



Факультет программной инженерии и компьютерной техники

Лабораторная работа №4

«Анализ уязвимостей веб-приложения с помощью OWASP ZAP»
по дисциплине «Информационная безопасность»

Выполнил:

Студент группы Р3432
Чмуррова М.В.

Преподаватель:

Рыбаков Степан Дмитриевич

Санкт-Петербург

2025

Задание

1. Установите OWASP ZAP (бесплатный инструмент).
2. Запустите встроенный браузер ZAP и перейдите на тестовый сайт (например, <http://testphp.vulnweb.com/>).
3. Проведите "Быстрое сканирование" (Quick Scan) сайта.
4. Проанализируйте результаты сканирования: найдите 3-5 различных типов уязвимостей (например, XSS, SQLi).
5. Сделайте скриншоты найденных уязвимостей и кратко опишите суть каждой.

Выполнение

Для выполнения был скачан OWASP ZAP на компьютер MacOS.

«Быстрое сканирование» проводилось на тестовом сайте <http://testphp.vulnweb.com> и с использованием браузера Chrome.

При открытии приложения открывается начальная страница:

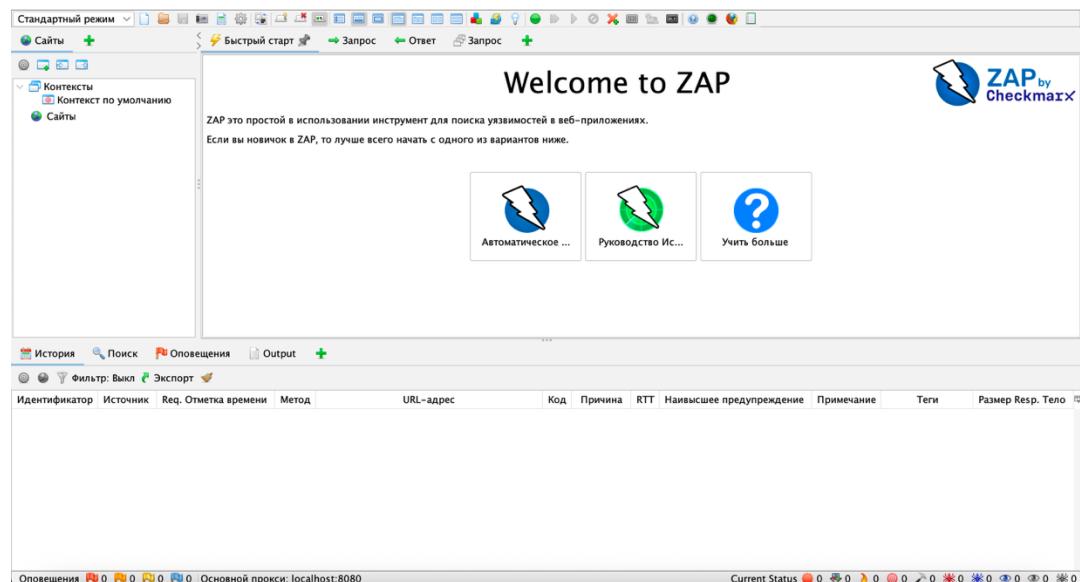


Рисунок 1. Начальный экран

Для дальнейшей работы выбирается вкладка «Быстрый старт» и выбирается режим «Автоматическое сканирование вашего приложения», которое выглядит следующим образом:

