**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе №5**

**по дисциплине «Базы данных»**

Тема: Тестирование БД на безопасность

|  |  |  |
| --- | --- | --- |
| Студент гр. 1303 |  | Бутыло Е.А. |
| Преподаватель |  | Заславский М.М. |

Санкт-Петербург

2023

**Цель работы.**

Сделать простой web-сервер и проверить реализованное API на безопасность.

**Текст задания**

Вариант 3

1. Сделать простой web-сервер для выполнения запросов из ЛР3, например с (express.js). Не обязательно делать авторизацию и т.п., хватит одного эндпоинта на каждый запрос, с параметрами запроса как query parameters.
2. Намеренно сделайте несколько (2-3) запроса, подверженных SQL-инъекциям
3. Проверьте Ваше API с помощью sqlmap (или чего-то аналогичного), передав эндпоинты в качестве целей атаки. Посмотрите, какие уязвимости он нашёл (и не нашёл), опишите пути к исправлению.

**Выполнение работы**

Для выполнения работы были использованы модели из лабораторной работы №3. Реализовано API для доступа к БД.

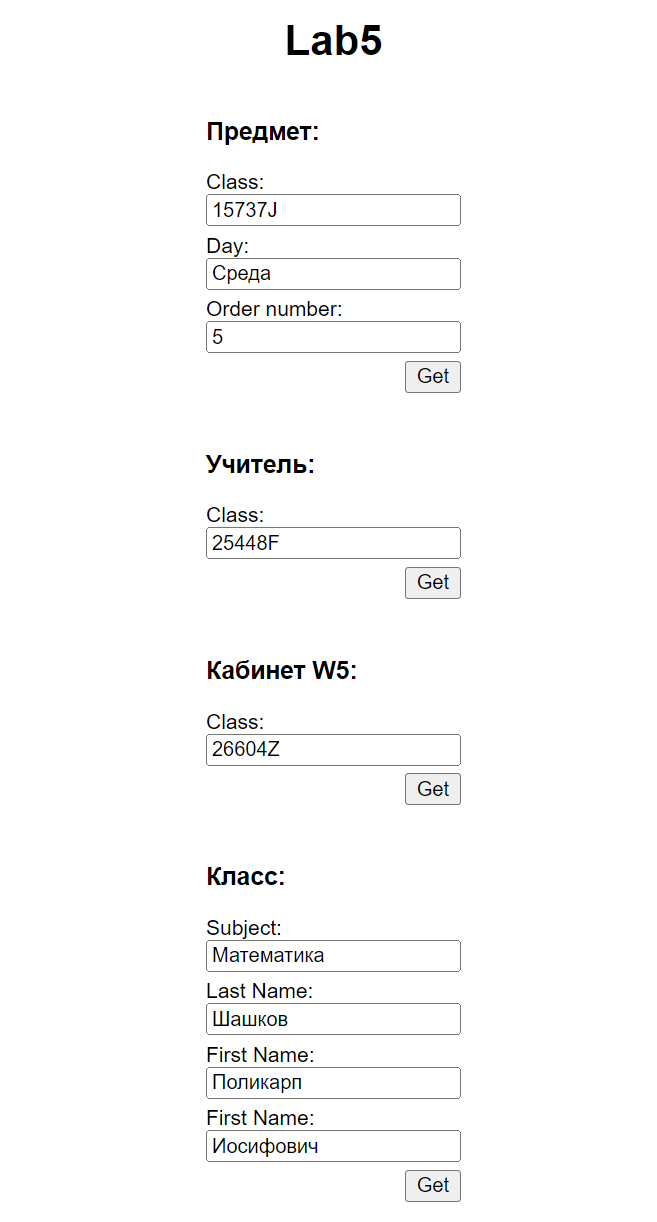


Рисунок 1. – Реализованный web-сервер для выполнения запросов.

