МОЛДАВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ

ДЕПАРТАМЕНТ ИНФОРМАТИКИ

**Лабораторная работа № 5**

**по курсу Securitatea Sistemelor Informatice**

**Тема:**

**Инструменты сканирования уязвимостей OWASP-ZAP**

Выполнил: Bogdanov Iurii,

студент группы I2302

Проверила: L. Novac,

doctor conferențiar universitar

Кишинёв, 2024

***Введение:***

**Цель работы:**

1. Использовать инструменты OWASP для сканирования уязвимостей в веб-приложениях (2-3 веб-приложения).
2. Определить, какие уязвимости встречаются, и описать их.
3. Каковы методы решения тех проблем, которые вызваны определенными уязвимостями?
4. Определить другие приложения для сканирования уязвимостей для веб-приложений.

**Теоретическая часть:**

OWASP (Open Web Application Security Project) — это международное сообщество, объединяющее компании, учебные учреждения и энтузиастов в области разработки бесплатных инструментов и стандартов для повышения безопасности веб-приложений. Организация зарегистрирована как некоммерческая и благотворительная структура (501(c)(3) в США и аналог в Европе).

Среди самых известных проектов OWASP можно выделить руководство по безопасности и OWASP Top 10, которое фокусируется на самых критичных рисках для веб-приложений. Кроме того, популярностью пользуются учебные платформы и такие инструменты, как WebScarab Proxy для анализа трафика. В проекте задействовано около 190 локальных отделений и тысячи участников по всему миру.

OWASP ASVS (Application Security Verification Standard) — важный стандарт, способствующий унификации требований к безопасности приложений. Все проекты OWASP распределены по категориям, включающим аспекты безопасности, тестирования и жизненного цикла разработки ПО.

OWASP ZAP (Zed Attack Proxy) — один из самых востребованных open-source инструментов для сканирования веб-приложений, который может работать через прокси-сервер для анализа трафика, и поддерживает использование на различных платформах.

**Практическая часть:**

Для того чтобы начать сканирование веб-приложения на наличие уязвимостей с помощью OWASP ZAP, выполните следующие действия:

**Переход на вкладку запроса**:

* В этой вкладке вы сможете увидеть структуру HTTP-запроса с заголовками по умолчанию.
* Здесь можно выбрать один из методов запроса: **GET** (выбран по умолчанию), **POST**, **PUT**, **DELETE**, **HEAD**, **OPTIONS** и другие в зависимости от цели запроса.

**Составление и отправка HTTP-запроса**:

* Настройте HTTP-запрос в соответствии с потребностями анализа. После этого нажмите кнопку «Отправить».
* Приложение начнет искать уязвимости в веб-приложении, выполняя ручное сканирование. Этот метод подходит для анализа отдельных запросов.

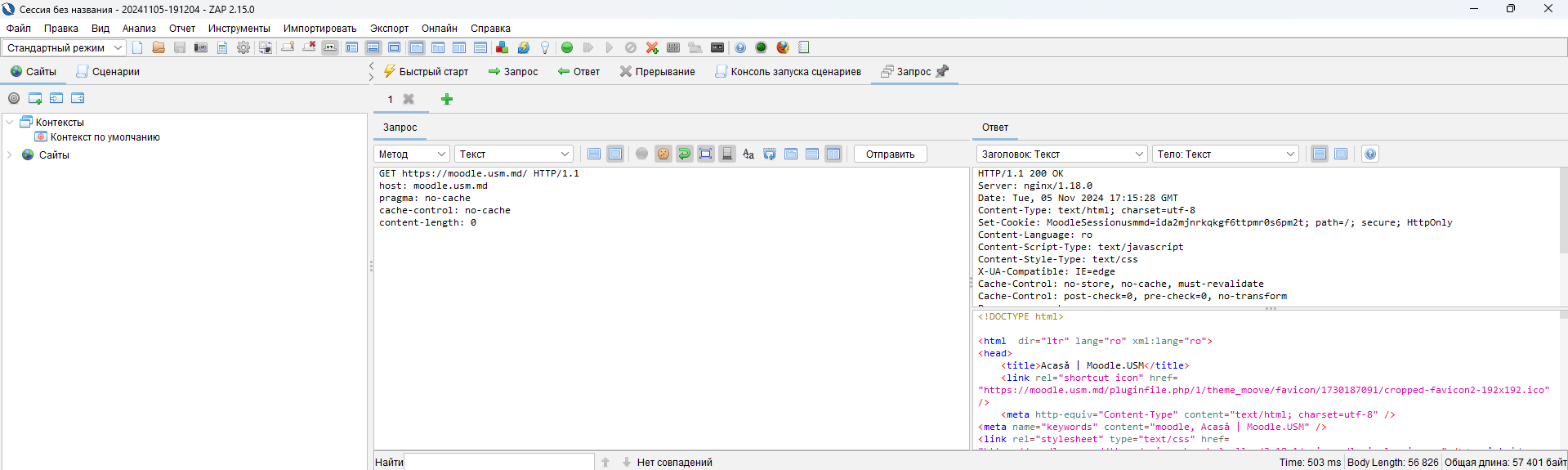
**Активное сканирование (автоматический режим)**:

* OWASP ZAP также предлагает «активное сканирование» — более автоматизированный подход к поиску уязвимостей. Для этого достаточно указать только базовый URL-адрес приложения.
* Инструмент самостоятельно найдет все доступные endpoints (точки взаимодействия с сервером), отправит к ним запросы и выполнит все необходимые проверки.
* Этот метод позволяет провести более масштабное и полное тестирование веб-приложения на наличие потенциальных уязвимостей, автоматизируя процесс поиска.

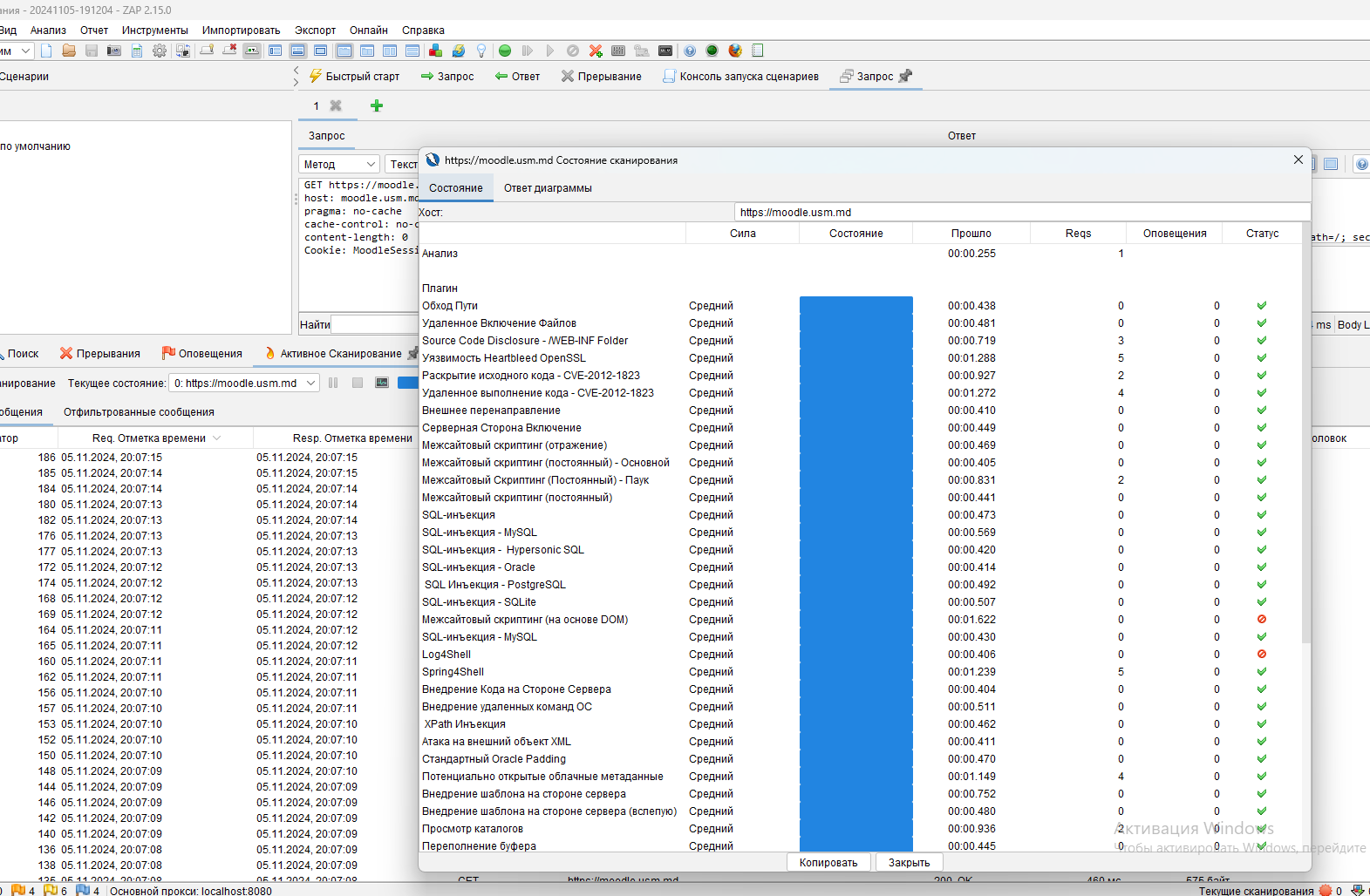
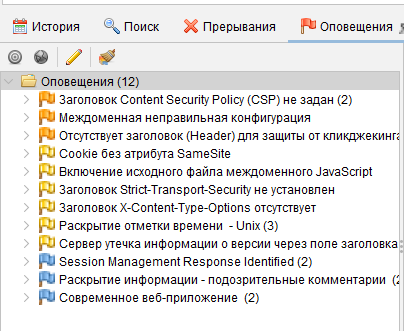
**Важно помнить**:

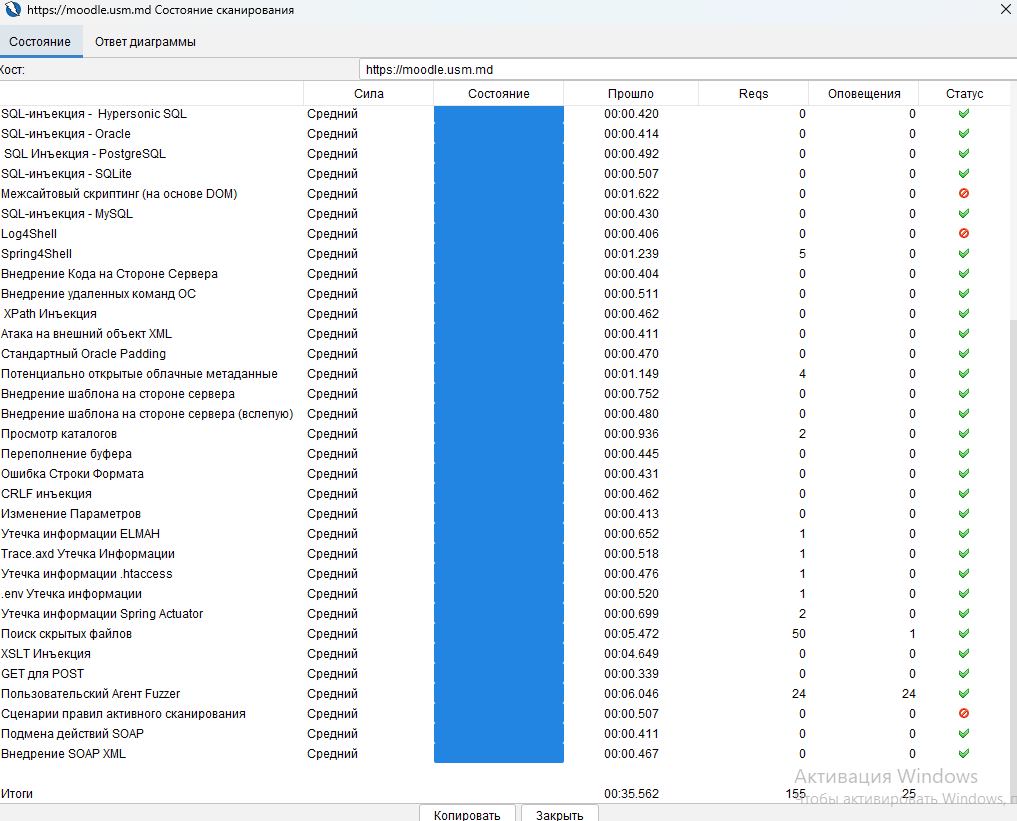
* Активное сканирование требует внимательной настройки. Поскольку оно генерирует большой объем запросов, неправильная конфигурация может вызвать нежелательные последствия для целевого веб-приложения, вплоть до возможного перегрузки сервера.

