|  |  |  |
| --- | --- | --- |
|  |  | УТВЕРЖДАЮ |
|  |  | Руководитель образовательной программы  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сергеева Е. Г.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.  М.П. |

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

на разработку

«Система автоматизированного тестирования безопасности веб-приложений»

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО |  | СОГЛАСОВАНО |
| Колледж ВятГУ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Вершинина Н. Д.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.  М.П. |  | Руководитель УП  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Долженкова М. Л.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

2025

Содержание

[1 Общие положение 1](#_Toc182717269)

[1.1 Полное наименование системы и ее условное обозначение 1](#_Toc182717270)

[1.2 Наименование организации-заказчика и организаций-участников работ 1](#_Toc182717271)

[1.3 Плановые сроки начала и окончания работы по создания системы 1](#_Toc182717272)

[1.4 Порядок оформления и предъявления заказчику результатов работ по созданию системы 1](#_Toc182717273)

[1.5 Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ 1](#_Toc182717274)

[1.6 Определения, обозначения и сокращения 2](#_Toc182717275)

[2 Назначение и цели создания системы 3](#_Toc182717276)

[2.1 Назначение системы 3](#_Toc182717277)

[2.2 Цели создания системы 3](#_Toc182717278)

[3 Характеристика объектов автоматизации 4](#_Toc182717279)

[3.1 Процессы, подлежащие автоматизации 4](#_Toc182717283)

[3.2 Пользователи, выполняющие процессы 4](#_Toc182717284)

[3.3 Существующее программное обеспечение 5](#_Toc182717285)

[3.4 Существующее техническое обеспечение 6](#_Toc182717286)

[3.5 Существующее нормативно-правовое обеспечение 6](#_Toc182717287)

[4. Описание предметной области 7](#_Toc182717288)

[Аналог 1: 8](#_Toc182717289)

[Аналог 2: 10](#_Toc182717290)

[Аналог 3: 12](#_Toc182717291)

[5 Требования к системе 14](#_Toc182717292)

[5.1 Требование к системе 14](#_Toc182717293)

[5.1.1 Требования к структуре и функционированию системы 14](#_Toc182717294)

[5.1.2 Показатели назначения 15](#_Toc182717295)

[**5.1** 15](#_Toc182717298)

[5.1.3 Требования к надежности 15](#_Toc182717299)

[5.1.4 Требования к безопасности 16](#_Toc182717300)

[5.1.5 Требования к защите информации от несанкционированного доступа 17](#_Toc182717301)

[5.1.6 Требования к патентной чистоте 17](#_Toc182717302)

[5.2 Требование к функциям (задачам), выполняемым системой 17](#_Toc182717303)

[5.3 Требования к видам обеспечения 20](#_Toc182717304)

[**5.3** 20](#_Toc182717307)

[5.3.1 Требования к математическому обеспечению 20](#_Toc182717308)

[5.3.2 Требование к информационному обеспечению 20](#_Toc182717309)

[5.3.3 Требования к лингвистическому обеспечению 20](#_Toc182717310)

[5.3.4 Требования к метрологическому обеспечению 20](#_Toc182717312)

[5.3.5 Требование к техническому обеспечению 20](#_Toc182717313)

[5.3.6 Требование к форматам хранения данных 21](#_Toc182717314)

[6 Состав и содержание работ по созданию системы 22](#_Toc182717315)

[7 Порядок контроля и приемки системы 24](#_Toc182717316)

[7.1 Виды испытаний 24](#_Toc182717317)

[7.2 Общие требования 24](#_Toc182717318)

[8 Требования с составу и содержанию работ по подготовке объекта автоматизации к вводу систмы в действие 25](#_Toc182717319)

[9 Требования к документированию 26](#_Toc182717320)

[10 Требования к приемо-сдаточным испытаниям 27](#_Toc182717321)

[10.1 Виды испытаний 27](#_Toc182717322)

[10.2 Общие требования 27](#_Toc182717323)

# Общие положение

## Полное наименование системы и ее условное обозначение

Полное название системы: «Система автоматизированного тестирования безопасности веб-приложений»

## Наименование организации-заказчика и организаций-участников работ

Заказчик: Колледж ВятГУ

Разработчик: Студент колледжа ВятГУ  
Вершинина Нелли Денисовна группы ИСПк-406-52-00

## Плановые сроки начала и окончания работы по создания системы

Разработка описываемого настоящим документом программного продукта должна быть осуществлена с 01.09.2025 по 01.12.2025.

## Порядок оформления и предъявления заказчику результатов работ по созданию системы

Система автоматизированного тестирования веб-безопасности передается Заказчику в виде готового программного обеспечения, размещенного на сервере разработчика или в облачной инфраструктуре.

Приемка системы осуществляется комиссией, состоящей из представителей Заказчика и Исполнителя. В ходе приемки проверяется соответствие реализованной функциональности требованиям настоящего ТЗ.

## Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ

При разработке Telegram-бота и создании проектно-эксплуатационной документации

Исполнитель руководствовался требованиями следующих нормативных документов:

* ГОСТ 19.201-78. ТЕХНИЧЕСКОЕ ЗАДАНИЕ. ТРЕБОВАНИЯ К СОДЕРЖАНИЮ И ОФОРМЛЕНИЮ;
* ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплексность и обозначение документов при создании автоматизированных систем;
* федеральный закон «Об информации, информационных технологиях и о защите информации» от 27.07.2006 N 149-ФЗ;
* ГОСТ 34.601-90. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.

## Определения, обозначения и сокращения

Настоящий документ содержит следующие аббревиатуры и определения:

* РоС — Proof of Concept;
* БД — База данных;
* ПК — Персональный компьютер;
* DAST – Dynamic Application Security Testing;
* SAST – Static Application Security Testing;
* API – Application Programming Interface;
* HTTP – Hypertext Transfer Protocol;
* CI/CD – Continuous Integration, Continuous Delivery;
* OWASP – Open Web Application Security Project;
* XSS – Cross-Site Scripting;
* SQL – Structured Query Language;
* CSRF – Cross-Site Request Forgery;
* LFI/RFI – Local/Remote File Inclusion;
* SSRF – Server-Site Request Forgery;
* GUI – Graphical User Interface;
* AJAX - Asynchronous Javascript and XML;
* ИБ – информационная безопасность;
* TLS - Transport Layer Security

# Назначение и цели создания системы

## Назначение системы

Система автоматизированного тестирования веб-безопасности предназначен для выявления уязвимостей в веб-приложениях, эмуляции атак и формирования отчётов с доказательствами эксплуатации.

Система обеспечивает автоматизированное проведение проверки, сводит результаты различных инструментов анализа и предоставляет разработчикам и специалистам по безопасности наглядные рекомендации по устранению проблем.

## Цели создания системы

Основными целями создания системы являются:

* минимизация ручного труда при поиске уязвимостей за счёт интеграции DAST, SAST, РоС;
* автоматическая проверка найденных проблем и сохранение доказательств (HTTP-запросы/ответы, скриншот браузера), что снижает число ложных срабатываний;
* предоставление интуитивного интерфейса (API и веб-панель) для запуска проверок и получения результатов;
* возможность подключения к CI/CD пайплайнам для проверок и получения результатов;
* генерация отчётов с классификацией и приоритизацией уязвимостей, что позволяет эффективно управлять устранением рисков;
* безопасное хранение конфиденциальных данных и строгое ограничение сканирования в рамках заданной области сканирования.

# Характеристика объектов автоматизации



## Процессы, подлежащие автоматизации

Процессы, подлежащие автоматизации, включают:

* сбор структуры веб-приложения: автоматический обход страниц и доступных точек входов;
* запуск и проведение атак: автоматизированная эмуляция наиболее распространенных атак с фиксацией признаков успешной эксплуатации;
* подтверждение уязвимостей: автоматическая проверка успешности атаки, сохранение скриншотов, HTTP-запросов/ответов и других артефактов;
* обработка запросов пользователей: обеспечение интерфейса для постановки задач сканирования, получения статусов и отчётов;
* агрегация и приоритизация результатов: автоматическое объединение найденных уязвимостей от разных модулей в единую базу с устранением дублей и классификацией по критичности;
* формирование отчётов: автоматическая генерация отчетов с указанием уязвимостей, РоС и рекомендаций по их устранению;
* интеграция с CI/CD: автоматический запуск сканирования при изменении кода или развертывании приложения и возврат статуса;
* хранение и доступ к артефактам: автоматизация процесса сохранения результатов сканирования и предоставление их пользователям.

## Пользователи, выполняющие процессы

Данные процессы выполняются следующими пользователями:

* специалист по информационный безопасности: запускают автоматизированные проверки приложений, анализируют отчёты, используют РоС для подтверждения уязвимостей;
* разработчики веб-приложений: получают отчёты с описанием уязвимостей и доказательствами эксплуатации, используют рекомендации для исправления кода;
* менеджеры проектов: просматривают сводные отчёты, контролируют устранение критических уязвимостей, используют метрики для управления качеством и безопасностью разработки;
* системы CI/CD: автоматически запускают процесс тестирования при обновлении кода, обеспечивая непрерывный контроль безопасности.

На рисунке 1 изображены процессы и пользователи, используемые в системе.

