Lab 01: SQL Injection Retrieving Hidden Data via WHERE Clause

Lab link:

https://portswigger.net/web-security/sql-injection/lab-retrieve-hidden-data

(Solved instance: https://0ae900ce046113c780504ee300e7006e.web-security-academy.net)

Objective

This lab demonstrates a basic SQL injection vulnerability in the WHERE clause of a product category filter.

The goal is to bypass a released = 1 condition and view unreleased products using SQL injection.

Vulnerable SQL Query

The original query used by the application is likely:

```
SELECT * FROM products WHERE category = 'Gifts' AND released = 1
```

We aim to modify this using a SQL injection payload:

```
category=Gifts'+OR+1=1--
```

Resulting in:

```
SELECT * FROM products WHERE category = 'Gifts' OR 1=1 --' AND released = 1
```

- OR 1=1 is always true, retrieving all rows
- -- comments out the remaining query

Exploitation Steps

Step 1: Accessed the lab and landed on the home page (1-access-lab.png)

Step 2: Clicked on a product category (Gifts) (2-click-category.png)

Step 3: Intercepted the original request with Burp Suite (3-intercepted-request.png)

Step 4: Modified the request with SQL injection payload (4-modified-request.png)

Step 5: Received a response with unreleased products visible (5-response-with-unreleased.png)

Step 6: Confirmed lab completion with the success banner (6-lab-solved.png)

Outcome

The payload successfully bypassed the released = 1 condition, revealing hidden products.

This demonstrates how insecure handling of user input in SQL queries can expose sensitive or internal data.

Notes

- This is Lab 01 of the SQL Injection series
- All screenshots are taken using Burp Suite Community Edition
- Documented in Australian English as part of a personal bug bounty portfolio