

SAST.V.SCA

Российское решение
для композиционного анализа приложений

ПОЧЕМУ SCA ВАЖЕН СЕЙЧАС

Использование Open Source - индустриальный стандарт: time-to-market сокращается, издержки снижаются, однако одновременно с этим растут риски:

- уязвимости в используемых компонентах,
- протестное ПО (protestware),
- лицензионные ограничения, конфликтующие с политикой компании.

СОВРЕМЕННЫЕ ЦЕПОЧКИ ПОСТАВОК ПРОГРАММНЫХ ПРОДУКТОВ СЛИШКОМ СЛОЖНЫ, ЧТОБЫ ПОЛАГАТЬСЯ ТОЛЬКО НА ДОВЕРИЕ – НУЖЕН КОНТРОЛЬ.

SASTAV.SCA

Мы разработали модуль SCA, который делает использование Open Source безопасным и управляемым. Его задача – дать вашей команде уверенность в компонентах, с которыми вы работаете, и освободить время для главного – разработки ценного продукта.

КЛЮЧЕВЫЕ ВОЗМОЖНОСТИ

Точность и полнота: собственная логика определения наличия уязвимости на базе качественных источников и экспертизы в protestware. Точнее находим – быстрее исправляем.

Лицензионный контроль: выявление рисков используемых в компании лицензий OS компонентов и несоответствий политики компании.

Широкий охват: поддержка исходного кода, дистрибутивных пакетов, Docker-образов, SBOM, директорий и отдельных файлов.

Интеграция: совместимость с Nexus, JFrog, Harbor и другими репозиториями артефактов.

Dependency Firewall: контроль открытых компонентов до их попадания в компанию.

Два режима работы: классический анализ и сетевой проху-режим для раннего перехвата.

Встраивание в SDLC: бесшовная интеграция в CI/CD, работа в закрытых контурах.

Управляемость: ролевая модель доступов, гибкие политики, блокирующие проверки безопасности и пороги качества.

Наглядность: интерактивный граф зависимостей и детальный состав компонентов: от пакета и версии до лицензии и риска.

РЕЖИМЫ РАБОТЫ

Два режима – один результат: ранний перехват рисков.

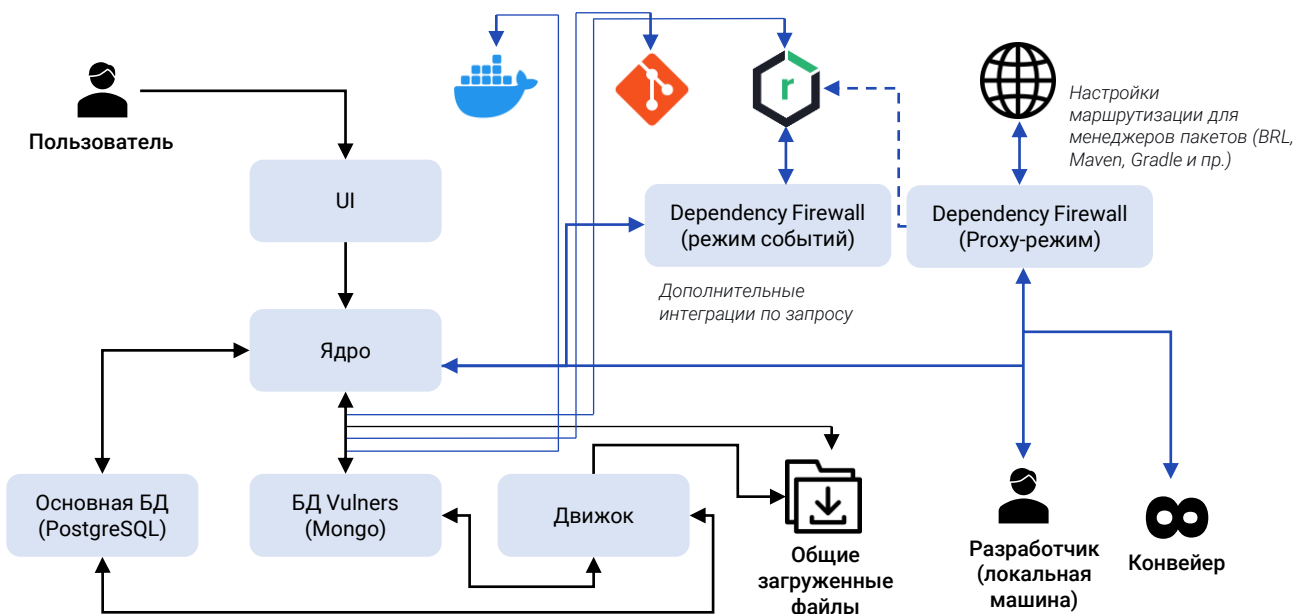
Классический SCA

- Высокоточный поиск и приоритизация уязвимостей;
- интеграция в CI/CD, работа в закрытых контурах;
- встроенные политики и пороги качества в релизном процессе.