**Проверка веб-приложения moodle:**Создадим запрос на проверку веб-приложения moodle.

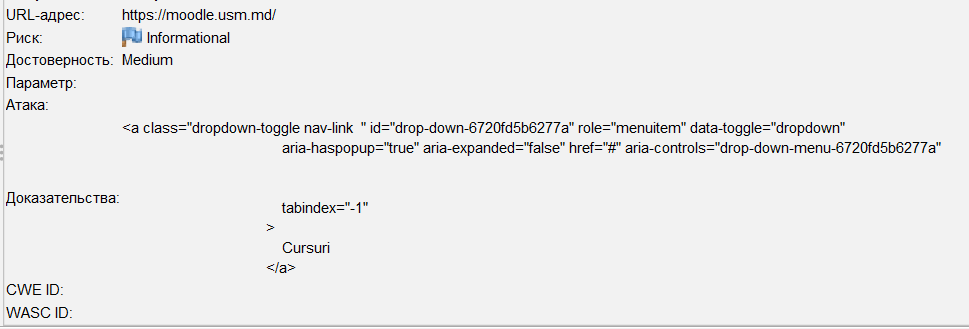
****

Запрос отправляется на веб-приложение moodle, и результаты появляются почти мгновенно. После его отправки приходит подтверждение успешного выполнения, а в теле ответа возвращается HTML-разметка основной страницы платформы moodle.

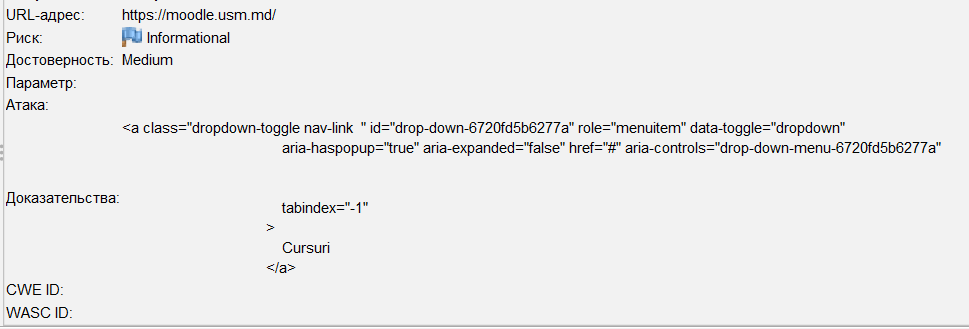
Когда OWASP ZAP завершает сканирование, он отображает результаты на вкладке под названием «Оповещения».



Результаты активного сканирования сайта Moodle:



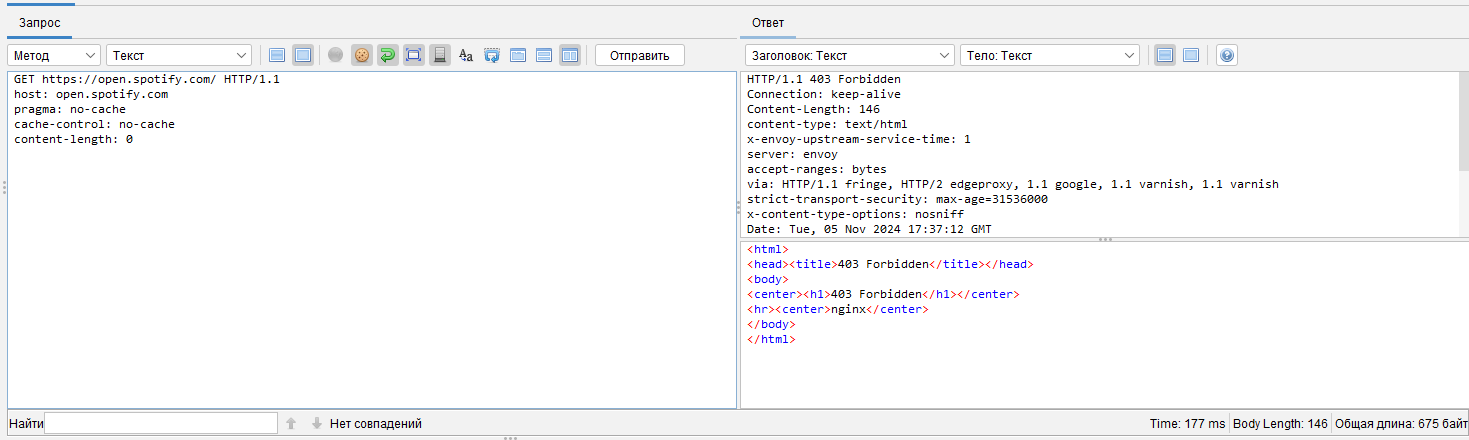
Далее мы получаем оповещение в результате обращение к URL сайта moodle. Здесь мы видим, что сайт имеет достоверность Medium.



