CODE SCORING

Корпоративный репозиторий для кода и артефактов это что?

Не просто корпоративный, но и безопасный

Алексей Смирнов, основатель CodeScoring

О спикере

- основал **CodeScoring**
- **20+** лет в коммерческой разработке
- один из авторов ГОСТа по композиционному анализу в ТК362 ФСТЭК России
- делаем душевные конференции DUMP, PyCon, RustCon, LetsGoConf
- астроном, преподавал python 15 лет

могу быть полезным «про РБПО»,
безопасность Open Source, аудит ПО, анализ кода

cs.groteck.ru/IB_2_2025

Дисклеймер

Доклад о проверке содержимого репозиториев.
НЕ будем говорить о способах защиты самого репозитория от злоумышленных воздействий

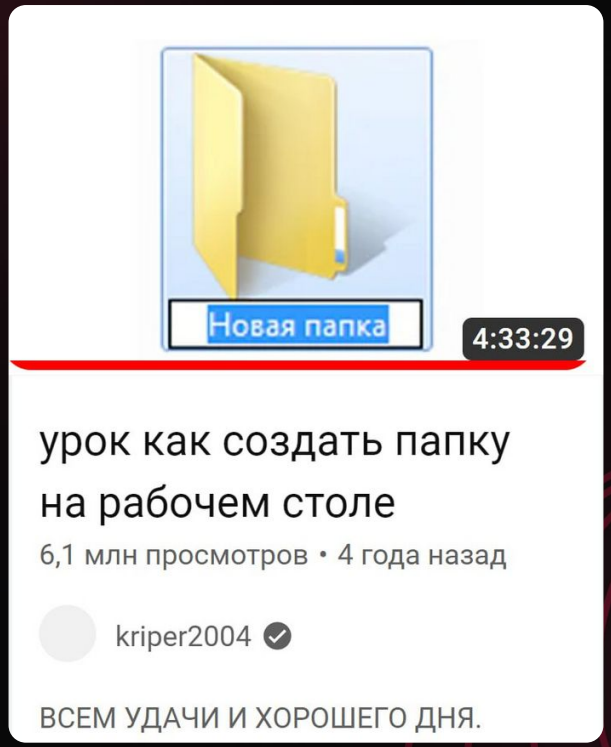
Репозиторий

Определение (одно из)

Репозиторий программных пакетов:
Замкнутая совокупность программных пакетов
и метаинформации о них. Репозиторий называется
замкнутым, если для каждого бинарного пакета можно
вычислить его замыкание, т. е. можно установить пакет
в систему с соблюдением всех его зависимостей.

(с) ГОСТ 54593—2011 «Свободное программное обеспечение»

Репозиторий, ты кто?

