

SQL Injection Vulnerability Report

**Lab: SQL Injection – Extracting
Hidden Data**

**Platform: PortSwigger Web Security
Academy**

Tester: Praisegod Aliyu

Date: [06 Jan, 2026]

Objective

The objective of this lab was to identify and exploit a SQL Injection vulnerability that allows extraction of hidden data from the database.

Vulnerability Type

SQL Injection (Boolean-based)

OWASP Category: A03 – Injection

Target Description

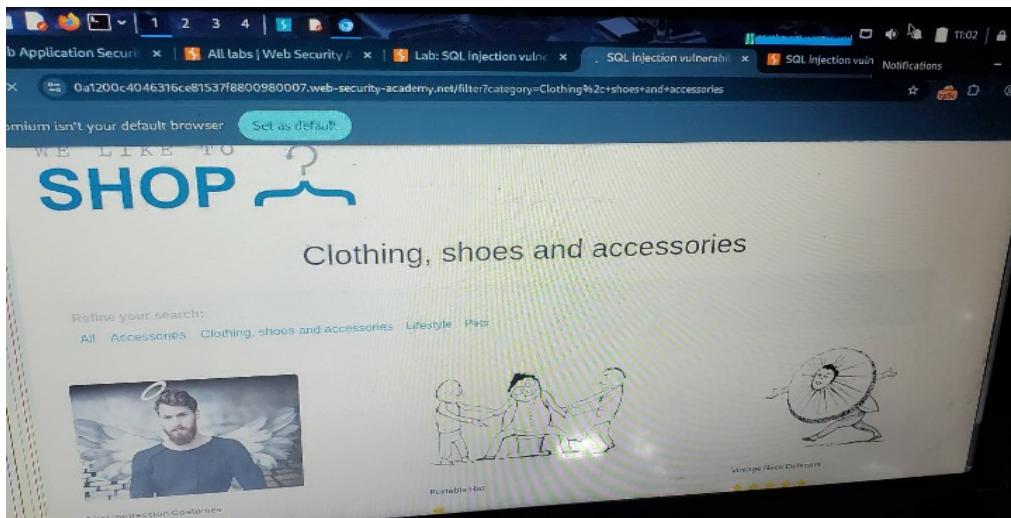
The target application contains a product listing page that uses a SQL query to retrieve products from the backend database based on category input.

Vulnerability Discovery

The category parameter was tested by injecting a single quote (' which resulted in an SQL error, indicating improper input sanitization.

Exploitation Steps

1. Intercepted the request using Burp Suite.
2. Identified the vulnerable 'category' parameter.
3. Injected the following payload: ' OR 1=1--
4. The application returned all products, including hidden items.



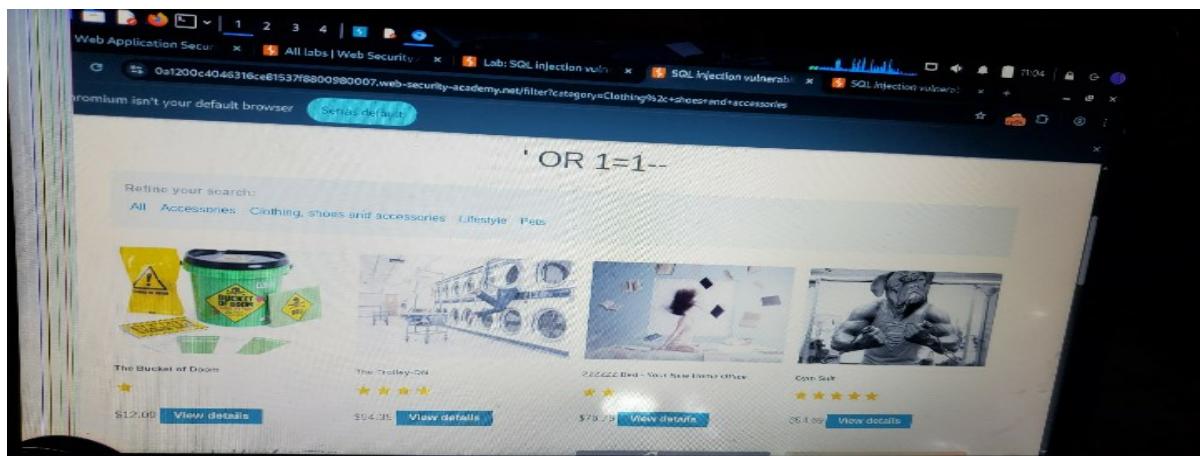
Before injection

Screenshot of Burp Suite showing a network request to https://0a1200c4046316ce8157f8800980007.web-security-academy.net/filter?category=Clothing%20+shoes+and+accessories. The request is a GET method with the URL https://0a1200c4046316ce8157f8800980007.web-security-academy.net/filter?category=Clothing%20+shoes+and+accessories.

The Request pane shows the raw HTTP request:

```
GET /filter?category=Clothing%20+shoes+and+accessories HTTP/2.0
Host: 0a1200c4046316ce8157f8800980007.web-security-academy.net
Cookie: Session=22_AqCkK17p+WBCHo2000L97Wv-kT
User-Agent: Chrome/133.0.6702.199, "NokiA-Brand", v=59
Sec-Cloud-Mobile: 70
Sec-Ch-Ua-Platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.6702.199 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Dnt: 1
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Content-Type: application/x-www-form-urlencoded
```

Injecting the payload



After injection (hidden items visible)

Impact

An attacker could exploit this vulnerability to access unauthorized data, potentially exposing sensitive business information.

Remediation

- Use prepared statements (parameterized queries)
- Implement proper input validation
- Avoid dynamic SQL queries