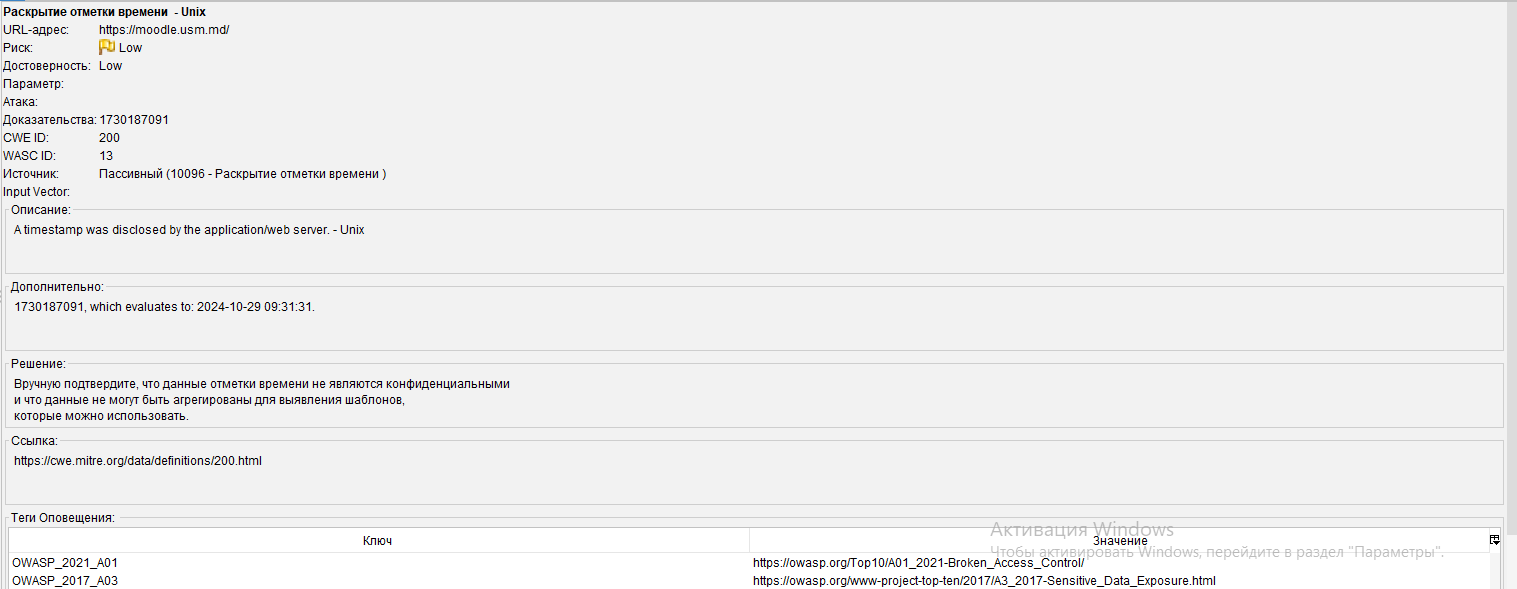
**Веб-приложение Spotify:**

Spotify — это популярное веб-приложение для стриминга музыки, которое позволяет пользователям слушать миллионы треков, создавать плейлисты, открывать новые релизы и персонализированные рекомендации. Приложение работает как в браузере, так и через мобильные и десктопные приложения. Spotify использует гибридную модель: бесплатный доступ с рекламой и премиум-подписки для прослушивания без рекламы, офлайн-доступа и улучшенного качества звука.

