevoli. Ciò potrebbe comportare perdite finanziarie, impatti sulla responsabilità legale, violazioni di conformità, danni alla reputazione, riduzione del valore aziendale e della fiducia dei clienti.

2 L'importanza della Sicurezza Applicativa

Sappiamo bene che durante il ciclo di sviluppo software (in seguito anche abbreviato come **SDLC**, ovvero Software Development Life Cycle) ci possono essere scelte di progettazione architetturale e funzionale che si traducono in vulnerabilità, così come il team di sviluppo può inconsapevolmente introdurre codice insicuro o dare per scontati alcuni comportamenti dell'utente che diventano bug di sicurezza. Tutto questo in pratica rende qualsiasi applicazione più esposta ad attacchi, soprattutto quando la stessa è in esecuzione.

> La sicurezza del proprio portfolio applicativo è un'esigenza, e spesso un obbligo, non più trascurabile.

Anzi, anche nella fase di esercizio ci possono essere ulteriori problemi di sicurezza, legati a mis-configurazioni dell'ambiente di produzione, che sia un application server on-premises o un cluster Kubernetes in cloud, oppure legati a mancati aggiornamenti dello stack tecnologico e del middleware utilizzati. Essendo attività basate su una forte componente di fattore umano, il rischio che tutto ciò accada non può mai essere considerato pari a zero.

Visto che le applicazioni sono un asset strategico e visto che i dati che esse ci consentono di trattare impongono delle responsabilità a qualsiasi organizzazione, **è imprescindibile affrontare il problema della loro sicurezza e farlo in modo proattivo e strutturato durante tutto il loro ciclo di vita**.

Quanto affermato è così importante che diversi enti o lo stesso legislatore si sono attivati già da tempo per richiedere direttamente o indirettamente che la sicurezza applicativa sia considerata prioritaria e imprescindibile.
Il panorama normativo europeo, infatti, è in rapida evoluzione a causa della pubblicazione, o dell'imminente rilascio, di regolamenti come **GDPR, PSD2, DORA, il Cyber Resilience Act o la direttiva NIS2**. In ambito finanziario e assicurativo, questi regolamenti impongono nuovi obblighi da rispettare. Oppure, spostandosi sugli standard di settore, ulteriori esempi sono PCI DSS, che stabilisce best practices per la sicurezza delle applicazioni nel requisito 6, o ISO 27001, che fornisce una serie di controlli in merito all'interno della sezione 8.

Tali iniziative spingono costantemente le organizzazioni ad adeguarsi a regole rigorose e a implementare misure di sicurezza più stringenti per proteggere i dati e garantire la conformità delle proprie applicazioni.

3 Il ciclo di sviluppo software e le sue criticità

Per indirizzare adeguatamente la sicurezza applicativa dobbiamo inevitabilmente allargare il nostro sguardo ad aspetti gestionali e problematiche del mondo dello sviluppo software, che emergono nella quotidianità. Ecco alcuni interrogativi da tenere presenti:

- **Come intervenire** in un ciclo di vita e di sviluppo che è articolato in diverse fasi, con tanti attori coinvolti?
- Parlando di stakeholder, **che ruolo ha ognuno** e quali sono le responsabilità per attuare Application Security e governare la sicurezza del portfolio applicativo nel tempo?
- La quasi totalità delle applicazioni si appoggia su componenti attinte all'esterno. Si pensi in particolare a quanto ormai sono indispensabili i diversi progetti opensource che i nostri team utilizzano costantemente. Ci siamo mai posti la domanda sulla loro **affidabilità**, sia in termini di manutenzione nel tempo sia in termini di risoluzione delle vulnerabilità e delle criticità?
- Infine, **ogni organizzazione ha una propria storia**, attenzione e sensibilità al tema della sicurezza applicativa che sono differenti, nonché una disponibilità di risorse per occuparsene che è necessario tenere in considerazione.
  Come quindi muoversi secondo un percorso di crescita graduale?

Affrontiamo brevemente questi punti.

3.1 La complessità del SDLC

Intervenire in un ciclo di sviluppo e di vita di un'applicazione richiede di considerare la sua articolazione. Ci sono aspetti di impostazione (governance) che sono presenti ancora prima di partire con un qualsiasi progetto software. Ovviamente poi si entra nel merito con opportune fasi di design, implementazione, test/collaudo e poi rilascio in produzione. A questo segue la necessaria manutenzione nel tempo.
Il tutto in modo iterativo, perché sappiamo che nessun applicativo è immutabile e quindi esistono procedure più o meno precise per il change management.

