

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

Информационная безопасность
Работа 3: Аудит безопасности веб-приложения

Выполнил:

Сандов Кирилл Алексеевич

Группа:


P3413















2025

Краткое резюме

Было найдено множество уязвимостей через DAST-инструмент ZAP.

Найденные уязвимости:

 Оповещения (14)

- >  SQL-инъекция
- >  Заголовок Content Security Policy (CSP) не задан (87)
- >  Идентификатор (ID) сеанса при перезаписи URL (107)
- >  Междоменная неправильная конфигурация (100)
- >  Отсутствует заголовок (Header) для защиты от клидджекинга (29)
- >  Уязвимость JS Библиотеки (Library)
- >  Включение исходного файла междоменного JavaScript (98)
- >  Заголовок Strict-Transport-Security не установлен (3)
- >  Заголовок X-Content-Type-Options отсутствует (111)
- >  Раскрытие отметки времени - Unix (162)
- >  Раскрытие частной ИС
- >  Получено из кеша (63)
- >  Раскрытие информации - подозрительные комментарии (4)
- >  Современное веб-приложение (50)

Но список далеко не полный. Создатели OWASP Juice Shop выписали все уязвимости приложения на странице Scoreboard:

3%
Hacking Challenges

0%
Coding Challenges

3/172
Challenges Solved

3/28 0/23 0/14
0/37 0/26 0/14

Difficulty Status Tags

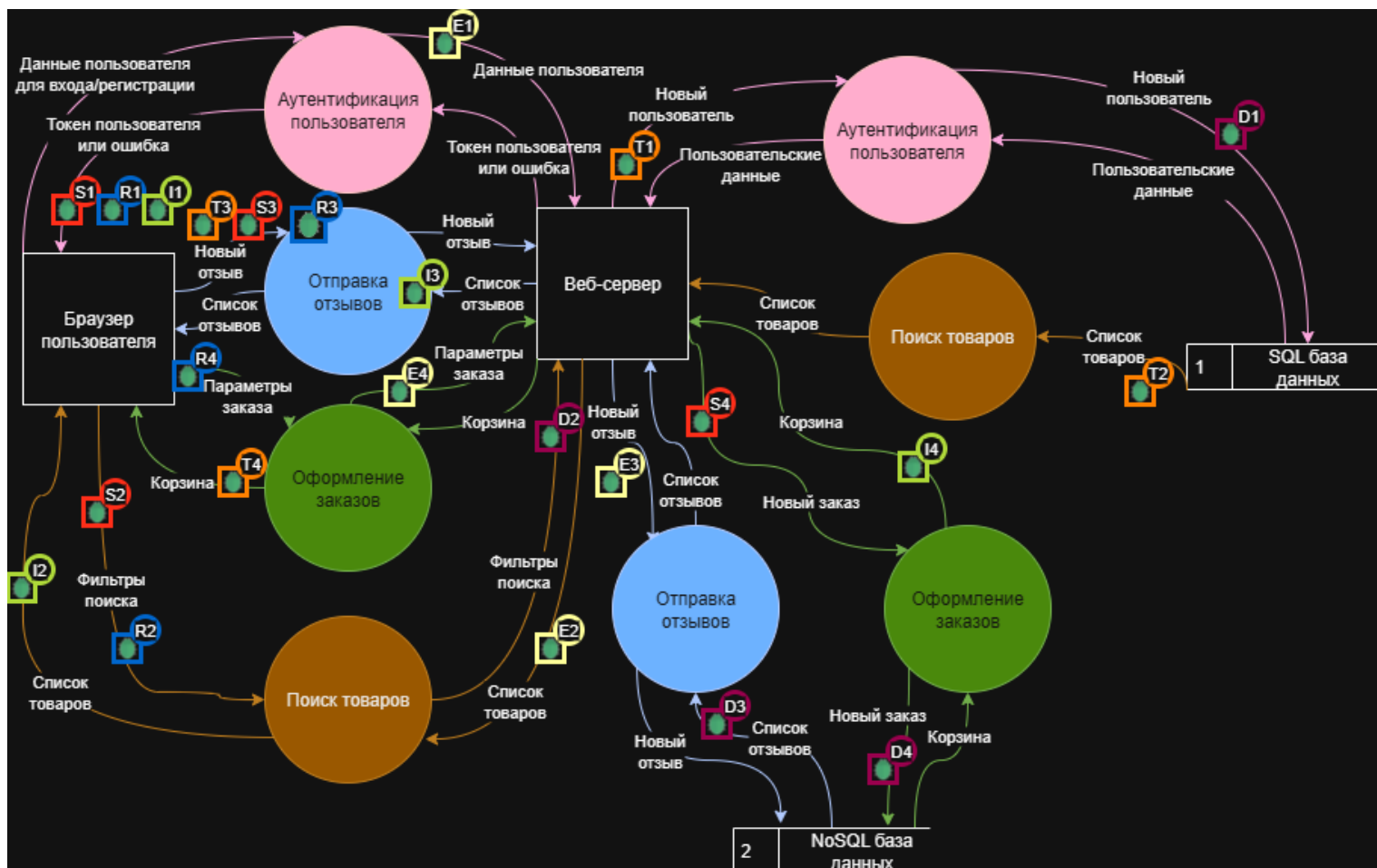
All XSS Sensitive Data Exposure Improper Input Validation Broken Access Control Unvalidated Redirects Vulnerable Components Broken Authentication Security through Obscurity Insecure Deserialization Miscellaneous

