MONHOUSE

FAILOVER BAR

Мониторинг уязвимостей тоже мониторинг?

Алексей Смирнов, CodeScoring

CODE SCORING

РОССИННО

SAYMON TECHNOLOGIES

ЦЕНТРАЛЬНЫЙ ПУЛЬТ

whoami & whoweare

Алексей Смирнов,
основатель CodeScoring,
решения композиционного анализа (SCA)

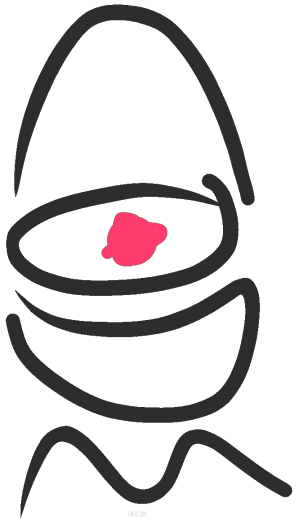
Делаем CodeScoring, tg: @codescoring

Просвещаем в OpenDataScience, tg: @codemining

- 3 года ведем конференцию про анализ кода
- Делаем бесплатные курсы
- Публикуемся и выступаем

План митапа

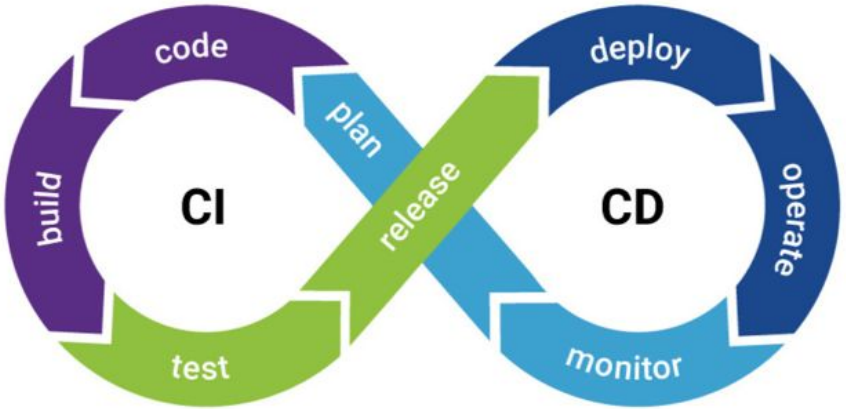
- Проблематика безопасной разработки с OSS
- Принципы выстраивания безопасной разработки с OSS
- Основные принципы работы систем композиционного анализа
- Обзор платформы CodeScoring
- Возможности практического применения CodeScoring



OSS & Security

DevOps -> DevSecOps

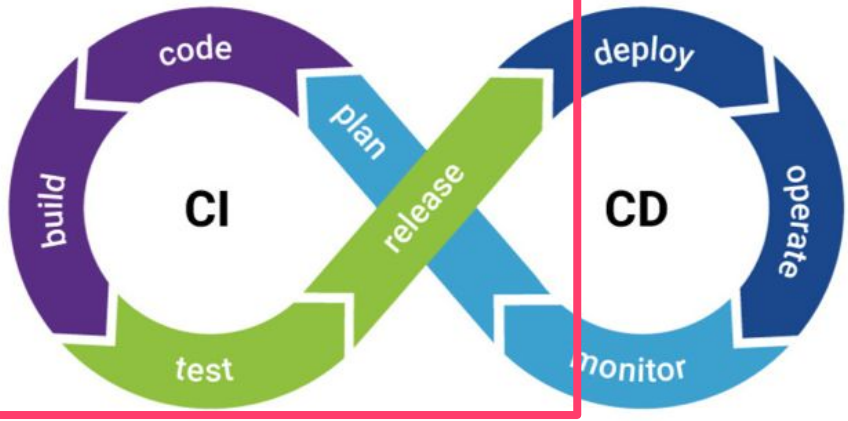
- **SAST**
  Static Application Security Testing
- **DAST/IAST**
  Dynamic/Interactive Application Security Testing
- **OSA**
  Open Source Analysis
- **SCA**
  Software Composition Analysis



- **BAST**
  Behavioral Application Security Testing
- **BCA**
  Bytecode & Container Analysis
- **WAF**
  Web Application Firewall

