Attacco SQL Injection

Introduzione

Questo progetto ha come obiettivo dimostrare un attacco di SQL Injection di tipo in-band, utilizzando input forniti dall'utente per iniettare comandi SQL malevoli. L'attacco è volto a compromettere almeno due delle proprietà della CIA (Confidentiality, Integrity, Availability) di un sistema di gestione database vulnerabile. Abbiamo implementato il progetto su un ambiente Docker con una configurazione vulnerabile di un server web e database.

Ambiente di Sviluppo

Piattaforma: MacOS con processore M1

Tecnologie utilizzate:

- **Docker** per la virtualizzazione degli ambienti di sviluppo.
- **MARIAdb** come sistema di gestione di database relazionale.
- **PHP** per la realizzazione dell'applicazione web vulnerabile.
- **phpMyAdmin** per la gestione del database tramite interfaccia web.

Esecuzione dell'Attacco SQL Injection

Attacco Tautologico

Inserire nel campo username: `admin' OR '1'='1`

Lasciare il campo password vuoto o inserire un valore qualsiasi.

Attacco con Commento di Fine Riga

Inserire nel campo username: `admin'; -`

Lasciare il campo password vuoto o inserire un valore qualsiasi.

Attacco Piggybacked

Inserire nel campo username: `admin'; DROP TABLE users; --`

Lasciare il campo password vuoto o inserire un valore qualsiasi

Risultati

Durante l'esecuzione dei test, gli attacchi SQL Injection di tipo Tautologico, Commento di Fine Riga e Piggybacked hanno funzionato correttamente, permettendo l'accesso non autorizzato al sistema. In particolare, l'attacco Piggybacked ha avuto successo nel cancellare la tabella `users`, dimostrando la vulnerabilità del sistema.

Conclusioni

Il progetto ha dimostrato con successo l'esecuzione di attacchi SQL Injection di tipo Tautologico, Commento di Fine Riga e Piggybacked, compromettendo la proprietà di integrità e disponibilità del sistema. Questo sottolinea l'importanza di implementare misure di sicurezza adeguate, come l'uso di query preparate e la sanitizzazione dell'input degli utenti, per prevenire tali attacchi.