Solar appScreener

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

Solar appScreener

Версия 3.15.2

Июнь 2025

СОДЕРЖАНИЕ

1. ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

Аббревиатура	Расшифровка
АРМ	Автоматизированное рабочее место
БД	База данных
ОС	Операционная система
ПО	Программное обеспечение
СУБД	Система управления базами данных
UI	User Interface – интерфейс пользователя
CLI	Command Line Interface – интерфейс командной строки
SBOM	Software Bill of Materials – документ, содержащий подробный перечень компонент и зависимостей
SSDLC	Secure Software Development Lifecycle – жизненный цикл безопасной разработки системы

2. ВВЕДЕНИЕ

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

3. СВЕДЕНИЯ О SOLAR APPSCREENER

3.1. Описание возможностей

3.1.1. 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.

3.2. Перечень эксплуатационной документации для ознакомления

В поставку Solar appScreener входят следующие эксплуатационные документы:

- Руководство системного администратора (для Windows и Linux);
- Руководство пользователя.

4. ТРЕБОВАНИЯ К СЕРВЕРНОЙ И КЛИЕНТСКОЙ ЧАСТЯМ

4.1. Требования к аппаратному обеспечению

4.1.1. Серверная часть

Система поставляется модулями. Для корректной работы требуется последовательно установить необходимый набор модулей. Модули можно установить на один сервер или на несколько серверов, связанных в одну сеть. Ссылки на скачивание модулей содержатся в инструкции по установке Solar appScreener.

- APP модуль - обязательный модуль, веб приложение, отвечает за логику работы всей системы;
- OSA модуль - анализатор сторонних компонент.

4.1.1.1. Минимальные характеристики оборудования для установки на одном сервере

Для функционирования Solar appScreener с модулем анализа сторонних (APP + OSA модуль) на одном сервере, требуется оборудование со следующими минимальными характеристиками:

- 8 ядерный процессор с тактовой частотой 2.2 ГГц;
- объём оперативной памяти – 32 ГБ;
- минимальный объём жёсткого диска – 512 ГБ SSD/ SAS HDD (при увеличении количества сканирований требуемый объём может увеличиться);
- поддерживаемые операционные системы (см. Требования к программному обеспечению).

В зависимости от количества сканирований и наличию или отсутствию SBOM файлов для анализа состава программного обеспечения (OSA) минимальные характеристики могут возрасти.

4.1.1.2. Минимальные характеристики для установки модулей на отдельных серверах

Для развертывания модулей на отдельных серверах для разных подсистем требуются следующие минимальные характеристики:

APP модуль

- 4 ядерный процессор с тактовой частотой 2.2 ГГц;
- объём оперативной памяти – 8 ГБ;
- минимальный объём жёсткого диска – 512 ГБ SSD/ SAS HDD (при увеличении количества сканирований требуемый объём увеличивается);
- поддерживаемые операционные системы (см. Требования к программному обеспечению).

OSA модуль

- 4 ядерный процессор с тактовой частотой 2.2 ГГц;

- объём оперативной памяти – 16 ГБ;
- минимальный объём жёсткого диска – 128 ГБ SSD/ SAS HDD;
- поддерживаемые операционные системы.

4.1.2. Клиентская часть

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

4.2. Требования к программному обеспечению

4.2.1. Серверная часть

Для функционирования Solar appScreener на серверном оборудовании должна быть установлена одна из операционных систем:

- Ubuntu 20.04 LTS;
- Ubuntu 22.04 LTS;
- Ubuntu 24.04 LTS;
- Debian 12;
- CentOS 7;
- AlmaLinux 8;
- Red Hat Enterprise Linux 8;
- Red Hat Enterprise Linux 9;
- Astra Linux Special Edition 1.7;
- Astra Linux Special Edition 1.8;
- RedOS 7;
- RedOS 8;
- ALT Linux server/server virtualization 10.

Также на серверной части должны выполняться следующие требования к аппаратно-программному комплексу:

- чистая операционная система без предустановленного стороннего ПО;
- отсутствие установленных систем контейнеризации;
- отсутствие ограничений для работы систем контейнеризации;
- версия ядра Linux 3.10 или выше, поддерживаются только generic ядра; рекомендуется использовать последнюю версию ядра;
- версия iptables 1.4 или выше;
- версия git 1.7 или выше;
- версия XZ Utils 4.9 или выше;
- исполняемый файл ps, предоставляемый procps или аналогичным пакетом;
- наличие и доступность пакетов из базовых репозиториев для операционной системы;
- привилегии пользователя root/administrator;
- свободные TCP-порты:
  - 80, 443 — для подключения NGINX и TLS протокола;
  - 61616 — для подключения Apache ActiveMQ Artemis.

4.2.2. Клиентская часть

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

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

Рекомендуем в настройках браузера разрешить выполнение **JavaScript** и сохранение файлов **cookies**.

5. ФУНКЦИОНАЛЬНАЯ СТРУКТУРА SOLAR APPSCREENER

Архитектура Solar appScreener обеспечивает быстродействие и отказоустойчивость системы. Взаимодействие с пользователем, распределение задач и анализ кода выполняется отдельными модулями.

Solar appScreener включает следующие модули:

- **Веб-приложение**:
  - **UI**;
  - **API**;
  - **Backend**;
  - **DB**.
- **Message broker**.
- **Daemon**.
- **Модули анализа**:
  - модуль анализа состава ПО.
- **Модули для интеграции с системами**:
  - CI/CD: Azure DevOps (TFS), Jenkins, TeamCity;
  - CLI: CLT;
  - отслеживания ошибок: Jira.

Веб-приложение. Модуль представляет собой веб-приложение, развёрнутое на сервере Apache Tomcat. Через **Веб-приложение** осуществляется взаимодействие с остальными компонентами Solar appScreener. **Веб-приложение** включает компоненты **UI** и **Backend**.

- **UI**. Компонента веб-приложения – пользовательский интерфейс. **UI** взаимодействует с **Backend**.
- **Backend.** Компонента веб-приложения. Выполняет все сложные/долгие операции:
  - обновление БД при запуске анализа;
  - взаимодействие с модулем **Message broker**:
    - добавление задачи в очередь;
    - опрос для установления статуса;
    - выгрузка результатов.
  - сохранение результатов в БД;
  - формирование списка уязвимостей, подготовка отчётов.

БД. База данных (PostgreSQL) для хранения информации.

Message broker. Реализует очередь с приоритетами и является промежуточным модулем между **Backend** и **Daemon**.

Daemon. Обращается к модулю **Message broker** для получения данных о задачах, запускает соответствующие модули анализа. Выполняет мониторинг работы модулей, отправляет в **Message broker** информацию об обновлениях статуса, а также результаты сканирования.

Модули анализа. Запускаются модулем **Daemon** через CLI. В выбранном формате фиксируют статус и результаты сканирования.

- Модуль OSA – анализ open-source, используемого в коде проекта.

CLI. Command Line Interface. Взаимодействует с модулем **Backend** по сети, предоставляет доступ к функциональности Solar appScreener через CLI.

6. ОПИСАНИЕ РАБОТЫ С SOLAR APPSCREENER

6.1. Установка Solar appScreener

6.1.1. Порядок установки

От того, какой вариант установки будет предпочтительным: на одном сервере или нескольких, зависят требования к аппаратным ресурсам сервера/серверов (см. Требования к аппаратному обеспечению). Количество серверов и работающих на них модулей можно скорректировать после установки, добавив или удалив необходимые элементы (сервера, модули) в систему.

Каждый модуль поставляется с полным окружением для работы. В зависимости от типа модуля это либо полностью контейнеризированная среда **Docker**, либо нативная среда исполнения для iccheck-win/macos в виде **JDK-11**. Все это устанавливается/обновляется автоматически. Более подробная информация к требованию для хостов, где будут развернуты модули, указана в инструкции по установке этих модулей.

Обратите внимание: на один сервер может быть установлено только по одному экземпляру модулей из списка ниже:

- Сначала следует установить на сервер модуль веб приложения — **APP**. Данный модуль отвечает за веб-интерфейс ПО, логику работы всей системы и взаимодействует с опциональными модулями(OSA). APP модуль может быть развёрнут только на одном сервере в единственном экземпляре, т.к. к нему привязывается лицензия, и он не может быть реплицирован.
- Модуль анализа состава программного обеспечения — **OSA** (опциональный), может быть установлен совместно с другими модулями или вынесен на отдельный сервер. Может быть развёрнут на нескольких серверах одновременно (увеличивает пропускную способность установки по сканированию).

6.1.2. Инструкция по установке системы

Первым устанавливается модуль веб-приложения — APP. Далее устанавливаются опциональные модуль OSA.

Установка APP модуля

1. Скачать и распаковать архив.
2. Открыть в терминале директорию распакованного архива.
3. Выполнить команду:

   ```
   sudo bash actions.sh | sudo tee -a /tmp/appscreener_APP.log
   ```

   и следовать инструкциям.

Установка опциональных модулей - OSA

1. Скачать и распаковать архив (архивы).
2. Открыть в терминале директорию распакованного архива.