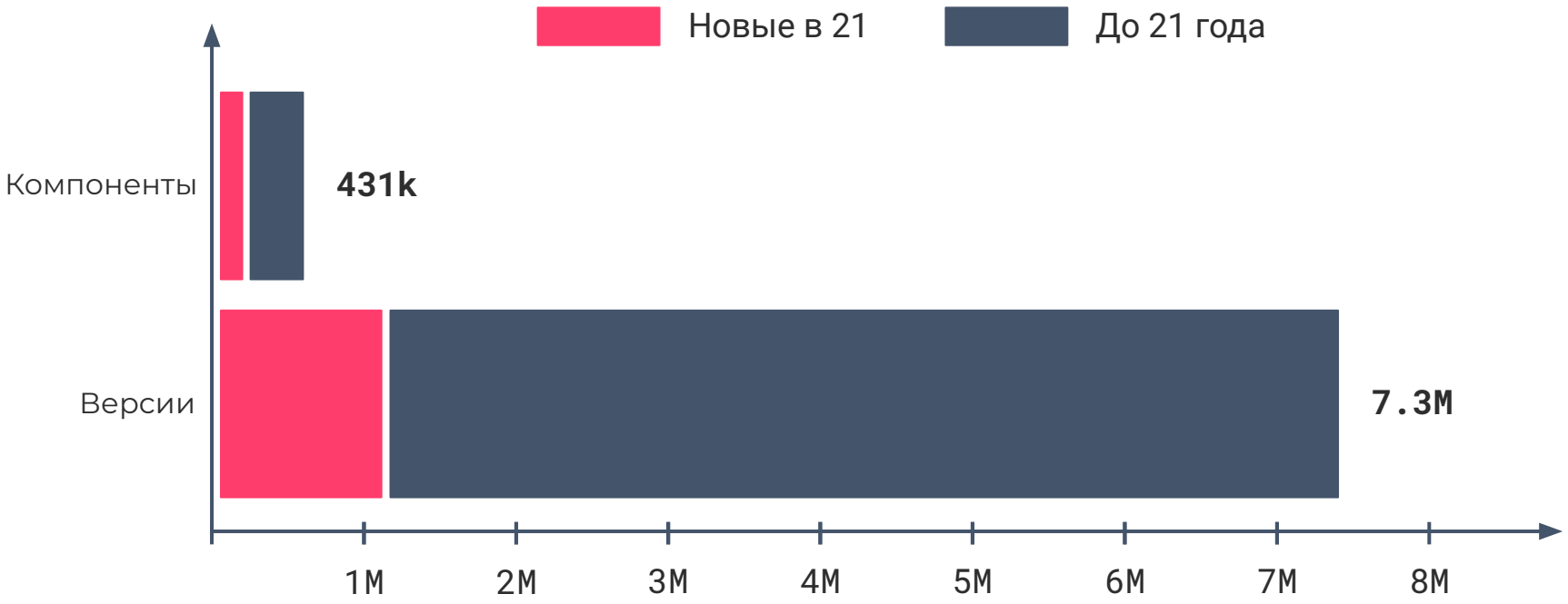
DevOps -> DevSecOps

- **SAST**
  Static Application Security Testing
- **OSA**
  Open Source Analysis
- **SCA**
  Software Composition Analysis

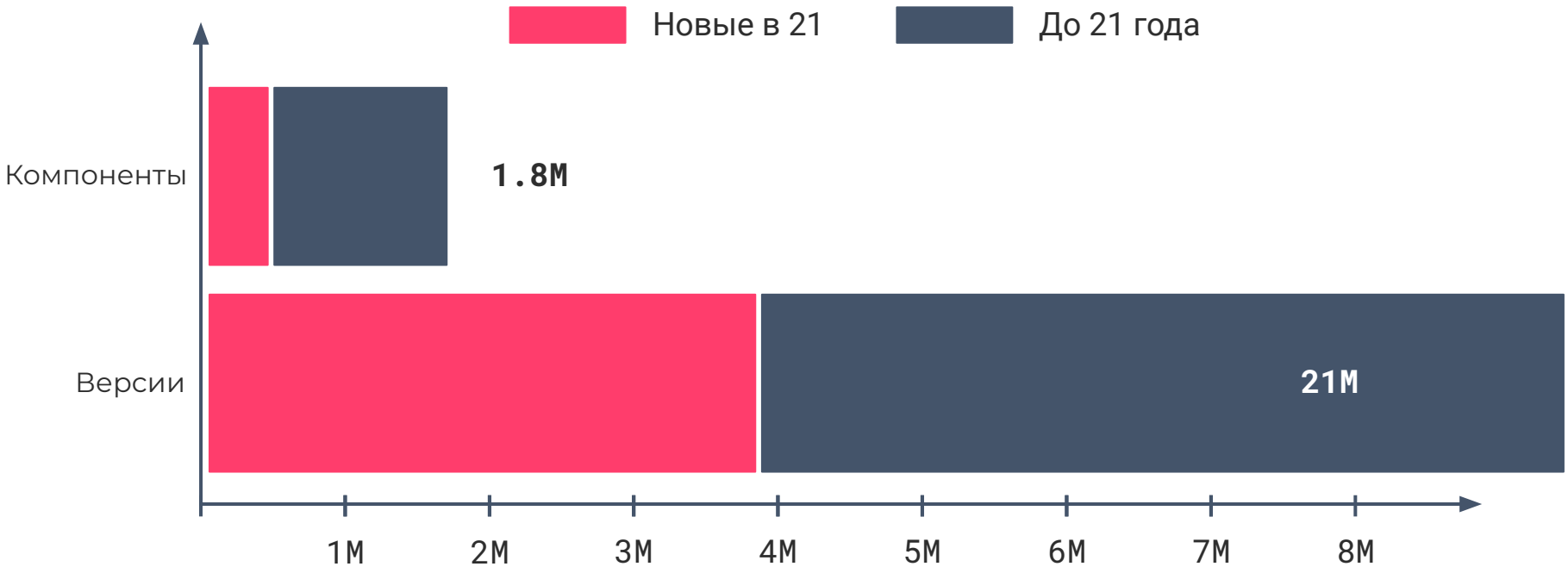


- **DAST/IAST**
  Dynamic/Interactive Application Security Testing
- **BAST**
  Behavioral Application Security Testing
- **BCA**
  Bytecode & Container Analysis
- **WAF**
  Web Application Firewall

