Министр науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет информационных технологий и программирования

Лабораторная работа №4

Безопасность WEB-приложений

Выполнил студент группы № M34041
Титов Даниил Ярославович

Проверил:
Батоцыренов Павел Андреевич

Санкт-Петербург

2024

Содержание

Код отчёта на Overleaf (Latex)

1 Broken Access Control

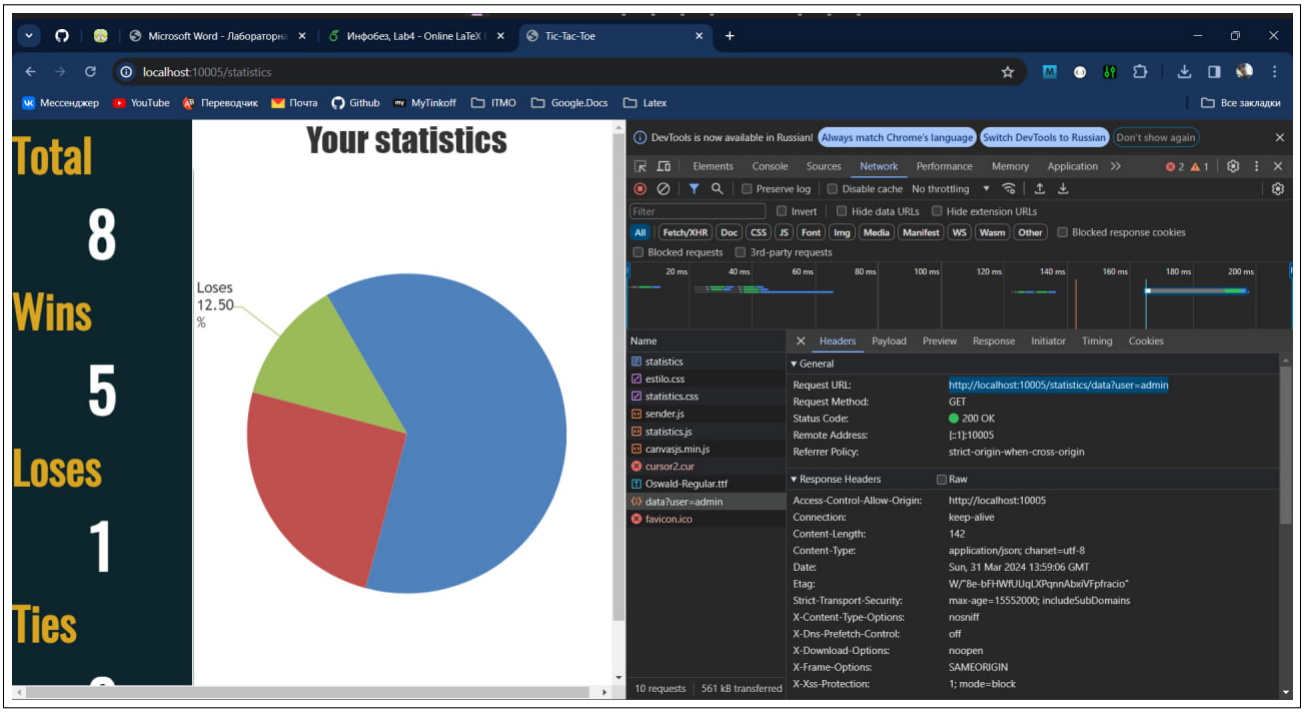
1.1 Создать двух пользователей

Я создал двух пользователей: admin и admin2

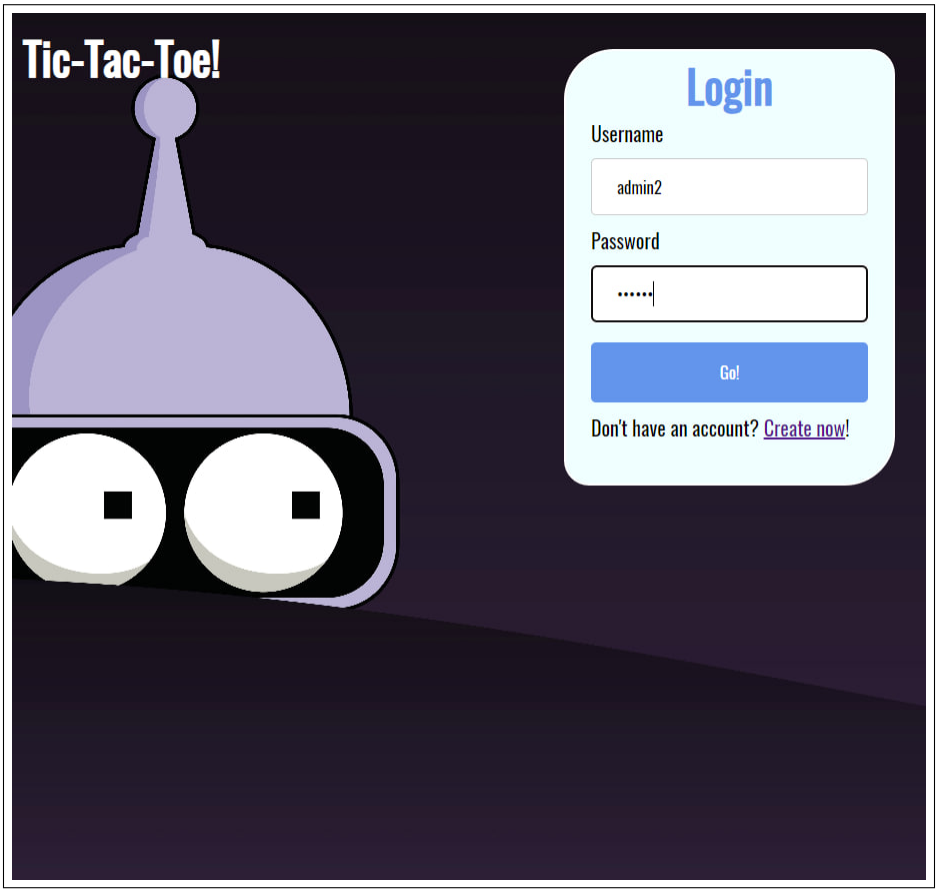
1.2 Сыграть несколько раз первым пользователем и посмотреть его статистику

