4 ESIGENZE ED AMBITI DI APPLICAZIONE

Secondo la fonte Gartner[1], in considerazione delle contromisure adottate negli ultimi anni per il controllo degli accessi alle infrastrutture e la messa in protezione dei dati, oltre il 75% degli attacchi sono stati indirizzati direttamente verso le applicazioni software, causando gravi danni di immagine e pesanti perdite finanziarie. Gli obiettivi degli attacchi sono le vulnerabilità applicative che si celano all'interno di queste implementazioni software. Le vulnerabilità applicative sono quasi sempre presenti poiché, ancora oggi, le politiche di qualità del software ed i relativi investimenti si concentrano soprattutto sulla correzione puntuale delle difettosità funzionali e sulle performance delle logiche applicative, trascurando l'attuazione di pratiche di progettazione e di sviluppo del software capaci di garantire un adeguato livello di sicurezza.
A titolo esplicativo, la matrice di seguito illustrata riporta un indice quantitativo di effort in termini di costi necessario nella risoluzione delle problematiche di sicurezza applicativa:

		PROBLEMATICHE DI SICUREZZA (VULNERABILITA')				
		Scoperte in fase di progettazione e disegno	Scoperte in fase di sviluppo	Scoperte in fase di integrazione	Scoperte in fase di collaudo	Scoperte in fase di esercizio
DOVUTE A	Architetturali	1X	5X	10X	20X	30X
	Di codifica		1X	10X	20X	30X
	Di integrazione			1X	10X	15X

Tabella 4 - Vulnerabilità dovute a errori

Il National Institute of Standards and Technology (NIST[2]) ha stimato che il costo del "code fixing" eseguito successivamente al rilascio in produzione del codice, può risultare 30 volte il costo che si avrebbe se tali difettosità fossero individuate e risolte nella fase di progettazione.
Il numero degli attacchi continua a crescere in maniera esponenziale e crescono anche le vulnerabilità considerate alte e critiche (fonte: "Rapporto 2019 sulla Sicurezza ICT"). Da qui la necessità di una maggiore diffusione delle fondamentali best practices in materia di sicurezza applicativa, le prime tra tutte riconducibili ad una buona ingegnerizzazione del software, una piena comprensione delle vulnerabilità e delle minacce maggiormente note, compresi i difetti propri dei linguaggi di programmazione, ma soprattutto una considerazione della problematica fin dalle prime fasi del ciclo di sviluppo del software.
Diversamente, i costi aggiuntivi possono portare ad una perdita significativa di produttività e di fiducia da parte degli utilizzatori finali. L'adozione di un ciclo di vita sicuro del software (SSDLC) - ***Allegato 1: Linee Guida per l'adozione di un ciclo di sviluppo di software sicuro*** - aiuta sistematicamente a considerare ed implementare le opportune pratiche/metodologie di sicurezza nel corso di tutte le sue fasi: analisi, progettazione, sviluppo, test e manutenzione, assicurando che le vulnerabilità siano più facilmente individuabili e misurate prima della distribuzione dell'applicazione, riducendo così il costo complessivo dello sviluppo del software.
Esistono diversi modelli e metodologie di ciclo di vita di sviluppo del software, ma ciascuna di queste in generale consiste di una serie di step o fasi predefinite. Indipendentemente dal modello SDLC scelto, è necessario integrare il concetto di sicurezza al fine di garantire una adeguata protezione del sistema e delle informazioni che questo dovrà trasmettere, elaborare e memorizzare.

1 https://www.gartner.com

2 https://www.nist.gov/