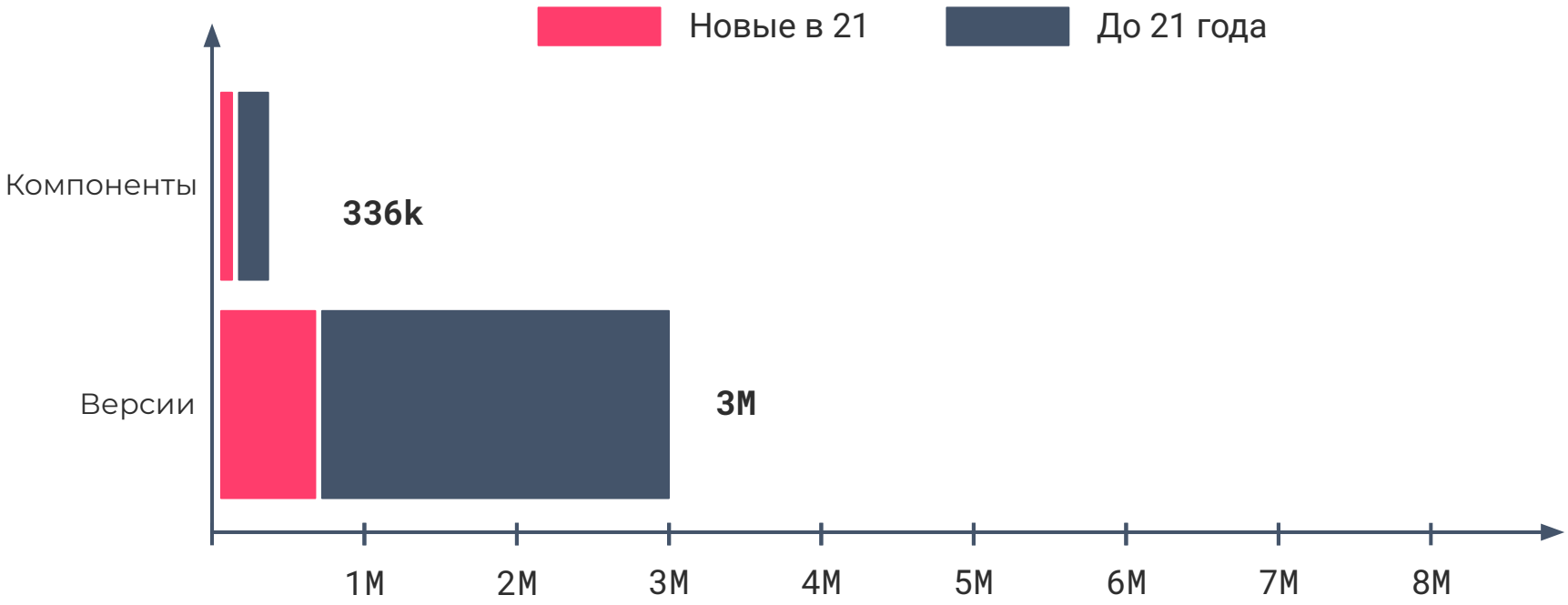
Open Source растёт /Java



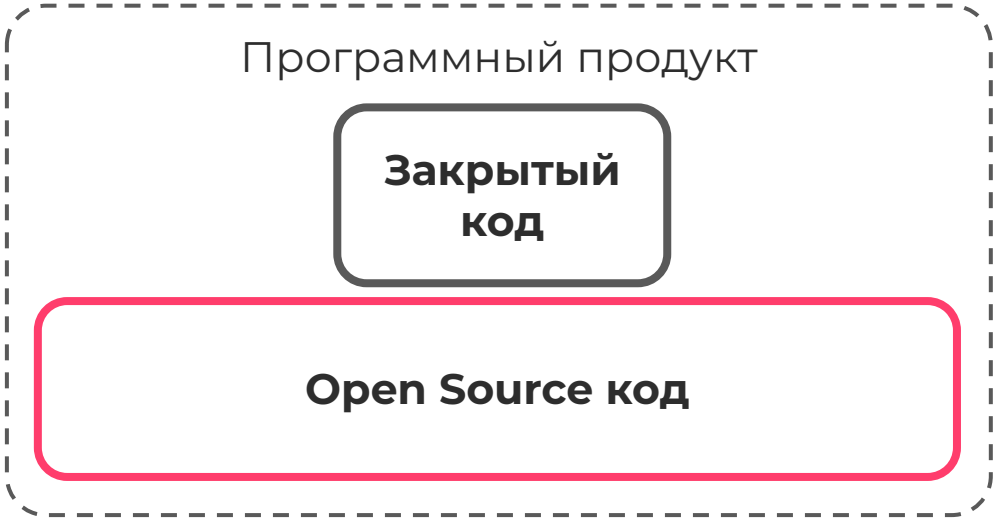
Open Source растёт /JavaScript



Open Source растёт /Python



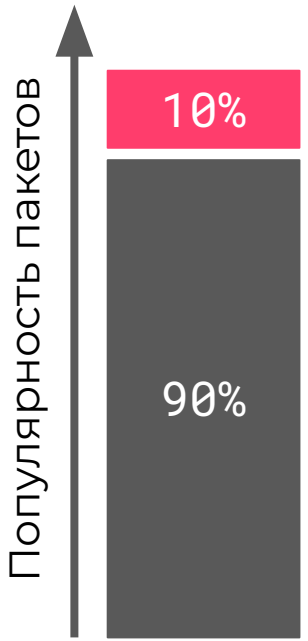
Open Source > Proprietary



Неизвестен
Нужно сканировать
Сложно исследовать

Известен
Идентифицируем
Исследуем

Популярное лучше изучено



в 36% проектов выявлены уязвимости

в 8% проектов выявлены уязвимости

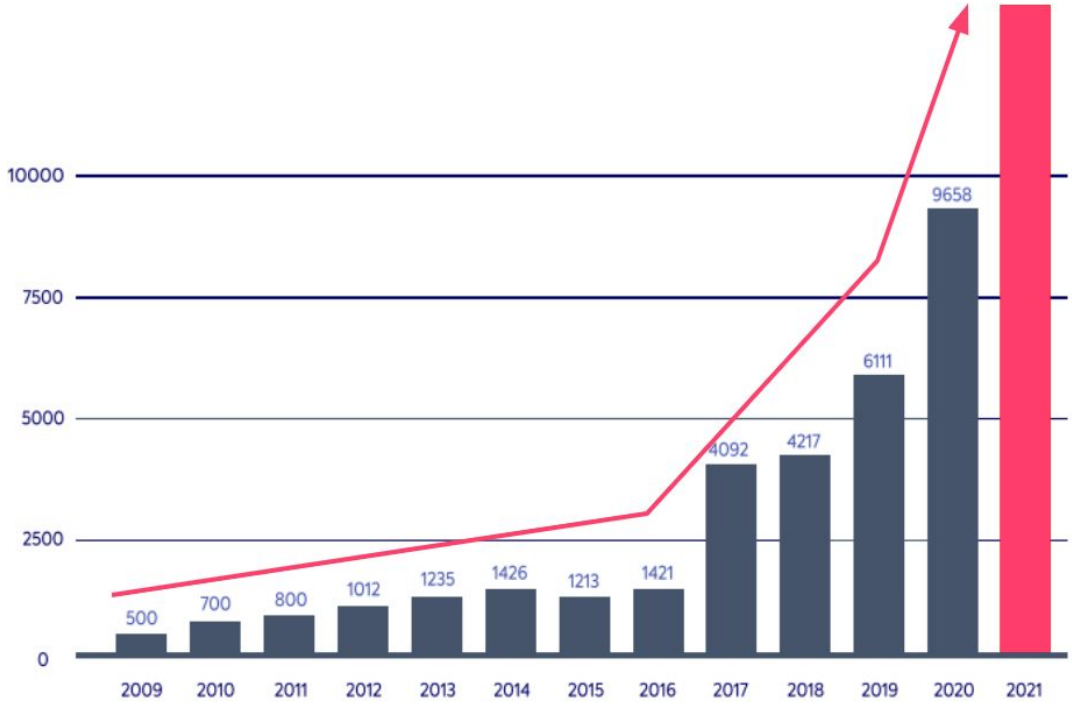
Open Source небезопасен 💔

- **Уязвимости**
  Появляются каждый день. На многие уязвимости есть эксплоит на GitHub. Время подготовки атаки существенно сократилось.

