Network for Security (NFS)
Université de Rennes
2025

PENTEST Web: Lab 3

I Introduction

This lab will introduce SQL Injection attacks.

SQLi: Injects malicious SQL into a backend query, causing the database to execute attacker-controlled commands. This can expose or alter stored data, bypass authentication, or disrupt the application's normal behaviour.

For this lab, I expect a brief PDF report in a write-up style, where each question is addressed with a clear explanation of your approach, including both successes and any encountered difficulties. Screenshots and code snippets should be added when they support your reasoning. Please send the report to gwendal.patat@irisa.fr, using the subject line [PENTEST LAB 3 RSSI], no later than December 14.

I.1 Prerequisites

For this lab, you must install or download the following tools/files:

- docker
- the lab zip file here: https://avalonswanderer.github.io/assets/zip/pentest/Labsetup_sqli.zip or the arm version here.

As the previous lab exercises, we need to deploy containers and configure DNS resolution in order to correctly resolve the site names and simulate a real-world scenario. You can take a look at the previous lab subjects for a step by step setup if you need to.

II SQL Injections

Start the dockers and visite the URL http://www.seed-server.com.

II.1 Authentication Bypass

A login page is often connected to a database on the back end to retreive existing user information and authenticate them.

To help you a bit, here is an idea of what the server side SQL request looks like when searching for the given name/password combinaison.

```sql
SELECT name, eid, salary, email FROM credential
WHERE name='(NAME)' and password='(PASSWORD)';
```

Question 1: Connect as alice using an SQLi.

Question 2: Connect as a user with a high salary (over 90000).

Question 3: With this knowledge, you should be able to flag this challenge pretty easily: SQL Injection Authentication

In the wild, protections can be added to sanitize the input provided by the user before launching the SQL request to avoid such problem.

In the following challenge, sanitization is being used but the title should help you find a way to bypass the authentication in a similar way as before.

Question 4: Exploit such subtleties to bypass the authentication again by flagging this challenge: SQL Injection Authentication GBK

II.2 Database Update

Bypassing an authentication is not the only thing you can do with SQL injections. Depending on the backend request, you can even modify the actual data server-side.

Question 5: Log on with Alice and try to change your salary to whatever you like without changing the other employees' salary.

Question 6: From Alice account, try now to change the salary of Ted to 1.

Once again, to give you an idea of what the regular server side request might looks like:

```
UPDATE credential
SET nickname='...', email='...', adress='...', ...
WHERE EID = 10000;
```

Acknowledgements

This work is inspired by seedlabs and the book *Internet Security: A Hands-on Approach, 3rd Edition,* by Du Wenliang.