Solar appScreener

Руководство пользователя Solar appScreener модуль анализа компонентов с открытым кодом (OSA)

Solar appScreener

Версия 3.15.2

Июнь 2025

СОДЕРЖАНИЕ

1. ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

Термин	Расшифровка
АРМ	Автоматизированное рабочее место
ОС	Операционная система
ПО	Программное обеспечение
CLI	Command Line Interface – интерфейс командной строки
CLT	Command Line Tool – инструмент командной строки
REST	Representational State Transfer – передача состояния представления
SBOM	Software Bill of Materials – документ, содержащий подробный перечень компонент и зависимостей
SDLC	System Development Life Cycle – жизненный цикл разработки системы
VCS	Version Control System – система управления версиями

2. ГЛОССАРИЙ

API (Программный интерфейс приложения, Application Programming Interface) — интерфейс, который определяет взаимодействие программы с другой программой.

CI/CD система — система, которая объединяет практики непрерывной интеграции, непрерывной доставки и непрерывного развёртывания. CI/CD системы могут быть встроены в процесс разработки по методике SDLC и SSDLC. Примеры: **TeamCity**, **Jenkins**.

Continuous Delivery (Непрерывная доставка, CD) — практика разработки программного обеспечения, при которой команды производят программное обеспечение в короткие циклы.

Continuous Deployment (Непрерывное развёртывание, CD) — практика разработки программного обеспечения, которая заключается в использовании автоматизированного тестирования для проверки правильности и стабильности изменений в коде для быстрого, автономного развёртывания в производственной среде.

Continuous Integration (Непрерывная интеграция, CI) — практика разработки программного обеспечения, которая заключается в слиянии рабочих копий в общую основную ветвь разработки и выполнении автоматизированных сборок проекта для выявления потенциальных ошибок и решения интеграционных проблем.

ID проекта (Project ID) — первые 6 символов UUID проекта. ID проекта может быть использован при поиске проекта в общем списке проектов, но UUID является полным идентификатором проекта. Пример ID проекта: d4d1e2.

LDAP (Lightweight Directory Access Protocol) — протокол прикладного уровня для доступа к службе каталогов. Протокол использует TCP/IP и позволяет производить операции аутентификации (bind), поиска (search) и сравнения (compare), а также добавления, изменения или удаления записей.

Software Bill of Materials (SBOM) — документ, содержащий подробный перечень компонент и зависимостей, используемых в проекте программного обеспечения. В нем также перечислены все библиотеки, фреймворки и их соответствующие версии, которые используются в программном обеспечении.

UUID (Universally unique identifier) — 128-битное число, которое используется для идентификации информации. Пример: d4d1e2da-6b82-4350-829b-d3883592f4c8.

Version Control System (VCS, Система контроля версий) — программный инструмент, который служит для записи изменений в файлы и отслеживания изменений, внесённых в код. Примеры: **Git**, **Subversion**.

VCS хостинг — веб-сервис для хостинга проектов и их совместной разработки. Примеры: **GitLab**, **GitHub**, **GitFlic**, **Bitbucket**.

Безопасность приложения (Application security) — набор мер, принимаемых для повышения безопасности приложения, часто путём обнаружения, исправления и предотвращения уязвимостей безопасности.

Интеграция (Integration) — обмен данными между системами с возможной последующей обработкой.

Интерфейс командной строки (Терминал, Консоль, Command-line interface, CLI) — интерфейс, который обрабатывает команды для программы в виде строк текста.

Инструменты командной строки (CLT) — программы без графического интерфейса, взаимодействие с которыми осуществляется через командную строку.

Плагин (Расширение) — независимо компилируемый программный модуль, динамически подключаемый к основной программе и предназначенный для расширения её функциональных возможностей.

Система отслеживания ошибок (Bug tracking system, Bug tracker) — программа, разработанная для учёта и контроля ошибок, найденных в программном обеспечении, которая позволяет следить за процессом устранения этих ошибок. Примеры: **Jira**, **Redmine**.

Скрипт (Script) — последовательность команд для автоматического выполнения задачи.

Токен авторизации API (Токен, API authorization token) — набор символов, который предназначен для аутентификации пользователей для выполнения действий в системе без использования пользовательского интерфейса.

3. ВВЕДЕНИЕ

Настоящий документ представляет собой руководство пользователя Solar appScreener модуль анализа компонентов с открытым кодом (OSA)(далее Solar appScreener)

4. СВЕДЕНИЯ О SOLAR APPSCREENER

4.1. Описание возможностей

4.1.1. Анализ сторонних компонент

Модуль анализа сторонних компонент может сканировать библиотеки с открытым исходным кодом, используемые в приложении, на наличие уязвимостей, рисков цепочки поставок и лицензионных рисков.

Solar appScreener предлагает следующие возможности:

- Идентификация и управление безопасностью компонент с открытым исходным кодом с интерактивной визуализацией дерева зависимостей для проектов, написанных на: C/C++, C#, Dart, Erlang, GO, Java, JavaScript, Kotlin, Objective-C, PHP, Python, Ruby, Rust, Scala, Swift, TypeScript, VB.NET;
- Непрерывная оценка состояния пакетов с открытым исходным кодом по 8 метрикам анализа цепочки поставок;
- Идентификация лицензионных рисков использования открытого кода в ваших проектах;
- Комбинированный анализ SAST/OSA для отслеживания выполнения функций библиотек в коде приложения.

Solar appScreener использует следующие базы данных для идентификации уязвимостей в открытом исходном коде:

- NVD;
- GitHub, GitLab;
- OSV;
- Собственные базы данных Solar appScreener..

4.2. Требования к АРМ пользователя

4.2.1. Требования к аппаратному обеспечению

АРМ пользователя Solar appScreener должно быть оборудовано персональным компьютером с подключением к внутренней сети компании.

4.2.2. Требования к программному обеспечению

В состав программного обеспечения компьютера для АРМ пользователя Solar appScreener должна входить программа-клиент, предоставляющая пользователю возможность навигации и просмотра веб-ресурсов (браузер). Рекомендуемые браузеры (актуальные версии):

- **Mozilla Firefox**;
- **Google Chrome**;
- **Safari**;
- **Microsoft Edge**.

5. ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ

5.1. Авторизация

Ссылка для входа в веб-интерфейс Solar appScreener (далее UI) предоставляется администратором. При переходе по ссылке пользователь попадает на страницу авторизации.

Чтобы войти в систему, введите логин и пароль и нажмите кнопку **Войти** (рис. 5.1).

Вход в систему может быть осуществлен с помощью логина (в формате <user> или <user@domain>) и пароля учётной записи **LDAP**. Чтобы настроить доступ с помощью **LDAP**, обратитесь к администратору.

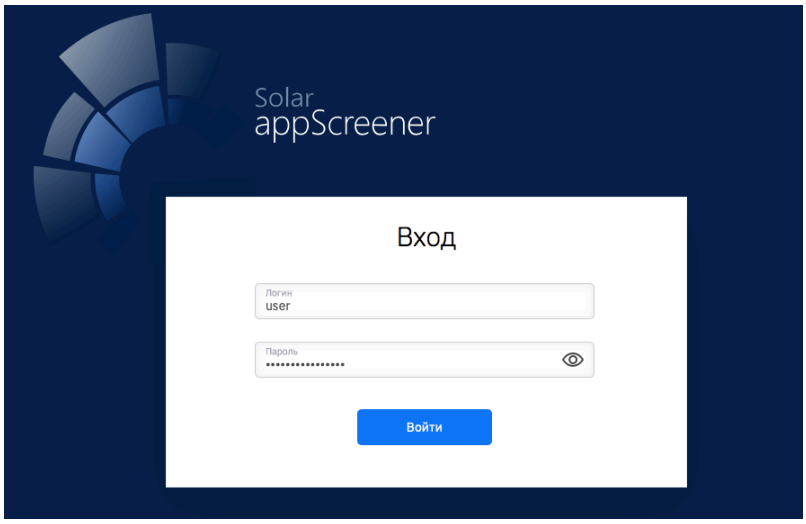

Рис. 5.1: Страница Авторизации

При введении неверных учётных данных на экране отобразится сообщение **Неверный логин и/или пароль.** При превышении числа попыток аутентификации с неверным паролем ваш аккаунт будет временно заблокирован. Количество попыток аутентификации и продолжительность блокировки устанавливается администратором системы (по умолчанию лимит попыток входа — 5, срок блокировки — 5 часов).

Прежде чем начать работу с Solar appScreener, ознакомьтесь с Пользовательским соглашением и нажмите **Принимаю**.

После успешного входа в систему отображается краткая инструкция. После просмотра инструкции отображается **Домашняя страница**. Повторный просмотр краткой инструкции доступен на странице **О продукте**.

5.2. Главное меню

В верхней части страницы расположено главное меню, которое предоставляет доступ к разделам **Домашняя страница**, **Проекты**, **Аналитика**, **О продукте**, **Личный кабинет**.

Для доступа к форме обратной связи нажмите .

5.2.1. Домашняя страница

Домашняя страница предназначена для загрузки и сканирования новых проектов. **Проектом** здесь и далее называется загрузка в Solar appScreener приложения и его сканирование с целью выявления уязвимостей. Под **сканированием** следует понимать анализ исходного или бинарного кода приложения и выявление фрагментов кода, содержащих уязвимости. **Уязвимости** — недостатки в коде приложения, которые могут быть использованы злоумышленниками и вызывать нарушение корректной работы приложения. Подробное описание запуска проекта представлено в разделе Создание проекта.

На **Домашней странице** также можно:

- ознакомиться со списком проектов — отображаются последние 4 проекта с последующей ссылкой на страницу **Проекты** с полным списком проектов;
- увидеть статистику по количеству сканирований с учётом статусов. Нажмите на статус, чтобы посмотреть список соответствующих проектов.

5.2.2. Проекты

Страница **Проекты** предназначена для управления проектами. Все проекты представлены в виде списка с краткими характеристиками.

Для каждого проекта отображаются следующие данные:

- логотип, название проекта, автор (пользователь, загрузивший проект), ID проекта (первые шесть символов UUID проекта);
- статус последнего сканирования;
- меню действий:
  - копировать UUID проекта;
  - посмотреть подробные результаты последнего сканирования;
  - запустить сканирование;
  - выгрузить отчёт;
  - настроить проект;
  - архивировать проект.
- кнопка добавления в Избранное;
- дата и время последнего сканирования;
- количество уязвимостей критического, среднего, низкого и информационного уровней, а также общее количество уязвимостей;
- рейтинг приложения.

Список можно отсортировать по названию, статусу последнего сканирования, по дате и по рейтингу. Для этого нажмите на соответствующий заголовок, повторное нажатие меняет порядок сортировки.

Для скрытия ненужных в данный момент проектов существует возможность архивации. Архивированный проект сохраняется в системе, но становится недоступным для работы. Выберите **Архивировать проект** в меню действий, чтобы добавить проект в архив.

Проект, находящийся в архиве, можно найти, нажав **Показать архив** на странице **Проекты**.

Для удобной навигации по проектам предусмотрены поиск и фильтры. Поиск позволяет искать проекты по названию, ID проекта или автору. Чтобы установить фильтры, нажмите на иконку фильтров и настройте один или несколько параметров:

- статус сканирования — выбрать из списка статусы сканирования;
- дата обновления — задать временной диапазон;
- рейтинг — задать диапазон для рейтинга последнего сканирования в проекте;
- количество уязвимостей каждого из уровней критичности — задать диапазон для количества уязвимостей критического, среднего, низкого или информационного уровня.

Чтобы установить фильтры, нажмите на кнопку **Применить**. После применения фильтров в правой части страницы появится количество отобранных проектов и кнопка **Сбросить**. Нажатие на кнопку отменит фильтрацию.

Чтобы перейти на страницу конкретного проекта, нажмите на его название в списке. Подробнее про управление конкретным проектом в разделе Управление проектом.

5.2.2.1. Очередь сканирований

На вкладке **Очередь сканирований** можно управлять очередью сканирований в системе, поднимать/опускать приоритет сканирований. Система поддерживает 4 уровня приоритета сканирований: Низкий, Средний, Высокий и Эксклюзивный. По умолчанию сканирования запускаются со **Средним** приоритетом.

Искать сканирование в очереди можно по ID, названию проекта и автору сканирования.

В основной части страницы находится список всех активных сканирований в системе. Он представлен в виде таблицы со столбцами:

- Название проекта — кликабельно, по клику происходит переход на страницу **Обзор** (при наличии доступа в проект у пользователя);
- Сканирование — первые 6 символов UUID сканирования (по кнопке может быть скопирован целиком) и автор сканирования;
- Дата создания;
- Статус;
- Приоритет.

Все столбцы поддерживают сортировку, по умолчанию отсортированы по приоритету. Сканирования с одинаковым приоритетом сортируются по дате создания: первым отображается и будет просканирован проект, запущенный *раньше*.

Обратите внимание: изменение приоритета сканирования затрагивает только проекты в очереди и не повлияет на прогресс сканирований, которые уже выполняются.

5.2.3. Группы проектов

Проекты в Solar appScreener можно объединять в группы. Используя группы, можно выполнять действия с несколькими логически связанными проектами одновременно.

Также для групп доступна сводная информация и аналитика. Работать с группами можно в разделе **Группы проектов** (рис. 5.2).

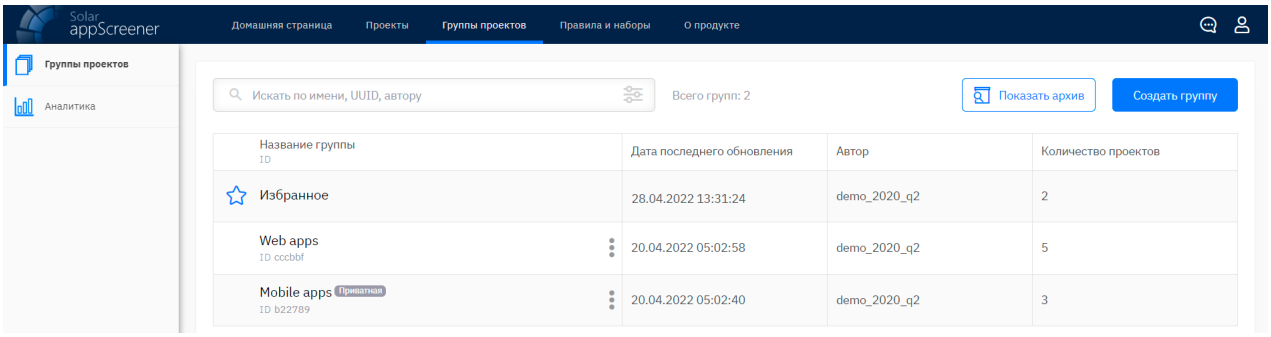

Рис. 5.2: Группы проектов

Список групп отображается в основной части страницы. Для каждой группы отображаются следующие данные:

- название группы;
- ID группы;
- видимость;
- меню действий:
  - копировать UUID группы;
  - проекты — перейти к списку проектов;
  - настроить группу;
  - архивировать группу.
- дата и время обновления;
- автор;
- количество проектов в группе.

Список можно отсортировать. Для этого нажмите на соответствующий заголовок, повторное нажатие меняет порядок сортировки. Для удобной навигации также предусмотрен поиск и фильтры. Поиск позволяет искать группы проектов по названию, UUID группы или автору. Чтобы установить фильтры, нажмите на иконку фильтров и настройте один или несколько параметров:

- видимость — выбрать публичные или приватные группы;
- количество проектов в группе — можно указать диапазон;
- дата создания;
- дата последнего обновления;
- наличие проектов — добавить проекты, которые должна содержать группа.*

* *Обратите внимание: при выборе нескольких проектов фильтр работает как условие ИЛИ, то есть результаты поиска будут содержать группы проектов с хотя бы одним из указанных проектов.*

Чтобы установить фильтры, нажмите на кнопку **Применить**. После применения фильтров в правой части страницы появится количество отобранных проектов и кнопка **Сбросить**. Нажатие на кнопку отменит фильтрацию.

5.2.3.1. Создание группы проектов

Чтобы создать группу проектов, нажмите **Создать группу**, укажите имя группы и выберите проекты, которые следует в неё включить. Также можно включить проекты из существующих групп. Чтобы группа отображалась у всех пользователей, выберите опцию **Публичная**. После выполнения этих действий нажмите **Сохранить** (рис. 5.3).

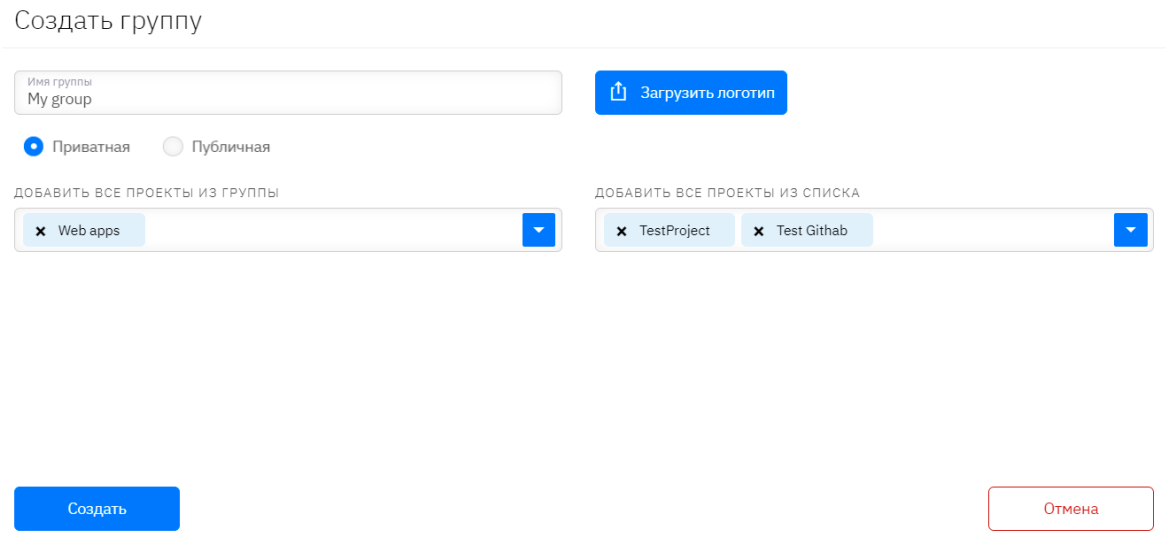

Рис. 5.3: Создание группы проектов

5.2.3.2. Работа с группой проектов

Чтобы перейти к конкретной группе проектов, кликните по её названию в списке групп.

На вкладке **Обзор** представлена общая статистика по сканированиям в группе. Динамику результатов сканирований проектов можно проследить на графиках. В верхней части страницы можно выбрать тип значений (суммарное или среднее) и период для отображения. Сводная информация по сканированиям представлена в таблице **Статистика группы**.

Действия с проектами в группе доступны на вкладке **Проекты**. Здесь можно просмотреть список всех проектов в группе, добавить/удалить проект или поместить/извлечь проект из архива группы. Обратите внимание: при удалении проекта происходит только его удаление из группы, но не из системы.

Управлять группой и правами пользователей в группе можно во вкладке **Настройки**. В подразделе **Управление группой** можно редактировать данные группы, поместить/извлечь группу из архива или удалить группу. В подразделе **Jira** можно привязать группу проектов Solar appScreener к проекту **Jira**.

5.2.3.3. Аналитика

Раздел **Аналитика** предназначен для просмотра общей статистики по всем сканированиям в системе и сравнения результатов анализа по группам.

На вкладке **Результаты** можно проследить динамику результатов сканирований проектов. Добавьте группу к статистике, чтобы сравнить результаты. Для просмотра

информации по всем проектам, выберите **Все проекты** в списке групп. Аналогично разделу **Статистика группы** тип значений и период для отображения можно настроить. Для каждой из групп на графиках отображаются данные:

- количество сканирований (при отображении суммарного значения) или рейтинг (при отображении среднего значения);
- продолжительность сканирований;
- количество уязвимостей (с учётом уровня критичности).

Для просмотра аналитики по сканированиям перейдите на вкладку **Сканирования**. В таблице отображаются данные о количестве проектов в группе, количестве сканирований и их статусах. Чтобы убрать группу из статистики, нажмите на иконку крестика в конце строки нужной группы.