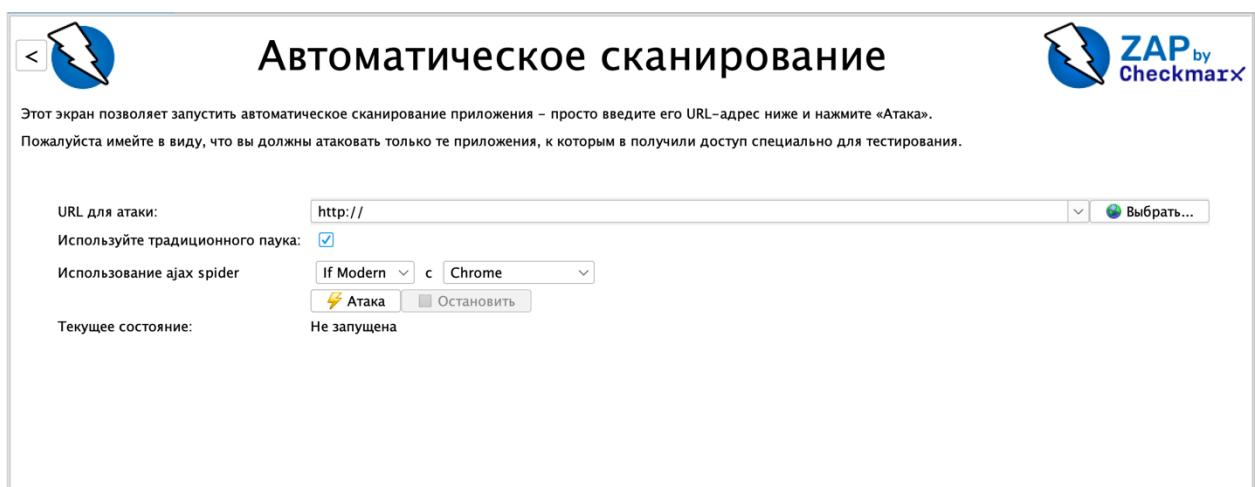


Рисунок 2. Автоматическое сканирование

Автоматическое сканирование имеет следующие поля для указания:

- URL для атаки: URL-адрес сайта, который необходимо проверить на уязвимости
- Используйте традиционного паука: использовать ли обычного паука. Паук – механизм обхода сайта. Он автоматически переходит по ссылкам, формам, редиректам, чтобы собрать все возможные URL и точки входа на сайт, которые потом будут проверяться на уязвимости
- Использование ajax spider: предлагается использования более глубокого анализа структуры сайта с использованием браузера. Ajax паук с помощью встроенного браузера выполняет JavaScript и имитирует действия пользователя, чтобы найти динамически загружаемые страницы и запросы на сайте

После нажатия кнопки «Атака» начинается процесс анализа сайта: нахождение всех URL и отправление запросов на поиски уязвимостей.

Так, паук нашел 126 URL:

| Обработано | Метод | URI | Отметки |
|------------|-------|--|---------|
| 1 | GET | http://testphp.vulnweb.com/Flash/add.flv | |
| 2 | GET | http://testphp.vulnweb.com/listproducts.php?artist=3 | |
| 3 | GET | http://testphp.vulnweb.com/listproducts.php?artist=1 | |
| 4 | GET | http://testphp.vulnweb.com/product.php?pic=6 | |
| 5 | GET | http://testphp.vulnweb.com/showimage.php?file=./pictures/6.jpg | |
| 6 | GET | http://testphp.vulnweb.com/showimage.php?file=./pictures/6.jpg&size... | |
| 7 | POST | http://testphp.vulnweb.com/cart.php | |
| 8 | POST | http://testphp.vulnweb.com/cart.php | |
| 9 | POST | http://testphp.vulnweb.com/cart.php | |
| 10 | POST | http://testphp.vulnweb.com/cart.php | |
| 11 | POST | http://testphp.vulnweb.com/cart.php | |
| 12 | POST | http://testphp.vulnweb.com/cart.php | |
| 13 | POST | http://testphp.vulnweb.com/cart.php | |
| 14 | POST | http://testphp.vulnweb.com/cart.php | |
| 15 | POST | http://testphp.vulnweb.com/cart.php | |
| 16 | POST | http://testphp.vulnweb.com/cart.php | |
| 17 | POST | http://testphp.vulnweb.com/cart.php | |
| 18 | POST | http://testphp.vulnweb.com/cart.php | |
| 19 | POST | http://testphp.vulnweb.com/cart.php | |
| 20 | POST | http://testphp.vulnweb.com/cart.php | |
| 21 | POST | http://testphp.vulnweb.com/cart.php | |
| 22 | POST | http://testphp.vulnweb.com/cart.php | |
| 23 | POST | http://testphp.vulnweb.com/cart.php | |
| 24 | POST | http://testphp.vulnweb.com/cart.php | |
| 25 | POST | http://testphp.vulnweb.com/cart.php | |
| 26 | POST | http://testphp.vulnweb.com/cart.php | |
| 27 | POST | http://testphp.vulnweb.com/cart.php | |
| 28 | POST | http://testphp.vulnweb.com/cart.php | |
| 29 | POST | http://testphp.vulnweb.com/cart.php | |
| 30 | POST | http://testphp.vulnweb.com/cart.php | |
| 31 | POST | http://testphp.vulnweb.com/cart.php | |
| 32 | POST | http://testphp.vulnweb.com/cart.php | |
| 33 | POST | http://testphp.vulnweb.com/cart.php | |
| 34 | POST | http://testphp.vulnweb.com/cart.php | |
| 35 | POST | http://testphp.vulnweb.com/cart.php | |
| 36 | POST | http://testphp.vulnweb.com/cart.php | |
| 37 | POST | http://testphp.vulnweb.com/cart.php | |
| 38 | POST | http://testphp.vulnweb.com/cart.php | |
| 39 | POST | http://testphp.vulnweb.com/cart.php | |
| 40 | POST | http://testphp.vulnweb.com/cart.php | |
| 41 | POST | http://testphp.vulnweb.com/cart.php | |
| 42 | POST | http://testphp.vulnweb.com/cart.php | |
| 43 | POST | http://testphp.vulnweb.com/cart.php | |
| 44 | POST | http://testphp.vulnweb.com/cart.php | |
| 45 | POST | http://testphp.vulnweb.com/cart.php | |
| 46 | POST | http://testphp.vulnweb.com/cart.php | |
| 47 | POST | http://testphp.vulnweb.com/cart.php | |
| 48 | POST | http://testphp.vulnweb.com/cart.php | |
| 49 | POST | http://testphp.vulnweb.com/cart.php | |
| 50 | POST | http://testphp.vulnweb.com/cart.php | |
| 51 | POST | http://testphp.vulnweb.com/cart.php | |
| 52 | POST | http://testphp.vulnweb.com/cart.php | |
| 53 | POST | http://testphp.vulnweb.com/cart.php | |
| 54 | POST | http://testphp.vulnweb.com/cart.php | |
| 55 | POST | http://testphp.vulnweb.com/cart.php | |
| 56 | POST | http://testphp.vulnweb.com/cart.php | |
| 57 | POST | http://testphp.vulnweb.com/cart.php | |
| 58 | POST | http://testphp.vulnweb.com/cart.php | |
| 59 | POST | http://testphp.vulnweb.com/cart.php | |
| 60 | POST | http://testphp.vulnweb.com/cart.php | |
| 61 | POST | http://testphp.vulnweb.com/cart.php | |
| 62 | POST | http://testphp.vulnweb.com/cart.php | |
| 63 | POST | http://testphp.vulnweb.com/cart.php | |
| 64 | POST | http://testphp.vulnweb.com/cart.php | |
| 65 | POST | http://testphp.vulnweb.com/cart.php | |
| 66 | POST | http://testphp.vulnweb.com/cart.php | |
| 67 | POST | http://testphp.vulnweb.com/cart.php | |
| 68 | POST | http://testphp.vulnweb.com/cart.php | |
| 69 | POST | http://testphp.vulnweb.com/cart.php | |
| 70 | POST | http://testphp.vulnweb.com/cart.php | |
| 71 | POST | http://testphp.vulnweb.com/cart.php | |
| 72 | POST | http://testphp.vulnweb.com/cart.php | |
| 73 | POST | http://testphp.vulnweb.com/cart.php | |
| 74 | POST | http://testphp.vulnweb.com/cart.php | |
| 75 | POST | http://testphp.vulnweb.com/cart.php | |
| 76 | POST | http://testphp.vulnweb.com/cart.php | |
| 77 | POST | http://testphp.vulnweb.com/cart.php | |
| 78 | POST | http://testphp.vulnweb.com/cart.php | |
| 79 | POST | http://testphp.vulnweb.com/cart.php | |
| 80 | POST | http://testphp.vulnweb.com/cart.php | |
| 81 | POST | http://testphp.vulnweb.com/cart.php | |
| 82 | POST | http://testphp.vulnweb.com/cart.php | |
| 83 | POST | http://testphp.vulnweb.com/cart.php | |
| 84 | POST | http://testphp.vulnweb.com/cart.php | |
| 85 | POST | http://testphp.vulnweb.com/cart.php | |
| 86 | POST | http://testphp.vulnweb.com/cart.php | |
| 87 | POST | http://testphp.vulnweb.com/cart.php | |
| 88 | POST | http://testphp.vulnweb.com/cart.php | |
| 89 | POST | http://testphp.vulnweb.com/cart.php | |
| 90 | POST | http://testphp.vulnweb.com/cart.php | |
| 91 | POST | http://testphp.vulnweb.com/cart.php | |
| 92 | POST | http://testphp.vulnweb.com/cart.php | |
| 93 | POST | http://testphp.vulnweb.com/cart.php | |
| 94 | POST | http://testphp.vulnweb.com/cart.php | |
| 95 | POST | http://testphp.vulnweb.com/cart.php | |
| 96 | POST | http://testphp.vulnweb.com/cart.php | |
| 97 | POST | http://testphp.vulnweb.com/cart.php | |
| 98 | POST | http://testphp.vulnweb.com/cart.php | |
| 99 | POST | http://testphp.vulnweb.com/cart.php | |
| 100 | POST | http://testphp.vulnweb.com/cart.php | |
| 101 | POST | http://testphp.vulnweb.com/cart.php | |
| 102 | POST | http://testphp.vulnweb.com/cart.php | |
| 103 | POST | http://testphp.vulnweb.com/cart.php | |
| 104 | POST | http://testphp.vulnweb.com/cart.php | |
| 105 | POST | http://testphp.vulnweb.com/cart.php | |
| 106 | POST | http://testphp.vulnweb.com/cart.php | |
| 107 | POST | http://testphp.vulnweb.com/cart.php | |
| 108 | POST | http://testphp.vulnweb.com/cart.php | |
| 109 | POST | http://testphp.vulnweb.com/cart.php | |
| 110 | POST | http://testphp.vulnweb.com/cart.php | |
| 111 | POST | http://testphp.vulnweb.com/cart.php | |
| 112 | POST | http://testphp.vulnweb.com/cart.php | |
| 113 | POST | http://testphp.vulnweb.com/cart.php | |
| 114 | POST | http://testphp.vulnweb.com/cart.php | |
| 115 | POST | http://testphp.vulnweb.com/cart.php | |
| 116 | POST | http://testphp.vulnweb.com/cart.php | |
| 117 | POST | http://testphp.vulnweb.com/cart.php | |
| 118 | POST | http://testphp.vulnweb.com/cart.php | |
| 119 | POST | http://testphp.vulnweb.com/cart.php | |
| 120 | POST | http://testphp.vulnweb.com/cart.php | |
| 121 | POST | http://testphp.vulnweb.com/cart.php | |
| 122 | POST | http://testphp.vulnweb.com/cart.php | |
| 123 | POST | http://testphp.vulnweb.com/cart.php | |
| 124 | POST | http://testphp.vulnweb.com/cart.php | |
| 125 | POST | http://testphp.vulnweb.com/cart.php | |
| 126 | POST | http://testphp.vulnweb.com/cart.php | |

