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

**ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ**

«Система автоматизированного тестирования безопасности веб-приложений»

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

2024

Содержание

[1 Объект испытаний 1](#_Toc182758774)

[2 Цель испытаний 2](#_Toc182758775)

[3 Требования к боту 3](#_Toc182758776)

[4 Требования к программной документации 4](#_Toc182758777)

[5 Методы испытаний 5](#_Toc182758778)

[6 Требования по испытаниям программных средств 8](#_Toc182758779)

[7 Перечень работ, проводимых после завершения испытаний 9](#_Toc182758780)

[8 Метрологическое обеспечение испытаний 10](#_Toc182758781)

[9 Отчетность 11](#_Toc182758782)

# Объект испытаний

Объектом испытаний является система для автоматизированного тестирования безопасности веб-приложений. Разрабатываемая система предназначена для ля специалистов в области информационной безопасности, разработчиков и DevOps-инженеров, заинтересованных в оперативном выявлении, анализе и подтверждении уязвимостей в веб-ресурсах и API-сервисах.

# Цель испытаний

Целью испытаний является проверка соответствия функциональности системы требованиям технического задания, оценка ее стабильности и надежности работы при различных сценариях взаимодействия с пользователями. Испытания включают проверку корректности обработки данных, удобства пользовательского интерфейса и соответствия требованиям безопасности.

# Требования к системе

Система автоматизированного тестирования безопасности должен реализовывать следующие функции:

* инициализация и авторизация пользователей: при входе пользователь проходит авторизацию;
* после успешного входа открывается основное меню интерфейса с доступом к основным модулям;
* запуск сканирования веб-приложений: пользователь может указать URL, параметры проверки (DAST, SAST, или комбинированный), глубину анализа, авторизационные данные и дополнительные опции;
* мониторинг задач и статусов: пользователь имеет возможность отслеживать прогресс выполнения проверки, видеть процент выполнения, время старта и статус;
* анализ и подтверждение уязвимостей: система автоматически проверяет обнаруженные уязвимости и формирует доказательства (PoC): логи, HTTP-запросы, ответы сервера, скриншоты;
* формирование отчетов: реализуется возможность формирования и просмотра отчётов с результатами анализа (типы найденных уязвимостей, уровень критичности, описание уязвимости и рекомендации по устранению, РоС);
* управление пользователями и ролями: система поддерживает распределение ролей (администратор, пользователь);
* интеграция с CI/CD: предусмотрена возможность автоматического запуска проверки при обновлении кода в репозитории. Запуск осуществляется через REST API и webhook-и;
* хранение и безопасность данных: все результаты, отчёты и логи сохраняются в защищённой базе данных PostgreSQL и файловом хранилище MinIO. Доступ регулируется ролевой моделью (RBAC), соединения защищены TLS;
* панель мониторинга системы: система отображает состояние всех сервисов, нагрузку, активные воркеры, очередь Redis, использование ресурсов и ошибки. Мониторинг реализуется через Prometheus и Grafana;
* настройки и конфигурации: пользователь (или администратор) может изменять параметры окружения, подключений и пороговые значения критичности уязвимостей. Все настройки сохраняются в .env и config.yml.

# Требования к программной документации

К моменту тестирования должны быть подготовлены следующие документы:

* техническое задание;
* руководство пользователя;
* руководство программиста.

# Методы испытаний

Испытания будут проводиться согласно следующему плану:

* проверка состава и качества сопроводительной документации: анализ полноты и соответствия предоставленной документации требованиям;
* функциональное тестирование: проверка корректности работы всех функций системы в соответствии с пользовательскими сценариями;
* тестирование пользовательского интерфейса: оценка удобства использования системы, правильности отображения элементов интерфейса;
* тестирование стабильности и производительности: проверка работы системы под нагрузкой, оценка ее реакции на множественные запросы;
* тестирование на различных устройствах: проверка работы системы в разных браузерах;
* тестирование безопасности: проверка защиты пользовательских данных и устойчивости к некорректным входным данным.

Методы испытаний представлены в Таблице 1.

Таблица 1.