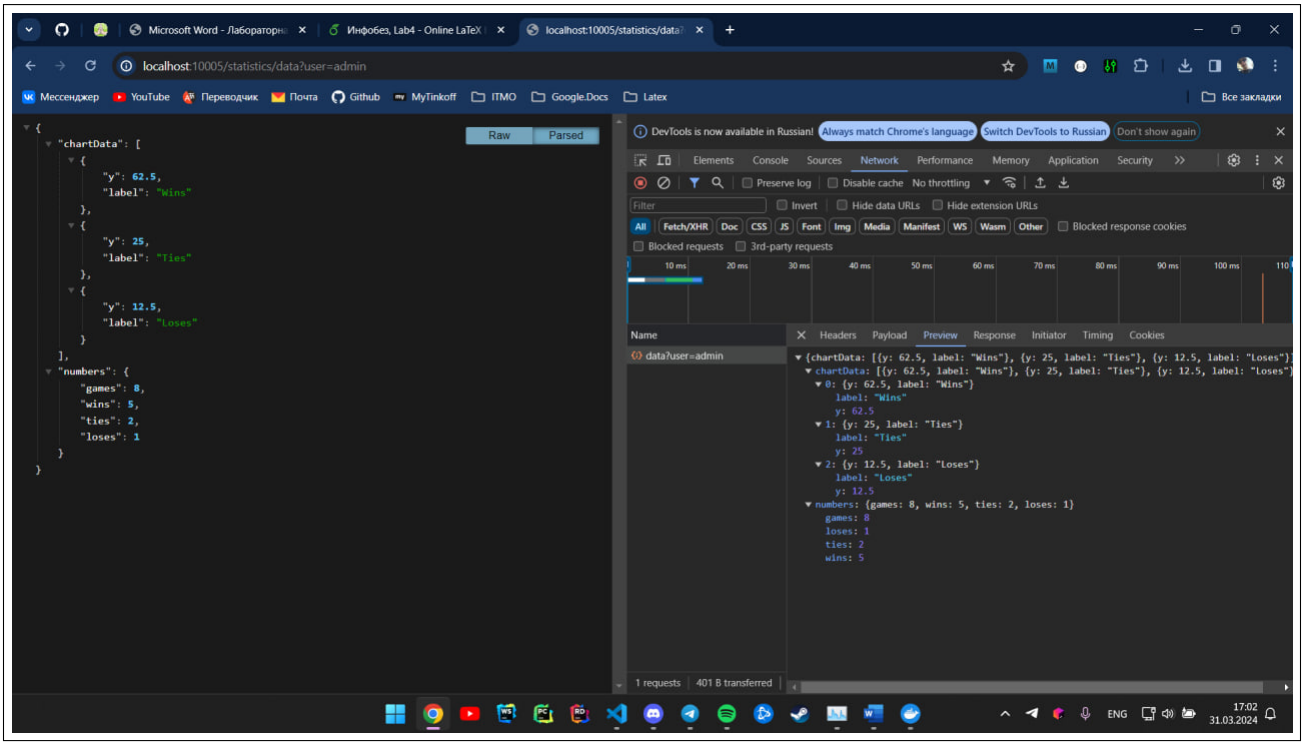
Сыграл 8 игр: 5 - win, 1 - lose, 2 - tie

1.3 Проверить как статистика получается с сервера через Developer Tools



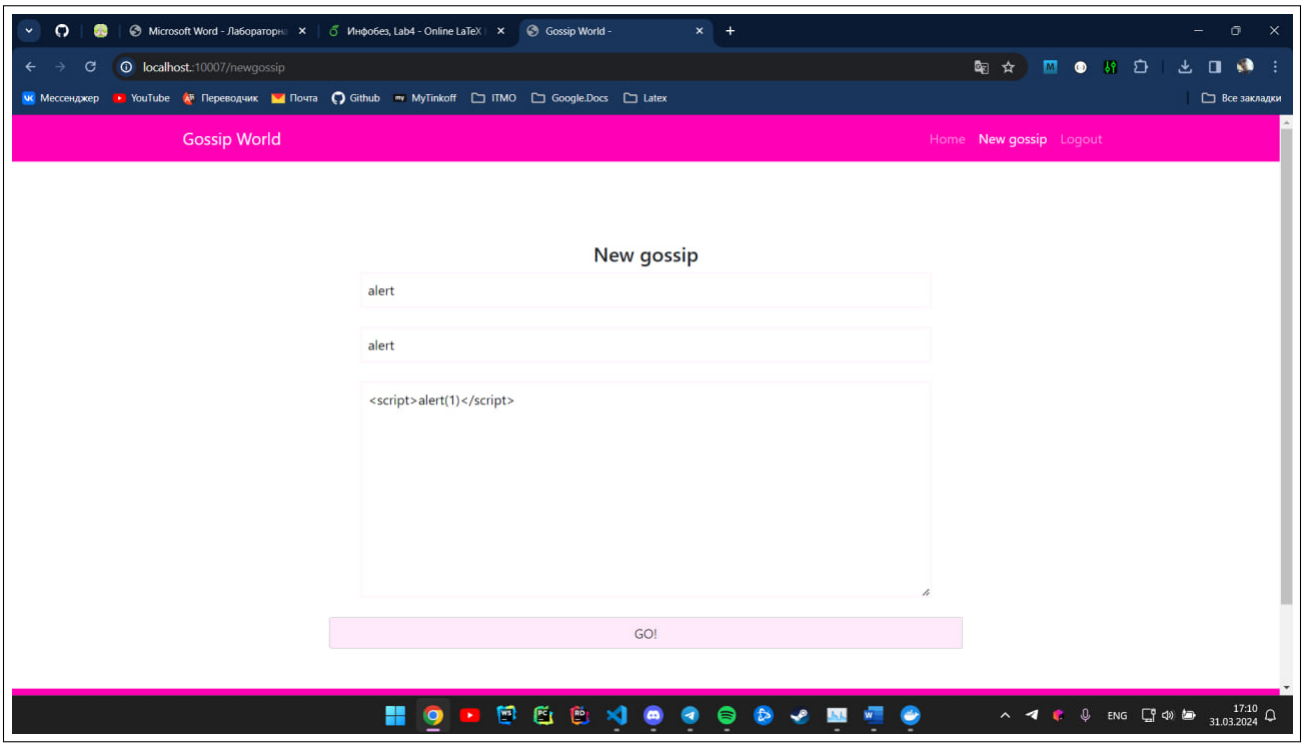
1.4 Используя GET-запрос и параметры сессии второго пользователя, вывести статистику первого пользователя.

