CODE SCORING

Контроль безопасности Open Source

от разработки до сопровождения ПО

Алексей Смирнов
Основатель CodeScoring

CodeScoring — платформа безопасной разработки

SCA
/композиционный анализ

OSA
/защита цепочки поставки

Secrets
/поиск секретов в коде с ML

TQI
/анализ качества кода

Российское решение на рынке с 2021 года. Запись в едином реестре ПО №13008

Чаще всего, ваш продукт выглядит так:

Известен
Идентифицируем
Исследуем

Неизвестен
Нужно сканировать
Сложно исследовать

Немного статистики про сторонний код

> 250 млн

проектов с открытым исходным кодом, а также:
8 млн. готовых пакетов;
100 млн. их версий

~90 млн

разработчиков хотя бы раз поучаствовали в разработке, а регулярно участвует ~6 млн.

protestware

саботаж и закладки
Пакеты: es5-ext, node-ipc, colors, faker, и ещё немного.
И Палестина.

x3

увеличилась скачиваемость открытых компонентов с 2023 на 2024

x13

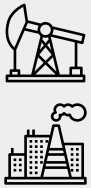
выросло количество известных атак на цепочки поставки: Dependency Confusion, Typosquatting, Namesquatting, Brandjacking, Malicious Code Injection и др.

> 900/мес

вредоносных пакетов выявляется экспертами по безопасности каждый месяц: кража параметров окружения, бэкдоры, шифровальщики и др.

Проблематика

Сырьевые и энергетические компании



Разрабатывают большое количество разнородного ПО для автоматизации.

Атаки могут привести к отказу критической инфраструтуры.

Важно обеспечить проверку ПО собственной и сторонней разработки.

Финансовые организации

Разрабатывают мобильные и веб-сервисы для физ. лиц и корпоративного сегмента.

Атаки могут привести к прямым финансовым потерям.

Важно внедрить культуру и контроль практики безопасной разработки.

Инвентаризация стороннего кода

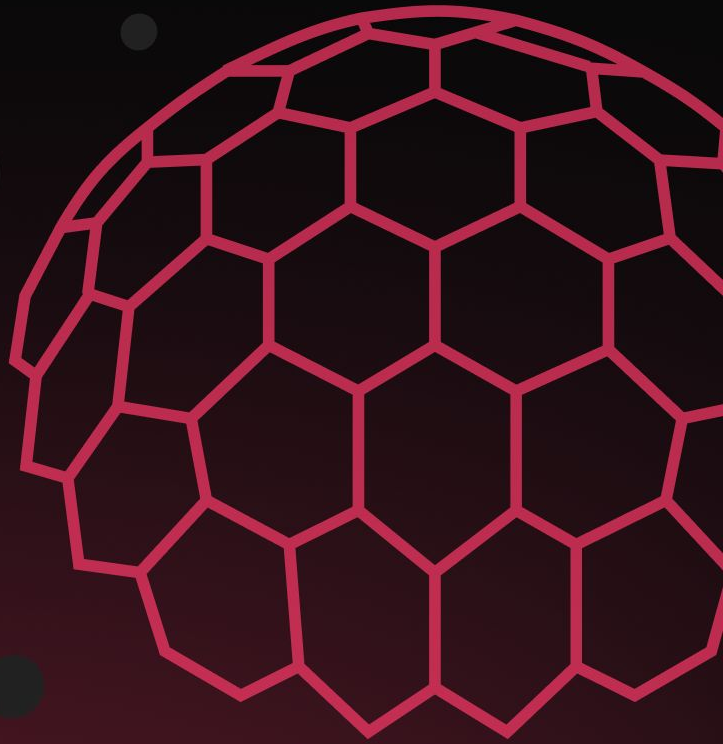
Перечень программных компонентов

/SBoM, software bill of materials

Машиночитаемый документ, содержащий в себе структурированную информацию о сторонних компонентах программного обеспечения и отношениях между ними

Что позволяет сделать SBoM?

- Фиксация единого компонентного состава версий программных продуктов
- Учет заимствованных **и привлекаемых** компонентов
- Фиксация информации об известных уязвимостях в сторонних компонентах
- Фиксация информации о лицензиях
- Отслеживание изменений в компонентах



Основные форматы SBoM

SWID
Software Identification Tags

- Становление: 2009 г.
- NIST
- Стандарт: ISO/IEC 19770-2
- Перевод: ГОСТ Р ИСО/МЭК 19770-2-2014
- больше про идентификаторы, чем про SBoM, ограниченная популярность
- nvd.nist.gov /products/swid

SPDX
The Software Package Data Exchange

- Становление: 2010 г.
- Linux Foundation
- Стандарт ISO/IEC 5926:2021
- Изначально ориентирован на работу с лицензиями, безопасность появилась позже
- spdx.dev