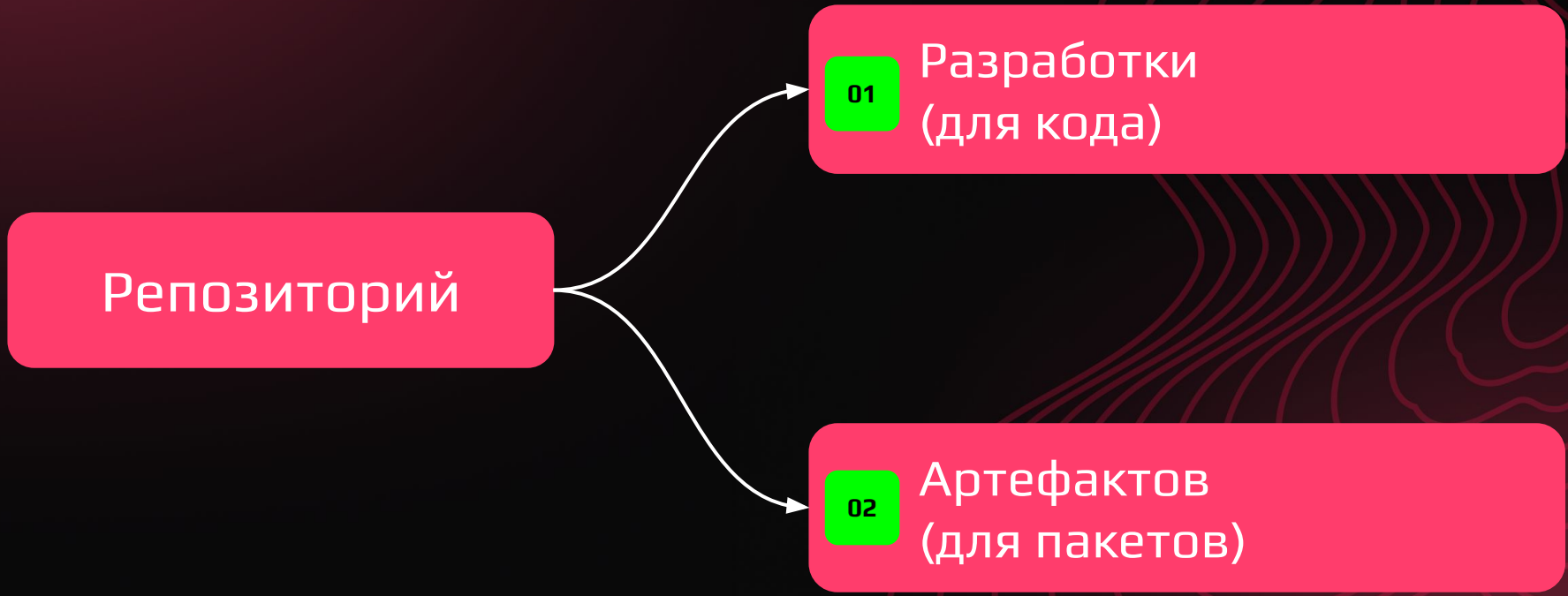


youtube.com/watch?v=5QZcOugHQ6s

Репозиторий, ты кто?

Папка с интерфейсами

Репозиторий для чего?



Безопасность

Квалификаторы безопасности

Зачастую эта терминология применяется к репозиторию артефактов, но также может быть применима и к репозиторию разработки

Безопасность чего?

Репозиторий разработки

?

Репозиторий пакетов

Всего, по возможности

Репозиторий разработки + Репозиторий пакетов

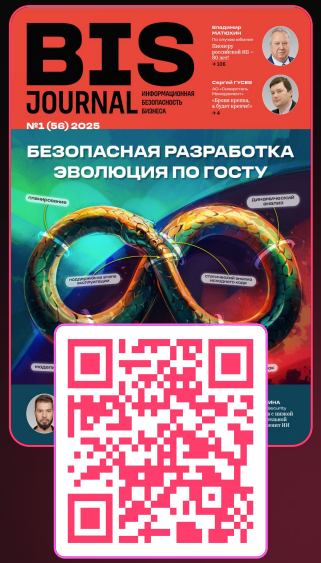
01 Чистый репозиторий

Сверхдоверенный репозиторий! Редкий вид

Но, уязвимости не появляются —
уязвимости обнаруживаются.

~100 уязвимостей в год «накапливает» ПО,
если не выходят обновления безопасности

ib-bank.ru/rbpo_pubs



02 Доверенный репо

Когда следует доверять по набору причин

Как правило, есть ответственный.

«А что такое ответственность?»

