DerScanner

Руководство системного администратора

DerScanner
Версия 11.2

Administrator Guide

Содержание

1 Введение 4
1.1 Описание возможностей 4
1.2 Архитектура 5
2 Общие требования 7
2.1 Требования к аппаратному обеспечению 7
2.1.1 Серверная часть для ОС Linux 7
2.1.2 Серверная часть для ОС Windows 9
2.1.3 Клиентская часть 10
2.2 Требования к программному обеспечению 10
2.2.1 Серверная часть для ОС Linux 10
2.2.2 Серверная часть для ОС Windows 11
2.2.3 Клиентская часть 11
3 Установка и обновление DerScanner с ОС Windows 12
4 Установка и обновление DerScanner с ОС Linux 13
4.1 Порядок установки 13
4.2 Инструкция по установке системы 14
4.3 Установка агента для анализа C/C++ и Objective-C проектов 15
4.3.1 Установка агента для macOS 15
4.3.2 Установка агента для для Windows 16
4.4 Установка Builder 17
5 Описание работы с DerScanner 19
5.0.1 Администрирование системы 19
5.0.2 Агенты сканирования 22
5.0.3 Общие настройки 23
5.0.4 Данные и память 25
5.0.5 Базы правил и уязвимостей 25
5.0.6 Подключение LDAP 27
5.0.7 Конфигурация Git 29
5.0.8 Лицензия 30

5.1 Шаблоны настроек сканирования 31

6 Расширенные настройки DerScanner 33
6.1 Смена языка компонента динамического анализа 33
6.2 Подключение встроенного почтового сервера 33
6.3 Изменение конфигурации компонента SAST с ОС Linux 34
6.4 Изменение количества потоков и памяти для анализа 35
6.5 Конфигурация агента сборки 39
6.6 Настройка работы агента сборки по HTTPS 39
6.7 Подключение приватных Git-репозиториев для установки с ОС Windows 40
6.7.1 Порядок подключения загрузки из приватных Git-репозиториев по SSH-ссылке 40
6.7.2 Порядок подключения загрузки из приватных Git-репозиториев по HTTPS-ссылке 41
6.8 Подключение приватных Git-репозиториев для установки с ОС Linux 42
6.9 Настройка сервиса NGINX 42
6.9.1 Установка HTTPS соединения 42
6.9.2 Настройка таймаута для соединения с сервером 43
6.9.3 Изменение размера клиентского запроса 43
6.9.4 Пользовательские настройки конфигурации NGINX 44
6.10 Установка HTTPS соединения (Windows) 45
6.11 Добавление самоподписных и корневых сертификатов в доверенные для работы через HTTPS и LDAPS для установки с ОС Windows 47
6.12 Добавление самоподписных и корневых сертификатов в доверенные для работы через HTTPS и LDAPS для установки с ОС Linux 48
6.13 Увеличение памяти для сервиса Tomcat 49
6.14 Что делать при ошибках в сканировании? 50
6.15 Система регистрации событий 50
6.15.1 Журналы событий 51
6.15.2 Отправка журналов событий ИБ по syslog 53
6.15.3 Log-bringer 53
6.15.4 Основные логируемые события 57
6.16 Резервирование данных 66
6.17 Миграция данных DerScanner 66
6.17.1 Миграция сервера на другой хост 66
6.17.2 Миграция сервера с Windows на Linux 67
6.17.3 Миграция базы данных на другой хост 68

6.18 Настройка интеграции с SAP-системами 69
6.18.1 Общая инструкция 69
6.18.2 Интеграция с SAP со стороны DerScanner 71
6.18.3 Пример практической реализации со стороны SAP 72

7 Получение технической поддержки 75

8 Список компонент анализа и поддерживаемых ими языков 76

9 Перечень сокращений 77

01 Введение

1.1 Описание возможностей

DerScanner — комплексное решение для анализа безопасности приложений на всех этапах жизненного цикла разработки. DerScanner поддерживает статический анализ кода с возможностью применения искуственного интеллекта для триажа и генерации рекомендаций по исправлениюю уязвимостей, оценку качества кода, динамический анализ веб-приложений, анализ состава программного обеспечения и выявление атак на цепочку поставок (Supply Chain атаки).

С помощью инструмента командной строки DerScanner взаимодействует с системами непрерывной интеграции (Continuous Integration), позволяя наладить непрерывный процесс контроля качества исходного кода и снижая временные затраты. Кроме того, DerScanner автоматизирует проверку новых сборок ПО и может быть встроен в процесс безопасной разработки (SSDLC).

Статический анализ в DerScanner предоставляет следующие возможности:

- Анализ кода на языках ABAP, Apex, ASP.NET, Bash, C#, C/C++, COBOL, Dart, Delphi, GO, Groovy, HTML, Java, Java для Android, JavaScript, JSP, Kotlin, LotusScript, Objective-C, Pascal, PHP, PL/SQL, PowerShell, Python, Perl, Ruby, Rust, Scala, Solidity, Swift, T-SQL, TypeScript, VB 6.0, VB.NET, VBA, VBScript, Visual Basic, Vyper, XML, YAML, TOML, 1C. Возможна загрузка исходных или бинарных файлов, анализ кода из архивов или по ссылке на репозиторий Git или Subversion.
- Анализ конфигурационных файлов.
- Анализ приложений для ОС Android (`apk`-файлов) и iOS (`ipa`-файлов).
- Генерация рекомендаций по исправлению уязвимостей (DerCodeFix).
- Триаж уязвимостей с применением искуственного интеллекта (DerTriage).

Вместе со статическим анализом кода DerScanner может выполнить **анализ качества кода** для проектов на JavaScript, TypeScript, Delphi и Pascal. Результаты включают оценки качества кода, которые помогают выявлять потенциальные проблемы и улучшить надежность кода.

Динамический анализ обеспечивает тестирование методом “чёрного ящика”, что позволяет обнаруживать уязвимости и закладки в веб-приложениях в момент их выполнения без доступа к исходному коду. Если исходный код приложения доступен, результаты динамического анализа могут быть сопоставлены с результатами статического анализа.

Компонентный анализ позволяет сканировать библиотеки с открытым исходным кодом на наличие уязвимостей, рисков цепочки поставок и лицензионных рисков. Функциональность доступна для проектов на C/C++, C#, Dart, Erlang, GO, Java, JavaScript, Kotlin, Objective-C, PHP, Python, Ruby, Rust, Scala, Swift, TypeScript, VB.NET. DerScanner использует базы данных NVD, GitHub, GitLab, OSV, а также собственные базы данных для обнаружения уязвимостей.

Для проектов, написанных на JavaScript/TypeScript, Python, C# или Java в рамках анализа состава ПО в DerScanner доступна опция **Гибридного анализа SCA + SAST** (анализа достижимости). Гибрийдный анализ отслеживает выполнение уязвимых функций библиотеки

в коде приложения.

Для всех типов анализа DerScanner позволяет просматривать и управлять результатами сканирований, генерировать отчёты.

1.2 Архитектура

В DerScanner функции анализа и работу пользовательского интерфейса отвечают отдельные программные компоненты. Это обеспечивает высокую производительность и стабильность системы.

Компоненты DerScanner:

- Web-serever — веб приложение, развернутое на сервере Apache Tomcat, которое включает в себя:
  - UI — пользовательский интерфейс приложения;
  - API — интерфейс взаимодействия с другими приложениям;
  - Backend — серверная часть приложения, которая взаимодействует с БД, взаимодействует с броккером сообщений для распределения задач анализа и получения результатов, формирует отчетность по результатам анализа;
- Relational Database Management System (RDBMS) — система усправления базой данных;
- Artemis MQ (MQ) — броккер сообщений, который реалирует очередь анализа и Daemon и серверную часть веб-приложения;
- SAST Daemon, DAST Daemon, SCA Daemon — компоненты, которые общаются с Artemis MQ для получения задач на сканирование.
- Компоненты анализа SAST:
  - Компонент для анализа исходного кода на языках ABAP, Apex, C#, COBOL, Dart, Delphi, GO, Groovy, HTML, JavaScript, LotusScript, Pascal, PHP, PL/SQL, Python, Perl, Ruby, Rust, Solidity, Swift, T-SQL, TypeScript, VB.NET, VBA, VBScript, Visual Basic, Vyper и 1C;
  - Компонент для анализа бинарных файлов Java, Scala, Kotlin и приложений `.apk`;
  - Компонент для анализа исходного кода на языках C, C++, Objective-C;
  - Компонент для анализа `.ipa`-файлов;
  - Компонент для анализа бинарных файлов `.exe` и `.dll`.
- Компонент анализа DAST;
- Компонент анализа SCA.

Интеграции DerScanner взаимодействуют с системой через API.

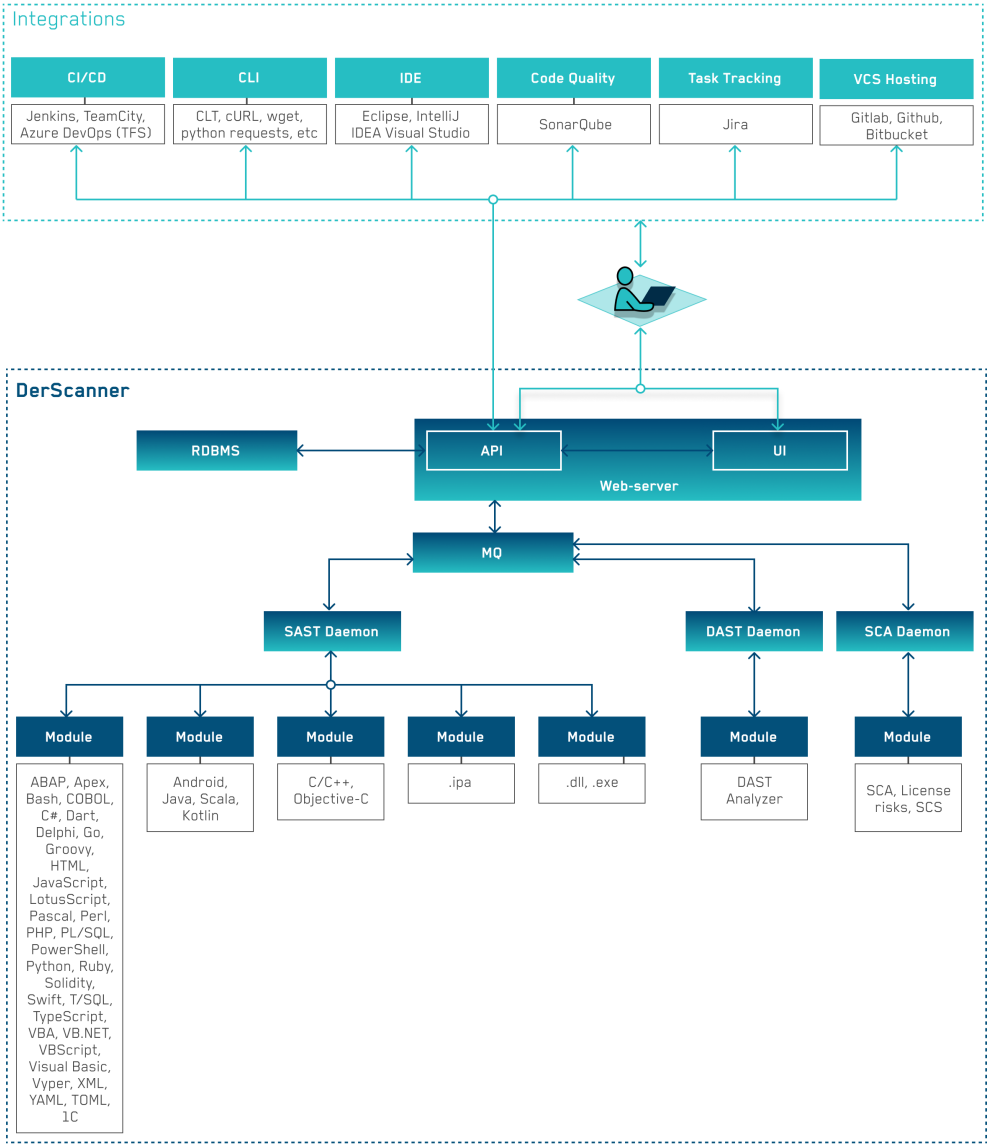


Рис. 1.1: Архитектура DerScanner

02 Общие требования

2.1 Требования к аппаратному обеспечению

2.1.1 Серверная часть для ОС Linux

Система поставляется в виде программных компонент. Для корректной работы требуется последовательно установить необходимый набор компонент:

- компонент APP - обязательный компонент, веб приложение, отвечает за логику работы всей системы;
- компонент SAST - статический анализатор;
- компонент DAST - динамический анализатор веб-приложений;
- компонент SCA - анализатор состава программного обеспечения;
- Агенты C/C++/Objective-C - статические анализаторы для проектов на C/C++/Objective-C для Windows или macOS;
- Builder - обеспечивает сборку проектов, написанных на Java, Scala, Kotlin, в пользовательской среде и передачу собранных проектов на анализ.
- DerAI - компонент для работы с ИИ-инструментами генерации рекомендаций по исправлению и триажу уязвимостей, найденных статическим анализом кода (к DerScanner может быть подключен компонент DerAI из облака или компонент, который развернут локально).

Компоненты можно установить на один сервер или на несколько серверов, связанных в одну сеть.

Компоненты APP, SAST, DAST, SCA поставляются с собственным окружением (представленным в виде системы контейнеризации). Агенты C/C++/Objective-C поставляются с окружением jdk-11 для Windows или macOS. Builder поставляется с окружением jdk-11 для Linux.

2.1.1.1 Минимальные характеристики оборудования установки на одном сервере

Для функционирования DerScanner на одном сервере, требуется оборудование со следующими минимальными характеристиками:

APP + SAST

- 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).

APP + SAST + DAST (следует увеличить объём преративной памяти на 8 ГБ, CPU на 2 ядра, объем диска на 200 ГБ)

- 10 ядерный процессор с тактовой частотой 2.2 ГГц;
- объём оперативной памяти – 40 ГБ;

- минимальный объём жёсткого диска – 512 ГБ SSD/ SAS HDD (при увеличении количества сканирований требуемый объём может увеличиться);
- поддерживаемые операционные системы.

APP + SAST + SCA (следует увеличить объём преративной памяти на 16 ГБ, CPU на 2 ядра, объем диска на 100 ГБ)

- 10 ядерный процессор с тактовой частотой 2.2 ГГц;
- объём оперативной памяти – 48 ГБ;
- минимальный объём жёсткого диска – 512 ГБ SSD/ SAS HDD (при увеличении количества сканирований требуемый объём может увеличиться);
- поддерживаемые операционные системы.

APP + SAST + DAST + SCA

- 12 ядерный процессор с тактовой частотой 2.2 ГГц;
- объём оперативной памяти – 64 ГБ;
- минимальный объём жёсткого диска – 512 ГБ SSD/ SAS HDD (при увеличении количества сканирований требуемый объём может увеличиться);
- поддерживаемые операционные системы.

В зависимости от количества сканирований, объёма кода для подсистемы статического анализа (SAST), сложности сканируемого ресурса для динамического анализа (DAST) и наличию или отсутствию SBOM файлов для анализа состава программного обеспечения SCA минимальные характеристики могут возрасти.

2.1.1.2 Минимальные характеристики оборудования установки на отдельных серверах

Для установки DerScanner на нескольких (отдельных) узлах, компонент **APP** устанавливается один раз и не реплицируется. Для развёртывания компонент на отдельных серверах для разных подсистем требуются следующие минимальные характеристики:

APP

- 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).

SAST

- 4 ядерный процессор с тактовой частотой 2.2 ГГц;
- объём оперативной памяти – 32 ГБ;
- минимальный объём жёсткого диска – 128 ГБ SSD/ SAS HDD;
- поддерживаемые операционные системы.

DAST

- 4 ядерный процессор с тактовой частотой 2.2 ГГц;
- объём оперативной памяти – 8 ГБ;
- минимальный объём жёсткого диска – 128 ГБ SSD/ SAS HDD;
- поддерживаемые операционные системы.

SCA

- 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.