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Название проверки** | **Выполняемые действия** | **Ожидаемый результат** |
| 1 | Проверка документации | Анализ предоставленных документов: технического задания, руководства пользователя, руководства программиста, программы и методики испытаний. | Документация соответствует требованиям, содержит всю необходимую информацию. |
| 2 | Инициализация системы | Запуск всех сервисов командой:  docker-compose up -d.  Проверка их состояния:  docker-compose ps.  Переход в браузере по адресу http://localhost:8000 и проверка загрузки главной страницы. | Все контейнеры запущены и имеют статус Up (healthy). Веб-интерфейс открывается, система готова к работе. |
| 3 | Авторизация пользователя | Ввести логин и пароль в форме авторизации. Проверить выдачу JWT-токена и переход в панель управления после входа. | Пользователь успешно проходит авторизацию, токен выдан, интерфейс загружается без ошибок. |
| 4 | Создание задачи сканирования | В разделе «Новая проверка» указать тестовый URL (например, https://example.com), выбрать тип анализа (DAST/SAST/комбинированный), задать глубину проверки и запустить задачу. | Система создаёт задачу, присваивает уникальный task\_id, статус отображается как «В очереди». |
| 5 | Мониторинг выполнения задачи | В разделе «Мониторинг задач» наблюдать изменение статуса и процент выполнения. Проверить корректность логов в реальном времени. | Статус задачи меняется последовательно («в очереди -> выполняется -> завершено»). Логи отображаются без ошибок. |
| 6 | Формирование отчета | После завершения сканирования открыть раздел «Отчёты», выбрать задачу по task\_id, просмотреть отчёт и скачать PDF/JSON. | Отчёт формируется корректно, содержит список уязвимостей, уровни критичности, доказательства (PoC) и рекомендации. |
| 7 | Просмотр Proof-of-Concept | Открыть отчёт по task\_id, выбрать уязвимость, нажать «Просмотреть PoC», проверить корректность отображения запроса, ответа и скриншота. | Скриншоты и ответы сервера отображаются корректно, доказательства уязвимости сохраняются в артефактах MinIO. |
| 8 | Проверка хранения данных | Подключиться к базе данных PostgreSQL и убедиться в наличии записей о задачах и результатах. Проверить сохранение артефактов в MinIO. | Все результаты и файлы успешно сохранены, структура данных соответствует описанию. |
| 9 | Интеграция с CI/CD | Отправить webhook-запрос на /api/v1/scan/start из тестового пайплайна GitHub Actions. Проверить автоматический запуск и получение отчёта. | Сканирование запускается автоматически при обновлении кода, отчёт возвращается корректно. |
| 10 | Тестирование интерфейса | Перейти по всем разделам веб-панели (Новая проверка, Мониторинг задач, Отчёты, Настройки), проверить навигацию, отображение элементов и адаптивность. | Веб-панель отображается корректно, все элементы кликабельны, переходы работают без ошибок. |
| 11 | Тестирование стабильности | Запустить несколько параллельных задач сканирования, наблюдать нагрузку и время отклика системы, убедиться в отсутствии сбоев и ошибок. | Система устойчива к нагрузке, обрабатывает множественные запросы без задержек и ошибок. |
| 12 | Тестирование производительности | Запустить 10 параллельных задач сканирования. Замерить среднее время выполнения, нагрузку на CPU и RAM. | Система сохраняет стабильность, все задачи завершаются успешно без зависаний и ошибок. |
| 13 | Проверка безопасности | Попробовать отправить в API некорректные данные (XSS, SQL-инъекции, спецсимволы). Проверить реакцию системы. | Ввод проверяется корректно, система отклоняет вредоносные данные, ошибок и утечек нет. |

# Требования по испытаниям программных средств

Испытания программных средств системы для автоматизированного тестирования безопасности веб-приложений проводятся в рамках функционального и нагрузочного тестирования. Других требований к испытаниям программных средств бота не предъявляется.

# Перечень работ, проводимых после завершения испытаний

По результатам испытаний делается заключение о соответствии системы требованиям технического задания и возможности его использования клиентами. При необходимости проводится доработка программных средств и документации.