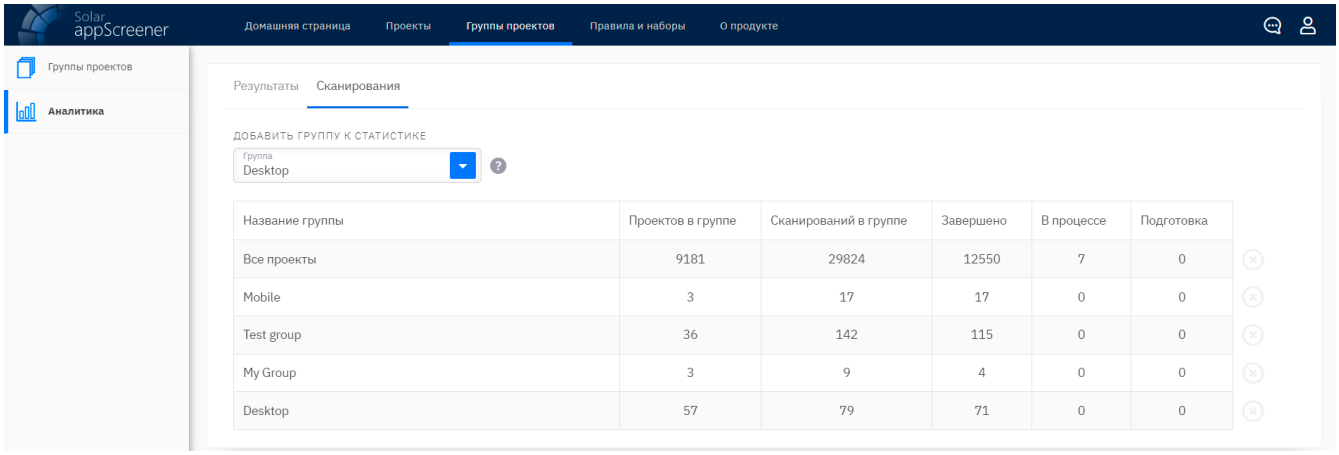

Рис. 5.4: Просмотр статистики по группам проектов

5.2.4. О продукте

Страница **О продукте** служит для предоставления пользователю общей информации о работе с Solar appScreener. В верхней части страницы можно переключаться между следующими разделами:

- Инструкция;
- Анализ сторонних компонент.

В разделе **Инструкция** представлено краткое описание запуска анализа. Из этого раздела можно скачать руководство пользователя и включить/отключить отображение подсказок в интерфейсе.

5.2.5. Личный кабинет

Личный кабинет (рис. 5.5) открывается при наведении курсора на иконку в правом углу верхнего меню. Появляется выпадающее меню с пунктами: **Профиль**, **Настройки**, **Выйти**.

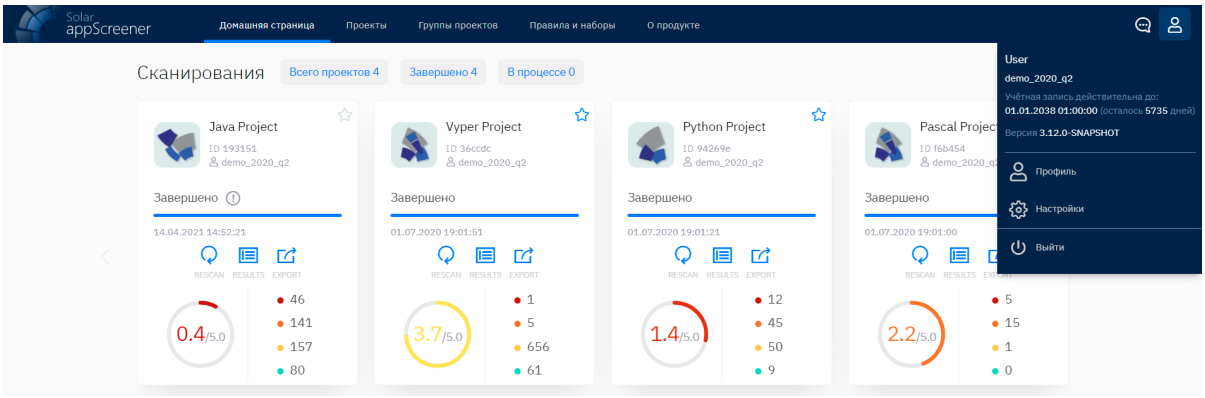

Рис. 5.5: Личный кабинет

5.2.5.1. Профиль

В разделе **Профиль** можно выполнить следующие действия:

- ознакомиться с информацией об учётной записи;
- ознакомиться с информацией об ограничениях лицензии;
- настроить оповещения;
- выбрать язык интерфейса.

Если вы хотите получать почтовые оповещения о завершённых сканированиях, воспользуйтесь переключателем. По желанию к оповещению можно добавить краткую информацию о результатах или текст ошибки в случае, если сканирование будет завершено с ошибкой.

5.2.5.2. Настройки доступа

5.2.5.2.1. Токен и пароль

На вкладке **Токен и пароль** можно получить активный токен авторизации и изменить пароль учётной записи.

Токен авторизации предназначен для аутентификации пользователя при выполнении действий в Solar appScreener без использования UI. Например, чтобы запускать сканирования напрямую через CLT или автоматизировать действия в системе с помощью скриптов. Чтобы получить токен авторизации API:

1. Нажмите **Создать токен**.
2. Введите пароль учётной записи.
3. Укажите время действия токена.
4. Нажмите **Получить активный токен**.

Токен авторизации появится в соответствующем поле. Ознакомиться с информацией о всех активных токенах можно в таблице.

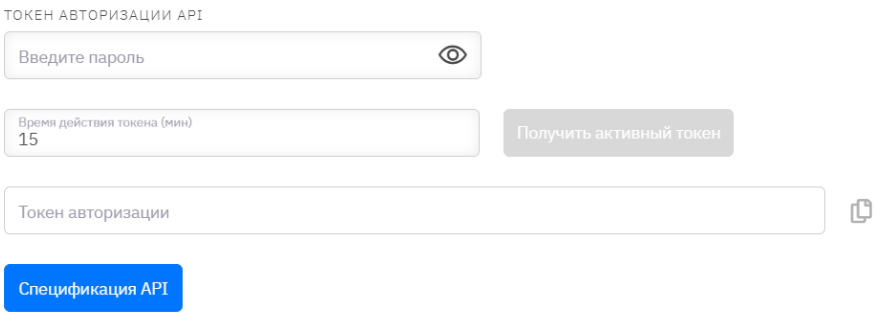

Рис. 5.6: Токен авторизации API

В соответствии с требованиями информационной безопасности **пароль** учётной записи должен регулярно обновляться. Незадолго до истечения срока действия текущего пароля вы получите уведомление.

Для **смены пароля:**

1. Укажите текущий пароль.
2. Укажите новый пароль и повторите его в следующем текстовом поле.
3. Нажмите **Сохранить**.

По истечению срока действия пароля произойдёт автоматический выход из системы на всех устройствах. Для повторного входа требуется установить новый пароль.

5.2.5.2.2. Таск-менеджер

Для того чтобы привязать аккаунт Jira или ТУРБО Трекинг (рис. 5.7):

1. Выберите нужный таск-менеджер из списка.
2. Введите URL сервера таск-менеджера.
3. Введите логин и пароль от аккаунта таск-менеджера.
4. Если вы выбрали ТУРБО Трекинг, укажите ID базы данных.
5. При необходимости активируйте опцию **Игнорировать самоподписанный сертификат**.
6. Нажмите **Привязать аккаунт**.

В результате этих действий в разделе **Личный кабинет** будет указан привязанный аккаунт таск-менеджера. В этом же разделе можно **Отвязать аккаунт** и **Проверить соединение**.

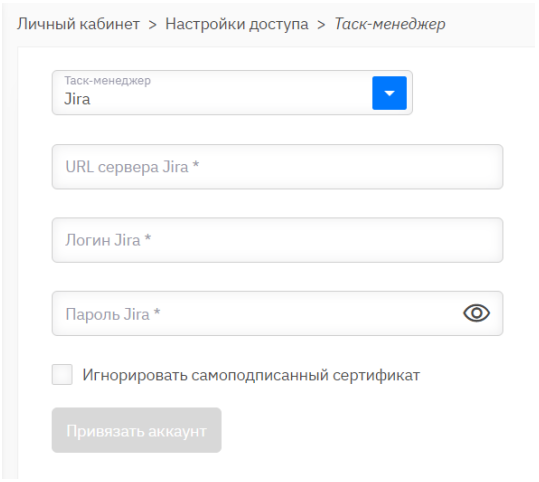

Рис. 5.7: Привязка аккаунта таск-менеджера

5.2.5.2.3. Приватный репозиторий

В разделе **Приватный репозиторий** можно работать с учётными данными, необходимыми для анализа файлов из закрытых репозиториев. Сохранённые в разделе учётные записи можно использовать в различных проектах в системе.

Вы можете добавить/редактировать учётные данные 5 типов:

- логин и пароль - укажите имя пользователя и пароль от ресурса, которому требуется аутентификация;
- токен доступа - предоставьте токен, используемый для авторизации на стороннем ресурсе;
- SSH-ключ - предоставьте приватный SSH-ключ (может быть введён вручную или загружен файлом) и при необходимости отредактируйте конфигурацию SSH клиента (доступно только при выключенном переключателе).

Добавленные учётные записи отображаются в виде списка на соответствующих вкладках. Чтобы отредактировать данные или настроить доступ к ним других пользователей системы, выберите нужную учётную запись из списка.

Данные учётной записи также можно заполнить перед началом сканирования. Выбор опции **Использовать данные при пересканировании проекта** сохранит данные в зашифрованном виде в настройках проекта для последующих сканирований.

5.2.5.3. Настройки системы

5.2.5.3.1. Экспорт отчёта

В подразделе **Экспорт отчёта** можно выполнять действия с шаблонами настроек экспорта отчёта. Шаблоны позволяют в один клик выставлять часто используемую конфигурацию отчёта. На странице можно:

- создать шаблон;
- внести изменения в существующие шаблоны;
- выбрать шаблон по умолчанию.

Для удобной навигации по шаблонам предусмотрена возможность поиска по названию или автору шаблона.

Создание шаблона экспорта отчёта

Чтобы создать шаблон:

1. Нажмите на кнопку **Создать шаблон**. После этого откроется форма создания шаблона экспорта отчёта.
2. Задайте название шаблона.
3. Отметьте чекбокс **Шаблон экспорта по умолчанию**, чтобы использовать выбранный шаблон по умолчанию при генерации отчёта.
4. При необходимости добавьте описание шаблона настроек.
5. Укажите, будет шаблон публичным или приватным. **Публичный шаблон** будет доступен для использования всем пользователям системы. **Приватный шаблон** будет доступен только автору шаблона и администратору системы.
6. Настройте видимость шаблона в списке на странице **Экспорт отчёта** проекта.
7. Шаблон хранит в себе информацию о конфигурации настроек пунктов отчёта. Подробнее о настройках экспорта см. Экспорт отчёта.
8. Нажмите **Сохранить**. После успешного сохранения система вернёт вас в подраздел **Экспорт отчёта**.

Изменение шаблона экспорта отчёта

Чтобы изменить шаблон экспорта:

1. Нажмите на название шаблона в списке. После этого откроется форма изменения шаблона.
2. Внесите желаемые изменения.
3. Нажмите **Сохранить**. После успешного сохранения система вернёт вас в подраздел **Экспорт отчёта** раздела **Настройки**.

Для удаления шаблона экспорта после выполнения шага 2 нажмите кнопку **Удалить**.

5.2.5.3.2. Миграция проектов

В подразделе **Миграция проектов** можно выполнять действия по переносу проектов с одной установки Solar appScreener на другую. К переносу доступны как проекты/группы проектов, так и отдельные сканирования.

Экспорт проектов

Настройте положение переключателя на проекты или отдельные сканирования и выберите элементы для экспорта.

Для **групп проектов** в архив будет добавлена следующая информация:

- группа проектов с входящими в неё проектами;

- настройки проектов (настройки приватного репозитория экспортируются в настройках проекта (ключи в разделе **Личный кабинет** > **Настройки доступа** не сохраняются); права пользователей не экспортируются);
- сканирования (архивированные сканирования не экспортируются);
- информация о сканированиях с файлом приложения, если при запуске была выбрана настройка **Сохранить загруженный файл**, и журналами событий (автор сканирования не сохраняется);
- базы SCA, которые были нарушены уязвимостями данных проектов (только для проектов SCA).

Для **проектов** в архив будет добавлена следующая информация:

- проект;
- настройки проекта (настройки приватного репозитория экспортируются в настройках проекта (ключи в разделе **Личный кабинет** > **Настройки доступа** не сохраняются); права пользователей не экспортируются);
- сканирования (архивированные сканирования не экспортируются);
- информация о сканированиях с файлом приложения, если при запуске была выбрана настройка **Сохранить загруженный файл**, и журналами событий (автор сканирования не сохраняется);
- базы SCA, которые были нарушены уязвимостями данных проектов (только для проектов SCA).

Для **сканирований** в архив будет добавлена следующая информация:

- сканирование;
- информация о сканировании с файлом приложения, если при запуске была выбрана настройка **Сохранить загруженный файл**, и журналами событий (автор сканирования не сохраняется);
- базы SCA, которые были нарушены уязвимостями данных проектов (только для проектов SCA).

Обратите внимание: Базы SCA переносятся таким образом, что их результаты отображаются только в импортированных проектах. Выбрать их для сканирования новых проектов не получится.

Импорт проектов

Чтобы импортировать проекты на новую установку Solar appScreener:

1. В разделе **Миграция проектов** > **Импорт проектов** загрузите архив, экспортированный с другой установки Solar appScreener.
2. Введите пароль шифрованного архива.
3. Выберите нужные файлы для импорта на новую установку.
4. Если в импортируемом архиве находятся отдельные сканирования, выберите проект на новой установке, в который будут помещены импортируемые сканирования.
5. Нажмите **Импорт**.

6. АНАЛИЗ СТОРОННИХ КОМПОНЕНТ

В Solar appScreener реализована возможность сканировать приложения с целью выявления уязвимых компонент и зависимостей в open-source библиотеках в режиме **Анализа сторонних компонент** (Open Sourse Analysis, OSA).

Сторонние библиотеки можно проверить на наличие уязвимостей, а также наличие лицензионных и Supply chain рисков.

6.1. Создание проекта

В интерфейсе Solar appScreener реализованы следующие способы создания проекта:

- запуск сканирования приложения, загруженного с локального компьютера;
- запуск сканирования приложения по ссылке на репозиторий;
- запуск сканирования приложения по SBOM файлу;
- создание пустого проекта, у которого нет сканирований.

6.1.1. Создание пустого проекта

Чтобы создать пустой проект, введите название и нажмите **Создать проект**. При необходимости нажмите **Показать настройки** и установите настройки анализа для будущих сканирований. Подробнее про настройки анализа в разделе Общие.

В созданном проекте можно настроить интеграции. Подробнее про интеграции Solar appScreener в разделе Автоматическое сканирование.

6.2. Запуск сканирования

Чтобы запустить новое сканирование в UI:

1. На Домашней странице, перейдите на вкладку **Анализ сторонних компонент**.
2. В блоке **SBOM файл** загрузите проект в виде SBOM файла в формате Cyclone DX (архив с SBOM файлом или ссылка на SBOM файл в репозитории приведет к ошибке сканирования).

   Solar appScreener может автоматически собрать SBOM файл из исходного кода для проектов, написанных на JavaScript/TypeScript, PHP, Python, Ruby, C#/VB.NET, C/C++/Objective-C, GO, Java/Kotlin/Scala, Rust, Swift, Erlang.
3. В настройках выберите необходимые типы анализа:
   - SCA — для поиска уязвимостей;
   - Supply chain анализ — для оценки рейтинга здоровья используемых библиотек на основе 8 метрик безопасности, отслеживания риска атак MavenGate, Starjacking или Typosquating;
   - Анализ лицензионных рисков — для проверки лицензионных политик.
4. При необходимости дополнительно настройте анализ (подробнее о **Настройках** в разделе Настройки).

5. Нажмите **Начать сканирование**.

Для успешной сборки SBOM необходимо, чтобы архив или репозиторий (Git, Subversion) содержали как исходный код, так и соответствующие манифесты. В Solar appScreener реализованы следующие способы сборки SBOM по исходному коду:

- **Онлайн** — если Solar appScreener имеет доступ в Интернет. Обратите внимание, что при онлайн-генерации SBOM-файла все необходимые компоненты загружаются из Интернета, что несёт в себе потенциальные риски для безопасности вашей системы. Чтобы собрать SBOM-файл из исходного кода, необходимо открыть доступ к следующим источникам:
  - https://www.ya.ru/
  - https://www.npmjs.com
  - https://github.com/
  - https://maven.pkg.github.com/
  - https://pkg.go.dev/
  - https://cocoapods.org/
  - https://packagist.org/
  - https://crates.io/
  - https://rubygems.org/
  - https://rebar3.org/

В зависимости от проекта Solar appScreener посетит один или несколько источников.

- **Автономный режим** — если Solar appScreener не имеет доступа к Интернету. В этом случае генератор собирает SBOM-файл только по тем компонентам, данные о которых уже доступны в его локальной базе данных.

6.3. Запуск комбинированного анализа SAST/OSA

Для проектов, написанных на **JavaScript/TypeScript**, **Python**, **C#** или **Java**, вы можете предоставить исходный код проекта для запуска комбинированного анализа SAST/OSA. Такой анализ необходим для сокращения времени на обработку уязвимых зависимостей. Это позволяет исключить недостижимые уязвимости, тем самым сокращая их общее количество.

Процесс SAST/OSA начинается с поиска известных уязвимостей среди компонент проекта (OSA). Затем включается модуль статического анализа (SAST), который определяет конкретные уязвимые импорты и вызовы функций сторонних компонент. В результате анализа строится граф вызовов импортов и уязвимых функций зависимостей, используемых в проекте, что явно демонстрирует достижимость той или иной уязвимости. Перемещаясь по элементам графа, можно отследить цепочку вызовов уязвимости от транзитивных зависимостей до основного кода и подтвердить или опровергнуть актуальность уязвимости для проекта.

На данный момент комбинированный модуль (SAST/OSA) поддерживает следующие языки:

1. **JS**/**TS**, **Python**: анализируются как директивные, так и транзитивные уязвимые зависимости.
2. **Java**, **C#**: анализируются только директивные уязвимые зависимости.

Для корректной работы модуля необходимо выполнить следующие подготовительные шаги:

- **JS/TS**, **Python**: Для определения достижимости уязвимостей в директивных и транзитивных зависимостях необходимо передать с основным кодом проекта папку, где будет лежать код сторонних зависимостей. По умолчанию, для Python это папка **venv**, для JS/TS – **node_modules**. Название папки по умолчанию можно изменить в **Общих настройках** при старте сканирования.
- **Java**: Для определения достижимости уязвимостей в директивных зависимостях также требуется папка с зависимостями проекта, в частности `.jar`-файлы. Для этого можно использовать Maven-плагин **dependency:copy-dependencies**, который копирует все зависимости проекта в указанный каталог. По умолчанию, для Java это папка **.m2/repository/**. Название папки по умолчанию можно изменить в **Общих настройках** при старте сканирования.
- **C#**: Для определения достижимости уязвимостей в директивных зависимостях достаточно загрузить на анализ только основной код проекта.

Чтобы настроить комбинированное сканирование:

1. На Домашней странице, перейдите на вкладку **Анализ сторонних компонент**.
2. В блоке **Исходный код** загрузите исходный код проекта в виде файла или ссылки на репозиторий.
3. В блоке **SBOM файл** загрузите SBOM файл проекта (опционально).
4. В настройках анализа активируйте опцию **Комбинированный анализ**.
5. Укажите дополнительные настройки при необходимости.
6. Нажмите **Начать сканирование**.

6.4. Инструкция по сборке SBOM файла

Проекты Swift/Objective-C (cocoapods)

Для проектов, написанных на Swift, Objective-C, можно воспользоваться инструментом **cyclonedx-cocoapods**. Пример команды, с помощью которой можно создать SBOM файл в формате CycloneDX:

```
cyclonedx-cocoapods --path /path/to/project --output /path/to/bom.xml
```

где `--path` путь до проекта, `--output` путь до файла SBOM.

В результате получится файл с расширением `.xml`. Чтобы конвертировать его в `.json`, можно использовать **cyclonedx-cli**. Пример команды для конвертации CycloneDX-XML в CycloneDX-JSON:

```
cyclonedx convert --input-file /path/to/bom.xml --input-format xml --output-file
/path/to/bom.json --output-format json
```

где:

`--input-file` - путь до конвертируемого файла;

`--input-format` - формат исходного файла;

`--output-file` - путь до итогового файла;

`--output-format` - итоговый формат файла.

Обратите внимание: данный генератор не строит дерево транзитивных зависимостей.

Для генерации SBOM файла для многих языков программирования можно воспользоваться инструментом **cdxgen**. Пример команд, с помощью которых можно создать SBOM файл:

Проекты JavaScript

```
cd /path/to/project
```

```
cdxgen -t node.js -o /path/to/sbom.json
```

где `-t` тип проекта, `-o` путь до файла SBOM.

Проекты Java/Scala/Kotlin (Maven/Gradle)

```
cd /path/to/project
```

```
cdxgen -t java -o /path/to/sbom.json
```

Проекты C/C++ (conan)

```
cd /path/to/project
```

```
cdxgen -t c/c++ -o /path/to/sbom.json
```

Обратите внимание: дерево транзитивных зависимостей будет простроено только если зависимости описаны в `conan.lock`.

Проекты PHP (Composer)

```
cd /path/to/project
```

```
cdxgen -t php -o /path/to/sbom.json
```

Проекты Swift (SwiftPM)

```
cd /path/to/project
```

```
cdxgen -t swift -o /path/to/sbom.json
```

Проекты C# (.Net)

```
cd /path/to/project
```

```
cdxgen -t .Net -o /path/to/sbom.json
```

Обратите внимание: дерево транзитивных зависимостей будет сгенерировано только в присутствии файлов `project.assets.json`, `packages.lock.json`.

