SQL Injection Attack Simulation

ABSTRACT.

Questo è un piccolo progetto ha l’obiettivo di simulare un attacco SQL Injection allo scopo di violare la CIA Triad.

## Introduzione

Per questo progetto è stata realizzata una simulazione di un **attacco SQL Injection** di tipo **in-band**, ovvero dove i dati estratti o l’effetto dell’attacco ritornano all’attaccante attraverso lo stesso canale da cui è partita la richiesta.

## 2.0 Project Structure

L’ambiente è stato costruito utilizzando **Docker**, con un **web server PHP** con Apache e un **database MySQL** vulnerabile, accessibile tramite una semplice **interfaccia di login web** creata ad hoc non protetta contro SQL Injection.

L’applicazione è quindi composta da due container:

* **php\_web:** contiene il web server + l’interfaccia web;
* **vulnerable\_db:** contiene il database MySQL con la tabella users.

## 3.0 Functionalities

* All’avvio viene eseguito il file **init.sql**, il quale crea la tabella users e la popola con dati iniziali composti da username e password.
* Viene mostrata all’utente una semplice interfaccia di login (**index.php**), con un form per inserire username e password per l’accesso. In questi campi è possibile inserire il codice SQL per far partire l’attacco.
* Una volta “eseguito” il login si accede ad una seconda interfaccia (result.php), la quale:
  + Mostra la **query** che viene eseguita;
  + Mostra l’**esito** del login (successo/fallimento);
  + Elenca i **dati ottenuti** dalla query eseguita (un semplice login di un utente legittimo restituirà una scritta di benvenuto con lo username e la password inseriti);
  + Visualizza lo **stato attuale del database** (la tabella users), al fine didattico di mostrare le modifiche effettuate.

## 4.0 Attack Simulation

Al fine di rendere il database vulnerabile a SQLi, sono stati presi due accorgimenti principali:

1. Il login viene gestito tramite query non parametrizzate
2. Le query vengono tutte gestite come multi query:

Andiamo ora a vedere nello specifico gli attacchi che sono stati simulati:

### 4.1 Violazione della Confidentiality