Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Информационная безопасность

**Работа 3: Аудит безопасности веб-приложения**

Выполнил:

Сандов Кирилл Алекссевич

Группа:

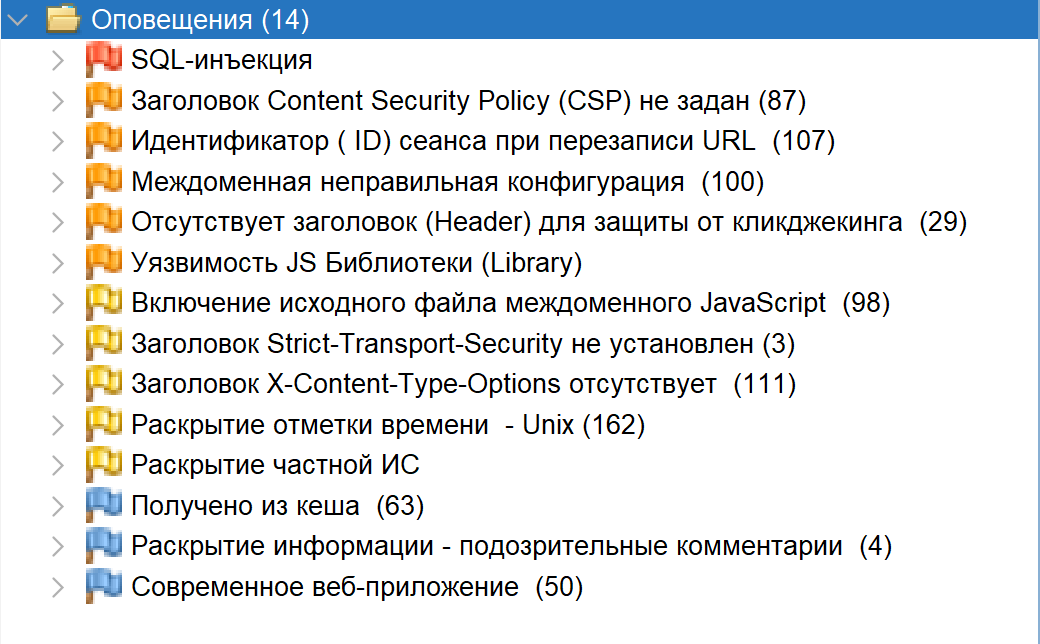