Рисунок 3. Демонстрация работы паука

А ajax паук нашел 2008 URL:

| Обработано | Идентификатор... | Req. Отметка врем... | Мет... | URL-адрес | Код состояния | Принцип | Размер Resp. | Заголовок | Размер Resp. | Темп... | Наивысшее предупрежде... | Примечани... | Теги |
|------------|----------------------|----------------------|--------|--|---------------|---------|--------------|--------------|--------------|---------|--------------------------|--------------|-------------------|
| 1 | Req. Отметка врем... | 1 | ... | http://testphp.vulnweb.com/ | 200 OK | ... | 222 байт | 4 720 ознат. | 4 720 ознат. | Средний | | | Form, Object, ... |
| 2 | ... | ... | ... | http://testphp.vulnweb.com/categories... | 200 OK | ... | 222 байт | 6 115 байт | 6 115 байт | Средний | | | Form, Object, ... |
| 3 | ... | ... | ... | http://testphp.vulnweb.com/product.ph... | 200 OK | ... | 222 байт | 6 368 байт | 6 368 байт | Средний | | | Form, Hidden, ... |
| 4 | ... | ... | ... | http://testphp.vulnweb.com/cart.ph... | 200 OK | ... | 222 байт | 4 903 байт | 4 903 байт | Средний | | | Form, Object, ... |
| 5 | ... | ... | ... | http://testphp.vulnweb.com/showimage... | 200 OK | ... | 208 байт | 3 324 байт | 3 324 байт | Низкий | | | |
| 6 | ... | ... | ... | http://testphp.vulnweb.com/ | 200 OK | ... | 222 байт | 4 958 байт | 4 958 байт | Средний | | | Form, Object, ... |
| 7 | ... | ... | ... | http://testphp.vulnweb.com/categories... | 200 OK | ... | 222 байт | 6 115 байт | 6 115 байт | Средний | | | Form, Object, ... |
| 8 | ... | ... | ... | http://testphp.vulnweb.com/categories... | 200 OK | ... | 222 байт | 6 115 байт | 6 115 байт | Средний | | | Form, Object, ... |
| 9 | ... | ... | ... | http://testphp.vulnweb.com/guestbook... | 200 OK | ... | 222 байт | 5 390 байт | 5 390 байт | Средний | | | Form, Hidden, ... |
| 10 | ... | ... | ... | http://testphp.vulnweb.com/categories... | 200 OK | ... | 222 байт | 6 115 байт | 6 115 байт | Средний | | | Form, Object, ... |
| 11 | ... | ... | ... | http://testphp.vulnweb.com/search.ph... | 200 OK | ... | 222 байт | 4 777 байт | 4 777 байт | Средний | | | Form, Object, ... |
| 12 | ... | ... | ... | https://content-autofill.googleapis.com... | 403 Forbidden | ... | 130 байт | 76 байт | 76 байт | | | | |
| 13 | ... | ... | ... | http://testphp.vulnweb.com/categories... | 200 OK | ... | 222 байт | 6 115 байт | 6 115 байт | Средний | | | Form, Object, ... |
| 14 | ... | ... | ... | http://testphp.vulnweb.com/categories... | 200 OK | ... | 222 байт | 6 115 байт | 6 115 байт | Средний | | | Form, Object, ... |
| 15 | ... | ... | ... | http://testphp.vulnweb.com/categories... | 200 OK | ... | 222 байт | 6 115 байт | 6 115 байт | Средний | | | Form, Object, ... |
| 16 | ... | ... | ... | ... | ... | ... | ... | ... | ... | | | | |