3. Выполнить команду в зависимости от устанавливаемого модуля:
   - OSA модуль: `sudo bash actions.sh | sudo tee -a /tmp/appscreener_OSA.log`

   и следовать инструкциям.
4. При установке модуля на сервер, отличный от сервера модуля APP, в файле /opt/appscreener/core/osa/configs/osa-daemon.env заменить значение поля **queueURI** на адрес сервера с установленным модулем APP, например: queueURI=tcp://10.10.10.10:61616 После чего перезапустить службу модуля командой:

   `sudo systemctl restart appscreener-osa.service`

Дальнейшая установка и настройка системы

1. На сервере с развёрнутым APP модулем в браузере перейти по адресу - `http://localhost` , или на рабочей станции в сети сервера в браузере открыть адрес - `http://<APP_module_installation_address>` .
2. Во всплывающем окне с предложением загрузить лицензию скопировать идентификатор установки и отправить его по почте для создания лицензии под конкретную инсталляцию.
3. Загрузить полученную лицензию через интерфейс.
4. Зайти в систему: логин: **admin**, пароль: ***put_admin_password_here***.
5. Сменить пароль учётной записи в **Личном кабинете**.
6. Загрузить файл **Rules.zip** (находится в архиве с APP модулем). Войти в систему как администратор, выбрать и загрузить файл Rules.zip на вкладке **Администрирование** > **Настройки системы** > **Правила**.

Руководство по работе с системой можно загрузить из интерфейса, на вкладке **О продукте**. Плагины для интеграций с TeamCity, Jenkins, Azure DevOps Server, SonarQube, Visual Studio, а также Command Line Tool расположены в директории на сервере с установленным APP модулем - **/opt/appscreener/app/plugins**.

Примеры интеграции с GitLab CI расположены в **/opt/appscreener/app/integration-patterns/gitlab-clt.zip** или **/opt/appscreener/app/integration-patterns/gitlab-api.zip**.

Лог установки/обновления располагается в файле **/tmp/appscreener_app.log**.

6.2. Вход в систему

Для входа в веб-интерфейс Solar appScreener (далее UI) введите в адресной строке браузера адрес **http://<host>**, где host – адрес сервера, на который был установлен Solar appScreener, или адрес сервера на котором установлен модуль веб-приложения (APP модуль), в случае если установка проводилась на нескольких серверах. Появится окно авторизации (рис. 6.1).

Для входа в систему введите логин, пароль и нажмите **Войти**.

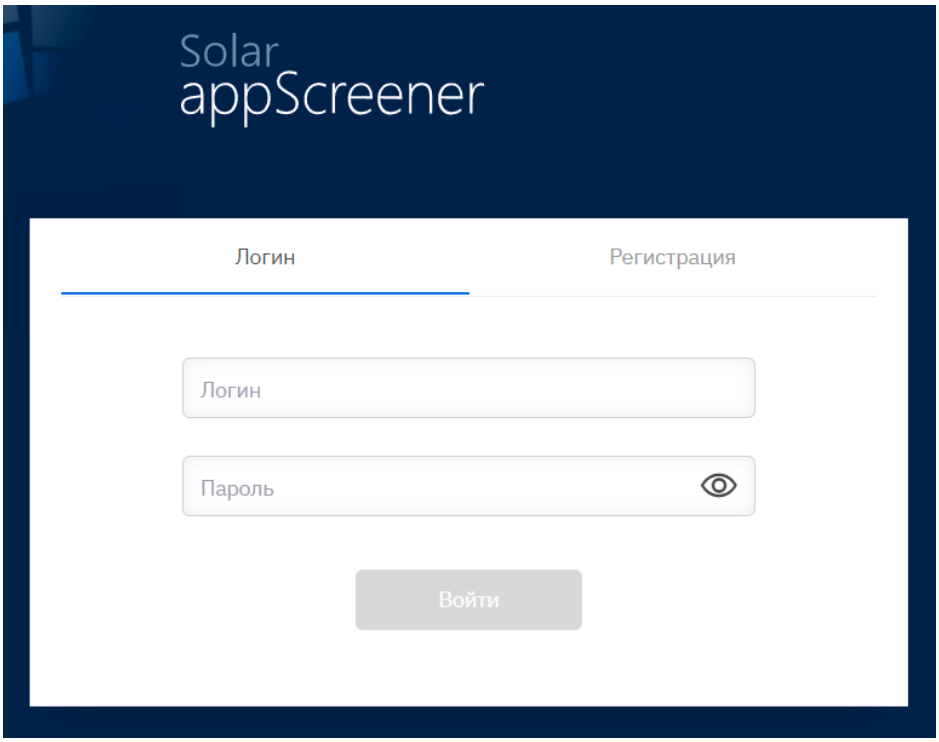

Рис. 6.1: Авторизация пользователя

При введении неверных идентификационных данных на экране отобразится сообщение **Неверный логин и/или пароль**.

Прежде чем начать использование Solar appScreener, необходимо ознакомиться с Пользовательским соглашением и нажать кнопку **Принимаю**

После успешного входа в систему отображается **Домашняя страница**.

При наличии созданных проектов **Домашняя страница** будет содержать до 4-х последних запущенных сканирований в проектах.

6.3. Управление учётными записями пользователей

В Solar appScreener реализован механизм ролевого разграничения доступа. За каждым пользователем может быть закреплена роль, определяющая его возможности в системе. На странице **Администрирование** > **Пользователи** > **Список пользователей** > *имя пользователя* можно настроить конкретные права пользователя в системе и доступы к проектам и группам проектов, выбрав для него одну из ролей или задав необходимые параметры вручную. Дополнительно, на странице **Администрирование > Настройки системы > Общие > Управление пользователями** > блок **Проект** можно настроить права по умолчанию для всех пользователей. Данные права присваиваются, если на странице пользователя вручную выдать доступ к необходимым проектам. Также можно настроить права и доступы для группы пользователей на странице **Администрирование** > **Пользователи** > **Группы пользователей** > *имя группы*.

Изменить настройки ролей на странице пользователя нельзя.

Если права, назначенные пользователю в настройках учётной записи и группе пользователей, не совпадают, ограничения и разрешения суммируются.

6.4. Управление ролями пользователей

Управление ролями пользователей производится на странице **Администрирование > Пользователи > Роли пользователей**.

В Solar appScreener реализованы следующие роли: администратор, модератор, пользователь. Также есть возможность создания пользовательских ролей для более гибкого разграничения возможностей. Доступ к существующим проектам в системе нужно выдавать отдельно на странице пользователя.

Администратор — пользователь с максимальными привилегиями. Ему доступны:

- раздел **Администрирование**;
- все проекты с максимально доступными правами;
- все доступные виды сканирования;
- все приватные сущности, созданные другими пользователями.

Также к его функциям относится управление учётными записями пользователей.

Модератор имеет следующий набор прав:

- работа с проектами и правилами;
- просмотр раздела **Администрирование**;
- все доступные виды сканирования;
- просмотр, изменение, удаление и сохранение всех приватных сущностей, которые создали другие пользователи.

Модератор может просматривать раздел Администрирование, однако не может изменять права и настройки пользователей.

Пользователь имеет стандартный набор прав, позволяющих выполнять анализ и проводить верификацию результатов. По умолчанию, пользователи имеют доступ только к тем проектам, которые они создали. Для доступа к другим проектам необходимо выбрать нужные проекты на странице самого пользователя или на странице группы, которой принадлежит пользователь.

При изменении шаблона роли на странице выбранной роли набор прав у пользователей с этой ролью также обновится в системе.

6.4.1. Управление пользователями

6.4.1.1. Создание учётной записи пользователя

Для создания пользователя:

1. Перейдите в раздел **Пользователи > Создать пользователя**.
2. Введите логин, пароль и ФИО.
3. Введите e-mail, название организации, веб-сайт организации, должность и телефон (опционально).
4. Поля **Учётная запись действительна с/до** позволяют регулировать срок доступа пользователя к системе. До наступления/после завершения срока пользователь не сможет совершить вход в систему, при этом его учётная запись будет доступна администратору. По желанию, администратор сможет изменить срок доступа.

5. Поля **Доступно сканирований OSA** позволяют регулировать количество доступных для пользователя сканирований соответствующего типа. При отсутствии доступных сканирований пользователь также не сможет создать пустой проект.
6. Выберите общие права для пользователя или выберите одну из существующих ролей:
   - **Работать с публичными шаблонами** позволяет создавать публичные шаблоны настроек сканирования и экспорта отчёта. При отсутствии права пользователь сможет создавать только приватные шаблоны;
   - **Работать с группами проектов для всех пользователей** позволяет создавать публичные группы проектов. При отсутствии права пользователь сможет создавать только приватные группы проектов;
   - **Создавать учётные записи через API** позволяет создавать учётные записи новых пользователей через API;
   - **Создавать новые проекты** позволяет создавать новые проекты OSA, включая пустые проекты. Без права пользователь сможет выполнять сканирования только в существующих проектах;
   - **Устанавливать эксклюзивный приоритет сканирований** позволяет запускать сканирования с максимальным уровнем приоритета;
   - **Экспортировать или импортировать проекты** позволяет выгружать или загружать проекты между разными установками Solar appScreener;
   - **Загружать приложения с локального компьютера или из репозитория** позволяет загружать проекты указанным способом;
   - **Запускать сканирования SCA** позволяет запускать сканирования анализа состава ПО;
   - **Запускать сканирования Supply chain** позволяет запускать сканирования цепочек поставок.
7. Настройте параметры, которые не будут отображаться для других пользователей в сканированиях, запущенных данным пользователем, в разделе **Ограничения видимости**.
8. Настройте доступ к проектам в системе.
9. Выберите права в проектах анализа состава ПО или в группах проектов. Права пользователя на проект, на группу проектов и полученные через **группу пользователей** работают по принципу объединения. Если пользователь является автором проекта, он получает все доступные права в проекте (за исключением ограничений, которые могут быть заданы в разделах **Общие права доступа**, **Ограничения видимости**).
10. Нажмите **Сохранить**.

