

Анализ атак межсайтовой подделки запросов (CSRF) и способов защиты*Болтовский Гавриил Александрович**Приамурский государственный университет им. Шолом-Алейхема**Студент***Аннотация**

В статье рассматриваются механизмы атак межсайтовой подделки запросов (CSRF) и их возможные последствия для веб-приложений. В качестве инструмента демонстрации используется OWASP Juice Shop, предоставляющий практические примеры эксплуатации уязвимостей. В результате проведенного анализа представлены уязвимости и рассмотрены основные методы защиты от CSRF-атак.

Ключевые слова: информационная безопасность, JavaScript, CSRF

Analyzing cross-site request forgery (CSRF) attacks and defenses*Boltovsky Gavriil Alexandrovich**Sholom-Aleichem Priamursky State University**Student***Abstract**

The paper discusses cross-site request forgery (CSRF) attack mechanisms and their possible consequences for web applications. OWASP Juice Shop is used as a demonstration tool, providing practical examples of vulnerability exploitation. As a result of the analysis, the vulnerabilities are presented and the main methods of defense against CSRF attacks are discussed.

Keywords: information security, JavaScript, CSRF

1 Введение**1.1 Актуальность исследования**

Атаки межсайтовой подделки запросов (CSRF) представляют серьезную угрозу информационной безопасности, так как позволяют злоумышленникам выполнять несанкционированные действия от имени пользователей без их ведома. Это может привести к компрометации учетных записей, несанкционированному изменению данных и выполнению критически важных операций, что ставит под угрозу конфиденциальность, целостность и доступность веб-ресурсов. Актуальность исследования обусловлена широкой распространенностью этой уязвимости и недостаточной осведомленностью разработчиков о механизмах ее эксплуатации и эффективных методах защиты. Изучение принципов работы атак CSRF и внедрение современных механизмов противодействия являются важными аспектами обеспечения безопасности веб-приложений.

1.2 Обзор исследований

В исследовании [1] рассматриваются вопросы тестирования веб-приложений с акцентом на важность сочетания инструментов автоматизированного анализа, таких как Acunetix, Invicti, Burp Suite Professional и OWASP ZAP, с методами ручного тестирования для повышения эффективности выявления уязвимостей, включая CSRF-атаки. А.Д. Чесноков в своей работе [2] анализирует различные аспекты информационной безопасности, выделяя ключевые направления обеспечения защиты данных, что имеет важное значение при разработке мер противодействия CSRF-угрозам. В исследовании «Detection of cross-site scripting (XSS) attacks using machine learning techniques: a review» [3] рассматриваются современные подходы к автоматизированному обнаружению атак, основанные на методах машинного обучения, которые могут быть адаптированы для выявления CSRF-атак. Вопрос эффективности механизмов защиты от CSRF-угроз в современных веб-фреймворках анализируется в работе «Where we stand (or fall): An analysis of CSRF defenses in web frameworks» [4], где проводится оценка применяемых стратегий и их устойчивости к атакам. В исследовании А.В. Ерохина, А.А. Гусарова, О.С. Сучкова [5] подробно рассматривается природа CSRF-атак, приводятся практические примеры их реализации, а также анализируются основные методы защиты, такие как токены безопасности и политика одного источника.

1.3 Цель исследования

Целью данного исследования – рассмотреть механизмы атак межсайтовой подделки запросов (CSRF) и их возможные последствия для веб-приложений.

1.4 Постановка задачи

Для достижения поставленной цели предполагается изучить теоретические основы атак межсайтовой подделки запросов, их классификацию и механизмы эксплуатации, а также выявить потенциальные векторы атак в целевом веб-приложении. В рамках исследования проводится анализ веб-приложения OWASP Juice Shop с целью выявления CSRF-уязвимостей, после чего разрабатываются сценарии атак, демонстрирующие практическую возможность эксплуатации обнаруженных недостатков безопасности.