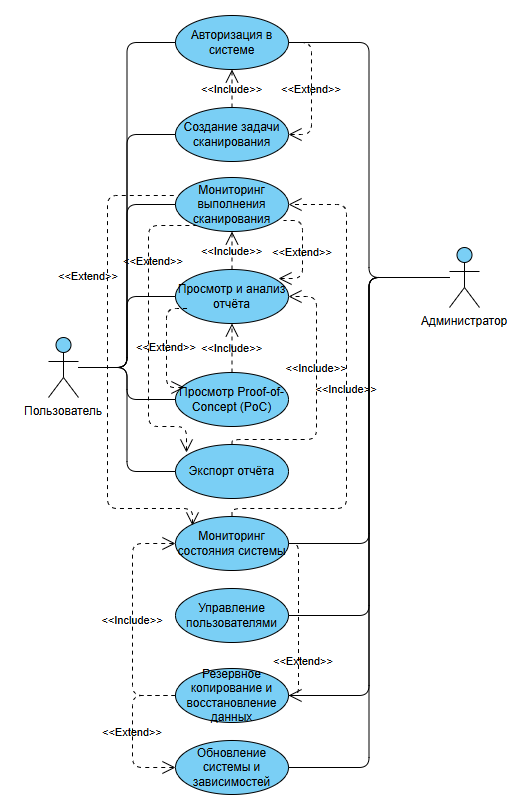


Рисунок 1 – Диаграмма Use Case

## Существующее программное обеспечение

В настоящее время для тестирования безопасности веб-приложений широко применяются инструменты, такие как OWASP ZAP, Nuclei, Burp Suite, Acunetix, Nessus. Однако большинство из них выполняют лишь отдельные типы анализов, например, только DAST или SAST, часто требуют высокой квалификации специалиста для правильного использования, не предоставляют в автоматическом режиме подтверждение уязвимости и сложны для интеграции в CI/CD без дополнительной настройки.

Разрабатываемая система решает эти проблемы за счёт объединения анализа и эмуляции атак в едином инструменте и формирования отчётов.

## Существующее техническое обеспечение

Для работы системы требуется:

* серверная инфраструктура или рабочая станция с установленными Docker;
* тестовые веб-приложения для проверки работы системы;
* интернет-соединение для загрузки шаблонов атак и обновлений инструментов анализа.

## Существующее нормативно-правовое обеспечение

К существующему нормативно-правовому обеспечению относятся:

* федеральный закон «Об информации, информационных технологиях и о защите информации» от 27.07.2006 N 149-ФЗ;
* федеральный закон «О персональных данных» от 27.07.2006 N 152-ФЗ;
* ГОСТ Р 56939-2016 «Защита информации. Методы и средства контроля защищённости информации. Общие положения»;
* стандарты OWASP и рекомендации по обеспечению безопасности веб-приложений.

Система обеспечит соответствие всем актуальным нормативным требованиям и стандартам по защите информации и персональных данных пользователей.

# Описание предметной области

Современные веб-приложения являются ключевым компонентом информационных систем в организациях, так как обеспечивают доступ пользователей к сервисам, базам данных и бизнес-процессам через интернет.

С развитием цифровизации возрастает количество атак на веб-приложения: злоумышленники стремятся использовать уязвимости в их коде или конфигурации для кражи персональных данных, финансовой информации, а также для получения несанкционированного доступа к инфраструктуре.

По данным аналитических отчётов наиболее распространёнными угрозами для веб-приложений являются:

* XSS: внедрение вредоносного JavaScript-кода в страницы приложения;
* SQL-инъекция: подмена SQL-запросов для получения или изменения данных в базе;
* CSRF: выполнение действий от имени пользователя без его ведома;
* LFI/RFI: чтение или выполнение файлов на сервере;
* SSRF: использование сервера жертвы для доступа к внутренним ресурсам;
* ошибки конфигурации: небезопасные заголовки, открытые API, неправильные права доступа.

Проверка защищённости веб-приложений в организациях проводится с помощью ручного тестирования или специализированных инструментов, однако:

* ручное тестирование требует высокой квалификации и занимает много времени;
* существующие сканеры выдают большое количество ложных срабатываний и не всегда предоставляют воспроизводимые доказательства уязвимостей;
* интеграция безопасности в процессы CI/CD разработки часто усложнена.

В этих условиях и возникает необходимость в системе автоматизированного тестирования веб-безопасности, которая позволит объединить в одном решении сбор краулинг, запуск атак, РоС, генерацию отчётов, что ускорить устранение проблем и повысит общий уровень безопасности веб-приложений.

## Аналог 1:

**OWASP ZAP.**

Это инструмент для тестирования безопасности веб-приложений. Он помогает разработчикам, тестироващикам и специалистам по безопасности оценивать уязвимости веб-приложений на этапе разработки и тестирования.