Broken Anti Automation Injection Security Misconfiguration Cryptographic Issues XXE

17 challenges are unavailable on Docker due to security concerns or technical incompatibility! Show them anyways

<div>Miscellaneous</div> <div>Score Board</div> <div>Find the carefully hidden "Score Board" page.</div> <div>Tutorial Code Analysis With Coding Challenge</div>	<div>XSS</div> <div>DOM XSS</div> <div>Perform a DOM XSS attack with <code><iframe src="javascript:alert('XSS')"></code>.</div> <div>Tutorial Good for Demos With Coding Challenge</div>	<div>XSS</div> <div>Bonus Payload</div> <div>Use the bonus payload <code><iframe width="1000" height="100" scrolling="no" frameborder="no" allow="autoplay" src="https://w.ovoynkzlog.com/01kxprj3"></code></div> <div>Shareables Tutorial With Coding Challenge</div>	<div>Miscellaneous</div> <div>Privacy Policy</div> <div>Read our privacy policy.</div> <div>Good Practice Tutorial Good for Demos</div>
<div>Miscellaneous</div> <div>Bully Chatbot</div> <div>Receive a coupon code from the support chatbot.</div> <div>Shareables Break Force</div>	<div>Sensitive Data Exposure</div> <div>Confidential Document</div> <div>Access a confidential document.</div> <div>Good for Demos With Coding Challenge</div>	<div>Security Misconfiguration</div> <div>Error Handling</div> <div>Provide an error that is neither very gracefully nor consistently handled.</div> <div>Prerequisites</div>	<div>Sensitive Data Exposure</div> <div>Exposed Metrics</div> <div>Find the endpoint that serves usage data to be scraped by a popular marketing system.</div> <div>Good Practice With Coding Challenge</div>
<div>Miscellaneous</div> <div>Mass Dispel</div> <div>Close multiple "Challenge solved"-notifications in one go.</div> <div>Shareables</div>	<div>Improper Input Validation</div> <div>Missing Encoding</div> <div>Retrieve the photo of Eijon's cat in "molese combat-mode".</div> <div>Shareables</div>	<div>Unvalidated Redirects</div> <div>Outdated Allowlist</div> <div>Let us redirect you to one of our crypto currency addresses which are not promoted any longer.</div> <div>Code Analysis With Coding Challenge</div>	<div>Improper Input Validation</div> <div>Repetitive Registration</div> <div>Follow the DRY principle while registering a user.</div> <div>Good Practice With Coding Challenge</div>
<div>Broken Access Control</div> <div>Web3 Sandbox</div>	<div>Improper Input Validation</div> <div>Zero Stars</div>	<div>Sensitive Data Exposure</div> <div>Exposed credentials</div>	<div>Injection</div> <div>Login Admin</div>