2 Методы исследования

Методы исследования атак межсайтовой подделки запросов (CSRF) и методов защиты включают анализ механизмов эксплуатации данной уязвимости и оценку эффективности существующих мер противодействия. В рамках исследования проводится изучение принципов работы CSRF-атак, основанных на автоматической передаче учетных данных пользователя, таких как cookies или токены аутентификации, что позволяет злоумышленнику инициировать запросы от имени жертвы.

Практическое исследование осуществляется с использованием специализированного веб-приложения OWASP Juice Shop, которое предоставляет возможности для моделирования и анализа CSRF-атак в контролируемой среде. Данный инструмент позволяет изучить механизмы реализации атак, включая использование скрытых HTML-форм и вредоносных ссылок, а также протестировать различные подходы к защите, такие как применение токенов анти-CSRF, проверка источника запросов и ограничение доверенных доменов.

3 Результаты и обсуждения

OWASP Juice Shop предоставляет несколько способов развертывания, одним из которых является установка через пакетный менеджер npm. Этот метод позволяет быстро и удобно запустить приложение на локальной машине без необходимости использования дополнительных инструментов контейнеризации. Для установки OWASP Juice Shop через npm необходимо предварительно установить Node.js, git и пакетный менеджер npm. После установки необходимых компонентов процесс развертывания заключается в выполнении команд:

```
git clone https://github.com/juice-shop/juice-shop.git
cd juice-shop
npm install
npm start
```

Альтернативным методом развертывания является использование контейнеризации с помощью Docker.

Установленное приложение доступно в браузере по адресу <http://localhost:3000> (рис. 1).

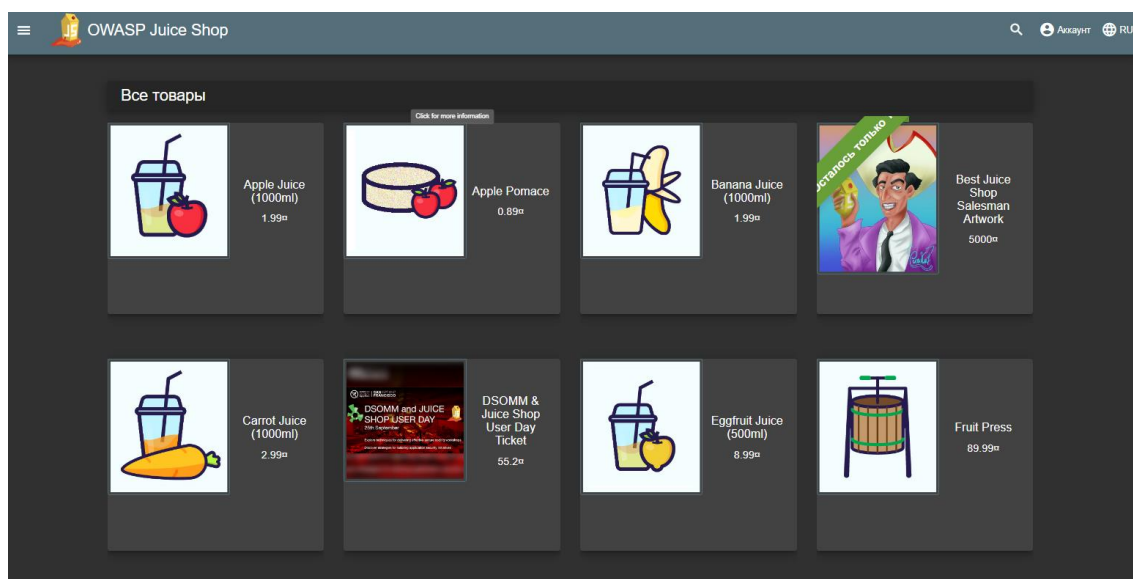


Рисунок 1 – Главная страница веб-приложения

Сценарий проведения CSRF атаки выглядит следующим образом: пользователь входит в свою учётную запись; злоумышленник отправляет пользователю ссылку на сторонний сайт с встроенным вредоносным кодом; пользователь переходит на сайт злоумышленника, на оригинальный сайт отправляется какой-либо запрос от имени пользователя, использующий активную сессию.