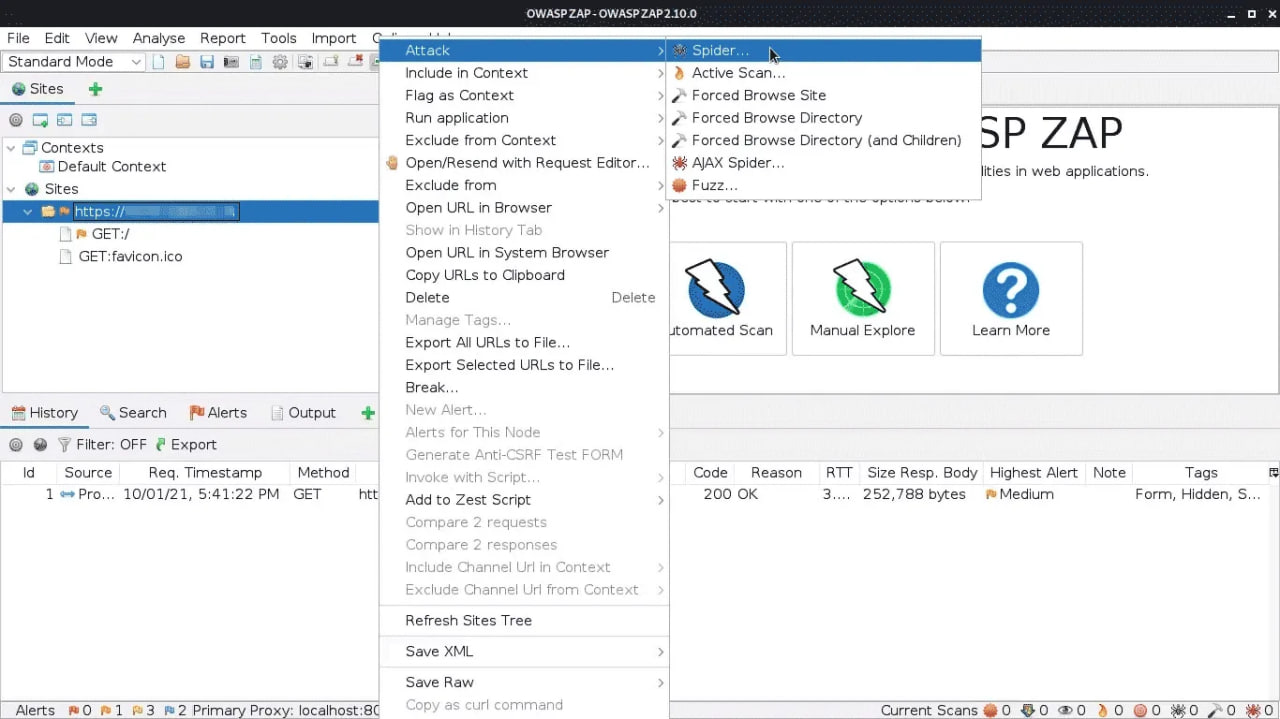


Рисунок 1 – Пример аналога 1

**Функции:**

* интерцепт и инспекция графиков: ZAP позволяет перехватывать и анализировать HTTP-трафик между браузером и целевым веб-приложением, что помогает идентифицировать потенциальные уязвимости;
* сканирование уязвимостей: инструмент осуществляет автоматическое сканирование на предмет распространённых уязвимостей;
* генерация отчётов о безопасности для документирования обнаруженных уязвимостей и предоставления информации для исправления;
* поддержка скриптов и API: ZAP предоставляет возможность разработки и использования скриптов для автоматизации тестирования и интеграции с другими инструментами через API.

**Особенности:**

* кроссплатформенность – поддержка всех основных ОС (Windows, Linus, MacOS);
* бесплатный проект с открытым исходным кодом;
* поддержка плагинов для расширения функционала;
* возможность работы через GUI и через интерфейс командой строки;
* обширный набор функций – от активного/пассивного сканирования до сканирования API и AJAX;
* подходит для начинающих специалистов в ИБ;

**Проблемы:**

* работает только как DAST, без SAST
* результат часто содержит много ложных срабатываний;
* нет удобной интеграции и сырые отчёты.

## Аналог 2:

**Nuclei.**

Это инструмент с открытым исходным кодом, который позволяет специалистам по ИБ проводить быстрые и эффективные сканирования на наличие уязвимостей в различных системах – от веб-приложений до сетевых устройств и облачных платформ.