Для редактирования/удаления пользователя либо редактирования его прав нажмите на логин пользователя в списке, внесите требуемые изменения и нажмите **Сохранить/Удалить пользователя**.

6.4.1.2. Блокировка пользователя

В форме редактирования пользователя предусмотрен механизм ручной блокировки пользователя. Чтобы заблокировать пользователя:

1. Кликните на логин пользователя в списке пользователей.

2. В открывшейся форме редактирования пользователя укажите причину блокировки. Причина отобразится пользователю при попытке входа.
3. Нажмите **Заблокировать**.

После нажатия кнопки **Заблокировать** причину блокировки указать будет нельзя.

Для разблокировки пользователя перейдите в форму редактирования заблокированного пользователя и нажмите **Разблокировать**.

6.4.2. Управление группами

Для удобства назначения ролей пользователи могут быть объединены в группы. Чтобы создать группу пользователей:

1. Перейдите в раздел **Группы пользователей > Создать группу**.
2. Введите имя группы.
3. Добавьте описание группы (опционально).
4. Выберите состав группы из списка пользователей.
5. Выберите общие права и права в проектах.
6. Нажмите **Сохранить**.

Для редактирования/удаления группы нажмите на название группы в списке, внесите требуемые изменения и нажмите **Сохранить/Удалить группу**.

6.5. Администрирование системы

В разделе **Администрирование > Настройки системы** можно управлять настройками Solar appScreener, загружать правила поиска уязвимостей, обновлять лицензию и базы уязвимостей, добавлять настройки LDAP.

6.5.1. Агенты сканирования

На странице отображается информация по подключённым агентам, а также можно просмотреть статус агента (активен/недоступен), характеристики и графики по используемым ресурсам.

В разделе **Настройки системы > Агенты сканирования** можно управлять агентами сканирования:

- **Остановить приём сканирований** — останавливает приём сканирований выбранных агентов;
- **Запустить приём сканирований** — запускает приём сканирований выбранных агентов;
- **Удалить** — удаляет остановленные выбранные агенты.

6.5.2. Общие настройки

На вкладке **Настройки системы > Общие** можно управлять настройками Solar appScreener.

6.5.2.1. Система

В разделе **Система** можно работать со следующими настройками:

- **Очистка директории анализа** — выбор чекбокса позволяет сохранять только те файлы, в которых были найдены уязвимости. Остальные файлы удаляются из системы.
- **Язык по умолчанию** — поле для выбора языка ответа на API-запросы.
- **Максимальный размер загружаемого файла** — в поле можно выставить лимит на размер загружаемого файла в битах.
- **Продолжительность сессии** — определяет время, по истечении которого нужно будет повторно авторизоваться в веб-интерфейсе Solar appScreener.
- **Приостановить систему** — для приостановки всех активных сканирований. Прогресс сохраняется.
- **Максимальный размер файла после распаковки** — ограничение на размер файлов после распаковки, в байтах.
- **Рекурсия** — глубина распаковки вложенных архивов, включая `.zip`, `.jar` и т.д.
- **Внешний адрес системы** — IP адрес или доменное имя хоста для доступа к веб-интерфейсу системы. Это необходимо для настройки автоматического сканирования с использованием webhook, работы Swagger и формирования ссылок в отчетах, таск-менеджере, отправляемых письмах.

6.5.2.2. Почта

В разделе **Почта** можно работать со следующими настройками:

- **Администратор** — на указанные здесь электронные адреса будут приходить уведомления о запусках сканирования и сбоях.
- **Обратная связь** — указанные здесь пользователи будут получать отзывы о работе системы.
- **От** — в этом поле можно указать адрес, который будет указан как отправитель.
- **Хост** — хост, используемый на почтовом сервере для подключения.
- **Localhost** — доменное имя почтового сервера.
- **Пароль** — пароль, используемый для аутентификации.
- **Порт** — порт, используемый на почтовом сервере для подключения.
- **SSL** — ssl для почтового сервера, принимает значения true/false.
- **TLS start** — starttls для почтового сервера, принимает значения true/false.
- **Пользователь** — логин, используемый для авторизации.

В разделе также можно настроить системные оповещения. Для оповещений об окончании лицензии и подписки на тех. поддержку можно настроить период и частоту напоминаний, а также выбрать вариант отображения: письма и/или оповещения в интерфейсе.

6.5.2.3. Управление пользователями

Права по умолчанию позволяет настроить:

- **Пользователь, созданный через API** — выбрать для пользователя, созданного через REST, роль и максимальное количество доступных сканирований.
- **Пользователь, синхронизированный из LDAP** — выбрать для пользователя, синхронизированного из AD, роль и максимальное количество доступных сканирований.
- **Пользователь, созданный администратором** — выбрать для пользователя, созданного администратором, роль и максимальное количество доступных сканирований.

Ротация пароля позволяет настроить:

- **Ротация пароля** — активация чекбокса включает принудительную смену пароля учётных записей пользователей с определённой периодичностью.
- **Срок действия пароля** — в поле можно указать продолжительность действия пароля.
- **Ежедневно уведомлять пользователя о истечении срока действия пароля** — в поле можно указать срок, в течение которого пользователь будет получать оповещения о необходимости поменять пароль учётной записи.

Блокировка пользователя при превышении лимита попыток входа позволяет настроить:

- **Блокировать пользователя при превышении лимита попыток входа** — активация чекбокса включает блокировку пользователя при превышении попыток входа с неверным паролем.
- **Срок блокировки** — в поле можно указать продолжительность действия блокировки пользователя.
- **Лимит попыток входа** — в поле можно указать лимит попыток авторизации с неверным паролем.

6.5.3. Данные и память

В разделе можно настроить автоматическое удаление сканирований в проектах для освобождения памяти от ненужных или устаревших сканирований или удалить вручную сканирования, подходящие под выставленные условия.

Автоматическое удаление сканирований — позволяет настроить:

- **Удалять сканирования** — сканирования полностью удаляются из системы.
- **Условие** — выбор удаления сканирований по достижению установленного значения, при его достижении первое сканирование будет удалено, а последнее сохранится, или по возрасту сканирования, при достижении установленного значения все сканирования старше этого срока будут удалены.

Ручное удаление сканирований позволяет настроить **Условие** — выбор удаления сканирований, выполненных до установленной даты, или по статусу сканирования.

6.5.4. Работа с правилами

6.5.4.1. Анализ сторонних компонент

Для обновления выберите агент, для которого нужно провести обновление, и загрузите новую версию баз с локального компьютера по кнопке **Загрузить локально**.

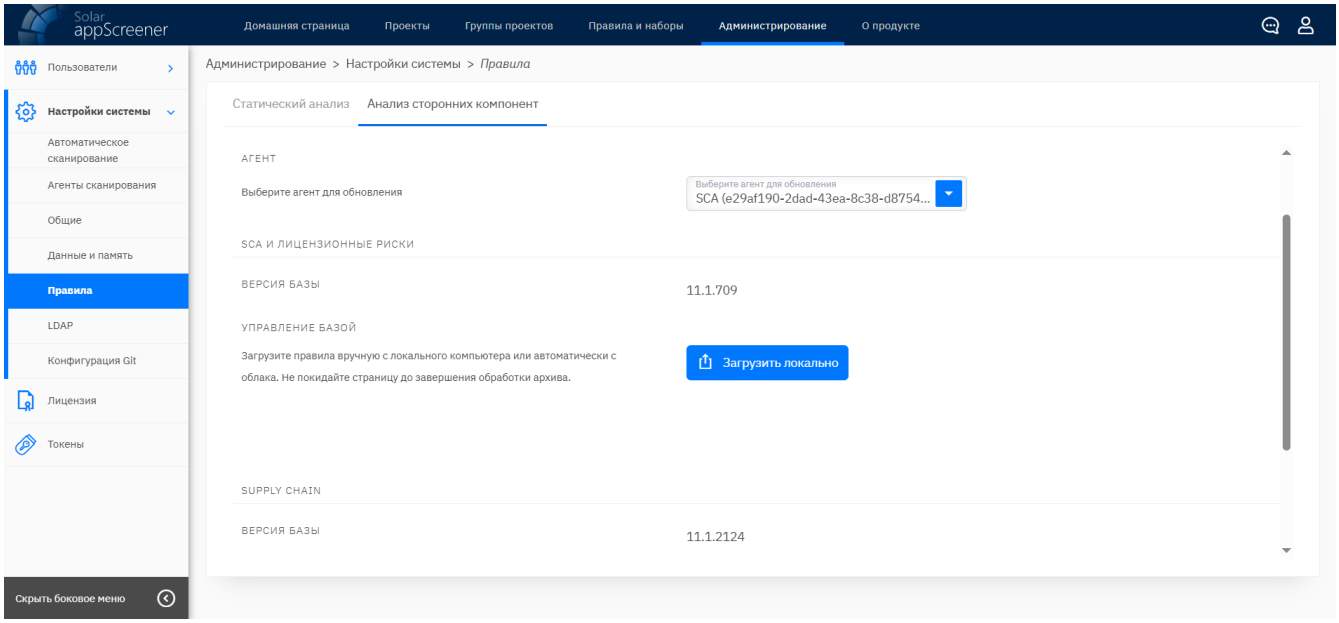

Рис. 6.2: Раздел Настройки системы > Правила > Анализ сторонних компонент

6.5.5. Настройка LDAP

Для добавления нового подключения **LDAP** (рис. 6.3):

1. Перейдите в **Настройки системы > LDAP**.
2. Нажмите **Добавить подключение**.
3. Укажите **параметры** нового подключения (в тултипе справа от поля можно найти подсказку). Обратите внимание, что в некоторых случаях для AD может требоваться подключение только по доменному имени (не по ip и другим адресам).
   - DNS сервер или `C:\windows\system32\drivers\etc\hosts` содержит адрес, совпадающий с главным доменным именем сертификата.
   - **Имя домена** содержит домен, соответствующий домену в учётных записях пользователей после “@”. Например, чтобы пользователь user@test-domain смог авторизоваться в LDAP через Solar appScreener, в поле **Имя домена** нужно указать **test-domain**.