Для проведения атаки можно использовать формулу для смены имени пользователя (рис. 2).

Email:
admin@juice-sh.op

Username:
USER

Set Username

Рисунок 2 – Поле для смены имени пользователя

При нажатии на кнопку «Set Username». На сайте происходит отправка POST запроса, в заголовке которого содержатся куки авторизованного пользователя, а так же данные вида поле:значение.

Проанализировать запросы, узнать их содержимое можно через инструменты разработчика в браузере (вкладка «Networks»). При отправке новых данных, можно увидеть, что на сайте появляется новый запрос (рис. 3).

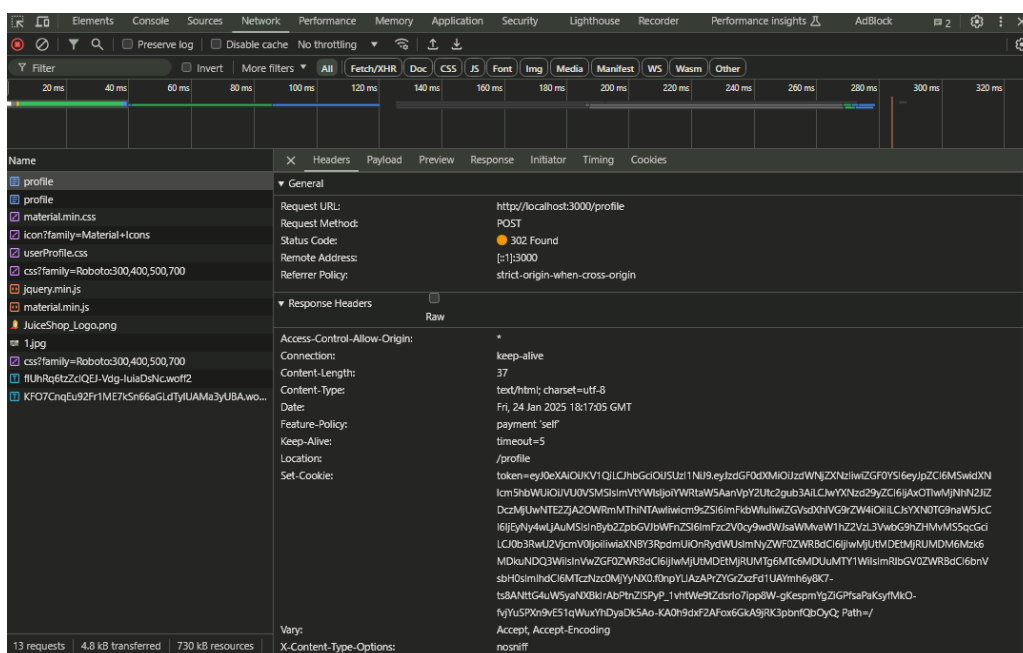


Рисунок 3 – Обзор запроса через инструменты разработчика в браузере

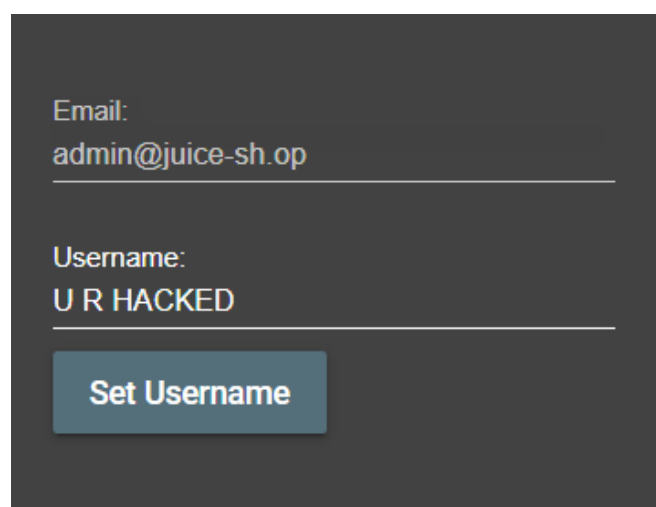
Теперь на стороннем веб-сервере размещается эксплойт. Ссылка на сервер передаётся пользователю с активной сессией. В результате перехода, у пользователя сменится имя на то, что задумал злоумышленник.