- **Лицензии**
  Их отсутствие, лицензионная (не)совместимость, риск смены лицензии на более строгую, риск введения экспортных ограничений.

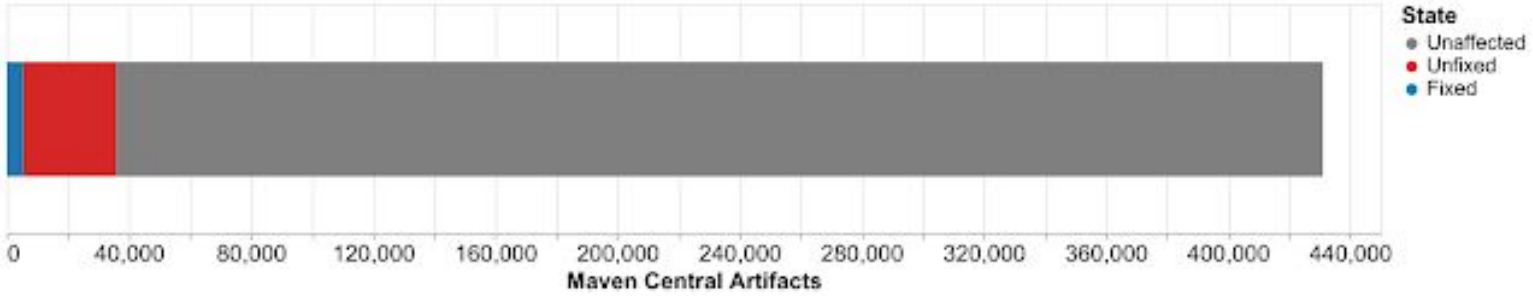
- **Закладки**
  Такие закладки могут быть активированы по разным признакам: locale, geoip, geoloc, etc.

Open Source vulns trend



Пример. log4j (log4shell)

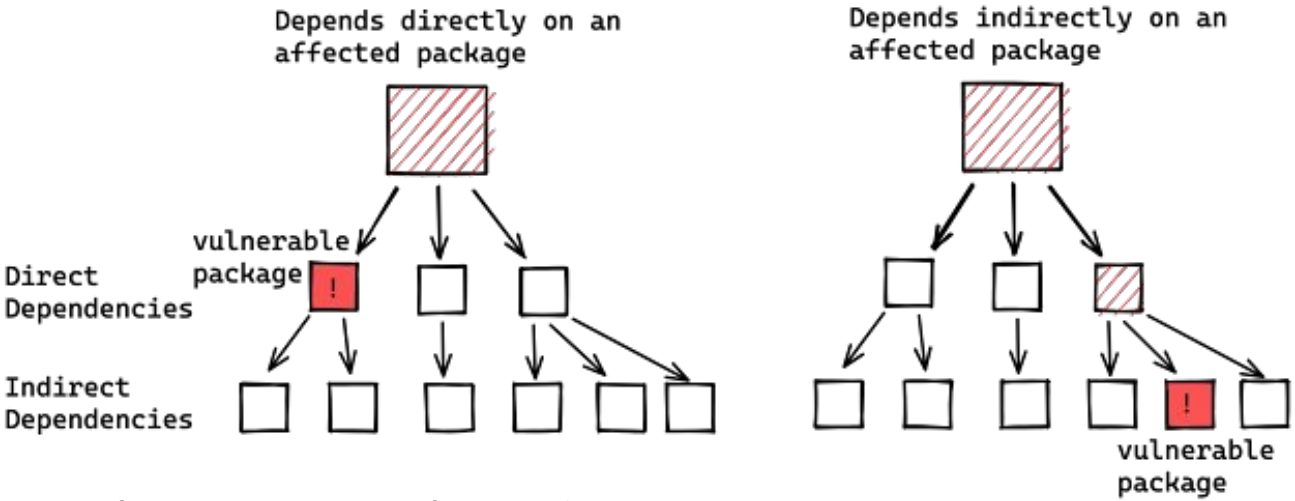
CVE-2021-44228, 10.12.2021 и CVE-2021-45046, 14.12.2021.
Выполнение произвольного кода на сервере
(Arbitrary Code Execution, ACE).



December 16, 2021. Google Security

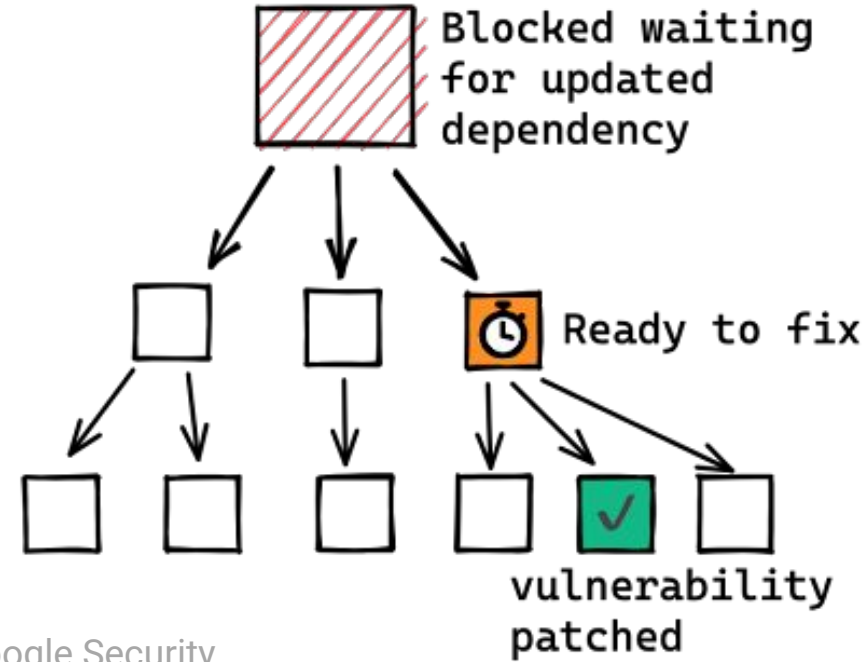
log4shell. Глубже!