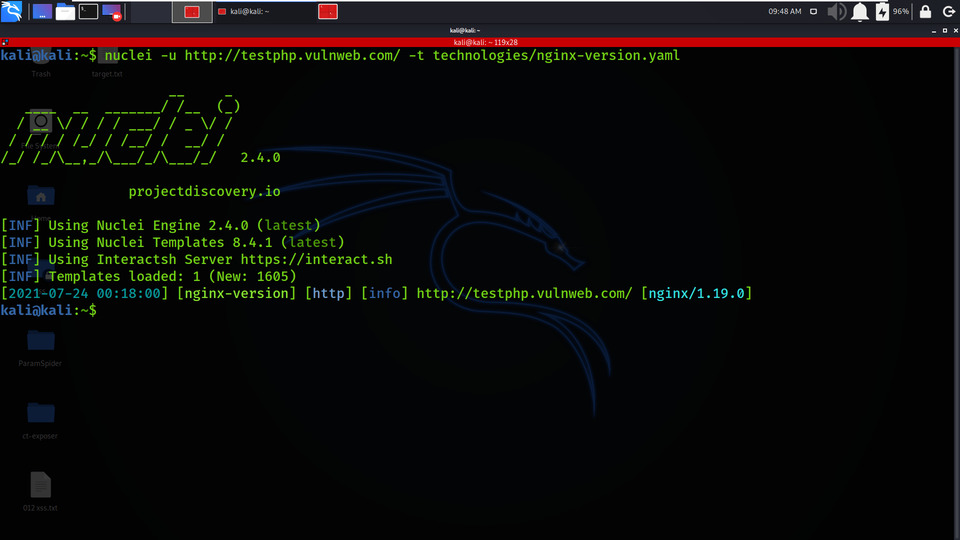


Рисунок 2 – Пример аналога 2

**Функции:**

* сканирование веб-приложений на наличие типичных уязвимостей, таких как SQL-инъекции, XSS и многие другие;
* с помощью инструмента можно проводить аудит серверов, сетевых устройств и других элементов инфраструктуры на наличие ошибок в конфигурации, сетевых портов и других потенциальных уязвимостей;
* поддерживает сканирование API на предмет уязвимостей и неправильных конфигураций;
* проверка безопасности сетевых устройств (маршрутизаторы, коммутаторы, межсетевые экраны).

**Особенности:**

* гибкость и настраиваемость;
* возможность создания собственных шаблонов для сканирования;
* инструмент имеет высокую скорость работы и производительность;
* активная поддержка сообществом и командой разработчиков.

**Проблемы:**

* инструмент имеет только DAST;
* не имеет анализа кода;
* инструмент не стоит РоС сам – лишь показывает факт уязвимости;
* нет дашборда.

## Аналог 3:

**Burp Suite Professional.**

Это инструмент для тестирования безопасности веб-приложений. Программа позволяет видеть, как данные передаются между браузером и сервером, и искать ошибки в этом процессе.

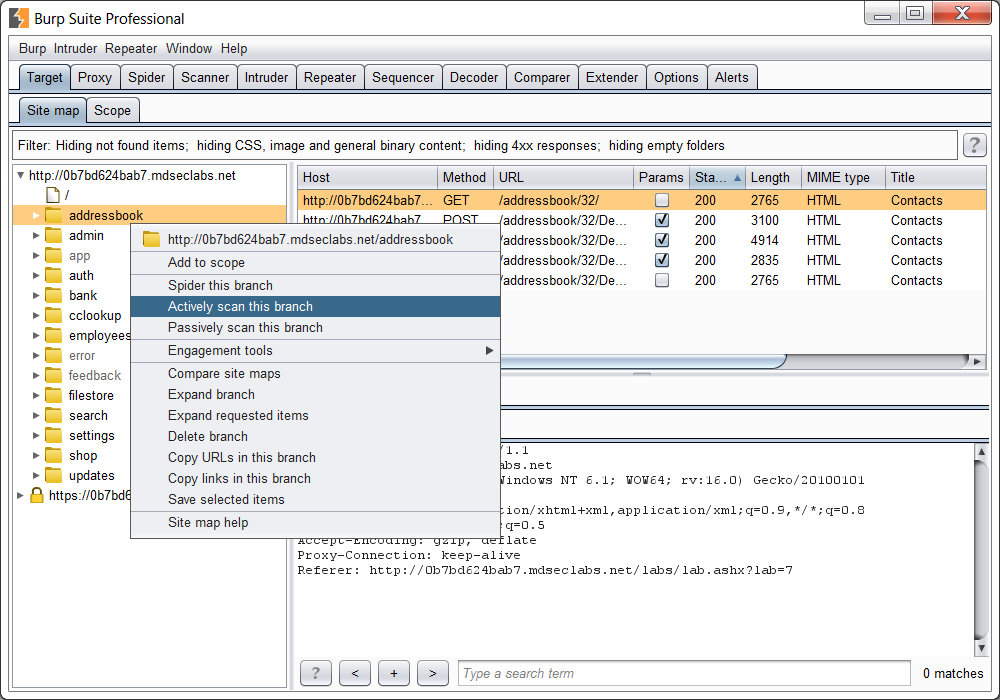


Рисунок 3 – Пример аналога 3

**Функции:**

* перехват прокси-сервера для мониторинга, инспектирования и модификации HTTP-запросов и ответов в реальном времени;
* автоматизированный сканер уязвимостей;
* анализ случайных токенов для выявления слабой криптографии;
* автоматическое обнаружение и картирование сайта, поиск файлов и директории.