P3413

2025

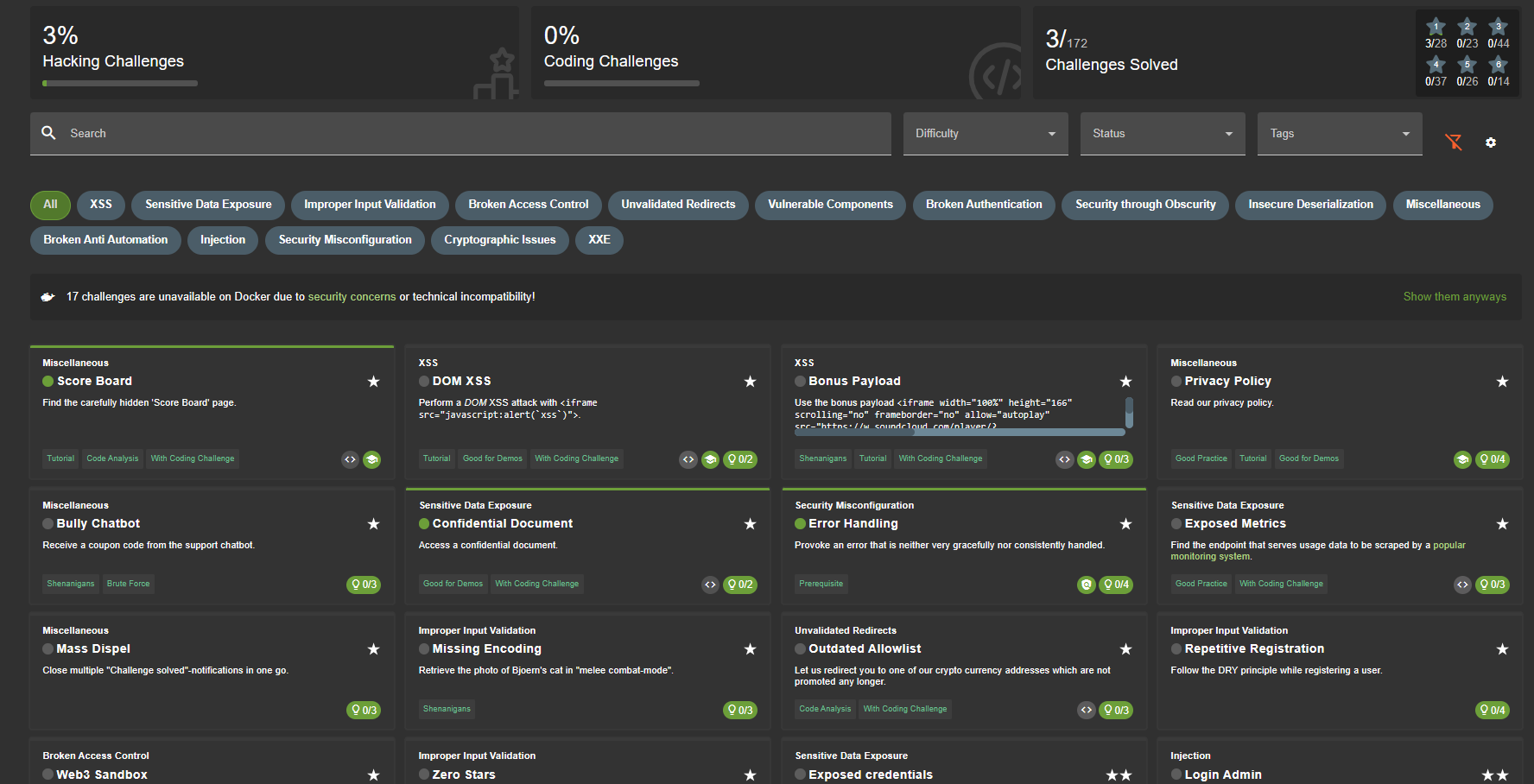
## Краткое резюме

Было найдено множество уязвимостей через DAST-инструмент ZAP.

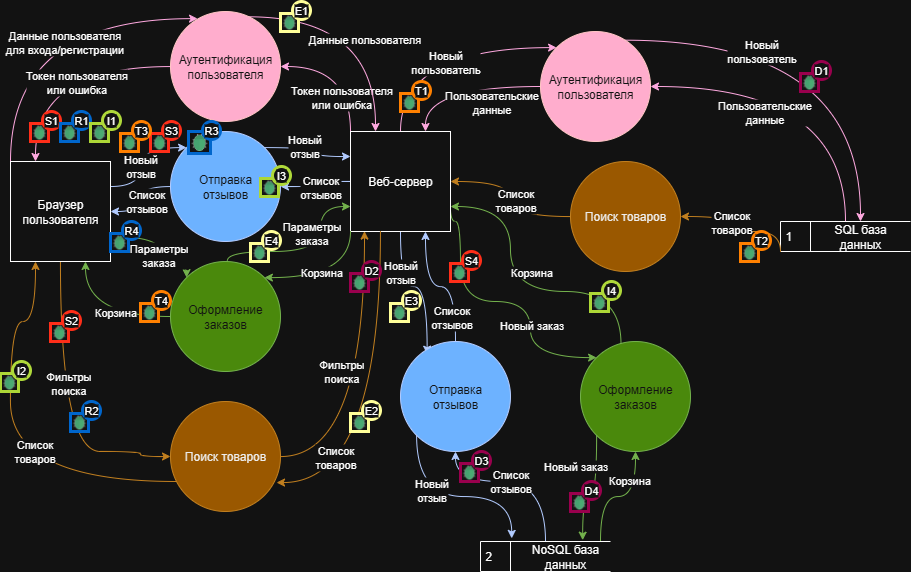
Найденные уязвимости:



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



## DFD



## Анализ угроз по методике STRIDE

|  |  |
| --- | --- |
| Тип | Описание |
| S Spoofing | S1) На потоке данных «Аутентификация пользователя» можно украсть сессионный токен пользователя из-за уязвимости “Заголовок Content Security Policy (CSP) не задан”, сделав XSS-атаку  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.  S3) На потоке данных «Отправка отзывов» можно опубликовать отзыв от имени другого пользователя из‑за уязвимости «Forged Feedback», подменив идентификатор автора в запросе.  S4) На потоке данных «Оформление заказов» можно оформить заказ от лица жертвы из‑за уязвимости «CSRF», отправив кросс‑сайтовый запрос из другого происхождения. |
| Tampering | T1) На потоке данных «Аутентификация пользователя» можно изменить чужие учётные данные, выполнив процесс «Forgot Password» из‑за недостаточной проверки прав.  T2) На потоке данных «Поиск товаров» можно изменить результаты и/или данные о товарах через SQL-инъекцию в параметре поиска (уязвимость «SQL Injection»).  T3) На потоке данных «Отправка отзывов» можно подменить автора и содержимое отзывов других пользователей (уязвимость «Forged Review»), отправив изменённый запрос.  T4) На потоке данных «Оформление заказов» можно изменить содержимое заказа другого пользователя (уязвимость «Manipulate Basket»), подложив дополнительные товары в его корзину. |
| Repudiation | 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.  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.  R3) На потоке данных «Отправка отзывов» можно опубликовать отзыв под именем другого пользователя из‑за уязвимости «Forged Feedback/Forged Review», что делает журналы недостоверными для установления авторства.  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. |
| Information Disclosure | 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.  I2) На потоке данных «Поиск товаров» можно посмотреть чужую корзину, подменив cookie с идентификатором корзины (уязвимость «View Basket»).  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.  I4) На потоке данных «Оформление заказов» можно эксфильтровать сведения о заказах других пользователей через уязвимость «NoSQL Exfiltration», передав инъекционный запрос, возвращающий чужие записи. |
| Denial of Service | 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.  D2) На потоке данных «Поиск товаров» можно “повесить” рендер/бэкенд через SSTi-пейлоад в параметре поиска (уязвимость «SSTi»), вызывающий ресурсоёмкие вычисления.  D3) На потоке данных «Отправка отзывов» можно сделать БД недоступной, добавив в тело создания отзыва поле с "$where": "sleep(...)" (уязвимость «NoSQL DoS»).  D4) На потоке данных «Оформление заказов» можно блокировать оформление, инъецировав "$where": "sleep(...)" в данные чекаута/применения купона в корзине (уязвимость «NoSQL DoS»). |
| Elevation of Privilege | E1) На потоке данных «Аутентификация пользователя» можно при регистрации сразу получить права администратора (уязвимость «Admin Registration») за счёт подмены признака роли.  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.  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.  E4) На потоке данных «Оформление заказов» можно незаконно получить привилегированный статус «Deluxe Membership» (уязвимость «Deluxe Fraud»), расширяющий права и льготы при заказах. |

## 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 исправленияю | Нужно установить заголовок Content-Security-Policy:    Он заблокирует вызовы eval(), запретит динамическое выполнение скриптов |

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

|  |  |
| --- | --- |
| Название | Forged Feedback – подмена логина |
| Описание | Эксплойт позволяет отправлять отзывы под именем другого пользователя  Нужно установить нужный логин пользователя в запросе /rest/products/<id>/reviews: |
| Уровень риска | 8,6 (Высокий) |
| Категория OWASP Top 10 | A07:2021-Identification and Authentication Failures |
| Предложения по исправленияю | Нужно определять пользователя по его JWT-токену из заголовка Authorization, а не через поле в запросе |

|  |  |
| --- | --- |
| Название | View Basket – отсутствие проверок владения сущностями |
| Описание | Эксплойт позволяет посмотреть чужую корзину.  Для этого надо в запросе GET /rest/basket/<id> использовать id искомой корзины |
| Уровень риска | 4.3 (Medium) |
| Категория OWASP Top 10 | A07:2021-Identification and Authentication Failures |
| Предложения по исправленияю | 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 | Эксплойт позволяет зарегистрировать пользователя как администратора.  Для этого надо к запросу POST /rest/Users добавить поле “role”: “admin” |
| Уровень риска | 7.5 (High) |
| Категория OWASP Top 10 | A01:2021-Broken Access Control |
| Предложения по исправленияю | 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

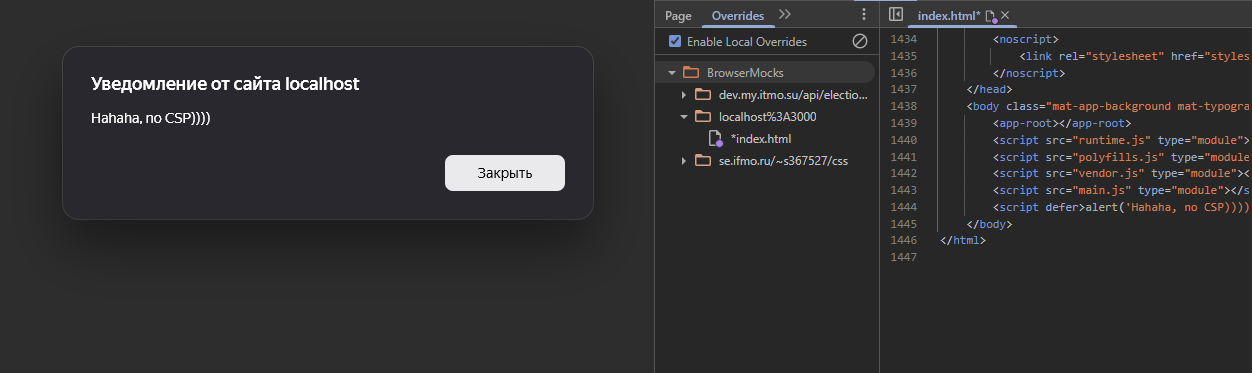
1. XSS-атака из-за отсутствия заголовков CSP