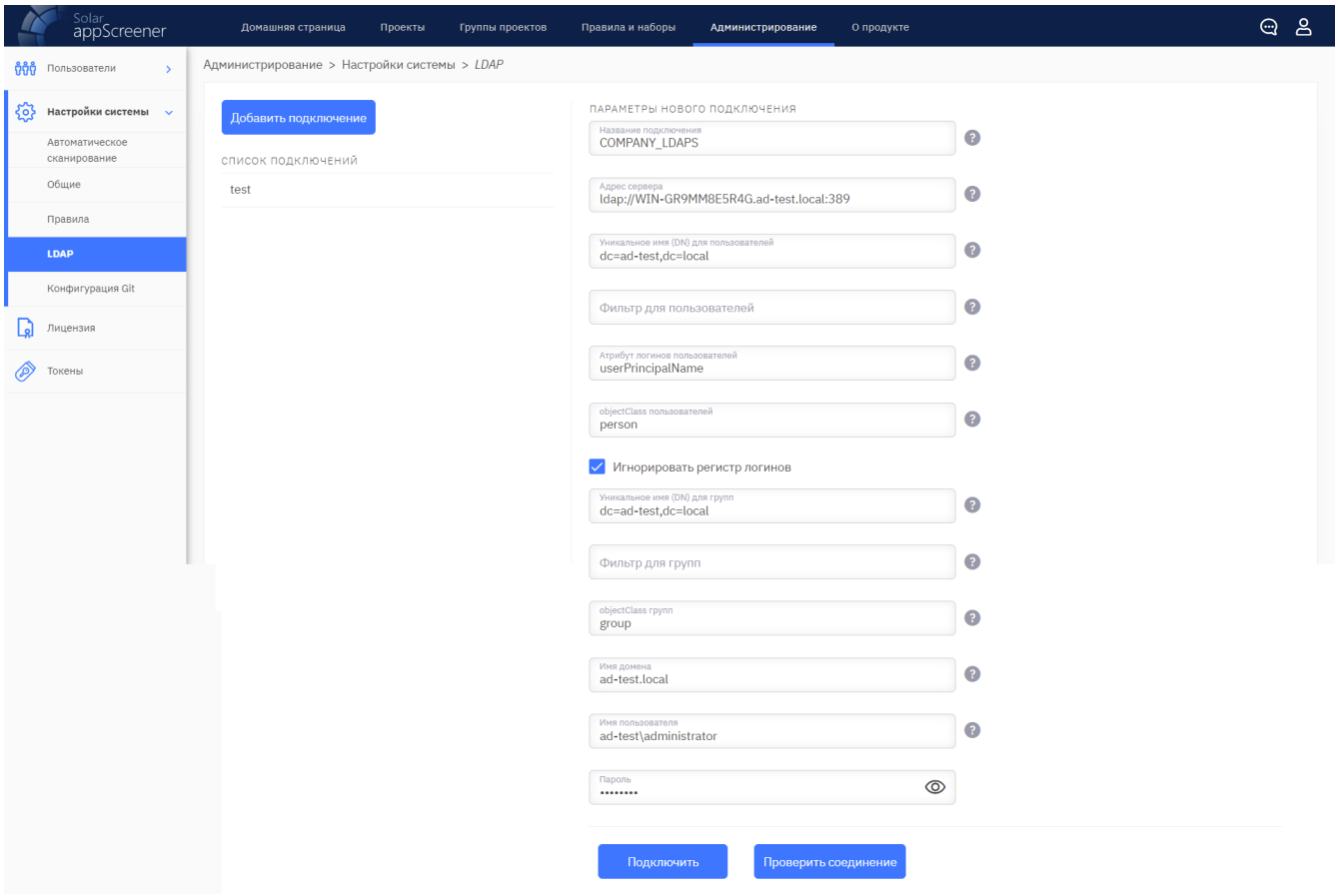

Рис. 6.3: Параметры подключения LDAP

4. Нажмите **Проверить соединение**, после успешной проверки нажмите **Подключить**.
5. Синхронизируйте пользователей LDAP. Для синхронизированных пользователей система создаст учётные записи в Solar appScreener. Работать с ними можно как с локальными пользователями. Ранее синхронизированные пользователи будут отображаться в таблице подключения.

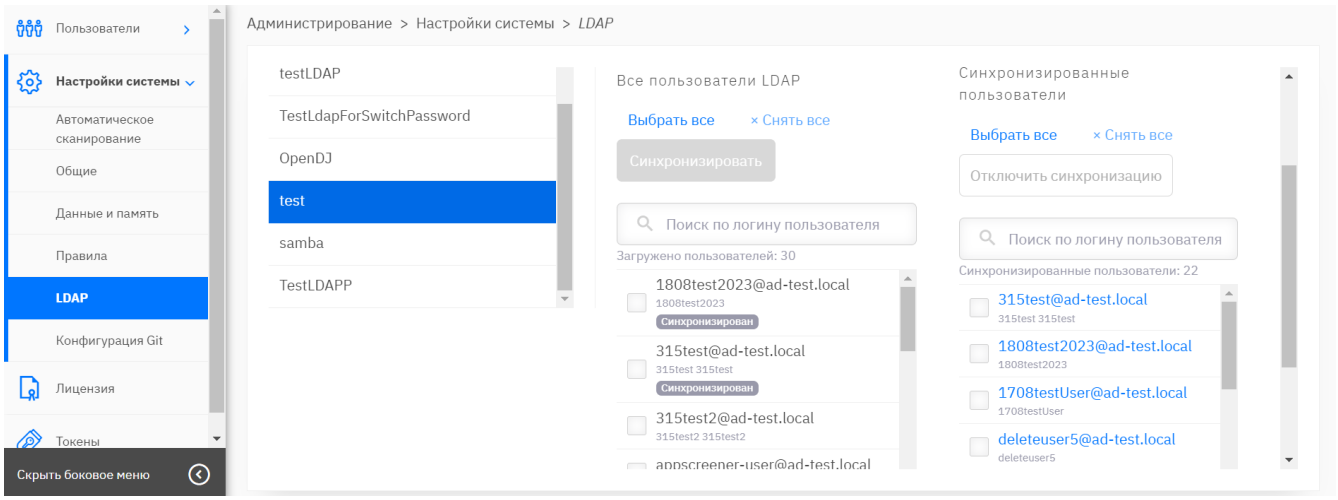

Рис. 6.4: LDAP: синхронизация пользователей

6.5.6. Конфигурация Git

Добавляйте и редактируйте настройки клиента Git в этом разделе. В системе может быть сохранена только одна конфигурация.

Для успешного сканирования проектов с подмодулями требуется заполнить поле `helper = cache --timeout`. По умолчанию Git не кеширует учётные данные. Каждое подключение будет запрашивать логин и пароль. В режиме cache учётные данные хранятся в памяти в течение установленного периода времени, после чего удаляются. Обратите внимание, что данные хранятся в вашем домашнем каталоге *в открытом виде*.

Пример конфигурации:

```
[user]
email = your@email.com
user = your_username
[credential]
helper = cache --timeout 30000
```

6.5.6.1. Игнорирование сертификата

Чтобы игнорировать валидацию самоподписанных сертификатов сервисом Git, добавьте в текст конфигурации условие: Для всех адресов:

```
[http]
sslVerify = false
```

Только для домена weak.example.com:

```
[http "https://weak.example.com"]
sslVerify = false
```

6.5.6.2. Проверка сертификата

Если требуется валидация самоподписанных сертификатов сервисом Git, выполните:

1. Создайте самоподписанный сертификат для доменного имени (опция **Common name**).
2. В интерфейсе Solar appScreener (Администрирование > Настройки системы > Конфигурация Git) добавьте в конфигурацию Git:

Добавить в доверенные Git для всех адресов:

```
[http]
sslCAInfo = /opt/backend/files/certs.crt
```

Добавить в доверенные Git только для домена weak.example.com:

```
[http "https://weak.example.com"]

sslCAInfo = /opt/backend/files/certs.crt
```

3. На сервере с модулем APP создайте файл /opt/appscreener/app/services/backend/files/certs.crt и добавьте содержимое самоподписанного сертификата в формате:

```
-----BEGIN CERTIFICATE-----

CERTIFICATE_SAMPLE_1

-----END CERTIFICATE-----
```

4. При необходимости, добавление новых самоподписанных сертификатов происходит путём редактирования файла /opt/appscreener/app/services/backend/files/certs.crt:

```
-----BEGIN CERTIFICATE-----

CERTIFICATE_SAMPLE_1

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

CERTIFICATE_SAMPLE_2

-----END CERTIFICATE-----

...
```

6.5.7. Лицензия

Для обновления **Лицензии**:

1. Перейдите в **Настройки системы > Лицензия**.
2. Выберите файл с лицензией.
3. Нажмите **Сохранить**.