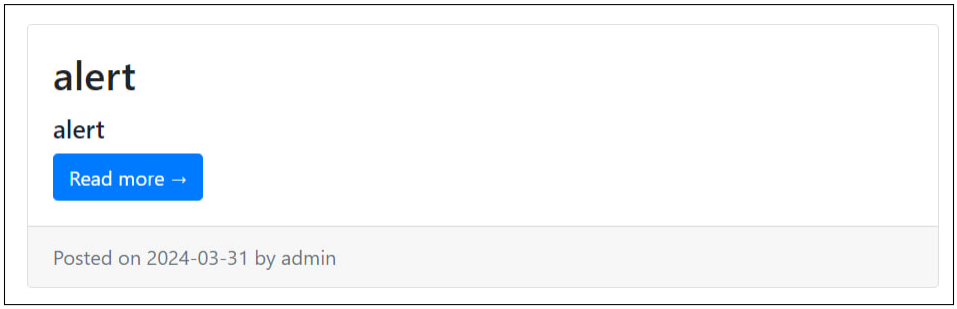


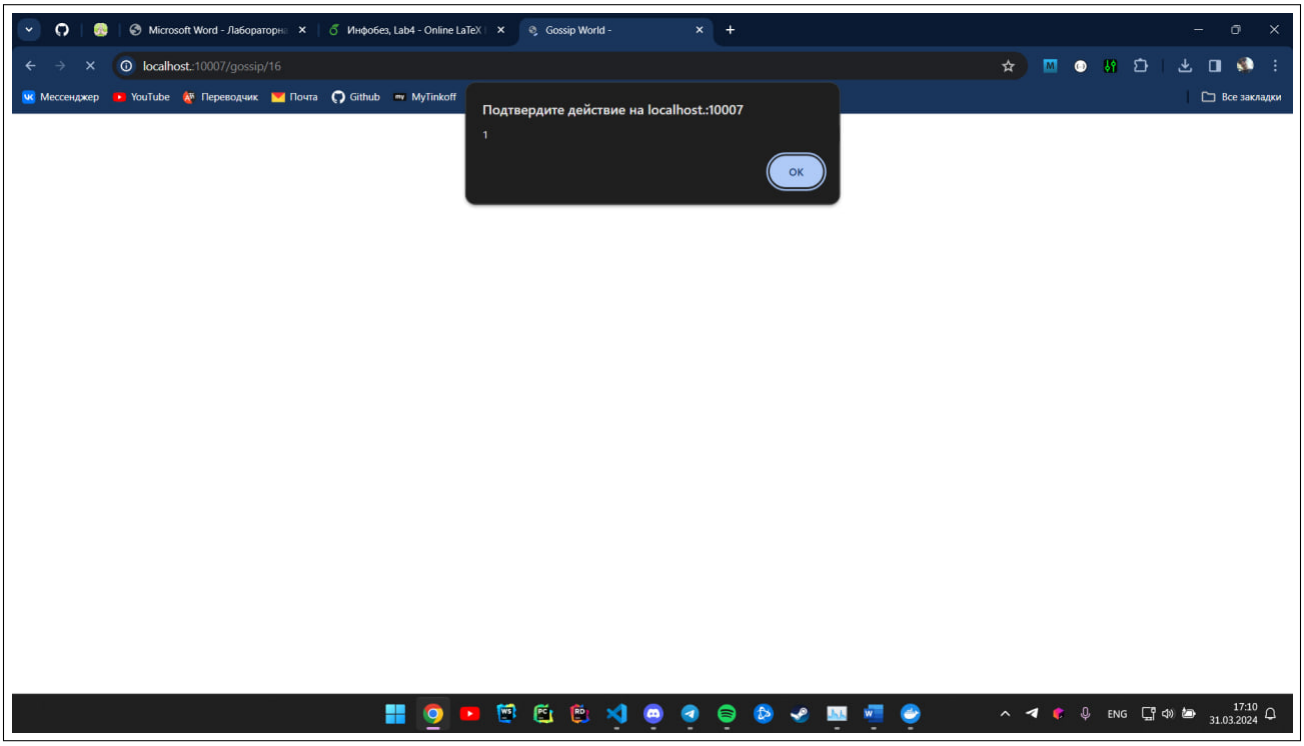


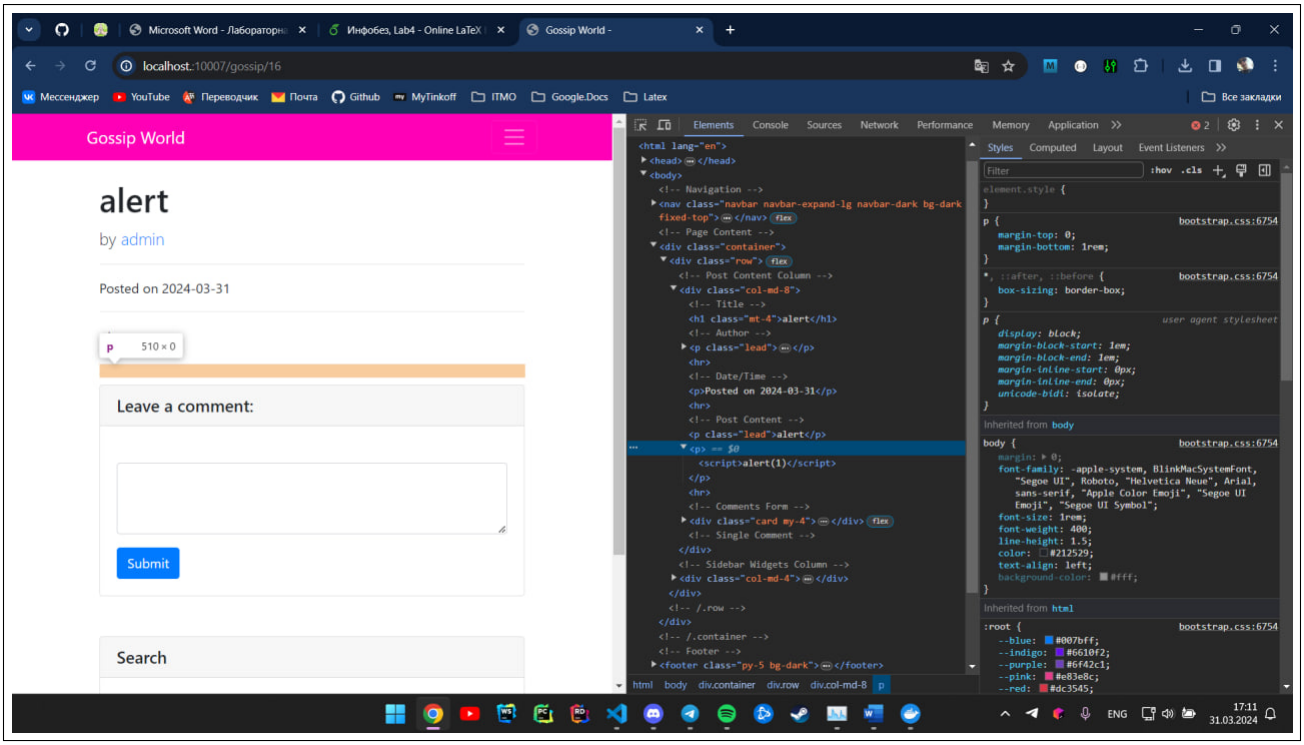
2 Cross-Site Scripting

2.1 Используя <script>alert(1)</script> проверить формы приложения



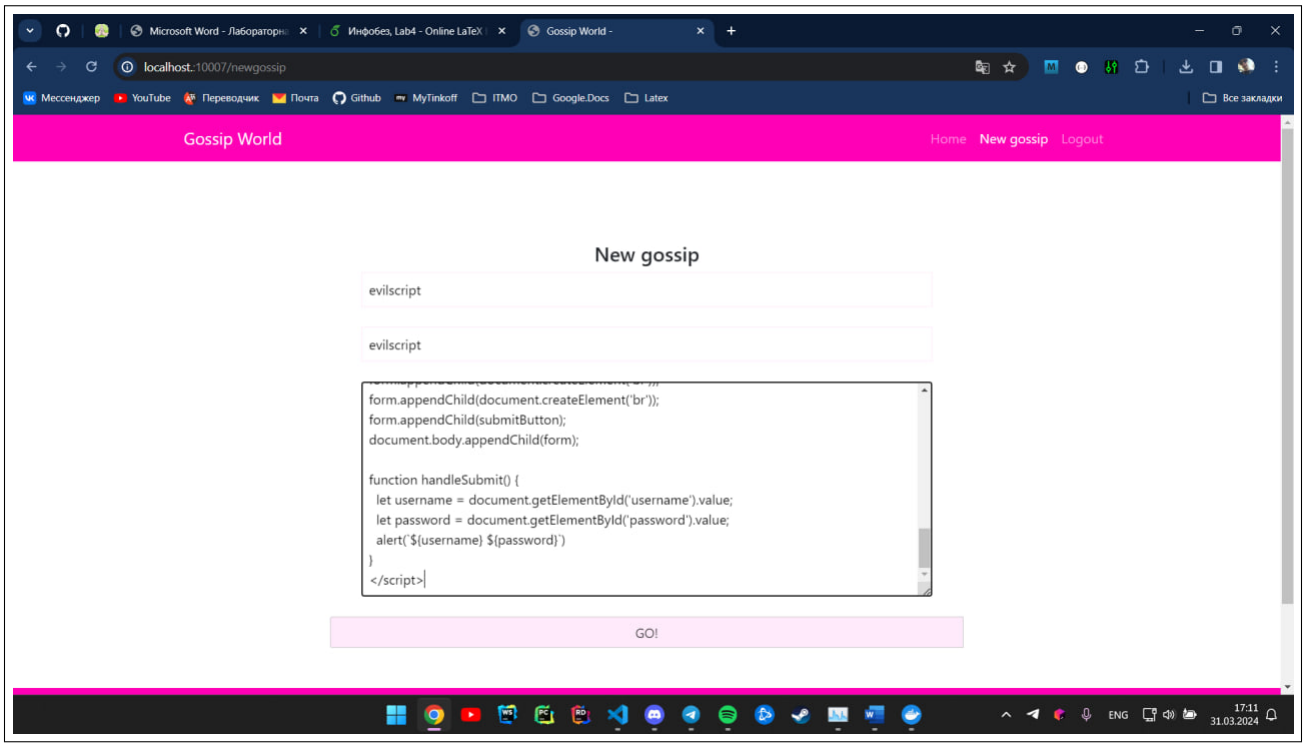


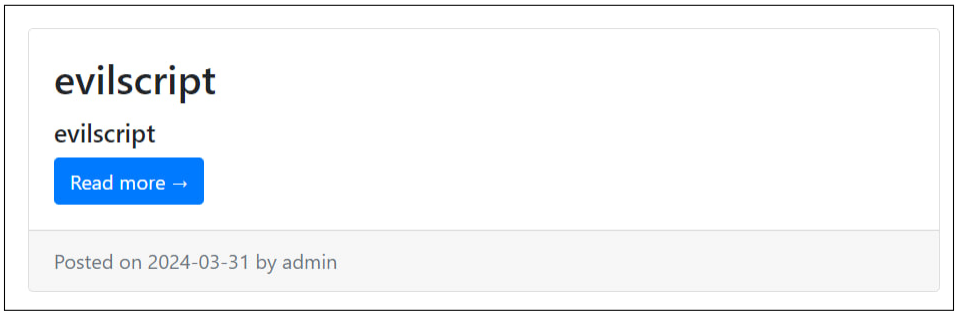

Подтвердите действие на localhost.:10007
