

Sicurezza nel ciclo di vita del SW

H1 Introduzione

H2 Progettare un SW che sia al contempo **sicuro** e **funzionale** è spesso complicato. È necessario fare delle scelte su cosa voler mettere in sicurezza. Queste scelte vengono fatte nella fase detta **analisi di sicurezza** (o **analisi del rischio**).

Il modello che *prevede l'analisi del rischio al suo interno* è il **modello a spirale**, che risulta quindi molto utilizzato nel caso di voler creare SW sicuro.

H3 Definizioni preliminari

Politica (*policy*)

H6 Essa è l'insieme di regole che stabiliscono *cosa è consentito e cosa non è consentito*, cioè la **security** del sistema

Spesso è definite in linguaggio naturale.

Meccanismo di sicurezza

H6 È il metodo, strumento o procedura usato/a per *applicare la politica*

H6 È importante notare come **gran parte della complessità non sta nel definire singole politiche e meccanismi di sicurezza, bensì nella loro integrazione nel sistema globale, in quanto ci possono essere conflitti e discrepanze.**

Sicurezza nel ciclo di vita

H3 Sicurezza e requisiti

H2

I **requisiti di sicurezza** fanno parte dei **requisiti non funzionali**. Questa categorizzazione non è molto definita, in quanto spesso i meccanismi di sicurezza influenzano le funzionalità.

Essi hanno lo scopo di stabilire:

- ciò che va protetto
- da chi va protetto
- per quanto tempo va protetto

Il documento dei requisiti deve specificare *cosa* un sistema deve fare, ma anche *perché* deve farlo:

Esempio:

"I numeri delle carte di credito vanno protette contro potenziali furti perché sono informazioni delicate"

H3 Analisi dei rischi

Essa ha l'obiettivo di *identificare i rischi possibili e valutare strategie* per prevenirli e affrontarli

I rischi vanno classificati in base alla loro *severità*, e tale classificazione sarà relativa al contesto.

Livello di protezione

Il **livello di protezione** è funzione della probabilità che un attacco si verifichi e degli effetti dell'attacco qualora succeda

H6

In altre parole, il livello di protezione dipende sia dalla probabilità che l'attacco si può verificare sia dalla gravità del danno che l'attacco potrebbe causare. Quest'ultima influisce maggiormente sul livello di protezione, in quanto spesso si preferisce proteggersi da un attacco improbabile ma grave piuttosto che da un attacco probabile ma che non comporterebbe gravi conseguenze.

H3 Sicurezza e specifica

La specifica deve contenere *possibili soluzioni* ai rischi individuati durante l'analisi. È utile avere una specifica:

- **dettagliata**, che descriva la reazione del sistema in situazioni critiche
- **formale**, completa e non ambigua, ma anche chiara e comprensibile
- **eseguibile**, per avere un feedback immediato

Inoltre, se durante la fase di specifica vengono individuati nuovi rischi, è giusto aggiungerli nei requisiti.

H3 Sicurezza e design

Nella fase di progettazione, dal punto di vista della sicurezza è importante identificare:

- *come i dati passano da una componente all'altra* per garantirne l'**integrità**
- *utenti, ruoli e diritti* sulla base delle politiche di **confidenzialità** e **privacy**
- **ogni soluzione** applicabile ai rischi individuati
- la *relazione di **trust*** tra ciascuna componente

Oltre al *Design Document*, si produce il **security plan** che contiene tali informazioni.

H3 Sicurezza e implementazione

In fase di implementazione è necessario introdurre delle accortezze poiché molti attacchi si basano su errori introdotti in fase di scrittura del codice. È quindi importante:

- applicare, se possibile, *metodi formali* per il testing
- predisporre tecniche per il controllo del codice (*code auditing*)

- scegliere linguaggi di programmazione che non siano inclini a problemi di sicurezza

H3 Sicurezza e testing

Nel momento in cui introduciamo la sicurezza in questa fase, avremo la seguente distinzione:

- **testing funzionale**

Mettere alla prova il sistema per determinare se esso fa ciò che si suppone in circostanze normali o critiche

- **testing di sicurezza**

Mettere alla prova il sistema allo stesso modo di un utente malizioso, sfruttando i punti di debolezza del sistema

Code coverage

Metodo che verifica quanta parte del codice viene eseguita dopo una certa esecuzione, quindi per un certo input

H6

Questo meccanismo è importante sia per il testing funzionale, sia per il testing di sicurezza in quanto il codice non eseguito potrebbe contenere debolezze.

Ciclo di vulnerabilità

Si tratta della sequenza *ciclica* di eventi che partono dalla *identificazione* di una **vulnerabilità** del sistema alla sua *eliminazione*

H2

Esso può avvenire in qualsiasi fase dello sviluppo, ma solitamente avviene dopo la consegna.

La sequenza di eventi è la seguente:

1. Viene scoperta una **nuova vulnerabilità** nel SW
2. Si cerca una **soluzione** al problema:
si *analizza* la vulnerabilità, si *sviluppa* una soluzione e la si *testa* in ambiente controllato, si *distribuisce* la soluzione
3. La **patch** distribuita viene *installata*
4. Si cercano vulnerabilità simili
5. Si ricomincia dal passo 1