**Особенности:**

* поддержка множества протоколов;
* расширение функционала через плагины для создания расширений;
* поддержка кастомых атак и анализ ответов.

**Проблемы:**

* инструмент платный;
* требует ручной работы;
* нет легкой интеграции CI/CD без платных надстроек.

# Требования к системе

## Требование к системе

В требование системы входит:

* показатели назначения;
* требования безопасности;
* требования к надежности;
* требования к патентной чистоте;
* требования к структуре и функционированию системы;

### Требования к структуре и функционированию системы

Перечень подсистем и их назначение:

* подсистема краулинга;
* подсистема эмуляции атак;
* подсистема подтверждения РоС;
* подсистема обработки пользовательских запросов;
* подсистема агрегации результатов;
* подсистема формирования отчётов;
* подсистема интеграции.

Перспективы развития и модернизации системы:

* добавление новых сценариев атак и шаблонов эксплойтов;
* расширение интеграции с другими инструментами безопасности;
* поддержка распределённого режима работы;
* внедрение машинного обучения для приоритизации уязвимостей и снижения числа ложных срабатываний;

Требования к режимам функционирования системы:

* возможность работы в режиме непрерывного сканирования и CI-интеграции;
* обеспечение стабильной работы в круглосуточном режиме, за исключением плановых обновлений и обслуживания.

Требования к способам и средствам связи для информационного обмена между компонентами системы:

* использование REST API для взаимодействия компонентов системы;
* обеспечение защищённого обмена данными;
* поддержка очередей сообщений для управления задачами.

Требования к характеристикам взаимосвязей создаваемой системы со смежными системами:

* возможность интеграции с системами CI/CD;
* поддержка импорта/экспорта результатов в формате JSON, CSV, PDF;
* возможность интеграции с корпоративными SIEM и баг-трекерами.

Требования по диагностике системы:

* логирование всех операций, ошибок и результатов в централизованный журнал;
* поддержка мониторинга состояния системы;
* возможность перезапуска и восстановления задач после сбоев.

### Показатели назначения

Показатели назначения системы:

* система должна обеспечивать доступность 24/7, за исключением запланированных периодов обслуживания;
* среднее время сканирования тестового приложения не должно превышать 15 минут в режиме «быстрой проверки»;
* возможность добавления новых атак и модулей, и поддержка увеличения объема данных без значительных изменений архитектуры системы;
* поддержка увеличения числа параллельных задач сканирования за счёт запуска дополнительных воркеров.



### Требования к надежности

Требования к надежности системы:

* среднее время наработки на отказ должно быть не менее 1000 часов;
* после сбоя система должна быть восстановлена в течение 30 минут;
* система должна корректно обрабатывать сетевые ошибки и падения внешних сервисов.

### Требования к безопасности

Система должна соответствовать требованиям информационной безопасности и защиты персональных данных, включая:

* доступ к интерфейсу должен осуществляться через безопасную аутентификацию;
* реализация ролевой модели;
* использование HTTP-запросов для всех API-запросов и TLS для соединений с БД;
* скриншоты и РоС должны храниться в защищённом хранилище с разграничением доступа;
* система должна работать только в пределах заданных адресов, исключая сканирование сторонних ресурсов без разрешений.

### Требования к защите информации от несанкционированного доступа

Система должна обеспечивать:

* все взаимодействия между компонентами системы должны осуществляться по защищённым каналам;
* каждый пользователь систем должен иметь индивидуальную учётную запись с разграничением прав доступа;
* использование ролевой модели для ограничения функций в зависимости от уровня доступа пользователя:
* учётные данные для тестируемых приложений, токены и ключи должны храниться в зашифрованном виде;
* система должна запускать сканы только в пределах разрешённого диапазона адресов, исключая несанкционированные тестирование сторонних ресурсов.

### Требования к патентной чистоте

Система автоматизированного тестирования веб-безопасности должна отвечать требованиям к патентной чистоте в соответствии с законодательством Российской Федерации, не нарушая прав на интеллектуальную собственность других компаний или лиц. Разработчик обязуется использовать только открытые API и лицензированные источники данных.

## Требование к функциям (задачам), выполняемым системой

Все функциональные требования представлены в Таблице 1.