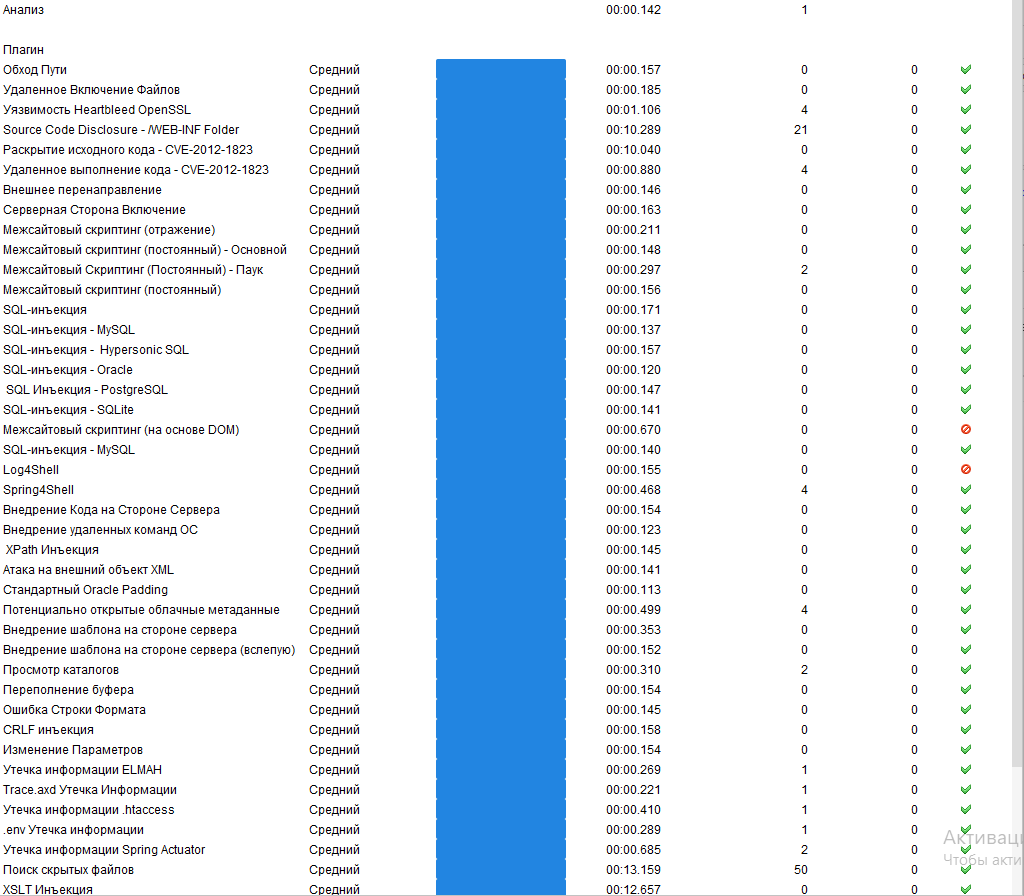
Давайте создадим запрос на данный сайт.



Оповещение о раскрытии отметки времени.



И наконец результаты активного сканирования приложения Spotify.



***Встречающиеся уязвимости:***

**Кликджекинг** (Clickjacking) — это техника обмана пользователей в интернете, при которой злоумышленник может получить доступ к приватной информации или даже контролировать устройство пользователя, заманивая его на внешне безвредную страницу или внедряя вредоносный код в безопасную страницу.

**CSRF** (Cross-Site Request Forgery), также известная как XSRF или «подделка межсайтовых запросов», — это атака, использующая уязвимости протокола HTTP. Когда пользователь посещает сайт, созданный злоумышленником, с его устройства втайне отправляется запрос на другой сервер (например, на сервер платежной системы), инициирующий нежелательное действие, такое как перевод средств на счет злоумышленника.

***Методы решения уязвимостей:***

### 1. Защита от кликджекинга (Clickjacking)

**Content Security Policy (CSP)**: CSP помогает предотвратить внедрение вредоносного контента, ограничивая домены, с которых может загружаться контент на ваш сайт. Например, frame-ancestors 'self' в CSP запрещает другим сайтам внедрять ваш сайт через iframe.

**Заголовок X-Frame-Options**: Этот заголовок добавляется в HTTP-ответ сервера, и поддерживается большинством современных браузеров. Он может принимать значения:

* + DENY: Полностью запрещает загрузку страницы в iframe.
  + SAMEORIGIN: Разрешает загрузку страницы в iframe, только если запрос исходит с того же домена.
  + ALLOW-FROM uri: Позволяет загружать страницу в iframe только на указанном URI (не все браузеры поддерживают).

**JavaScript-защита**: Вы можете добавить JavaScript-код для проверки, отображается ли страница внутри iframe. Если обнаружено, что страница загружена в iframe, можно перенаправить пользователя на основной URL вашего сайта. Это дополнительный уровень защиты для браузеров, которые не поддерживают X-Frame-Options.

