Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Информационная безопасность

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

Выполнил:

Сандов Кирилл Алекссевич

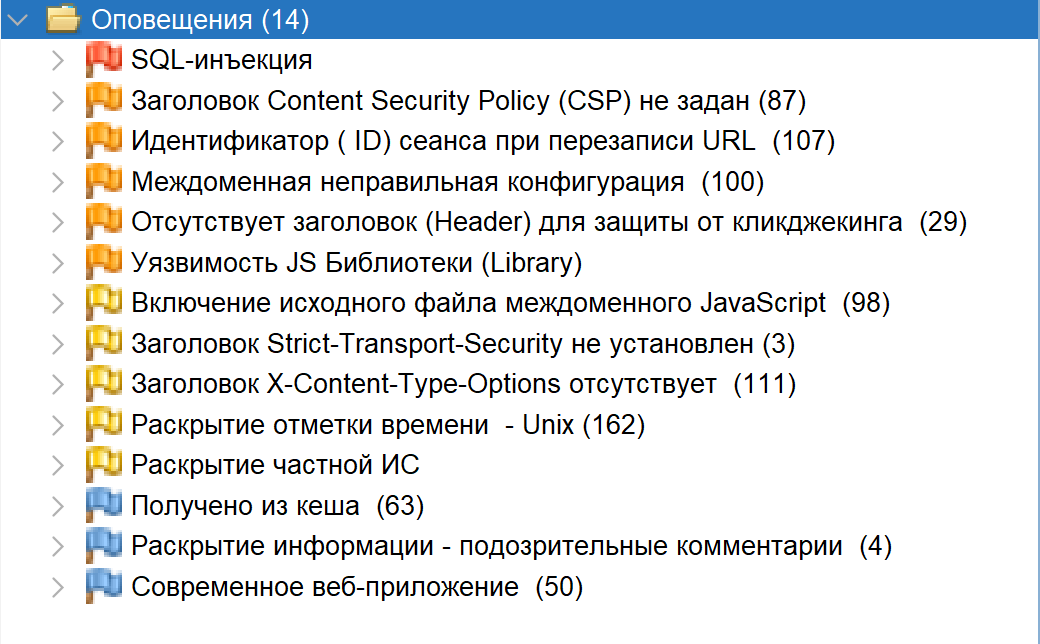
Группа:

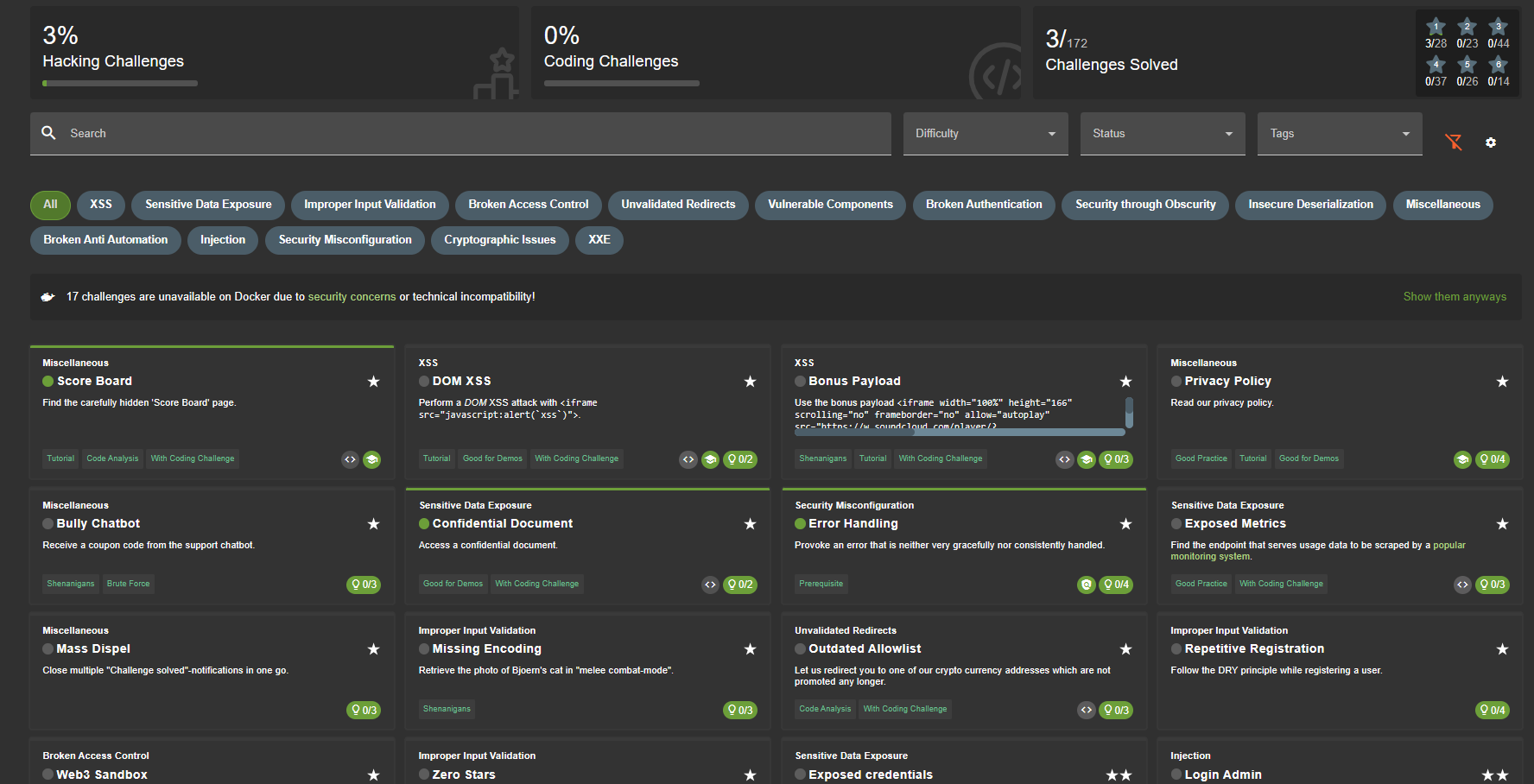
P3413

2025

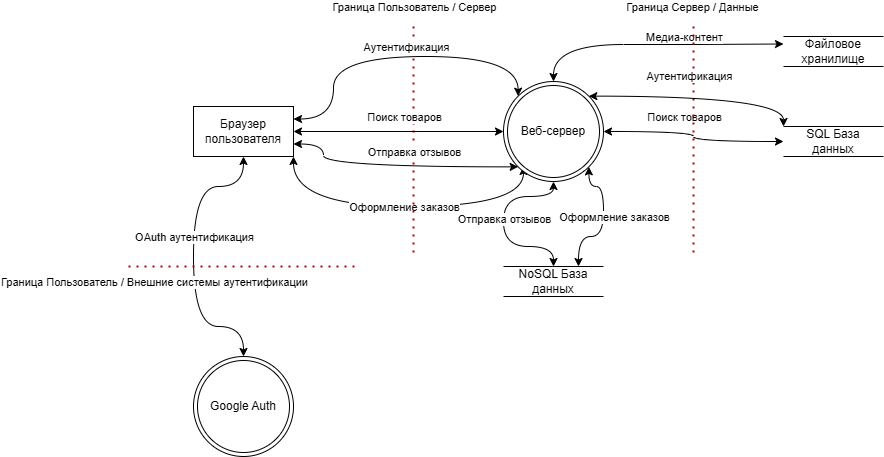
## Краткое резюме

Найденные уязвимости:





## DFD



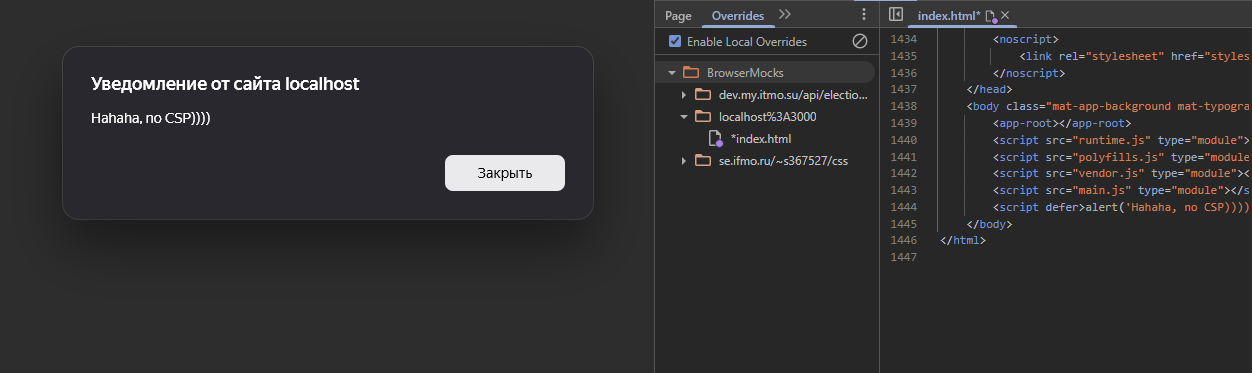
## Анализ угроз по методике STRIDE

|  |  |
| --- | --- |
| Тип | Описание |
| Spoofing | На потоке данных Авторизация можно украсть сессионный токен пользователя из-за уязвимости “Заголовок Content Security Policy (CSP) не задан”, сделав XSS-атаку |
| Tampering | На потоке данных Поиск товаров можно изменить товары через SQL-инъекцию |
| Repudiation | На потоке данных Отправка отзывов можно опубликовать отзыв под именем другого пользователя |
| Information Disclosure | На потоке данных Поиск товаров можно посмотреть чужую корзину  Там надо подменить cookie с id корзины на ту, которую хотим посмотреть |
| Denial of Service | На потоке данных Отправка отзывов можно инъециоровать NoSQL-запрос с функцией sleep(), которая сделает всю БД недоступной |
| Elevation of Privilege | На потоке данных Авторизация можно дополнить запрос так, чтобы при регистрации стразу стать администратором |

## Таблица уязвимостей

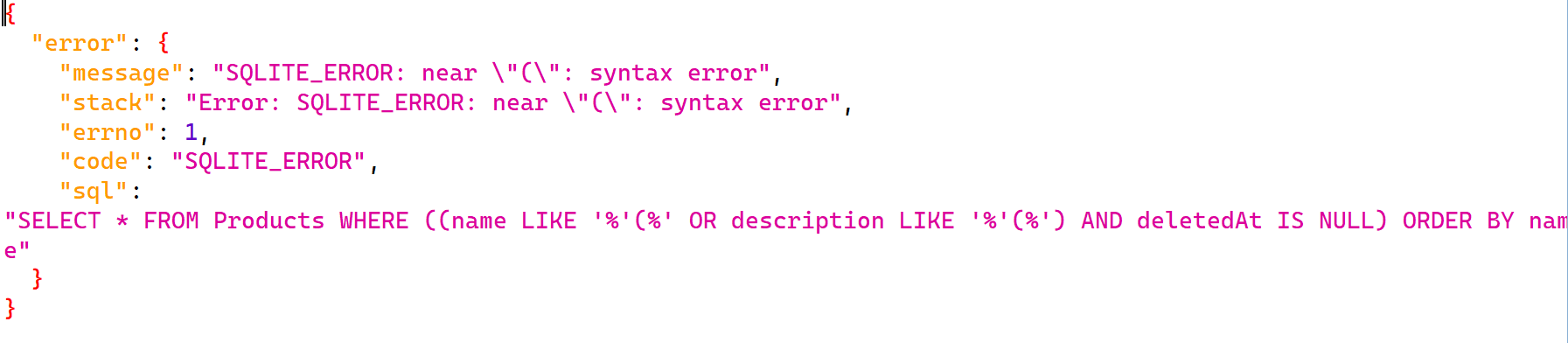
|  |  |
| --- | --- |
| Название | XSS-атака из-за отсутствия заголовков CSP |
| Описание | Можно встроить в страницу JS-код, который перешлёт злоумышленнику локальные данные пользователя |
| Уровень риска | 6.2 (Средний) |
| Категория OWASP Top 10 | A03:2021-Injection |
| Предложения по исправленияю | Нужно установить заголовок Content-Security-Policy:    Он заблокирует вызовы eval(), запретит динамическое выполнение скриптов |