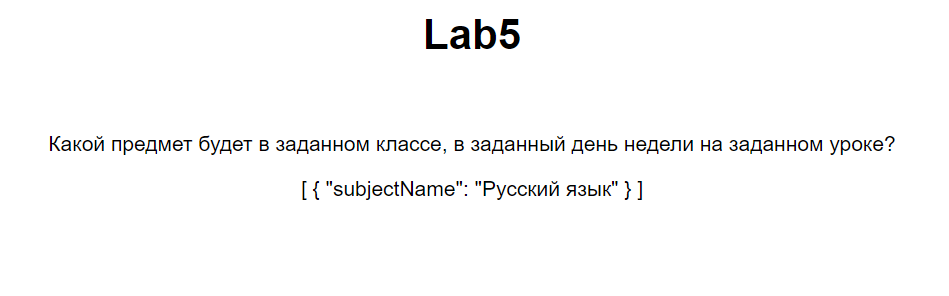
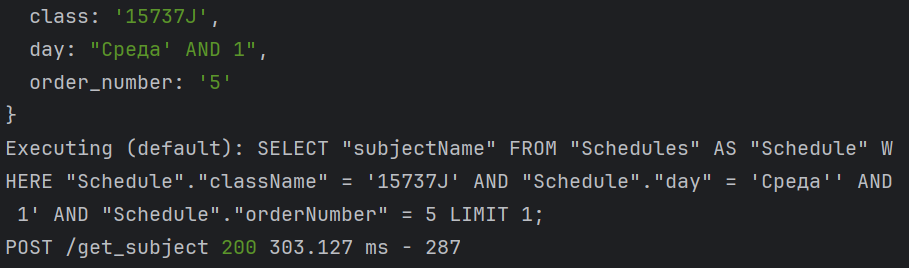
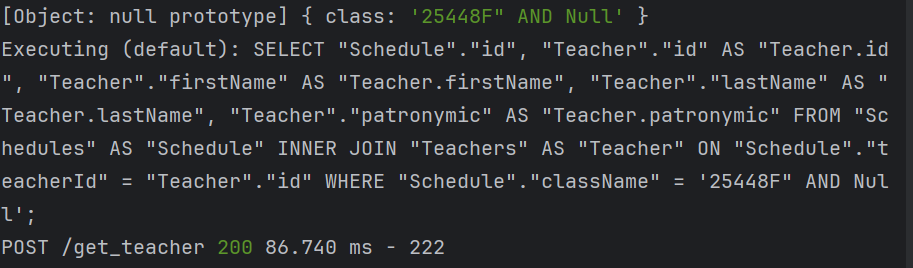


Рисунок 2. – Пример ответа web-сервера.

Намеренно сделано несколько запросов, подверженных SQL-инъекциям:

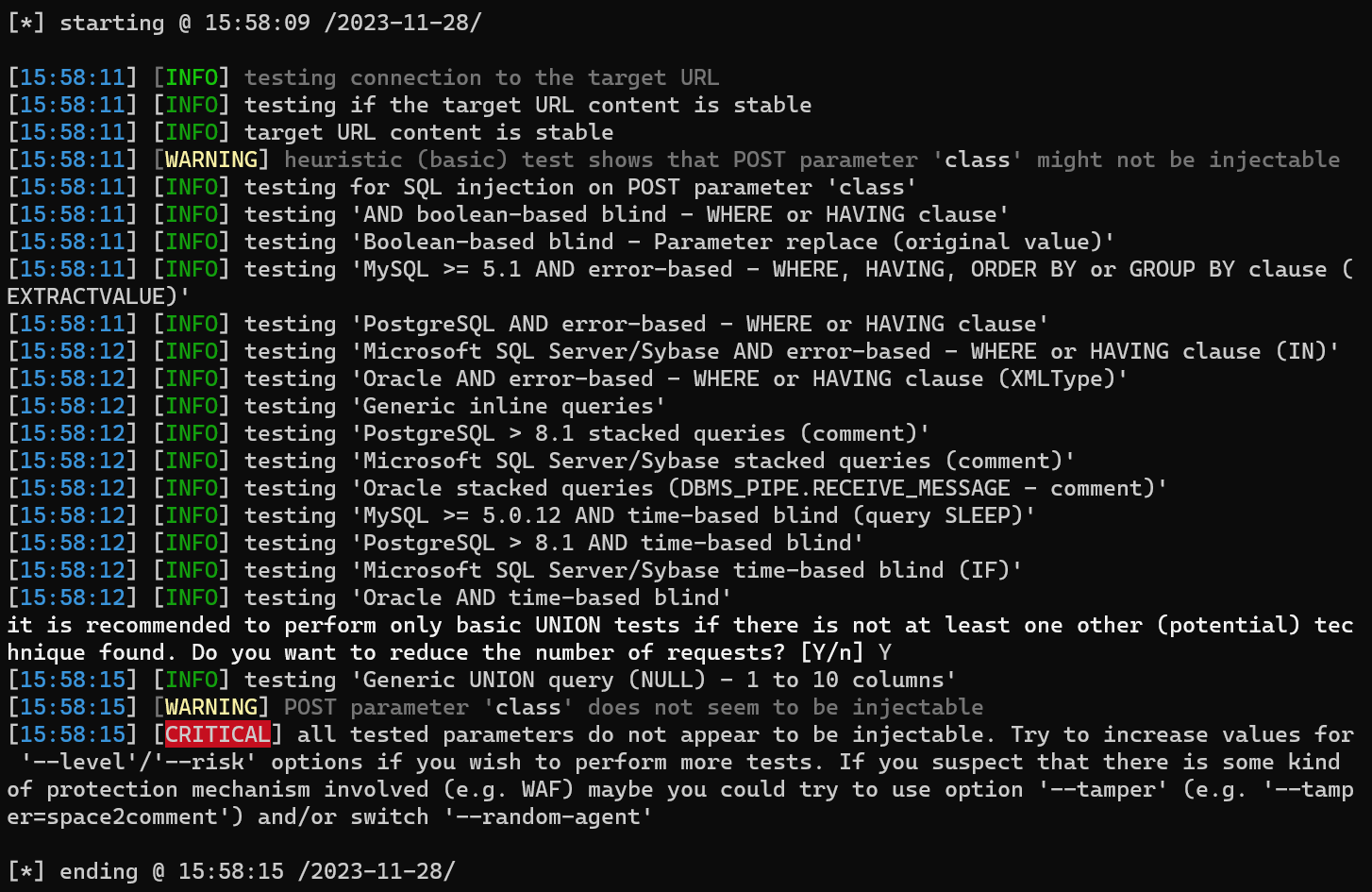




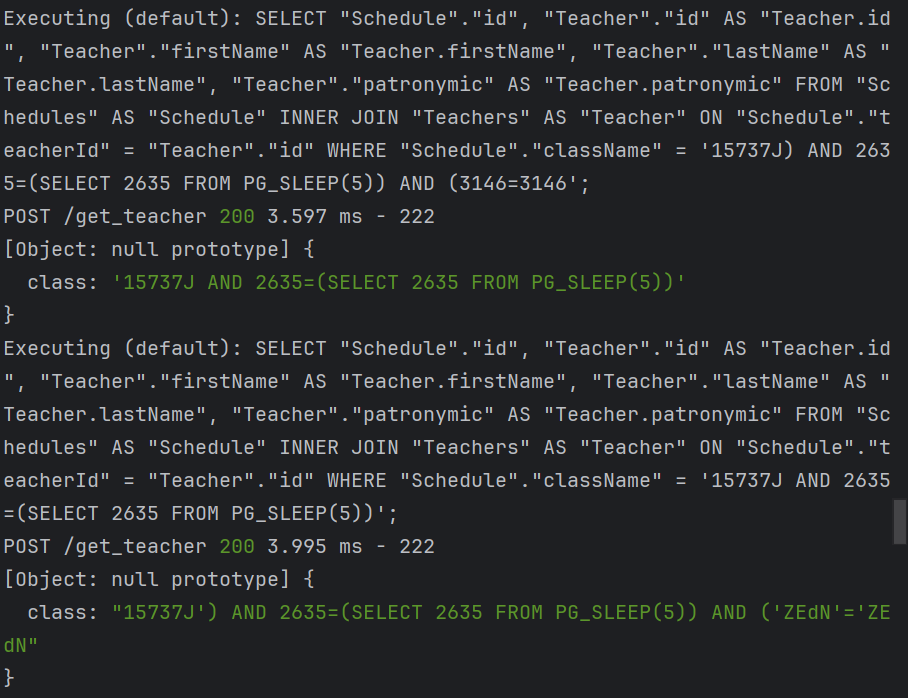
Как видим никаких особых результатов запросы не принесли.

Проверим API с помощью sqlmap:

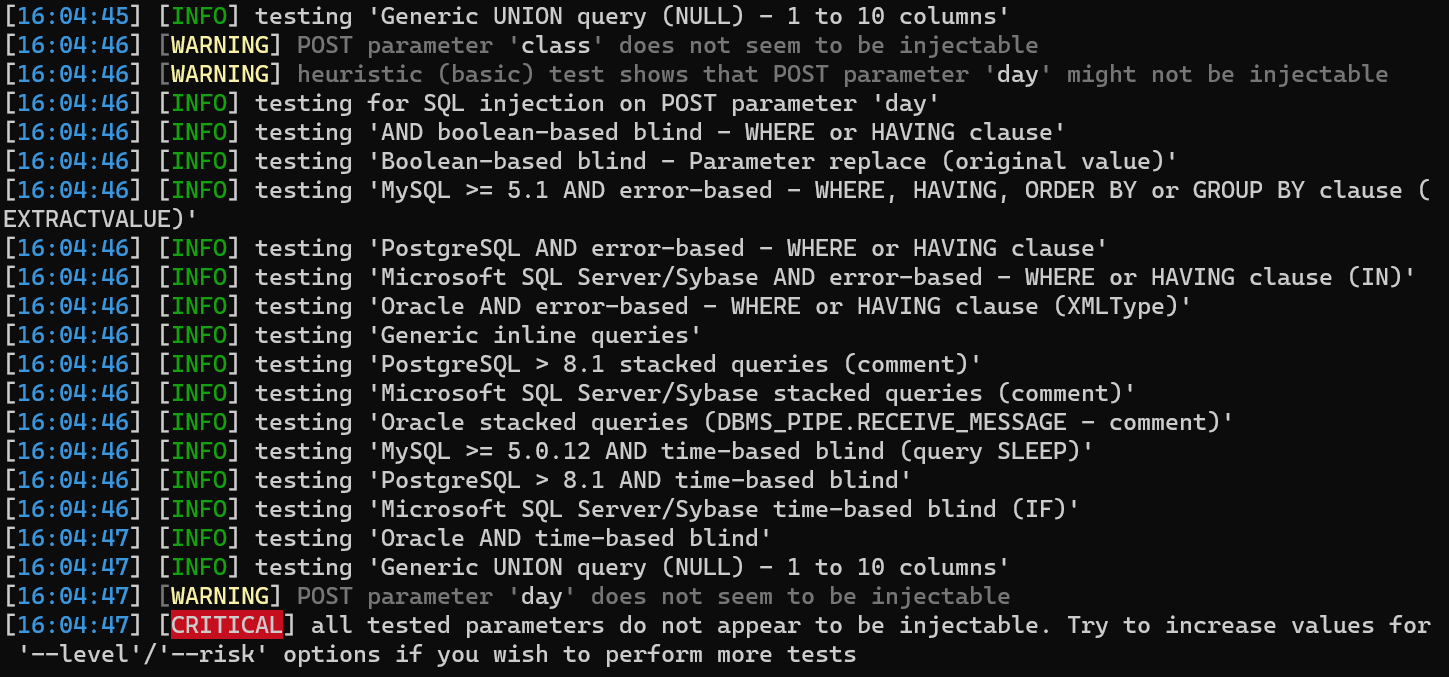
1. python .\sqlmap.py --method POST -u "http://localhost:3000/get\_teacher" --data "class=15737J" -p "class"



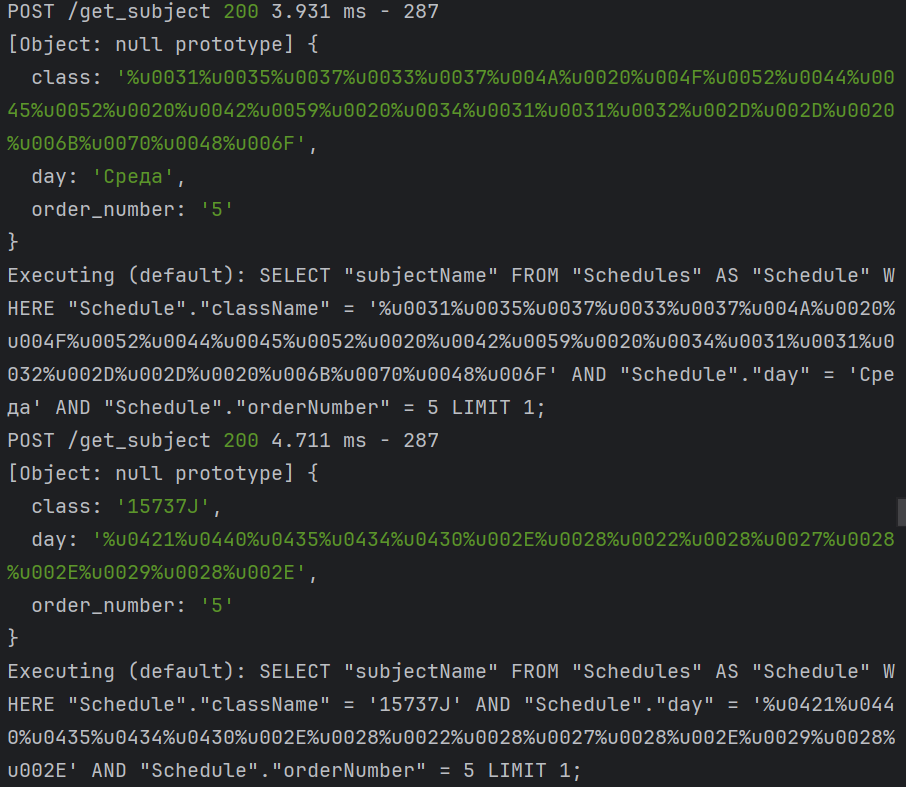
Поля не подвержены инъекции. Далее пример запросов, для проверки угрозы инъекции:



1. python .\sqlmap.py --method POST -u "http://localhost:3000/get\_subject" --data "class=15737J&day=Среда&order\_number=5" -p "class,day" --tamper=charunicodeencode



Поля не подвержены инъекции. Далее пример запросов, для проверки угрозы инъекции:



**Выводы.**

Протестировав доступ к БД с помощью ORM Sequelize на безопасность, получили, что система успешно справляется с угрозами SQL-инъекции.

**ПРИЛОЖЕНИЕ А**

Pull request: <https://github.com/moevm/sql-2023-1303/pull/61>