Dependency Firewall

- Установка в качестве активного или пассивного контроля за репозиториями артефактов;
- режимы работы проху для потока данных и event для реакции на события под разные сценарии и процессы компании;
- блокирующие проверки безопасности и пороги качества на уровне артефактов в репозиториях.

АРХИТЕКТУРА SCA



SAST.V Артефакт demo.jar

demo.jar
r765756756

Дата создания: 17.08.2025, 17:22:42
Последнее обновление: 17.08.2025, 17:27:34

Уязвимости: Critical - 1, High - 12, Medium - 5, Low - 4, Info - 0

Сработки политик: 1 (Блокирующие - 1, Индикативные - 0, Информационные - 0)

Топ технологий: java

Топ лицензий: [График]

Превышены пороговые значения политики по уязвимостям: Critical - 0 High - 1 Medium - 2

Компонентный состав

Введите название ... Поиск

- pkg:maven/org.apache.tomcat.embed/tomcat-embed-core@10.1.17
- pkg:maven/org.springframework/spring-web@6.1.2
- pkg:maven/org.springframework/spring-context@6.1.2
- pkg:maven/org.springframework/spring-webmvc@6.1.2
- pkg:maven/ch.qos.logback/logback-core@1.4.14
- pkg:maven/org.springframework.boot/spring-boot@3.2.1
- pkg:maven/org.springframework/spring-core@6.1.2

Информация о компоненте

Наименование: tomcat-embed-core
Версия: 10.1.17
Purl: pkg:maven/org.apache.tomcat.embed/tomcat-embed-core@10.1.17
Технология: java
Путь: /demo.jar:BOOT-INF/lib/tomcat-embed-core-10.1.17.jar
Вендор:
Лицензии: <https://www.apache.org/licenses/LICENSE-2.0.txt>

Обнаруженные уязвимости

Уязвимость	Критичность	CVSS 2	CVSS 3.1	Статус
GHSA-83qj-6f2-vhqg	CRITICAL	Не указано	9.8	CRITICAL

КРАТКОЕ ОПИСАНИЕ
Apache Tomcat: Potential RCE and/or information disclosure and/or information corruption with partial PUT

РАСШИРЕННОЕ ОПИСАНИЕ
Path Equivalence: 'file.Name' (Internal Dot) leading to Remote Code Execution and/or Information disclosure and/or malicious content added to /uploaded files via write enabled Default Servlet in

ХАРАКТЕРИСТИКИ
Вектор атаки: NETWORK
Сложность эксплуатации: LOW

Админов А.А. SCA UI v1.0.0

В разделе просмотра результатов сканирования отображается статистика по сканированию, информация о найденных компонентах (компонентный состав) и их глубине, информация по выбранной компоненте, уязвимости выбранной компоненты и информация о выбранной уязвимости.

SAST.V Артефакт Dockerfile

Dockerfile
54323

Дата создания: 14.08.2025, 14:27:51
Последнее обновление: 14.08.2025, 14:28:58

Уязвимости: Critical - 0, High - 2, Medium - 0, Low - 0, Info - 0

Сработки политик: 0 (Блокирующие - 0, Индикативные - 0, Информационные - 0)

Топ технологий: deb, apk, python, go

Топ лицензий: Other, MIT

Скачать граф зависимостей

- корневые компоненты
- прямые зависимости корневых компонентов
- промежуточные транзитивные зависимости
- крайние транзитивные зависимости

Компонентный состав

Информация о компоненте

Админов А.А. SCA UI v1.0.0

В разделе просмотра результатов сканирования есть подраздел с графом зависимостей, на котором помечены уровни зависимостей и их взаимосвязанность.

SAST.V < Политики Добавить

Проекты
Реестр уязвимостей
Реестр зависимос...
Политики
Настройки

Админов А.А.
SCA UI v1.0.0

Структура

- Хранилище проектов
 - Демо
 - События
 - Nexus Events**
 - Proxy Events

Информация об узле

Наименование: Nexus Events
Тип: Репозиторий артефактов
Дата создания: 14.08.2025
Дата обновления: 14.08.2025

Введите название политики ... Поиск

Примененные политики

Наименование	Критичность	Тип	Область применения	Автор	Дата создания
PROTESTWARE	Блокирующая	По уязвимости	Организация	admin	14.08.2025
Main Org	Блокирующая	По кол-ву уязвимостей	Организация	admin	15.08.2025

Кол-во элементов на странице 10 1-2 из 2

В разделе настройки политик безопасности отображается иерархия сущностей системы, к которым применяются политики, политики и информация по ним.

Разработка компании ShiftLeft Security

115114, г. Москва, ул. Дербеневская,
д. 15Б, помещ. 2/1, 3 этаж, офис 306

+7 (499) 502-13-75
support@sastav.ru



sastav.ru