Проекты на других языках

Список генераторов для разных языков программирования представлен по ссылке. Все генераторы поддерживают формат SBOM CycloneDX.

6.5. Запуск сканирования из командной строки

*Для того чтобы посмотреть раздел **help**, выполните команду: `java -jar clt.jar -help`.*

Запуск сканирования из командной строки доступен только при условии предустановленной Java 11 или новее.

Для создания проекта из командной строки следует выполнить команду:

```
java -jar clt.jar -rest [rest URL] -token [token] scaProjectCreate [options]
```

- **-rest** адрес API (URL).
- **-token** токен авторизации API; можно получить в интерфейсе пользователя в разделе **Личный кабинет** (см. раздел Личный кабинет).

Аргументы (options):

Обязательные аргументы:

- **-name** название проекта.

Для запуска сканирования из командной строки следует выполнить команду:

```
java -jar clt.jar -rest [rest URL] -token [token] scaScanCreate [options]
```

- **-rest** адрес API (URL).
- **-token** токен авторизации API; можно получить в интерфейсе пользователя в разделе **Личный кабинет** (см. раздел Личный кабинет).

Аргументы (options):

Обязательные аргументы:

- **-project** ID проекта.
- **-path** путь к каталогу или файлу для анализа (не должен использоваться, если указан параметр link).
- **-link** URL репозитория (не должен использоваться, если указан параметр path).

Необязательные аргументы:

- **-sca** выполнить SCA анализ (по умолчанию включено).
- **-licenseRisks** выполнить анализ лицензионных рисков (по умолчанию включено).
- **-supplyChain** выполнить Supply Chain анализ (по умолчанию включено).
- **-priority** приоритет выполнения сканирования.
- **-saveFile** полностью сохранить загружаемый файл.

- **-vcs.branch** ветка репозитория, если не master (при анализе приложений по ссылке на репозиторий).
- **-vcs.account** UUID репозитория.
- **-vcs.login** логин репозитория.
- **-vcs.password** пароль репозитория.
- **-vcs.token** токен аутентификации репозитория.
- **-vcs.tokenId** UUID токена аутентификации.
- **-vcs.sshKey** путь до ключа SSH репозитория.
- **-vcs.sshKeyId** UUID ключа SSH.
- **-vcs.sshDefault** использовать стандартную конфигурацию SSH-клиента. Если значение false, то указать конфигурацию SSH-клиента.
- **-vcs.saveCredentials** использовать учётные данные при пересканировании.
- **-iconPath** путь к логотипу проекта.

Пример:

```
java -jar clt.jar -rest [rest_url] -token [token] scaScanCreate
-link [link] -project PROJECT
```

Для проверки статуса сканирования из командной строки следует выполнить команду:

```
java -jar clt.jar -rest [rest URL] -token [token] scaScanInfo [options]
```

- **-rest** адрес API (URL).
- **-token** токен авторизации API; можно получить в интерфейсе пользователя в разделе **Личный кабинет** (см. раздел Личный кабинет).

Аргументы (options):

- **-scan** ID сканирования.

Пример:

```
java -jar clt.jar -rest [rest_url] -token [token] scaScanInfo -scan SCAN
```

Для экспорта отчёта в формате PDF из командной строки следует выполнить команду:

```
java -jar clt.jar -rest [rest URL] -token [token] scaExport [options]
```

Для экспорта отчёта в формате CSV из командной строки следует выполнить команду:

```
java -jar clt.jar -rest [rest URL] -token [token] scaExport [options]
'-general.format' CSV
```

- **-rest** адрес API (URL).
- **-token** токен авторизации API; можно получить в интерфейсе пользователя в разделе **Личный кабинет** (см. раздел Личный кабинет).

Аргументы (options):

Аргументы включаются в отчёт, если они принимают значение **true**. По умолчанию аргументы принимают значение **false**, если не указано иное.

Обязательные аргументы:

- **-path** путь до директории, куда следует поместить отчёт.
- **-project** ID проекта.

ID проекта можно получить в боковом меню проекта в интерфейсе. Справа от логотипа проекта отображается ID (первые шесть символов ID проекта). Чтобы скопировать в буфер полный ID, нажмите на иконку .

Опциональные аргументы:

- **-scans** ID сканирований, которые следует включить в отчёт. Если сканирований несколько, то перечислять их следует через запятую.
- **-filter.vulnerabilities** включить в отчёт уязвимости (включается по умолчанию).
- **-filter.critical** включить в отчёт уязвимости критического уровня (включается по умолчанию).
- **-filter.medium** включить в отчёт уязвимости среднего уровня (включается по умолчанию).
- **-filter.low** включить в отчёт уязвимости низкого уровня.
- **-filter.info** включить в отчёт уязвимости информационного уровня.
- **-filter.licenseRisks** включить в отчёт лицензионные риски (включается по умолчанию).
- **-filter.supplyChain** включить в отчёт Supply chain риски (включается по умолчанию).
- **-filter.vulnerableDependencies** включить в отчёт компоненты с уязвимостями (включается по умолчанию).
- **-filter.safeDependencies** включить в отчёт компоненты без уязвимостей.
- **-filter.tasks** включить в отчёт уязвимости с созданными задачами в таск-менеджере (включается по умолчанию).
- **-fuzzy.included** включить в отчёт данные FLE. При включении укажите:
  - **-fuzzy.critical** настройка значения FLE для критических уязвимостей (по умолчанию 0).
  - **-fuzzy.medium** настройка значения FLE для уязвимостей среднего уровня (по умолчанию 0).
  - **-fuzzy.low** настройка значения FLE для уязвимостей низкого уровня (по умолчанию 0).
  - **-fuzzy.info** настройка значения FLE для уязвимостей информационного уровня (по умолчанию 0).
- **-general.contents** включить оглавление (включается по умолчанию).
- **-general.format** формат отчёта. Принимает значения: PDF, CSV, DOCX, JSON, HTML. По умолчанию принимает значение PDF.
- **-general.included** включить настройки экспорта (включается по умолчанию).
- **-general.locale** настроить язык отчёта (по умолчанию английский).
- **-general.logo** использовать пользовательский логотип. По умолчанию не используется. Если пользователь хочет использовать собственный логотип, необходимо указать путь до логотипа.
- **-general.statuses** отображать статусы уязвимостей (по умолчанию выключено).
- **-projectInfo.scanHistory** количество последних сканирований в отчёте (по умолчанию 0):
  - -1— не выгружать историю сканирований;
  - 0 — выгрузить всю историю сканирований;
  - >0 — число последних сканирований.
- **-projectInfo.securityDynamic** включить динамику уровня безопасности (включается по умолчанию).

- **-projectInfo.vulnerabilityDynamic** включить динамику количества уязвимостей (включается по умолчанию).
- **-results.included** включить подробные результаты (включается по умолчанию). При включении укажите:
  - **-results.action** включить действия с уязвимостями (включается по умолчанию).
  - **-results.comment** включить комментарии к уязвимостям (включается по умолчанию).
  - **-results.entryNum** настроить количество вхождений уязвимости (по умолчанию 0):
    - -1 — не выгружать вхождения;
    - 0 — выгрузить все вхождения;
    - >0 — выгрузить не более чем вхождений.
  - **-results.statuses** включить уязвимости со статусами (указать статусы). По умолчанию все уязвимости, кроме отклоненных.
  - **-results.taskInfo** включить информацию о задачах в таск-менеджере (включается по умолчанию).
  - **-results.dependencies** включить метки зависимостей (включается по умолчанию).
  - **-results.dependencyTree** включить дерево зависимостей компоненты.
- **-scanInfo.included** включить настройки сканирования (включается по умолчанию). При включении укажите:
  - **-scanInfo.foundChart** включить диаграмму найденных уязвимостей (включается по умолчанию).
  - **-scanInfo.settings** включить настройки сканирования (включается по умолчанию).
  - **-scanInfo.dependenciesChart** включить диаграмму найденных зависимостей (включается по умолчанию).
- **-table.included** включить список уязвимостей (включается по умолчанию). При включении в отчёт укажите:
  - **-table.entryNum** список вхождений уязвимости в списке уязвимостей (по умолчанию 0):
    - -1 — не выгружать вхождения;
    - 0 — выгрузить все вхождения;
    - >0 — выгрузить введённое количество вхождений.
  - **-table.statuses** включить уязвимости со статусами (указать статусы). По умолчанию все уязвимости, кроме отклоненных.
  - **-table.dependencyTree** включить полное дерево зависимостей.
- **-comparison.included** включать ли в отчёт сравнение с предшествующим сканированием. При включении в отчёт укажите:
  - **-comparison.ScanId** ID сканирования для сравнения (обязательный параметр).
  - **-comparison.fixed** включить в отчёт устранённые уязвимости.
  - **-comparison.newIssue** включить в отчёт новые уязвимости (включается по умолчанию).
  - **-comparison.saved** включить в отчёт сохранившиеся уязвимости (включается по умолчанию).
  - **-comparison.entryNum** включить количество вхождений уязвимости (по умолчанию 0):
    - -1 — не выгружать вхождения;
    - 0 — выгрузить все вхождения;
    - >0 — выгрузить введённое количество вхождений.
  - **-comparison.scanSettings** включить в отчёт настройки сравнения сканирований (включается по умолчанию).

Пример:

```
java -jar clt.jar -rest http://<installation_address>/app/api/v1 -token
kljkjljlkjljklkjk scaExport -scan ec59395b-4372-47b1-95a2-4d48b044ff0b
-path C:\test -default
```

Важно обратить внимание:

Раздел **Информация о сканировании** не будет включён в отчёт, если вы не укажете ID необходимого сканирования (аргумент **-scans**). ID сканирования можно получить в информации о сканировании. Чтобы скопировать в буфер ID сканирования, нажмите на иконку .

Описанная выше функциональность также доступна через REST.

6.6. Управление проектом

Управление проектом состоит из разделов **Обзор**, **Подробные результаты**, **Сканирования**, **Экспорт отчёта**, **Сравнение сканирований** и **Настройки**. Переключение между этими разделами осуществляется через меню в левой части страницы.

Справа от логотипа проекта отображается ID (первые символы UUID проекта). Чтобы скопировать в буфер полный UUID, нажмите на .

На страницу **Обзор** > **Сводные результаты** можно перейти, нажав на название проекта на странице **Проекты** в разделе **OSA** или на **Домашней странице** (если проект входит в шесть последних запущенных проектов).

На страницы **Подробные результаты** или **Экспорт отчёта** можно перейти, нажав на соответствующие кнопки быстрой навигации на странице **Проекты** или на **Домашней странице** (если проект входит в шесть последних запущенных проектов).

6.6.1. Обзор

В разделе **Обзор** содержится статистика сканирования и интерактивное дерево зависимостей. Сканирование, для которого будут отображаться данные, можно выбрать в правом верхнем углу страницы. Нажмите на иконку , чтобы отобразились параметры запуска анализа для выбранного сканирования.

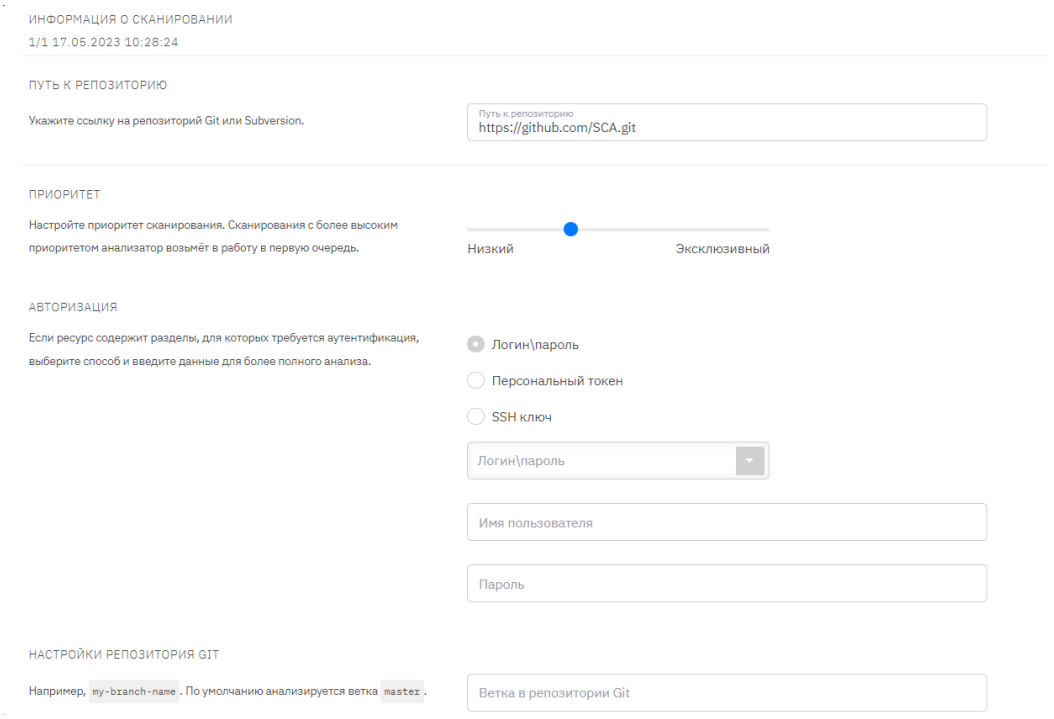

Рис. 6.1: Параметры запуска анализа

На вкладке **Сводные результаты** представлена следующая информация:

- рейтинг;
- статус сканирования;
- продолжительность сканирования;
- общее количество компонент;
- количество уязвимых компонент;
- графическая информация по сканированию и проекту:
  - диаграмма с количеством уязвимостей каждого уровня критичности в сканировании;
  - график уровня безопасности проекта;
  - график количества уязвимостей в проекте;
  - диаграмма с наиболее уязвимыми компонентами;
  - график с типами уязвимостей.

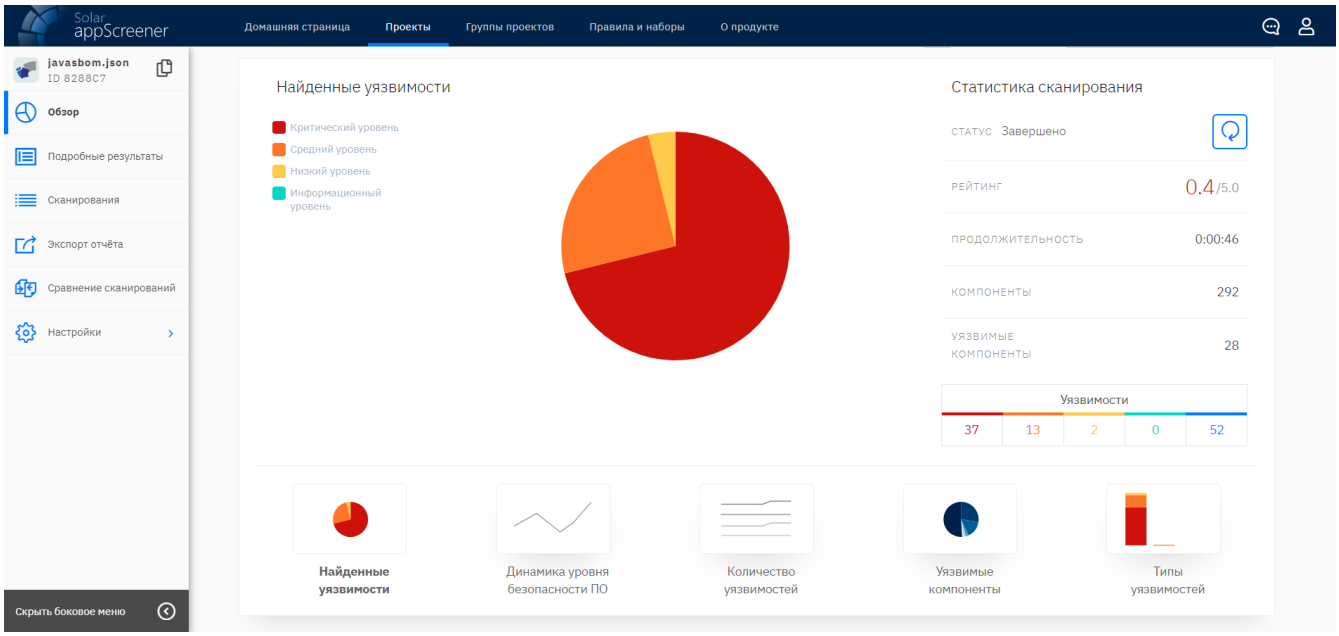

Рис. 6.2: Сводные результаты

Если в данный момент приложение не сканируется, можно запустить новое сканирование, нажав на иконку . Если сканирование находится в процессе анализа, его можно остановить, нажав на иконку .

На вкладке **Дерево зависимостей** представлен интерактивный граф зависимостей всех компонент в проекте. Чтобы скрыть ветку графа, нажмите на **-** на ребре. Чтобы отменить действие, нажмите на **+**.

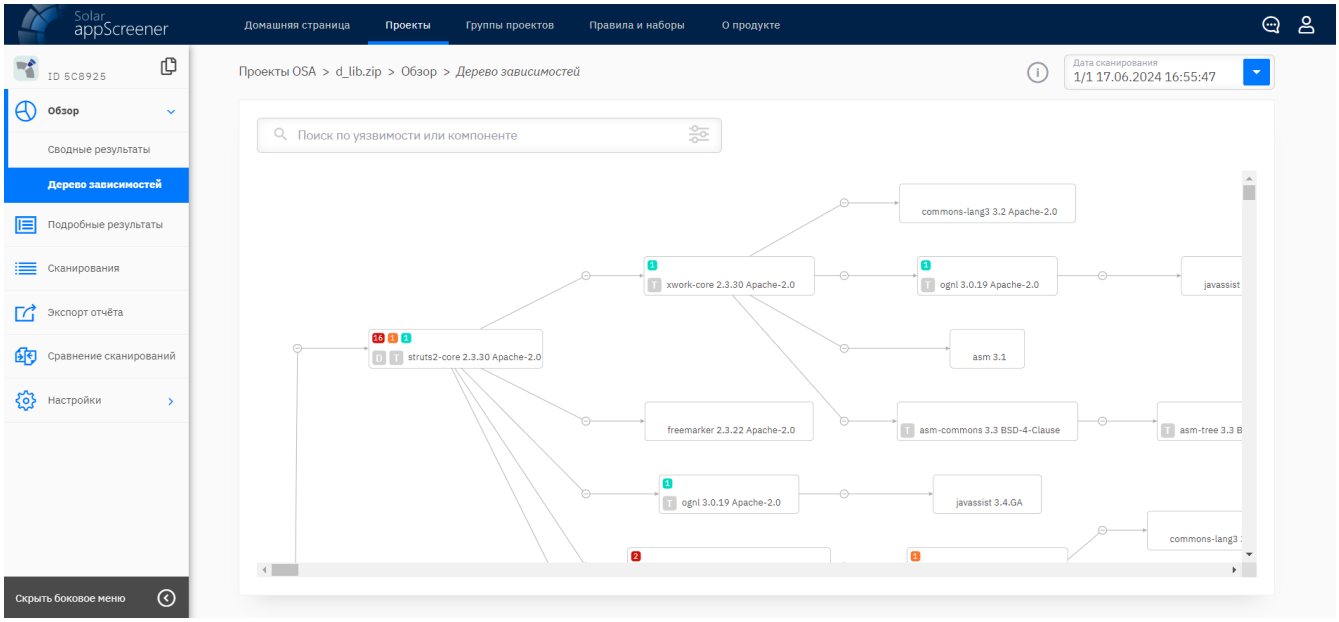

Рис. 6.3: Дерево зависимостей

Для каждой компоненты указывается название, версия и лицензия, а также уязвимости при наличии. Список уязвимостей можно развернуть по клику на компоненту. Клик по

уязвимости осуществляет переход к данной уязвимости на странице **Подробные результаты**.

Для удобной навигации предусмотрен поиск по названию уязвимости или компоненте, а также фильтры.

6.6.2. Подробные результаты

На вкладке **Подробные результаты** отображается информация по каждой из обнаруженных уязвимостей для выбранного сканирования. Переключаться между результатами разных сканирований можно с помощью списка сканирований в правом верхнем углу.