Скриншот:



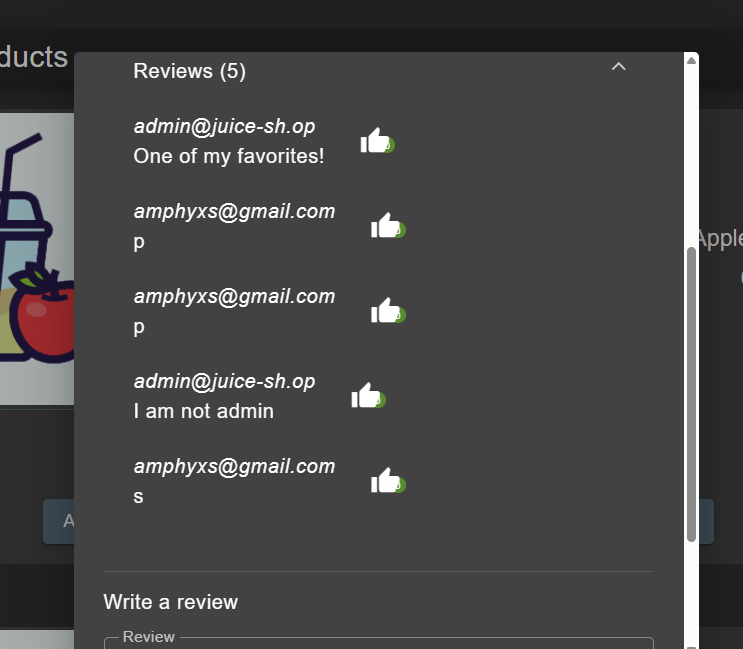
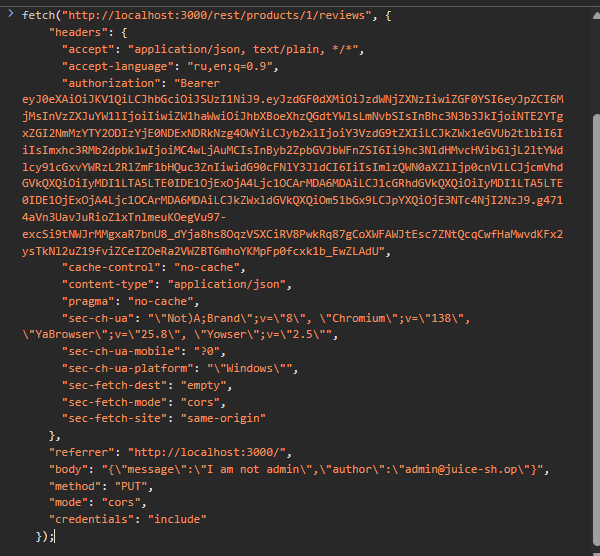
|  |  |
| --- | --- |
| Название | SQL-инъекция через запрос поиска |
| Описание | Можно изменить записи в таблице Products через SQL-инъекцию, которую нужно передать в query-параметре запроса:  /rest/products/search?q=<SQL-инъекция тут> |
| Уровень риска | 9,8 (Критический) |
| Категория OWASP Top 10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

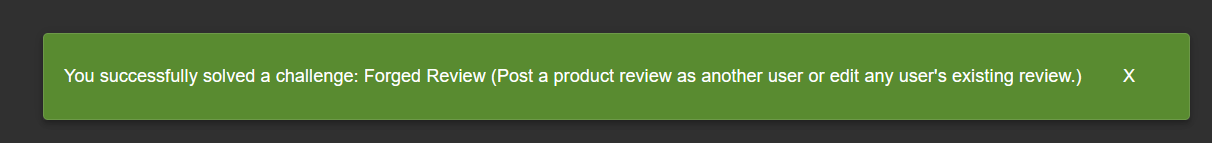
Скриншот:



|  |  |
| --- | --- |
| 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 |

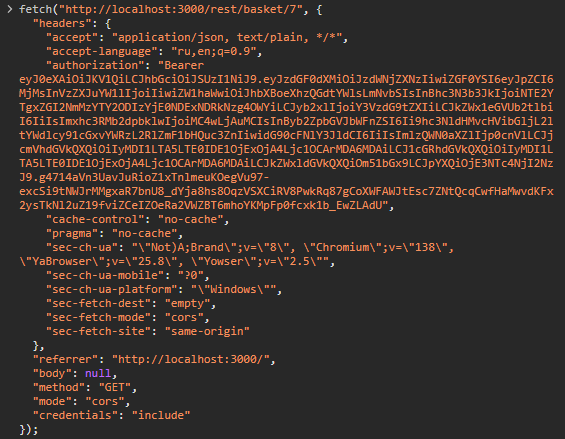
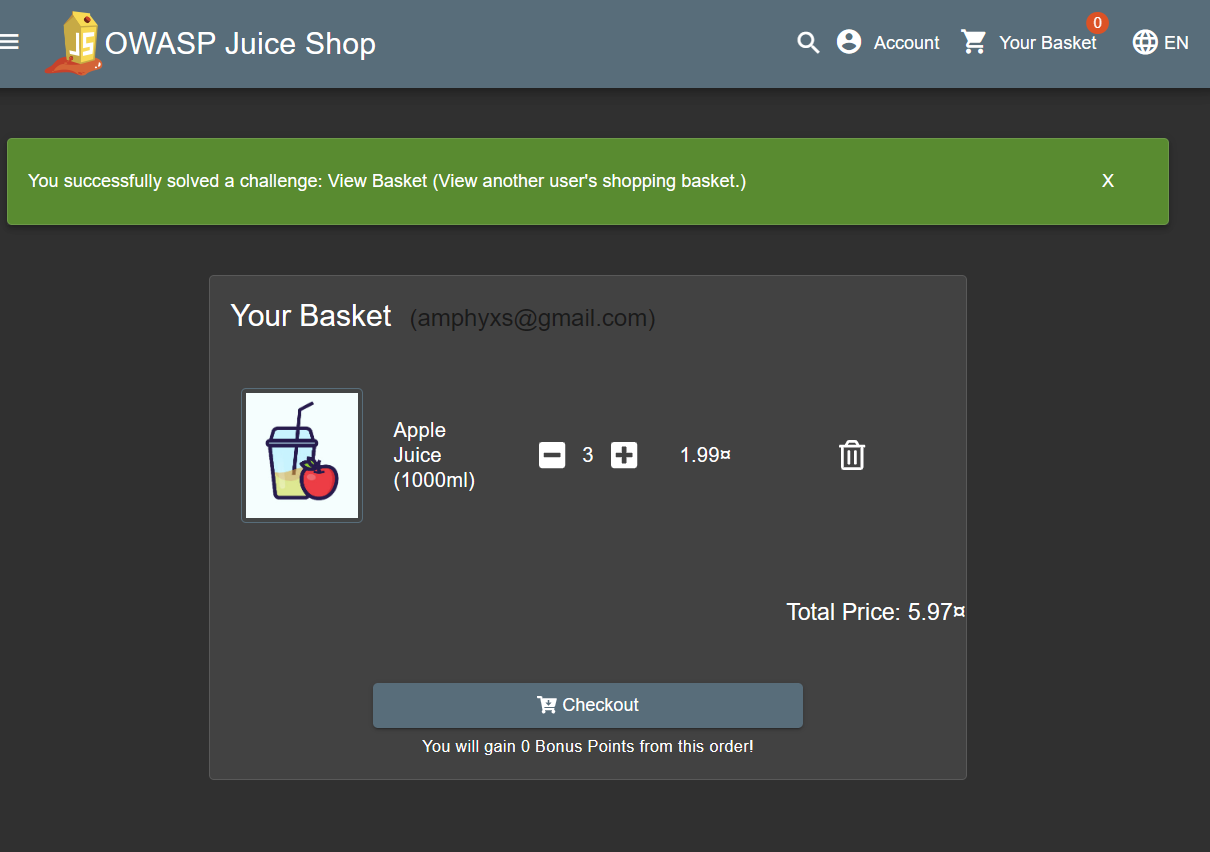
Скриншоты:



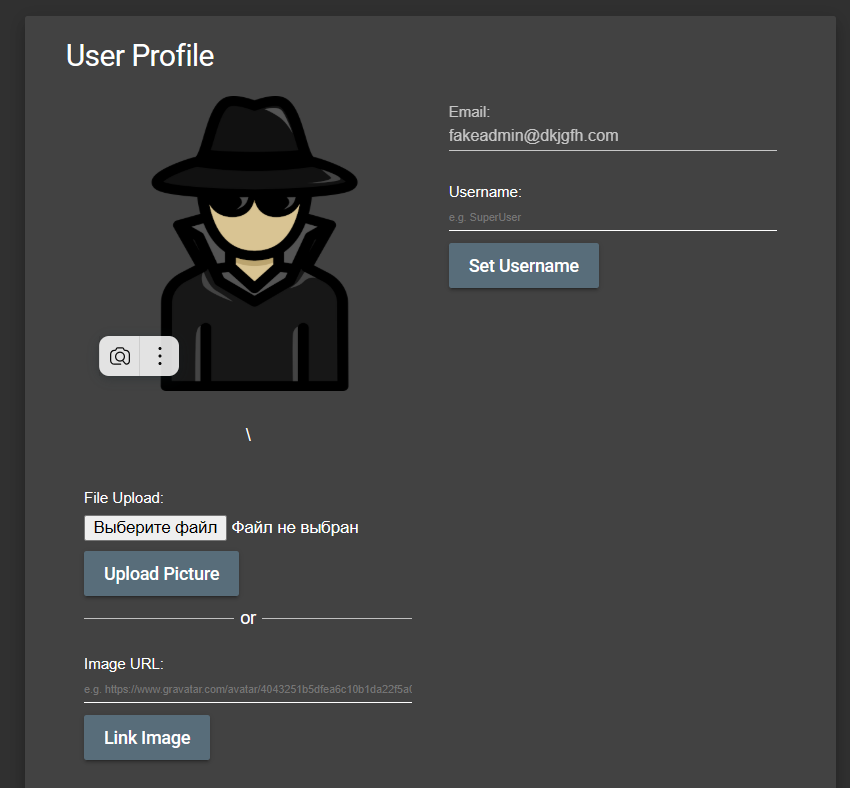
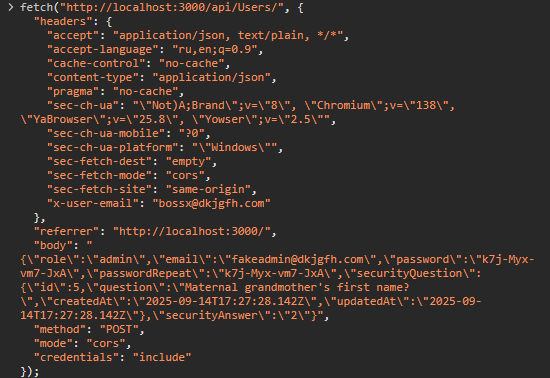
|  |  |
| --- | --- |
| Название | View Basket – отсутствие проверок владения сущностями |
| Описание | Эксплойт позволяет посмотреть чужую корзину.  Для этого надо в запросе GET /rest/basket/<id> использовать id искомой корзины |
| Уровень риска | 4.3 (Medium) |
| Категория OWASP Top 10 | A07:2021-Identification 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 |

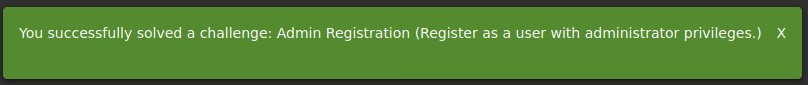
Скриншоты:



|  |  |
| --- | --- |
| 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:



## 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.pdf>