Oltre alla frammentazione, abbiamo altri elementi di complessità, soprattutto in organizzazioni che hanno più di un team di sviluppo:

- **Processi e metodologie** di sviluppo non consolidati e non uniformi.
- **Strumenti differenti** e non centralizzati.
- Una pluralità di **attori coinvolti**: sviluppatori, analisti, loro responsabili, owner delle applicazioni (ad es. product manager), il team di Operations, la funzione Cybersecurity, l'Internal Audit, il marketing, senza tralasciare il management o il board esecutivo. Tutti hanno, a seconda del loro ruolo e posizione, diversi compiti e responsabilità legate all'Application Security.

3.2 Diversi stakeholder e diverse priorità

Oltre ad avere più figure coinvolte con una applicazione, queste hanno esigenze differenti che le portano a "parlare lingue" che spesso non sono intelligibili tra loro:

- **Gli owner applicativi** – si interfacciano con il business; stabiliscono priorità e tempistiche per lo sviluppo; sono consapevoli delle tipologie di dati trattati e dei requisiti normativi; ricevono la pressione del marketing per andare "live" con una nuova funzionalità, ecc.
- **I team di sviluppo ed esercizio** – sanno come definire e implementare i requisiti funzionali, nonché gestire le applicazioni nel quotidiano, ma sono sempre a risorse limitate e quindi costretti a correre per un nuovo rilascio.
- **La funzione Cybersecurity** – conosce ed è in grado di valutare quali vulnerabilità possono essere introdotte nel SDLC, le loro severità, i rischi associati e gli impatti in termini di compliance; deve però "lottare" perché siano svolti Security Test (come Vulnerability Assessment, Penetration Test o Code Review) in un regime di tempi contingentati.

Questi sono solo alcuni esempi, ma è chiaro che allo stesso tavolo ci sono stakeholder che hanno priorità diverse e non convergenti ed è necessario trovare una modalità di confronto e la definizione di azioni condivise, che siano sufficientemente flessibili.

3.3 Opensource e Supply Chain Security

Consideriamo un dato di fatto: la maggior parte, se non tutte, le applicazioni di un'organizzazione contengono codice di natura open source, anche per funzionalità fondamentali per il business, e questo avviene in misura sempre maggiore.
Un modus operandi ragionevole, perché è inutile reinventare la ruota, ma che espone a possibili rischi in termini di affidabilità e sicurezza:

- **I progetti opensource** che integriamo hanno delle community solide alle spalle? Esisteranno ragionevolmente ancora tra cinque o dieci anni? Oppure sono basati sul lodevole sforzo di tre o quattro sviluppatori nel loro tempo libero?

> Costriuiamo le nostre applicazioni con componenti affidabili anche da un punto di vista di sicurezza?

- Se emerge una vulnerabilità critica in una componente opensource (si pensi ad **Heartbleed** per OpenSSL o a **Log4Shell** per Apache Log4j), sappiamo dove abbiamo utilizzato quella specifica versione, visto che le nostre applicazioni potenzialmente ereditano questo problema? La community di solito rilascia patch di sicurezza in tempi rapidi?

Domande che sono di assoluta importanza, perché se una dipendenza opensource ha una vulnerabilità critica nota, anche la base di codice in cui è inserita eredita tale problema.

Per questo l'argomento delle dipendenze software e della relativa **Supply Chain Security** sta diventando sempre più pressante. Si parla sempre più di **Software Bill of Materials** (SBoM), ovvero di inventariare durante tutto il ciclo di vita e sviluppo ciò che si sta incorporando nelle proprie basi di codice. Questo è già un obbligo nell'ambito delle agenzie federali statunitensi, ma anche l'Unione Europea con la direttiva NIS2 si muove in tal senso (e la versione 2022 dello standard ISO 27001 nel controllo 5.21 non si esime dall'affrontare il tema).

3.4 Maturità differenti

Parlando di sicurezza applicativa è necessario tenere presente quanto ha già fatto e sta facendo un'organizzazione con i suoi team. Esistono infatti sensibilità e maturità differenti. Come succede anche in altri contesti, moltissime realtà sono "all'anno zero", alcune hanno svolto un Penetration Test, altre hanno uno sviluppatore che si interessa di vulnerabilità applicative e cerca di spiegare i problemi ai colleghi, o poco altro ancora.