6.6. Система регистрации событий

Подсистема регистрации событий Solar appScreener регистрирует события для каждого модуля (APP, OSA).

Подсистема регистрации событий фиксирует следующие события:

- действия пользователей:
  - запуск, остановка, просмотр результатов и другие действия со сканированиями и проектами;
  - авторизация;
  - действия с правилами;
  - действия администратора;
- технические события системы:
  - состояние запущенных сканирований;
  - состояние работы подключенных модулей анализа;

- блокировка/разблокировка пользователей;
- отправка e-mail сообщений;
- проверка ограничений пользователя;
- сообщения об ошибках системы;
- файловые операции.

Все события заносятся в журналы событий.

6.6.1. Журналы событий

Solar appScreener может создавать несколько файлов журналов событий. Они отличаются уровнем детализации:

- ERROR - события с ошибками системы;
- WARN - события с предупреждениями, которые потенциально могут привести к событиям с ошибкой;
- INFO - события с информацией состояния/действий системы;
- DEBUG - события с отладочными данными для анализа работы системы разработчиками.

Все модули записывают и хранят журналы событий в файловой системе сервера, на котором они функционируют.

Журналы событий APP модуля

В составе веб-приложения (APP модуль) работают нескольких подсистем:

- NGINX - отвечает за хранение каркаса портала и служит reversproxy для обращений извне. Файлы журнала событий NGINX находятся на установке системы по пути **/opt/appscreener/app/services/frontend/logs** и состоят из:
  - access.log (журнал обращений клиентов к веб-серверу);
  - error.log (журнал ошибок веб-сервера).
- BACKEND - отвечает за логику работы веб-приложения. Файлы журнала событий подсистемы находятся на сервере по пути **/opt/appscreener/app/services/backend/logs** и состоят из:
  - appscreener-error.log (только события с ошибками);
  - appscreener-warn.log (события с ошибками и предупреждениями);
  - appscreener-debug.log (события с ошибками, предупреждения, информация о работе системы и отладочные данные).
- ACTIVEMQ ARTEMIS - отвечает за контроль работы анализаторов (очередь сканирований). Файлы журнала событий ActiveMQ Artemis можно получить на сервере с работающим модулем APP, выполнив команду:

  ```
  sudo docker logs artemis
  ```

- POSTFIX - отвечает за работу почтового сервера, для отправки из системы Solar appScreener сообщений. Файлы журнала событий postfix можно получить на сервере с работающим модулем APP, выполнив команду:

  ```
  sudo docker logs postfix
  ```

Журналы событий OSA модуля

Модуль отвечает за работу алгоритмов анализа состава программного обеспечения. При работе модуля на нескольких серверах, журналы событий будут храниться в файловой системе сервера отдельно для каждого экземпляра модуля. Журналы модуля хранятся в директории **/opt/appscreener/core/osa/services/osa-daemon/logs**.

Все журналы производят ежедневную ротацию и архивирование в аллоцированные директории.

6.6.2. Отправка журналов событий ИБ по syslog

Логируемые события:

- сведения об операциях входа/выхода в/из системы;
- сведения об операциях по управлению учётными записями;
- сведения об операциях по управлению ролями и привилегиями пользователей;
- сведения об изменении настроек конфигурации;
- сведения об ошибках/сбоях системы.

Чтобы отправлять журналы событий Solar appScreener на другой сервер по протоколу **syslog** в формате `.cef`:

1. Откройте файл **/opt/appscreener/app/configs/backend.env**.
2. Добавьте в него 3 переменные среды:
   - **syslog.host** — IP-адрес сервера, на который необходимо отправлять журналы событий;
   - **syslog.port** — порт, по которому будут отправляться журналы событий;
   - **syslog.protocol** — протокол, используемый для отправки журналов событий.

   Пример значений:

   ```
   syslog.host=10.208.1.1
   syslog.port=514
   syslog.protocol=TCP
   ```
3. Сохраните изменения.
4. Перезагрузите сервис APP модуля:

   ```
   sudo systemctl restart appscreener-app
   ```

6.6.3. Log-bringer

Log-bringer необходим для автоматического сбора журналов событий системы. Программа собирает все журналы и полезную системную информацию в один архив. Log-bringer поставляется с каждым модулем и встраивается в систему как системная программа. Чтобы запустить, необходимо предоставить Log-bringer права на выполнение.

Если установлен Solar appScreener версии 3.15 или выше, программа для сбора журналов событий вшита в модули системы (APP, OSA) по умолчанию. Если

установлена версия 3.14 и ниже, актуальная программа предоставляется по запросу в техническую поддержку.

Для запуска необходимо выполнить команду: `sudo log-bringer`

Программа запросит выбор даты, начиная с которой нужно собрать журналы событий. Выберите нужную дату. Программа автоматически соберет журналы событий с выбранной даты по текущую и создаст архив с ними.

Результатом выполнения является архив с журналами событий. Архив защищен паролем. В таблицах указаны пути и содержание архива.

Общие журналы событий

Путь	Описание
log_bringer.log	ошибки при исполнении программы
system/cpuinfo.log	информация о процессоре
system/meminfo.log	информация об оперативной памяти
system/os_info.log	информация об ОС
system/disk_space.log	информация о размере диска
system/kernel.log	информация об ошибках в ядре ОС
system/processes.log	список запущенных процессов в системе
system/vm_state.log	информация об использованных ресурсах
system/ip_addr.log	информация о сетевых настройках сервера
system/loadavg.log	информация о средней загрузке процессора
system/uptime.log	информация о времени работы с момента последнего перезапуска системы

Docker журналы событий

Путь	Описание
docker/docker_info.log	информация о версии docker и docker compose
docker/daemon.json	информация о настройке сервера docker (/etc/docker/daemon.json)
docker/docker_ps.log	таблица с основной информацией о контейнерах (docker ps -a)
docker/container_top.log	расширенная информация о контейнерах
docker/docker_images.log	информация о загруженных образах
docker/docker_volumes.log	информация о созданных томах в docker
docker/ docker_system_disk_free.log	информация об использованном пространстве docker на диске
docker/ docker_system_disk _free_detailed.log	детальная информация об использованном пространстве docker на диске
docker/docker_stats.log	информация об использованных ресурсах docker контейнеров

Журналы событий модуля APP

Путь	Описание
app-module	каталог с информацией о Frontend, Backend, ActiveMQ Artemis, Postgres, Postfix для модуля APP
app-module/app.compose.yml	docker-compose конфигурация для модуля APP
app-module/ systemd_app_service.log	информация об APP systemd сервисе
app-module/app-db/postgres.env	конфигурация для app-db сервиса; первая инициализация соединения с БД
app-module/app-db/ container_app-db.log	журналы из контейнера app-db; основной файл журнала postgresql
app-module/frontend	конфигурации и журналы для NGINX
app-module/frontend/ default.conf.template	конфигурация NGINX (старая версия)
app-module/frontend/ frontend.env	конфигурация NGINX (новая версия, начиная с 3.14.5)
app-module/frontend/ logs/access.log	журналы доступа к NGINX
app-module/frontend/ logs/error.log	журналы ошибок NGINX
app-module/backend	каталог, в котором находятся журналы и конфигурации для Backend
app-module/backend/license.xml	лицензия на продукт
app-module/backend/backend.env	конфигурация для Backend
app-module/backend/logs	каталог, основные журналы веб-приложения
app-module/artemis/ container_artemis.log	журналы из контейнера ActiveMQ Artemis
app-module/postfix/ container_postfix.log	журналы из контейнера Postfix

Журналы событий модуля OSA

Путь	Описание
osa-module	каталог с конфигурациями и журналами для модуля OSA
osa-module/ osa.compose.yml	конфигурация docker-compose для модуля OSA
osa-module/ systemd_osa_service.log	информация о службе OSA systemd
osa-module/osa-daemon/ osa-daemon.env	конфигурация для службы osa-daemon
osa-module/osa-daemon/ container_osa-daemon.log	журналы из контейнера osa-daemon

Путь	Описание
osa-module/osa-daemon/logs	каталог, журналы событий osa-daemon
osa-module/osa/container_osa.log	журналы из контейнера OSA
osa-module/osa-db/container_osa-db.log	журналы из контейнера osa-db
osa-module/sbom-generator/container_sbom-generator.log	журналы из контейнера SBOM-генератора

