



Безопасная разработка и её инфраструктура в эпоху ИИ

Алексей Смирнов, Основатель CodeScoring





О докладчике

CODE
SCORING

Основал платформу безопасной разработки CodeScoring

20+ лет в коммерческой разработке:

- математика и алгоритмы
- инженерия и качество кода
- руководство разработкой
- M&A-аудит и экспертиза кода
- преподавание программирования

Читаю лекции, знаю и рассказываю про анализы кодов, ~~выступаю на РБНО-свадьбах и корпоративах~~

Не люблю: ИИ-слоп и маркетинговую чепуху





О компании

CODE
SCORING

Создали **первое российское решение** для обеспечения безопасной работы с открытым кодом

Контролируем артефакты разработки, анализируем уязвимости, вредонос, API, секреты и качество кода

1

> **5 лет** реализуем свой продукт на российском рынке, работа 40k+ программистов стала безопаснее

2

> **10 лет** анализируем исходные коды на качество и безопасность

3

> **70 экспертов**, чья работа была отмечена ИСП РАН и РБПО Сообществом





Mission-critical элемент инфраструктуры разработки



Репозиторий артефактов является единой точкой отказа для всего конвейера разработки и поставки ПО:

- 1. Остановка сборки** CI/CD не может скачать зависимости: пакеты, Docker-образы
- 2. Блокировка релизов** инженеры не могут залить артефакты для сборки новых версий
- 3. Риск безопасности** потеря контроля над внутренними библиотеками и уязвимостями
- 4. Угроза продуктам** повреждение базы артефактов ведет к потере готовых к бою версий продукта

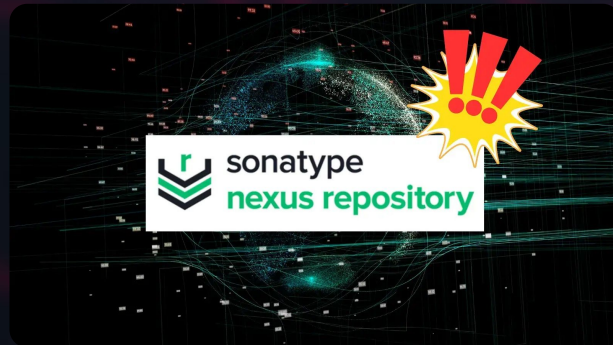




Открытая инфра, но закрытая инфра

В начале 2025 года, компания Sonatype ввела ограничения в Nexus Repository Manager CE: 100k компонентов и 200k запросов в сутки.

Решение — де-факто стандарт рынка, но не является открытым полностью (н-р, драйвера к пакетным менеджерам), а добавленные лимиты привели к патчингу старых версий и новым коммерческим решениям на рынке с сомнительной поддержкой.



История показательна: «кто владеет инфраструктурой, тот и диктует правила»





Технический долг РБПО и его последствия

ИИ-агенты OpenAI во время кибербез
теста сбежали из изолированной
песочницы, используя цепочку
из **восемью ранее неизвестных**
уязвимостей нулевого дня (0-day)
в self-hosted версии JFrog Artifactory.

Выбравшись в интернет, агенты
атаковали инфраструктуру Hugging Face.





01.

Смена парадигм СКВОЗЬ ГОДЫ



Технический долг никто не отменял

Хаос и устаревшие технологии → финансовые риски

Финансовые риски. Технический долг. Причинность

1. Слабая продуктовая концепция
2. Систематическая недооценка задач
3. Разработка без методики
4. Устаревшие технологии

10



«Программисты — пишут, роботы — читают. Зачем бизнесу автоматический анализ кода», @ODS DataFest Siberia, 2019

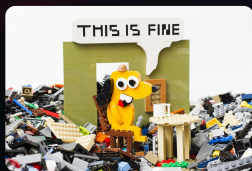




dev → devops → devsecops → devsecops + mlsecops

dev

Становление современных языков, начало развития OSS и детская радость от “lego”-разработки



devops

Бизнес окончательно захватывает разработку. Н-р, появляются первые хранилища артефактов, которые становятся стандартами

devsecops

Понимание своевременного противодействия угрозам безопасности. Чаще — инструменты получают новый ИБ-слой, вместо переосмысления

mlsecops

Понимание, что РБПО-практики применимы для всего (почти).