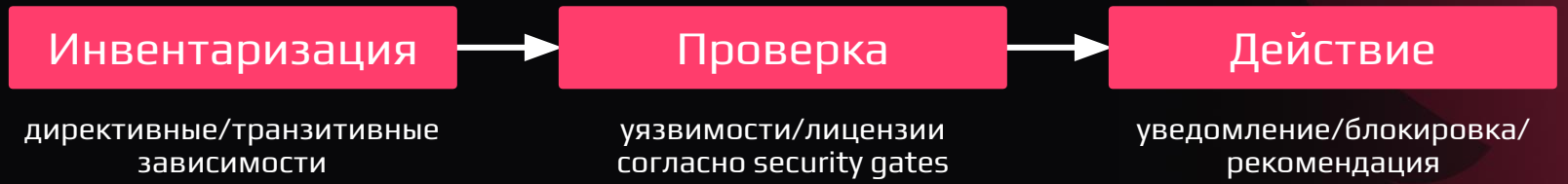
CycloneDX

- Становление: 2017 г.
- OWASP, ECMA (TC54), ТК362 ФСТЭК России
- Стандарт: на пути к ISO
- Есть развитие: SaaSBOM, CBOM, ML-BOM, OBOM, ...
- Учтены огрехи предшественников, поддержка в сообществе и среди инструментов
- cyclonedx.org

Композиционный анализ

Анализ, основанный на инвентаризации сторонних компонентов ПО, определении особенностей их использования, составлении перечня известных уязвимостей и/или иных недостатков компонентов. (англ. Software Composition Analysis, SCA)

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



Признание в России

Согласно ГОСТ 56939-2024,
композиционный анализ встает в один ряд
со статическим и динамическим анализами
в части практик безопасной разработки.

На завершающей стадии находится проект стандарта
«Композиционный анализ. Общие требования»

Где разработчик применяет проверки стороннего кода?

CODE SCORING

Написание кода

Проверка в редакторе кода (IDE)
Локальная проверка сборки

Сборка продукта

Проверка сборок в конвейере:
ночные, предрелизные, релизные

Пост-релизная проверка

Обнаружение вновь выявленных уязвимостей, выпуск исправлений

Внедрение SBoM

«Построить SBoM, вырастить SDL-политики, воспитать культуру безопасной разработки»
@IT IS Conf 2023

Как потребитель ПО может выполнять проверки?

Потребитель должен получать от поставщика информацию об используемых компонентах для контроля безопасности на своей стороне