Рисунок 4. Демонстрация работы ajax-паука

После пассивного сканирования (паук + ajax паук) начинается активное сканирование, которое генерирует специальные запросы к сайту с целью

обнаружения уязвимостей. Оно уже более агрессивное и глубокое. Выглядит следующим образом:

| http://testphp.vulnweb.com Состояние сканирования | | | | | | |
|---|----------------------------|-----------|-----------|------|------------|--------|
| Состояние | Ответ диаграммы | | | | | |
| Хост: | http://testphp.vulnweb.com | | | | | |
| | Сила | Состояние | Прошло | Reqс | Оповещения | Статус |
| Анализ | | | 00:07.646 | 24 | | |
| Плагин | | | | | | |
| Обход Пути | Средний | | 00:53.541 | 666 | 0 | ✓ |
| Удаленное Включение Файлов | Средний | | 00:34.119 | 400 | 0 | ✓ |
| Source Code Disclosure - /WEB-INF Folder | Средний | | 00:00.748 | 3 | 0 | ✓ |
| Уязвимость Heartbleed OpenSSL | Средний | | 00:03.006 | 0 | 0 | ✓ |
| Раскрытие исходного кода - CVE-2012-18... | Средний | | 00:03.005 | 55 | 0 | ✓ |
| Удаленное выполнение кода - CVE-2012-... | Средний | | 00:04.068 | 174 | 0 | ✓ |
| Внешнее перенаправление | Средний | | 00:30.761 | 359 | 0 | ✓ |
| Серверная Сторона Включение | Средний | | 00:13.675 | 160 | 0 | ✓ |
| Межтаймовый скрипting (отражение) | Средний | | 00:10.310 | 137 | 20 | ✓ |
| Межтаймовый скрипting (постоянный) - Ос... | Средний | | 00:03.396 | 40 | 0 | ✓ |
| Межтаймовый Скрипting (Постоянный) - П... | Средний | | 00:02.056 | 87 | 0 | ✓ |
| Межтаймовый скрипting (постоянный) | Средний | | 00:00.028 | 0 | 0 | ✓ |
| SQL-инъекция | Средний | | 01:07.163 | 644 | 13 | ✓ |
| SQL Injection - MySQL (Time Based) | Средний | | 00:01.430 | 29 | 0 | ⬇ |
| SQL Injection - Hypersonic SQL (Time Based) | Средний | | | 0 | 0 | ⬇ |
| SQL Injection - Oracle (Time Based) | Средний | | | 0 | 0 | ⬇ |
| SQL Injection - PostgreSQL (Time Based) | Средний | | | 0 | 0 | ⬇ |
| SQL Injection - SQLite (Time Based) | Средний | | | 0 | 0 | ⬇ |
| Межтаймовый скрипting (на основе DOM) | Средний | | | 0 | 0 | ⬇ |
| SQL Injection - MsSQL (Time Based) | Средний | | | 0 | 0 | ⬇ |
| Log4Shell | Средний | | | 0 | 0 | ⬇ |
| Spring4Shell | Средний | | | 0 | 0 | ⬇ |
| Внедрение Кода на Стороне Сервера | Средний | | | 0 | 0 | ⬇ |
| Внедрение удаленных команд ОС | Средний | | | 0 | 0 | ⬇ |
| XPath Инъекция | Средний | | | 0 | 0 | ⬇ |
| Атака на внешний объект XML | Средний | | | 0 | 0 | ⬇ |
| Стандартный Oracle Padding | Средний | | | 0 | 0 | ⬇ |
| Потенциально открытые облачные метадан... | Средний | | | 0 | 0 | ⬇ |
| Внедрение шаблона на стороне сервера | Средний | | | 0 | 0 | ⬇ |
| Внедрение шаблона на стороне сервера (в... | Средний | | | 0 | 0 | ⬇ |
| Remote OS Command Injection (Time Based) | Следящий | | | 0 | 0 | ⬇ |