6.6.4. Основные логируемые события

Тип события	Атрибуты	Файл
Попытка входа в систему	время и ID события, логин пользователя, имеет ли пользователь права администратора, LDAP URL, ID пользователя	appscreener-debug.log
Инициализация поиска соответствия данных с БД	время и ID события, логин пользователя	appscreener-debug.log
Создание токена сессии	время и ID события, логин пользователя, имеет ли пользователь права администратора, LDAP URL, ID пользователя	appscreener-debug.log
Получение токена в Личном Кабинете	время и ID события, логин пользователя, время действия токена в минутах, имеет ли пользователь права администратора, LDAP URL, ID пользователя	appscreener-debug.log
Попытка изменить пароль	время и ID события, логин пользователя, имеет ли пользователь права администратора, LDAP URL, ID пользователя	appscreener-debug.log
Ошибка ввода неверного пароля	время и ID события, логин пользователя	appscreener-debug.log
Ошибка ввода неверного логина	время и ID события, логин пользователя	appscreener-debug.log

Тип события	Атрибуты	Файл
Обновление времени действия токена сессии	время и ID события, логин пользователя, время действия токена в минутах, имеет ли пользователь права администратора, LDAP URL, ID пользователя	appscreener-debug.log
Попытка использования просроченного токена сессии	время и ID события, время истечения действия токена, текущее время, разница в мс	appscreener-debug.log
Запрос на получение списка проектов	время и ID события, вид и порядок сортировки, требуются ли архивированные проекты, требуются ли проекты без сканирований, название проекта, статусы проектов имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	appscreener-debug.log
Запрос информации о проекте	время и ID события, UUID проекта, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	appscreener-debug.log
Запрос информации о сканировании	время и ID события, UUID сканирования, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	appscreener-debug.log
Запрос информации о списке сканирований	время и ID события, UUID проекта и сканирований, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	appscreener-debug.log
Запрос информации о статистике	время и ID события, UUID сканирования, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	appscreener-debug.log
Запрос информации об уязвимости	время и ID события, ID уязвимости, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	appscreener-debug.log

Тип события	Атрибуты	Файл
Запрос на скачивание отчёта	время и ID события, UUID сканирования, UUID проекта, UUID всех уязвимостей, название проекта, ссылка на подробные результаты, путь до логотипа, количество сканирований, дата создания проекта, версия приложения, автор проекта, скрытые настройки пользователя: ID пользователя, UUID настроек, ограничения видимости, язык, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	appscreener-debug.log
Запрос на получение списка пользователей	время и ID события, настройки фильтрации пользователей: сортировка, направление сортировки, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	appscreener-debug.log
Создание пользователя	время и ID события, объект userDto: логин, email, ФИО, организация, должность, номер телефона, веб-сайт, дата истечения срока действия учётной записи, дата активации учётной записи, максимальное число сканирований, дата блокировки учётной записи, дата окончания блокировки учётной записи, дата окончания действия пароля, ограничение на неправильный ввод пароля, скрытые настройки пользователя: ID пользователя, UUID настроек, ограничения видимости, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	appscreener-debug.log

Тип события	Атрибуты	Файл
Изменение пользователя	время и ID события, объект userDto: логин, email, ФИО, организация, должность, номер телефона, веб-сайт, дата истечения срока действия учётной записи, дата активации учётной записи, максимальное число сканирований, дата блокировки учётной записи, дата окончания блокировки учётной записи, дата окончания действия пароля, ограничение на неправильный ввод пароля, скрытые настройки пользователя: ID пользователя, UUID настроек, ограничения видимости, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	appscreener-debug.log
Удаление пользователя	время и ID события, UUID удаляемого пользователя, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	appscreener-debug.log
Создание проекта	время и ID события, настройки проекта: ссылка, логин репозитория, название файла, ветка, название проекта, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	appscreener-debug.log
Изменение проекта	время и ID события, UUID настроек проекта, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	appscreener-debug.log

Тип события	Атрибуты	Файл
Удаление проекта	время и ID события, UUID проекта, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	appscreener-debug.log
Запуск сканирования	время и ID события, настройки проекта: ссылка, логин репозитория, название файла, ветка, название проекта, путь до временного файла (при сканировании архива), команда клонирования репозитория (при сканировании из VCS), путь до source.zip, UUID проекта, UUID сканирования, UUID всех запущенных задач, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	appscreener-debug.log
Приостановка сканирования	время и ID события, UUID сканирования, UUID всех незавершённых задач, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	appscreener-debug.log
Остановка сканирования	время и ID события, UUID сканирования, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	appscreener-debug.log
Удаление сканирования	время и ID события, UUID сканирования, директория с ресурсами проекта, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	appscreener-debug.log

Тип события	Атрибуты	Файл
Изменение атрибутов найденной уязвимости	время и ID события, UUID уязвимости, UUID сканирования, UUID проекта, старый и новый комментарий (в случае изменения комментария), старый и новый уровень критичности (при изменении уровня критичности), старый и новый статус (при изменении статуса), имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	appscreener-debug.log
Загрузка правил в систему	время и ID события, название архива, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	appscreener-debug.log
Ввод неверного пароля	время и ID события	appscreener-debug.log
Ввод неверного логина	время и ID события	appscreener-debug.log
Загрузка проекта по ссылке	время и ID события, ссылка	appscreener-debug.log
Попытка создания пользователя, который уже существует	время и ID события	appscreener-debug.log
Изменение пароля: новые пароли не совпадают	время и ID события	appscreener-debug.log
Ошибка загрузки архива правил	время и ID события	appscreener-debug.log
Ошибка лицензии: кончилась, доступен только просмотр	время и ID события	appscreener-debug.log
Ошибка места жёсткого диска	время и ID события	appscreener-debug.log
Нехватка оперативной памяти	время и ID события	appscreener-debug.log
Отсутствие ответа демона	время и ID события, UUID задачи	appscreener-debug.log
Отсутствие ответа БД	время и ID события	appscreener-debug.log
Ошибки модуля Matcher	время и ID события, путь до файла	log.log (/opt/appscreener/files/d/{taskuuid}/.state/log.log)

6.7. Резервирование данных

Для резервирования данных системы требуются копии:

1. **Базы данных приложения**. Дамп базы данных можно получить, выполнив на хосте с APP модулем:

   ```
   sudo docker exec app-db pg_dump -U backend backend > db_dump.sql
   ```

2. **Директорий**:

   - /opt/appscreener/app/services/backend/files/
   - /opt/appscreener/app/services/frontend/nginx/ssl/

3. **Файлов конфигурации**:

   - На сервере с APP модулем:
     - /opt/appscreener/app/configs/backend.env
     - /opt/appscreener/app/configs/postgres.env
     - /opt/appscreener/app/services/frontend/nginx/templates/default.conf.template
   - На сервере с OSA модулем:
     - /opt/appscreener/core/osa/configs/osa-daemon.env

4. **Логов приложения** (опционально, см. Журналы событий).

7. ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ О РАБОТЕ С SOLAR APPSCREENER

7.1. Подключение встроенного почтового сервера

В дистрибутив поставки входит почтовый сервер и клиент **Postfix**, который можно использовать как релей для пересылки писем. Клиент использует TLS-протокол для подключения к сторонним почтовым серверам.

Обратите внимание:

- Во внешних сервисах письма могут приходить в папку **Спам**, если явно не указать фильтрацию входящих писем.
- Письма могут приходить с некоторой задержкой (из-за попадания в «Серый список»).
- Журнал событий сервера можно посмотреть с помощью команды, которую необходимо запустить на хосте, где развёрнут **APP** модуль:

```
sudo docker logs postfix
```

Для настройки встроенного почтового сервера:

1. Перейдите в раздел **Администрирование** > **Настройки системы** > **Общие** > **Почта**.
2. Заполните поля **Администратор**, **Обратная связь** и **От**. В поле **Хост** добавьте значение **postfix**, в поле **Порт** — значение **25**, остальные поля оставьте пустыми.

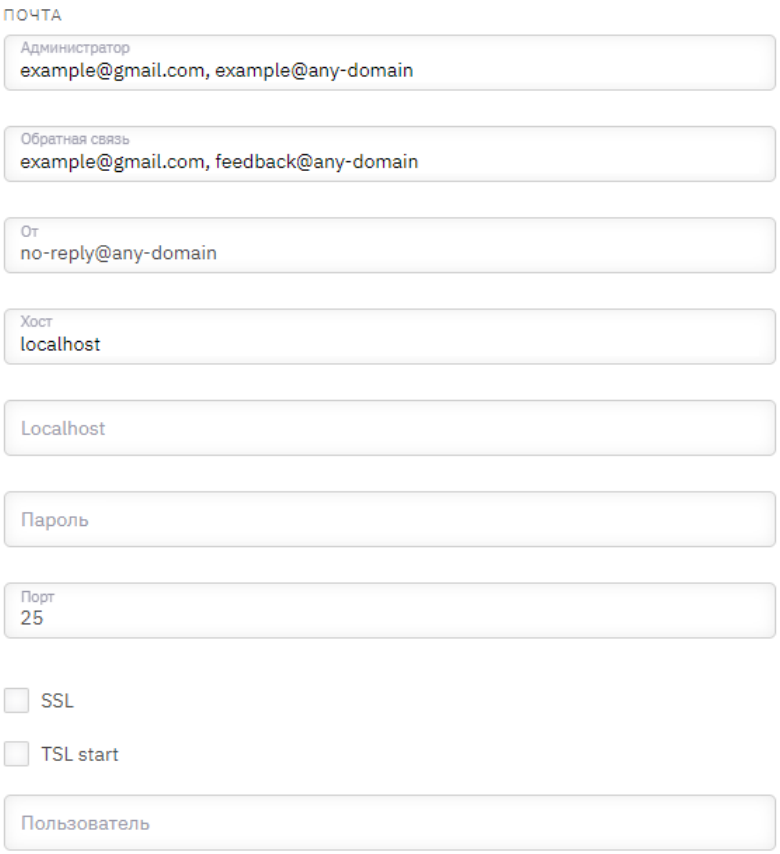

Рис. 7.1: Настройка почтового сервера

3. Настройте политику оповещений в разделе **Администрирование** > **Настройки системы** > **Общие** > **Оповещения**.
4. Нажмите **Сохранить**.