Esistono all'opposto realtà più virtuose, dove sono state definite delle policy di Application Security e un relativo programma di iniziative, dove si svolgono attività periodiche di **SAST** (Static Analysis Security Testing) e **DAST** (Dynamic Analysis Security Testing), formando e coinvolgendo opportunamente gli sviluppatori. Rimane, magari, da ottimizzare l'attuazione delle azioni di rimedio.

Sono semplici descrizioni di quanto osserviamo nella nostra esperienza con diverse organizzazioni, non solo in Italia, ma è evidente che per gestire adeguatamente un percorso di crescita graduale della sicurezza applicativa, la maturità attuale (e le conseguenti risorse a disposizione) è un fattore determinante.

3.5 Il nodo della remediation

Da quanto scritto finora emerge l'importanza di individuare le vulnerabilità che possono manifestarsi in un'applicazione in esercizio, già dai momenti iniziali della sua progettazione (o anche prima) e di protrarre la ricerca durante l'implementazione del codice o nei momenti di collaudo funzionale.

Questo continuo Security Testing e i relativi risultati, per quanto siano utili in termini di consapevolezza, se non sono accompagnati da una successiva attività di **Remediation**, rischiano di rimanere inefficaci. È infatti inutile accumulare una massa di problemi se poi non viene allocato tempo per permettere ai team di sviluppo di attuare gli opportuni fix o, se necessario, per incrementare le proprie competenze al fine di sapere come risolvere i problemi.

Siamo davanti al punto nevralgico di qualsiasi percorso di sicurezza applicativa: ogni organizzazione deve avere capacità di remediation e deve saperla governare.

4 Un riferimento metodologico per non improvvisare

Se finora abbiamo osservato gli elementi di criticità di un SDLC, come possiamo capire quali siano gli interventi opportuni per attuare un programma di sicurezza applicativa efficace?

Nel mare della conoscenza condivisa a livello internazionale dagli esperti di settore, sicuramente spicca il progetto OWASP, realtà a partecipazione collaborativa che cura diverse linee guida di Security Testing e best practices in ambito web, mobile o IoT. Tra queste, parlando di impostazione metodologica per definire e mantenere un Secure SDLC, emerge OWASP SAMM (https://owaspsamm.org/).

> Lo sviluppo sicuro: processi, persone e tecnologia. Le best practices aiutano a tenere insieme tutto.

L'acronimo significa "Software Assurance Maturity Model" e questa metodologia definisce che cosa fare, in che fase di un SDLC, da parte di chi e quali siano gli output attesi in un programma di Application Security. Questo è ottenuto modulando le iniziative in base al livello di maturità attuale e quello che si desidera raggiungere, secondo una adeguata roadmap, ad esempio a due o tre anni, e operando a livello di tutte le componenti convolte: processi, persone e strumenti. Partendo dal momento zero, cioè la Governance, passando per Construction, Implementation, Verification e Operations.

Sulle fondamenta di SAMM proponiamo un percorso sostenibile, che consenta a qualsiasi organizzazione, avendo ben presente la sua maturità e le sue esigenze, di modulare e attuare una crescita graduale specifica per la propria Application Security: il **SOC Applicativo**.

5 SOC Applicatico: un percorso graduale e sostenibile

SOC è una sigla molto conosciuta in ambito Cybersecurity e significa **Security Operations Center**. Normalmente è declinato in un contesto infrastrutturale IT o OT: si tratta di un centro di competenze che presidia il perimetro network di un'organizzazione, le sue postazioni utente o altri endpoint e opera in modo reattivo a fronte di segnalazioni opportune, mettendo in atto procedure di Incident Management.

5.1 Ma che cosa è un SOC Applicativo?

Nel contesto di un Secure SDLC, vogliamo invece introdurre il concetto di **SOC Applicativo**.

È sempre un competence center ma **focalizzato per presidiare la sicurezza delle applicazioni, di qualunque natura, e che si muove con un approccio proattivo**, modulando il suo intervento secondo le possibilità che vedremo nei prossimi paragrafi. Si tratta di un team, molto spesso esterno all'azienda a causa del grado di specializzazione che deve avere e mantenere, che opera sia da un punto di vista tecnico, sia soprattutto come supervisore e facilitatore della delicata, ma ineludibile, attività di remediation.