Рисунок 5. Активное сканирование

Все активное сканирование заняло ~13 минут.

По окончании работы программы были найдены следующие уязвимости:

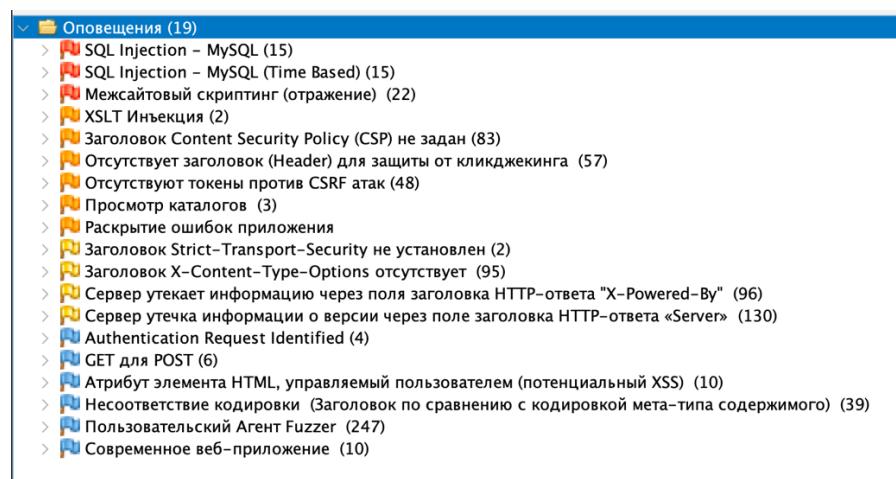


Рисунок 6. Найденные оповещения

Всего 19 оповещений. Из них:

- Высокоприоритетных: 3,
 - Среднеприоритетных: 6,

- Низкоприоритетных: 4,
- Информационных: 6

Выберем несколько найденных уязвимостей и проанализируем их, кликнув на уязвимость два раза для получения дополнительной информации

Анализ найденных уязвимостей

1. SQL Injection

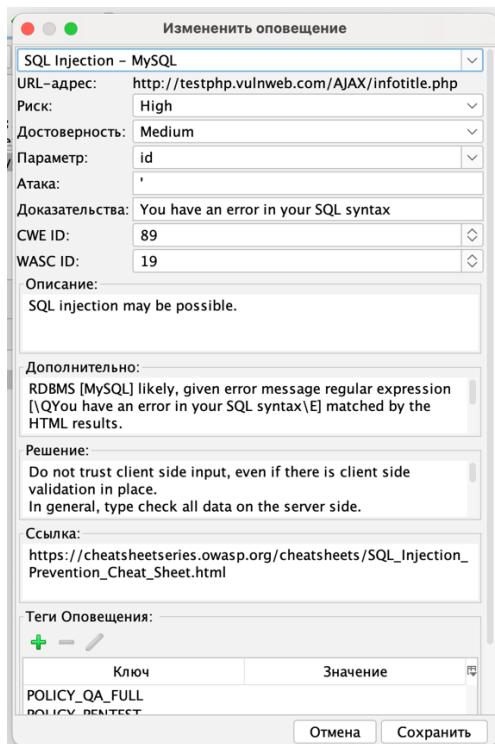


Рисунок 7. Уязвимость SQL Injection

SQLi (SQL injection) – уязвимость при которой злоумышленник может изменить или вставить запрос, чтобы получить неавторизованный доступ к данным или изменить их.

Конкретно в найденной ошибке получаем сообщение «You have an error in your SQL syntax» означает, что строка SQL, которую сервер пытается выполнить, стала синтаксически некорректной — обычно, потому что входные данные от пользователя попали внутрь SQL-строки напрямую.

Для решения этой проблемы необходимо использовать параметризованные запросы (prepared statements) и серверную валидацию/приведение типов — ни в коем случае не конкатенировать пользовательский ввод в SQL.