2.1.1.3 Требования к конфигурации объема дискового пространства

Корректная работа дистрибутива требует соблюдения одного из следующих условий при разбиении диска:

- объем всего диска выделяется на корневую директорию **(/)**;
- при специфическом дисковом разбиении на директорию **/opt** необходимо выделить не менее 90% от требуемого дискового пространства для каждого компонента.

В ходе эксплуатации DerScanner в зависимости от использования, через некоторое время может потребоваться расширение дискового пространства.

2.1.2 Серверная часть для ОС Windows

Система поставляется отдельными программными компонентами. Для корректной работы требуется последовательно установить необходимый набор компонент. Компоненты можно установить на один сервер или на несколько серверов, связанных в одну сеть. Ссылки на скачивание компонент содержатся в инструкции по установке DerScanner.

- APP - обязательный компонент, веб приложение, отвечает за логику работы всей системы;
- SAST - статический анализатор.

2.1.2.1 Минимальные характеристики оборудования для установки на одном сервере

Для функционирования DerScanner с компонентом статического анализа **APP + SAST** 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 объёма кода для подсистемы статического анализа (SAST) минимальные характеристики могут возрасти.

2.1.2.2 Минимальные характеристики оборудования установки на отдельных серверах

Для развёртывания компонент на отдельных серверах для разных подсистем требуются следующие минимальные характеристики:

APP

- 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).

SASTь

- 4 ядерный процессор с тактовой частотой 2.2 ГГц;
- объём оперативной памяти – 32 ГБ;
- минимальный объём жёсткого диска – 128 ГБ SSD/ SAS HDD;
- поддерживаемые операционные системы.

2.1.3 Клиентская часть

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

2.2 Требования к программному обеспечению

2.2.1 Серверная часть для ОС Linux

Для функционирования DerScanner 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.

2.2.2 Серверная часть для ОС Windows

Для функционирования DerScanner на серверном оборудовании должна быть установлена одна из операционных систем: - Microsoft Windows Server 2019, 2022 x64; - Microsoft Windows 10 x64, 11 x64.

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

- чистая операционная система без предустановленного стороннего ПО;
- отсутствие установленных сторонних баз данных и Java-машин;
- привилегии пользователя administrator;
- свободные TCP-порты: 80, 443, 8005, 8009, 8080, 5432, 61616.

2.2.3 Клиентская часть

Чтобы начать работу с DerScanner, потребуется компьютер с доступом к внутренней корпоративной сети, где установлен DerScanner, и один из браузеров Mozilla Firefox, Google Chrome, Safari или Microsoft Edge. В настройках браузера необходимо рразрешить выполнение JavaScript и сохранение файлов cookies.

03 Установка и обновление DerScanner с ОС Windows

1. Загрузите установочный файл по ссылке из инструкции по установке.
2. Запустите установочный файл от имени администратора и внимательно следуйте инструкциям.
3. Перейдите в браузере по адресу http://localhost или, при доступности машины по сети извне, по http://installation_address с другого компьютера сети.
4. Во всплывающем окне с предложением загрузить лицензию скопируйте идентификатор установки и отправьте его по почте для создания лицензии под конкретную инсталляцию.
5. Загрузите полученную лицензию через интерфейс.
6. Войдите в систему:
   - логин: **admin**
   - пароль: ***put_admin_password_here***
7. После чего смените пароль в разделе **Аккаунт**.
8. Загрузите правила на вкладке **Настройки** > **Система** > **Правила**.
9. Перейдите в раздел **Домашняя страница** и начните пользоваться системой.

Плагины для интеграций с Teamcity, Jenkins, Azure DevOps Server, IntelliJ IDEA, Visual Studio, SonarQube, а также CLT (command line tool) расположены в %DERSCANNER_HOME%\plugins.

04 Установка и обновление DerScanner с ОС Linux

4.1 Порядок установки

Количество серверов и работающих на них компонент можно скорректировать после установки, добавив или удалив необходимые элементы (сервера, компоненты) в систему.

Компонент **APP** следует устанавливать первым. Этот компонент отвечает за веб-интерфейс ПО, логику работы всей системы и взаимодействует с опциональными компонентами (SAST, DAST, SCA). APP компонент может быть развёрнут только на одном сервере в единственном экземпляре, т.к. к нему привязывается лицензия, и он не может быть реплицирован.

Компонент **SAST** (опциональный), может быть установлен совместно с другими компонентами или вынесен на отдельный сервер. Может быть развёрнут на нескольких серверах одновременно (увеличивает пропускную способность установки по статическому сканированию).

Компонент **DAST** (опциональный), может быть установлен совместно с другими компонентами или вынесен на отдельный сервер. Может быть развёрнут на нескольких серверах одновременно (увеличивает пропускную способность установки по динамическому сканированию).

Компонент **SCA** (опциональный), может быть установлен совместно с другими компонентами или вынесен на отдельный сервер. Может быть развёрнут на нескольких серверах одновременно (увеличивает пропускную способность установки по сканированию).

Дополнительные агенты для компонента SAST следует устанавливать, если есть необходимость проведения статического анализа платформо-зависимых проектов на языках C/C++, Objective-C. Если требуется анализ кода, написанного на C/C++, Objective-C, под платформу **Windows x86-64** (Инструкция по установке компонента для анализа C/C++ и Objective-C проектов для Windows) или **macOS** (Инструкция по установке компонента для анализа C/C++ и Objective-C проектов для macOS).

Дополнительный компнент Builder (для SAST) предоставляет возможность сборки проектов на языках Java, Scala, Kotlin в определённо настроенном окружении или с использованием определенных команд. Компонент необходимо разворачивать в том случае, если есть необходимость анализа такх проектов и на одном сервере вместе со сборщиком (Инструкция по установке компонента Builder).

Дополнительный компонент DerAI (для SAST), который обеспечивает работу ИИ инструментов (DerCodeFix и DerTriage).

4.2 Инструкция по установке системы

Первым устанавливается компонент веб-приложения — APP. Далее устанавливаются опциональные компоненты (SAST, DAST, SCA, Builder, DerAI), порядок их установки не важен.

Установка APPя

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

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

   и следовать инструкциям.

Установка опциональных компонент - SAST, DAST, SCA

1. Скачать и распаковать архив (архивы).
2. Открыть в терминале директорию распакованного архива.
3. Выполнить команду в зависимости от устанавливаемого:

   - SAST: `sudo bash actions.sh | sudo tee -a /tmp/derscanner_SAST.log`
   - DAST: `sudo bash actions.sh | sudo tee -a /tmp/derscanner_DAST.log`
   - SCA: `sudo bash actions.sh | sudo tee -a /tmp/derscanner_OSA.log`

   и следовать инструкциям.
4. При установке компонента на сервер, отличный от сервера компонента APP, в файле /opt/derscanner/core/sast/configs/sast-daemon.env заменить значение поля **queueURI** на адрес сервера с установленным компонентом APP, например: queueURI=tcp://10.10.10.10:61616 После чего перезапустить службу компонента командой (для примера используется команда для компонента SAST):

   ```
   sudo systemctl restart derscanner-sast.service
   ```

Дальнейшая установка и настройка системы

1. На сервере с развёрнутым компонентом APP 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: ***admin_password***.
5. Сменить пароль учётной записи в **Аккаунт > Доступы > Токен и пароль**.
6. Выбрать и загрузить файл Rules.zip на вкладке **Настройки** > **Система** > **Правила**.
7. В разделе **Настройки** > **Система** > **Общие** указать IP-адрес/доменное имя развернутоного локально или облачного компонента **DerAI**.

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 компонентом APP - **/opt/derscanner/app/plugins**.

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

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

4.3 Установка агента для анализа C/C++ и Objective-C проектов

4.3.1 Установка агента для macOS

Установка агента является опциональной и необходима только в случае, если для требуется проводить статический анализ проектов, написанных на языках C/C++, Objective-C, для платформы macOS. Ссылка на скачивание компонента содержится в инструкции по установке DerScanner.

Требования к системе:

- macOS версии mojave и выше;
- 4-cores CPU;
- 16 ГБ RAM;
- 100 ГБ SSD/SAS HDD;
- права root;
- отсутствие установленных Java-машин и переменных среды $JAVA_HOME;
- сетевое соединение с основным сервером установки;
- наличие установленных Xcode, clang, CMake, cocoapods;
- доступ к сети интернет;
- наличие установленных Git и/или Subversion, при необходимости загрузки библиотек из внешних источников;
- подготовленное окружение, в котором успешно собирается анализируемый проект.

Обратите внимание: Данный компонент необходимо устанавливать на машину, которая обеспечивает полную автоматическую сборку анализируемого проекта. Поддерживаются сборки проектов с использованием Xcode и GNU CMake.

4.3.1.1 Порядок установки

1. Скачайте архив.
2. Распакуйте содержимое в желаемую директорию, например **/Users/user/derscanner/**.
3. Установите JDK из архива, выбирая всё по умолчанию.
4. В терминале перейдите в директорию с распакованным архивом, выполните `iccheck: bash iccheck.sh` и следуйте инструкциям.
5. Измените настройки **daemon.properties**, проверьте доступность значений всех ключей.
6. Убедитесь в правильности путей и измените их при необходимости:
   - **queueURI** адрес установки основного сервера. Пример: **tcp://88.88.88.88:61616**

- **workDir** путь до директории временных файлов анализатора. Пример: /Users/user/aderscanner/files/d/
- **CFamilyNix.module** путь до исполняемого файла анализатора. Пример: /Users/user/derscanner/iccheck/iccheck-make
- **CFamily.iccheckxx** путь до исполняемого файла анализатора. Пример: /Users/user/derscanner/iccheck/iccheck++
- **CFamily.iccheck** путь до исполняемого файла анализатора. Пример: /Users/user/derscanner/iccheck/iccheck
- **CFamily.rules** путь до правил анализатора. Пример: /Users/user/derscanner/iccheck/share/iccheck/assets/queries/

7. Запустите из терминала: `java -jar daemon.jar`.

Внутри архива приложен пример **plist** для установки службы с помощью гипервизора процессов **launchd**, в котором необходимо заменить значения на корректные.

4.3.1.2 Порядок обновления

1. Убедитесь в отсутствии активных задач у анализатора и завершите его процесс.
2. Сохраните копию конфигурационного файла **daemon.properties**.
3. Переименуйте директорию с текущей версией анализатора для создания резервной копии.
4. Скачайте архив.
5. Распакуйте содержимое в необходимую директорию, так чтобы имя совпадало с предыдущей версией, например **/Users/user/derscanner/**.
6. В терминале перейдите в директорию с распакованным архивом и выполните скрипт `iccheck: bash iccheck.sh` и следуйте инструкциям.
7. В случае полного совпадения ключей в старой и новой конфигурациях, скопируйте сохраненную копию в директорию к **daemon.jar**, чтобы заменить **daemon.properties**.
8. В случае различия конфигураций, измените настройки **daemon.properties** на корректные. Проверьте правильность значений всех ключей.
9. Запустите из терминала: `java -jar daemon.jar`.

Внутри архива приложен пример **plist** для установки службы с помощью гипервизора процессов **launchd**, в котором необходимо заменить значения на корректные.

4.3.2 Установка агента для для Windows

Установка производится опционально, если для статического анализа требуется анализ проектов, написанных на C/C++, Objective-C, для платформы Windows (Visual Studio). Ссылка на скачивание компонента содержится в инструкции по установке DerScanner.

Требования к системе.

- ОС Windows 10 x64/Windows 11 x64/Windows Server 2019 x64/Windows Server 2022 x64;
- 4-cores CPU 2.2GHz;
- 16 GB RAM;

- 100 GB HDD;
- права администратора;
- наличие установленных Visual Studio или Microsoft Build Tools версии, на которой происходит сборка проекта;
- отсутствие установленных Java-машин и переменных среды $JAVA_HOME;
- сетевое соединение с сервером с установленным компонентом APP;
- подготовленное окружение, в котором успешно собирается анализируемый проект.

Обратите внимание: данный компонент необходимо устанавливать на машину, на которой возможна полная автоматическая сборка анализируемого проекта.

4.3.2.1 Порядок установки и обновления

1. Проверить сетевую доступность порта 61616 сервера с установленным компонентом APP.
2. Запустить установочный файл от имени администратора и внимательно следовать инструкциям.

Обратите внимание:

- Анализатор управляется службой derscanner-daemon;
- Ключ daemonThreads в %DERSCANNER_HOME%\daemon\daemon.properties регулирует количество одновременных задач для анализа. Изменять его необходимо через добавление новой системной переменной среды **daemonThreads** со значением желаемого количества потоков, после чего перезапустить сервис анализатора **derscanner-daemon**;
- При запуске анализа C/C++ проектов из веб-интерфейса системы в дополнительных настройках выберите опцию **Windows**;
- Поддерживается анализ со сборкой для проектов Visual Studio и GNU CMake;
- Переменная %DERSCANNER_HOME% — путь до директории DerScanner.s

4.4 Установка Builder

Установка производится опционально, если сборки в проектов на Java, Scala и Kotlin в процессе анализа, требуется определённым образом настроенная среда или пользовательские команды. Может быть реплицирован на нескольких узлах.

Требования к системе.

- привилегии root-пользователя;
- свободный TCP порт: 8080 (для установки по умолчанию);
- сетевая доступность хоста с компонентом APP;
- окружение, в котором успешно собирается Java/Scala/Kotlin проект.

Обратите внимание:

- перед установкой компонента APP необходимо обновить до последней версии;
- после установки компонента Builder сервис будет работать по протоколу HTTP;
- информация по настройке Builder агентов и изменение протокола на HTTPS описана в Настройка работы агента сборки по HTTPS.

Порядок установки и обновления:

1. Скачать и распаковать архив.

2. Открыть в терминале директорию распакованного архива.
3. Выполнить команду:

   ```
   sudo bash actions.sh
   ```

   и следовать инструкциям.
4. Изменить учетную запись агента: имя пользователя и пароль для подключения к агенту сборки (см. Конфигурация агента сборки).
5. Подключить агент в UI (см. Агенты сборки).

Учетные данные агента по умолчанию и порт:

логин: builder

пароль: OhFe9iewaeN0wu5e

порт: 8080

За работу агента на хосте сборки отвечает служба **derscanner-builder.service**. Проверить статус службы можно, выполнив команду:

```
sudo systemctl status derscanner-builder.service
```

Рабочая директория — **/opt/derscanner/builder**.

Процесс запущен от имени пользователя **derscanner**, который создается при запуске скрипта установки агента Builder.

05 Описание работы с DerScanner

5.0.1 Администрирование системы

5.0.1.1 Создание учётных записей

Для создания учетной запиши пользователя DerScanner:

1. Откройте раздел **Пользователи** и нажмите на **Создать пользователя**.
2. Введите логин, пароль и ФИО, e-mail, название организации, веб-сайт организации, должность и телефон (опционально).
3. 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.
4. Поля **Доступно сканирований SAST/DAST/SCA** позволяют регулировать количество доступных для пользователя сканирований соответствующего типа. При отсутствии доступных сканирований пользователь также не сможет создать пустой проект.
5. Выберите общие права для пользователя или выберите одну из существующих ролей (рис. 5.1):
   - **Работать с правилами** позволяет создавать новые правила, копии правил и пользовательские паттерны;
   - **Работать с наборами правил для всех пользователей** позволяет создавать публичные (доступные другим пользователям) наборы правил. При отсутствии права пользователь сможет создавать только приватные наборы, доступные ему и администратору системы;
   - **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 SAST, DAST или SCA, 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 DerScanner;

- **Загружать приложения с локального компьютера или из репозитория** позволяет загружать проекты указанным способом;
- **Запускать сканирования DAST** позволяет запускать сканирования динамическим типом анализа;
- **Запускать сканирования SCA** позволяет запускать сканирования анализа состава ПО;
- **Запускать сканирования Supply chain** позволяет запускать сканирования цепочек поставок.