7.2. Подключение приватных Git-репозиториев

7.2.1. Логин/пароль

Подключение приватного репозитория в отдельном сканировании или проекте:

1. Откройте настройки сканирования и перейдите в пункт **Настройки приватного репозитория**.
2. Добавьте логин и пароль приватного репозитория.
3. Активируйте опцию **Использовать данные при пересканировании проекта** (*опционально*). Даёт доступ к приватному репозиторию всем сканированиям в данном проекте.

Сохранение данных для подключения приватного репозитория в учётную запись пользователя:

1. В веб-интерфейсе перейдите в раздел **Профиль > Настройки доступа > Приватный репозиторий > Логин и пароль**.
2. Нажмите **Добавить учётную запись**.

3. Добавьте данные приватного репозитория.
4. Настройте права на использование данных приватного репозитория отдельным пользователям (опционально).
5. Откройте настройки сканирования и перейдите в пункт **Настройки приватного репозитория**.
6. Выберите добавленную учётную запись.

7.2.2. Токен авторизации

Подключение приватного репозитория в отдельном сканировании или проекте:

1. Откройте настройки сканирования и перейдите в пункт **Настройки приватного репозитория**.
2. Добавьте токен приватного репозитория.
3. Активируйте опцию **Использовать данные при пересканировании проекта** (*опционально*). Даёт доступ к приватному репозиторию всем сканированиям в данном проекте.

Сохранение данных для подключения приватного репозитория в учётную запись пользователя:

1. В веб-интерфейсе перейдите в раздел **Профиль > Настройки доступа > Приватный репозиторий > Токен доступа**.
2. Нажмите **Добавить токен**.
3. Добавьте токен в поле **Ключ**.
4. Настройте права на использование токена отдельным пользователям (опционально).
5. Откройте настройки сканирования и перейдите в пункт **Настройки приватного репозитория**.
6. Выберите добавленный токен.

7.2.3. SSH-ключ

Подключение приватного репозитория в отдельном сканировании или проекте:

1. Откройте настройки сканирования и перейдите в пункт **Настройки приватного репозитория**.
2. Добавьте файл приватного ключа или вставьте содержимое в поле **Ключ**.
3. Активируйте опцию **Использовать данные при пересканировании проекта** (*опционально*). Даёт доступ к приватному репозиторию всем сканированиям в данном проекте.

Сохранение данных для подключения приватного репозитория в учётную запись пользователя:

1. В веб-интерфейсе перейдите в раздел **Профиль > Настройки доступа > Приватный репозиторий > SSH-ключ**.

2. Нажмите **Добавить ключ**.
3. Добавьте приватный SSH-ключ.
4. Измените стандартную конфигурацию SSH-клиента (опционально).
5. Настройте права на использование ключа отдельным пользователям (опционально).
6. Откройте настройки сканирования и перейдите в пункт **Настройки приватного репозитория**.
7. Выберите добавленный SSH-ключ.

7.2.4. Подключение загрузки из приватных Git-репозиториев

Конфигурацией для работы с Git-репозиториями можно управлять в UI (**Администрирование** > **Настройки системы** > **Конфигурация Git**).

Пожалуйста, используйте техническую учётную запись, так как данные будут храниться на сервере в открытом виде. В случае приватного репозитория BitBucket скачивание по логину/паролю недоступно. Пожалуйста, используйте токен авторизации или SSH-ключ.

Пример конфигурации:

```
[user]
email = your@email.com
user = your_username
[credential]
helper = cache --timeout 30000
```

7.3. Настройка сервиса NGINX

В составе программного обеспечения используется reverse-proxy сервер **NGINX**. Конфигурационный файл **NGINX** не предусматривает внесение пользовательских изменений. Вместо этого корректировать работу прокси сервера можно изменяя переменные, описанные в файле **/opt/appscreener/app/configs/frontend.env**. Детальная информация о настройке прокси сервера и переменных для конфигурирования описана в данной главе.

7.3.1. Установка HTTPS соединения

По умолчанию приложение работает по протоколу **HTTP** без шифрования. Для смены работы по защищённому протоколу необходимо наличие файла сертификата (например, *cert.cer*) и приватного ключа (например, *private.key*) без пароля для выбранного доменного имени.

Их необходимо загрузить на хост с модулем APP в директорию **/opt/appscreener/app/services/frontend/nginx/ssl** (приватный ключ в формате доступа 600).

После этого нужно отредактировать файл **/opt/appscreener/app/configs/frontend.env**, заменив в нём значение поля **SSL_STATE** на **on** и добавив два дополнительных поля со значениями:

1. Поле **SSL_CERT_NAME**, со значением имени добавленного сертификата.
2. Поле **SSL_KEY_NAME**, со значением имени приватного ключа.

Пример.

```
...
SSL_STATE=on
SSL_CERT_NAME=my_certificate.cer
SSL_KEY_NAME=my_private_key.key
```

Чтобы изменения вступили в силу, перезапустите сервис модуля APP:

```
sudo systemctl restart appscreener-app.service
```

Обратите внимание: редирект запросов с HTTP на HTTPS дополнительно настраивать не требуется, при использовании HTTPS соединения данный функционал включается автоматически.

7.3.2. Настройка таймаута для соединения с сервером

Для конфигурации **NGINX** используются следующие директивы настройки таймаутов:

```
    proxy_connect_timeout       ${TIMEOUT_SCOPE};
    proxy_send_timeout          ${TIMEOUT_SCOPE};
    proxy_read_timeout          ${TIMEOUT_SCOPE};
    send_timeout                ${TIMEOUT_SCOPE};
```

Все параметры регулируются переменной TIMEOUT_SCOPE. Чтобы изменить таймаут, отредактируйте значение TIMEOUT_SCOPE в файле конфигурации **/opt/appscreener/app/configs/frontend.env**. Значение по умолчанию — 600.

Чтобы изменения вступили в силу, перезапустите сервис модуля APP:

```
sudo systemctl restart appscreener-app.service
```

7.3.3. Изменение размера клиентского запроса

Для ограничения размера клиентского запроса в конфигурации **NGINX** используется директива *client_max_body_size*. Чтобы изменить лимит, отредактируйте значение CLIENT_MAX_BODY_SIZE в файле конфигурации **/opt/appscreener/app/configs/frontend.env**. Значение по умолчанию — 4GB.

Чтобы изменения вступили в силу, перезапустите сервис модуля APP:

```
sudo systemctl restart appscreener-app.service
```

Обратите внимание:

С помощью данной переменной можно регулировать максимальный размер архивов, загружаемых на сервер для сканирования, но для этого также необходимо внести соответствующие изменения в интерфейсе Solar appScreener в разделе **Администрирование** > **Настройки системы** > **Общие** > *Максимальный размер загружаемого файла (байты)*. Значение CLIENT_MAX_BODY_SIZE должно быть >= значению в поле **Максимальный размер загружаемого файла**.

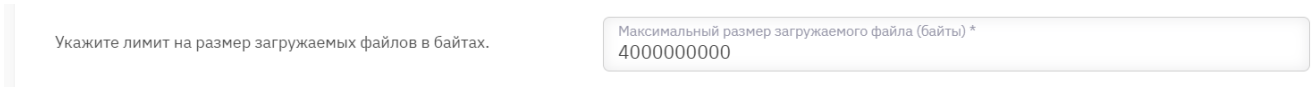

Рис. 7.2: Настройки системы > Общие > Максимальный размер загружаемого файла

7.3.4. Пользовательские настройки конфигурации NGINX

Для кастомизации файла конфигурации NGINX (директивы, контекст) необходимо добавить переменные с многострочным значением в файл **/opt/appscreener /app/configs/frontend.env**. Каждая переменная отвечает за добавление настроек в определённую секцию шаблонного файла конфигурации NGINX.

Секция	Имя переменной
server	SERVER_BASE
location /	LOCATION_ROOT
location /app	LOCATION_APP
location /app/ws	LOCATION_APP_WS
server(redirect)	SERVER_REDIRECT

Для просмотра шаблонного файла конфигурации NGINX выполните команду:

```
sudo docker exec frontend cat /etc/nginx/templates/default.conf.template
```

Для каждой секции в шаблонном файле настроек NGINX есть стандартно настроенные директивы и вложенные секции, переопределение которых запрещено. Переменная должна содержать строку в одинарных или двойных кавычках с директивой (директивами) для конфигурации NGINX. Если значение директивы заключается в кавычки, следует использовать различные типы кавычек (одинарные/двойные) для строки многострочной переменной и значения директивы.