Кроме того, были найдены time-based SQL инъекции - разновидность слепой SQL-инъекции, при которой атакующий не получает данные напрямую, но заставляет базу данных выполнять задержку, и по времени

ответа извлекает информацию: задержка = «1» (истина), отсутствие задержки = «0» (ложь). За счет этого появляется возможность извлечь данные, даже если приложение не возвращает ошибки/результаты

В моем сканировании было найдено 15 различных time-based MySQL Injection уязвимостей:



Рисунок 8. Time-based уязвимости

2. Межсайтовый скрипting (отражение)

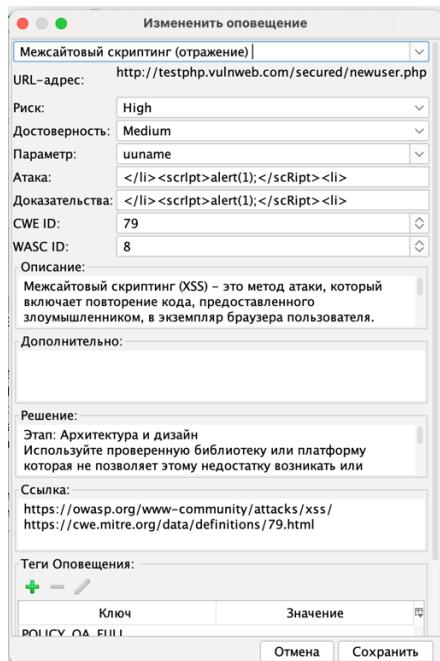


Рисунок 9. Межсайтовый скрипting (отражение)

XSS – вид атаки при которой сайт отображает на странице данные, введённые пользователем, без проверки и экранирования, из-за чего злоумышленник может вставить свой JavaScript-код, который выполнится в браузере других пользователей.

На скриншоте видно, что ZAP протестировал параметр uuname (например, поле формы) и вставил туда вот такой тестовый скрипт: <scrIpt>alert(1);</scRipt>. Сервер вернул этот код обратно в ответ, и браузер его выполнил. Это означает, что сайт уязвим к отражённому XSS (Reflected XSS)

Для устранения данной уязвимости необходимо всегда экранировать/санитизировать данные перед вставкой в HTML (включая атрибуты и контент), применять серверную валидацию/white-list, использовать безопасные шаблонизаторы или фреймворки (которые делают автоэкранирование), установить Content Security Policy и флаги куки (HttpOnly, Secure, SameSite)

При переходе на URL найденный в одной из XSS уязвимости видно, что вставленный скрипт действительно вызывает alert(1):