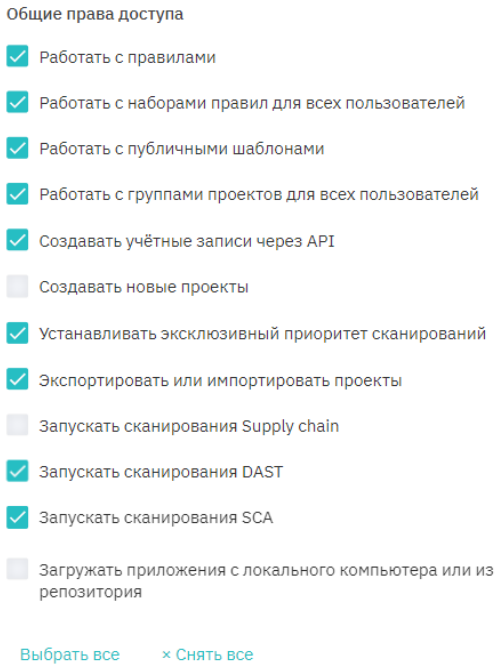


Рис. 5.1: Общие права пользователя

6. Настройте параметры, которые не будут отображаться для других пользователей в сканированиях, запущенных данным пользователем, в разделе **Ограничения видимости**.
7. При необходимости ограничьте языки программирования, для которых пользователь сможет запускать анализ, в разделе **Анализировать языки**.
8. Настройте доступ к проектам в системе.
9. Выберите права в проектах статического, динамического анализа и анализа состава ПО, или в группах проектов (рис. 5.2). 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**, **Анализировать языки**).

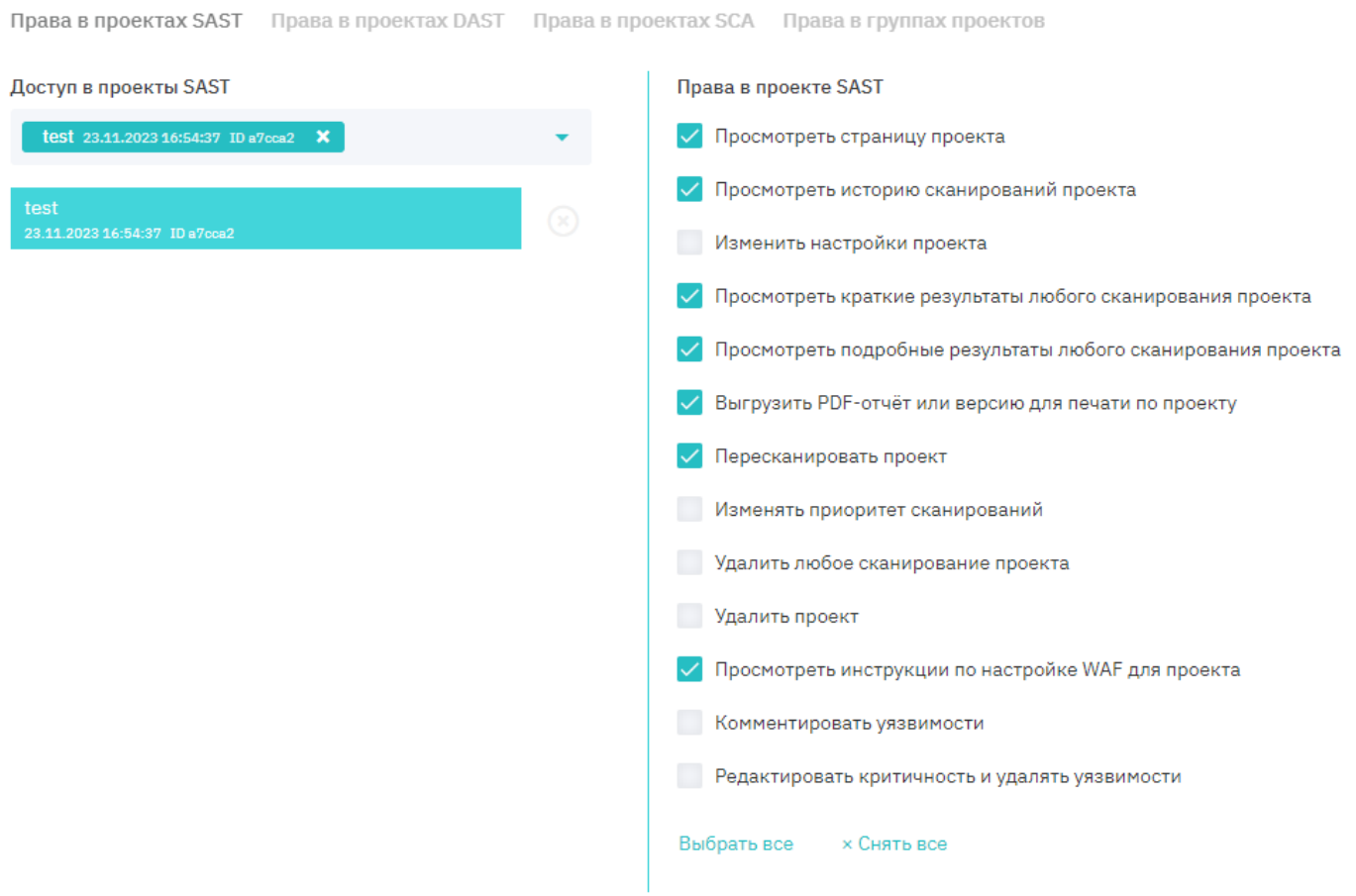


Рис. 5.2: Права пользователя в определённом проекте

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**.

5.0.1.2 Блокировка пользователя

Чтобы заблокировать учетную запись пользователя DerScanner:

1. Нажмите на логин пользователя в разделе **Пользователи**.
2. Укажите причину блокировки (опционально). 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**.

5.0.1.3 Управление группами пользователей

Пользователи могут быть объединены в группы. В итоге пользователь получит объединение личных прав и прав группы.

Чтобы создать группу пользователей:

1. Разделе **Группы пользователей** нажмите на кнопку **Создать группу**.
2. Введите имя группы и описание.
3. Выберите состав группы из списка пользователей.
4. Выберите общие права и права в проектах.
5. Нажмите **Сохранить**.

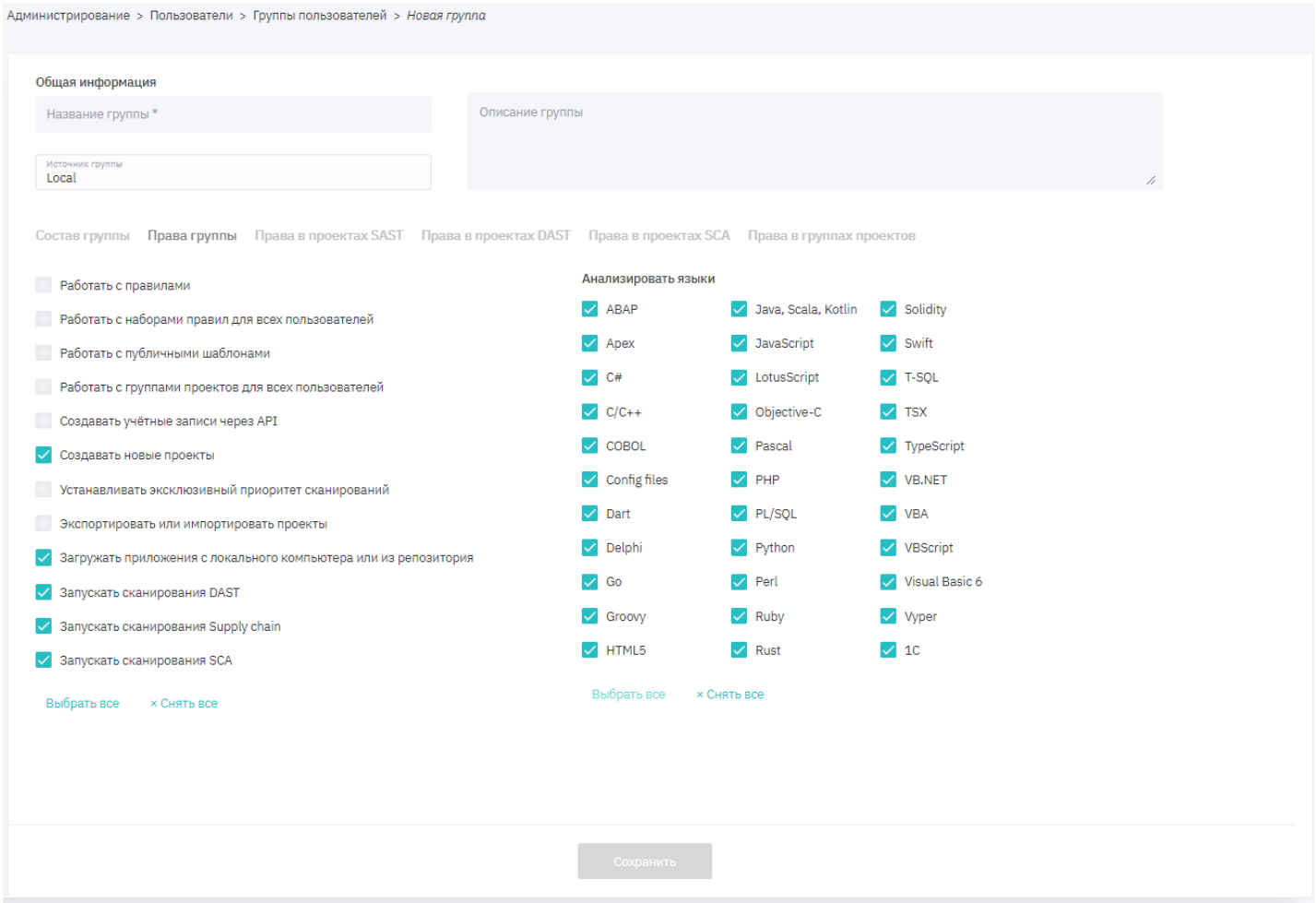


Рис. 5.3: Создание группы пользователей

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

5.0.2 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 **Система > 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.

5.0.3 Общие настройки

На вкладке **Система > Общие** можно управлять настройками DerScanner.

5.0.3.1 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.
- **Размер кода для подсветки** — в поле можно указать значение для подсветки синтаксиса для многострочных файлов на странице **Подробные результаты** проекта.
- **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 DerScanner.
- **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.
- **DerAI** — IP-адрес или доменное имя компонента **DerAI**, которое необходимо для настройки доступа к компоненту.

5.0.3.2 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.

5.0.3.3 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.

5.0.3.4 Агенты сборки

Для добавления агента (установленного компонента Builder):

1. В поле **Название агента** укажите название агента сборки.
2. В поле **Адрес** укажите адрес сервера, на котором развернут агент.
3. В поле **Порт** укажите порт агента.
4. В поле **Имя пользователя** укажите логин учётной записи, используемый для подключения к агенту сборки.
5. В поле **Пароль** укажите пароль учётной записи, используемый для подключения к агенту сборки.
6. В поле **Таймаут (минуты)** укажите таймаут для сборки в минутах. Если время сборки превысит указанное значение, анализ начнется без сборки.

7. Нажмите **Сохранить**.

Проверьте соединение с указанным агентом, нажав **Проверка соединения**.

После добавления агента вы можете использовать его для анализа ваших проектов на Java, Scala и Kotlin. Для этого:

1. В настройках сканирования (**Настройки языков** > **Java, Scala, Kotlin**) выберите опцию **Анализ исходного кода с собственной сборкой**.
2. Выберите добавленный агент из списка и укажите команду, которая будет использоваться для сборки проекта.

После начала сканирования проект будет отправлен агенту.

Обратите внимание: команды сборки, переданные через UI DerScanner, вместе с исходным кодом запускаются сервисом с правами пользователя **derscanner** в рабочей директории агента.

5.0.4 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 только исходный код** — удаление исходного кода со страницы **Подробных результатов**, на вкладке **Трасса** пропадает трасса, пропадает возможность скачать исходный проект.
- **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.

5.0.5 Базы правил и уязвимостей

5.0.5.1 Статический анализ

Для обновления **Правил** (рис. 5.4):

1. Перейдите в **Система > Правила**.
2. Выберите файл с архивом правил и нажмите **Загрузить правила**.
3. Нажмите **Сохранить**.

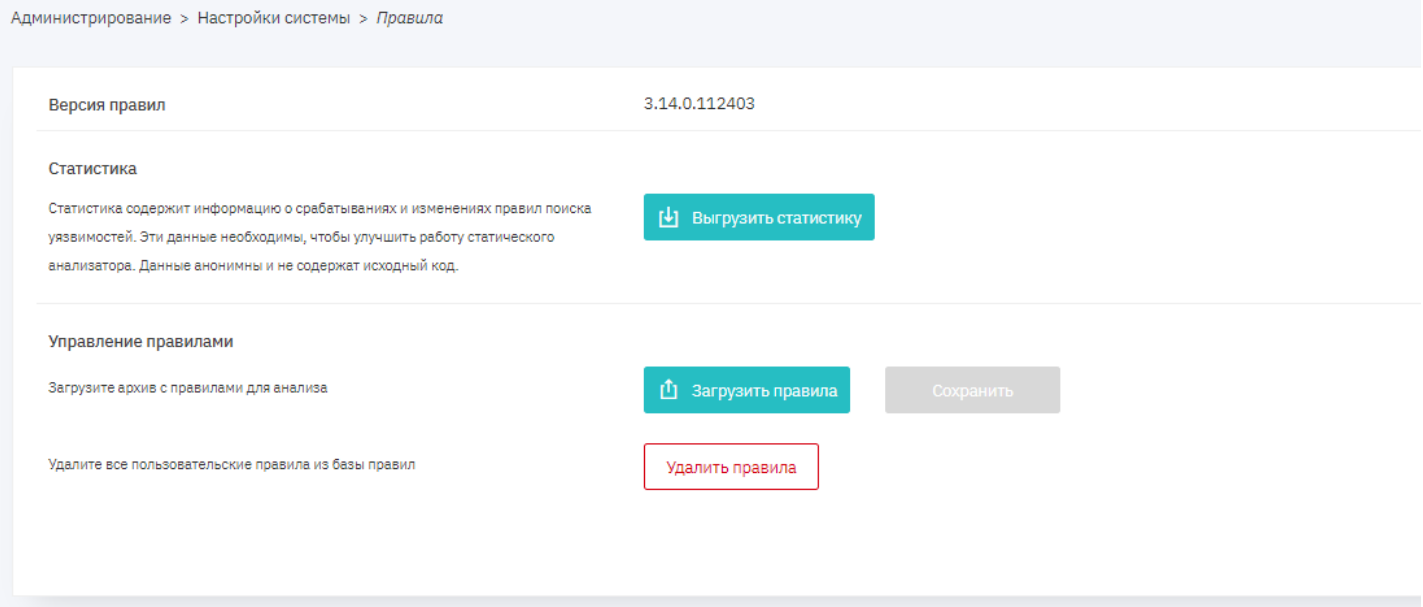


Рис. 5.4: Раздел Настройки > Система > Правила > Статический анализ

При обновлении **Правил** пользовательские правила сохраняются в системе. Если вы хотите удалить все пользовательские правила, нажмите **Удалить правила**.

5.0.5.2 Компонентный анализ

На странице можно выбрать ручное или автоматическое обновление баз SCA.

Для автоматического обновления баз **SCA**, **Лицензионных рисков** и **SCS** воспользуйтесь соответствующими переключателями.

Для ручного обновления выберите агент, для которого нужно провести обновление, и загрузите новую версию баз с локального компьютера или с облака (**Загрузить локально** или **Загрузить с облака** соответственно).