Таблица 1 – Функциональные требования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название функции** | **Входные параметры** | **Выходные параметры** | **Описание функции** | **Подсистема** |
| Запуск сканирования | URL, scope, creds, профиль | Task\_id, статус | Полный цикл: краул -> атаки -> РоС -> отчет | Обработка запросов, эмуляция атак |
| Получение статуса задачи | Task\_id | Статус, прогресс, логи | Онлайн-статус выполнения | Обработка запросов |
| Эмуляция атак | Список эндпоинтов, профиль атак | Результат по эндпоинтам | XSS, SQLi, CRSF, LFI/SSRF и др. | Эмуляция атак |
| Подтверждение РоС | Vuln\_id, endpoint, payload | HTTP-артефакты, скриншоты | АвтопоС | Подтверждение РоС |
| Агрегация и нормализация | Результаты ZAP/Neclei/SemgreP | Единый список с уровнями серьёзности | Дупликация, приоритизация | Агрегация результатов |
| Формирование отчета | Task\_id | JSON/HTML/PDF | Отчёт с РоС и рекомендации | Отчёты |
| Хранение артефактов | Скриншоты и логи | Ссылка или ID артефакта | Сохранение доступа к артефактам | Хранилище |
| Установка пользователями | Данные пользователя, роль | Токен, профиль | RBAC, роли | Пользователи |
| Интеграция с CI/CD | Webhook/API | Код ответа, ссылки на отчеты | Автозапуск | Интеграция |
| Мониторинг и диагностика | События | Метрики, алерты | Логи | Диагностика |
| Управление целями/проектами | Проект, таргеты, теги, правила контроля доступа | ID проекта или таргета | Каталог целей, импорт доменов/подсетей, область охвата | Управление целями |
| Сценарии аутентификации | Шаги логина, селекторы, 2FA | Валидная сессия | Скрипты логина/обновление сессий | Аутентификация/сессия |
| Импорт API-спецификаций | OpenAPI/Swagger URL/файл, Postman, GraphQL | Список эндпоинтов API | Точное API-сканирование | API-скан |
| Профиль и политика скана | Профиль и пороги | ID политики | Настройка глубины, активных атак, fail-rules для CI | Политики |
| Фаззинг и пойлоады | Набор пэйлоадов, лимиты | Результат фаззинга | Генерация/мутации пэйлоадов, «безопасный режим» | Эмуляция атак |
| Анти-DoS | RPS, concurrency, паузы | Применённые лимиты | Контроль нагрузки | Оркестратор |
| Триаж ложных срабатываний | Vuln\_id, метка FP, комментарий | Обновлённый статус | Пометка FP, назначение ответственности | Триаж |
| Сравнение сканов | Scan\_id A/B | Список изменений | Регрессии/улучшения, тренды по времени | Аналитика |
| Интеграция с трекерами | Jira/YouTrack creds, vuln\_id | ID задачи в трекере | Автосоздание тикетов с PoC | Интеграции |
| Экспорт форматов | Формат (SARIF/CSV/JSON/PDF) | Файл/ссылка | Совместимость с внешними системами | Отчеты |
| Обновление шаблонов | Источники Nuclei/Semgrep | Статус обновления | Авто-апдейт правил и шаблонов | Обслуживание |
| Маскирование чувств, данных | Отчет или артефакт | Обезличенный отчёт | PII-редакция скриншотов/логов | Безопасность отчетов |
| Ретешн/бэкап | Политика хранеия | Удалённые/заархивированные записи | Жизненный цикл артефактов | Хранилище |
| Управление секретами | Креды, токены | Секрет-id | Безопасное хранение/ротация секретов | Безопасность |
| Self-health | - | Статусы компонентов | Проверка доступности БД, Redis, ZAP, и т.д. | Диагностика |
| Аудит действий | Событие, пользователь | Запись аудита | Журнал действий (кто что запускал/менял) | Безопасность |
| Мультипроектность | Организация или проект | Access scope | Изоляция данных по проектам/командам | Управление целями |
| Импорт/экспорт конфигов | Профили/политики/объем работы | Архив и импорт | Переезд конфигов между стендами | Администрирование |

## Требования к видам обеспечения

В требования к видам обеспечения:

* требование к информационному обеспечению;
* требование к лингвистическому обеспечению;
* требование к техническому обеспечению;
* требования к математическому обеспечению;
* требования к метрологическому обеспечению;
* требование к форматам хранения данных.



### Требования к математическому обеспечению

Требований не предъявляются.

### Требование к информационному обеспечению

Система должна поддерживать удобный и интуитивно понятный пользовательский интерфейс. Система должна осуществлять хранение информации о задачах, найденных уязвимостях, артефактах. Должно быть реализована возможность поиска, фильтрации и сортировки найденных уязвимостей по типу, критичности, дате обнаружения и статусу. Система должна формировать отчеты в различных форматах (JSON, HTML, PDF).

### 5.3.3 Требования к лингвистическому обеспечению

Интерфейс модуля должен быть на русском языке, с поддержкой стандартных терминов и понятий.



### Требования к метрологическому обеспечению

Требований не предъявляются.

### Требование к техническому обеспечению

Серверная часть системы должна быть развёрнута на надежной инфраструктуре. Система должна поддерживать работу в контейнеризированной среде. Все компоненты должны функционировать на современных операционных системах. Для хранения данных должна использоваться промышленная СУБД – PostgreSQL. Для управления задачами используется брокер сообщений. Для эмуляции атак и РоС используется поддержка headless-браузера.

