Guida della vulnerabilità SQL Injection

Introduzione

Questa è la guida della vulnerabilità di tipo **SQL Injection**, seguendo questa guida riuscirai a sfruttare la vulnerabilità all'interno di HackerLab.

Questo tipo di vulnerabilità permette ad un malintenzionato di poter eseguire del codice malevolo sfruttando delle falle nella validazione degli input da parte dell'utente e della costruzione di query al database.

Requisiti

- Browser (Nella guida viene utilizzato Chrome)
  - https://support.google.com/chrome/answer/95346

Guida

Per eseguire questa vulnerabilità è richiesta una conoscenza basilare del linguaggio SQL. Per trovare una falla di questo tipo all'interno di qualsiasi sito web si deve provare manualmente o in modo automatico ad inserire delle query sql o dei caratteri specifici all'interno di tutti i campi che possono essere inviati al server da parte dell'utente. Eseguendo questa procedura all'interno di HackerLab si può notare che inserendo dei caratteri speciali utilizzati da SQL per la creazione di query viene generato un errore, questo avvisa chi sta cercando la falla della sua stessa presenza.
In questo caso ho utilizzato il carattere ' (apice singolo) all'interno della barra di ricerca di HackerLab, il risultato ottenuto è il seguente:

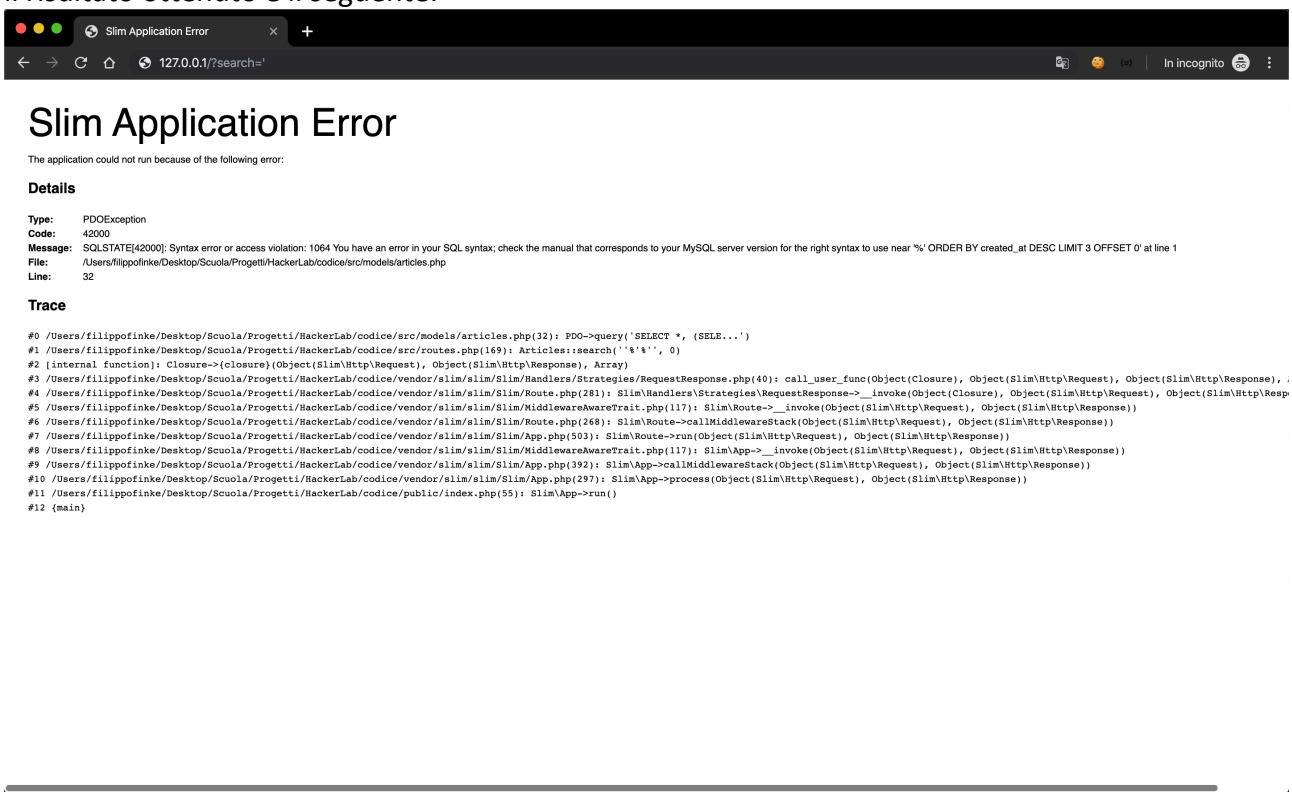

Figura 1 Schermata di errore.