Il SOC Applicativo può operare in tutte le fasi di un ciclo di vita e sviluppo software, per indirizzare le iniziative di Application Security in termini di processi, persone e strumenti. Si può occupare di attività che spaziano dalla predisposizione di policy per un Secure SDLC, alla stesura di linee guida e formazione specialistica sul Secure Coding per gli sviluppatori o sul Threat Modeling per gli analisti, ma soprattutto si concentra sulla ricerca metodica di vulnerabilità, facendo leva su un adeguato Security Testing o sull'integrazione di piattaforme SAST e DAST nella toolchain che i team di sviluppo utilizzano.

Il SOC Applicativo è la direzione verso cui dovrebbe tendere qualsiasi organizzazione per raggiungere l'obiettivo di portare in produzione applicazioni e app sempre più sicure.

5.2 La necessità del Security Testing

Il Security Testing è un momento fondamentale per capire come le proprie applicazioni si comportano a fronte di attacchi cyber. È basato su Vulnerability Assessment (VA), ovvero attività automatizzate per censire vulnerabilità, e Penetration Test (PT), dove l'azione di ricerca manuale di opportuni specialisti mira a capire se le vulnerabilità siano sfruttabili. Sono definiti anche **DAST**, cioè Dynamic Analysis Security Testing, proprio perché svolti contro applicativi in esecuzione.

A quanto sopra si affianca lo Static Analysis Security Testing, o **SAST**, dove invece si effettua una revisione del codice dell'applicazione alla ricerca di ulteriori problemi di sicurezza (ad es. credenziali o segreti hard-coded).

È a questo livello che si può operare anche la Software Composition Analysis, o **SCA**, al fine di creare un opportuno inventario delle dipendenze opensource e delle vulnerabilità associate, ovvero il SBoM.
SAST e SCA possono partire o dal codice binario di un'applicazione (in particolare laddove esista una compilazione) o dal suo codice sorgente (ad es. per linguaggi interpretati), come le piattaforme più evolute e gli specialisti opportunamente formati consentono di attuare.

Lavorare, quando possibile, sulla forma binaria di un applicativo ha un vantaggio che riteniamo utile evidenziare: **consente infatti di valutare e validare il livello di sicurezza di software commerciali**, per i quali non si dispone solitamente del codice sorgente, o di applicativi rilasciati da propri fornitori, perché questi potrebbero evitare di rendere disponibile il sorgente, soprattutto in assenza di specifici obblighi contrattuali.
Inoltre, una binary code review permette di avere benefici come una maggiore precisione perché contiene solitamente meno rumore (cioè falsi positivi) e la possibilità di individuare problematiche di sicurezza introdotte durante la compilazione.

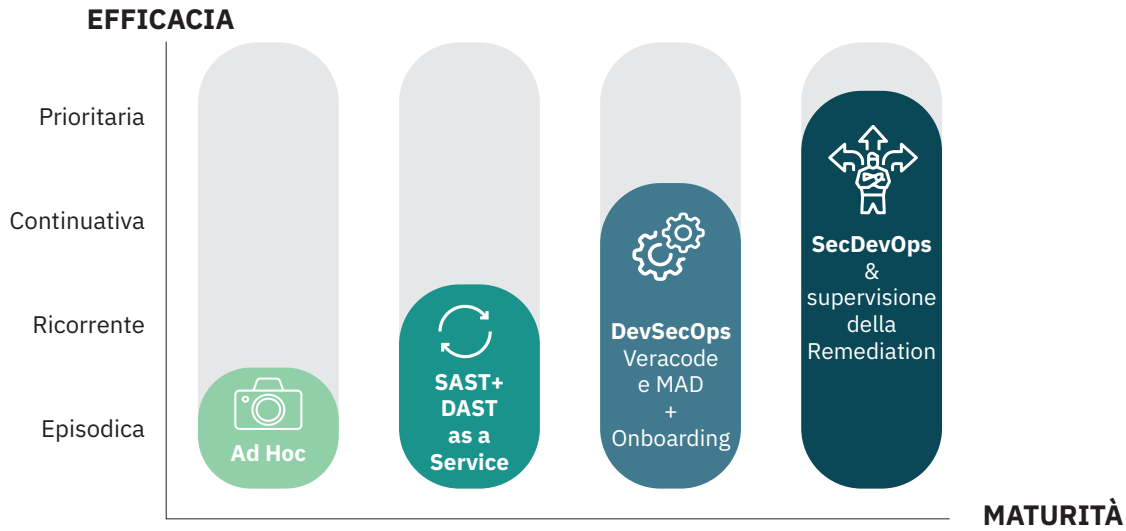
SAST, SCA e DAST sono attività complementari, non si escludono l'un l'altra:

- Ognuna, infatti, consente di **identificare vulnerabilità** che più difficilmente emergono con le altre, ottimizzando così la capacità di ricerca, soprattutto per quelle applicazioni che hanno un profilo di rischio maggiore e sui cui è opportuno investire in adeguati test di sicurezza.
- Se DAST può essere effettuato solo nelle fasi di UAT o collaudo, perché l'applicativo deve essere portato in esecuzione, SAST e SCA si possono svolgere **durante lo sviluppo anche su versioni intermedie**. Consentono quindi di mettere in pratica il paradigma dello "Shift Security Left": in un SDLC identificare problemi di sicurezza prima consente di avere un effort e un costo per il loro fix inferiori, rispetto all'indirizzare il medesimo problema una volta che si è passati in produzione.

5.3 I passi del percorso

Vediamo ora come declinare **SAST**, **SCA** e **DAST** nel **SDLC** con il supporto di team specializzati (interni o più facilmente esterni) e secondo un percorso graduale, tenendo ben presente il livello di maturità di un'organizzazione e di conseguenza le sue esigenze e risorse a disposizione, per operare in modo sostenibile.

Avremo come riferimento il seguente grafico, che andremo a illustrare nei prossimi paragrafi.



5.3.1 Approccio "ad hoc"

Il primo livello di modulazione di un SOC Applicativo è basato su test di sicurezza delle applicazioni eseguiti da uno specifico team di specialisti, attività che possiamo definire "episodiche".

Il Security Testing, sotto forma di SAST, SCA o DAST (cioè VAPT) è un pilastro di qualsiasi iniziativa di Application Security: **consente di acquisire consapevolezza dei problemi**, tramite opportuni report di output e permette di ottenere un set di indicazioni che, se implementate, innalzano il livello di sicurezza tecnologico di un'applicazione.

Il SOC Applicativo può facilitare l'attuazione di remediation fornendo ai team di sviluppo i chiarimenti necessari e un confronto sulle modalità migliori per il fix, solitamente attraverso incontri dedicati.
Ma ricordiamoci che **le attività episodiche rimangono sempre e solo una fotografia** in un determinato istante.

5.3.2 Proattività e modalità "as a Service"

Dato che qualsiasi applicazione è oggetto di continui cambiamenti o miglioramenti, e l'apparire di nuove vulnerabilità è all'ordine del giorno, diviene fondamentale non limitarsi ad attività di Security Testing ad hoc, ma è indispensabile iterare queste attività nel tempo (e in alcuni casi è addirittura mandatorio, ad esempio per chi tratta dati di carte di credito, dove lo standard PCI DSS impone di eseguire VA trimestralmente e PT semestralmente o annualmente).

Iterazione che ha senso raccordare alle pianificazioni dei rilasci, in modo che lo stato di vulnerabilità del proprio portfolio applicativo sia costantemente aggiornato.

Svolgere verifiche ricorrenti e frequenti è fondamentale quando si parla di sicurezza applicativa.

La proattività è l'elemento che caratterizza il secondo livello su cui può essere modulato un SOC Applicativo, che definiamo come SAST, SCA o DAST "**as a Service**".

Queste sono sì eseguite in modo ricorrente, producendo i report attesi, ma in questo caso il team di SOC Applicativo aggiunge ulteriore valore perché osserva l'evoluzione delle applicazioni nel tempo. Diventano così di semplice realizzazione le seguenti richieste:

- **Confrontare** un'analisi con la successiva, isolando nuove problematiche di sicurezza introdotte nell'ultimo rilascio e confermando la chiusura di precedenti criticità.
- Avendo serie storiche a disposizione, produrre **KPI e dashboard** che rappresentano la dinamica dello stato di vulnerabilità dell'intero portfolio applicativo, utili per il governo dell'Application Security.
- **Seguire nel tempo i team di sviluppo** sull'attuazione delle remediation, trovando le modalità migliori di condivisione dei risultati dei Security Test, ad esempio convertendoli in ticket per il sistema di bug tracking in uso e già familiare (come Jira, ServiceNow o altro).