Уязвимый компонент может быть в транзитивных зависимостях



December 16, 2021. Google Security

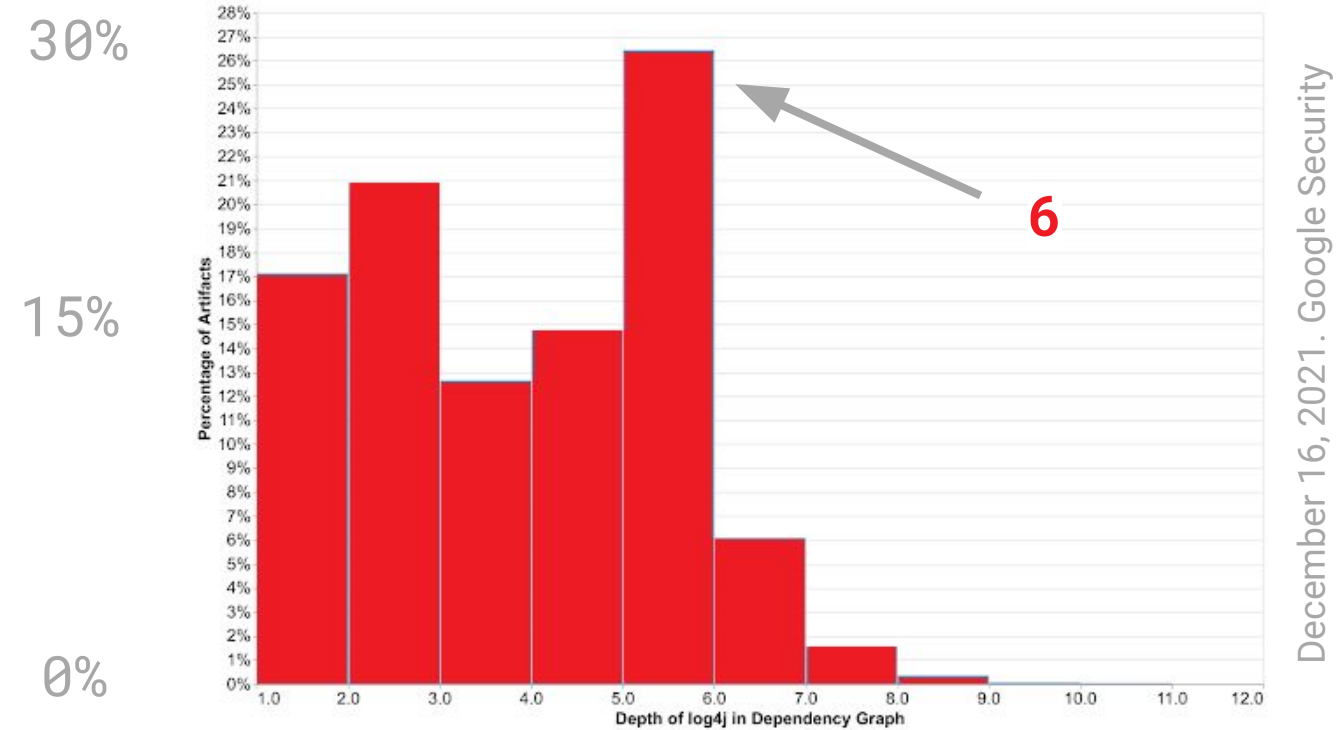
log4shell. А кто починит транзит?



December 16, 2021. Google Security

log4shell. Глубина проникновения

Процент артефактов



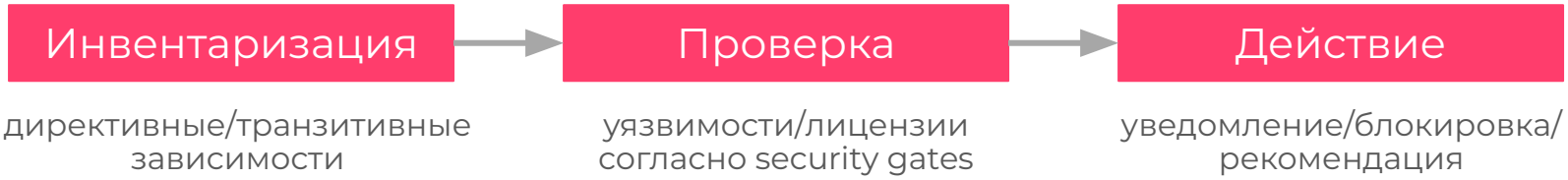
Глубина графа зависимостей

December 16, 2021. Google Security

Software Composition Analysis, SCA

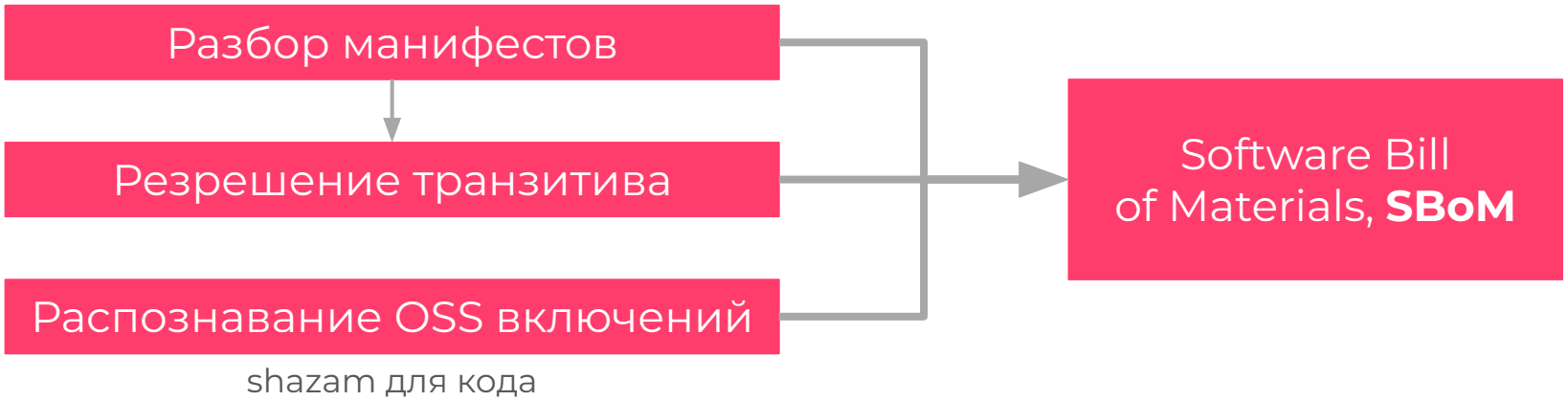
SCA — процесс определения компонентов, составляющих программное обеспечение.

