Secure software development: la sicurezza come fattore abilitante

La filiera del *software development life cycle* (SDLC) va ripensata per rendere la sicurezza l'elemento chiave, comune e trasversale alle diverse fasi o settori di attività, permettendo alle organizzazioni di partecipare con serenità e successo alla trasformazione digitale in atto.

Troppo spesso, ancora oggi, la sicurezza è ritenuta una risorsa da attivare solo quando si manifestano i primi problemi: è invece evidente come l'intero ciclo di vita di prodotti e servizi vada sviluppato in un'ottica sicura, senza per questo sacrificare la competitività. Chiunque crei, venda o utilizzi strumenti software – ormai onnipresenti in qualsiasi tipologia di business – ha il massimo interesse ad abbracciare una visione che anticipi le problematiche all'origine, riducendone sensibilmente l'impatto, anziché inseguirle in ottica di *remediation*.

Proteggere l'SLDC per mettere in sicurezza la propria attività

I dati mostrano che le compromissioni software possono avere effetti "a catena" devastanti per le aziende colpite. Concepire la sicurezza quale fattore abilitante permette di razionalizzare tempi e costi, tanto economici quanto reputazionali, legati ad eventuali stop o rallentamenti in fase di rilascio; prevenendo l'ipotesi, ancora peggiore, che le vulnerabilità siano rilevate (se non sofferte) dagli utenti nel corso della loro esperienza.

Durante gli ultimi trent'anni, nel tentativo di rendere il processo di sviluppo sempre più rapido e performante, si sono

diffusi vari modelli di *Software Development Life Cycle*. Tuttavia ognuno di essi prevede alcune fasi e passaggi imprescindibili:

- pianificazione e individuazione dei requisiti;
- architettura e design;
- programmazione dei test;
- *coding*;
- *testing* ed elaborazione dei risultati;
- rilascio e manutenzione.

I primi modelli di SDLC prevedevano che le attività legate alla sicurezza partissero dalla fase di *testing* o ancora dopo, spesso portando a rilasciare prodotti caratterizzati da difetti o vulnerabilità. Ma, evidentemente, più tardi si scopre l'esistenza di un problema e più complesso (nonché costoso) diventa risolverlo: per questo la frontiera dei controlli di sicurezza è stata progressivamente anticipata, allineandola al processo di sviluppo sin dalla sua progettazione.

Dopo questa integrazione verticale, lo scenario odierno richiede un'ulteriore evoluzione in senso orizzontale che inserisca ruoli e processi di cybersecurity tra le attività di *Risk Management*, includendola nei flussi di lavoro di tutti i team coinvolti, per arrivare davvero a concepire la sicurezza come *mission* condivisa e strutturale.

I vantaggi di una simile concezione appaiono evidenti:

- maggiore affidabilità dei software;
- *stakeholder* più consapevoli delle politiche di sicurezza previste;
- precoce scoperta (e correzione) di eventuali debolezze o difetti nel codice;
- drastica riduzione del livello di rischio;
- conseguente contenimento dei costi.

Le basi per costruire un SDLC sicuro

Integrare l'ottica di sicurezza nella catena del software implica diverse attività, dall'integrazione dei requisiti di sicurezza tra quelli di funzionalità alla previsione della *risk analysis* già in fase di design. Al riguardo esistono autorevoli framework di riferimento (tra cui quello del NIST) e modelli notoriamente solidi, come il Microsoft Security Development Lifecycle (MS SDL) che suggerisce alle organizzazioni 12 buone pratiche per implementare un software più sicuro.



Quale che sia il modello o la fonte prescelta, è bene partire sempre da:

- la formazione di tutte le persone coinvolte in materia di secure coding practices e frameworks di sicurezza;
- una scrupolosa *risk analysis* preliminare sull'architettura del software;
- l'inclusione della sicurezza nella pianificazione e realizzazione di *Test Cases*;
- il ricorso a strumenti di *code scanning* per potenziare i test di sicurezza con metodologie di analisi statica, dinamica e interattiva.

Il passaggio successivo prevede di approfondire il livello di analisi per aggiornare la strategia di sicurezza aziendale, rimodellandola in base alle priorità riscontrate.

Come attuare (e valutare) le proprie attività di *secure software development?*

Dopo aver definito un *software security program* (SSP) e/o individuato gli obiettivi della propria *software security initiative* (SSI), è opportuno:

- formalizzare i relativi processi;
- includere metriche puntuali per la valutazione dei risultati;
- investire su tecnologie sempre aggiornate;
- prevedere una continua formazione dei team di sviluppo, anche tramite risorse esterne.

Per chi invece ritiene già di applicare un approccio sicuro allo sviluppo del proprio software, è estremamente utile poterne valutare affidabilità ed efficacia.

In questo senso può venir loro in aiuto *Building Security In Maturity Model* (BSIMM), che da oltre 10 anni osserva le policy di sicurezza aziendali per offrire una panoramica *data-driven* sui relativi punti di forza e fragilità: uno strumento prezioso per conoscere meglio il contesto, mettersi a confronto con realtà similari e identificare le risorse necessarie a perfezionare la propria postura di sicurezza.

Condividi sui Social Network:
https://www.ictsecuritymagazine.com/notizie/secure-software-development-la-sicurezza-come-fattore-abilitante/