Come possiamo vedere viene mostrato a schermo un errore riguardanti la sintassi della query SQL che l'applicativo utilizza per interrogare il database.
Lo possiamo notare dal messaggio di errore:
SQLSTATE[42000]: Syntax error or access violation: 1064 You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '%' ORDER BY created_at DESC LIMIT 3 OFFSET 0' at line 1
Grazie a questo possiamo anche pensare alla struttura della query, potremmo pensare che la query sia la seguente:
SELECT colonne FROM tabella WHERE colonna LIKE '%VALORE_DI_RICERCA%' ORDER BY created_at DESC LIMIT 3 OFFSET 0.
Possiamo risalire a questa query basandoci sul messaggio di errore che mostra chiaramente il suo formato.
Grazie a questo possiamo confermare la presenza di questa vulnerabilità e considerare il fatto di averla già sfruttata. Se il malintenzionato ha conoscienze più approfondite di SQL può procedere ad eseguire delle query SQL molto più complesse, come per esempio la seguente query:
a%' UNION ALL SELECT '', '', CONCAT(full_name, " ", email), '','','' FROM users;--
Questa query unisce la tabella degli articoli presenti nel sito web con la tabella degli utenti. In questo modo possiamo mostrare nella lista degli articoli anche tutti gli utenti registrati, risultato:

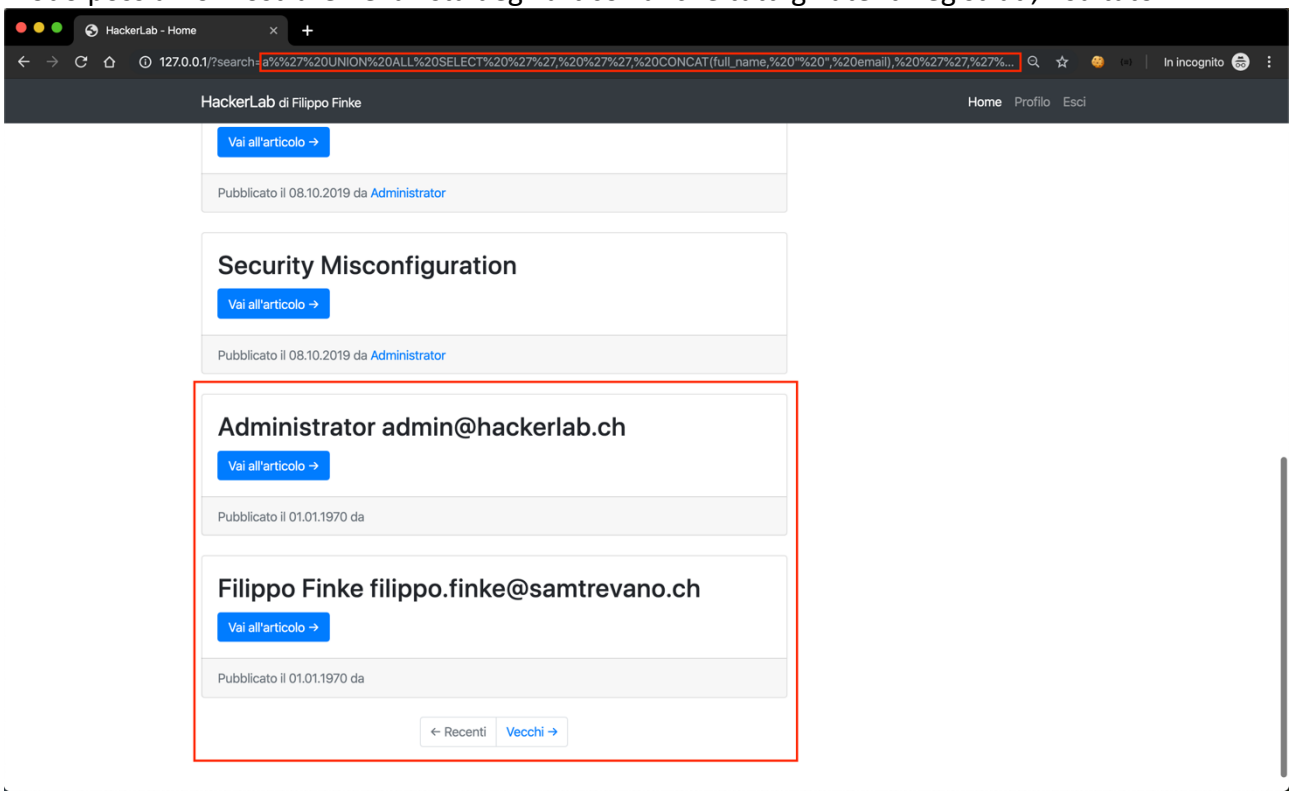

Figura 2 Esecuzione query complessa.

Come si può vedere dal risultato oltre che gli articoli otteniamo anche la lista di utenti presenti nel sito web.
Questa query è quindi molto pericolosa ma anche difficile da sfruttare a pieno, richiede delle competenze elevate e anche molta ricerca sulla struttura del sito web che si vuole attaccare.