SCA инструменты помогают в управлении рисками, связанными с безопасной разработкой.



Инвентаризация ПО

Найти манифесты, разобрать их, идентифицировать компоненты. А ещё бывают «включения» Open Source.



Как обычно применяется OSA/SCA?

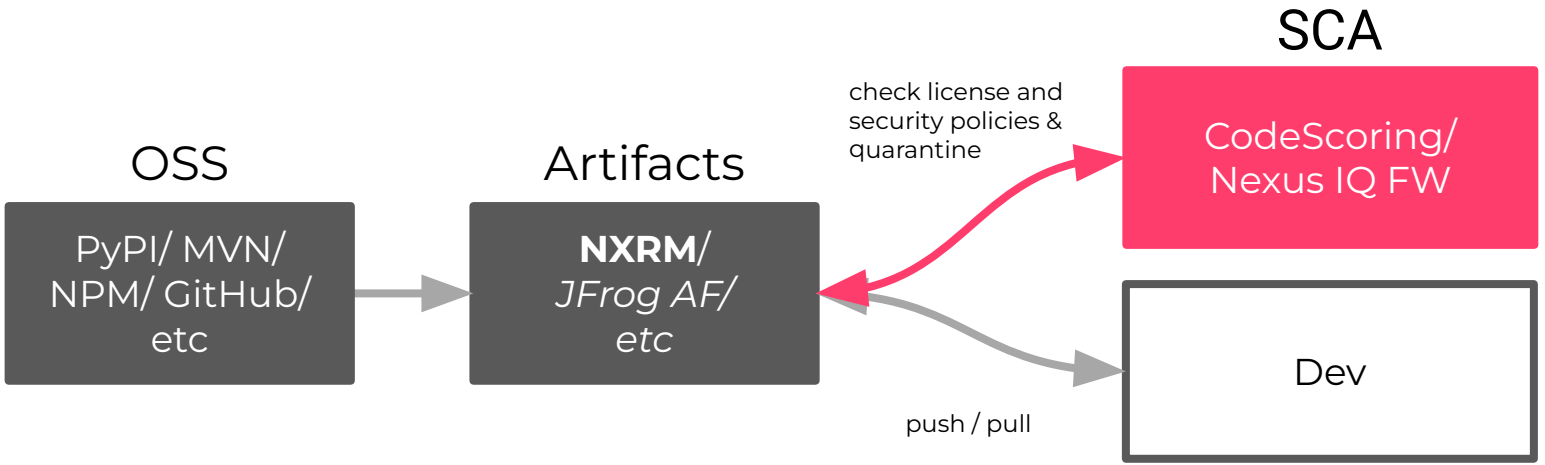
- **SCA Firewall**
  Блокирование нежелательных компонентов в прокси-репозиториях (хранилища артефактов)

- **CI/CD & security gates**
  Проверка и блокирование сборок в CI/CD pipelines (агент)

- **Continuous monitoring**
  Непрерывный мониторинг веток/тегов репозиториев (shift-left)

SCA.Firewall

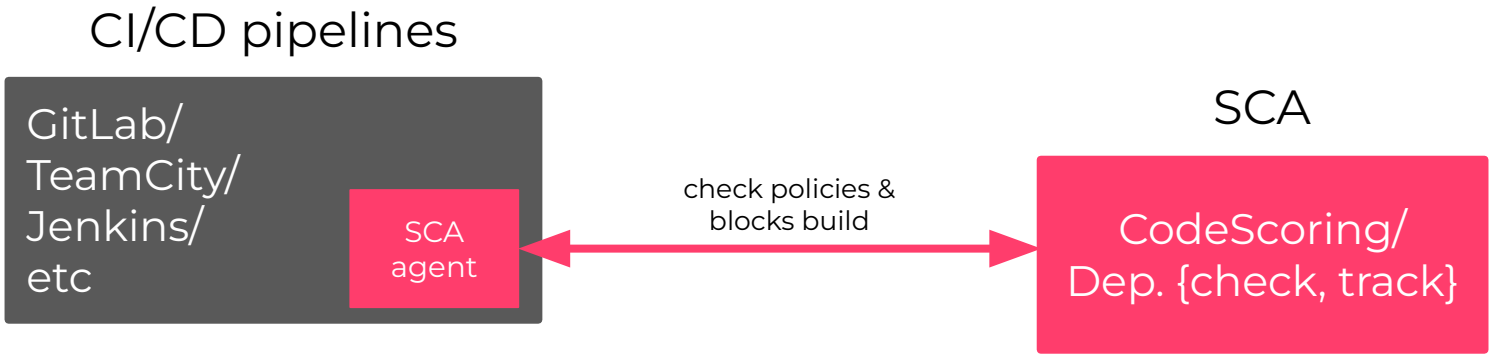
Блокирование нежелательных компонентов
в прокси-репозиториях (хранилища артефактов).



CodeScoring интегрируется с прокси-репозиториями через специальный плагин.