В рамках исследования, сервер-эксплойт был запущен локально с использованием Python. Код скрипта представлен на рисунке (рис. 4).

```
1 from http.server import HTTPServer, BaseHTTPRequestHandler
2
3 class CSRFHandler(BaseHTTPRequestHandler):
4     def do_GET(self):
5         # HTML-страница с CSRF-атакой
6         html = """
7         <!DOCTYPE html>
8         <html>
9             <head>
10                <title>CSRF Attack</title>
11            </head>
12            <body>
13                <form action="http://localhost:3000/profile" method="POST">
14                    <input type="hidden" name="username" value="U R HACKED">
15                    <input type="submit" value="Submit CSRF">
16                </form>
17                <script>
18                    document.forms[0].submit();
19                </script>
20            </body>
21        </html>
22        """
23        # Отправляем HTML как ответ
24        self.send_response(200)
25        self.send_header(keyword="Content-type", value="text/html")
26        self.end_headers()
27        self.wfile.write(html.encode("utf-8"))
28
29 # Запуск сервера
30 host = "localhost" # Адрес сервера
31 port = 8080 # Порт сервера
32 server = HTTPServer(server_address=(host, port), CSRFHandler)
33 print(f"Server running on http://{host}:{port}")
34 server.serve_forever()
35
```

Рисунок 4 – CSRF эксплойт, запускаемый локально

После перехода по адресу, имя пользователя на сайте меняется (рис. 5).



The screenshot shows a web interface with a dark background. At the top, it says "Email:" followed by "admin@juice-sh.op" in a light blue font. Below this is a horizontal line. Then it says "Username:" followed by "U R HACKED" in a light blue font. Below this is another horizontal line. At the bottom, there is a blue button with the text "Set Username" in white.

Рисунок 5 – Имя пользователя изменено

Защита от атак типа CSRF (Cross-Site Request Forgery) представляет собой комплекс мер, направленных на предотвращение несанкционированных действий от имени пользователя на веб-сайте. Одним из ключевых методов является внедрение уникальных токенов, привязанных к каждому сеансу или запросу, что позволяет серверу удостовериться в легитимности источника данных. Важную роль играет настройка заголовков политики безопасности, таких как SameSite для файлов cookie, ограничивающих передачу данных только в рамках одного домена. Дополнительную защиту обеспечивают строгие проверки источника запроса, анализ реферера и происхождения. Комбинированный подход, включающий многослойные механизмы защиты, позволяет минимизировать риски эксплуатации уязвимости CSRF и обеспечить высокий уровень безопасности веб-приложений.

3. Выводы

Таким образом были продемонстрированы XSS и CSRF атаки на веб-приложение.

Библиографический список

1. Riegler M., Sametinger J. Mode switching for secure web applications—a juice shop case scenario //Database and Expert Systems Applications-DEXA 2021 Workshops: BIODDD, IWCFS, MLKgraphs, AI-CARES, ProTime, AISys 2021, Virtual Event, September 27–30, 2021, Proceedings 32. Springer International Publishing, 2021. С. 3-8.
2. Чесноков А. Д. Информационная безопасность // StudNet. 2022. Т. 5. №. 1. С. 478-489.
3. Kaur J., Garg U., Bathla G. Detection of cross-site scripting (XSS) attacks using machine learning techniques: a review //Artificial Intelligence Review. 2023. Т. 56. №. 11. С. 12725-12769.
4. Likaj X., Khodayari S., Pellegrino G. Where we stand (or fall): An analysis of CSRF defenses in web frameworks //Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses. 2021. С. 370-385.
5. Ерохин А. В., Гусаров А. А., Сучков О. С. Межсайтовая подделка запроса CSRF // Роль и место информационных технологий в современной науке. 2017. С. 46-49.
6. OWASP Juice Shop. URL: <https://owasp.org/www-project-juice-shop/> (дата обращения: 25.01.2025).