DFD



Анализ угроз по методике STRIDE

Тип	Описание
Spoofing	<p>S1) На потоке данных «Аутентификация пользователя» можно украсть сессионный токен пользователя из-за уязвимости "Заголовок Content Security Policy (CSP) не задан", сделав XSS-атаку</p> <p>S2) На потоке данных «Поиск товаров» можно выдавать себя за другого пользователя, подделав токен из-за уязвимости «Unsigned JWT» (или «Forged Signed JWT») и отправлять поисковые запросы от его имени.</p> <p>S3) На потоке данных «Отправка отзывов» можно опубликовать отзыв от имени другого пользователя из-за уязвимости «Forged Feedback», подменив идентификатор автора в запросе.</p>

	<p>S4) На потоке данных «Оформление заказов» можно оформить заказ от лица жертвы из-за уязвимости «CSRF», отправив кросс-сайтовый запрос из другого происхождения.</p>
Tampering	<p>T1) На потоке данных «Аутентификация пользователя» можно изменить чужие учётные данные, выполнив процесс «Forgot Password» из-за недостаточной проверки прав.</p> <p>T2) На потоке данных «Поиск товаров» можно изменить результаты и/или данные о товарах через SQL-инъекцию в параметре поиска (уязвимость «SQL Injection»).</p> <p>T3) На потоке данных «Отправка отзывов» можно подменить автора и содержимое отзывов других пользователей (уязвимость «Forged Review»), отправив изменённый запрос.</p> <p>T4) На потоке данных «Оформление заказов» можно изменить содержимое заказа другого пользователя (уязвимость «Manipulate Basket»), подложив дополнительные товары в его корзину.</p>
Repudiation	<p>R1) На потоке данных «Аутентификация пользователя» можно выполнить действия под поддельной учётной записью из-за уязвимости «Unsigned JWT/ Forged Signed JWT» — логи зафиксируют фальшивый идентификатор, что позволяет отрицать авторство.</p> <p>R2) На потоке данных «Поиск товаров» можно инициировать запросы от имени жертвы через «Reflected/DOM XSS» в параметрах поиска — действия будут записаны как выполненные жертвой, что позволяет злоумышленнику отрицать причастность.</p> <p>R3) На потоке данных «Отправка отзывов» можно опубликовать отзыв под именем другого пользователя из-за уязвимости «Forged</p>

	<p>Feedback/Forged Review», что делает журналы недостоверными для установления авторства.</p> <p>R4) На потоке данных «Оформление заказов» можно оформить заказ от лица жертвы через «CSRF», и логи покажут действия как инициированные жертвой, что позволяет злоумышленнику отрицать участие.</p>
Information Disclosure	<p>I1) На потоке данных «Аутентификация пользователя» можно раскрыть действительные тестовые учётные данные из-за уязвимости «Exposed credentials» (жёстко прописаны на клиенте) и получить доступ к аккаунту жертвы.</p> <p>I2) На потоке данных «Поиск товаров» можно посмотреть чужую корзину, подменив cookie с идентификатором корзины (уязвимость «View Basket»).</p> <p>I3) На потоке данных «Отправка отзывов» можно раскрыть секрет внешнего сервиса (уязвимость «Leaked API Key»), извлекая API-ключ из загруженного фронтенд-кода/сетевых запросов страницы отзывов.</p> <p>I4) На потоке данных «Оформление заказов» можно эксфильтровать сведения о заказах других пользователей через уязвимость «NoSQL Exfiltration», передав инъекционный запрос, возвращающий чужие записи.</p>
Denial of Service	<p>D1) На потоке данных «Аутентификация пользователя» можно вызвать отказ в обслуживании массовыми попытками входа (отсутствует жёсткий rate limiting/lockout): непрерывный брутфорс «Password Strength/Login Admin» перегружает обработку аутентификации.</p> <p>D2) На потоке данных «Поиск товаров» можно «повесить» рендер/бэкенд через SSTi-пейлоад в параметре поиска (уязвимость «SSTi»),</p>