Использовать переменную SERVER_REDIRECT следует только в случае, если настроено соединение по HTTPS и требуется внести дополнения в секцию *server*, отвечающую за перенаправление с HTTP на HTTPS.

Пример использования многострочной переменной для секции *server*, файл **/opt/appscreener/app/configs/frontend.env**

```
CLIENT_MAX_BODY_SIZE=4G

TIMEOUT_SCOPE=600

SSL_STATE=off

SERVER_BASE="
```

```
add_header X-Frame-Option DENY;
add_header X-Content-Type-Options nosniff;
add_header X-XSS-Protection '1; mode=block';"
```

```
    keepalive_timeout 5;

    merge_slashes on;
    if ($request_uri ~ ^[^?]*//) {
        rewrite ^ $uri permanent;
    }

### SERVER_BASE section beginning ###

add_header X-Frame-Option DENY;
add_header X-Content-Type-Options nosniff;
add_header X-XSS-Protection '1; mode=block';
### SERVER_BASE section end ###

    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }

    root   /usr/share/nginx/html;
```

Рис. 7.3: Пример сформированной конфигурации

Чтобы изменения вступили в силу, перезапустите сервис модуля APP:

```
sudo systemctl restart appscreener-app.service
```

Информация о том, что многострочная переменная настроена в файле конфигурации выводится в лог контейнера *frontend*. Посмотреть лог можно выполнив команду:

```
sudo docker logs frontend
```

```
user@Ubuntu-22:~/nginx$ sudo docker logs frontend
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: /etc/nginx/conf.d/default.conf differs from the packaged version
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
20-envsubst-on-templates.sh: Running envsubst on /etc/nginx/templates/default.conf.template to /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/100-ast-preparation.sh
The SERVER_BASE is configured
/docker-entrypoint.sh: Configuration complete; ready for start up
```

Рис. 7.4: Корректный лог после внесения изменений

7.4. Добавление самоподписных и корневых сертификатов в доверенные для работы через HTTPS и LDAPS

Для добавления сертификата в доверенные необходимо:

1. Загрузить на сервер где развёрнуто веб-приложение (APP.module) системы, файл сертификата в формате PEM. Этот формат представляет собой ASCII-файл, закодированный по схеме Base64. Возможные расширения: `.pem`, `.crt`, `.cer`. Для примера используется `.cer`-файл сертификата LDAPS: **example.cer**.
2. Выполнить команды:

```
sudo docker cp example.cer backend:/

sudo docker exec backend keytool -importcert -noprompt -cacerts
-storepass

changeit -file "/example.cer" -alias "example-cer"
```

3. Проверить наличие только что добавленных сертификатов в списке доверенных:

```
sudo docker exec backend keytool -cacerts -storepass changeit -list |

grep example-cer
```

4. Остановить процессы сканирования или дождаться их завершения. Перезапустить работу APP модуля:

```
sudo systemctl restart appscreener-app.service
```

7.5. Увеличение памяти для сервиса Tomcat

Чтобы увеличить объём памяти для сервиса Tomcat:

1. На хосте с модулем **APP** откройте конфигурационный файл: **/opt/appscreener/app/configs/backend.env**
2. В следующей строке замените значение `Xmx4096M` на `Xmx8192M`:
   `CATALINA_OPTS="-Xms1024M -Xmx4096M"`

   Должно получиться: `CATALINA_OPTS="-Xms1024M -Xmx8192M"`
3. Закройте файл, сохранив изменения.
4. Перезапустите модуль **APP**:

```
sudo systemctl restart appscreener-app.service
```

7.6. Что делать, если сканирование завершилось со статусом «Ошибка»?

При возникновении ошибок во время сканирования обратитесь в службу поддержки. К письму приложите:

1. Скриншот страницы сканирования с описанием ошибок. Если ошибок несколько, приложите несколько скриншотов.
2. Файлы журналов событий системы, собранные Log-bringer (подробнее см. Log-bringer).

7.7. Миграция данных Solar appScreener

7.7.1. Миграция сервера на другой хост

Для миграции сервера Solar appScreener с **Host_1** на **Host_2** необходимо:

1. На Host_1 выполнить:

   ```
   sudo docker exec app-db pg_dump -U backend backend > db_dump.sql
   ```

2. Развернуть новую установку Solar appScreener на Host_2.
3. Перенести дамп базы на Host_2.
4. Перенести директорию **/opt/appscreener/app/services/backend/files/** с Host_1 на Host_2 по тому же пути.
5. На Host_2 выполнить:

   ```
   sudo systemctl stop appscreener-app.service
   ```

   ```
   sudo docker volume rm -f app-postgres-data
   ```

   ```
   sudo docker compose -f /opt/appscreener/app/app.compose.yml up -d app-db
   ```

   ```
   sudo docker cp db_dump.sql app-db:/
   ```

   (*выполнить команду из директории, где находится дамп базы данных*)

   ```
   sudo docker exec app-db bash -c "psql -U backend backend < db_dump.sql"
   ```

   ```
   sudo systemctl start appscreener-app.service
   ```

7.7.2. Миграция сервера с Windows на Linux

Чтобы выполнить миграцию сервера Solar appScreener с Windows на Linux, необходимо:

На **Windows_host**:

1. Завершить или дождаться завершения всех активных сканирований.
2. В домашней директории открыть файл **pg_dump** в PowerShell:

   C:\appscreener\3rd-party\PostgreSQL\13\bin\pg_dump -E UTF-8 -f C:\appscreener\db_dump.sql -U backend backend

3. При запросе пароля, ввести его при подключении к БД (расположен в **Environment Variables > System variables > hibernate.connection.password**).
4. Перенести на **Linux_host** следующие файлы и папки:

- **C:\appscreener\db_dump.sql**
- **C:\appscreener\files\b**
- **C:\appscreener\files\s**

На **Linux_host**:

1. Установить систему Solar appScreener.
2. Копировать с **Windows_host** на **Linux_host** файлы и папки для миграции в удобную директорию.
3. Остановить работу веб-приложения:

```
sudo docker stop backend
```

4. Удалить БД новой установки:

```
sudo docker exec app-db dropdb -p 5432 -h localhost -U backend
--maintenance-db=postgres -f -e backend
```

5. Создать пустую БД:

```
sudo docker exec app-db createdb -p 5432 -h localhost -U backend
backend
```

6. Перейти в директорию с файлами для миграции (**db_dump.sql, /b, /s**) и провести миграцию файлов из дампа:

```
sudo cat db_dump.sql | sudo docker exec -i app-db psql -p 5432 -h
localhost -U backend -d backend
```

7. Провести копирование директорий **b** и **s**:

```
sudo cp -r s/ /opt/appscreener/app/services/backend/files
```

```
sudo cp -r b/ /opt/appscreener/app/services/backend/files
```

8. Перезапустить службу приложения **APP**:

```
sudo systemctl restart appscreener-app.service
```

7.7.3. Миграция базы данных на другой хост

Для миграции базы данных Solar appScreener на новый хост необходимо:

На хосте с модулем **APP**:

1. Завершить или дождаться завершения всех активных сканирований.
2. Остановить сервис приложения:

```
sudo systemctl stop appscreener-app
```

3. Сделать дамп базы данных:

```
sudo docker exec app-db pg_dump -U backend backend > db_dump.sql
```

4. Запомнить или записать значение поля *hibernate.connection.password* в **/opt/appscreener/app/configs/backend.env**.

На хосте, где вы хотите развернуть СУБД **PostgreSQL**:

5. Установить СУБД PostgreSQL версии не ниже 13.0.
6. Создать в БД пользователя с именем *backend* и паролем из поля *hibernate.connection.password*.
7. Создать пустую базу данных с именем *backend* от пользователя с именем *backend*.
   - *для хоста*: `sudo createdb -p 5432 -h localhost -U backend backend` (может потребоваться пользователь postgres, `sudo su - posrgres`)
   - *для Docker*: `sudo docker exec <Container-Name> createdb -p 5432 -h localhost -U backend backend`
8. Выполнить загрузку дампа БД (*db_dump.sql*) в пустую базу:
   - *для хоста*: `sudo cat db_dump.sql sudo psql -p 5432 -h localhost -U backend -d backend` (может потребоваться пользователь postgres, `sudo su - posrgres`)
   - *для Docker*: `sudo cat db_dump.sql | sudo docker exec -i <ИМЯ_КОНТЕЙНЕРА> psql -p 5432 -h localhost -U backend -d backend`
9. Сделать подключение к серверу доступным извне.

На хосте с модулем **APP**:

10. Открыть файл **/opt/appscreener/app/configs/backend.env** для редактирования.
11. Изменить **hibernate.connection.url=jdbc:postgresql:/app-db:5432/backend?ssl=falsee**, изменив имя подключения к контейнеру **app-db** на адрес сервера с развёрнутой БД, например: **hibernate.connection.url=jdbc:postgresql:/####.###.#.####:####/backend?ssl=falsee**.
12. Сохранить изменения и запустить сервис приложения:

    ```
    sudo systemctl start appscreener-app
    ```

8. ПОЛУЧЕНИЕ ТЕХНИЧЕСКОЙ ПОДДЕРЖКИ

Для получения консультации по техническим вопросам можно обратиться по адресу support.appscreener@rt-solar.ru.