1
OK

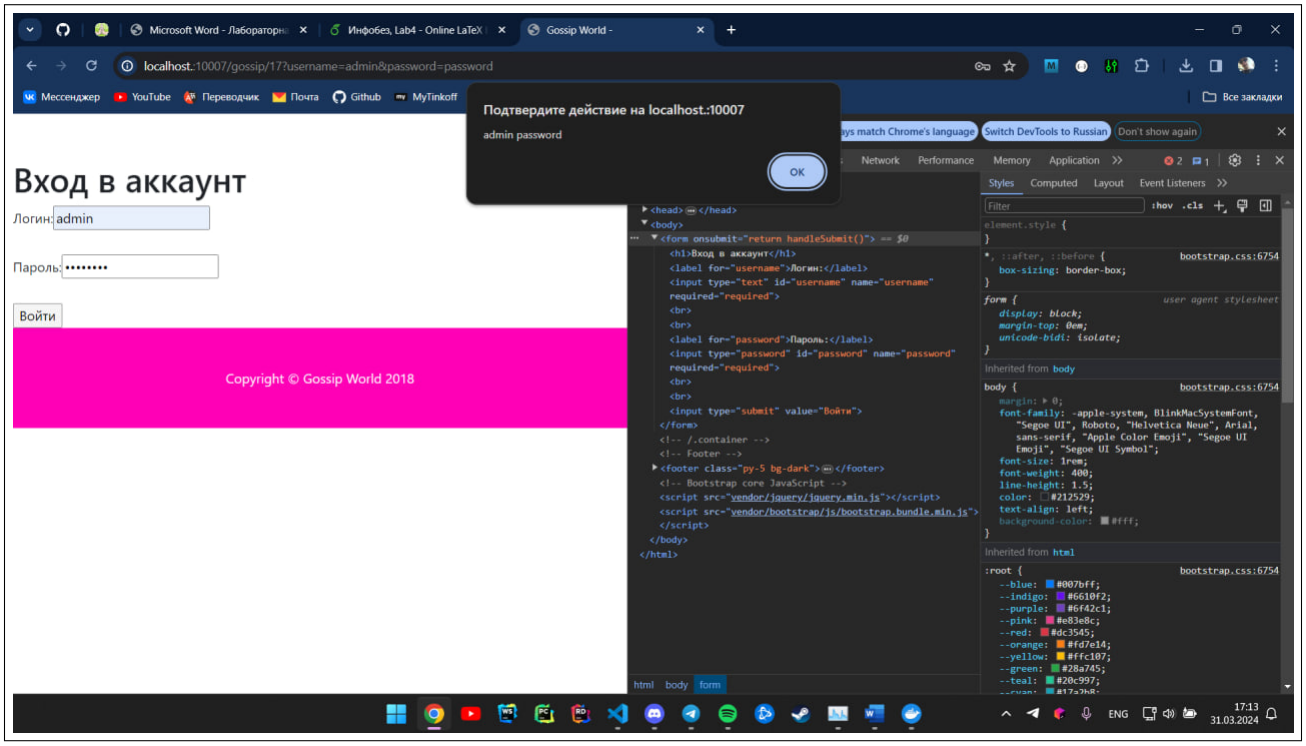

Gossip World
alert
by admin
Posted on 2024-03-31
Leave a comment:
Submit
Search

2.2 Как можно развить атаку?

Развить атаку можно многими способами, но я выбрал подмену страницы с постом на совершенно другую. В моём случае я просто сделал одну из самых простых формочек с логином и паролем. Условно считаем, что пользователь вводит туда свои реальные логин и пароль.
Далее у меня просто выводятся эти данные через alert, дабы не усложнять лабораторную работу, но очевидно, что можно было бы отправлять эти данные куда угодно для дальнейшего переиспользования и взлома попавшегося на эту уязвимость пользователя.