	<p>вызывающий ресурсоёмкие вычисления.</p> <p>D3) На потоке данных «Отправка отзывов» можно сделать БД недоступной, добавив в тело создания отзыва поле с "\$where": "sleep(...)" (уязвимость «NoSQL DoS»).</p> <p>D4) На потоке данных «Оформление заказов» можно блокировать оформление, инъецировав "\$where": "sleep(...)" в данные чекаута/применения купона в корзине (уязвимость «NoSQL DoS»).</p>
Elevation of Privilege	<p>E1) На потоке данных «Аутентификация пользователя» можно при регистрации сразу получить права администратора (уязвимость «Admin Registration») за счёт подмены признака роли.</p> <p>E2) На потоке данных «Поиск товаров» можно подменить JWT (уязвимость «Unsigned JWT/Forged Signed JWT») и искать/просматривать скрытые или админ-только позиции как администратор.</p> <p>E3) На потоке данных «Отправка отзывов» можно получить фактические модераторские полномочия и удалить все 5-звёздочные отзывы (уязвимость «Five-Star Feedback») при отсутствии корректной проверки прав.</p> <p>E4) На потоке данных «Оформление заказов» можно незаконно получить привилегированный статус «Deluxe Membership» (уязвимость «Deluxe Fraud»), расширяющий права и льготы при заказах.</p>

Таблица уязвимостей

Название	XSS-атака из-за отсутствия заголовков CSP
Описание	Можно встроить в страницу JS-код, который перешлёт злоумышленнику локальные

	данные пользователя
Уровень риска	6.2 (Средний)
Категория OWASP Top 10	A03:2021-Injection
Предложения по исправлению	<p>Нужно установить заголовок Content-Security-Policy:</p> <pre>Content-Security-Policy: default-src 'self'; script-src 'self' 'unsafe-eval'</pre> <p>Он заблокирует вызовы eval(), запретит динамическое выполнение скриптов</p>

Название	SQL-инъекция через запрос поиска
Описание	<p>Можно изменить записи в таблице Products через SQL-инъекцию, которую нужно передать в query-параметре запроса:</p> <p>/rest/products/search?q= <SQL-инъекция тут></p>
Уровень риска	9,8 (Критический)
Категория OWASP Top 10	A03:2021-Injection
Предложения по исправлению	Нужно использовать prepared-statements в SQL-запросах, чтобы не выполнять сторонний SQL-запрос

Название	Forged Feedback – подмена логина
Описание	<p>Эксплойт позволяет отправлять отзывы под именем другого пользователя</p> <p>Нужно установить нужный логин пользователя в запросе /rest/products/<id>/reviews:</p> <pre>author: "amphyxs@gmail.com" message: "p"</pre>
Уровень риска	8,6 (Высокий)
Категория	A07:2021-Identification and Authentication

OWASP Top 10	Failures
Предложения по исправлениям	Нужно определять пользователя по его JWT-токену из заголовка Authorization, а не через поле в запросе

Название	View Basket – отсутствие проверок владения сущностями
Описание	<p>Эксплойт позволяет посмотреть чужую корзину.</p> <p>Для этого надо в запросе GET /rest/basket/<id> использовать id искомой корзины</p>
Уровень риска	4.3 (Medium)
Категория OWASP Top 10	A07:2021-Identification and Authentication Failures
Предложения по исправлениям	Нужно определять пользователя по его JWT-токену из заголовка Authorization и выдавать соответствующую ему и только ему корзину, а не запрашивать её id из клиента. Либо хотя бы проверять перед запросом, является ли пользователем владельцем корзины

