**Техническое задание**

на разработку веб-приложения Sploit2Me

Команда: МЫ

*Жмайло Михаил Александрович (тимлид)*

*Пономарёва Анастасия Юрьевна (аналитик, дизайнер)*

*Саетбатталов Павел Равильевич (backend-разработчик)*

*Репин Михаил Евгеньевич (frontend-разработчик)*

**Содержание**

**1. Общие положения**

* **Полное наименование системы**

Полное наименование — Веб-сервис Sploit2Me для анализа и своевременного уведомления бизнеса о наличии новых уязвимостей в комопнентах программного обеспечения.

* **Определения, обозначения и сокращения**

БД – база данных.

ОС – операционная система.

Система – разрабатываемый веб-сервис со всеми программными компонентами.

СУБД – система управления базами данных.

Python — интерпретируемый язык программирования, используемый в разработке.

Информационная система — система обработки информации совместно с соответствующими организационными ресурсами, которая обеспечивает и распространяет информацию.

Flask — фреймворк (набор инструментов) для создания веб-приложения.

Недопустимое событие — ситуация, при которой появляется возможность компрометации чувствительной информации, воздействия на цифровой ресурс, иной метод, приводящий к проблемам с точки зрения безопасности обрабатываемых данных, в том числе персональных.

Уязвимость — недостаток программного обеспечения, приводящий к определенным недопустимым событиям.

БДУ ФСТЭК — база данных уязвимостей федеральной службы по техническому и экспортному контролю.

Эксплойт — программа, эксплуатирующая уязвимость. Исполнение эксплойта приводит к возможности совершения недопустимого события, либо автоматического совершения оного.

JSON — JavaScript Object Notation. Текстовый формат передачи данных.

Telegram — платформа для обмена сообщениями (т.н. мессенджер).

Telegram Бот, Бот — компонент платформы Telegram, позволяющий разработчикам автоматизировать процесс отправки сообщений пользователям.

Токен — уникальное значение, однозначно идентифицирующее или авторизирующее пользователя или бота.

Подписка — процесс уведомления пользователей о появившейся уязвимости посредством бота.

Фильтр — применяемые к массиву входных данных преобразования с целью выделения необходимой для пользователя информации.

Логин — уникальный строковый идентификатор пользователя.

Хеш-функция — функция, осуществляющая преобразование массива входных данных произвольной длины в выходную битовую строку установленной длины, выполняемое определённым алгоритмом.

Хеш — результат исполнения хеш-функции.

Хешированный пароль, или пароль в хешированном виде — результат выполнения хеш-функции по отношению к парольной фразе.

* **Назначения и цели создания системы**
* **Назначение системы**

Создаваемая система предназначена для автоматического извлечения информации о появляющихся уязвимостей в программном обеспечении с информационных ресурсов БДУ ФСТЭК России и последующим уведомлением о наличии уязвимости посредством отправки сообщения в Telegram. Механизм системы осуществляется посредством следующей функциональности:

* Механизм контроля доступа: регистрация, идентификация, аутентификация, авторизация;
* Механизм подписки на уведомления посредством бота;
* Механизм отправки уведомлений пользователям посредством бота;
* Механизм отключения подписки на уведомления;
* Механизм получения информации с информационных ресурсов БДУ ФСТЭК России.

* **Цели создания системы**

Первоочередной целью разработки системы является своевременное уведомление пользователей о появившихся уязвимостях в программных компонентах информационных систем путем сбора информации с ресурсов публично доступной базы данных уязвимостей.

Цель проекта осуществляется посредством:

* изучение необходимого набора навыков и инструментария путем изучения WEB-ресурсов и технической литературы;
* подготовка графической составляющей, разработка UI;
* написание программного кода с необходимым функционалом;
* тестирование

В частности, следует решать задачи по:

* обеспечению механизма аутентификации, авторизации и контроля доступа пользователей;
* предоставление возможности извлечения обработанного массива данных с информационного ресурса БДУ ФСТЭК;
* осуществлять фильтрацию массива данных по требованию пользователя;
* гарантировать своевременное уведомление пользователей о появившейся уязвимости посредством извещения в мессенджере «Telegram»
* **Требования к системе**
* **Нефункциональные требования**

Функционирование системы должно осуществляться посредством размещения в сети Интернет на доменном имени. Хранение, запуск и управление функциональным составляющим информационной системы должно осуществляться внутри изолированных Docker-контейнеров. Помимо этого, система должна поддерживать бота в мессенджере «Telegram», посредством которого будет осуществляться уведомление пользователя о новой уязвимости.

Интерфейс пользователя должен быть простым в освоении, минималистичным. Во время работы пользователь может подключить аккаунт, на который будут отсылаться уведомления, отключить аккаунт, применить правила фильтрации. Пользователи могут быть ограничены в возможности добавления аккаунтов для уведомления об уязвимостях.

Механизм аутентификации и авторизации пользователей, а также хранения информации о подключенных аккаунтах для уведомления об уязвимостях должен быть осуществлен с использованием реляционной базы данных. В частности, разрешается использование SQLite, PostgreSQL.