SCA.CI/CD

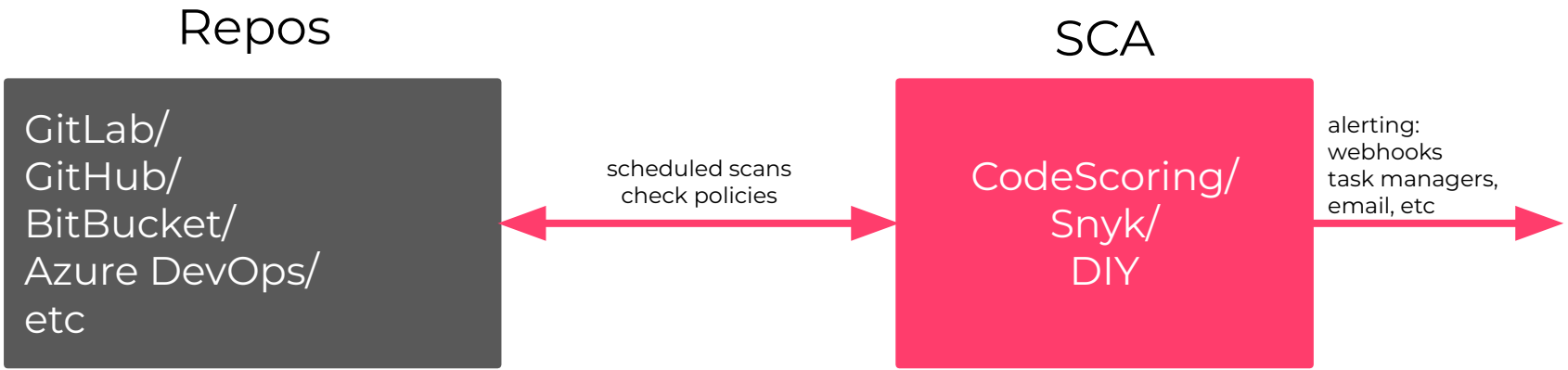
Проверка и блокирование сборок в CI/CD pipelines (агент).



CodeScoring интегрируется в CI/CD пайплайн через универсального агента.

SCA.Continuous monitoring

Непрерывный мониторинг веток/тегов репозиториев (shift-left).



CodeScoring отслеживает старый и новый код на появление новых рисков.

SCA — фундамент

Композиционный анализ дает нам основу и фундамент
для построения безопасных процессов разработки:

- безопасность компонентной базы, в первом приближении
- информированность об уязвимостях в фундаменте
- понимание лицензионной чистоты
- регулярный контроль ситуации
- автоматизация и понимание, что делать дальше

CodeScoring.SCA

CodeScoring.SCA

- инвентаризация компонентной базы программных продуктов (SBoM);
- поиск уязвимостей в Open Source компонентах;
- лицензионный ландшафт и лицензионная чистота;
- полноценное встраивание в жизненный цикл разработки (SDLC);
- режим непрерывного мониторинга угроз “от кода”;
- встраивание в CI/CD пайплайны;
- интеграция с прокси-репозиториями (хранилищами артефактов);
- гибкая настройка политик уведомлений и блокировки;
- риск-отчетность о компонентной базе и не только.

\+ Teams & Quality Intelligence (TQI)

Инвентаризация ПО

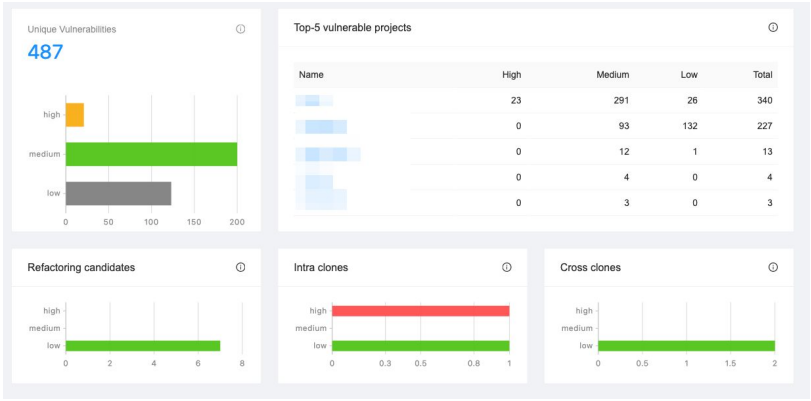
- Ведение реестра компонентных связей проектов (SBoM)

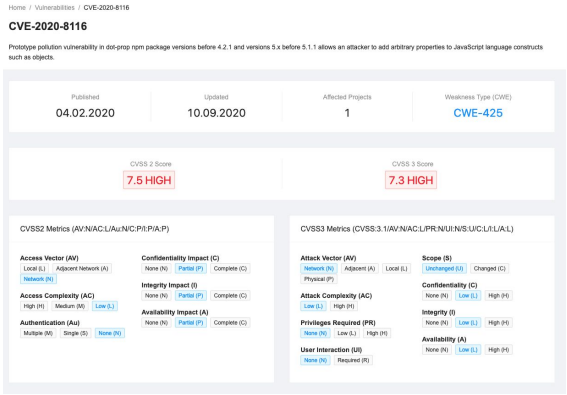
- Автообнаружение зависимостей
  - По файлам конфигураций (манифестам)
  - Разрешение транзитивных зависимостей
  - Включений OSS
  - По мета-данным (хэшам): md5, sha1, sha2

- Отчеты:
  - Списки проектов
  - Карта всех проектов
  - Компонентный состав всех проектов
  - Компонентная база проекта

Уязвимости

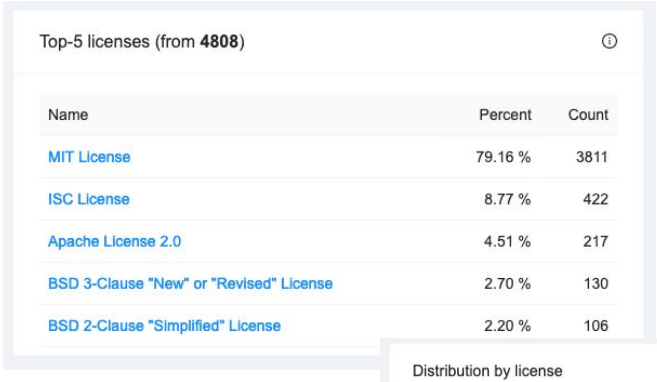
- Выявление уязвимостей:
  CVE, GHSA
- Рекомендации по исправлению
- Ведение базы уязвимостей
- Классификация**:** CVSS2, CVSS3
- Автоматическое обновление
- Отчеты:
  - Новые уязвимости
  - Уязвимости по проектам
  - Уязвимости по компании
  - Наиболее уязвимые проекты
  - Распределение уязвимостей по уровню риска