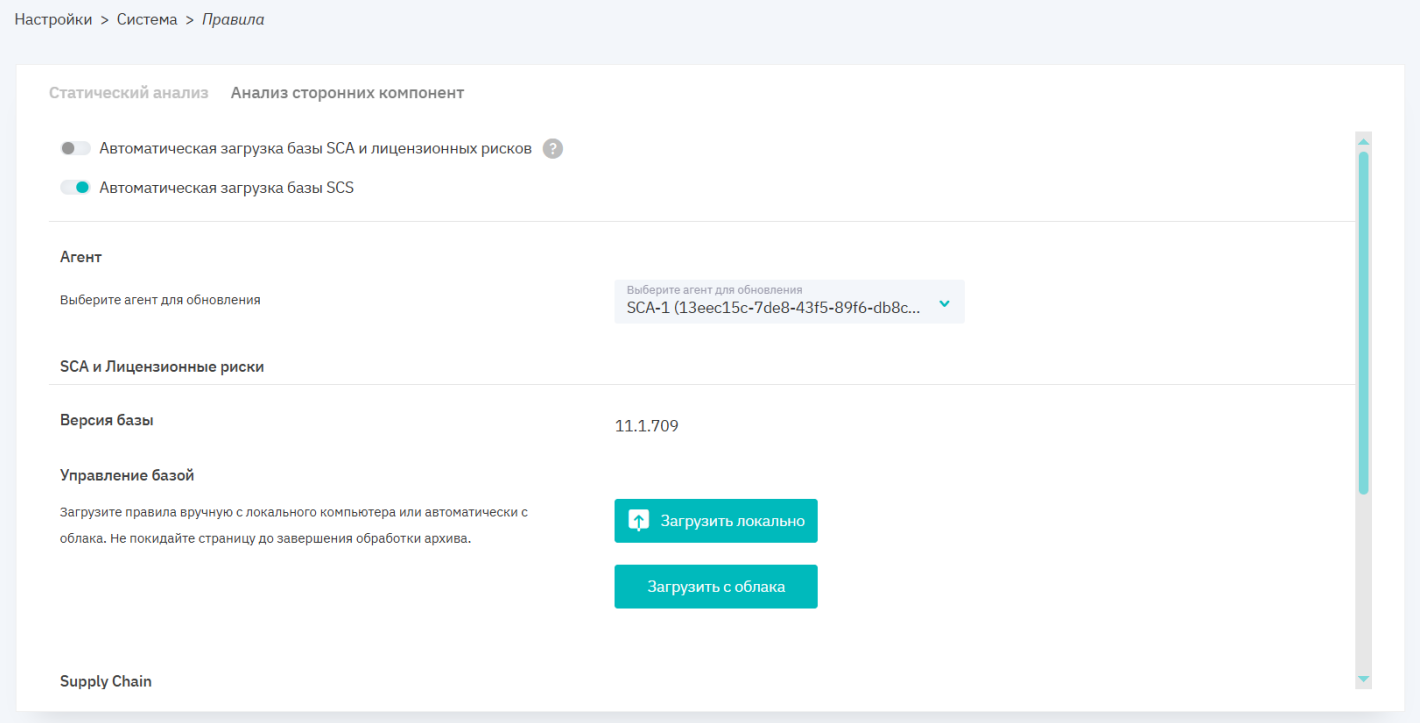


Рис. 5.5: Раздел Настройки > Система > Правила > Анализ сторонних компонент

5.0.6 Подключение LDAP

Для добавления нового подключения **LDAP** (рис. 5.6):

1. Перейдите в **Система > 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 DerScanner, в поле **Имя домена** нужно указать **test-domain**.

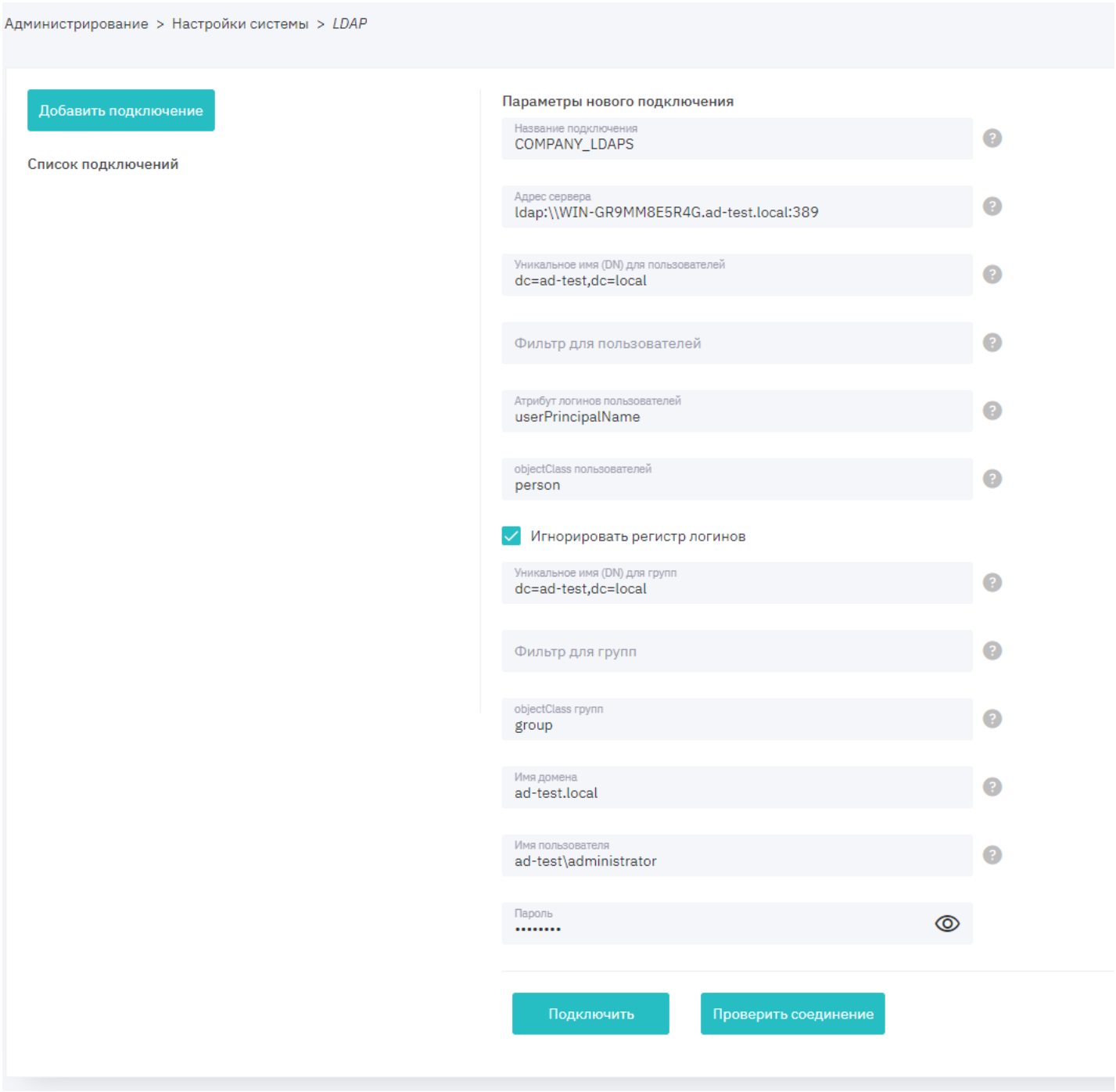


Рис. 5.6: Параметры подключения LDAP

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

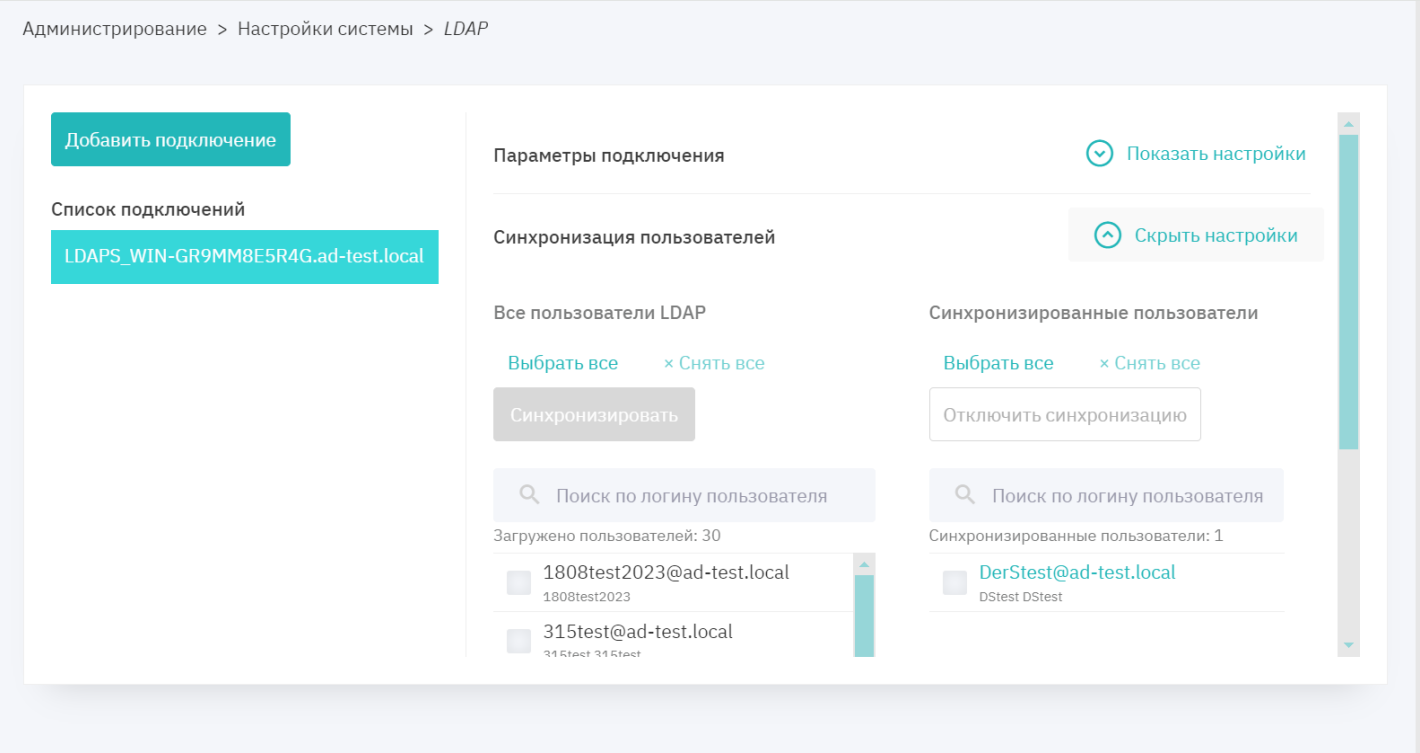


Рис. 5.7: LDAP: синхронизация пользователей

5.0.7 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
```

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
```

Если требуется валидация самоподписанных сертификатов сервисом Git, выполните:

1. Создайте самоподписанный сертификат для доменного имени (опция **Common name**).
2. В интерфейсе DerScanner (Настройки > Система > 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 компонентом APP создайте файл /opt/derscanner/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/derscanner/app/services/backend/files/certs.crt:

```
-----BEGIN CERTIFICATE-----
CERTIFICATE_SAMPLE_1
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
CERTIFICATE_SAMPLE_2
-----END CERTIFICATE-----
...
```

5.0.8 Лицензия

Для обновления **Лицензии** (рис. 5.8):

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

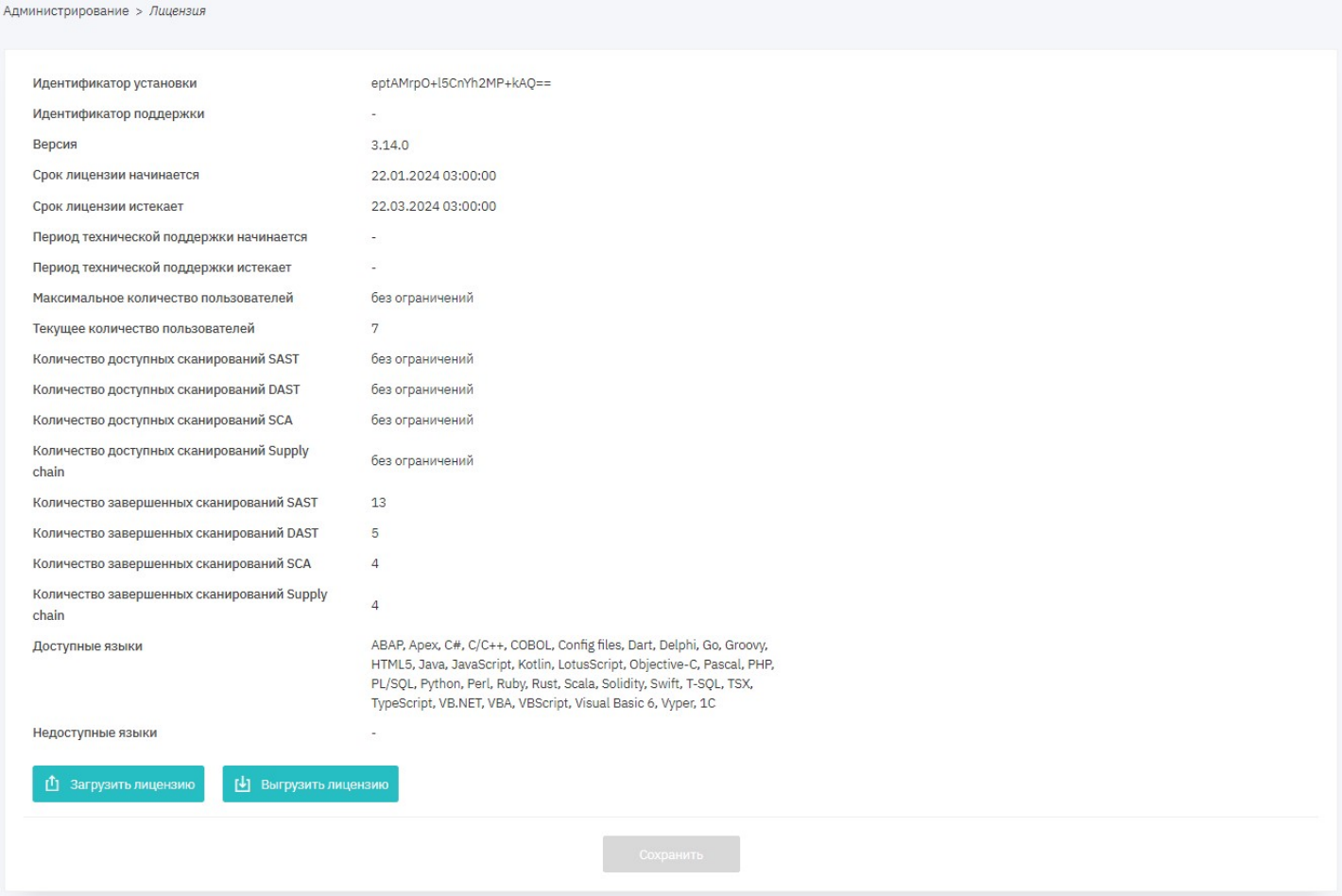


Рис. 5.8: Раздел Настройки > Лицензия

5.1 Шаблоны настроек сканирования

Шаблоны позволяют не настраивать конфигурацию сканирования вручную перед каждым запуском, а в один клик выставлять часто используемые настройки для анализа. Со списком доступных шаблонов можно ознакомиться в разделе **Аккаунт > Шаблоны > Сканирование**.

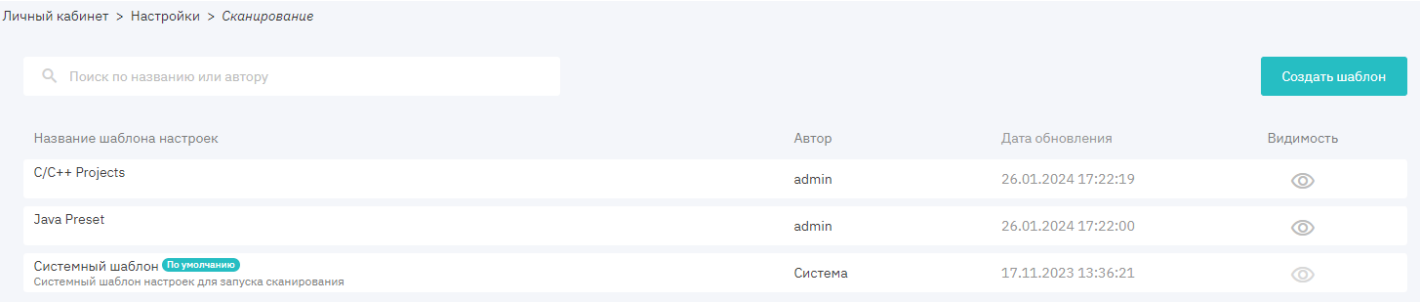


Рис. 5.9: Аккаунт > Шаблоны > Сканирование

Инструкцию по созданию/редактированию шаблонов настроек сканирования можно найти в Руководстве пользователя (раздел **Аккаунт** > **Шаблоны**).