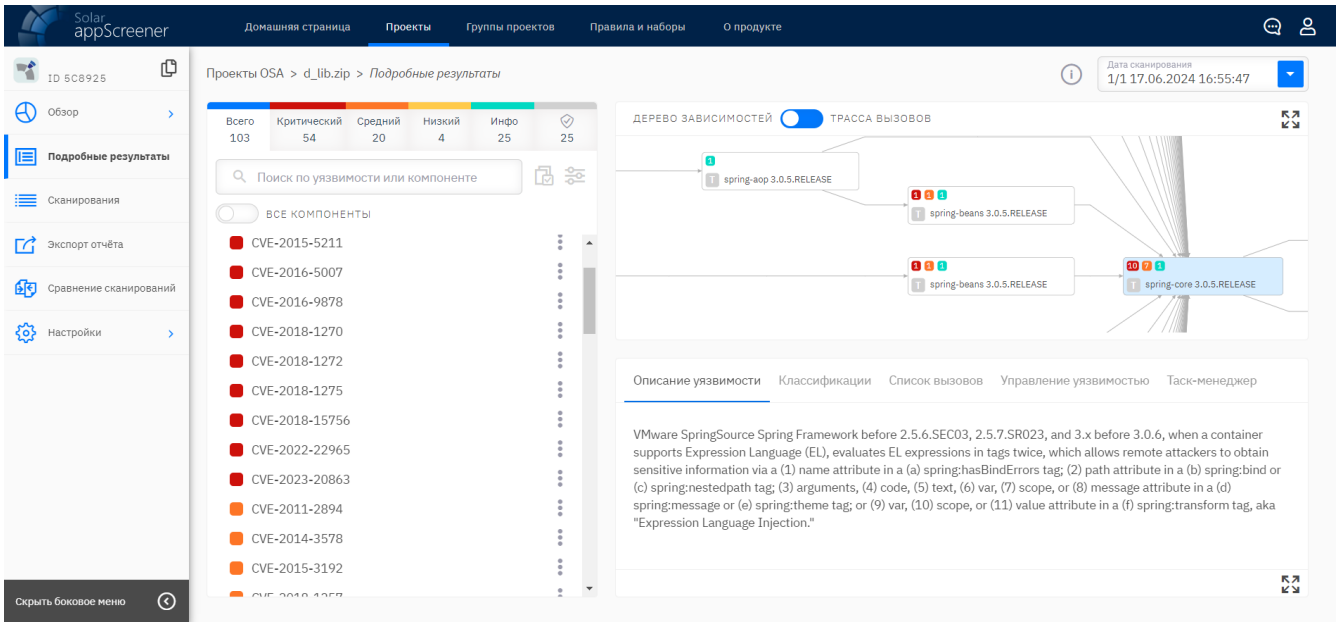

Рис. 6.4: Подробные результаты

В левой части страницы представлен список вхождений уязвимостей, сгруппированный по названию библиотек и версий. Если компонента содержит зависимости, они будут отмечены соответствующими тэгами: **D** для прямых, **T** для транзитивных зависимостей. Связанные зависимости отображаются по наведению курсора на тэг.

Если компонента содержит **Supply chain** или **лицензионные** риски, они будут отмечены соответствующими тэгами: **S** для Supply chain, **L** для лицензионных рисков. Чтобы ознакомиться с подробной информацией о рисках, нажмите на название библиотеки в списке компонент и на название риска в открывшемся списке вхождений.

В верхнем меню можно выбрать, уязвимости какого уровня требуется отобразить. Для удобной навигации по уязвимостям предусмотрен поиск по названию уязвимости или компоненте, а также фильтры (рис. 6.5).

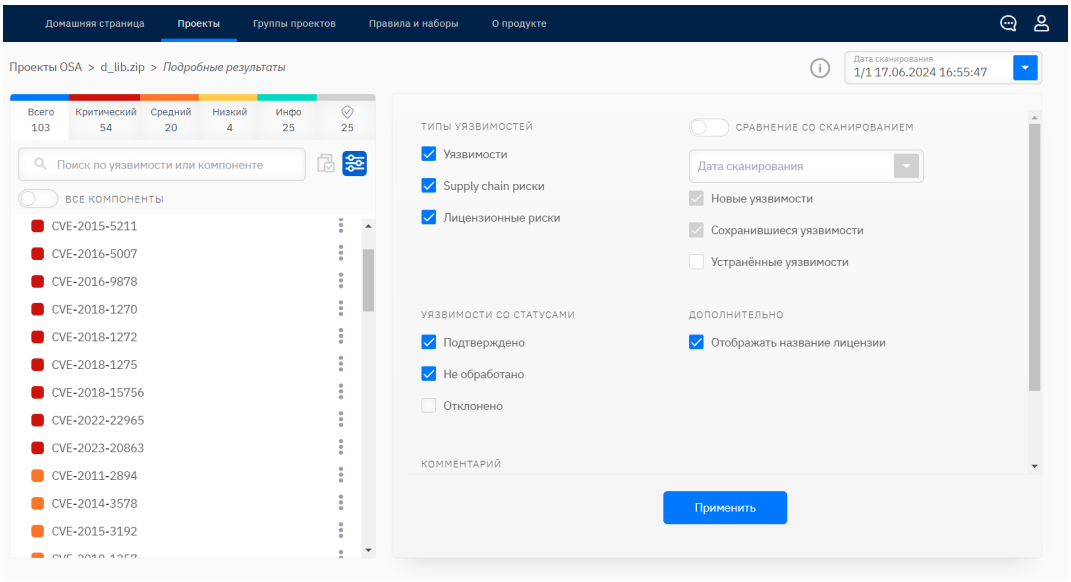

Рис. 6.5: Фильтры результатов

Фильтровать результаты можно по следующим параметрам:

- типы уязвимостей:
  - уязвимости;
  - supply chain риски;
  - лицензионные риски.
- статусы уязвимостей для отображения:
  - подтверждено;
  - не обработано;
  - отклонено.
- включить Fuzzy Logic Engine с одним из режимов:
  - только истинные — отображаются уязвимости с высокой вероятностью того, что вхождения являются реальной уязвимостью;
  - только важные — отображаются уязвимости, на которые надо обратить внимание в первую очередь;
  - пользовательский — позволяет настроить чувствительность Fuzzy Logic Engine, перемещая ползунок в разные положения. Крайнее левое положение ползунка характеризует вхождения с самой высокой вероятностью корректного срабатывания, крайнее правое отобразит уязвимости для любой вероятности;
  - динамический — позволяет настроить значение EPSS, перемещая ползунок в разные положения. Крайнее левое положение ползунка (1) характеризует вхождения с самой высокой вероятностью эксплуатации, крайнее правое (0) отобразит уязвимости для любой вероятности.
- наличие комментария:
  - с комментариями;
  - без комментариев.
- при наличии двух и более успешных сканирований в проекте, можно сравнить текущее сканирование с одним из предшествующих и отобразить уязвимости в соответствии с их статусом. Для этого выберите соответствующие настройки:
  - новые уязвимости — новые уязвимости, по отношению к выбранному из списка сканированию;
  - сохранившиеся уязвимости — уязвимости, обнаруженные в выбранном из списка

сканировании и в текущем сканировании;
- устранённые уязвимости — уязвимости, обнаруженные в выбранном из списка сканировании, но не обнаруженные в текущем сканировании.

Фильтры применяются после нажатия на кнопку **Применить**.

Нажмите на три точки рядом с названием уязвимости, чтобы изменить критичность и статус. При изменении статуса и уровня критичности уязвимости пересчитывается уровень безопасности приложения. Уязвимости со статусом **Отклонено** не учитываются при подсчёте количества уязвимостей и рейтинга безопасности. При пересканировании изменения сохраняются.

После выбора конкретной уязвимости в центральной части страницы отображается соответствующее ей дерево зависимостей компоненты. В нижней части представлена следующая информация (рис. 6.6): **Описание уязвимости**, **Рекомендации**, **Ссылки**, **Классификации**, **Управление уязвимостью**, **Таск-менеджер**.

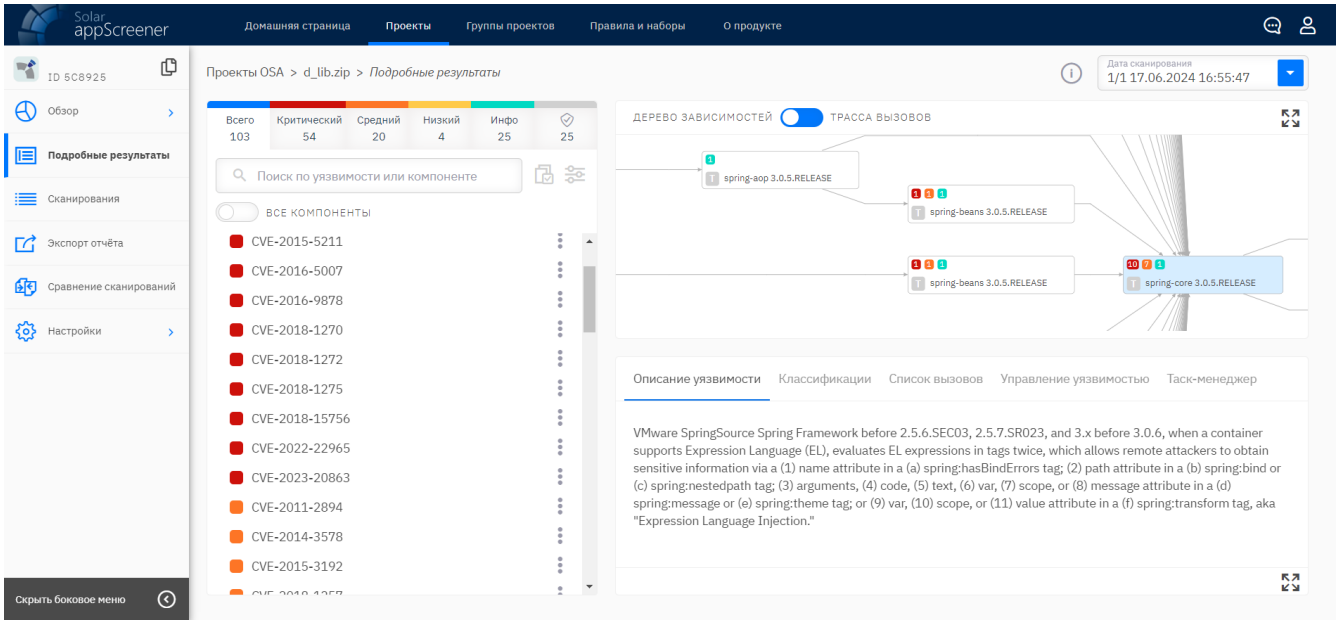

Рис. 6.6: Свойства уязвимости

В случае, когда вместе с SBOM файлом был просканирован исходный код проекта, список вызовов и импортов компоненты будет доступен на вкладке **Список вызовов**. Для просмотра необходимо выбрать элемент на трассе вызовов. Переключение между режимами просмотра частичного дерева зависимостей и трассы вызовов происходит с помощью переключателя над областью с деревом.

Для рисков Supply chain предоставляется **Описание** с оценкой по метрикам, **Управление уязвимостью**, **Таск-менеджер**.

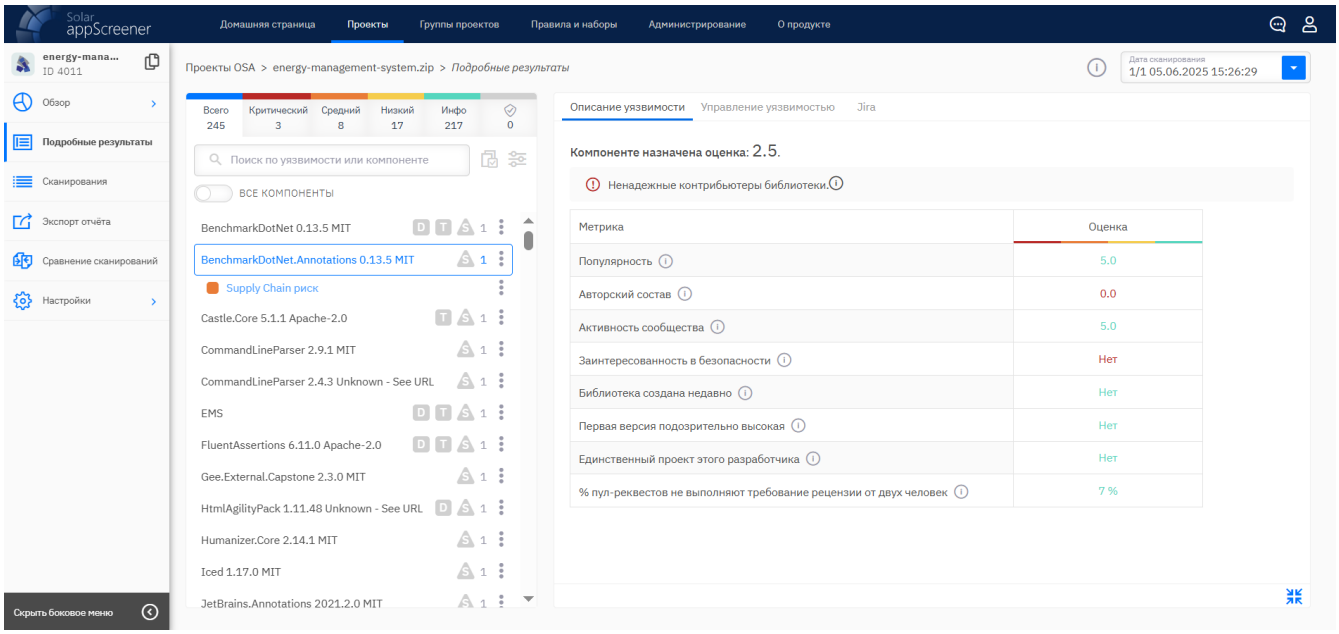

Рис. 6.7: Оценка компоненты

Для лицензионных рисков предоставляется **Описание**, **Управление уязвимостью**, **Таск-менеджер**.

На вкладке **Управление уязвимостью** (рис. 6.8) можно изменить уровень критичности и статус, добавить комментарий к уязвимости и посмотреть журнал событий с оставленными ранее комментариями и действиями.

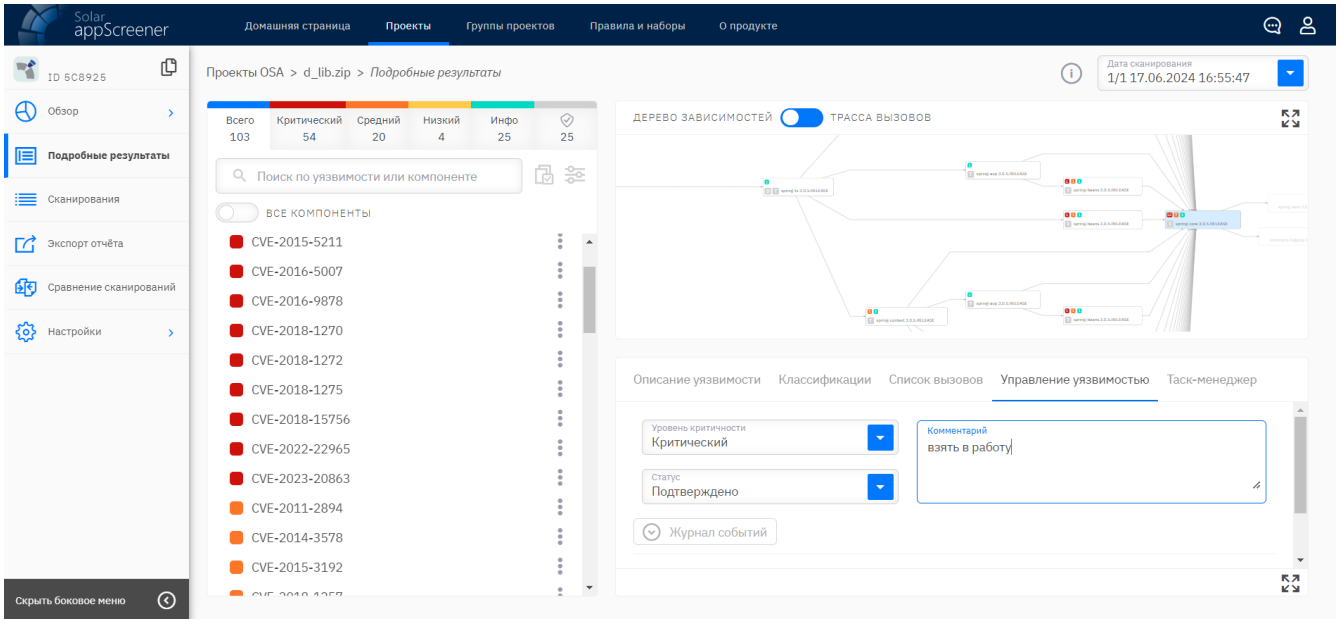

Рис. 6.8: Управление уязвимостью

На вкладке также отображаются EPSS и CVSS оценки для вхождения. Они показывают уверенность в том, что данная уязвимость может быть проэксплуатирована: чем выше оценка, тем меньше вероятность, что вхождение ложно-положительное.

6.6.3. Сканирования

Раздел **Сканирования** предназначен для управления сканированиями в рамках одного проекта. Для каждого сканирования отображаются следующие данные:

- дата и время сканирования, при нажатии на иконку отображается информация о параметрах запуска анализа;
- меню действий:
  - выгрузить отчёт;
  - архивировать сканирование;
  - удалить сканирование.
- статус сканирования;
- продолжительность сканирования;
- общее количество компонент;
- количество уязвимых компонент;
- количество уязвимостей критического, среднего, низкого и информационного уровня;
- рейтинг приложения.

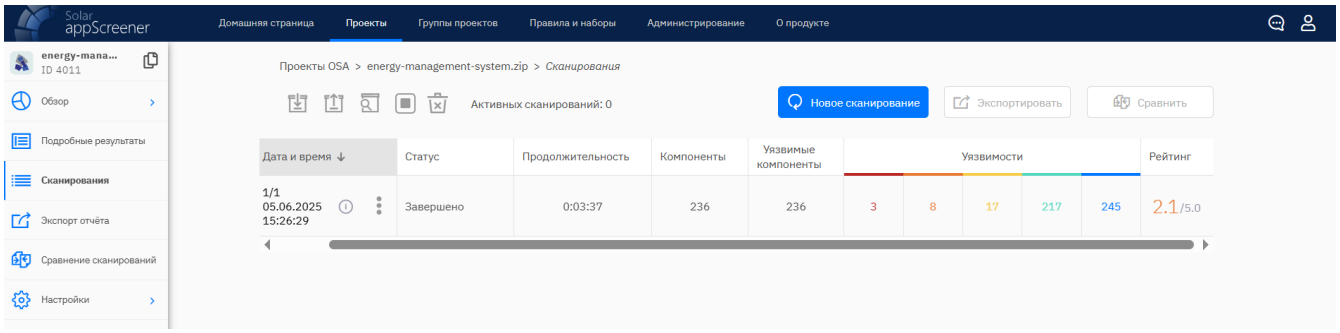

Рис. 6.9: Сканирования

Список можно сортировать по дате сканирования, продолжительности сканирования или рейтингу, общему количеству компонент или уязвимым компонентам. Для этого нажмите на соответствующий заголовок, повторное нажатие меняет порядок сортировки.

Сравнить результаты двух выбранных сканирований можно, нажав на кнопку **Сравнить**. Сканирования, которые находятся в архиве, можно скрыть из списка, нажав на **Скрыть архив**, или отображать в списке, нажав на **Показать архив**.

Для проведения повторного сканирования в рамках одного проекта нажмите **Новое сканирование**.

В Solar appScreener можно запустить сразу несколько сканирований в одном проекте с разными настройками. Отслеживать статусы сканирований можно в графе **Статус**.

6.6.4. Экспорт отчёта

В разделе **Экспорт отчёта** можно выгрузить результаты сканирования в отчёт в формате PDF, HTML, JSON, CSV или DOCX. Выберите один из готовых шаблонов настроек или задайте информацию для экспорта вручную.

Настройки отчёта включают следующие блоки:

- сканирования;

- сравнить со сканированием;
- информация о проекте;
- информация о сканировании;
- фильтр уязвимостей;
- список уязвимостей;
- подробные результаты;
- общие настройки отчёта.

Сканирования

Для экспорта отчёта выберите одно или несколько сканирований. Чтобы получить только сводную информацию по проекту, удалите все сканирования из списка.

Сравнить со сканированием

Выберите одно сканирование, чтобы опция **Сравнить со сканированием** стала доступна. В отчёт будут включены таблица сравнения, график и статистика по новым, сохранившимся и устранённым уязвимостям.

Выберите статусы уязвимостей (новые, сохранившиеся и/или устранённые) и укажите количество вхождений каждой уязвимости.

Информация о проекте

В отчёт можно включить динамику уровня безопасности и историю сканирований.

Информация о сканировании

По умолчанию будет добавлена статистика сканирования: статус, рейтинг, продолжительность, количество компонент и уязвимостей.

Выберите дополнительную информацию о сканировании:

- диаграмма найденных уязвимостей;
- диаграмма уязвимых компонент;
- настройки запуска сканирования.

Фильтр уязвимостей

Выберите уязвимости по уровню критичности и типу, а также компоненты для отображения. Воспользуйтесь фильтром Fuzzy Logic Engine в одном из режимов: **Только истинные**, **Только важные**, **Пользовательский** или **Динамический** режимы.

Список уязвимостей

Выберите статусы уязвимостей и задайте количество их вхождений, а также настройте добавление полного дерева зависимостей.