Название	Admin Registration – нелегальное становление администратором
Описание	<p>Эксплойт позволяет зарегистрировать пользователя как администратора.</p> <p>Для этого надо к запросу POST /rest/Users добавить поле "role": "admin"</p>
Уровень риска	7.5 (High)
Категория OWASP Top 10	A01:2021-Broken Access Control
Предложения по исправлениям	Нужно задать для эндпоинта DTO без поля role в ней, проверяя его валидность. А также сделать отдельный эндпоинт для изменения роли пользователей, который доступен только для других администраторов

Скриншоты

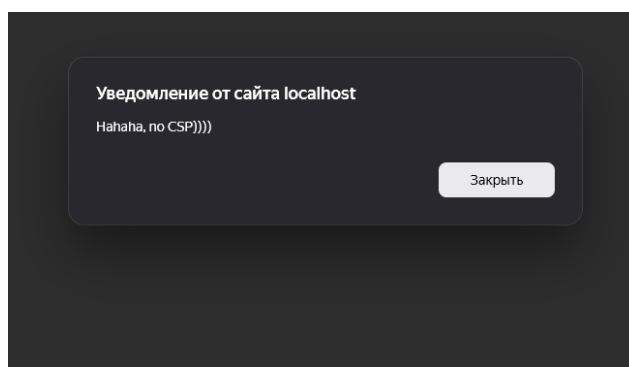
1) XSS-атака из-за отсутствия заголовков CSP

В поисковую строку надо вставить:

```
1. <iframe src="javascript:alert(`Hahaha, no CSP`))`)">
```

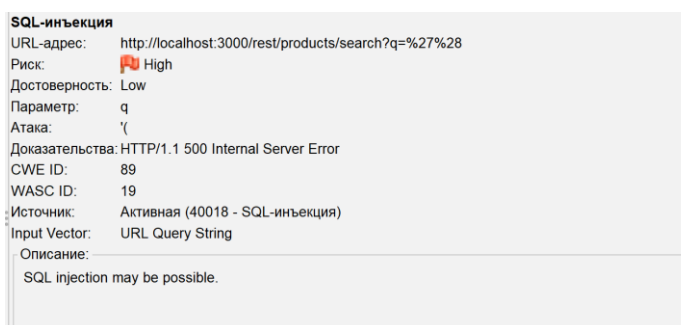


Выполнится JS-код и отобразится алерт:



2) SQL-инъекция через запрос поиска

Была найдена в ZAP, и эта ZAP её выполнил:



ZAP отправил `'(` в запросе поиска, и сервер исполнил это как SQL, вернув HTTP 500 ошибку с результатом исполнения SQL-запроса:

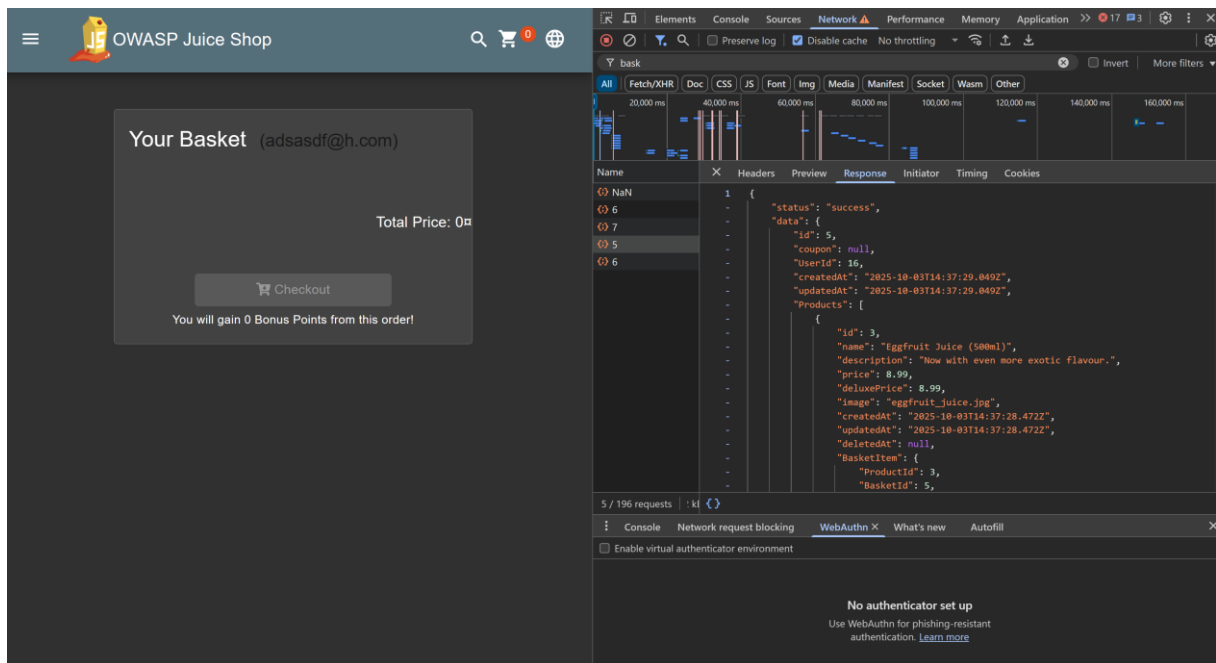
HTTP/1.1 500 Internal Server Error

3) Forged Feedback – подмена логина

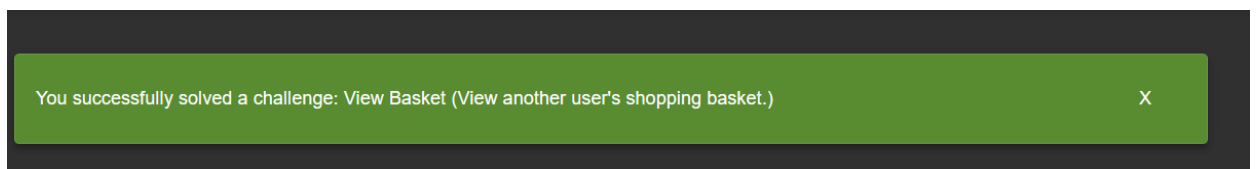
Можно опубликовать отзыв будучи другим пользователем. Тут был активный аккаунт amphyxs@gmail.com, но указав в поле author запроса создания отзыва email admin@juice-sh.op:

[illegible]

Можно было увидеть отзывы от чужого аккаунта:



И появился факт выполнения challenge:



5) Admin Registration – нелегальное становление администратором

Был отправлен запрос на регистрацию пользователя с параметром "role: admin":

```
> fetch("http://localhost:3000/api/Users/", {
  "headers": {
    "accept": "application/json, text/plain, */*",
    "accept-language": "ru,en;q=0.9",
    "cache-control": "no-cache",
    "content-type": "application/json",
    "pragma": "no-cache",
    "sec-ch-ua": "\"(Not)A;Brand\";v=\"8\", \"Chromium\";v=\"138\", \"YaBrowser\";v=\"25.8\", \"Yowser\";v=\"2.5\"",
    "sec-ch-ua-mobile": "70",
    "sec-ch-ua-platform": "\"Windows\"",
    "sec-fetch-dest": "empty",
    "sec-fetch-mode": "cors",
    "sec-fetch-site": "same-origin",
    "x-user-email": "bossx@dkjgfh.com"
  },
  "referrer": "http://localhost:3000/",
  "body": {
    "role": "admin",
    "email": "fakeadmin@dkjgfh.com",
    "password": "k7j-Myx-vm7-JkA",
    "passwordRepeat": "k7j-Myx-vm7-JkA",
    "securityQuestion": "id:5,question:Maternal grandmother's first name?",
    "createdAt": "2025-09-14T17:27:28.142Z",
    "updatedAt": "2025-09-14T17:27:28.142Z",
    "securityAnswer": "2",
    "method": "POST",
    "mode": "cors",
    "credentials": "include"
  }
});
```

И зайдя в зарегистрированный аккаунт отобразился интерфейс для администратора (аватар со шпионом есть только у администратора):