Администратор DerScanner может назначить шаблон настроек сканирования по умолчанию

для всех пользователей системы. Чтобы сделать это, активируйте опцию **Шаблон настроек по умолчанию для всех пользователей** в форме создания/редактирования шаблона (доступно только для **публичных** шаблонов).

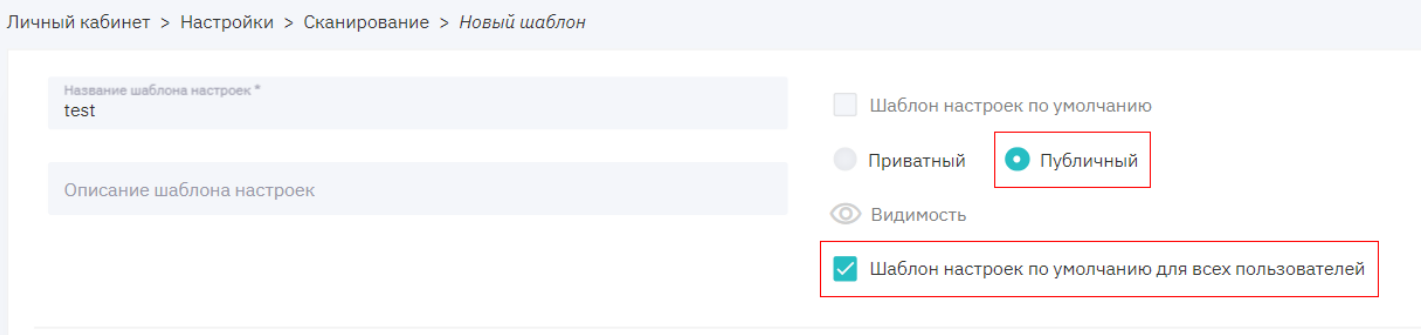


Рис. 5.10: Выбор шаблона по умолчанию

Обратите внимание: пользователи системы могут самостоятельно выбирать, изменять и удалять свое значение шаблона по умолчанию для запуска анализа. Если вы не выберете иной шаблон, то по умолчанию для запуска сканирования будет использоваться системный шаблон.

06 Расширенные настройки DerScanner

6.1 Смена языка компонента динамического анализа

Чтобы сменить язык для отображения результатов в интерфейсе:

1. Подключитесь к хосту, где развёрнут компонент **DAST**.
2. Откройте конфигурационный файл **/opt/derscanner/core/dast/configs/zap.env**.
3. Измените значение ключа ZAP_LOCALE на любое из поддерживаемых, например ru_RU.
4. Перезапустите сервис компонент **DAST**:

   ```
   sudo systemctl restart derscanner-dast.service.
   ```

6.2 Подключение встроенного почтового сервера

В дистрибутив поставки для Linux 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 компонент **APP**:

  ```
  sudo docker logs postfix
  ```

Для настройки встроенного почтового сервера:

1. Перейдите в раздел **Настройки** > **Система** > **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.

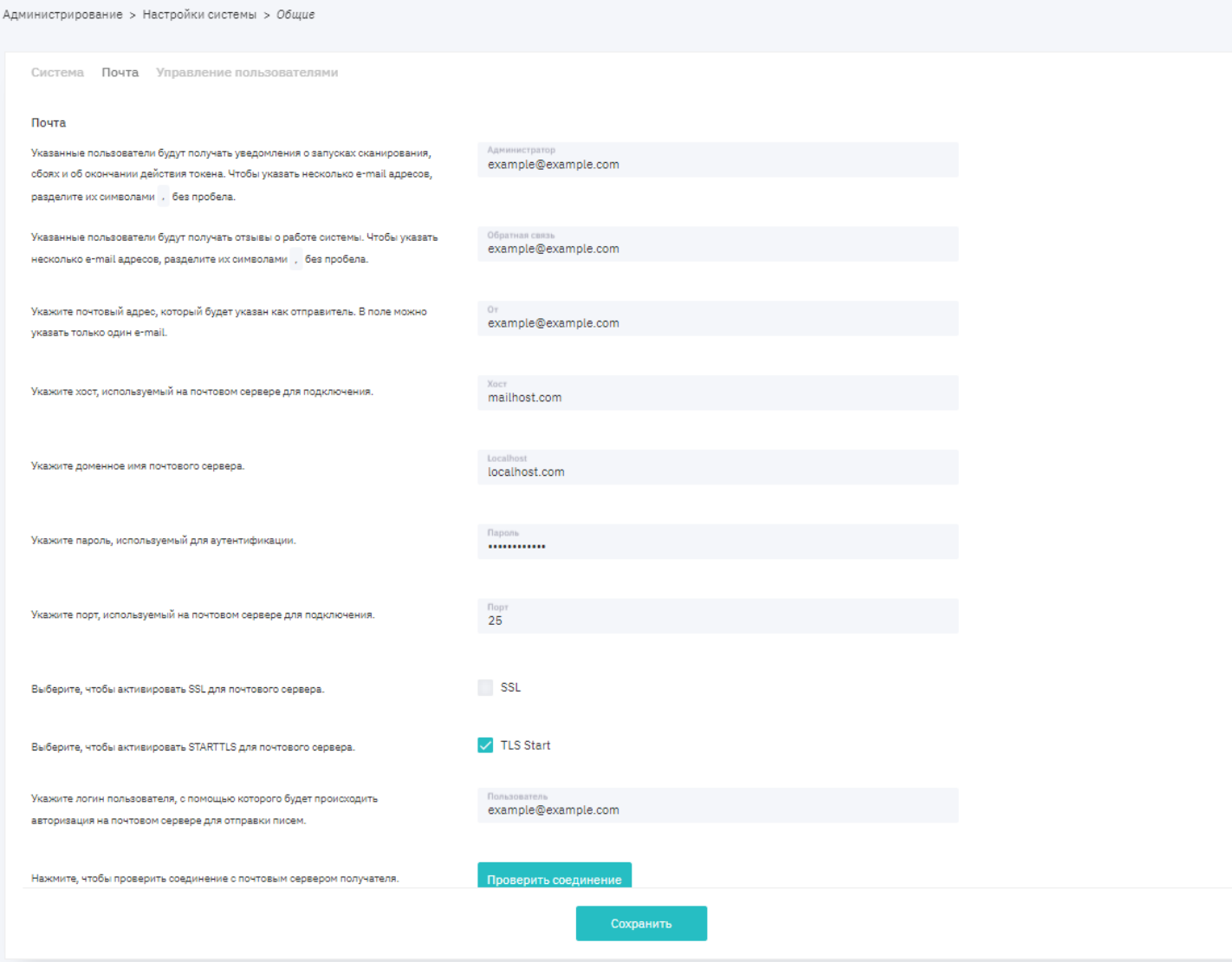


Рис. 6.1: Настройка почтового сервера

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

6.3 Изменение конфигурации компонента SAST с ОС Linux

Для компонента SAST можно изменить типы анализаторов, работающих на хосте. Список анализаторов, отвечающих за анализ определённых языков/файлов:

- **Java.module**: Java, Scala, Kotlin, Android;
- **Matcher.module**: ABAP, Apex, C#, COBOL, Dart, Delphi, GO, Groovy, HTML, JavaScript, LotusScript, Pascal, PHP, PL/SQL, Python, Perl, Ruby, Rust, Solidity, Swift, T SQL, TypeScript, VB.NET, VBA, VBScript, Visual Basic, Vyper, 1C;
- **Regex.module**, **Xml.module**: конфигурационные файлы;
- **CFamilyLinux.module**: C, C++;
- **CcppBin.module**: файлы с расширениями `.exe`, `.dll` (только для языков C/C++);
- **iOS.bin.module**: `.ipa`-файлы.

Для изменения конфигурации отредактируйте файл **/opt/derscanner/core/sast/configs/sast-daemon.env** на хосте с установленным компонентом SAST. По умолчанию могут быть запущены все анализаторы. Для исключения анализатора необходимо добавить в файл строку с наименованием компонента и пустым значением строки (““). Чтобы изменения вступили в силу, перезагрузите сервис компонента SAST:

```
sudo systemctl restart derscanner-sast.service
```

Пример: Если на хосте с установленным компонентом SAST следует удалить анализатор, отвечающий за анализ Java, Scala, Kotlin, Android (Java.module), в конец файла следует дописать строку: **Java.module=““**

Если на хосте с установленным компонентом SAST следует удалить анализаторы, отвечающие за анализ исходных кодов C/C++ (CFamilyLinux.module) и Java, Scala, Kotlin, Android (Java.module), в конце файла следует дописать строки: **Java.module=““ CFamilyLinux.module=”“**

Все изменения будут применены только на выбранном хосте.

6.4 Изменение количества потоков и памяти для анализа

Ниже приведено описание для ОС Windows.

По умолчанию на машину выделяется минимум 36 GB RAM и анализы идут в 1 поток в 28 GB (обязательно оставить около 8GB на остальные процессы системы).

От количества потоков анализа зависит, сколько сканирований может быть запущено одновременно, при этом ресурсы машины делятся между потоками (например, на машине с 36 GB RAM при наличии 3 потоков на каждый поток стоит выделить не более 9-9.5 GB). От количества памяти на поток зависит скорость анализа, а также объём кода, который можно проанализировать.

Поток – это логическая сущность, которая означает возможность запуска одного анализатора в один момент времени. Количество потоков анализа можно увеличить путём настройки многопоточности анализатора на одном сервере или подключения внешних агентов.

Для работы система использует автоматическую очередь задач **ActiveMQ Artemis**, которая взаимодействует с установленными на главном сервере или внешнем агенте **Daemon** (сущностями, которые запускают анализаторы), и, при наличии задачи на сканирование (task), отдаёт его в свободный **Daemon**. Когда **Daemon** получает задачи, он запускает анализатор определённого типа, соответственно занимает один поток (выбор анализатора зависит от языка программирования). Когда анализ задачи заканчивается, поток высвобождается и **Daemon** вновь может принимать новые задачи.

Количество потоков в системе определяет количество задач на сканирование, которые могут быть запущены одновременно. Задач на сканирование может быть несколько в пределах одного проекта, поэтому количество потоков соответствует количеству одновременных сканирований, а не проектов.

В системе можно реализовать многопоточность любого компонента (SAST, DAST, SCA) двумя способами. С помощью увеличения количества потоков демона на одном сервере или с помощью добавления внешних агентов. Чтобы установить внешний агент для SAST, нужно установить компонент SAST на сторонний хост, следуя инструкции по установке системы. Затем в файле /opt/derscanner/core/sast/configs/sast-daemon.env заменить значение поля

queueURI на адрес сервера с установленным APP, например: queueURI=tcp://10.10.10.10:61616 После чего перезапустить службу компонента командой:

```
sudo systemctl restart derscanner-sast.service
```

Чтобы изменить количество потоков анализа и количество памяти на каждый поток:

1. Файл %DERSCANNER_HOME%\daemon\daemon.properties содержит базовые настройки конфигурации для сервиса **derscanner-daemon** и представляет собой список в формате ключ-значение.

```
queueURI=tcp://localhost:61616/
queueLogin=dummy_daemon_login
queuePassword=dummy_daemon_hash
queueTimeout=1800
queue.priority.min=1
queue.priority.max=8
workDir=C:/DerScanner/files/d/
cleanDir=true
daemonThreads=1
cache.dir=C:/DerScanner/files/d/cache/
cache.size=10
python.path=C:/DerScanner/3rd-party/python/python.exe
retry.interval.initial=5
retry.attempts=30
retry.interval.max=3600
termination.timeout=180
```

Рис. 6.2: daemon.properties

2. Изменение значений происходит путём создания новых системных переменных в формате, представленном в **daemon.properties.**

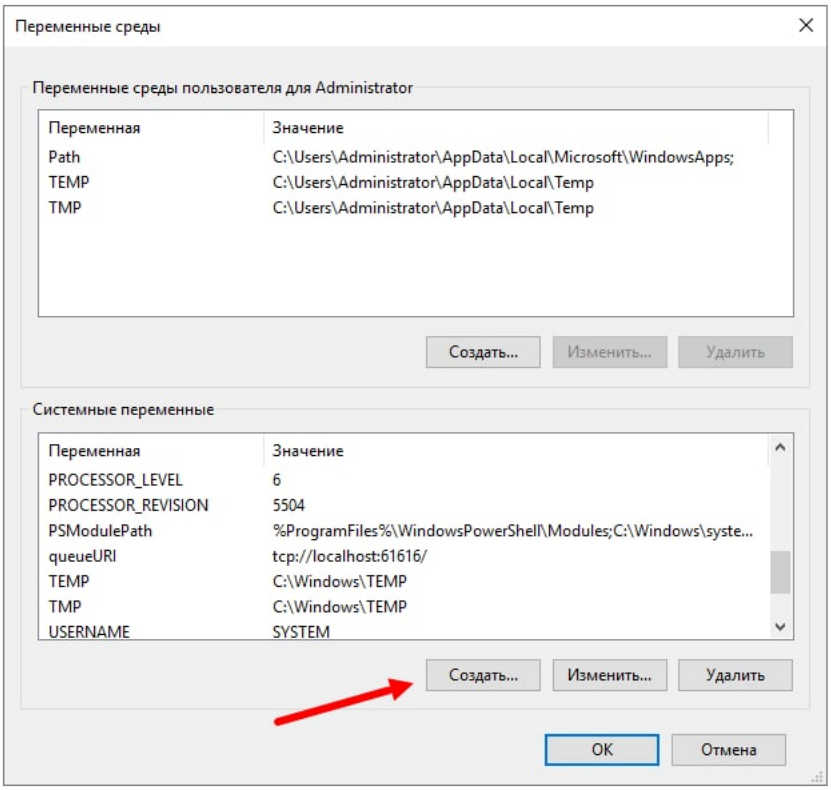


Рис. 6.3: Системные переменные

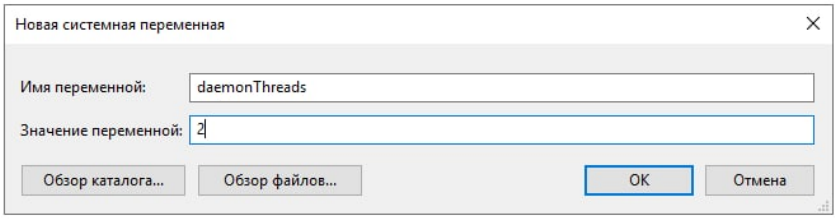


Рис. 6.4: Создание новой системной переменной

3. Например, для компонента **Matcher** необходимо создать новую системную переменную **Matcher.memory** с желаемым значением.

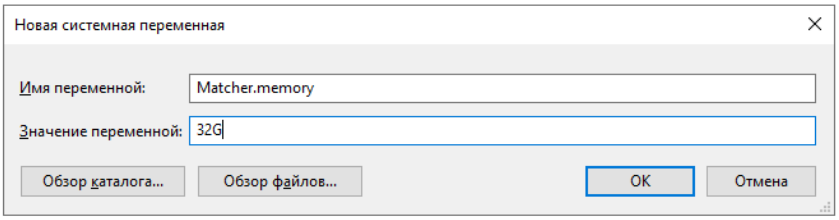


Рис. 6.5: Создание системной переменной для компонента Matcher

4. Перейти в **Службы** и перезапустить сервис **derscanner-daemon**.

Ниже приведено описание для ОС Linux.

Во время работы компонент SAST запускает различные статические анализаторы, такие как: Java, Matcher, Regex, Xml, CFamilyNix, CcppBin, iOS.bin, CFamilyWin. Количество используемой анализаторами оперативной памяти ограниченно в конфигурационных файлах. По умолчанию настроена работа компонента SAST в 1 поток, что означает возможность запускать только одну задачу (анализатор) в один момент времени.

Для увеличения пропускной способности анализатора (возможности сканировать сложные/объёмные конструкции языка), следует увеличить количество возможной доступной памяти для него.