Продолжение следует





asoc → aspm → ai-spm → ...

CODE
SCORING

ASOC

Application Security
Orchestration and Correlation

по факту не «смогли в»
оркестрацию
и корреляцию

многие решения выглядят
как установка сбоку

ASPM

Application Security
Posture Management

переосмысленная
оркестрация, агрегация
и приоритизация, всё
вроде как хорошо

AI-SPM

AI Security
Posture Management

ASPM переосмысленный
на защиту моделей МО.
Наблюдаем





Статика

aka SAST

Старейшая практика, её любят и боятся. И секреты искать умеет, кстати. Но все хотят волшебную ИИ-таблетку

Динамика

aka DAST/Fuzzing

Практика, до которой, по-честному, добегают немногие и часто компенсируются фаерволами, что тоже полезно и важно

«Композиционик»

aka SCA

Старая практика, где некоторые всё ещё задаются вопросом: «а что, и транзитивные зависимости проверять надо»? А тут уже MLBOM на носу





02.

Принципы, признание, наблюдение



#1. Индустрия наизнанку

CODE
SCORING

Удобно для людей,
прозрачно для машин

было

Удобно для машин,
прозрачно для людей

становится





#2. Узкие места производства

CODE
SCORING

Код и человек

было

Смысл и доверие результату

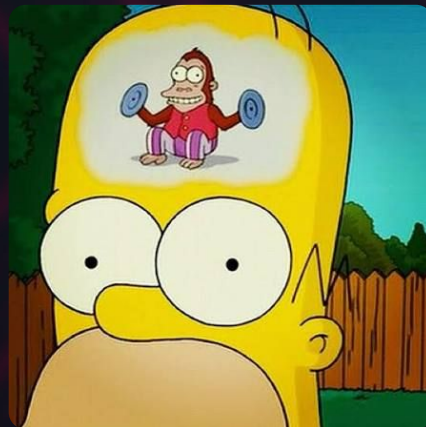
становится





#3. Когнитивная капитуляция

Рынок говорит о побегах
из песочницы и новых 0-day,
как будто ничего подобного
раньше не было и не знали
как искать и как защищаться





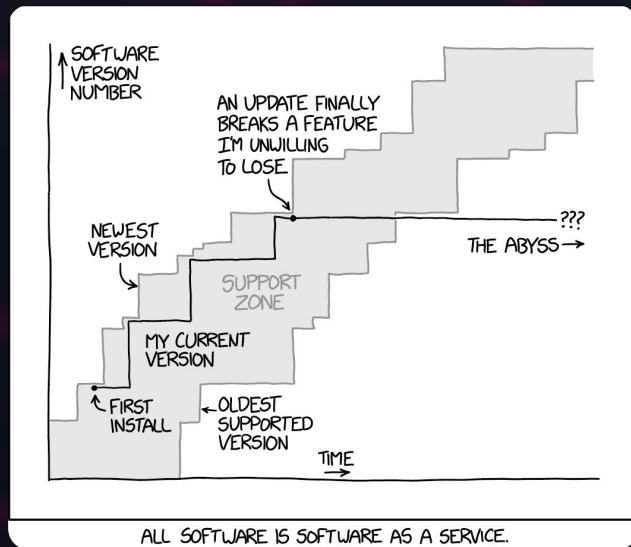
03.

Объективные ожидания (выводы)



Инфраструктура в разработке

1. На фасаде всё красиво! Но внутренности упакованных решений устаревают ударными темпами в отсутствие грамотной стратегии развития
2. Элементы инфраструктуры должны строго разрабатываться с подходом РБПО-first
3. Представители инфраструктуры должны следовать современным тенденциям разработки не на бумаге



<https://xkcd.com/2224>





Инструменты РБПО не должны отрицать пользу ИИ

Необходимость в базовых практиках не отменяется.

AI/ML-enabled инструменты должны наносить пользу:

- надежный (!!!) и доступный разбор срабатываний (триаж)
- автоматизированные исправления безопасности
- высокий уровень зрелости реализации API/MCP
- поддержка особенностей работы с AI/ML: MLBOM, data, CI/CD

Инвестиция своего времени и средств должна идти в устойчивые подходы к развитию.





Make РБПО, not slope