### Требование к форматам хранения данных

Все результаты тестов, артефакты и отчёты должны храниться в базе данных (PostgreSQL) и файловом хранилище.

Отчёты должны поддерживать хранение и выгрузку в форматах JSON, CSV, HTML, PDF.

Скриншоты и артефакты PoC должны храниться в графическом формате PNG/JPEG, логи – в текстовом формате .log / .txt.

Взаимодействие с внешними сервисами и API должно осуществляться только через защищённые каналы связи.

# Состав и содержание работ по созданию системы

Этапы разработки модуля указаны в таблице 2.

Таблица 2 – Этапы разработки

|  |  |  |  |
| --- | --- | --- | --- |
| № этапа | Наименование этапа | Состав работ | Результат |
| 1 | Настройка рабочего окружения | Выбор язык для написания программы; выбор среды разработки, на котором будет писаться данная программа; выбор библиотек, которые будут использоваться для написания программы. | Акт выполненных работ; готовое к написанию кода рабочее место. |
| 2 | Техническое задание | Описание предметной области; анализ требований | Техническое задание. |
| 3 | Проектирование | Проектирование модуля. | Акт выполненных работ. |
| 4 | Написание кода программного обеспечения | Написание прототипа; написание готовой программы; | Акт выполненных работ; программное обеспечение. |
| 5 | Тестирование программы | Протестировать приложение на основе методики тестирования. | Акт выполненных работ; список недоработок и ошибок в работе программного обеспечения. |
| 6 | Доработка программы | Исправление ошибок в программе, которые были выявлены в тестировании. | Акт выполненных работ. |
| 7 | Подготовка эксплуатационной документации | Написание руководство пользователя; написание руководства программиста (оператора); написание программы и методики испытаний; написание отчета по практике | Руководство пользователя; руководство программиста (оператора); программа и методика испытания; отчет по учебной практике |
| 8 | Внедрение в эксплуатацию | Презентация и защита программы и программной документации | Защита учебной практики |

# Порядок контроля и приемки системы

## Виды испытаний

Во время испытаний проверить работу системы автоматизированного тестирования веб-безопасности по следующим позициям:

* набор функциональных тестов;
* корректное функционирование заданных в техническом задании функций;
* производительность и масштабируемость;
* надежность работы;
* безопасность системы;
* совместимость и переносимость;
* интеграционные испытания;
* испытания пользовательского интерфейса.

## Общие требования

Сдача-приемка осуществляется комиссией, в состав которой входят представители Заказчика и Исполнителя. По результатам приемки подписывается акт приемочной комиссии, подтверждающий соответствие реализованной функциональности требованиям настоящего ТЗ.

# Требования с составу и содержанию работ по подготовке объекта автоматизации к вводу систмы в действие

Состав и содержание работ показано в Таблице 3.

Таблица 3 – Состав работ

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Название работ** | **Содержание работ** |
| 1 | Анализ требований | Сбор, анализ и документирование требований к системе, согласование с заказчиком |
| 2 | Настройка рабочего окружения | Подготовка и настройка необходимого программного обеспечения и инструментов |
| 3 | Проектирование | Разработка архитектуры системы, планирование функциональных модулей и интерфейсов |
| 4 | Разработка программного обеспечения | Реализация кода системы на основе утвержденных требований и архитектуры |
| 5 | Тестирование | Проверка системы на наличие ошибок, тестирование функциональности и безопасности |
| 6 | Доработка программы | Исправление ошибок, выявленных в ходе тестирования |
| 7 | Подготовка эксплуатационной документации | Написание руководства пользователя, руководства программиста, методики испытаний и отчета по практике |
| 8 | Внедрение в эксплуатацию | Размещение системы, настройка, презентация и защита проекта |

# Требования к документированию

Для системы автоматизированного тестирования веб-безопасности на различных стадиях создания программного продукта должны быть выпущены следующие документы, из числа предусмотренных в ГОСТ 34.201-89, приведены ниже:

* аналитическая записка;
* отчет по учебной практике;
* программа и методика испытания;
* руководство пользователя;
* руководство программиста (оператора);
* техническое задание.

# Требования к приемо-сдаточным испытаниям

## Виды испытаний

Во время испытаний проверить работу программы по следующим позициям:

* набор функциональных тестов;
* корректное функционирование заданных в техническом задании функций;
* возможность функционирования на ЭВМ с указанными минимальными системными требованиями.

## Общие требования

Испытания проводятся согласно Программе и Методике Испытаний комиссией, включающей представителей заказчика:

* руководитель образовательной программы, Сергеева Е. Г.
* руководитель учебной практики, Долженкова М.Л.

Комиссии должны быть предъявлены эксплуатационные документы, MVP ИС и доклад. Оценка результатов осуществляется комиссией коллегиально.