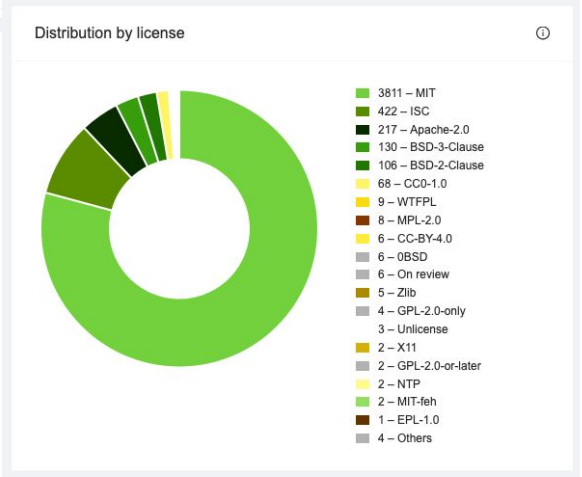




Лицензии

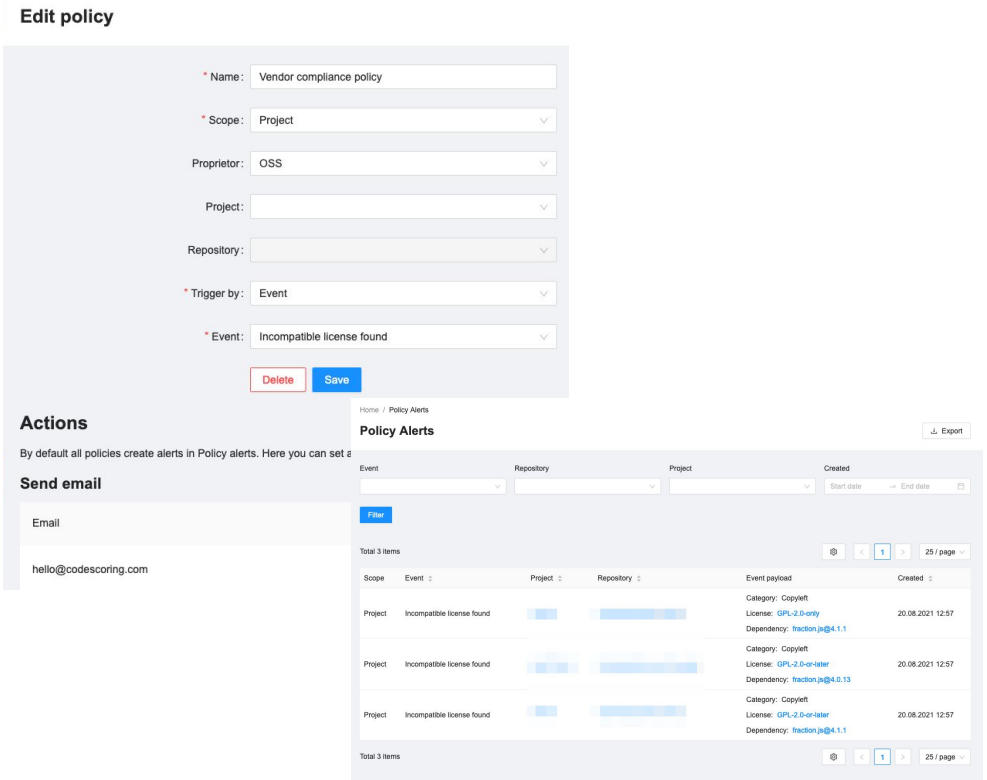
- Определение лицензий
- Лицензионная совместимость компонент (license compliance)
- Ведение реестра лицензий
- Автоматическое обновление
- Отчеты:
  - Лицензионный состав проекта
  - Лицензионный состав всех проектов
  - Нарушения политик лицензирования
  - Лицензии по группам
  - База лицензий





Политики уведомлений

- Иерархические политики
- Настройка по параметрам
- Логические объединения
- Отправка уведомлений
  на почту и в менеджер задач
- Блокировка сборки по событию
- inbox-механизм управления
- Выгрузка отчетности
  - нарушения политик по компании
  - нарушения политик по продукту
  - журнал сработки политик



Примеры политик

IF This	Then	That
• Нарушена лицензионная чистота (найдена неподходящая лицензия)	Тогда	Поставить задачу на замену библиотеки и уведомить юристов
• Найдена критическая уязвимость	Тогда	Заблокировать сборку, поставить задачу на третью линию
• Выпущена версия пакета после конкретной даты	Тогда	Поместить в карантин и поставить задачу на SAST-ревью
• Появился пакет из стоп-листа	Тогда	Поместить пакет в карантин
• Найдена уязвимость из OWASP Top Ten	Тогда	Эскалировать задачу

Поддерживаемые технологии

Языки программирования

- Java
- Kotlin
- Swift
- Objective-C
- JavaScript
- TypeScript
- Python
- Ruby
- PHP
- C
- C++
- C#
- Bash
- Go
- Perl
- PLSQL
- PGSQL
- Scala

Манифесты

- Maven
- Gradle
- NPM
- NuGet
- PyPi
- Go
- Ruby Gems
- Cocoapods
- Packagist
- CONAN

Полный перечень поддерживаемых технологий размещен по адресу: docs.codescoring.ru.

Интеграции в SDLC

Системы контроля версий (VCS)

- Github
  - Community
  - Enterprise
- Gitlab
  - Community
  - Enterprise
- Bitbucket
  - Server
- Azure DevOps
  - Cloud
  - Server

CI/CD конвейер

- Универсальный агент

Прокси-репозитории

- Nexus Repository Manager

Таск-менеджеры

- Jira

Почтовый сервер

Алгоритм

Алгоритм безопасной работы с OSS

Каждый может и должен построить правила для безопасной работы с компонентной базой. Хорошо, когда есть от чего оттолкнуться.


SBoM, Software bill of materials
Есть вероятность угрозы безопасности? (первая/новая/ странная версия или автор, уязвимости и т. п.)
да
Уязвимости известные?
да
Обновление доступно? (разумное)
да
Обновиться по сценарию
нет
нет
Ревью компоненты обнаружило проблемы?
да
Проверка на допустимость уязвимости
нет
да
Настроить непрерывный мониторинг версии ПО
Возможно исправить в коде?
да
Компенсирующие действия в ПО, написание автотестов
нет
Не используем + обновление политик
нет
Возможно исправить в окружении?
да
Компенсирующие меры (Configs, WAF и т. п.)
нет

Спасибо за внимание!

Алексей Смирнов,
основатель CodeScoring,
решения композиционного анализа (SCA)

alexey@codescoring.ru

@alsmirn — докладчик
@codescoring — CS-новости
@codemining — анализ кода