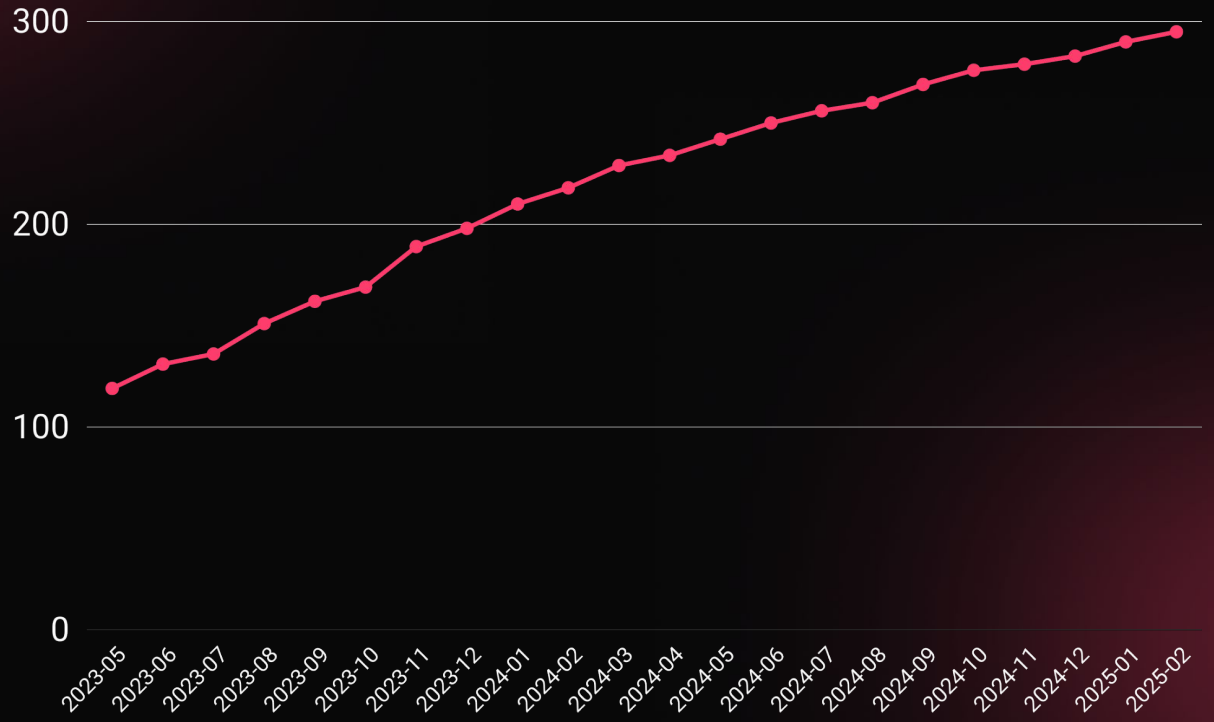
100

уязвимостей в год

накапливает ПО, если не выходят обновления безопасности

ib-bank.ru/rbpo_pubs

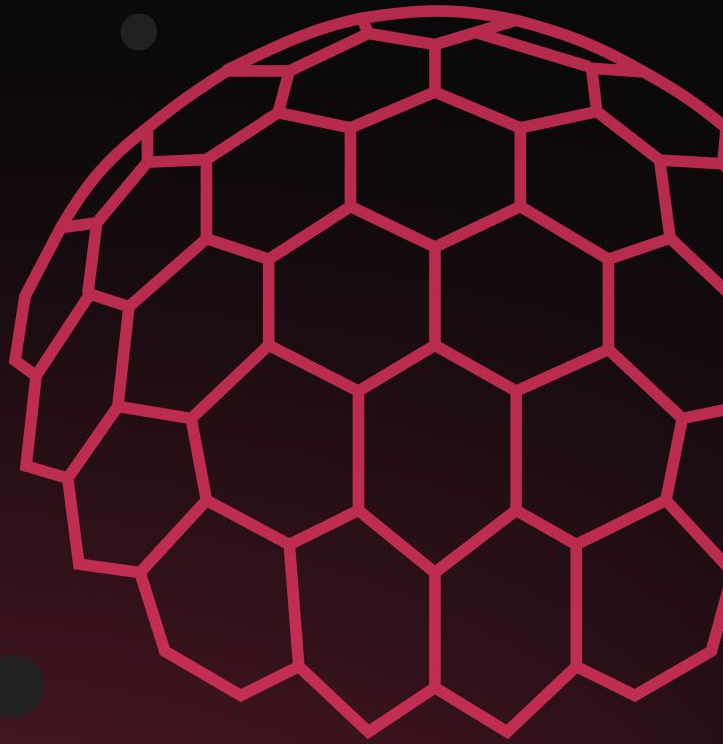
Пример: обнаружение уникальных уязвимостей в компонентах GitLab CE 15.11



Выводы

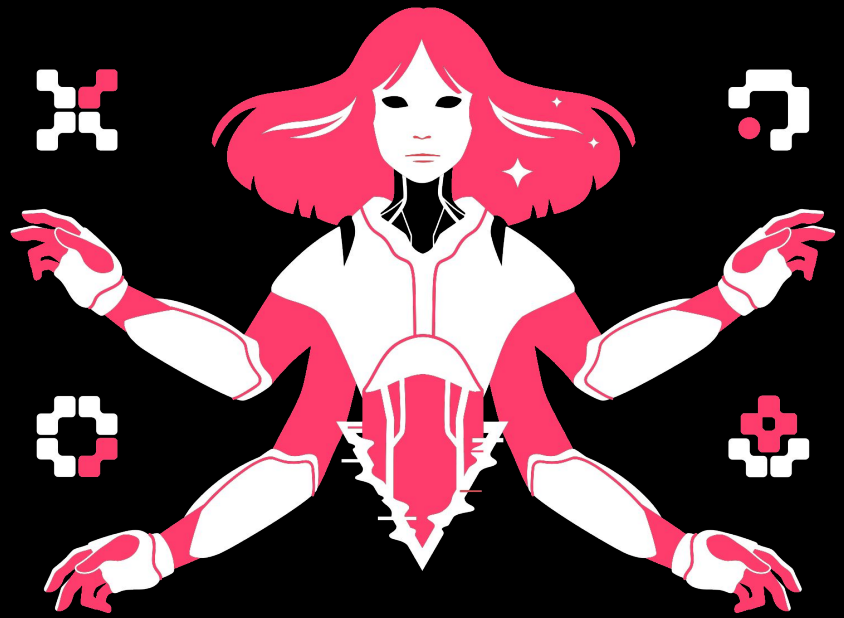
Польза для безопасности

- Знание своих и чужих продуктов
- Информированность об уязвимостях
- Понимание, что делать дальше
- Контроль ранее выпущенных продуктов
- Автоматизация процесса защиты цепочки поставки



Композиционный анализ

Поиск секретов в коде



Защита цепочки поставки

Анализ качества кода

alexey@codescoring.ru — контакт

@codescoring — новости продукта

@codemining — анализ кода

Образование:

youtube.com/@codescoring

vkvideo.ru/@codescoring

Полезные материалы

01 Композиционный анализ (SCA)

Проблема отцов и детей: аналитика и триаж транзитивных зависимостей, PHD'24

Построить SBoM, вырастить SDL- политики, воспитать культуру безопасной разработки, IT IS Conf'23

Protestware. Как много в этом слове! Devopsconf'22

02 Защита цепочки поставки (OSA)

Таксономия атак на цепочку поставки ПО: тренды и предпосылки новых трендов, GigaConf'24

Мифы и факты о цепочке поставки программного обеспечения, CyberCamp'23

PyPI сегодня — радости статистики и печали безопасности, PyCon'22