5.3.3 DevSecOps: lavoriamo nel continuo

Il bisogno di assicurare un time-to-market ridotto per le nuove versioni dei propri software, al fine di ottenere un vantaggio competitivo, spinge sempre più organizzazioni ad automatizzare la propria toolchain di sviluppo, dando vita al ciclo DevOps.

Per quanto descritto in precedenza si può facilmente comprendere che integrare la sicurezza nelle pipeline DevOps diventa quindi un elemento fondamentale e distintivo. L'obiettivo è rilasciare quotidianamente codice prodotto in modo sicuro, "by design", adeguatamente controllato, e implementare rapidamente aggiornamenti per correggere bug di sicurezza. Va da sé che i team di Development e di Operations hanno ora un ruolo attivo e precise responsabilità nella ricerca continua di vulnerabilità.

> Un ulteriore passo avanti: rendiamo quotidiano l'impegno per la sicurezza delle nostre applicazioni.

Trasformare un SDLC in un Secure SDLC, e DevOps in ciò che viene chiamato DevSecOps, è il terzo livello su cui si può modulare l'azione di un SOC Applicativo.

DevSecOps si basa fortemente sull'integrazione nella toolchain di CI/CD di piattaforme per attuare SAST, SCA o DAST, che non solo aggiungono automazione ai test di sicurezza, ma permettono anche ai programmatori di cambiare la loro mentalità, imparando a considerare la sicurezza nel loro lavoro quotidiano e a scrivere codice secondo le best practices che renderanno l'applicazione più resistente agli attacchi.

Le piattaforme più avanzate di SAST, DAST, IAST e SCA permettono:

- Di **integrarsi in modo rapido** attraverso l'uso di funzionalità native o di API appropriate, introducendo dei security gateway nel SDLC, a partire dal desktop dello sviluppatore, passando per i sistemi di building (come Jenkins, Gitlab, Azure DevOps, ecc.), fino ad arrivare alle fasi di UAT o collaudo, da cui poi far scattare il deployment.
- Di **ridurre i tempi necessari per le modifiche sul codice** richieste da attività di rimedio grazie all'uso di strumenti di intelligenza artificiale generativa, che può proporre al programmatore come riscrivere un insieme di istruzioni vulnerabili. Questo consente di contenere il costo di remediation per unità di codice scritto dal team di sviluppo.

In tale contesto il team di SOC Applicativo **lavora per agevolare queste integrazioni, sia in termini tecnologici sia raccordandosi ai processi di sviluppo software in essere, facilitando l'onboarding dei team di sviluppo** sulle piattaforme, in modo che inizino a beneficiarne quotidianamente.

5.3.4 SecDevOps: governance della remediation e priorità della sicurezza

In tutti i passi del percorso di SOC Applicativo finora descritti emerge l'importanza della remediation e si sarà compreso che risulta essere un punto nevralgico. Infatti, individuare vulnerabilità e classificarne la severità, anche in modo ricorrente o continuo, se non è supportato da un'adeguata attività di rimedio, rimane qualcosa di fine a sé stesso. **Ogni organizzazione deve investire sulla propria capacità di remediation e deve governarla adeguatamente**.

I team di sviluppo o i fornitori necessitano di confrontarsi sulle vulnerabilità e sulle diverse opzioni per risolverle. Devono allocare risorse e tempo e trovano difficolta a comunicare questa necessità all'interno dell'organizzazione. In pratica, emerge dai team un'esigenza di ricevere supporto costante sulla remediation.

Allo stesso modo la funzione Cybersecurity aziendale ha necessità di capire se le iniziative attuate stiano dando il risultato atteso. Il team di SOC Applicativo aiuta a raggiungere il livello più efficace e più maturo in un programma di Application Security e che definiamo come **SecDevOps**.

Sec non a caso è anteposto a DevOps, perché la sicurezza diventa una esigenza costante per tutti gli stakeholder coinvolti con le applicazioni, in tutte le fasi della loro vita, e il team di SOC Applicativo favorisce la definizione delle priorità di intervento. Consente quindi di trovare un linguaggio comune tra tutti gli interlocutori presenti al tavolo.