* **Требования к интерфейсу**

Интерфейс Системы должен быть прост, нагляден, интуитивно понятен и легок в освоении, и должен отвечать следующим требованиям:

* Чистота дизайна: Интерфейс должен быть минималистичным, без лишних элементов, которые могут отвлекать пользователя;
* Интуитивная навигация: Пользователь должен легко понимать, как перемещаться по интерфейсу приложения. Разместить элементы навигации в доступных местах, использовать стандартные иконки и кнопки (например, меню для доступа к дополнительным разделам);
* Добавление аккаунта: следует осуществить реализацию функции добавления аккаунта, на который будут приходить уведомления о новых уязвимостях.
* Легкость добавления фильтрации: пользователи должны иметь возможность добавлять фильтры на получаемые посредством информирования через Telegram бота данные об уязвимостях.
* **Требования к обеспечению безопасности персональных данных**

Система должна обеспечивать реализацию следующих мер по обеспечению безопасности персональных данных:

* Аутентификация и авторизация: Обеспечьте надежную систему аутентификации для пользователей;
* Резервное копирование: рекомендуется создавать резервные копии данных. В частности, базы данных;
* Обучение пользователей: обучайте пользователей основам информационной безопасности, объясняйте важность использования сложных паролей, риски перехода по внешним ссылкам и открытия подозрительных вложений на почте.
* **Требование к производительности системы**

Система должна соответствовать следующим требованиям по производительности:

* Кеширование информации. Полученные сведения об уязвимостях должны кешироваться локально на сервере. Кеширование позволяет применять фильтры к передаваемым на клиента данным. Резервное копирование: рекомендуется создавать резервные копии данных. В частности, базы данных;
* Обработка ошибок и перегрузок. Стоит рассмотреть все возможные проблемы, которые могут появиться в процессе эксплуатации сервиса. Недоступность сетевой инфраструктуры, невозможность сохранения целостности получаемых данных, некорректные данные.
* Отзывчивость и интерактивность. Следует использовать многопоточную модель при проектировании программного продукта с целью предотвращения издержек, связанных с долгим ожиданием пользователя при попытке использовать какой-либо функционал.
* **Функциональные требования**

Каждая страница системы должна отображать в верхней правой части вкладку информации о пользователе, при нажатии на которую возможно просмотреть профиль пользователя. Система должна предоставлять возможность пользователю изменять пароль учетной записи, имя (логин), обновлять аккаунт для уведомлений об уязвимостях, отключать аккаунт, применять фильтры.

* **Требования к разграничению доступа**

Система должна реализовать роль «Пользователь». Данная роль назначается каждому зарегистрированному в системе пользователю. Функциональные возможности роли следующие:

* добавление аккаунта для уведомления об уязвимостях;
* удаление аккаунта для уведомления об уязвимостях;
* изменение имени текущего пользователя;
* изменения пароля текущего пользователя;
* добавление фильтров к получаемому массиву данных;
* удаление фильтров.
* **Требования к регистрации и аутентификации**

Система должна предоставлять роли «Пользователь» возможность проходить процесс аутентификации. Роль «Пользователь» назначается автоматически при создании нового пользователя через механизм регистрации. Механизм регистрации доступен без авторизации.

В процессе регистрации заполняются следующие поля:

* Логин пользователя;
* Почта;
* Секретная парольная фраза (дважды).

В процессе аутентификации пользователь должен заполнить следующие поля:

* Логин;
* Пароль.

Технические требования к реализации конкретного механизма представлены в пунктах ниже.

* **Требования к механизму регистрации пользователей**

Следует обеспечить:

* простоту процесса регистрации;
* отказоустойчивость системы при регистрации пользователя;
* предотвращения регистрации пользователей с одинаковыми логинами или почтами;
* хранение информации о парольной фразе пользователя в реляционной базе данных в виде хешированных данных;
* безопасность используемой парольной фразы.
* **Требования к механизму аутентификации пользователей**

Следует обеспечить:

* безопасность механизма аутентификации пользователей, защита от уязвимостей внедрения;
* отказоустойчивость системы при аутентификации пользователей;
* проверка пары логина и парольной фразы посредством обращение к реляционной базе данных
* **Требования к механизму добавления пользовательского аккаунта для уведомления об уязвимостях**

Следует обеспечить:

* возможность добавления Telegram-аккаунта, на который будут отправляться уведомления при появлении информации о новой уязвимости;
* хранить имя Telegram-аккаунта в реляционной базе данных.
* **Требования к механизму удаления информации о пользовательском аккаунте для уведомления об уязвимостях**

Следует обеспечить:

* надежное удаление информации об аккаунте из реляционной базы данных.
* **Требования к уведомлениям и индикации**

Следует обеспечить:

* целостность передаваемой информации;
* передавать информацию об уязвимостях посредством оповещения путем отправки сообщения через Telegram бота;
* применять настроенные пользователем правила фильтрации.