Sviluppare la sicurezza (parte IV)

La rubrica **Digital Crime**, a cura di Paolo Galdieri, Avvocato e Docente di Diritto penale dell'informatica, si occupa del cybercrime dal punto di vista normativo e legale. **Clicca qui** per leggere tutti i contributi.
Nel precedente articolo *Sviluppare la resilienza* abbiamo parlato delle contromisure da attuare nel ciclo di sviluppo del software per mitigare i rischi più comuni da considerare per l'adesione al paradigma "security by design". Come di solito avviene in contesti tecnologici, l'approccio corretto al problema è stato formalizzato in metodologie divenute ormai standard di fatto per lo sviluppo di software "sicuro".

Il paradigma Dev(Sec)Ops

Schierare le contromisure di sicurezza previste prima, durante e dopo lo sviluppo tenendo conto del principio della *segregazione dei compiti* descritto in precedenza significa predisporre una pluralità di attori (e quindi di team IT) addetti alla governance di ciascuna funzione specifica: un approccio organizzativo in grado di garantire una ottima *security posture*, ma anche un inevitabile aggravio di tempi e costi.

Continuous Integration (CI)

A partire dai primi anni 2000, la necessità di ottimizzare gli aspetti legati a *source control management*, *testing*, *logging* e *monitoring* durante lo sviluppo software ha portato alla definizione della pratica nota come *Continuous Integration* (CI), traducibile come *integrazione continua*, la quale prevede

una serie di test automatici effettuati ogni volta che il codice sorgente viene aggiornato allo scopo di verificare che tale integrazione non abbia introdotto errori nel software esistente. L'obiettivo principale della CI è quello di individuare e risolvere automaticamente i problemi di integrazione nella fase preliminare del ciclo di sviluppo, minimizzando il rischio di bug e conflitti nel codice senza determinare impatti significativi sui costi.

Continuous Deployment (CD)

Negli anni successivi la CI è stata estesa includendo un ulteriore set di automatismi volti a prevedere il rilascio automatico del codice aggiornato in produzione, a patto che le modifiche abbiano superato tutti i test di integrazione previsti. Questa pratica complementare, nota come *Continuous Deployment*, viene oggi considerata a tutti gli effetti parte integrante della prima, all'interno del binomio noto come CI/CD.

La necessità di dotarsi di questo tipo di automatismi è alla base del paradigma *DevOps*, termine formato dalla crasi delle parole *development* e *operations*, che a loro volta identificano le due macro-categorie a cui appartengono i team di professionisti che tradizionalmente formano il reparto IT.

La metodologia DevOps, illustrata per la prima volta nel 2009 da John Allspaw e Paul Hammond, ha avuto un impatto dirompente e rivoluzionario nell'organizzazione del lavoro dei moderni team di sviluppo software proprio perché consente di adottare le principali contromisure proprie dell'approccio *Security by Design* senza rinunciare alla *separazione dei compiti* dei componenti dello staff IT, il tutto tenendo i costi sotto controllo.

Shift-Left: da DevOps a DevSecOps

Gli aspetti prettamente connessi alla sicurezza informatica non erano prioritari in un recente passato, quando le prestazioni e la stabilità del software erano gli indicatori preferiti per la qualità del software, ma hanno assunto un ruolo preponderante negli ultimi anni, a seguito dell'aumento esponenziale delle minacce informatiche, con la transizione, terminologica e fattuale, da *DevOps*a *DevSecOps*. Il nuovo paradigma estende il contesto originario focalizzando l'attenzione su tutte le *best practices* di sicurezza descritte in questo articolo come un requisito fondamentale dell'intero ciclo di sviluppo del software.

Questo concetto, denominato *Shift Left* (spostamento a sinistra, immaginando il ciclo di vita come una linea lungo la quale le fasi si susseguono da sinistra a destra), è a tutti gli effetti un'applicazione concreta del paradigma *Security by Design*.

Conclusioni

Risulta evidente come l'adozione delle *best practices* necessarie per la realizzazione di progetti "security by design", comporti inevitabilmente un incremento di costi. Tuttavia, se consideriamo l'entità crescente dei danni economici e di immagine prodotti dai *data breach* a livello mondiale, l'aumento di valore intrinseco dei dati stessi e la necessità di aderire a normative di *data protection* e *data security* sempre più rigorose e stringenti, possiamo ragionevolmente sostenere che si tratta di un investimento che vale la pena sostenere, soprattutto per le organizzazioni che gestiscono dati e informazioni ad alto rischio.

Come sempre, ferma restando la necessità di rispettare i principi di progettazione sicura imposti dalle normative vigenti, la scelta non può prescindere da un'attenta analisi

costi-benefici. Nell’effettuare questa analisi è importante tenere in considerazione la sfida posta dal periodo attuale, contraddistinto da un progressivo livello di digitalizzazione della società e da un aumento esponenziale delle minacce connesse alle nuove tecnologie, e quindi sempre più esigente in termini di sicurezza. Per questo motivo risulta fondamentale dotarsi del know-how e delle tecnologie necessarie per adottare modelli e metodologie in grado di garantire il miglior livello di sicurezza raggiungibile con il proprio budget.

https://www.key4biz.it/sviluppare-la-sicurezza-parte-iv/485251/