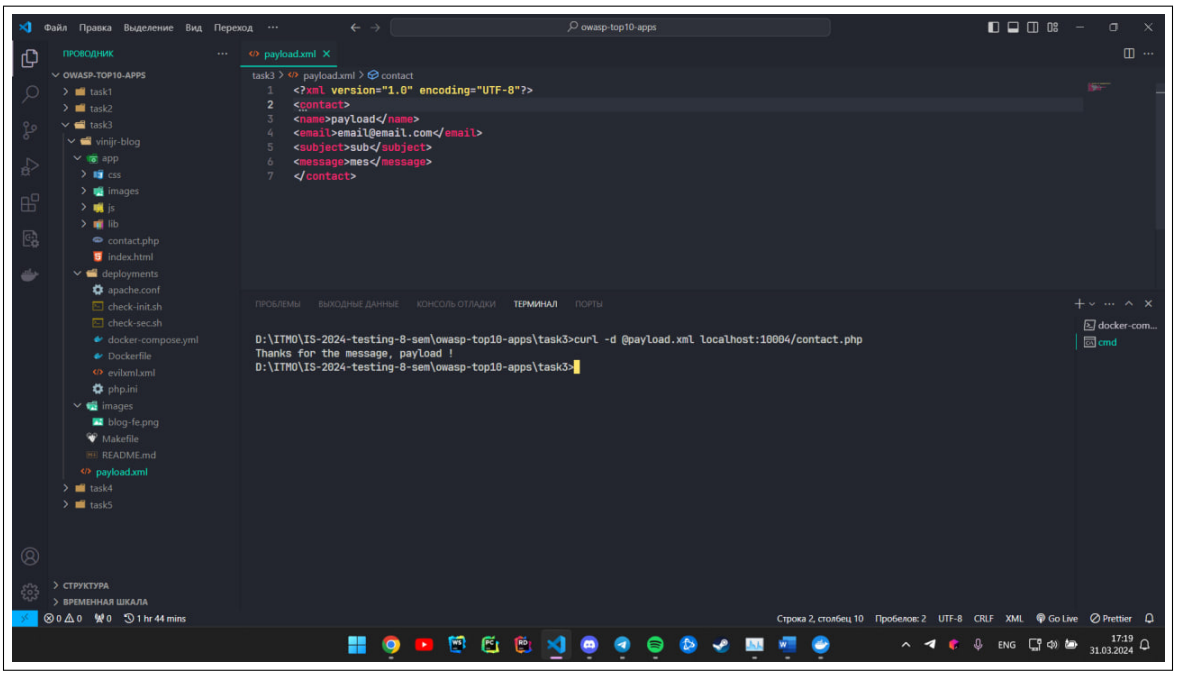

Подтвердите действие на localhost.:10007
admin password
OK
Вход в аккаунт
Логин: admin
Пароль:
Войти
Copyright © Gossip World 2018

3 Security Misconfiguration

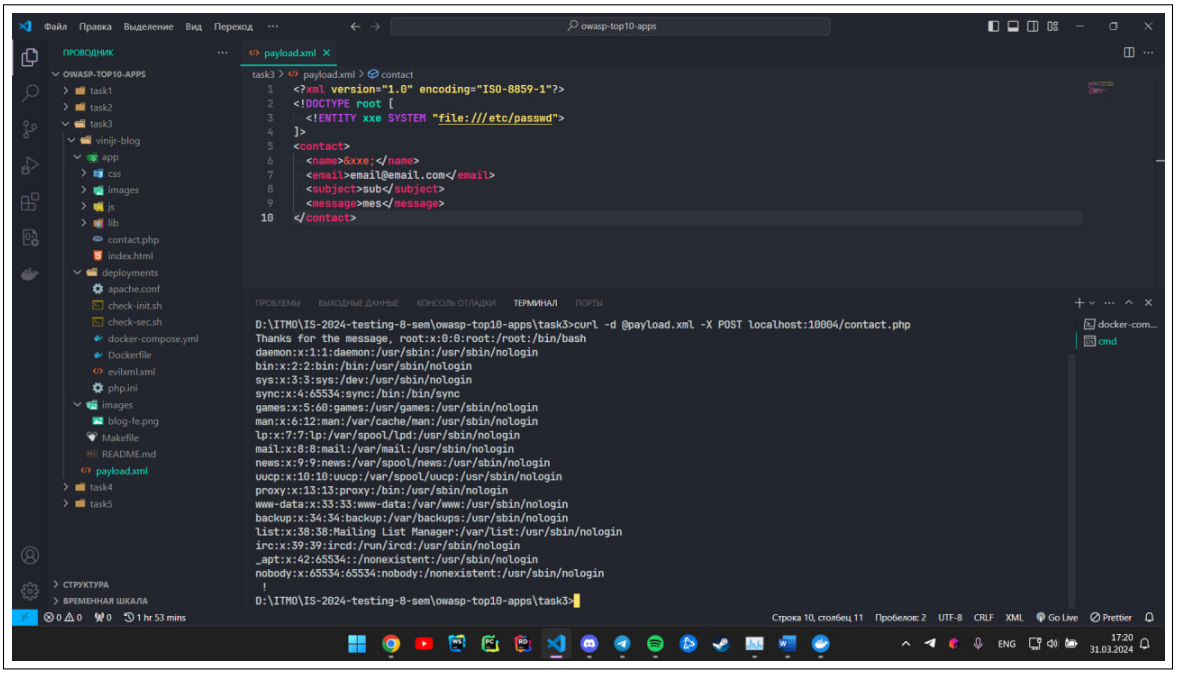
3.1 Изучить Request Payload в Developer Tools

Изучил

3.2 Используя curl и повторяя структуру POST, создать файл payload.xml; запустить curl -d @payload.xml localhost:10004/contact.php