Для увеличения скорости обработки проекта с несколькими языками или возможности запускать параллельно несколько проектов (при условии, что проекты запускают только одну задачу (анализатор), следует увеличить количество потоков на хосте с компонентом SAST или добавить дополнительный сервер (сервера) с развёрнутым компонентом SAST. Один сервер с компонентом SAST с 2 настроенными потоками равноценен двум серверам с компонентами SAST с одним настроенным потоком на компонент. В обоих случаях в системе будет возможность запустить 2 задачи (анализатора) в один момент времени.

Обратите внимание:

Если настраивается работа в несколько потоков (например 2) на одном хосте, следует учесть, что в один момент времени на этом сервере может запуститься две задачи (анализатора), и они могут запросить у операционной системы оперативную память, указанную в настройках статических анализаторов. Поэтому нужно оставлять операционной системе 20% RAM от памяти, выделенной на анализаторы. Если на хосте работают другие компоненты анализа (APP, DAST, SCA), также следует учитывать потребление ресурсов этими мкомпонентами.

Пример: Для сервера с установленным компонентом SAST на отдельном сервере, для включения двух потоков потребуется увеличить оперативную память исходя из расчётов: 22GB на поток * 2 потока = 44 GB + 20% на работу JVM = 53 + 2~4 GB на работу ОС. Итого 57 GB RAM. Количество CPU так же следует увеличить в два раза.

Чтобы изменить количество потоков анализа и количество памяти на каждый поток:

1. Подключитесь к хосту, где развёрнут компонент **SAST**.
2. Откройте конфигурационный файл: **/opt/derscanner/core/sast/configs/sast-daemon.env**
3. Измените количество потоков анализа на желаемое: daemonThreads=....

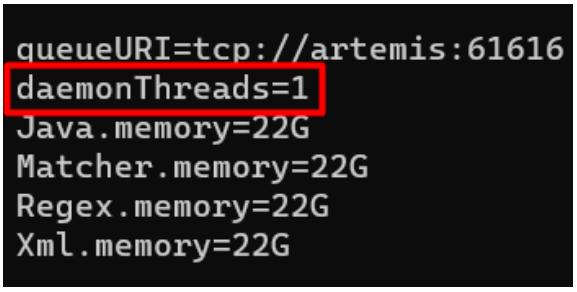


Рис. 6.6: daemonThreads

4. Измените количество памяти на поток анализа на желаемое: Matcher.memory=....

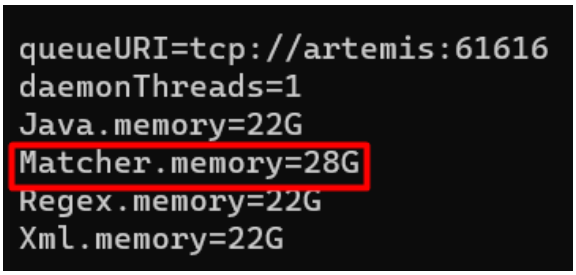


Рис. 6.7: Matcher.memory

5. Перезапустите сервис компонента **SAST** командой:

   ```
   sudo systemctl restart derscanner-sast.service
   ```

6.5 Конфигурация агента сборки

Для изменения настроек агента Builder:

1. Откройте файл на сервере, где развернут агент сборки **/opt/derscanner/builder/configs/builder.env.**
2. Измените необходимые параметры переменных среды:

   AST_BUILDER_PORT — порт на котором работает агент сборки;

   AST_BUILDER_USERNAME — имя пользователя для подключения к агенту сборки;

   AST_BUILDER_PASSWORD — пароль для подключения к агенту сборки.
3. Сохраните изменения.
4. Перезапустите агент сборки:

   ```
   sudo systemctl restart derscanner-builder.service
   ```
5. Отредактируйте соединение с агентом сборки в интерфейсе приложения **Настройки > Система > Общие > Агенты сборки** (см. Агенты сборки).

6.6 Настройка работы агента сборки по HTTPS

По умолчанию агент Builder устанавливается на сервер сборки и работает по протоколу **HTTP**. Для генерации ключа и сертификата используется утилита keytool, которая включена в состав среды функционирования **JDK 11**.

Для настройки работы агента Builder по протоколу **HTTPS** необходимо выполнить следующие действия:

На стороне сервера с агентом Builder

1. Сгенерировать хранилище ключей - **keystore.jks** с сертификатом для Builder внутри. Абсолютный путь **keytool**-утилиты: **/opt/derscanner/builder/environment/openjdk11/bin/keytool**. Пример команды:

   ```
   keytool -genkey -alias builder -keyalg RSA -storepass
   ```

```
keystore-file-password -keystore keystore.jks -validity 3650
```

Параметры **storepass** (пароль для хранилища ключей) и **validity** (количество дней действия хранилища ключей) выбираются пользователем на свое усмотрение.

2. Извлечь самоподписной сертификат **builder.cer** из хранилища ключей, выполнив команду в директории с созданным файлом хранилища ключей **keystore.jks**:

```
keytool -export -alias builder -storepass password

-file builder.cer -keystore keystore.jks
```

3. Изменить файл **/opt/derscanner/configs/builder.env**, указав путь до файла хранилища ключей и пароль от хранилища ключей, например:

```
JKS_PASSWORD=keystore-file-password

JKS_PATH=/opt/derscanner/builder/keystore.jks
```

4. Опционально изменить номер порта работы сервиса в файле **/opt/derscanner/configs/builder.env**, например:

```
AST_BUILDER_PORT=8443
```

5. Перезапустить службу, выполнив команду:

```
sudo systemctl restart derscanner-builder.service
```

На стороне сервера с компонентом APP

1. Перенести на сервер полученный файл сертификата **builder.cer** в виде уникального имени файла для этого агента, например, **builder-agent-1.cer**.
2. Изменить файл **/opt/derscanner/app/configs/backend.env** добавив в поле **CATALINA_OPTS** следующее значение **-Djdk.internal.httpclient.disableHostnameVerification**, например:

```
CATALINA_OPTS="-Xms1024M -Xmx8192M

-Djdk.internal.httpclient.disableHostnameVerification"
```

3. Добавить сертификат **builder-agent-1.cer** в доверенные (см. Добавление самоподписных и корневых сертификатов в доверенные для работы через HTTPS и LDAPS).
4. Перенастроить соединение с агентом сборки на протокол HTTPS в интерфейсе приложения **Настройки > Система > Общие > Агенты сборки** (см. Агенты сборки).

6.7 Подключение приватных Git-репозиториев для установки с ОС Windows

6.7.1 Порядок подключения загрузки из приватных Git-репозиториев по SSH-ссылке

Порядок подключения загрузки из приватных Git-репозиториев по SSH-ссылке следующий:

1. Создать пару ключей без пароля, с помощью которых будет происходить авторизация в репозиториях.

2. Добавить публичную часть ключа в настройках Git-платформы аккаунта, от лица которого будет происходить загрузка.
3. Приватную часть поместить по пути: `C:\Windows\System32\config\systemprofile\.ssh` (если директории `.ssh` не существует, её необходимо создать).
4. В директории `C:\Windows\System32\config\systemprofile\.ssh` создать файл **config** со следующим содержимым:

```
Host *
    StrictHostKeyChecking no
    UserKnownHostsFile NUL
```

5. Перейти в веб-интерфейс системы, добавить SSH-ссылку во вкладке **Домашняя страница > Ссылка на приложение / Путь к репозиторию**.
6. Убедиться, что указанная в настройках сканирования ветка (по умолчанию **master**) существует.
7. Запустить сканирование.

6.7.2 Порядок подключения загрузки из приватных Git-репозиториев по HTTPS-ссылке

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.

Порядок подключения загрузки из приватных **Git-репозиториев** следующий:

1. Подключиться к машине, где развёрнута система.
2. Запустить **cmd** от Администратора и перейти в директорию %DERSCANNER_HOME%\3rd-party
3. Запустить команду: `PsExec.exe -i -s cmd.exe`
4. В новой оболочке ввести:

   ```
   cmdkey /generic:git:https://your_hostname /user:your_username
   /password:your_password
   ```

   (для подключения к GitHub репозиторию в качестве **password** необходимо указать токен аккаунта).
5. После ввода этой команды получить сообщение: *CMDKEY. Учётные данные добавлены успешно.* Проверить это можно также с помощью команды: `cmdkey /list`
6. Закрыть обе командные оболочки, запустить сканирование по HTTPS-ссылке из веб-интерфейса.

6.8 Подключение приватных Git-репозиториев для установки с ОС Linux

Конфигурацией для работы с Git-репозиториями можно управлять в UI (**Настройки** > **Система** > **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
```

6.9 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/derscanner/app/configs/frontend.env**. Детальная информация о настройке прокси сервера и переменных для конфигурирования описана в данной главе.

6.9.1 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 компонентом APP в директорию **/opt/derscanner/app/services/frontend/nginx/ssl** (приватный ключ в формате доступа 600).

После этого нужно отредактировать файл **/opt/derscanner/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 компонента APP:

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

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

6.9.2 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/derscanner/app/configs/frontend.env**. Значение по умолчанию — 600.

Чтобы изменения вступили в силу, перезапустите сервис компонента APP:

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

6.9.3 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/derscanner/app/configs/frontend.env**. Значение по умолчанию — 4GB.

Чтобы изменения вступили в силу, перезапустите сервис компонента APP:

```
sudo systemctl restart derscanner-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 DerScanner в разделе **Настройки** > **Система** > **Общие** > *Максимальный размер загружаемого файла (байты)*. Значение CLIENT_MAX_BODY_SIZE должно быть >= значению в поле **Максимальный размер загружаемого файла**.

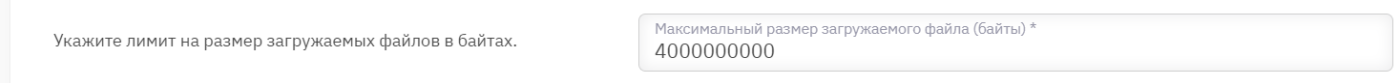


Рис. 6.8: Настройки > Система > Общие > Максимальный размер загружаемого файла

6.9.4 Пользовательские настройки конфигурации NGINX

Для кастомизации файла конфигурации NGINX (директивы, контекст) необходимо добавить переменные с многострочным значением в файл **/opt/derscanner /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/derscanner/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. 6.9: Пример сформированной конфигурации

Чтобы изменения вступили в силу, перезапустите сервис компонента APP:

```
sudo systemctl restart derscanner-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. 6.10: Корректный лог после внесения изменений

6.10 Установка HTTPS соединения (Windows)

Приложение по умолчанию работает по протоколу **HTTP** без шифрования.

Для смены работы по защищённому протоколу необходимо наличие файла сертификата и приватного ключа без пароля для выбранного доменного имени. Их необходимо загрузить на

машину и указать путь до файлов в уже существующем разделе **server** в %DERSCANNER_HOME%\nginx\conf\nginx.conf.

Пути и адрес замените на следующие:

server { listen 443 ssl;

ssl_certificate C:\derscanner\nginx\ssl\example.crt; ssl_certificate_key C:\derscanner\nginx\ssl\example.key;

(*Примечание*. В файле конфигурации перед символом **n** необходимо указывать двойной обратный слэш. В ином случае, сервис NGINX будет воспринимать одинарный слэш как указание к началу новой строки.)

```
http {
    include       mime.types;
    default_type  application/octet-stream;

    #log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
    #                  '$status $body_bytes_sent "$http_referer" '
    #                  '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log  logs/access.log  main;

    sendfile        on;
    #tcp_nopush     on;
    keepalive_timeout  65;

    #gzip  on;

    upstream tomcat {
    server localhost:8080 fail_timeout=0;
   }

    server {
    listen       443 ssl;

    ssl_certificate C:\DerScanner\\nginx\ssl\example.crt;
    ssl_certificate_key C:\DerScanner\\nginx\ssl\example.key;
```

Рис. 6.11: Файл nginx.conf. Настройка HTTPS соединения

Чтобы настроить автоматическое принудительное перенаправление любых запросов с **HTTP** на **HTTPS**, добавьте директиву **server**, содержащую следующие строки:

```
server {
listen 80;


return 301 https://$host$request_uri;
}
```

```
http {
    include       mime.types;
    default_type  application/octet-stream;

    #log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
    #                  '$status $body_bytes_sent "$http_referer" '
    #                  '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log  logs/access.log  main;

    sendfile        on;
    #tcp_nopush     on;
    keepalive_timeout  65;

    #gzip  on;

    upstream tomcat {
    server localhost:8080 fail_timeout=0;
   }

    server {
    listen 80;

    return 301 https://$host$request_uri;
    }

    server {
    listen       443 ssl;

    ssl_certificate C:\DerScanner\\nginx\ssl\example.crt;
    ssl_certificate_key C:\DerScanner\\nginx\ssl\example.key;
```

Рис. 6.12: nginx.conf. Настройка перенаправления с HTTP на HTTPS

После чего перейдите в **Службы** Windows и перезапустите сервис **NGINX**.

6.11 Добавление самоподписных и корневых сертификатов в доверенные для работы через HTTPS и LDAPS для установки с ОС Windows

Для добавления сертификата в доверенные необходимо:

1. Загрузить на хост с основной системой 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: **example.cer**.
2. Запустить **cmd** от имени Администратора и перейти в директорию с JVM: `cd JAVA_HOME/bin` (если переменная `JAVA_HOME` установлена).

3. Выполнить команду, чтобы добавить сертификат в доверенные для Java:

```
keytool -importcert -noprompt -cacerts -storepass changeit -file
"/example.cer"
-alias "example-cer"
```

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

```
keytool -cacerts -storepass changeit -list | findstr example-cer
```

5. Убедиться, что доменное имя сертификата правильно определяется DNS-сервером машины DerScanner:

```
nslookup fqdn.connection.address
```

6. Если имя определяется неверно, можно добавить соответствие в `C:\Windows\System32\drivers\etc\hosts`:

   `IP-address fqdn.connection.address`

   Например: `10.208.64.37 WIN-GR9MM8E5R4G.ad-test.local`

7. Перезапустить ОС.

6.12 Добавление самоподписных и корневых сертификатов в доверенные для работы через HTTPS и LDAPS для установки с ОС Linux

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:/
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 компонента APP:

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

6.13 Увеличение памяти для сервиса Tomcat

Чтобы увеличить объём памяти для сервиса Tomcat в установке с ОС Windows нужно:

1. Подключиться к машине, где развёрнута система.
2. Запустить файл %DERSCANNER_HOME%\tomcat\bin\DERSCANNER-tomcatw.exe.
3. Во вкладке **Java** изменить значение **Maximum memory pool**.

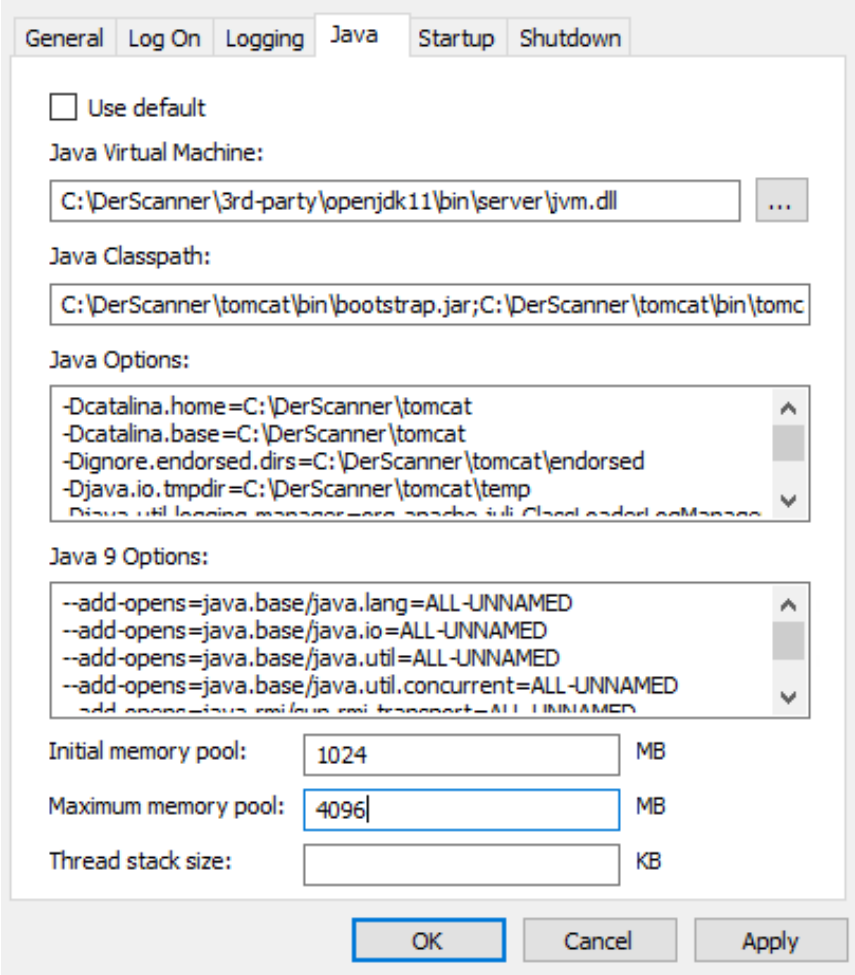


Рис. 6.13: derscanner-tomcat Properties

4. Сохранить изменения.
5. Перезапустить сервис анализатора **derscanner-tomcat** в службах системы.

Чтобы увеличить объём памяти для сервиса Tomcat в установке с ОС Linux:

1. На хосте с компонентом **APP** откройте конфигурационный файл: **/opt/derscanner/app/configs/backend.env**
2. В следующей строке замените значение `Xmx4096M` на `Xmx8192M`:

   `CATALINA_OPTS="-Xms1024M -Xmx4096M"`

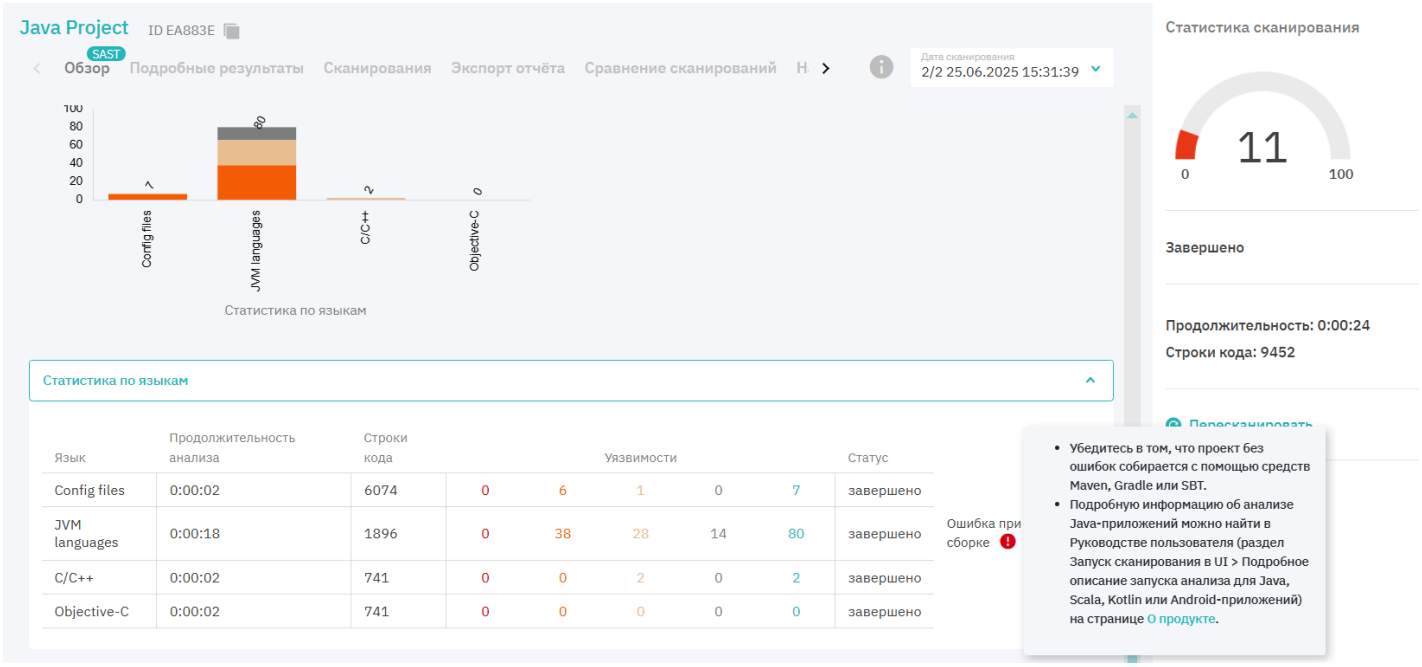
   Должно получиться: `CATALINA_OPTS="-Xms1024M -Xmx8192M"`
3. Закройте файл, сохранив изменения.

4. Перезапустите компонент **APP**:

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

6.14 Что делать при ошибках в сканировании?

При возникновении ошибок во время сканирования обратитесь в службу поддержки. К письму приложите скриншот страницы сканирования с описанием ошибок по языкам программирования. Если ошибок несколько, то приложите несколько скриншотов.



Язык	Продолжительность анализа	Строки кода	Уязвимости					Статус
Config files	0:00:02	6074	0	6	1	0	7	завершено
JVM languages	0:00:18	1896	0	38	28	14	80	завершено
C/C++	0:00:02	741	0	0	2	0	2	завершено
Objective-C	0:00:02	741	0	0	0	0	0	завершено

Рис. 6.14: Ошибки сканирования

В более сложных случаях для диагностики ошибок необходимы файлы журнала событий системы:

- для установки с ОС Windows:

  %DERSCANNER_HOME%/daemon/logs %DERSCANNER_HOME%/tomcat/logs %DERSCANNER_HOME%/nginx/logs

- для установки с ОС Linux можно направить журналы событий системы, собранные Log-bringer (подробнее см. Log-bringer).

6.15 Система регистрации событий

Подсистема регистрации событий DerScanner регистрирует события для каждого компонента (APP, SAST, DAST, SCA).

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

Все события заносятся в журналы событий.

6.15.1 Журналы событий

DerScanner ведет 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 компоненты записывают и хранят журналы событий в файловой системе сервера, на котором они функционируют.

Журналы событий компонента APP

В составе веб-приложения (компонента APP) работают несколько подсистем:

- NGINX - отвечает за хранение каркаса портала и служит reversproxy для обращений извне. Файлы журнала событий NGINX находятся на установке системы по пути **%DERSCANNER_HOME%\nginx\logs** (для Windows) или **/opt/derscanner/app/services/frontend/logs** (для Linux) 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 **%DERSCANNER_HOME%\tomcat\logs** (для Windows) или **/opt/derscanner/app/services/backend/logs** (для Linux) и состоят из:
  - derscanner-error.log (только события с ошибками);
  - derscanner-warn.log (события с ошибками и предупреждениями);
  - derscanner-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 по пути: %DERSCANNER_HOME%\artemis\log. Для Linux файлы журнала событий ActiveMQ Artemis можно получить на сервере с работающим компонентом APP, выполнив команду:

```
sudo docker logs artemis
```

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

```
sudo docker logs postfix
```

Журналы событий компонента SAST

Компонент отвечает за работу алгоритмов статического анализа. При работе компонента на нескольких серверах журналы событий будут храниться в файловой системе сервера отдельно для каждого экземпляра компонента. Журналы daemon (агента, отвечающего за контроль работы анализаторов) разделяются по глубине детализации информации в порядке возрастания:

- %DERSCANNER_HOME%\daemon\logs\daemon.log;
- %DERSCANNER_HOME%\daemon\logs\daemon-debug.log.

или

- /opt/derscanner/core/sast/services/sast-daemon/logs/daemon.log;
- /opt/derscanner/core/sast/services/sast-daemon/logs/daemon-debug.log.

Журналы работы анализаторов разделены по языкам сканирования и находятся в директории **%DERSCANNER_HOME%\daemon\logs\languages** (для Windows) или **/opt/derscanner/core/sast/services/sast-daemon/logs/languages** (для Linux). Для каждого проведённого анализа существует отдельный лог-файл с датой и идентификатором анализа.

Переменная %DERSCANNER_HOME% — путь до директории системы DerScanner, например:

C:\DerScanner; D:\sast\DerScanner.

Журналы событий компонента DAST

Компонент отвечает за работу алгоритмов динамического анализа. При работе компонента на нескольких серверах, журналы событий будут храниться в файловой системе сервера отдельно для каждого экземпляра компонента. Журналы компонента хранятся в директории **/opt/derscanner/core/dast/services/dast-daemon/logs**.

Журналы событий компонента SCA

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

Все журналы производят ежедневную ротацию и архивирование в аллоцированные директории.

6.15.2 Отправка журналов событий ИБ по syslog

Отправка журналов событий ИБ по syslog в системах с ОС Linux. 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 DerScanner на другой сервер по протоколу **syslog** в формате `.cef`:

1. Откройте файл **/opt/derscanner/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 компонента APP:

   ```
   sudo systemctl restart derscanner-app
   ```

6.15.3 Log-bringer

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

Если установлен DerScanner версии 11 или выше, программа для сбора журналов событий вшита в компоненты системы (APP, SAST, DAST, SCA) по умолчанию. Если установлена версия 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

Путь	Описание
docker/docker_stats.log	информация об использованных ресурсах docker контейнеров

Журналы событий компонента APP

Путь	Описание
app-module	каталог с информацией о Fronted, Backend, ActiveMQ Artemis, Postgres, Postfix для компонента APP
app-module/app.compose.yml	docker-compose конфигурация для компонента 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/app-db/scans.log	докер образ app-db с информацией о сканировании SAST из таблицы DB
app-module/app-db/tasks.log	докер-образ app-db с информацией о задачах SAST из таблицы DB
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)
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

Путь	Описание
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 компонента SAST

Путь	Описание
sast-module	каталог с конфигурациями и журналами для компонента SAST
sast-module/sast.compose.yml	конфигурация docker-compose для компонента SAST
sast-module/systemd_sast_service.log	информация о службе sast systemd
sast-module/sast-daemon/daemon.env	конфигурация для службы демона
sast-module/sast-daemon/logs	каталог, содержащий журналы SAST на сервере

Журналы событий компонента DAST

Путь	Описание
dast-module	каталог с конфигурациями и журналами для компонента DAST
dast-module/dast.compose.yml	конфигурация docker-compose для компонента DAST
dast-module/systemd_dast_service.log	информация о службе DAST systemd
dast-module/daemon/dast-daemon.env	конфигурация для службы компонента DAST

Путь	Описание
dast-module/daemon/container_dast_daemon.log	журналы событий из контейнера компонента DAST
dast-module/daemon/logs	каталог, содержащий журналы из компонента DAST
dast-module/zap/zap.env	конфигурация для службы компонента DAST
dast-module/zap/container_zap.log	журналы из контейнера компонента DAST

Журналы событий компонента SCA

Путь	Описание
osa-module	каталог с конфигурациями и журналами для компонента SCA
osa-module/osa.compose.yml	конфигурация docker-compose для компонента SCA
osa-module/systemd_osa_service.log	информация о службе SCA 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
osa-module/osa-daemon/logs	каталог, журналы событий osa-daemon
osa-module/osa/container_osa.log	журналы из контейнера SCA
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

6.15.4 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	derscanner-debug.log
Инициализация поиска соответствия данных с БД	время и ID события, логин пользователя	derscanner-debug.log
Создание токена сессии	время и ID события, логин пользователя, имеет ли пользователь права администратора, LDAP URL, ID пользователя	derscanner-debug.log
Получение токена в Аккаунте	время и ID события, логин пользователя, время действия токена в минутах, имеет ли пользователь права администратора, LDAP URL, ID пользователя	derscanner-debug.log
Попытка изменить пароль	время и ID события, логин пользователя, имеет ли пользователь права администратора, LDAP URL, ID пользователя	derscanner-debug.log
Ошибка ввода неверного пароля	время и ID события, логин пользователя	derscanner-debug.log
Ошибка ввода неверного логина	время и ID события, логин пользователя	derscanner-debug.log
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	derscanner-debug.log
Попытка использования просроченного токена сессии	время и ID события, время истечения действия токена, текущее время, разница в мс	derscanner-debug.log

Тип события	Атрибуты	Файл
Запрос на получение списка проектов	время и ID события, вид и порядок сортировки, требуются ли архивированные проекты, требуются ли проекты без сканирований, языки сканирований, название проекта, статусы проектов имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	derscanner-debug.log
Запрос информации о проекте	время и ID события, UUID проекта, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	derscanner-debug.log
Запрос информации о сканировании	время и ID события, UUID сканирования, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	derscanner-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	derscanner-debug.log
Запрос информации о статистике	время и ID события, UUID сканирования, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	derscanner-debug.log
Запрос информации об уязвимости	время и ID события, ID уязвимости, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	derscanner-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	derscanner-debug.log
Запрос на получение списка правил	время и ID события, язык, настройки фильтрации правил: ключевые слова, список языков, список классификаций, авторы, сортировка, направление сортировки, группировка, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	derscanner-debug.log
Запрос на получение списка наборов	время и ID события, настройки фильтрации наборов: ключевые слова, языки, авторы, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	derscanner-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	derscanner-debug.log

Тип события	Атрибуты	Файл
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	derscanner-debug.log
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	derscanner-debug.log

Тип события	Атрибуты	Файл
Удаление пользователя	время и ID события, UUID удаляемого пользователя, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	derscanner-debug.log
Создание проекта	время и ID события, настройки проекта: ссылка, логин репозитория, название файла, ветка, название проекта, список языков, список наборов правил, флаг использовать пользовательские паттерны, селектор файлов, кодировка исходных файлов, кодировка названий файлов, флаг анализировать стандартные библиотеки, флаг не компилировать проект, флаг инкрементального анализа, флаг проекта Visual Studio, флаг указывающий на сканирование стандартных библиотек, флаг сохранения исходных файлов после анализа, UUID шаблона сканирования, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	derscanner-debug.log
Изменение проекта	время и ID события, UUID настроек проекта, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	derscanner-debug.log
Удаление проекта	время и ID события, UUID проекта, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	derscanner-debug.log

Тип события	Атрибуты	Файл
Запуск сканирования	время и ID события, настройки проекта: ссылка, логин репозитория, название файла, ветка, название проекта, список языков, список наборов правил, флаг использовать пользовательские паттерны, селектор файлов, кодировка исходных файлов, кодировка названий файлов, флаг анализировать стандартные библиотеки, флаг не компилировать проект, флаг инкрементального анализа, флаг проекта Visual Studio, флаг указывающий на сканирование стандартных библиотек, флаг сохранения исходных файлов после анализа, UUID шаблона сканирования, 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	derscanner-debug.log
Приостановка сканирования	время и ID события, UUID сканирования, UUID всех незавершённых задач, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	derscanner-debug.log
Остановка сканирования	время и ID события, UUID сканирования, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	derscanner-debug.log

Тип события	Атрибуты	Файл
Удаление сканирования	время и ID события, UUID сканирования, директория с ресурсами проекта, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	derscanner-debug.log
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	derscanner-debug.log
Создание правила	время и ID события, уровень критичности, описания на русском и английском языках, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	derscanner-debug.log
Изменение правила	время и ID события, уровень критичности, описания на русском и английском языках, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	derscanner-debug.log
Создание набора правил	время и ID события, UUID набора правил, язык программирования, название набора, флаг является ли набор публичным, список ID правил включённых в набор, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	derscanner-debug.log

Тип события	Атрибуты	Файл
Изменение набора правил	время и ID события, UUID набора правил, язык программирования, название набора, флаг является ли набор публичным, список ID правил включённых в набор, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	derscanner-debug.log
Загрузка правил в систему	время и ID события, название архива, имеет ли пользователь права администратора, LDAP URL, ID пользователя, логин	derscanner-debug.log
Ввод неверного пароля	время и ID события	derscanner-debug.log
Ввод неверного логина	время и ID события	derscanner-debug.log
Загрузка проекта по ссылке	время и ID события, ссылка	derscanner-debug.log
Попытка создания пользователя, который уже существует	время и ID события	derscanner-debug.log
Изменение пароля: новые пароли не совпадают	время и ID события	derscanner-debug.log
Ошибка загрузки архива правил	время и ID события	derscanner-debug.log
Ошибка лицензии: кончилась, доступен только просмотр	время и ID события	derscanner-debug.log
Ошибка места жёсткого диска	время и ID события	derscanner-debug.log
Нехватка оперативной памяти	время и ID события	derscanner-debug.log
Отсутствие ответа демона	время и ID события, UUID задачи	derscanner-debug.log
Отсутствие ответа БД	время и ID события	derscanner-debug.log
Ошибки компонента Matcher	время и ID события, путь до файла	log.log (/opt/derscanner/files/d/{taskuuid}/.state/log.log)

Тип события	Атрибуты	Файл
OutOfMemory для Java	время и ID события	/opt/derscanner/sast-daemon/languages/ {LANG}/{DATE}_{TASK_START_TIME}_{SCAN_UUID}.log (либо .log.zip)

6.16 Резервирование данных

Для резервирования данных системы с ОС Linux требуются копии:

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

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

2. **Директорий**:
   - /opt/derscanner/app/services/backend/files/
   - /opt/derscanner/app/services/frontend/nginx/ssl/
3. **Файлов конфигурации**:
   - На сервере с компонентом APP:
     - /opt/derscanner/app/configs/backend.env
     - /opt/derscanner/app/configs/postgres.env
     - /opt/derscanner/app/services/frontend/nginx/templates/default.conf.template
   - На сервере с компонентом SAST:
     - /opt/derscanner/core/sast/configs/sast-daemon.env
   - На сервере с компонентом DAST:
     - /opt/derscanner/core/dast/configs/dast-daemon.env
     - /opt/derscanner/core/dast/configs/zap.env
   - На сервере с компонентом SCA:
     - /opt/derscanner/core/osa/configs/osa-daemon.env
4. **Логов приложения** (опционально, см. Журналы событий).

6.17 Миграция данных DerScanner

6.17.1 Миграция сервера на другой хост

Для миграции сервера DerScanner 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 DerScanner на Host_2.
3. Перенести дамп базы на Host_2.

4. Перенести директорию **/opt/derscanner/app/services/backend/files/** с Host_1 на Host_2 по тому же пути.
5. На Host_2 выполнить:

```
sudo systemctl stop derscanner-app.service
sudo docker volume rm -f app-postgres-data
sudo docker compose -f /opt/derscanner/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 derscanner-app.service
```

6.17.2 Миграция сервера с Windows на Linux

Чтобы выполнить миграцию сервера DerScanner 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:\derscanner\3rd-party\PostgreSQL\13\bin\pg_dump -E UTF-8 -f C:\derscanner\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:\derscanner\db_dump.sql**
   - **C:\derscanner\files\b**
   - **C:\derscanner\files\s**

На **Linux_host**:

1. Установить систему DerScanner.
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/derscanner/app/services/backend/files
   ```

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

8. Перезапустить службу приложения **APP**:

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

6.17.3 Миграция базы данных на другой хост

Для миграции базы данных DerScanner на новый хост необходимо:

На хосте с компонентом **APP**:

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

   ```
   sudo systemctl stop derscanner-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/derscanner/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 компонентом **APP**:

10. Открыть файл **/opt/derscanner/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 derscanner-app
```

6.18 Настройка интеграции с SAP-системами

Для DerScanner с ОС Linux доступна настройка интеграции с SAP-системами.

Обратите внимание: DerScanner не предоставляет возможность интеграции с SAP-системами «из коробки». Для настройки потребуется запустить дополнительные скрипты.

Процесс анализа исходного кода в DerScanner для SAP-систем, как правило, состоит из следующих этапов:

1. Выгрузка исходного кода из SAP-систем в виде файлов.
2. Адаптация формата выгруженных файлов для анализа и очистка директории.
3. Отправка архива с исходным кодом на сервер DerScanner.
4. Проверка полученного кода.
5. Отправка кода на анализ.

Наиболее востребованной является реализация анализа кода по расписанию в автоматическом режиме (приведено далее в примерах). Если анализ будет запускаться вручную, лишние шаги настройки для SAP-систем необходимо опустить.

Требования к реализации интеграции:

- unix-совместимые утилиты и оболочки для работы (unix-подобные ОС на серверах для обеих систем SAP и DerScanner);
- права администратора;
- наличие сетевого соединения между серверами.

6.18.1 Общая инструкция

Для доступа к скриптам в этой инструкции обратитесь в техническую поддержку (sales@dergrup.pro).

Обычно проверка кода проводится для ландшафтов всех систем, за исключением продуктовых. При наличии синхронизации настроек между тестовой и продуктовой средой происходит перезаписывание настроек тестовой среды, поэтому необходимо также провести настройку продуктовой среды. Имена и значения сред, путей и других переменных приведены для примера и могут изменяться по желанию.

6.18.1.1 Настройки на стороне SAP

1. Установить экстрактор кода: https://login.derscanner.com/dist/software/sap-mass-abap-download-program-master.zip
2. Создать программу с именем `ZDTP_MASSDOWNLOAD`
3. Создать внешнюю команду `ZDTX_MKDIR` (внешняя команда запускает unix-команду **mkdir /usr/sap/trans/derscanner/**).
4. Создать внешнюю команду `Z_CLEANUP_ANALYZER_DIR` (внешняя команда запускает unix-команду **rm –rf /usr/sap/trans/derscanner/**).[^1]
5. Создать логическое имя **Z_DERSCANNER_FILE** (физический путь **/usr/sap/trans/REPORT_work/derscanner/**).
6. Создать внешнюю команду `ZTRANSFER_BACKDOOR` (команда запускает скрипт для архивации файлов и передачи на сервер проверки кода, скрипт в приложении[^2], в нём необходимо подставить значение **SID**).
7. В зависимости от системы, для настройки ролей может потребоваться добавить значение `ZDTX_MKDIR` и `ZTRANSFER_BACKDOOR` для ОП `S_LOG_COM` в роли `ZLO_CUSTOMIZE`.
8. Создать три варианта запуска программы **ZDTP_MASSDOWNLOAD** по аналогии с SAP ERP: **CLASSES**, **FM** и **PROGRAMS**. Описание приводится ниже в описании вариантов.
9. Передача архива на сервер проверки кода происходит с помощью команды `scp`. Для этого публичный ключ SAP-сервера добавляется на сервер проверки кода, устанавливая одностороннее доверенное соединение. Необходимо передать публичный ключ на почту администратору сервера DerScanner.
10. Создать задание **DerScanner** и настроить запуск задания по расписанию в ночное время (в 01:00) от имени технической учётной записи вида `BJOB_BC_<имя_системы>` (например, `BJOB_BC_ER`, `BJOB_BC_PD` и т.п.). Задание состоит из следующих шагов:
    1. Запуск внешней команды по очистке директории `Z_CLEANUP_ANALYZER_DIR`.
    2. Запуск программы **ZDTP_MASSDOWNLOAD** с вариантом **CLASSES**.
    3. Запуск программы **ZDTP_MASSDOWNLOAD** с вариантом **FM**.
    4. Запуск программы **ZDTP_MASSDOWNLOAD** с вариантом **PROGRAMS**.
    5. Запуск внешней команды `ZTRANSFER_BACKDOOR` (ОС – Unix, система, например, er0cidb, er1cidb и т.п.).

6.18.1.2 Описание вариантов

1. **CLASSES**: в разделе **Objects to download** выбирается опция **Classes**.
2. **FM**: в разделе **Objects to download** выбирается опция **Function modules**.
3. **PROGRAMS**: в разделе **Objects to download** выбирается опция **Programs**.

Не указанные ниже параметры остаются заполненными по умолчанию.

1. В поле **Package** в первом подполе указывается значение **Y***, установлен выбор **Больше или равно**.
2. Отмечается настройка **Only customer objects**.

3. В разделе **Additional downloads for programs, function modules and classes** выбираются только следующие опции:
   - **Include programs** (с опцией **Recursive search**);
   - **Function modules** (с опцией **Recursive search**);
   - **Class recursive search**.
4. Указывается опция **Text document**.
5. Указывается опция **Download to server** с указанием пути **Z_DERSCANNER_FILE**.

6.18.2 Интеграция с SAP со стороны DerScanner

Для получения ABAP-кода с серверов SAP-систем используется защищённый протокол SCP. Для осуществления взаимодействия необходимо создать linux-пользователя на основном сервере DerScanner, который будет:

1. Получать исходный код с серверов SAP на сервер DerScanner.
2. Запускать задачу анализа полученного кода в системе.

Пользователь создаётся командой:

```
sudo useradd –U -m -s /bin/bash -d /opt/sap-transfer sap-transfer
```

После чего необходимо создать директорию `/opt/sap-transfer/.ssh` и файл `/opt/sap-transfer/.ssh/authorized_keys`, в который будут помещаться публичные части SSH-ключей для каждого сервера SAP-систем сред разработки и тестирования. Директория и все файлы в ней должны принадлежать пользователю `sap-transfer`, но не должны быть доступны для группы пользователя и других пользователей, в связи с чем необходимо выполнить команды:

```
sudo chmod go-rwx /opt/sap-transfer/.ssh

sudo chown sap-transfer:sap-transfer /opt/sap-transfer/.ssh
```

Для изоляции кода, который приходит для анализа от различных SAP-систем, необходимо в домашнем каталоге `/opt/sap-transfer` создать директории с названием систем, например: `er0`, `if1` и т.п.

В каждую директорию помещается скрипт, который обрабатывает полученный архив и отправляет его на анализ. В приложении[^3] необходимо подставить значение переменной `sap_system_name`, а также добавить суффикс в название, соответствующее директории и полному имени SAP-системы, архив которой отправляется на анализ.

После этого от лица пользователя `sap-transfer` создаётся задача по проверке наличия вновь пришедшего для анализа архива в `cron`. Для этого необходимо выполнить команду:

```
sudo su sap-transfer -c "crontab –e"
```

где вписать:

```
*/10 * * * * /opt/sap-transfer/<SID>/check_and_transfer_<SID>.sh
```

подставив необходимое имя системы.

Скрипт для запуска анализа кода по расписанию можно запросить у поддержки

6.18.3 Пример практической реализации со стороны SAP

1. Создать в системе отчет `ZDTP_MASSDOWNLOAD`. Исходный текст в приложении[^5].
2. Поместить в пакет `ZBC*` (если нет, создать пакет) и активировать.
3. Создать в транзакции `FILE` логический путь:
   - **Z_DERSCANNER_FILE**
   - Присвоить логическому пути физический:
   - `UNIX`
   - **/usr/sap/trans/derscanner/**
   - сохранить, поместить в запрос, полученный выше
4. Создать логическое имя файла:
   - **Z_DERSCANNER_FILE**
   - присвоить ему логический путь: **Z_DERSCANNER_FILE**
   - на хосте создать каталог **/usr/sap/trans/derscanner** (есть ограничения на длину абсолютного имени файла) со следующими правами - `sidadm:sapsys 644`
5. В SM49 создать команду: `ZDTX_MKDIR (UNIX) mkdir` (можно положить в запрос для удобства)
6. В SM49 создать команду:
   - `ZTRANSFER_BACKDOOR`
   - `UNIX`
   - `/usr/sap/scripts/pack_n_transfer_code.sh` (Приложение - Скрипт архивирования и отправки кода с сервера SAP на DerScanner[^2]). Можно положить в запрос для удобства.
7. В SM49 создать команду:
   - `Z_CLEANUP_ANALYZER_DIR`
   - `UNIX`
   - **/usr/sap/scripts/derscanner/transfer_dir_cleanup.sh** (Приложение - Скрипт предварительной очистки директории с выгруженным кодом[^1]). Можно положить в запрос для удобства.
8. Выложить скрипты на хост системы. В скрипте `pack_n_transfer_code.sh` (Приложение - Скрипт архивирования и отправки кода с сервера SAP на DerScanner[^2]) необходимо заменить значение **SID** на **SID** той системы, для которой выполняется настройка. Как указано на (рис. 6.15), **SAP system ID** в скрипте по отправке ABAP-кода, необходимо использовать нижний регистр. Необходимо поставить **X** (выполнение) в разрешениях у скриптов. Также если это AIX, то в скрипте нужно поменять `–iname` на `–name`.

```
#!/bin/bash

working_directory=/usr/sap/trans/
system=SID
archive_name=$system.zip
host=MSK-DPRO-APP538
target_dir=/opt/sap-transfer
user_name=sap-transfer
```

Рис. 6.15: Скрипт архивирования и отправки кода с сервера SAP на DerScanner

9. От лица `sidadm` сгенерировать `id_rsa.pub`:
   - `ssh-keygen -t rsa`
   - Принимать все вопросы по умолчанию до результата: **Your identification has been saved in /home/if0adm/.ssh/id_rsa. Your public key has been saved in /home/if0adm/.ssh/id_rsa.pub. The key fingerprint is: SHA256:NUIxP4n/phY2jVC5iAWtafpPavHXU8wDZpFCb0FQqms if0adm@if0cidb**
   - Сохранить: `/home/if0adm/.ssh/id_rsa.pub`
10. Отправить публичную часть в письме администраторам сервера DerScanner с просьбой добавить на сервер **DERSCANNER-SERVER-HOSTNAME** публичный ключ.
11. Далее, создать 3 варианта для программы `ZDTP_MASSDOWNLOAD`, как описано в инструкции по аналогии с `ER0` (Описание вариантов):
    - FM
    - CLASSES
    - PROGRAMS
12. После добавления публичного ключа в авторизованные на сервер DerScanner, выполнить от лица `sidadm`:

    ```
    ssh sap-transfer@DERSCANNER-SERVER-HOSTNAME ls /opt/sap-
    transfer
    ```

13. В появившемся диалоге с добавлением в `known_hosts` выбрать добавление -— это необходимо для работы скрипта.
14. Запланировать задание по аналогии под используемым техническим пользователем. **Z_DERSCANNER_DAILY** на 01.00 на ежедневной основе.

    Пример в FK4, Задание **Z_DERSCANNER_DAILY** для FK4:

№	Name of program/command	Program type	Parameter	User	Lang
1	Z_CLEANUP_BKDR_DIR	External comm.		BJOB_BC_FK4	
2	ZDTP_MASSDOWNLOAD	ABAP	CLASSES	BJOB_BC_FK4	EN
3	ZDTP_MASSDOWNLOAD	ABAP	PROGRAMS	BJOB_BC_FK4	EN
4	ZDTP_MASSDOWNLOAD	ABAP	FM	BJOB_BC_FK4	EN
5	ZTRANSFER_BACKDOOR	External comm.		BJOB_BC_FK4	

1. Запустить и проверить корректное выполнение задания.

2. С хоста под `sidadm` проверить, что файлы были получены, с помощью команды:

   ```
   ssh sap-transfer@DERSCANNER-SERVER-HOSTNAME ls -ltr sid
   ```

   (`sid (system id)` строчными буквами). Пример:

   ```
   ssh sap-transfer@DERSCANNER-SERVER-HOSTNAME ls -ltr er0
   ```

3. Выполнить данные настройки для всех ландшафтов, кроме продуктива.
4. Скопировать запрос в продуктив для сохранения выставленных настроек.

В конечном итоге должно осуществляться ежедневное задание из 5-ти шагов по отправке кода на сервер для всех систем. В продуктивных средах должны присутствовать все настройки, кроме передачи кода.

07 Получение технической поддержки

Для получения консультации по техническим вопросам можно обратиться по адресу sales@dergrup.pro.

08 Список компонент анализа и поддерживаемых ими языков

Таблица 8.1: Список компонент анализа и поддерживаемых ими языков

Компонент	Языки/Расширения
Java	Java, Scala, Kotlin, Java для Android
Matcher	ABAP, Apex, C#, COBOL, Dart, Delphi, GO, Groovy, HTML, JavaScript, LotusScript, Pascal, PHP, PL/SQL, Python, Perl, Ruby, Rust, Solidity, Swift, T SQL, TypeScript, VB.NET, VBA, VBScript, Visual Basic, Vyper, 1C
Regex	конфигурационные файлы
Xml	конфигурационные файлы
CFamilyLinux	C, C++ для Linux
CFamilyWin	C, C++ для Windows
CcppBin	`.dll`, `.exe` файлы для C, C++
iOS.bin	`.ipa` файлы

09 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