# Метрологическое обеспечение испытаний

Программа испытаний системы автоматизированного тестирования безопасности веб-приложений не требует использования специализированного измерительного оборудования.

# Отчетность

Результаты испытаний системы фиксируются в протоколах, которые включают следующие разделы:

* назначение испытаний и номер раздела требований ТЗ для бота, по которому проводят испытание;
* состав технических и программных средств, используемых при испытаниях;
* указание методик, в соответствии с которыми проводились испытания, обработка и оценка результатов;
* условия проведения испытаний и характеристики исходных данных;
* средства хранения и условия доступа к тестирующей программе;
* обобщённые результаты испытаний;
* выводы о результатах испытаний и соответствии бота требованиям технического задания.

В протоколах также могут быть зафиксированы замечания пользователей и тестировщиков по удобству эксплуатации системы.

Этап проведения предварительных испытаний завершается оформлением акта о результатах предварительных и приемочных испытаний системы для автоматического тестирования безопасности веб-приложений.

**ПРОТОКОЛ**

**Предварительных и приемочных испытаний «Система для автоматического тестирования безопасности веб-приложений»**

В соответствии с требованиями технического задания были проведены испытания «системы для автоматического тестирования безопасности веб-приложений» в соответствии с утвержденной «Программой и методикой испытаний».

Общие сведения об испытаниях приведены в таблице 2.

Результаты испытаний приведены в таблице 3.

Таблица 2 – Общие сведения об испытаниях

|  |  |  |
| --- | --- | --- |
| **Испытываемый образец:** | **Telegram-бот для мониторинга рыночных активов** | |
| Дата проведения испытаний: | « » 20\_\_г. | |
| Место проведения испытаний |  | |
| Испытания проводили: | Фамилия И.О. | Должность |
| От Исполнителя | Вершинина Н.Д. | Студент |
| От Заказчика | Долженкова М.Л. | Руководитель УП |

Таблица 3 – Результаты испытаний

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Шаг испытаний** | **№ пункта методики** | **Отметка о прохождении (Да/Нет)** | **Примечания** |
| 1 | Проверка документации | 1 | Да | Документация соответствует требованиям, содержит все необходимые разделы. |
| 2 | Инициализация системы | 2 | Да | Все контейнеры запущены, интерфейс доступен, система готова к работе. |
| 3 | Авторизация пользователя | 3 | Да | Авторизация проходит успешно, пользователь получает JWT-токен и доступ к панели управления. |
| 4 | Создание задачи сканирования | 4 | Да | При вводе URL и параметров проверки система корректно создаёт задачу и присваивает task\_id. |
| 5 | Мониторинг выполнения задачи | 5 | Да | Статус задачи изменяется последовательно («в очереди -> выполняется -> завершено»), логи отображаются корректно. |
| 6 | Формирование отчета | 6 | Да | Отчёт формируется корректно, содержит перечень уязвимостей, уровень критичности и рекомендации по устранению. |
| 7 | Просмотр Proof-of-Concept | 7 | Да | Скриншоты и ответы сервера отображаются корректно, доказательства уязвимости сохраняются в артефактах MinIO. |
| 8 | Проверка сохранения данных | 8 | Да | Результаты сканирования успешно записываются в PostgreSQL, артефакты хранятся в MinIO. |
| 9 | Интеграция с СI/CD | 9 | Да | При отправке webhook-запроса из GitHub Actions происходит автоматический запуск сканирования и генерация отчёта. |
| 10 | Тестирование интерфейса | 10 | Да | Веб-панель отображается корректно, все элементы кликабельны, переходы работают без ошибок. |
| 11 | Тестирование стабильности | 11 | Да | Система устойчива к нагрузке, обрабатывает множественные запросы без задержек и ошибок. |
| 12 | Тестирование на разных устройствах и браузерах | 12 | Да | Система функционирует корректно на всех протестированных устройствах и браузерах. |
| 13 | Тестирование безопасности | 13 | Да | Некорректные данные обрабатываются безопасно, система устойчив к попыткам введения скриптов. |