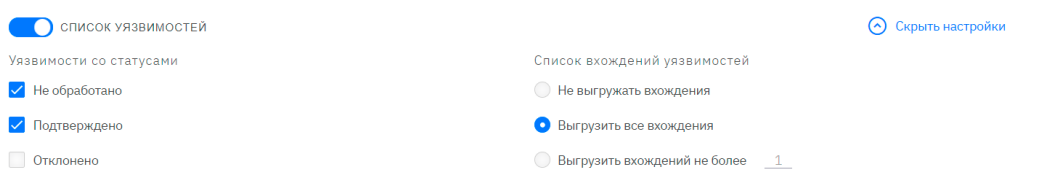

Рис. 6.10: Список уязвимостей

Подробные результаты

По умолчанию для уязвимостей будут добавлены описание, рекомендации по устранению, ссылки. Также можно настроить:

- статусы уязвимостей: **Не обработано**, **Подтверждено**, **Отклонено** (подробнее в разделе Подробные результаты);
- количество уязвимостей компоненты;
- отображение журнала событий:
  - не включать;
  - включить только комментарии;
  - включить только действия;
  - журнал целиком;
- отображение дополнительной информации.

Общие настройки отчёта

Выберите язык, формат отчёта и при необходимости включите в него настройки экспорта и оглавление. Также можно настроить отображение статусов уязвимостей в отчёте и установить пользовательский логотип.

Обратите внимание:

Для корректного отображения данных CSV-отчёта в **Microsoft Excel** необходимо вручную выбрать в выпадающем списке **Обнаружение типов данных** опцию **Не обнаруживать типы данных** во время импорта файла. Настройка отображения статусов уязвимостей недоступна для этого формата.

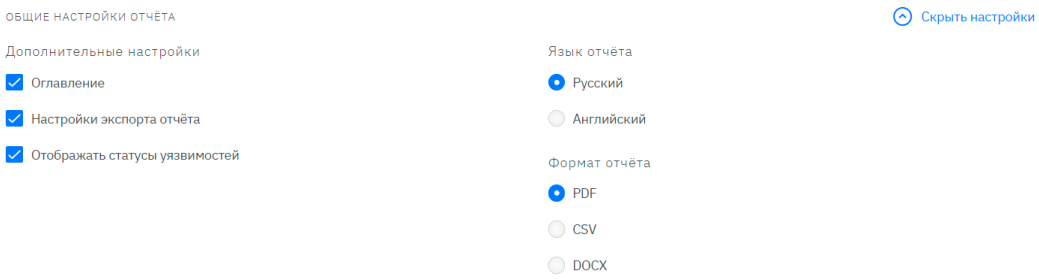

Рис. 6.11: Общие настройки отчёта

Чтобы скачать отчёт, нажмите **Скачать**.

Чтобы отправить отчёт по почте, нажмите **Отправить по e-mail**. В открывшейся форме укажите список адресов получателей и при необходимости отредактируйте текст письма.

6.6.5. Сравнение сканирований

В разделе **Сравнение сканирований** можно производить сравнение результатов сканирований. Чтобы сравнить результаты, выберите два сканирования в верхней части страницы. На странице отобразится количество устраненных, новых и сохранившихся уязвимостей на графике и в таблице. Также будет представлена таблица со сравнением по дате сканирования, продолжительности, общему количеству и количеству уязвимых компонент, количеству уязвимостей с учётом уровня критичности и рейтингу.

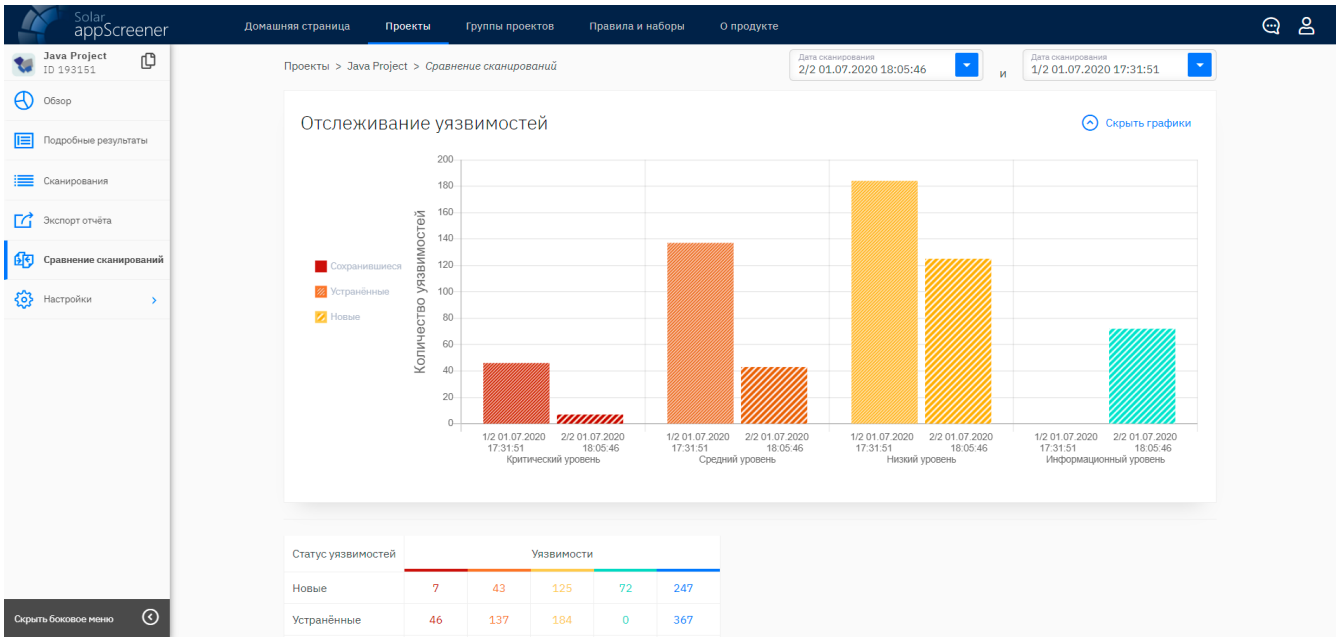

Рис. 6.12: Сравнение сканирований

6.6.6. Настройки

В разделе **Настройки** отображаются настройки проекта. В этом разделе можно работать с сущностями: **Общие**, **Права пользователей**, **Автоматическое сканирование** и **Управление проектом**.

6.6.6.1. Общие

В подразделе **Общие** (рис. 6.13) можно задать настройки для последующих сканирований:

- указать ссылку на репозиторий Git или Subversion;
- выбрать типы анализа: SCA, Supply chain, лицензионных рисков;
- задать приоритет сканирования;
- выбрать агент сканирования, который исполнит анализ;
- настроить сохранение загружаемых на анализ файлов;
- для проектов с исходным кодом (написанных на JS/TS, Java, Python, C#) настроить проведение комбинированного анализа;
- для проектов с исходным кодом (Gradle, Maven) настроить сборку SBOM файла только с зависимостями, напрямую включёнными в проект;
- выбрать способ авторизации и заполнить необходимые данные в случае, если проект загружается на анализ из приватного репозитория;
- указать ветку для сканирования в Git-репозитории.

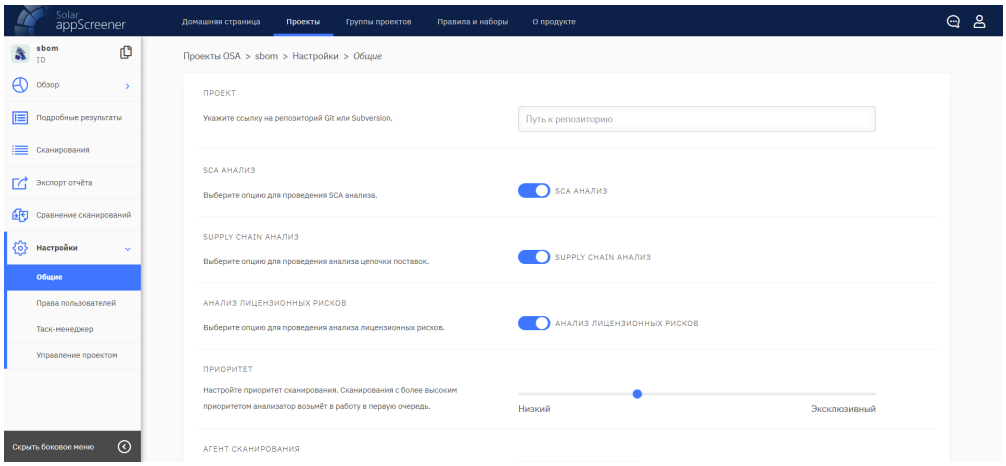

Рис. 6.13: Общие

В подразделе **Права пользователей** можно быстро выдать доступ к проекту другим пользователям системы и настроить их права в проекте.

6.6.6.2. Управление проектом

В подразделе **Управление проектом** можно редактировать данные проекта, а также архивировать или удалить проект. Архивированные проекты продолжают храниться в системе. Чтобы удалить проект без возможности восстановления, нажмите **Удалить проект** и подтвердите действие.

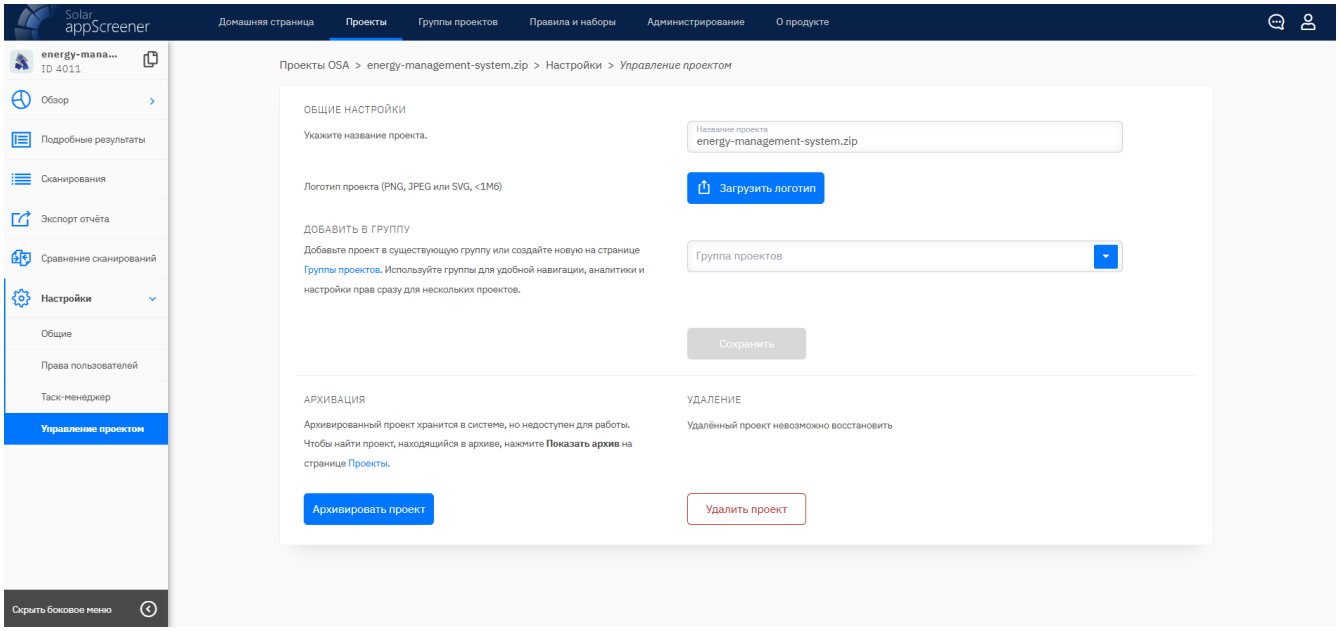

Рис. 6.14: Управление проектом

7. РАБОТА С API

Доступ к функциональности Solar appScreener доступен через API. Спецификация API отвечает стандарту OpenAPI. Веб-интерфейс реализован с помощью Swagger Codegen. Для доступа к API:

1. Перейдите в раздел **Личный кабинет** > **Настройки доступа** > **Токен и пароль**.
2. Нажмите **Спецификация API**.

Чтобы сделать запрос:

1. Нажмите на нужный запрос > **Try it out**.
2. Укажите параметры.
3. Нажмите **Execute**.

Соответствующий cURL-запрос и ответ сервера отобразятся ниже.

Доступ к спецификации в формате JSON может быть получен по протоколу HTTP с использованием GET-запроса **/app/api/v1/openapi.json** по токену авторизации API. Например:

```
curl -H 'Authorization: Bearer <token>'
<installation_address>/app/api/v1/openapi.json -o openapi.json
```

В результате в текущей директории будет сохранён файл **openapi.json**, который далее может быть передан инструментальному средству генерации библиотек доступа к JSON API.

Для генерации Java-библиотеки доступа к JSON API с использованием OpenAPI Generator можно использовать следующую команду:

```
java -jar openapi-generator-cli.jar generate -g java -i openapi.json
```

Более подробная информация о стандарте OpenAPI и возможностях инструментальных средств генерации библиотек доступа к API может быть получена в официальной документации:

- спецификация OpenAPI;
- OpenAPI Generator;
- Swagger Codegen;
- другие реализации и инструментальные средства.

7.1. Создание проекта

Чтобы создать новый проект из API:

1. Для проекта OSA, кликните по кнопке **/sca_projects** в разделе **sca**.
2. Нажмите **Try it out**.
3. Для создания проекта в параметрах достаточно указать имя проекта.
4. Нажмите **Execute**.

7.2. Запуск сканирования

Чтобы запустить сканирование **OSA** из API:

1. Перейдите в раздел **sca**.
2. Выберите нужный запрос в зависимости от вида исходных данных для анализа:
   - **/sca_projects/{id}/scans/archive** — для анализа по архиву с исходным кодом + SBOM-файлу проекта;
   - **/sca_projects/{id}/scans/url** — для анализа по ссылке на архив с исходным кодом + SBOM-файлу проекта;
   - **/sca_projects/{id}/scans/vcs** — для анализа по ссылке на репозиторий + SBOM-файлу проекта;
   - **/sca_projects/{id}/scans/url** — для анализа по SBOM-файлу проекта.
3. Нажмите **Try it out**.
4. В параметрах запроса укажите ID проекта OSA, в котором будет запущено новое сканирование.
5. В зависимости от выбранного запроса, в теле запроса выберите файлы для анализа или укажите ссылку на проект. Если вы загружаете файл, важно либо запустить анализ из директории с файлом, либо прописать путь к файлу.
6. Укажите дополнительные настройки.
7. Нажмите **Execute**.

7.3. Выгрузка отчёта

Чтобы выгрузить отчёт сканирования проекта **OSA** из API:

1. Перейдите в раздел **sca**.
2. Выберите способ получения отчёта: скачать (**/sca_reports/{id}/file**) или получить на почту (**/sca_reports/{id}/email**).
3. Нажмите **Try it out**.
4. В теле запроса введите ID нужного проекта и сканирования и укажите настройки отчёта.
5. Нажмите **Execute**.

8. ИНТЕГРАЦИИ SOLAR APPSCREENER

Все плагины для интеграции с Solar appScreener и файл clt.jar находятся на сервере Solar appScreener в директории /opt/Solar appScreener/app/plugins (Linux) и C:\Solar appScreener\plugins (Windows).

8.1. Jira

Для интеграции с **Jira** плагин не требуется. Solar appScreener использует для интеграции **Jira REST API v2**.

8.1.1. Как привязать проект в Solar appScreener к проекту в Jira

Для того чтобы привязать проект в Solar appScreener к проекту в Jira:

1. В проекте перейдите на вкладку **Таск-менеджер** в разделе **Настройки** бокового меню.
2. Нажмите на кнопку **Добавить конфигурацию**.

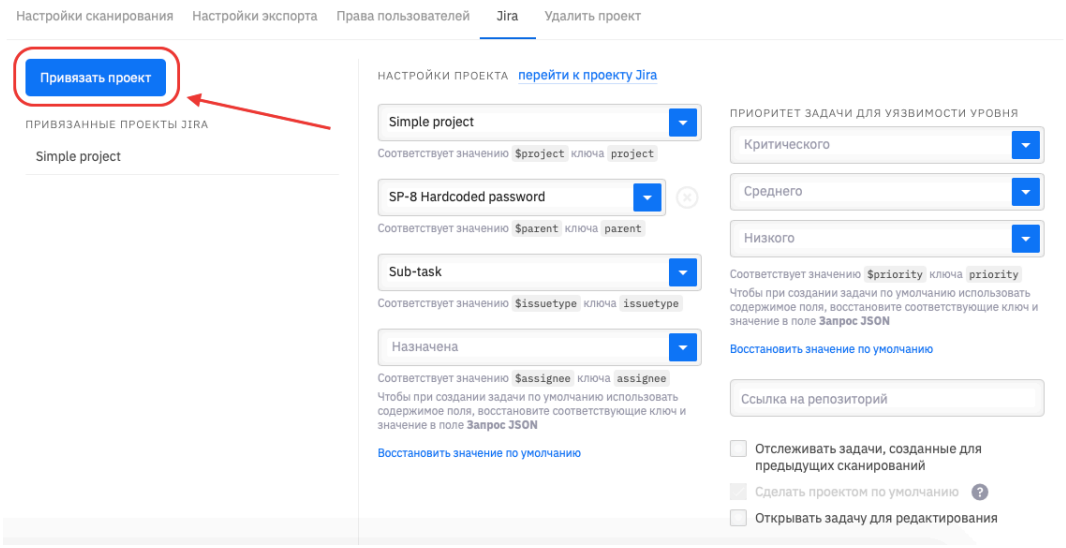

Рис. 8.1: Привязать проект Jira

3. Выберите из списка проект Jira, задайте название конфигурации и укажите тип задачи по умолчанию при создании новых задач Jira.
4. Заполните необязательные поля. Эта информация также будет использоваться по умолчанию при создании новых задач.
5. Настройте автоматическое создание задач в Jira по результатам сканирования, если необходимо.

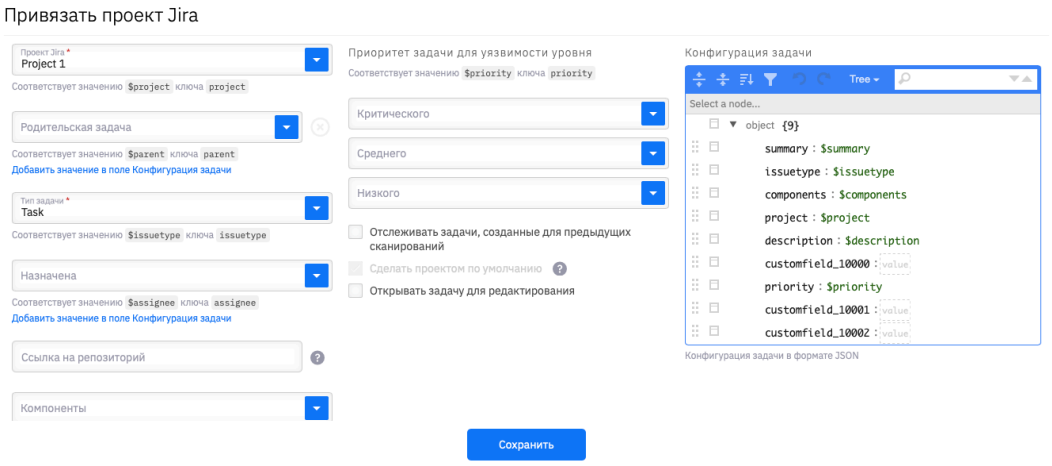

Рис. 8.2: Настроить параметры проекта Jira

Чтобы настроить уже привязанный проект, выберите его из списка и отредактируйте поля формы.

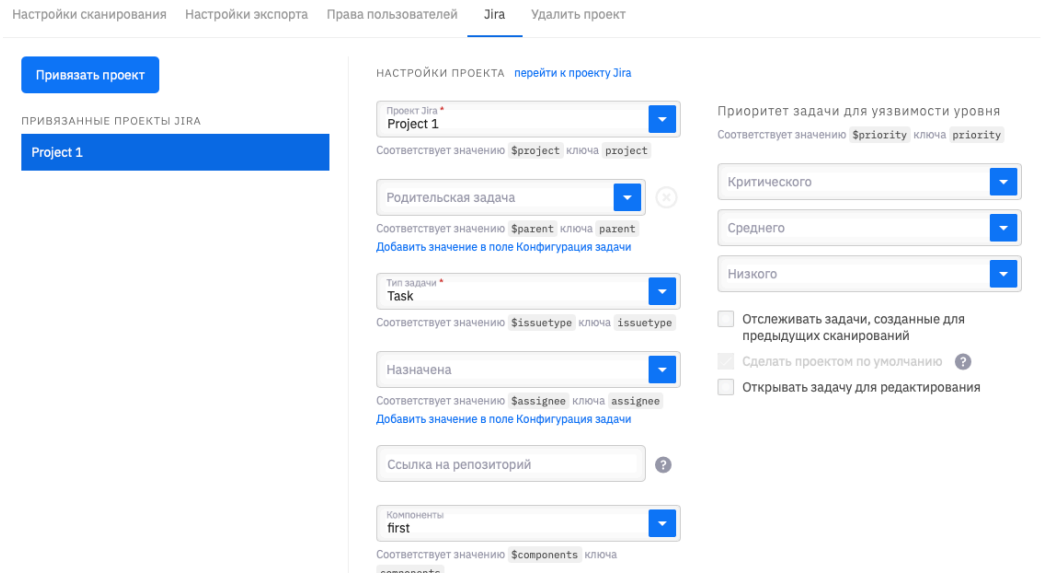

Рис. 8.3: Изменить параметры проекта Jira

8.1.2. Создание задачи в Jira

Если проект в Solar appScreener привязан к проекту в Jira, через интерфейс Solar appScreener можно создавать задачи в Jira. Для этого выполните действия:

1. Перейдите в раздел **Подробные результаты** проекта.
2. Выберите уязвимость в списке.
3. Перейдите на вкладку **Таск-менеджер**, которая располагается в правой нижней части страницы (на данной вкладке отображается список уже созданных задач в таск-менеджере).
4. Нажмите на кнопку **Создать задачу**.