Questo passo conclusivo del percorso proposto si orienta su queste caratteristiche:

- **Flessibilità** – considerando il portfolio di applicazioni e il profilo di rischio di ognuna, nonché le risorse disponibili, ci si potrà muovere su alcune commissionando attività SAST, SCA e DAST al team di SOC Applicativo mentre su altre sarà opportuno coinvolgere gli sviluppatori (o ancora prima gli analisti durante la progettazione), dotando loro degli opportuni processi e piattaforme di DevSecOps.
- **Agevolazione** – il team di SOC Applicativo segue in modo cadenzato o costante le attività e gli sviluppatori, in modo da fornire chiarimenti e verificare che le remediation siano comprese e messe in opera (anche attraverso specifici test puntuali di follow-up).
- **Supervisione** – lo stesso team si relaziona periodicamente con i responsabili dello sviluppo, gli owner delle applicazioni, la funzione Cybersecurity ed eventualmente l'Internal Audit, per evidenziare l'efficacia dell'azione di remediation, o le sue carenze, e la necessità di interventi correttivi.

Il fine di tutto il percorso esposto è riuscire a governare in modo proattivo e con un processo strutturato, se possibile continuo, il "corpus" di vulnerabilità del proprio portfolio applicativo e mantenerlo al di sotto di un'adeguata soglia di accettazione del rischio.

5.4 Il ruolo delle piattaforme

Da quanto finora descritto, in un programma di sicurezza applicativa l'automazione dei test è un fattore essenziale per garantire applicazioni più resistenti senza acquisire incertezza sui tempi di sviluppo e di rilascio.
Ciò è possibile solo facendo leva su piattaforme specifiche con caratteristiche avanzate, che abbiamo in parte già delineato:

- La capacità di **attuare in modo olistico** diverse modalità di Security Testing, come SAST, SCA e DAST.
- La possibilità di **partire dal codice binario**, per beneficiare di risultati più precisi e per operare anche nei casi in cui le applicazioni siano sviluppate esternamente e c'è necessità di validarne la sicurezza.
- La flessibilità per consentire **integrazioni nelle pipeline di sviluppo** e con gli strumenti usati dai team per produrre le applicazioni.
- Reporting efficace, per **confrontare analisi ricorrenti** nel corso del tempo e rapidamente evidenziare vulnerabilità introdotte in una nuova release e quanto invece è stato fissato.
- Rappresentazioni di adeguati KPI per **monitorare il processo di Security Testing e di remediation**, avendo così gli elementi decisionali opportuni in ottica di Governance e di gestione del rischio.

5.5 Deployment: hardening, security dei container e di IaC

La sicurezza non è confinata solo alle fasi di sviluppo, ma riguarda tutto il ciclo di vita di un'applicazione e quindi anche il momento del deployment in produzione.
Se infatti il contenitore che ospita il nostro applicativo ha dei problemi, siamo comunque ancora esposti a dei rischi. Per questo le attività di un SOC Applicativo si estendono anche al mondo delle Operations.
Senza pretesa di esaustività, sottolineiamo due azioni di cui un team di SOC Applicativo, con il supporto di adeguate piattaforme, può occuparsi:

- **La revisione dell'hardening**, che andando ad esaminare le configurazioni dello stack tecnologico utilizzato ed evidenziando i gap rispetto a precisi standard di riferimento (uno su tutti, i CIS Benchmarks), consente di valutare impostazioni che riducono la superficie di attacco a disposizione.
- **L'analisi del livello di sicurezza** dei container e degli script "Infrastructure as Code" usati per il Continuous Deployment, perché l'uso di componenti vulnerabili, le mis-configurazioni, la presenza di password, chiavi crittografiche o altri segreti hard-coded sono problemi sempre dietro l'angolo.

Le piattaforme del nostro SOC Applicativo

I servizi di SOC Applicativo erogati dal team di specialisti di CryptoNet Labs si basano su alcune soluzioni specifiche per analisi SAST, SCA e DAST, tra cui evidenziamo:

- **Veracode** (https://www.veracode.com/), nostro partner strategico, la cui piattaforma consente di intervenire nelle fasi di un SDLC, fornendo ai diversi team coinvolti gli strumenti adeguati a introdurre opportuni security gateway, tramite le funzionalità di IDE Scan, Pipeline Scan e Policy Scan, che operano partendo dal codice binario (laddove applicabile) o dal codice sorgente dell'applicazione.
- **MAD** (Mobile App Driller - https://www.appdriller.it/) è la piattaforma all-in-one proprietaria di CryptoNet Labs per l'analisi della sicurezza delle mobile app, progettata per effettuare Security Test completi in maniera automatica utilizzando il pacchetto binario dell'app e senza restrizioni rispetto al sistema operativo impiegato (la piattaforma supporta sia Android, sia iOS).