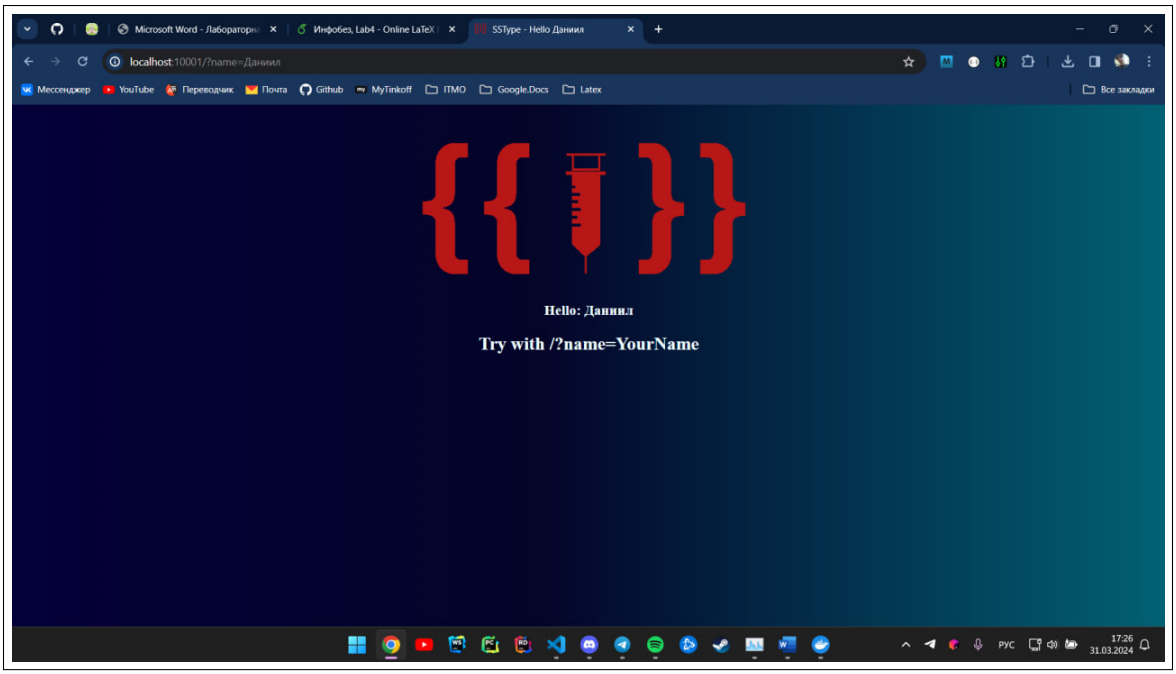


3.3 Доработать файл @payload.xml и получить от сервера /etc/passwd

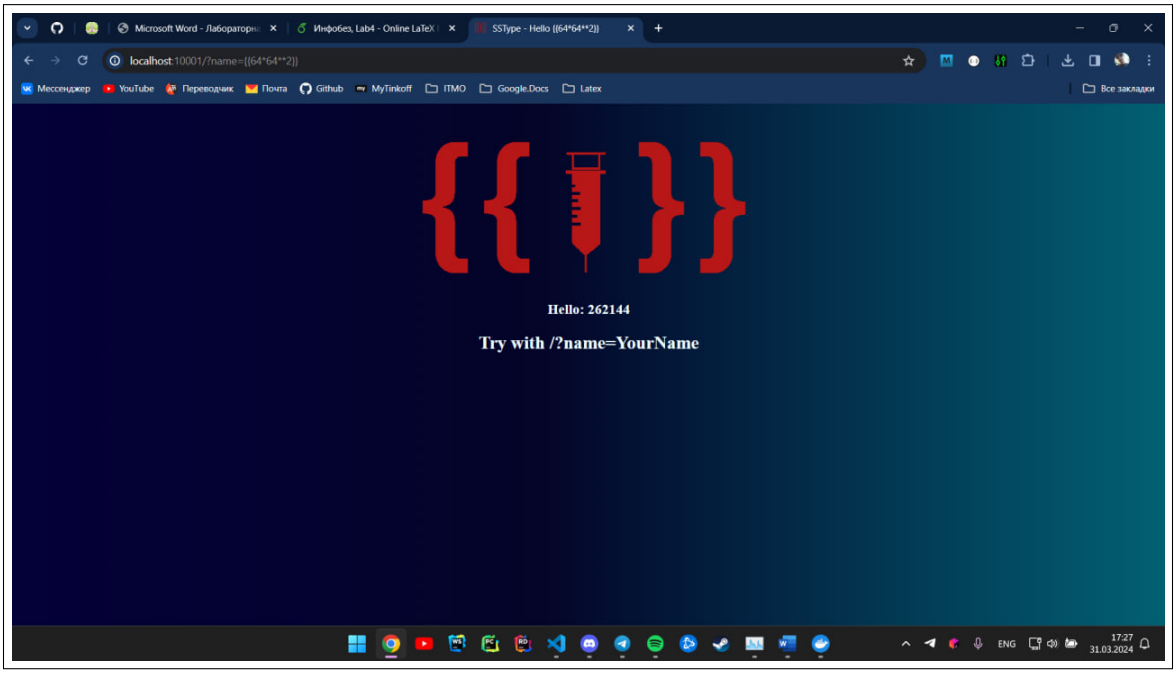


4 Server Side Injection

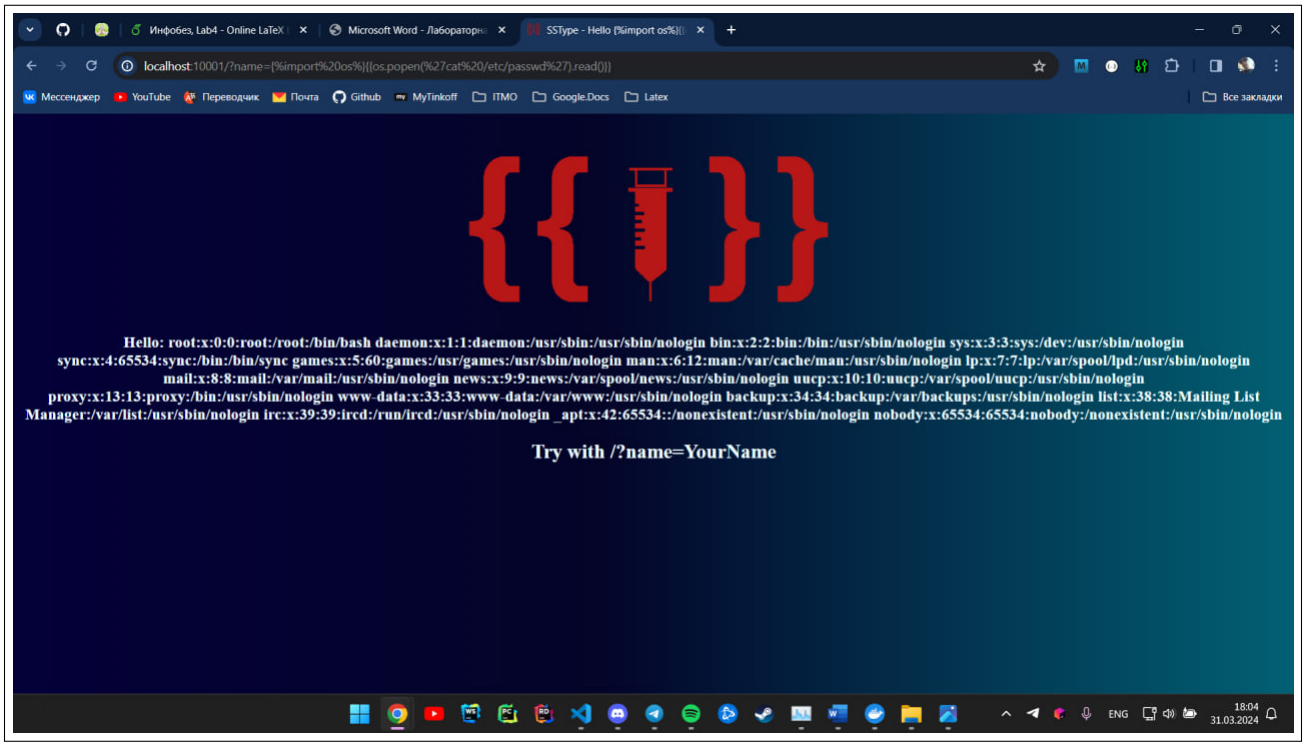
4.1 Исследовать работу приложения curl http://localhost:10001/?name=ваше_имя



4.2 Используя операцию умножения в параметре name проверить некорректную обработку пользовательского ввод



4.3 Вывести файл /etc/passwd используя в строке запроса Python команду (os.popen)



5 NoSQL Injection