5. Укажите проект, родительскую задачу (опционально), тип задачи, компоненты (опционально), тему задачи, приоритет (опционально), исполнителя (опционально), описание задачи в формате Jira (опционально).
6. Если в задаче есть другие обязательные поля, для них в поле **Конфигурация задачи** будут сгенерированы пары ключ и значение. Явно укажите значение, которое необходимо задать в задаче.
7. Нажмите на кнопку **Создать**.

Для просмотра задачи в Jira кликните на название задачи в списке. Для удаления задачи из интерфейса Solar appScreener нажмите на кнопку удаления.

Вы также можете создать задачи для нескольких уязвимостей одновременно. Для этого выберите нужные вхождения в списке в режиме выбора нескольких уязвимостей и кликните **Создать задачи** в меню действий.

8.2. ТУРБО Трекинг

Для интеграции с **ТУРБО Трекинг** плагин не требуется.

8.2.1. Как привязать проект в Solar appScreener к проекту в ТУРБО Трекинг

Чтобы привязать проект в Solar appScreener к проекту в ТУРБО Трекинг:

1. В проекте перейдите на вкладку **Таск-менеджер** в разделе **Настройки** бокового меню.
2. Нажмите на кнопку **Добавить конфигурацию**.
3. Выберите из списка проект ТУРБО Трекинг, задайте название конфигурации и укажите тип задачи по умолчанию при создании новых задач ТУРБО Трекинг.
4. Заполните необязательные поля. Эта информация также будет использоваться по умолчанию при создании новых задач.
5. Настройте автоматическое создание задач в ТУРБО Трекинг по результатам сканирования, если необходимо.

Чтобы настроить уже привязанный проект, выберите его из списка и отредактируйте поля формы.

8.2.2. Создание задачи в ТУРБО Трекинг

Если проект в Solar appScreener привязан к проекту в ТУРБО Трекинг, через интерфейс Solar appScreener можно создавать задачи в ТУРБО Трекинг. Для этого:

1. Перейдите в раздел **Подробные результаты** проекта.
2. Выберите уязвимость в списке.

3. Перейдите на вкладку **Таск-менеджер**, которая располагается в правой нижней части страницы (на данной вкладке отображается список уже созданных задач в таск-менеджере).
4. Нажмите на кнопку **Создать задачу**.
5. Укажите проект, родительскую задачу (опционально), тип задачи, компоненты (опционально), тему задачи, приоритет (опционально), исполнителя (опционально), описание задачи в формате ТУРБО Трекинг (опционально).
6. Если в задаче есть другие обязательные поля, для них в поле **Конфигурация задачи** будут сгенерированы пары ключ и значение. Явно укажите значение, которое необходимо задать в задаче.
7. Нажмите на кнопку **Создать**.

Для просмотра задачи в ТУРБО Трекинг кликните на название задачи в списке. Для удаления задачи из интерфейса Solar appScreener нажмите на кнопку удаления.

Вы также можете создать задачи для нескольких уязвимостей одновременно. Для этого выберите нужные вхождения в списке в режиме выбора нескольких уязвимостей и кликните **Создать задачи** в меню действий.

8.3. Gitlab CI

Если вы хотите встроить Solar appScreener в ваш GitLab CI в качестве Quality Gate, потребуется более детальная настройка с использованием скриптов. Примеры скриптов поставляются вместе с дистрибутивом системы.

8.3.1. Пример настройки с использованием API

В рамках тестового пайплайна (gitlab-api.zip) проект анализируется модулем OSA.

Stages:

- `build` — сборка проекта;
- `test` — запуск unit-тестов;
- `deploy` — развертывание сервиса в каком-либо окружении;
- `OSA` — OSA анализ.

Module's jobs:

- `OSA:create_project` — создание OSA проекта;
- `OSA:create_sbom` — генерация SBOM-файла;
- `OSA:start_scan_sbom` — запуск OSA сканирования SBOM-файла;
- `OSA:start_scan_archive` — запуск OSA сканирования архива.

Common jobs:

- `status_scan` — проверка статуса сканирования с ограничением в виде job's timeout;
- `check_rate_scan` — проверка результатов сканирования на соответствие определенным условиям;
- `get_summary` — генерация кратких результатов сканирования;
- `report` — генерация отчетов в форматах PDF и CSV.

Список переменных, которые редактируются пользователем:

- **API_V1_URL** — REST URL приложения, например: http://10.0.2.6/app/api/v1;
- **TOKEN** — токен аутентификации;
- **OSA_SCAN_TYPE** — выбор типа сканирования для OSA: архив (`archive`) или SBOM-файл (`sbom`);
- **MIN_RATING_OSA** — минимальная оценка рейтинга безопасности для OSA;
- **MAX_CRIT_OSA** — максимально допустимое количество критических уязвимостей для OSA.

Обратите внимание: чувствительные переменные необходимо хранить либо в специализированных сервисах, либо в скрытых переменных проекта/группы.

Замечания:

Для запуска сканирования OSA с использованием архива или SBOM-файла необходимо указать значение переменной **OSA_SCAN_TYPE** `sbom` или `archive` соотвественно.

8.3.2. Пример настройки с использованием CLT

В рамках тестового пайплайна (gitlab-clt.zip) проект анализируется модулем OSA.

Stages:

- `build` — сборка проекта;
- `test` — запуск unit-тестов;
- `deploy` — развертывание сервиса в каком-либо окружении;
- `OSA` — OSA анализ.

Module's jobs:

- `OSA:create_project` — создание OSA проекта;
- `OSA:create_sbom` — генерация SBOM-файла;
- `OSA:start_scan_sbom` — запуск OSA сканирования SBOM-файла;
- `OSA:start_scan_archive` — запуск OSA сканирования архива.

Common jobs:

- `status_scan` — проверка статуса сканирования с ограничением в виде job's timeout;
- `check_rate_scan` — проверка результатов сканирования на соответствие определенным условиям;
- `report` — генерация отчетов в форматах PDF и CSV.

Список переменных, которые редактируются пользователем:

- **API_V1_URL** — REST URL приложения, например: http://10.0.2.6/app/api/v1;
- **TOKEN** — токен аутентификации;
- **CLT_URL** — ссылка для загрузки clt.jar;
- **OSA_SCAN_TYPE** — выбор типа сканирования для OSA: архив (`archive`) или SBOM-файл (`sbom`);
- **MIN_RATING_OSA** — минимальная оценка рейтинга безопасности для OSA.

Обратите внимание: чувствительные переменные необходимо хранить либо в специализированных сервисах, либо в скрытых переменных проекта/группы.

Замечания:

Для запуска сканирования OSA с использованием архива или SBOM-файла необходимо указать значение переменной **OSA_SCAN_TYPE** `sbom` или `archive` соотвественно.

8.4. DefectDojo

Для интеграции с DefectDojo плагин не требуется. В DefectDojo можно загрузить отчёт из Solar appScreener в формате CSV или SARIF.

Для загрузки **CSV отчёта** в DefectDojo:

1. Перейдите на страницу проекта в DefectDojo.
2. На вкладке **Findings** нажмите **Import Scan Results**.
3. Заполните все необходимые поля. В поле **Scan Type** выберите **Solar appScreener Scan**.
4. В Solar appScreener скачайте отчёт в формате CSV и найдите файл **Detailed_Results.csv** в архиве с отчётом > **Scan information**.
5. В DefectDojo нажмите **Import**.

Для загрузки **SARIF отчёта** в DefectDojo:

1. Перейдите на страницу проекта в DefectDojo.
2. На вкладке **Findings** нажмите **Import Scan Results**.
3. Заполните все необходимые поля. В поле **Scan Type** выберите **SARIF**.
4. В Solar appScreener скачайте отчёт в формате SARIF.
5. В DefectDojo нажмите **Import**.

Для OSA сканирований переносятся следующие столбцы из отчёта:

- Vulnerability (название уязвимости);
- Description (описание уязвимости);
- Links (ссылки на сторонние ресурсы);
- Severity Level (уровень угрозы уязвимости).

8.5. Jenkins

Solar appScreener поддерживает **Jenkins 2.164** и более поздние версии.

8.5.1. Установка расширения в Jenkins

1. Перейдите на страницу настроек Jenkinks **Manage Jenkins** (**http://<installation_address>:8080/jenkins/manage**, **<installation address>** адрес машины, на которой установлен Jenkins).

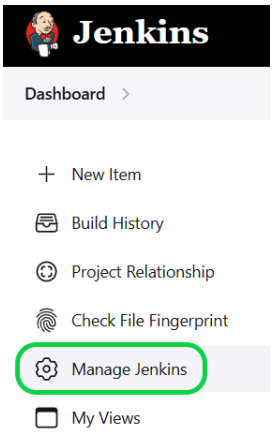

Рис. 8.4: Jenkins: Настройки Jenkins

2. Выберите **Manage Plugins** (**Управление плагинами**).

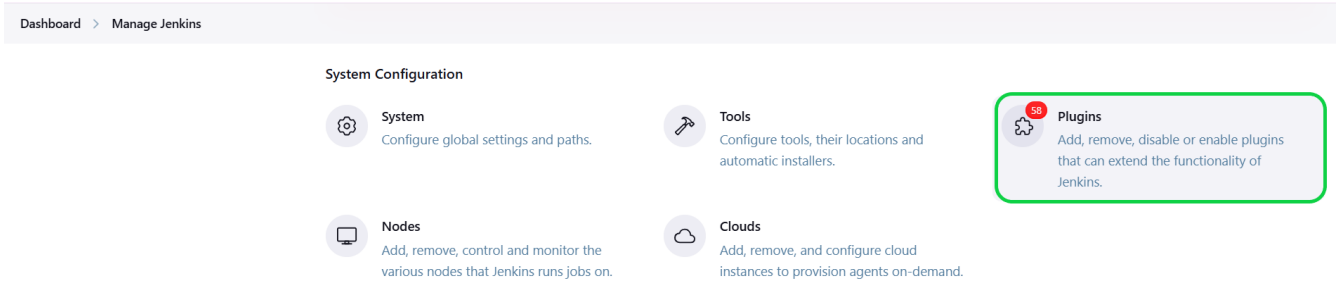

Рис. 8.5: Jenkins: Управление плагинами

3. Выберите вкладку **Advanced settings**.

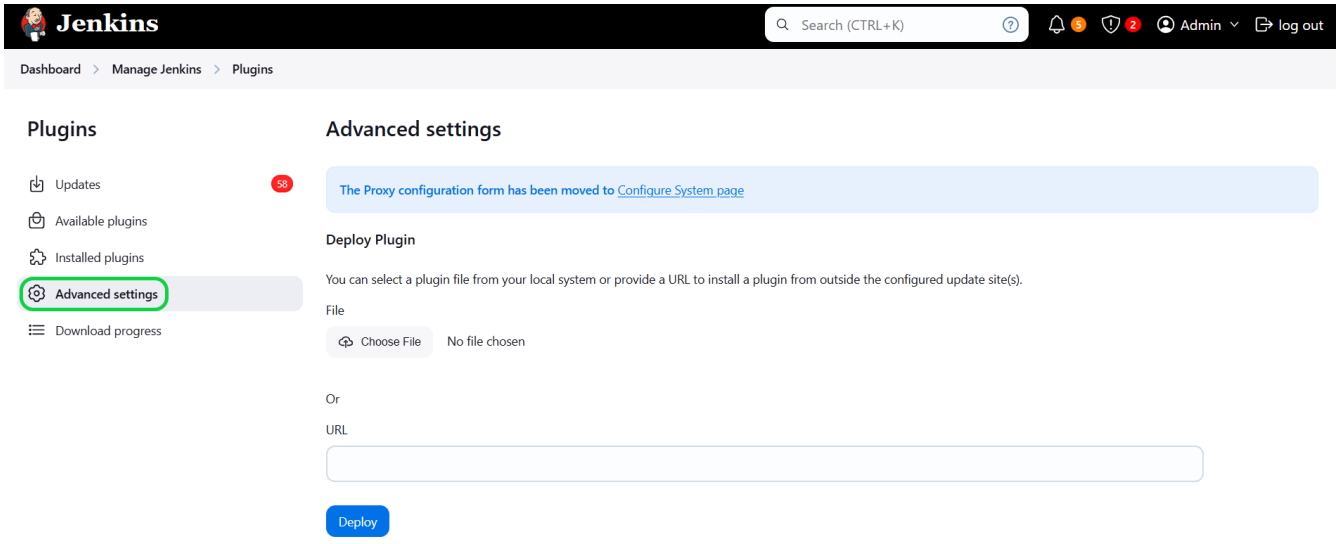

Рис. 8.6: Jenkins: Дополнительные настройки

4. В пункте **Deploy Plugin** (**Загрузить плагин**) нажмите кнопку **Choose File** (**Выберите файл**).

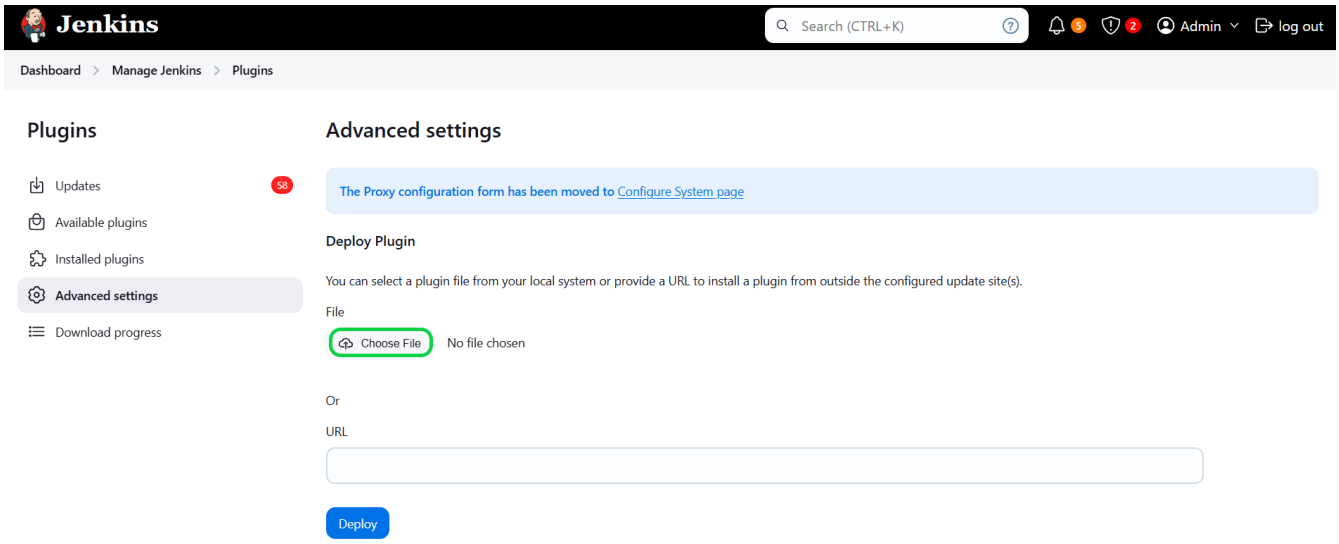

Рис. 8.7: Jenkins: Загрузить плагин

5. Выберите файл расширения в открывшемся всплывающем окне.
6. Нажмите **Upload** (**Загрузить**).

8.5.2. Общие настройки расширения

1. Перейдите на страницу настроек Jenkinks **Manage Jenkins** (**http://<installation_address>:8080/jenkins/manage**, **<installation_address>** адрес машины, на которой установлен Jenkins).
2. Выберите **System** в разделе **System Configuration** (**Конфигурация системы**).
3. Найдите пункт **Solar appScreener Analysis** (**Анализ Solar appScreener**) и заполните поля:
   - введите Solar appScreener URL, например **http://<installation_address>**, **<installation_address>** адрес машины, на которой установлен Solar appScreener;
   - введите адрес REST API для Solar appScreener, например **http://<installation_address>/app/api/v1/**, **<installation_address>** адрес машины, на которой установлен Solar appScreener;
   - введите токен, его можно получить в интерфейсе самого Solar appScreener в разделе **Личный кабинет** (**Account**);
   - введите время ожидания завершения сканирования;

Поле **Configuration ID** (**ID конфигурации**) будет заполнено автоматически после сохранения настроек. Оно указывается в качестве значения параметра **configUuid** в **Jenkins Pipeline Script** и хранит в себе глобальные настройки плагина.

4. Нажмите кнопку **Save** (**Сохранить**) внизу страницы.

8.5.3. Конфигурация анализа и отчёта

Конфигурацию анализа и отчёта можно осуществить двумя способами:

- через задачу со свободной конфигурацией;
- через **Jenkins Pipeline**.

8.5.3.1. Задача со свободной конфигурацией

1. Выберите нужный **Item** (далее нажмите **Configure** (**Настройки**) и перейдите к пункту 5) или создайте новый **New Item** (**Создать Item**).

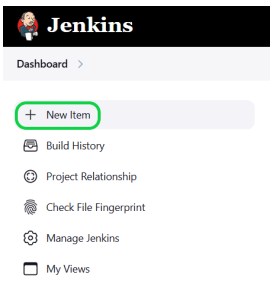

Рис. 8.8: Jenkins: Новый Item

2. Введите имя Item'а.
3. Выберите **Freestyle project** (**Создать задачу со свободной конфигурацией**).

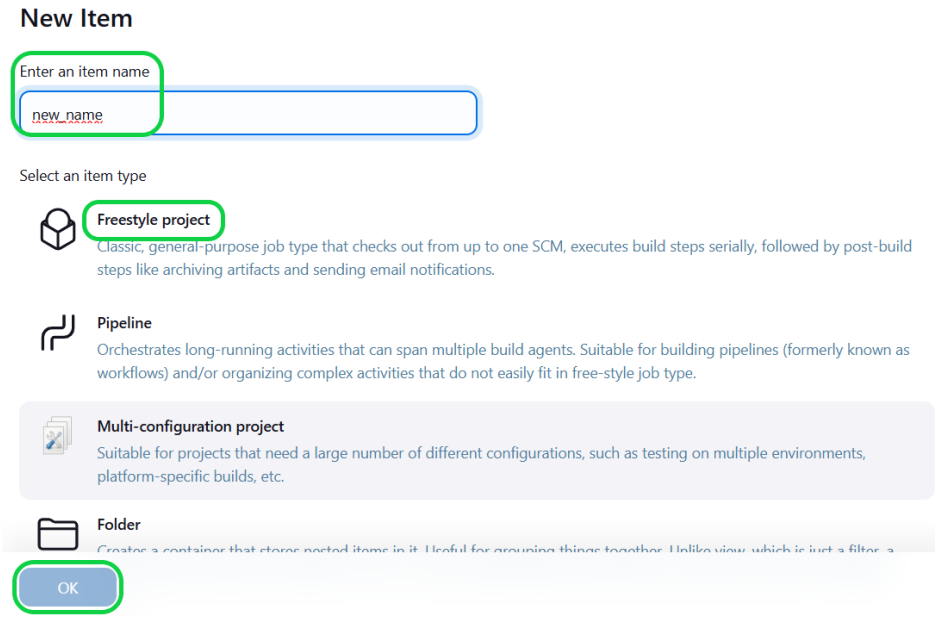

Рис. 8.9: Jenkins: Задача со свободной конфигурацией

4. Нажмите **OK**.
5. В пункте **Build Steps** (**Шаги сборки**) в поле **Add build step** (**Добавить шаг сборки**) выберите **Execute Solar appScreener Software Composition Analysis**.
6. В раскрывшемся меню:
   1. Введите UUID проекта.

      Project UUID это идентификатор существующего проекта Solar appScreener. Скопировать **Project UUID** можно в боковом меню проекта. Пример: 9feefaf0-4c17-47fe-b1f5-f7f64d4da722. Настройка конфигураций ниже перенастроит проект Solar appScreener, если текстовое поле **Project UUID** не пустое. Если текстовое поле **Project UUID** пустое, плагин Jenkins создаст новый проект в Solar appScreener.
   2. Выберите сервер.
   3. Настройте остальные конфигурации сканирования (если потребуется). Подробнее о конфигурациях сканирования см. Настройки.
7. В пункте **Post-build Actions** (**Послесборочные операции**) в поле **Add Build Step** (**Добавить шаг сборки**) выберите **Solar appScreener Software Composition Analysis report export** (**Экспорт pdf-отчёта анализа состава ПО Solar appScreener**) и сделайте настройку (см. раздел Экспорт отчёта).
8. Нажмите **Save** (**Сохранить**).