### 2. Защита от CSRF (Cross-Site Request Forgery)

**CSRF-токены**: Каждый запрос, особенно те, которые изменяют данные, должен содержать уникальный токен, генерируемый сервером. Токен добавляется в запрос (чаще всего в теле POST-запроса или в заголовке) и проверяется на сервере. Если токен отсутствует или неверен, сервер отклоняет запрос. Например, при отправке формы можно добавить скрытое поле с CSRF-токеном.

**Проверка заголовков Origin и Referer**: Сервер может проверять, с какого сайта пришел запрос, сверяя заголовок Origin или Referer с доверенными доменами. Если запрос пришел с неизвестного сайта, его можно отклонить. Этот метод полезен, но менее надежен, так как не всегда поддерживается.

**Аутентификация для критических операций**: Для чувствительных операций (например, смена пароля или перевод средств) можно потребовать дополнительную аутентификацию пользователя. Например, повторный ввод пароля или отправка кода подтверждения на телефон.

### 3. Защита от XSS (Cross-Site Scripting)

**Экранирование пользовательского ввода**: Перед отображением данных, введенных пользователем, убедитесь, что специальные символы (<, >, &, и т.д.) экранированы. Это предотвратит интерпретацию данных как HTML или JavaScript. Например, вместо <script> сервер должен отображать &lt;script&gt;.

**Content Security Policy (CSP)**: CSP может помочь ограничить выполнение JavaScript на страницах, блокируя скрипты с неразрешенных источников и предотвращая выполнение инъекций. Настройки CSP можно задать, чтобы блокировать выполнение JavaScript из неавторизованных источников.

**Фильтрация и валидация ввода**: Прежде чем использовать пользовательские данные, проверяйте их на соответствие нужному формату. Например, для числовых полей используйте только числа, для адресов электронной почты — только допустимые символы и т.д. Это уменьшает шансы на внедрение вредоносного кода.

### 4. Защита от SQL-инъекций

**Подготовленные выражения (Prepared Statements)**: Подготовленные выражения позволяют передавать параметры запроса отдельно от SQL-кода. Это предотвращает внедрение вредоносных данных, так как сервер будет воспринимать параметры только как значения, а не как часть SQL-команды. Например, вместо SELECT \* FROM users WHERE id = ' + userInput, используйте SELECT \* FROM users WHERE id = ?.

**Использование ORM (Object-Relational Mapping)**: ORM-библиотеки, такие как Hibernate, SQLAlchemy и другие, абстрагируют работу с базой данных, автоматически экранируя значения параметров. Это снижает риск SQL-инъекций, так как ORM не позволяет напрямую вставлять пользовательский ввод в SQL-код.

**Проверка и ограничение пользовательского ввода**: Ограничивайте ввод данных пользователями до ожидаемых значений. Например, если ожидается числовое значение, проверяйте, что введенные данные действительно являются числом. Для текстовых полей можно установить максимальную длину, чтобы предотвратить большие вставки вредоносного кода.

### 5. Общие меры безопасности

**Обновление ПО**: Всегда следите за обновлениями серверного ПО, фреймворков и библиотек, чтобы своевременно устранять уязвимости. Использование устаревших компонентов увеличивает риск взлома через известные уязвимости.

**Аутентификация и контроль доступа**: Внедряйте строгие механизмы аутентификации (например, двухфакторная аутентификация) и разграничение прав доступа для пользователей, чтобы минимизировать возможность несанкционированного доступа к конфиденциальным данным.

**Шифрование**: Для защиты данных в транзите используйте HTTPS, обеспечивая шифрование между клиентом и сервером. Конфиденциальные данные на сервере должны храниться в зашифрованном виде, чтобы даже при утечке злоумышленник не смог их использовать.

**Логирование и мониторинг**: Настройте логи для отслеживания подозрительных активностей и возможных атак. Инструменты мониторинга помогут выявлять аномалии, такие как повышенная активность или попытки обхода авторизации, и позволят быстро реагировать на угрозы.

**Варианты других приложений для сканирования уязвимостей**

### 1. ****Burp Suite****

**Описание**: Burp Suite — это комплексный инструмент для тестирования безопасности веб-приложений, разработанный компанией PortSwigger. Он предоставляет широкий набор возможностей для перехвата и анализа трафика, поиска уязвимостей и создания пользовательских атак. Burp Suite активно используется профессионалами по безопасности и является стандартом в тестировании на проникновение.