Entrambe le piattaforme svolgono numerose tipologie di controlli atomici, coprendo anche l'analisi delle dipendenze al fine di generare il relativo Software Bill of Materials (SBoM).

L'automazione, la rapidità d'esecuzione e le capacità di integrazione CI/CD di Veracode e di MAD permettono agli sviluppatori di effettuare controlli di vulnerabilità a ogni nuovo rilascio e nel corso di tutto il ciclo di sviluppo delle applicazioni. L'output di SAST, SCA e DAST è strutturato per fornire risultati utili alle differenti tipologie di utenti, dagli specialisti software ai manager, sia presentando gli adeguati dettagli tecnici sia supportando la generazione di evidenze per la compliance con standard di settore e normative (sono supportati **OWASP, PCI DSS, GDPR, DORA, ecc.**).

Appoggiandosi a queste piattaforme, **il SOC Applicativo di CryptoNet Labs struttura e offre una serie di servizi modulari e incrementali** che consentono di inserire nel modo più efficace i test di vulnerabilità nei processi di sviluppo del cliente e di prendersi carico delle azioni necessarie per la remediation.

Chi è CryptoNet Labs

CryptoNet Labs è uno dei riferimenti nel panorama della Cybersecurity, vantiamo un'esperienza trentennale con diversi clienti e casi di successo sia sul mercato italiano che estero.
Siamo il centro di competenza tecnologico sulla Cybersecurity del Gruppo DIGITAL360, di cui facciamo parte. Il nostro team di esperti e consulenti affianca i clienti per definire, valutare e implementare i percorsi di crescita più adeguati al fine di proteggere gli asset digitali strategici e il valore del business.

CryptoNet Labs supporta i clienti nella creazione di un sistema informativo più sicuro, nel rispetto degli standard e delle normative, offrendo l'attenzione e la costanza che servono alla coltivazione delle relazioni professionali di lungo termine.
La nostra offerta è mirata alla costante ricerca di tecnologie innovative per rendere più sicuri e competitivi i nostri clienti spaziando dalla consulenza, ai servizi di Offensive e Defensive Security.

Le competenze e l'esperienza degli specialisti di CryptoNet Labs sono rivolte anche alla realizzazione di proprie piattaforme tecnologiche per la Cybersecurity, attraverso una costante ricerca dell'innovazione, per affrontare in modo adeguato la complessità delle esigenze dei propri clienti.

DIGITAL 360
EMPOWERING INNOVATION

CryptoNet Labs è parte del Gruppo DIGITAL360, Società Benefit che promuove l'innovazione digitale come motore di sviluppo sostenibile e inclusivo dell'economia e di rinnovamento delle Imprese e Pubblica Amministrazione del Paese.

6 Conclusioni

L'esigenza di tutelare la sicurezza delle applicazioni passa attraverso tutte le fasi di un SDLC, spaziando tra Security Testing, revisione del codice, esame delle dipendenze, impostazione dei processi di sviluppo e delle policy associate, definizione di linee guida, formazione e molto altro ancora.

In particolare, è essenziale integrare best practice e strumenti di controllo già all'interno del ciclo di sviluppo software, aiutando i programmatori a creare codice più sicuro e ad accrescere le proprie competenze in merito. Diventano importanti fattori quali l'automazione dei test, l'adozione di metodologie come DevSecOps, il supporto e il governo della remediation da parte di un team specializzato (il SOC Applicativo) che consentono di non rallentare i cicli di rilascio, garantendo agli utenti finali versioni sempre aggiornate e il più possibile sicure delle applicazioni che usano ovunque e quotidianamente.

Fattori che inoltre permettono di dare evidenza internamente ed esternamente della conformità a standard di settore o a normative cogenti e di raggiungere e mantenere nel tempo i livelli di rischio tecnologico del proprio portfolio applicativo al di sotto di adeguate soglie di accettazione.

POWERED BY

CRYPTONET

a DIGITAL360 company LABS

CryptoNet Labs s.r.l.

Via Agostino Bertani,6
20154 Milano

WEB www.cryptonetlabs.it
TEL +39.02.91527802
PIVA 09939700960

SALES CONTACT
sales@cryptonetlabs.it