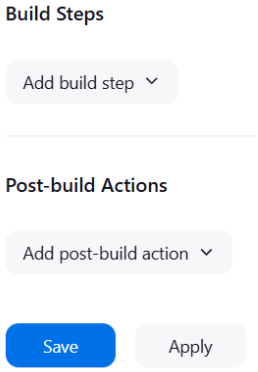

Рис. 8.10: Jenkins: Послесборочные операции

8.5.3.2. Pipeline

1. Выберите нужный **Item** (далее нажмите **Configure** (**Настройки**) и перейдите к пункту 5) или создайте новый **New Item** (**Создать Item**).
2. Введите имя Item'а.
3. Выберите **Pipeline**.
4. Нажмите **OK**.
5. В пункте **Pipeline** нажмите **Pipeline Syntax**.

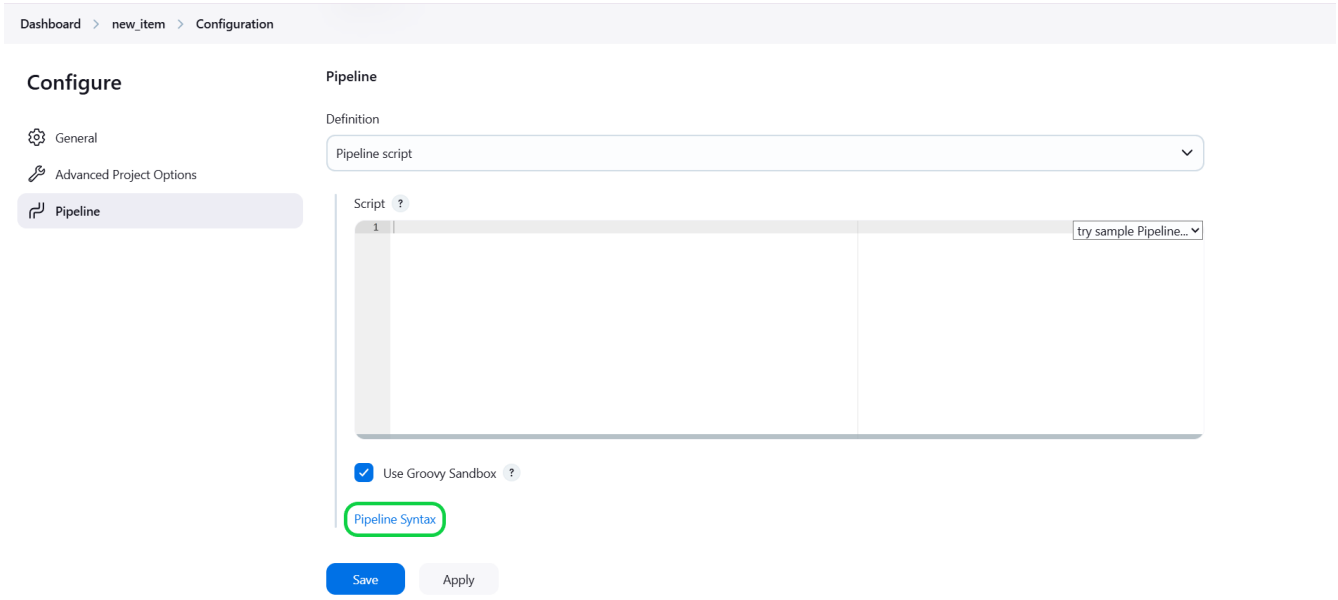

Рис. 8.11: Jenkins: Pipeline Syntax

6. В открывшейся вкладке в браузере в пункте **Steps** в поле **Sample Step** выберите **step: General Build Step**.
7. В пункте **Build Step** выберите **Execute Solar appScreener Software Composition Analysis** (**Выполнить анализ состава ПО Solar appScreener**).
8. В появившемся пункте **Solar appScreener project ID** (**Идентификатор проекта Solar appScreener**):

1. Введите UUID проекта.

   Project UUID это идентификатор существующего проекта Solar appScreener. Скопировать **Project UUID** можно в боковом меню проекта. Пример: 9feefaf0-4c17-47fe-b1f5-f7f64d4da722 . Настройка конфигураций ниже перенастроит проект Solar appScreener, если текстовое поле **Project UUID** не пустое. Если текстовое поле **Project UUID** пустое, плагин Jenkins создаст новый проект в Solar appScreener.

2. Выберите сервер.

3. Настройте остальные конфигурации сканирования (если потребуется). Подробнее о конфигурациях сканирования см. Настройки.

9. Нажмите **Generate Pipeline Script**.

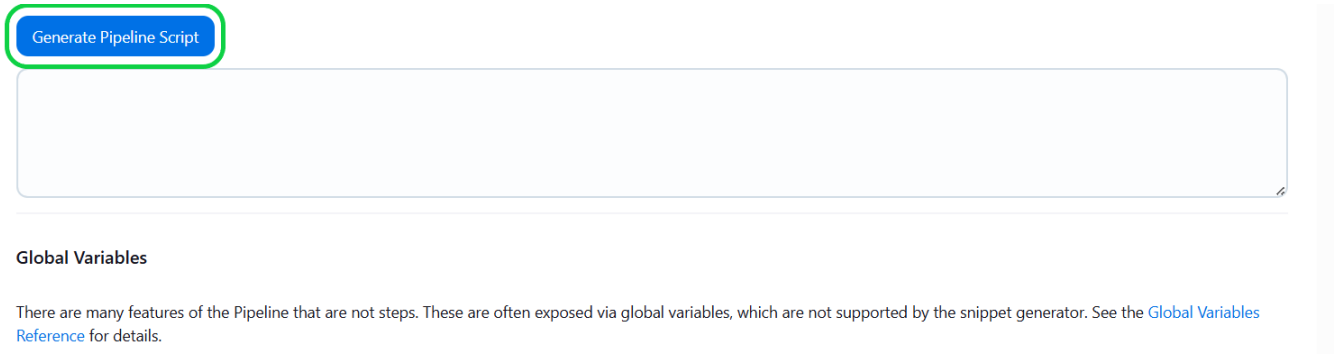

Рис. 8.12: Jenkins: Генерация скрипта Pipeline

10. Скопируйте появившийся скрипт для его использования на странице конфигурации проекта (шаг 5) в поле **Script**.

11. Вернитесь во вкладку **Pipeline Syntax**.

12. В пункте **Build Step** выберите **Solar appScreener Software Composition Analysis report export** (**Экспорт отчёта анализа состава ПО Solar appScreener**).

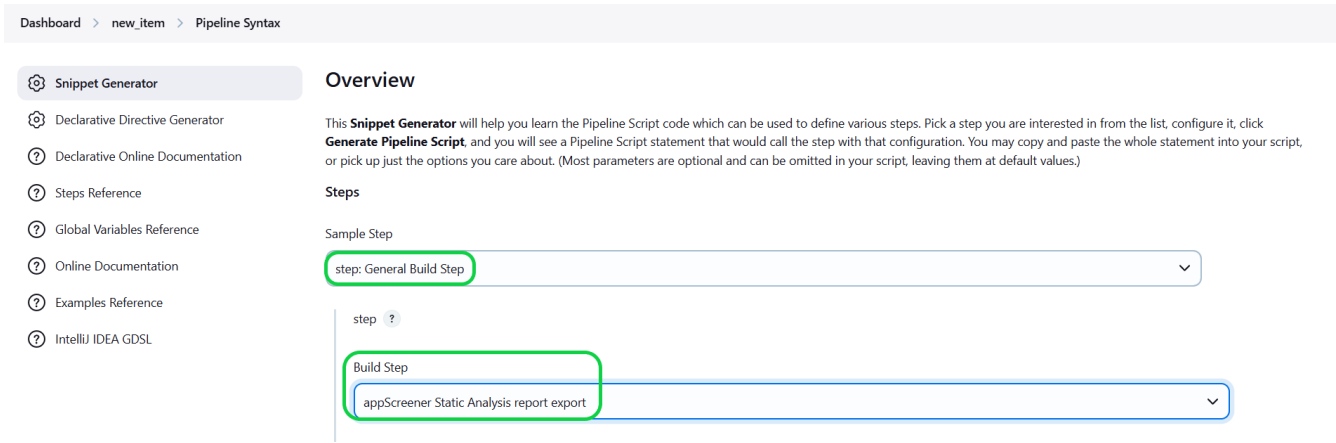

Рис. 8.13: Jenkins: Экспорт отчёта

13. В появившемся пункте настройте конфигурации формирования отчёта (если потребуется).

14. Нажмите **Generate Pipeline Script**.
15. Скопируйте появившийся скрипт для его использования на странице конфигурации проекта (шаг 5) в поле **Script**.
16. Нажмите **Save** (**Сохранить**).

8.5.4. Конфигурация Build Failure Conditions

Для того, чтобы создавать **build failure conditions** на основе переменных в виде метрик в **post-build step**:

1. Выберите нужный **Item** из списка.

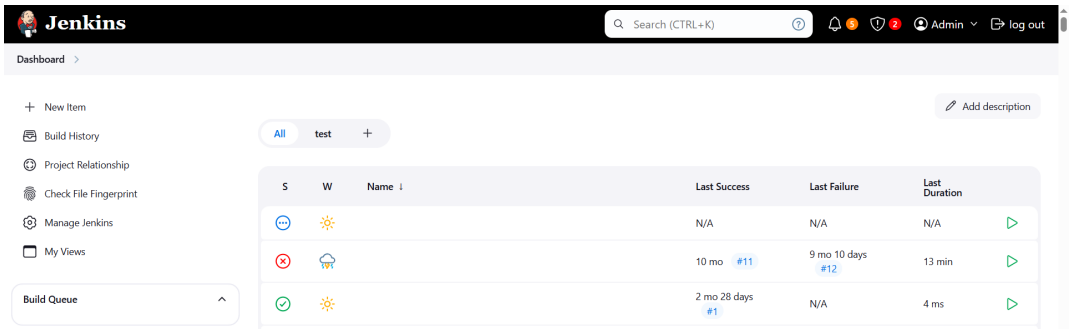

Рис. 8.14: Jenkins: Выбор Item'а

2. Нажмите **Configure** (**Настройки**).

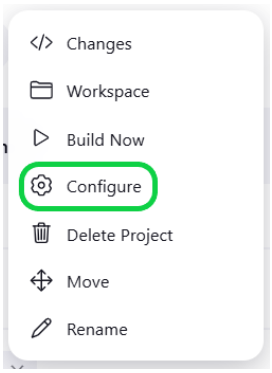

Рис. 8.15: Jenkins: Выбор настроек

3. В пункте **Post-build Actions** (**Послесборочные операции**) нажмите **Add post-build action** (**Добавить шаг после сборки**).

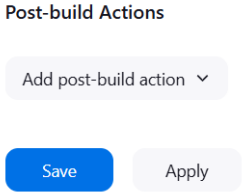

Рис. 8.16: Jenkins: Выбор послесборочных операций

4. Выберите **Post build task**.
5. Выберите появившийся раздел **Post build task**.
6. В разделе **Post build task** укажите **build failure conditions** в формате bash скрипта. Например, скрипт:

```
if [[ $SCORE < 3.5 || $CRITICAL > 10 || $LOW > 30 ]]; then
  echo "vulnerable app"
  exit 1
else
  exit 0
fi
```

означает, что сборка не будет завершена для проектов с рейтингом ниже 3,5, или количеством критических уязвимостей больше 10, или количеством уязвимостей низкого уровня больше 30.

Используйте глобальные переменные:

- Плагин создает переменные среды для сборки:
  - **PROJECT_ID** UUID проекта в Solar appScreener;
  - **SCAN_ID** UUID сканирования в Solar appScreener;
  - **SCAN_URL** ссылка на результаты сканирования в Solar appScreener;
  - **SERVER** адрес REST API;
  - **SERVER_UUID** идентификатор сервера, присваиваемый при настройке глобальной конфигурации;
  - **PDF_URL** адрес PDF-отчёта;
  - **SCORE** рейтинг проекта в Solar appScreener;
  - **TOTAL** общее количество уязвимостей в проекте;
  - **CRITICAL** количество уязвимостей критического уровня;
  - **MEDIUM** количество уязвимостей среднего уровня;
  - **LOW** количество уязвимостей низкого уровня;
  - **INFO** количество уязвимостей информационного уровня.
- Следующие параметры рассчитываются при установке чекбокса **Включить в отчёт сравнение со сканированием**. Они рассчитываются относительно сканирования, UUID которого указан в поле **UUID сканирования для сравнения**.
  - **NEW_TOTAL** общее количество новых уязвимостей в проекте;
  - **NEW_CRITICAL** количество новых уязвимостей критического уровня;
  - **NEW_MEDIUM** количество новых уязвимостей среднего уровня;
  - **NEW_LOW** количество новых уязвимостей низкого уровня;
  - **NEW_INFO** количество новых уязвимостей информационного уровня.

Эти переменные можно потом использовать для других шагов сборки.

7. Активируйте чекбокс **Escalate script execution status to job status**.

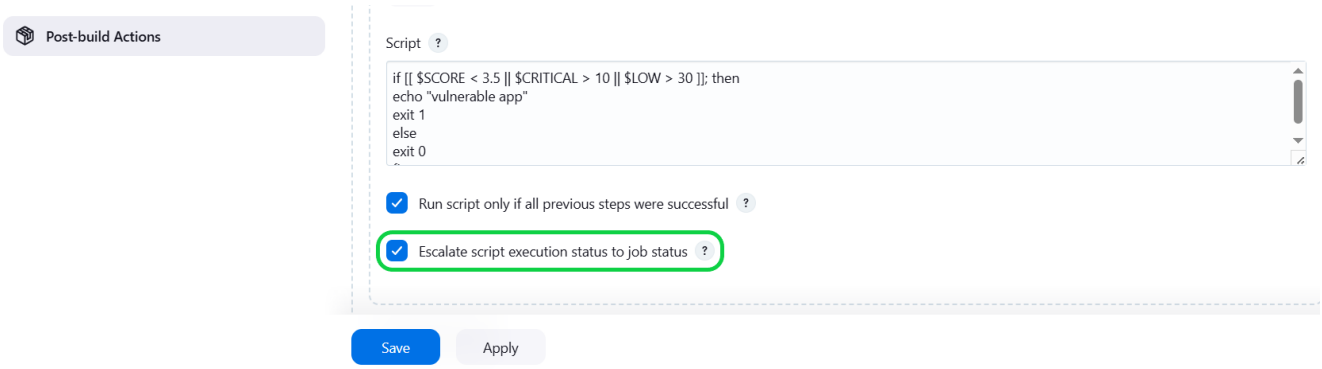

Рис. 8.17: Jenkins: Сохранение послесборочных операций

8. Нажмите **Save** (**Сохранить**).

8.5.5. Результаты и отчёт

Для того чтобы просмотреть результаты сканирования:

1. Выберите нужный **Item** из списка.

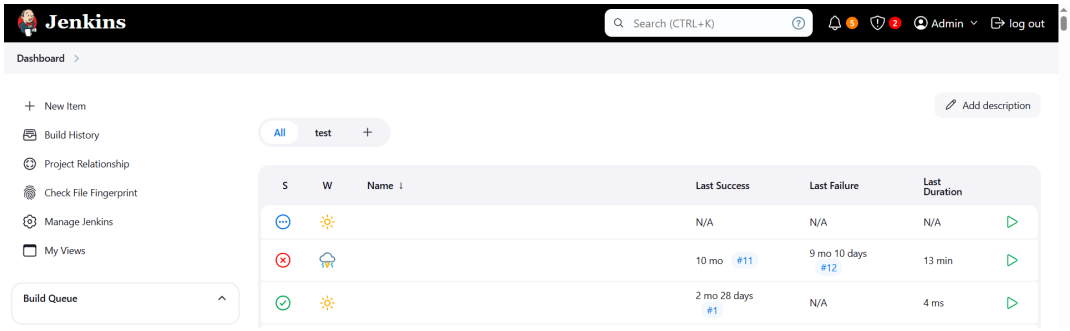

Рис. 8.18: Jenkins: Выбор Item'а

2. Для просмотра отчёта нажмите **Solar appScreener report** (**отчёт Solar appScreener**) или скачайте по ссылке в результатах сканирования (шаг 5).

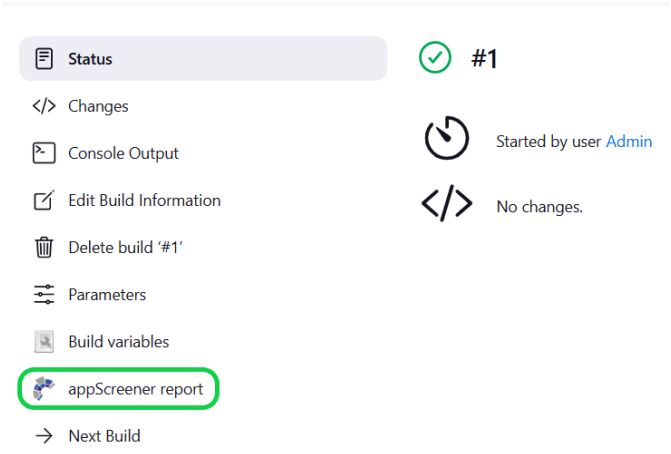

Рис. 8.19: Jenkins: отчёт

3. Нажмите на нужный номер сборки в блоке **Builds** (**История сборок**) располагающемся под боковым меню.

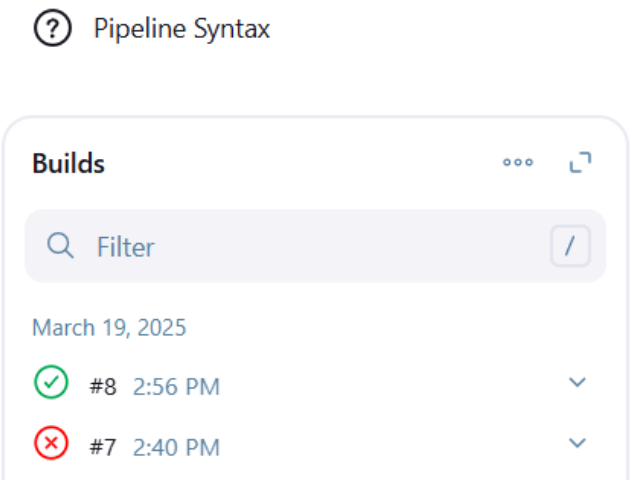

Рис. 8.20: Jenkins: Номер сборки

4. Для просмотра журнала событий (logs) нажмите **Console Output** (**Вывод консоли**) в боковом меню.

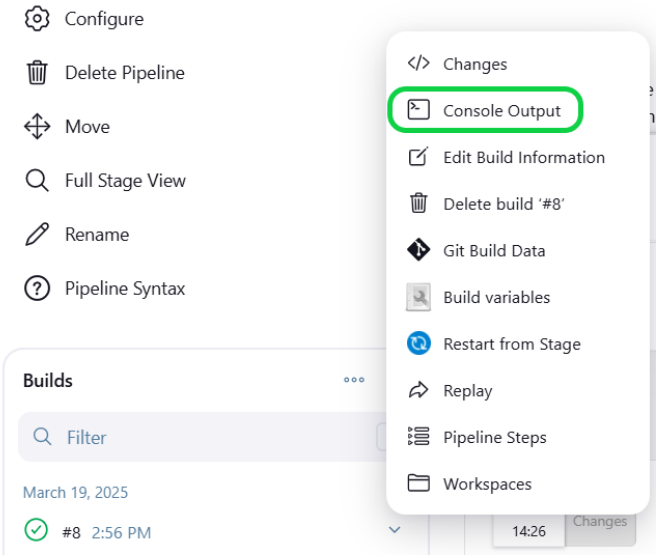

Рис. 8.21: Jenkins: Консоль

5. Для просмотра результатов сканирования нажмите **Build variables** (**Переменные сборки**). В пункте PDF_URL находится ссылка на скачивание PDF-отчёта.

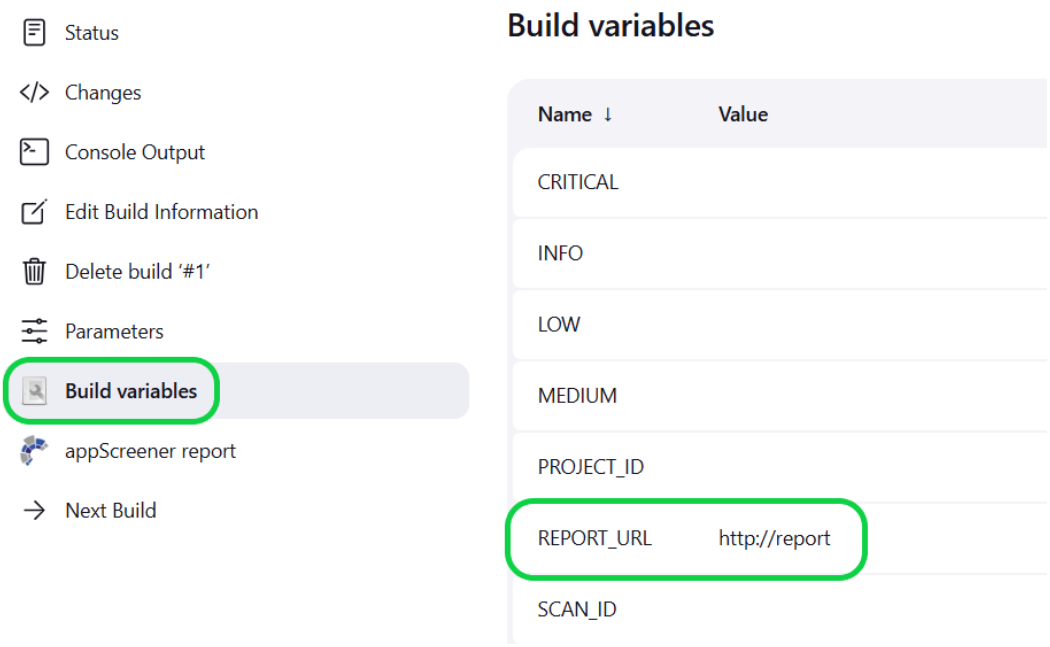

Рис. 8.22: Jenkins: Результаты сканирования

8.6. Azure DevOps Server

Solar appScreener поддерживает **Team Foundation Server 2018**, **Azure Devops Server 2019**, **Azure Devops Server 2020** и **Azure DevOps Services**.