**Основные функции**:

* **Перехват трафика**: Burp Suite позволяет перехватывать и изменять HTTP/HTTPS-запросы и ответы между клиентом и сервером, что помогает анализировать и тестировать безопасность приложения.
* **Автоматическое сканирование**: Burp Suite Pro (платная версия) поддерживает автоматическое сканирование веб-приложений для поиска уязвимостей, таких как SQL-инъекции, XSS, CSRF и другие.
* **Intruder**: Это мощный инструмент для автоматизированного тестирования входных данных, позволяющий находить слабые места путем отправки различных комбинаций параметров.
* **Repeater**: Позволяет вручную отправлять повторяющиеся HTTP-запросы с изменением параметров, что удобно для ручного тестирования.
* **Extender**: Burp Suite поддерживает плагины, которые можно загрузить через BApp Store или создать собственные с помощью Burp Extender API. Это делает инструмент гибким и настраиваемым для специфических нужд.

**Преимущества**:

* Интуитивно понятный интерфейс и отличная документация, что делает его доступным даже для начинающих.
* Возможность тонкой настройки сканирования и тестирования.
* Интеграция с CI/CD для автоматизации тестирования на этапах разработки.

**Недостатки**:

* Бесплатная версия ограничена в возможностях и не поддерживает автоматическое сканирование.
* Стоимость Pro версии может быть высокой для небольших компаний и команд.

### ****Кому подходит****: Burp Suite подходит как для профессиональных тестировщиков безопасности, так и для новичков. Его бесплатная версия полезна для обучения и базового тестирования, а Pro версия предоставляет весь функционал для комплексного анализа. 2. ****Nessus****

**Описание**: Nessus — это мощный инструмент для сканирования уязвимостей, разработанный компанией Tenable. Он используется для проверки сетевых устройств, серверов и веб-приложений на наличие уязвимостей. Nessus помогает быстро выявлять потенциальные проблемы в инфраструктуре, предоставляет расширенные отчеты и рекомендации по устранению уязвимостей. Этот инструмент широко используется для регулярного сканирования сетей в корпоративной среде.

**Основные функции**:

* **База уязвимостей**: Nessus поддерживает обширную базу данных уязвимостей, которая регулярно обновляется, что позволяет находить уязвимости в операционных системах, приложениях и сетевых устройствах.
* **Сканирование конфигурации и соответствия стандартам**: Проверяет устройства на соответствие стандартам безопасности (например, CIS, NIST, HIPAA), что важно для выполнения требований соответствия.
* **Поддержка различных типов сканирования**: От сетевого сканирования до веб-приложений и облачной инфраструктуры. Nessus позволяет настраивать типы тестов и профили для различных нужд.
* **Отчеты и аналитика**: Nessus генерирует подробные отчеты, включая описание уязвимостей, их уровень риска и рекомендации по устранению. Возможна настройка отчетов под нужды компании.

**Преимущества**:

* Регулярные обновления базы уязвимостей и мощная аналитика.
* Простой в использовании интерфейс и гибкость настройки сканирования.
* Подходит для сканирования не только веб-приложений, но и всей сетевой инфраструктуры.

**Недостатки**:

* Бесплатная версия (Nessus Essentials) ограничена 16 IP-адресами для сканирования, что делает её непригодной для больших сетей.
* Полная версия Nessus Professional платная и может быть дорогостоящей для небольших компаний.

**Кому подходит**: Nessus оптимален для компаний, которые хотят регулярно проверять сеть и приложения на наличие уязвимостей. Особенно полезен для организаций, которым необходимо соответствие требованиям безопасности и регулярные отчеты о состоянии инфраструктуры.

**Вывод:**

В ходе лабораторной работы были изучены ключевые возможности OWASP ZAP для выявления уязвимостей в веб-приложениях. Сканирование показало, что такие уязвимости, как кликджекинг и CSRF, являются частыми угрозами для современных веб-сервисов. Использование OWASP ZAP позволило выявить типы уязвимостей, которые злоумышленники могут использовать для кражи данных или распространения вредоносного кода. Были предложены методы защиты, включая заголовки безопасности и фильтрацию ввода.

### Также были рассмотрены другие инструменты, такие как ****Burp Suite**** и ****Nessus****, полезные для комплексной безопасности. Данная работа подчеркивает важность использования инструментов сканирования и превентивных мер для снижения рисков и защиты веб-приложений.

**Библиография:**

* <https://www.zaproxy.org/download/>
* <https://owasp.org/Top10/>
* <https://portswigger.net/burp>
* <https://www.tenable.com/products/nessus>
* [https://owasp.org/](https://owasp.org/" \t "_new)