5.1 Изучить файл db.js

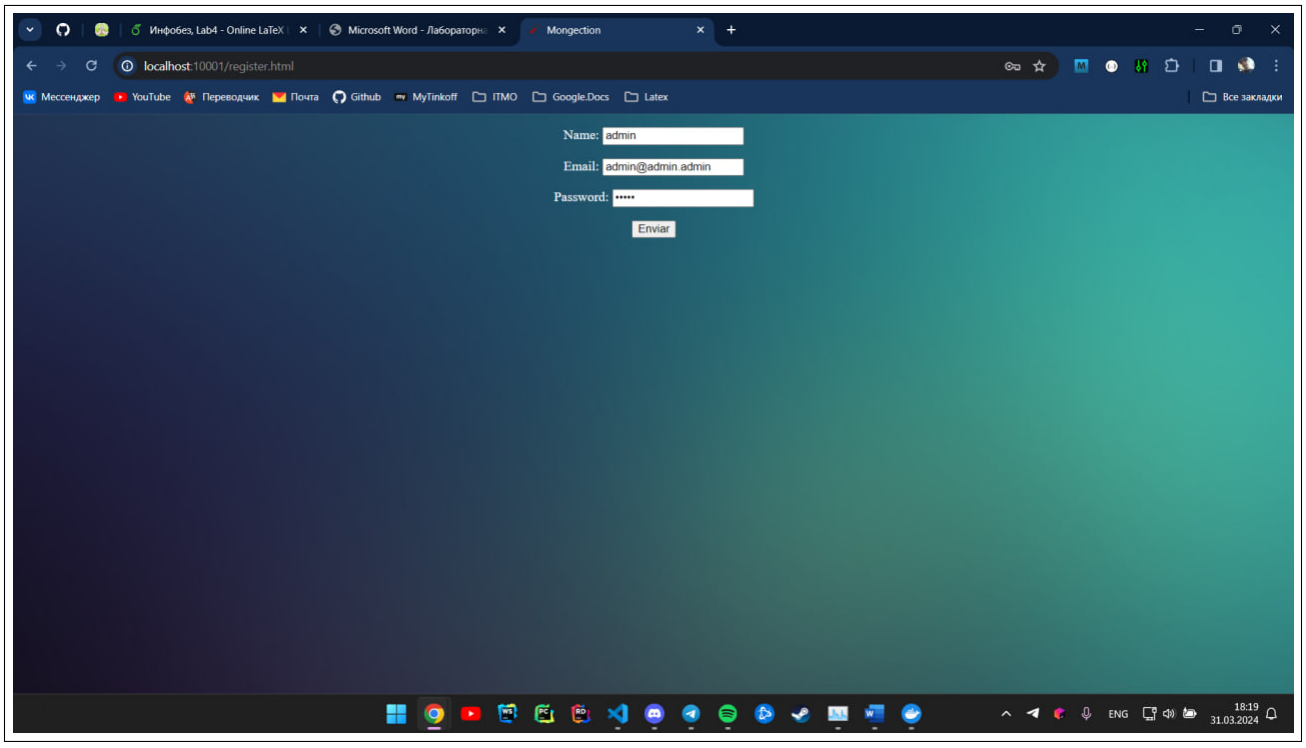
```
const existsUser = await User.find({$and: [ { email: email}, { password: password} ]});
```

5.2 Ответить на вопрос за что отвечают $ne и $gt в MongoDB

$ne — сравнение значений, не равных указанному значению.
$gt — сравнение значений, превышающих указанное значение.

5.3 Обойти аутентификацию, создав вредоносный запрос с использованием {"$ne": } в полях email и password

5.3.1 Регистрация нового пользователя



5.3.2 Обход аутентификации