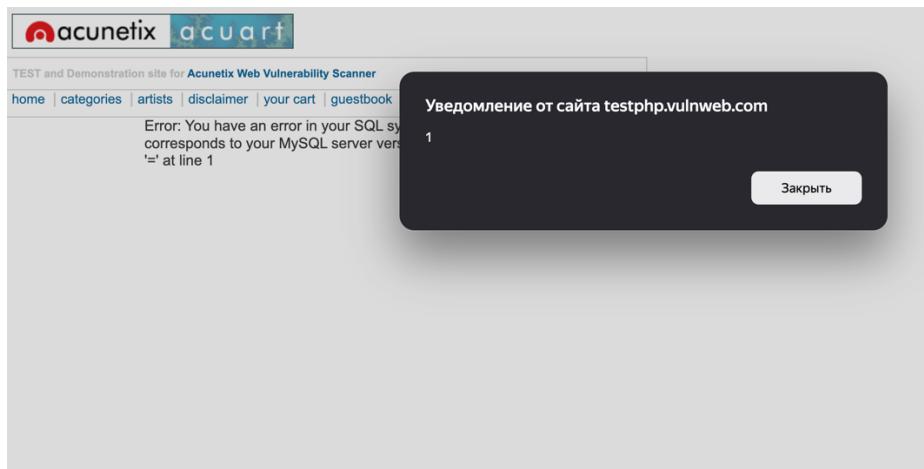


Рисунок 10. Выполнение alert(1) при XSS уязвимости

3. Заголовок Content Security Policy (CSP) не задан

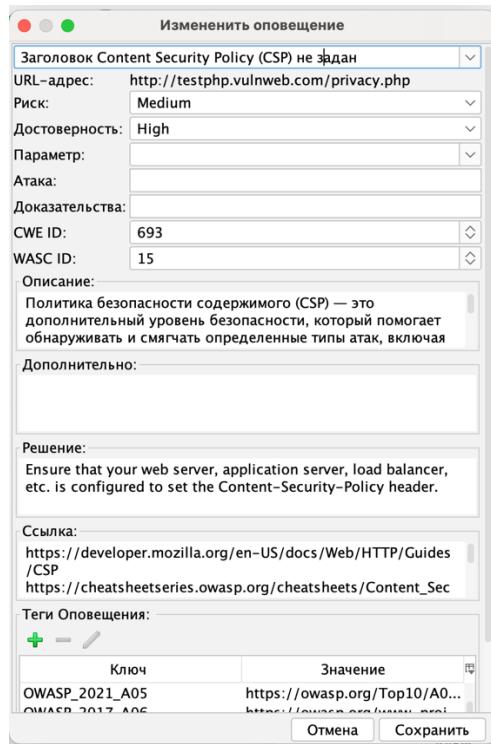


Рисунок 11. Заголовок CSP не задан

Отсутствие заголовка CSP означает отсутствие политики безопасности содержимого в HTTP-ответах сервера. Это уменьшает защиту от XSS, подмены скриптов и загрузки злонамеренных ресурсов.

Для решения этой проблемы необходимо установить HTTP-заголовок Content-Security-Policy (CSP) на веб-сервере или в приложении. Указать разрешённые источники для скриптов, стилей и контента, например:

```
Content-Security-Policy: default-src 'self'; script-src 'self';
```

Это ограничит выполнение внешних скриптов и снизит риск XSS и внедрения вредоносного кода.

4. Отсутствуют токены против CSRF атак

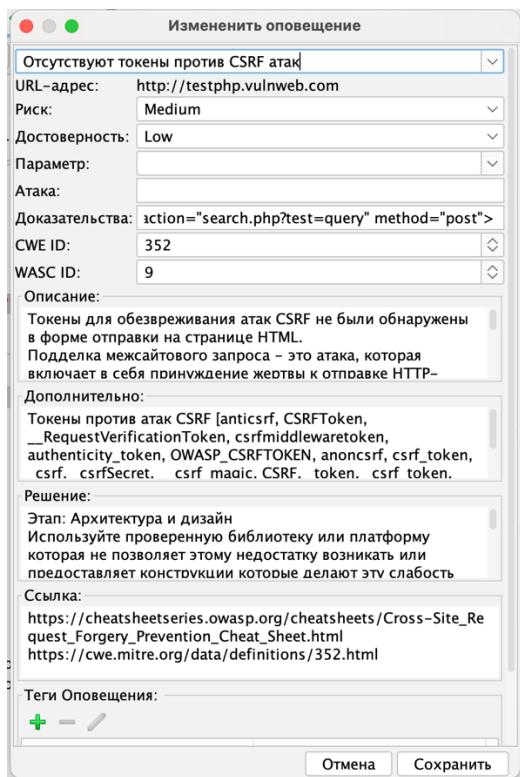


Рисунок 12. Уязвимость отсутствия токенов от CSRF атак

CSRF (Cross-Site Request Forgery) - атака, при которой злоумышленник заставляет браузер жертвы выполнить нежелательный запрос от имени пользователя на доверенный сайт, где он уже авторизован. Например, если пользователь вошёл в свой аккаунт, злоумышленник может с помощью поддельной формы заставить браузер отправить запрос на изменение пароля, перевод денег и т. д., без ведома пользователя.

ZAP обнаружил, что на странице (в форме search.php) отсутствует CSRF-токен — специальный уникальный параметр, который должен добавляться в каждую форму или запрос. Без этого токена сервер не отличит легитимный запрос пользователя от поддельного

Для решения этой проблемы необходимо добавить CSRF-токен в каждую форму. Например:

```
<input type="hidden" name="csrf_token" value="random_string">
```

Или использовать какие-либо встроенные механизмы фреймворков (аннотации). Кроме того, необходимо сначала обеспечить защиту от XSS-атак, так как, если этого не сделать, то csrf-токен может быть украден

Вывод

В ходе данной лабораторной работы было проведено сканирование с помощью OWASP ZAP сайта <http://testphp.vulnweb.com>. В процессе было найдено несколько уязвимостей различного уровня, а также 2 высокоприоритетные и 2 среднеприоритетные были использованы для подробного анализа. Наиболее опасными оказались: отражённый XSS (позволяет выполнить произвольный JS в браузере жертвы) и SQL-инъекция (позволяет модифицировать/читать данные БД)