8.6.1. Установка расширения

1. Перейдите на страницу управления расширениями (**http://<installation_address>/_gallery**, **<installation_address>** адрес машины, на которой установлен Azure DevOps Server).
2. Нажмите на кнопку **Manage Extensions** (**Управление расширениями**) в правой нижней части страницы.

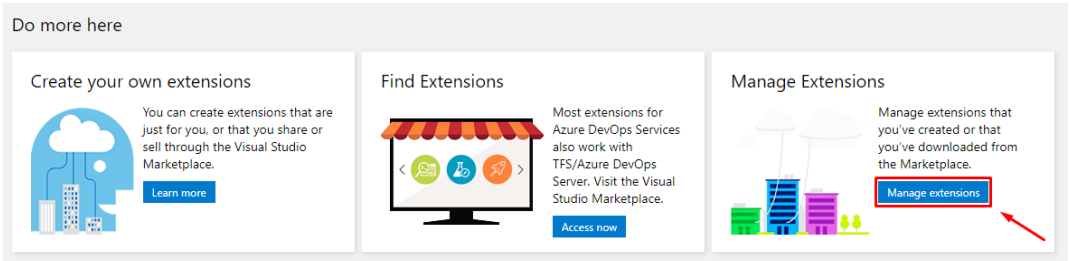

Рис. 8.23: Azure DevOps Server: Управление расширениями

3. Нажмите на кнопку **Upload extension** (**Загрузить расширение**).

Рис. 8.24: Azure DevOps Server: Отправить новое расширение

4. Выберите файл расширения в открывшемся всплывающем окне.
5. Нажмите на кнопку **Upload** (**Загрузить**).

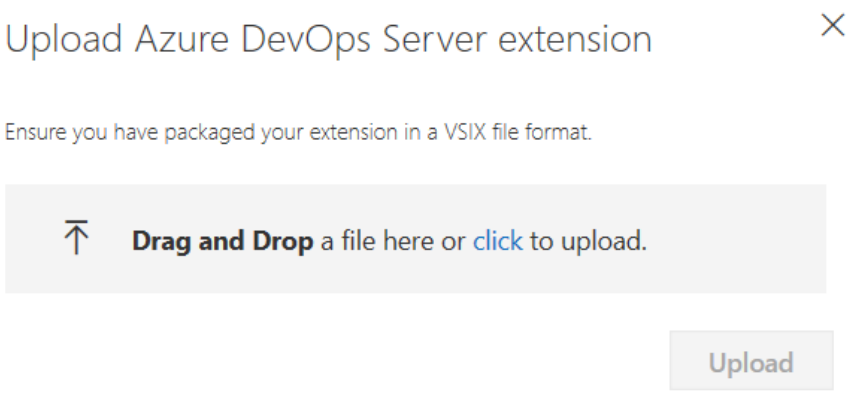

Рис. 8.25: Azure DevOps Server: Отправка

6. Выберите из списка расширение Solar appScreener.

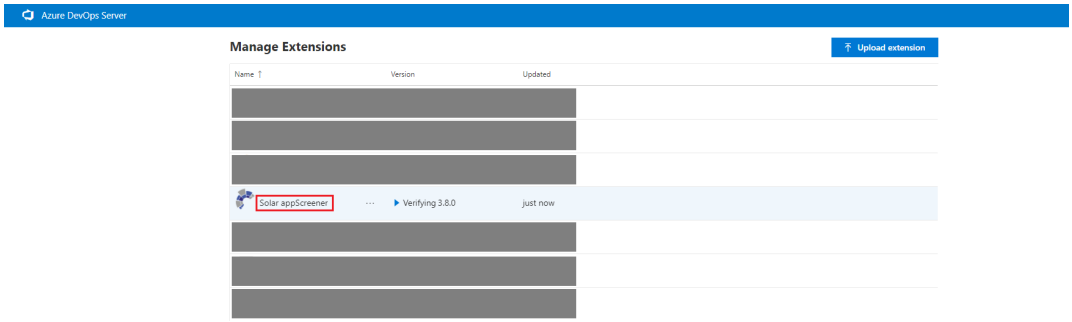

Рис. 8.26: Azure DevOps Server: Выбор расширения

7. Нажмите на кнопку **Get it free** (**Получить бесплатно**) и выберите коллекцию, для которой хотите установить расширение, например **DefaultCollection**.
8. Нажмите на кнопку **Install** (**Установка**).

   Если все шаги выполнены, на странице отобразится текст **You are all set!** (**Все готово**). В любом проекте выбранной коллекции будет доступна задача как шаг определения сборки.

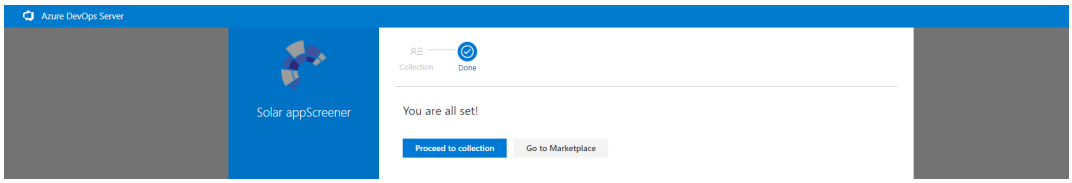

Рис. 8.27: Azure DevOps Server: Подтверждение

8.6.2. Добавление шага для сборки в Azure DevOps Server

1. Перейдите на страницу нужного проекта коллекции, для которого было установлено расширение (**http://<installation_address>/DefaultCollection/projectexample**,

<installation_address> адрес машины, на которой установлен Azure DevOps Server).

2. Перейдите по пути **Pipelines** -> **Pipelines** (**Сборка и выпуск** -> **Сборки**).

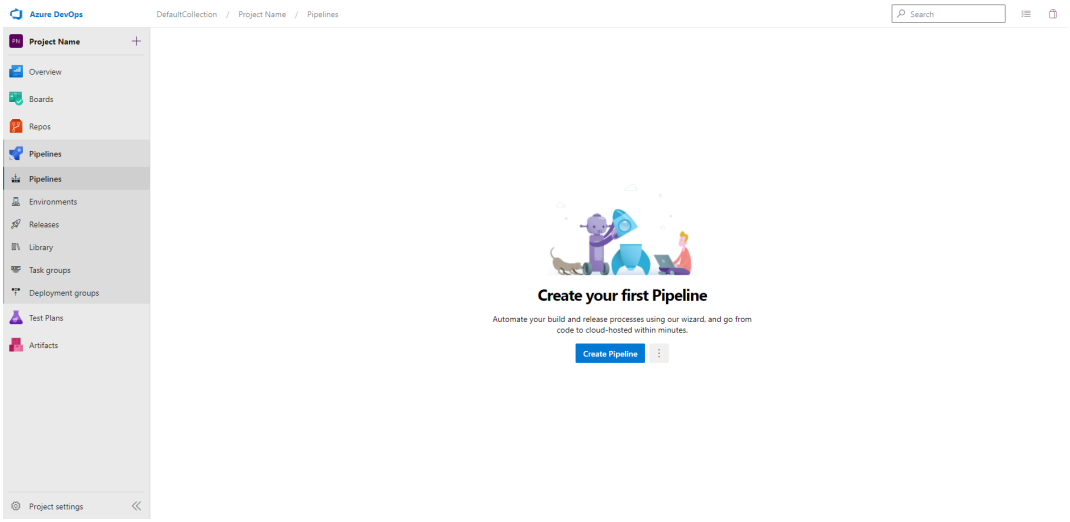

Рис. 8.28: Azure DevOps Server: Сборка и выпуск

3. Нажмите на три точки и нажмите **Edit** (**Изменить**) для имеющегося определения сборки или создайте новое, нажав **New Pipeline** (**Создать сборку**). Если вы выбрали новую сборку, выберите **Classic Redactor** (**классический редактор**). Далее нажмите на **Продолжить** > в выборе шаблона укажите **Empty Job** (**пустое задание**).
4. Нажмите **Add Task** (**Добавить задачу**).

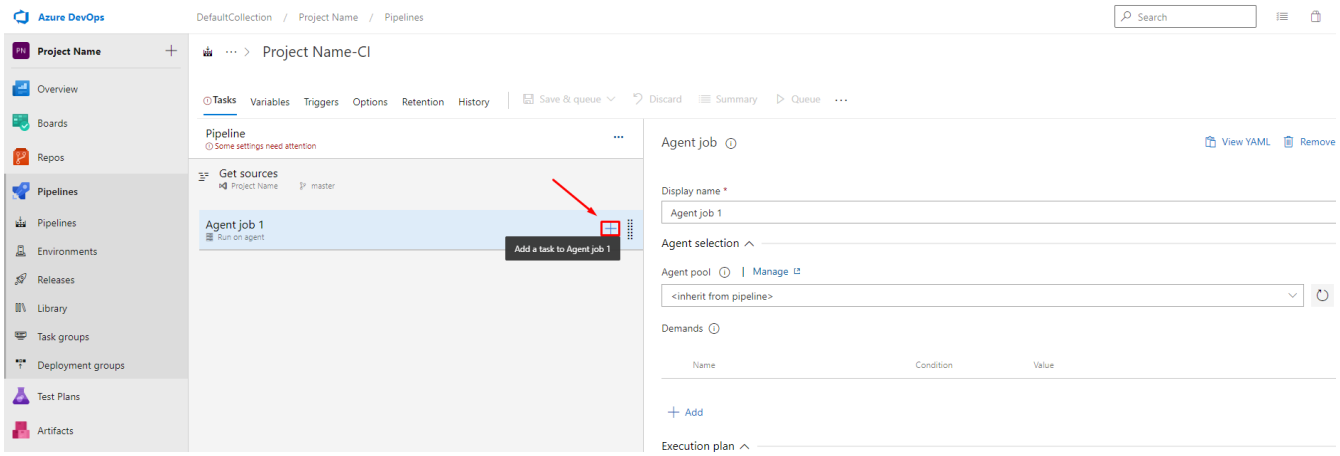

Рис. 8.29: Azure DevOps Server: Добавление задачи

5. Найдите **Run Solar appScreener SCA** и нажмите **Add** (**Добавить**).
6. Выберите добавленный шаг сборки.
7. Добавьте подключение к серверу Solar appScreener из списка или создайте новое:
   1. Справа от **Solar appScreener server end point** нажмите на кнопку **New** (**Создать**).
   2. В появившемся окне введите адрес API (например, **http://<installation_address>/app/api/v1/ <installation_address>** адрес машины, на которой установлен Solar appScreener) и токен. Токен можно получить в разделе

Личный кабинет (при получении токена рекомендуется установить длительное время действия токена).

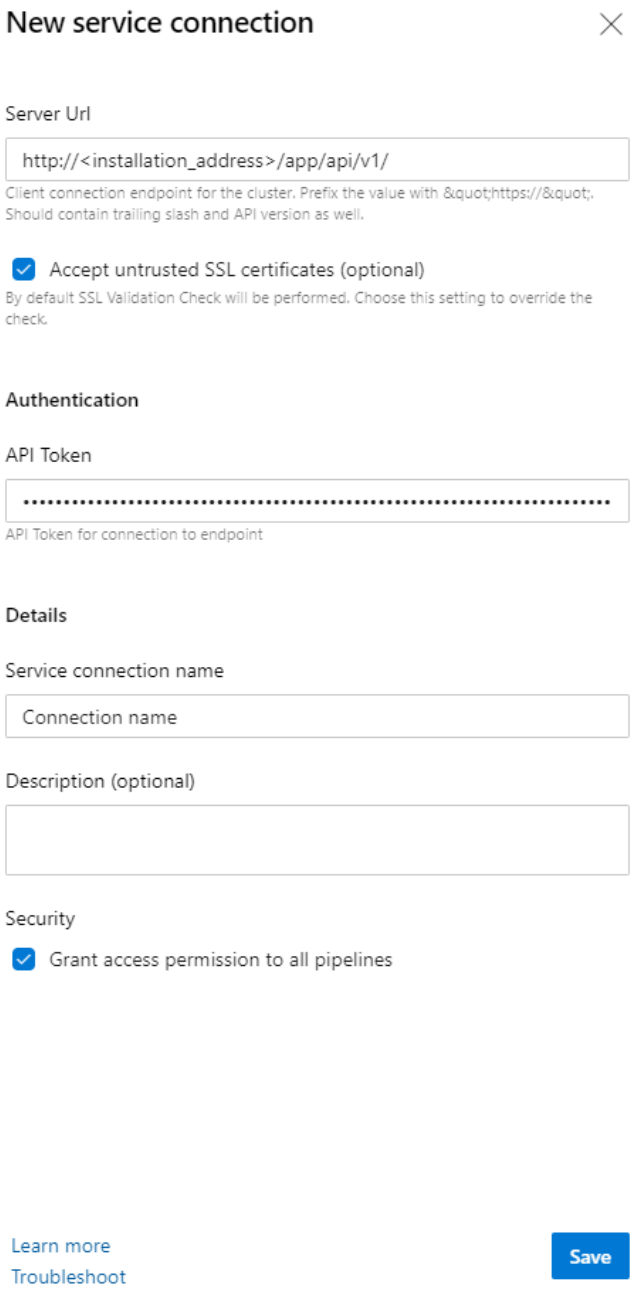

Рис. 8.30: Azure DevOps Server: Добавление подключения

3. Нажмите **Save**.

8. Настройте необходимые параметры анализа. Подробнее о настройках анализа в разделе Общие.
9. Укажите дополнительные параметры в меню **General analysis settings**.
10. Настройте **Failure Conditions**:
11. В разделе **Task failure conditions** активируйте опцию **Enable failing on condition**.
12. Установите **Failure Conditions** в зависимости от значений (Score condition, Critical

issues number condition, Medium issues number condition, Low issues number condition, Info issues number condition).

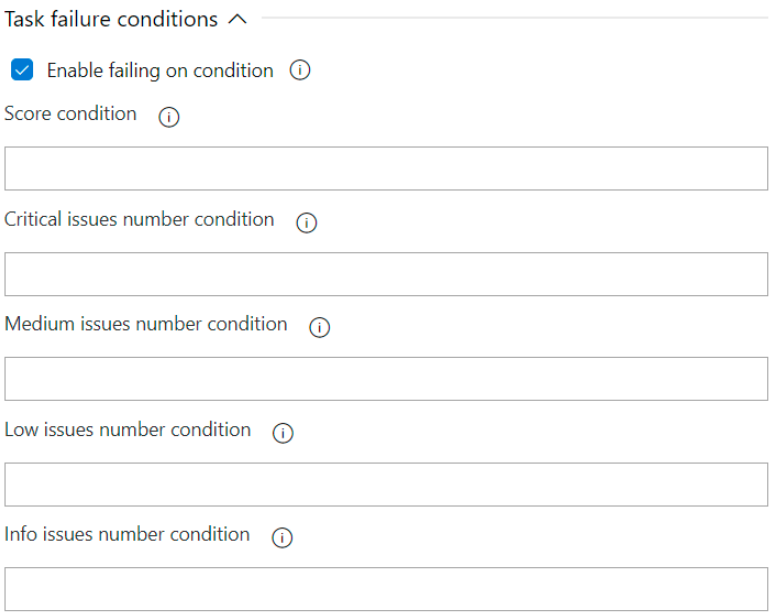

Рис. 8.31: Azure DevOps Server: Failure conditions

13. Нажмите в верхнем правом углу **Save and queue** (**Сохранить и поместить в очередь**) и затем ещё раз **Save and queue** (**Сохранить и поместить в очередь**).

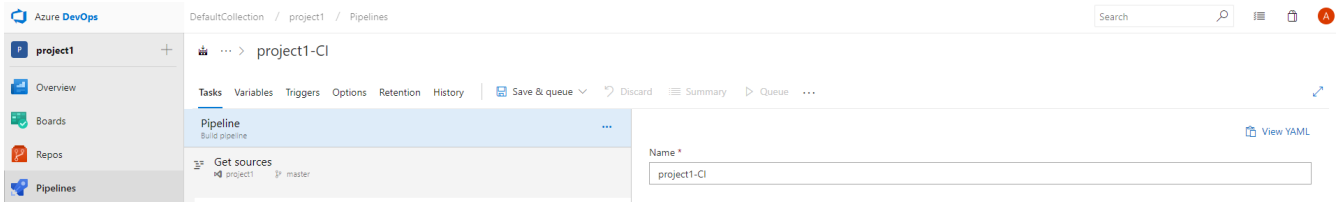

Рис. 8.32: Azure DevOps Server: Сохранение

12. Дождитесь окончания сборки и перейдите на страницу результатов.
13. Чтобы скачать отчёт, перейдите во вкладку **Solar appScreener code analysis results** и откройте ссылку на отчёт в новой вкладке браузера.

8.7. TeamCity

Solar appScreener поддерживает **TeamCity 2017.2.2.** и более поздние версии.

8.7.1. Интеграция Solar appScreener через плагин

Инструкция по использованию плагина к TeamCity:

1. Перейдите на страницу, на которой установлен TeamCity, например: **http://<installation_address>:8111/teamcity/**

2. Установите плагин: перейдите по пути **Administration->Plugins List->Upload plugin zip**. В некоторых случаях для корректной работы плагина может потребоваться удаление его предыдущих версий.
3. Настройте соединение:
    1. Перейдите по пути **Administration->Integrations->Solar appScreener**.
    2. В разделе **Solar appScreener Software Composition Analysis** введите адрес API (должен заканчиваться слешем, например: **http://<installation_address>/app/api/v1/**, **<installation_address>** адрес машины, на которой установлен Solar appScreener) и токен, который можно получить в веб-интерфейсе Solar appScreener на странице аккаунта в разделе **Токен** (см. раздел Личный кабинет).
    3. Нажмите на кнопку **Test connection** (при успешной проверке подключения появится надпись **Successful**).
    4. Нажмите **Save**.
4. Добавьте **Solar appScreener SCA Build Step** в настройках сборки и укажите для него необходимые настройки.
5. Добавьте **Build Features** (доступны только при наличии **Solar appScreener SCA Build Step**):
    - **Solar appScreener PDF report** настройте экспорт отчёта с результатами сканирования (см. раздел Экспорт отчёта);
    - **Solar appScreener statistics** включает в себя оценку безопасности, количество уязвимостей каждого уровня критичности, продолжительность сканирования (посмотреть статистику можно в **Build->Parameters->Reported statistic values**);
6. Настройте **Failure Conditions**:
    1. В настройках сборки нажмите **Failure Conditions**.
    2. Нажмите **Add failure condition** и выберите **Fail build on metric change**.
    3. Установите **Failure Conditions** в зависимости от значений **Solar appScreener statistics** (Solar appScreener LOC; info, low, medium, critical vulnerabilities; scan duration; score).

9. МАНИФЕСТЫ, ПОДДЕРЖИВАЕМЫЕ ДЛЯ ГЕНЕРАЦИИ SBOM

Таблица 9.1: Манифесты, поддерживаемые для генерации SBOM

Язык	Манифесты
Java/Kotlin/Scala	`pom.xml`, `ivy.xml`, `*.gradle`, `gradle.lockfile`
JavaScript/TypeScript	`package.json`, `package-lock.json`, `yarn.lock`, `pnpm-lock.yaml`
Python	`setup.py`, `Pipfile`, `Pipfile.lock`, `pyproject.toml`, `poetry.lock`, `requirements.txt`, `requirements.pip`, `requires.txt`
C/C++	`conanfile.txt`, `conan.lock`
Go	`go.mod`, `go.sum`
PHP	`composer.json`, `composer.lock`
Ruby	`Gemfile`, `Gemfile.lock`, `*.gemspec`
C#/VB.NET	`*.nuspec`, `packages.lock.json`, `Project.json`, `Project.lock.json`, `packages.config`, `paket.dependencies`, `paket.lock`, `*.csproj`, `*.fsproj`, `*.vbproj`, `project.assets.json`, `sln`
Objective-C/Swift	`Podfile`, `Podfile.lock`, `*.podspec`
Rust	`Cargo.lock`, `Cargo.toml`
Dart	`pubspec.yaml`, `pub.lock`
Erlang	`rebar.config`, `rebar`, `rebar.lock`