В поисковую строку надо вставить:

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

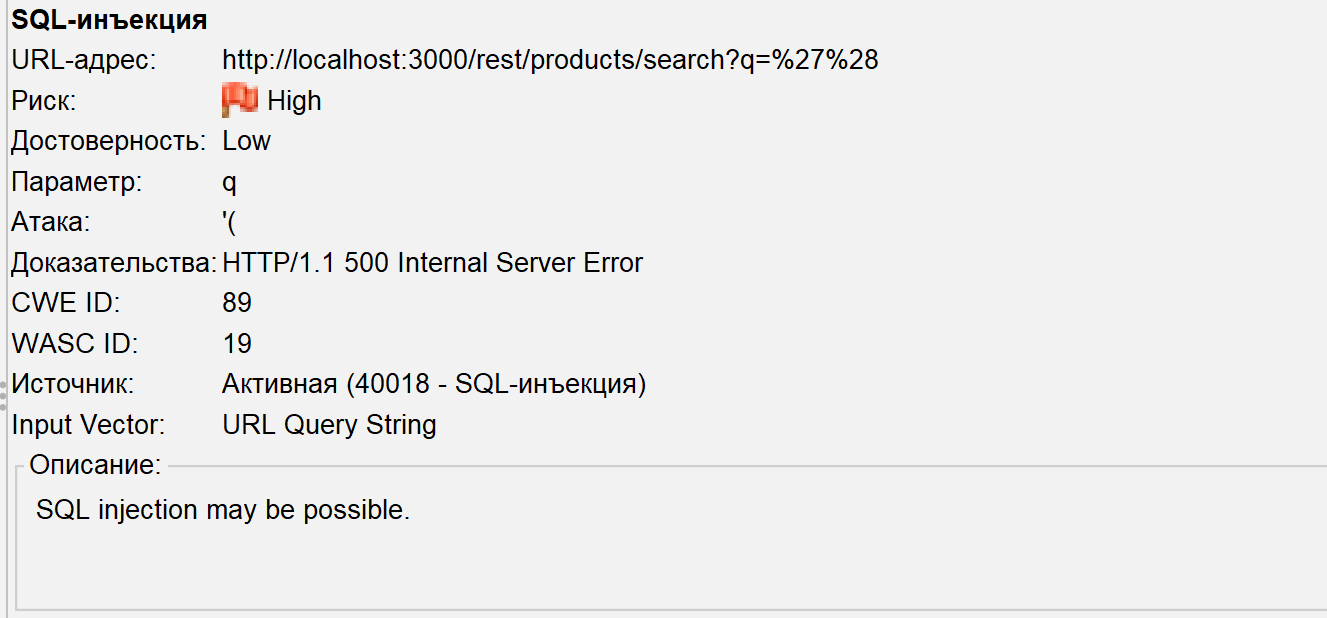


Выполнится JS-код и отобразится алерт:

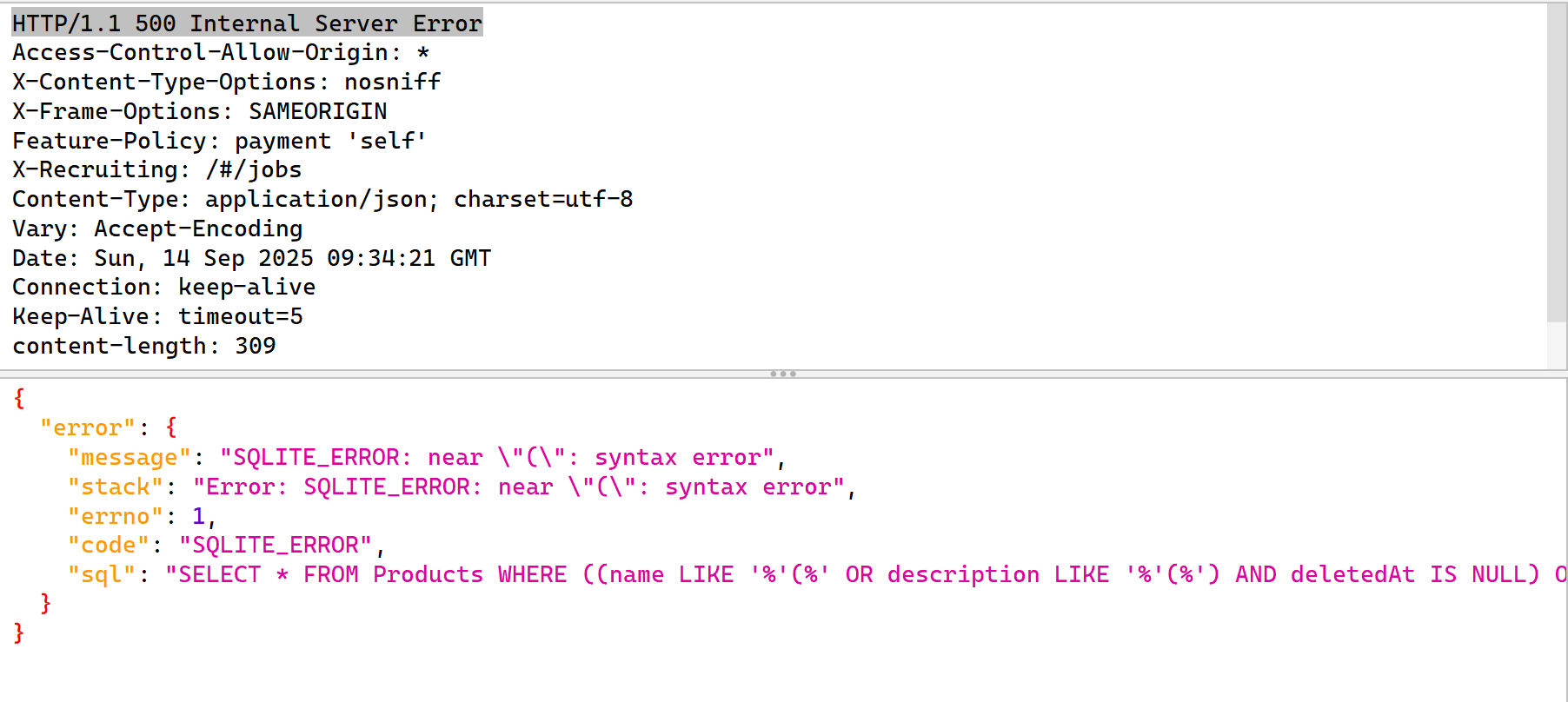


1. SQL-инъекция через запрос поиска

Была найдена в ZAP, и эта ZAP её выполнил:

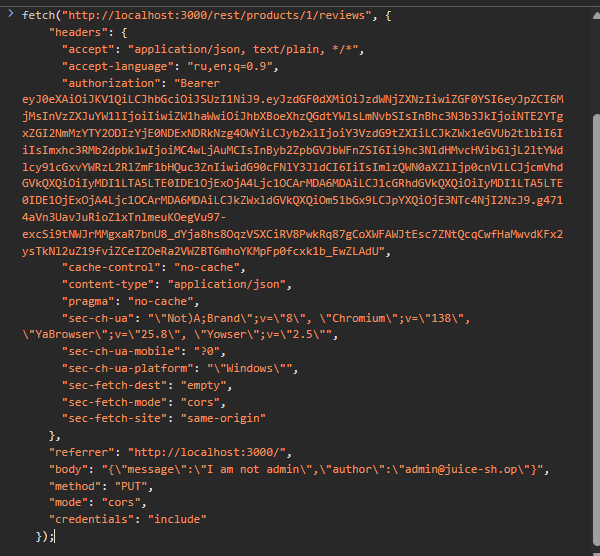


ZAP отправил ‘( в запросе поиска, и сервер исполнил это как SQL, вернув HTTP 500 ошибку с результатом исполнения SQL-запроса:

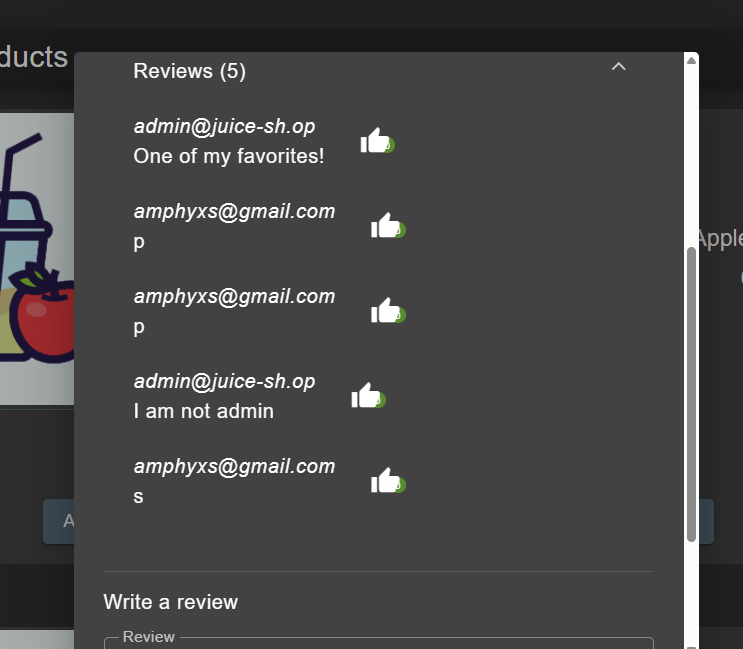


1. Forged Feedback – подмена логина

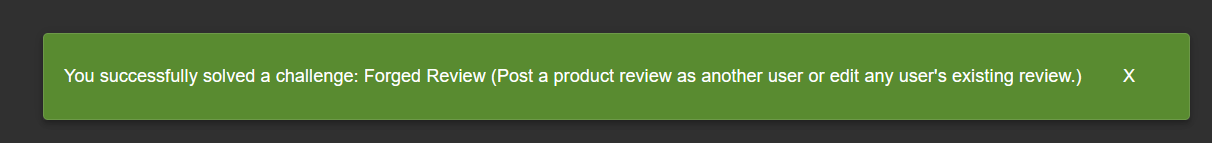
Можно опубликовать отзыв будучи другим пользователем. Тут был активный аккаунт [amphyxs@gmail.com](mailto:amphyxs@gmail.com), но указав в поле author запроса создания отзыва email [admin@juice-sh.op](mailto:admin@juice-sh.op):



Можно было увидеть отзывы от чужого аккаунта:

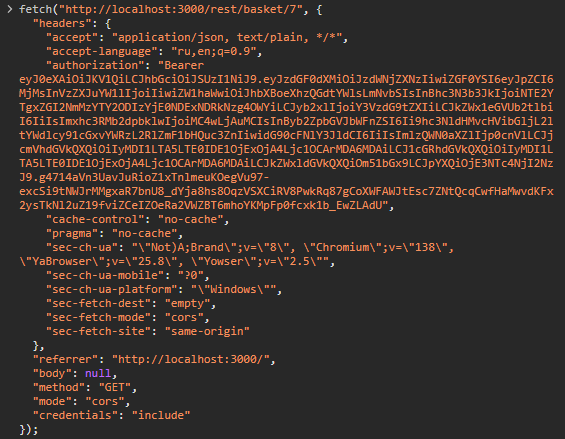


OWASP Juice Shop Scoreboard показал, что challenger по уязвимости был выполнен:

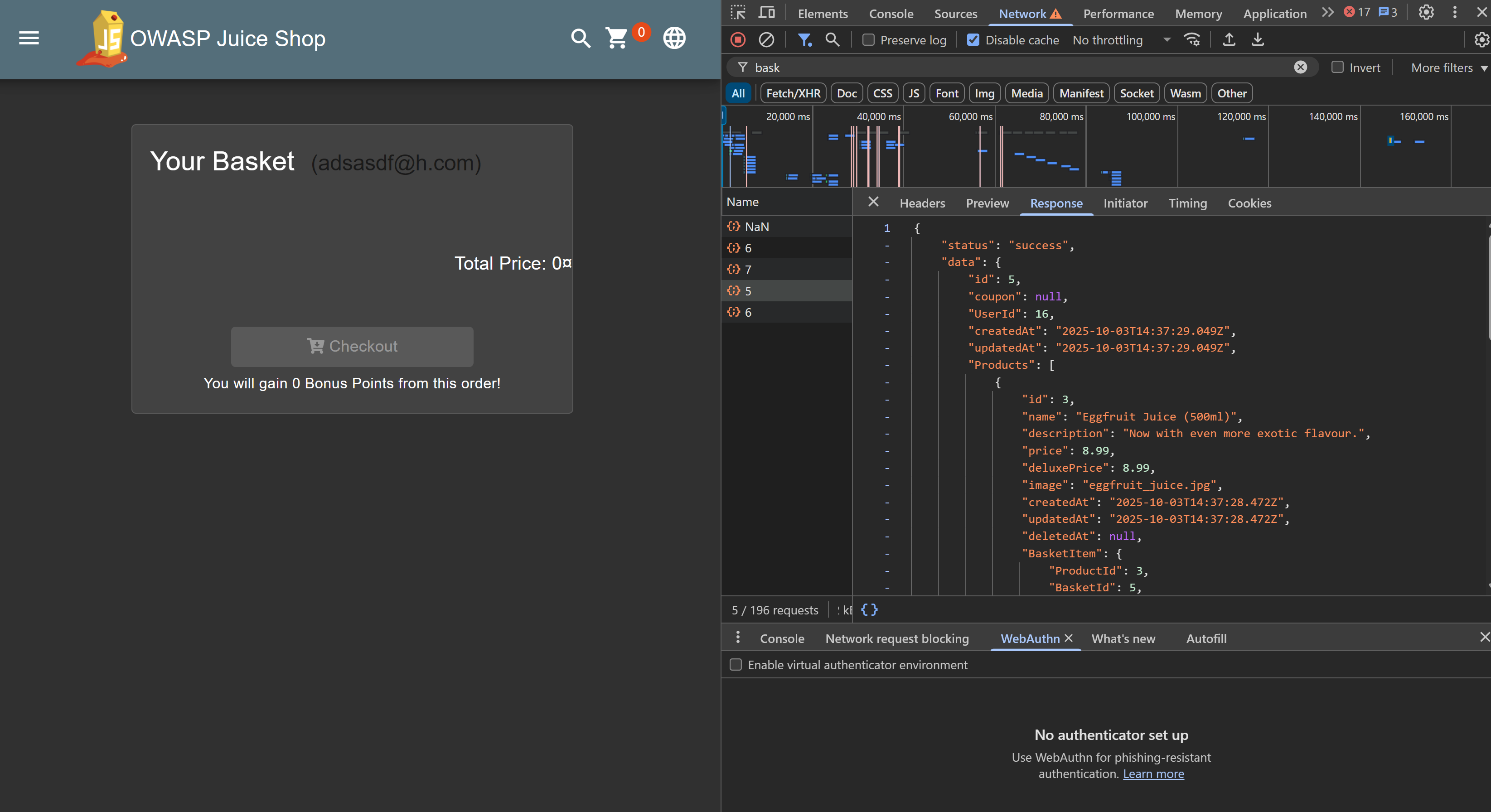


1. View Basket – отсутствие проверок владения сущностями

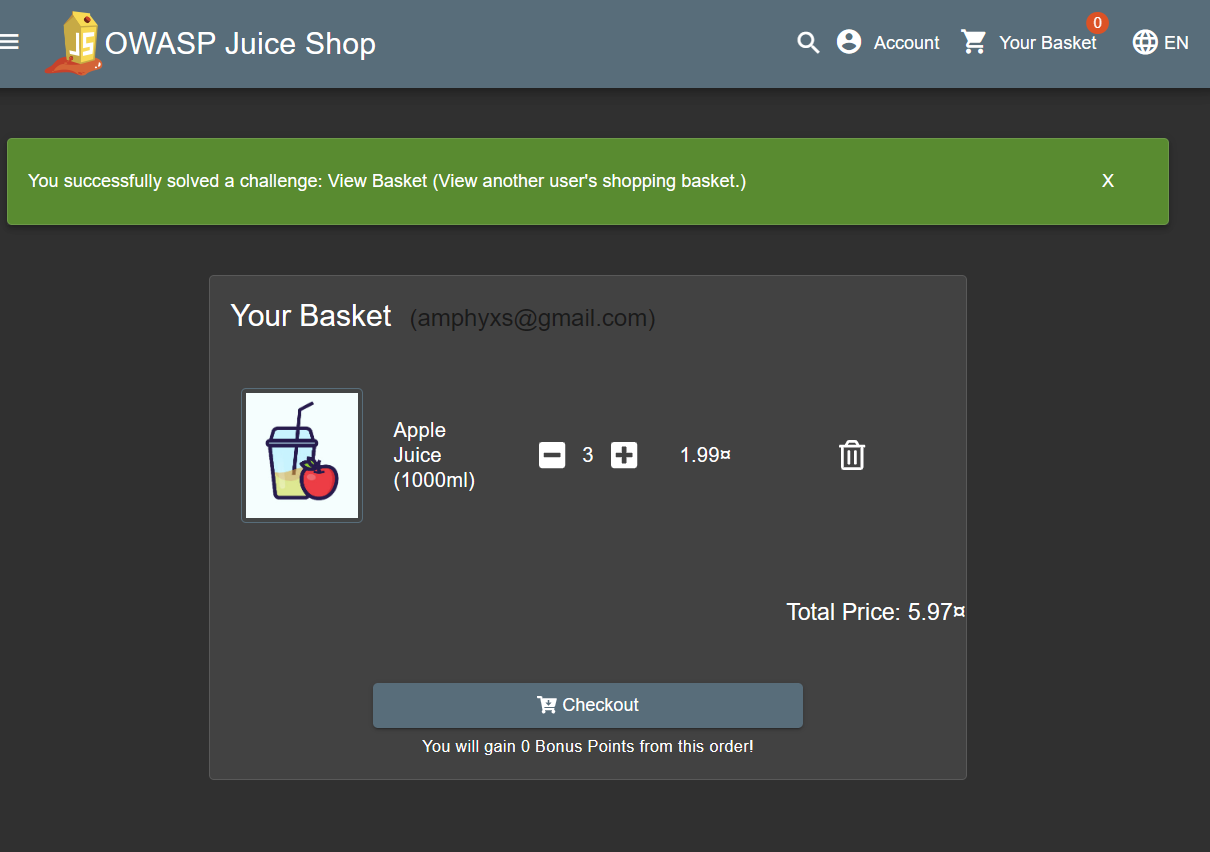
Был отправлен запрос корзины с id, который не равен id корзины текущего пользователя:



В результате на скриншоте видно, что корзина пользователя пуста, но он в запросе из консоли браузера отображается в теле ответа полная корзина чужого пользователя:

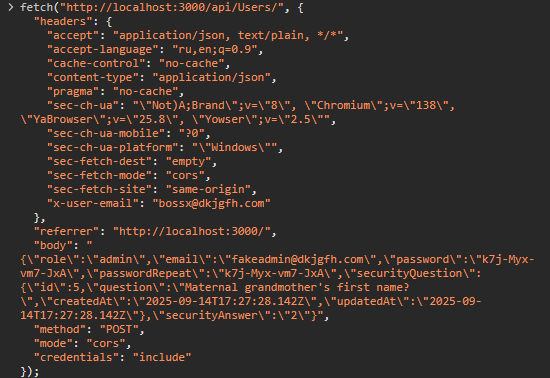


И появился факт выполнения challenge:

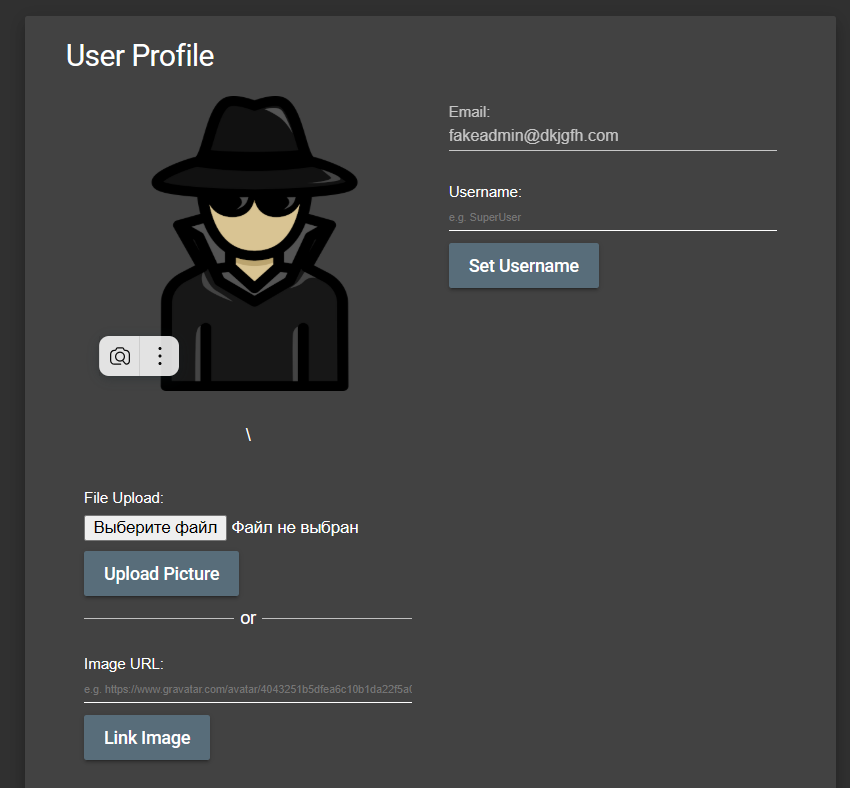


1. Admin Registration – нелегальное становление администратором

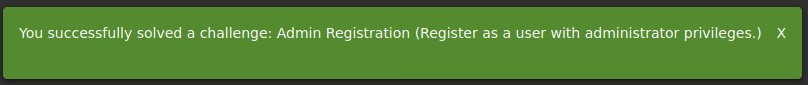
Был отправлен запрос на регистрацию пользователя с параметром “role: admin”:



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



И появилось достижение выполненного challenge:



## Рекомендации по устранению рисков

1. Нужно ко всем запросам добавить соответствующие заголовки CSP и CORS
2. Нужно включить в ORM-фреймворке автоматическую подготовку SQL-запросов, либо вручную подготавливать их в коде
3. Нужно проверять как сущности, так и эндпоинты, на возможность использования конкретным пользователем, чтобы избежать ситуаций с просмотром чужого контента
4. Использовать DAST/SAST-утилиты для проверки безопасности, они бы помогли выявить данные дыры
5. Верно настроить БД так, чтобы долгие запросы не влияли на её доступность

## Отчёт ZAP

<https://github.com/amphyxs/infosec-lab-3/blob/main/zap-report.html>