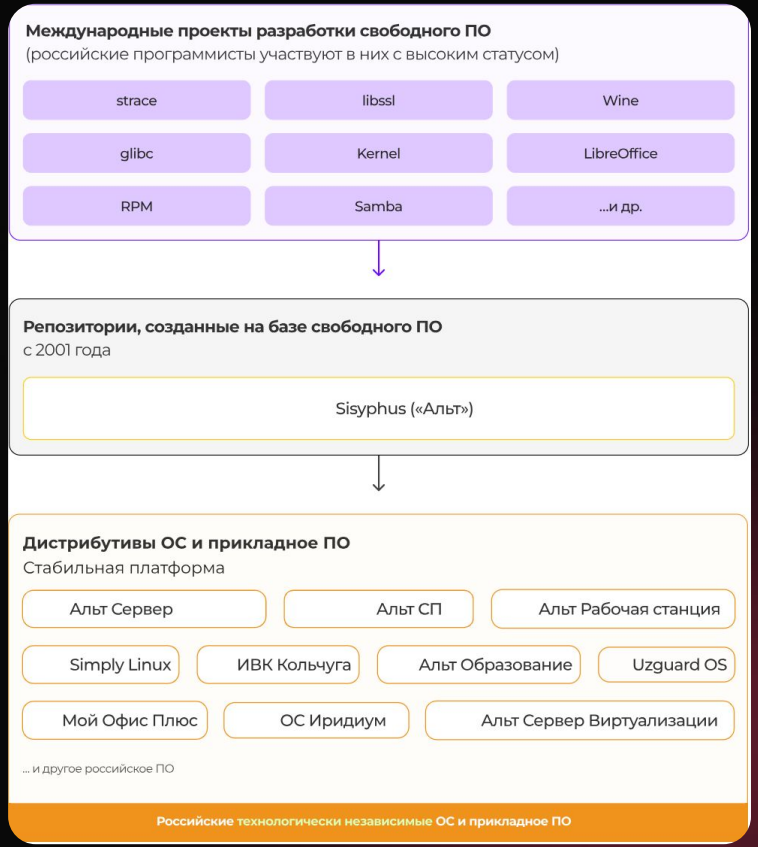
02

Пример: Сизиф

1. Старейший и крупнейший
   Участие для всех желающих

2. **Сборки и исходные коды связаны. Всё доступно**

3. **Поддержание стабильной ветки** от Базальт СПО: тестирование на безопасность и консистентность



02 Пример: ИСП РАН

Центр исследований безопасности системного ПО

Центр исследований безопасности системного ПО

Точка входа: https://gitlab.community.ispras.ru/cc-portal/intro

- .NET6 Runtime + ASP .NET Core
- OpenSSL
- Qemu + libvirt
- Podman
- Nginx
- Python3
- NodeJS
- Lua
- UEFI

В пилотном режиме:

- Redis
- ClickHouse
- Kubernetes
- Qt5
- PostgreSQL
- PHP
- chrony
- udisk
- Consul
- CUPS
- ActiveMQ
- ApacheDS

Основные принципы

- Репозиторий
  - Зеркалирование веток международного сообщества
  - Ветки с дополнительными исправлениями
    - в крайнем случае
    - на период эмбарго
- Настройка процессов автоматического анализа
  - статический анализ (SVACE)
  - фаззинг-тестирование (AFL++, LibFuzzer, ИСП Crusher)
  - функциональное тестирование
- Регрессионный анализ обновлений
- Систематический процесс экспертного анализа результатов
  - двухнедельные итерации
  - разметка предупреждений статического анализа
    + кросс-верификация
  - разбор падений
  - расширение покрытия функциональными и фаззинг-тестами
- Отслеживание уязвимостей, выявляемых в международном сообществе
- Подготовка рекомендаций по безопасному использованию ядра

02

Пример: AXIOM Repo

1. **Проверка и контроль пакетов** java-мира от экспертов и машин

2. **Сборки и исходные коды связаны**

3. **Техподдержка и SLA:** гарантии сроков; исправление уязвимостей; содержание форков, если авторы недоступны

03 Контролируемый репо

У ВАС применяются средства контроля

Средства анализа бывают разными:
статический (в т.ч. секреты), динамический (фаззинг),
композиционный (проверка стороннего кода),
антивирусный контроль и пр.

03 Контролируемый репо

Контроль кода репозиториев —
понятный свод инструментов

Базовые практики:

- статика, в т.ч. поиск секретов
- динамика и фаззинг
- композиционный (проверка стороннего кода)

03

Пример: контроль артефактов

1. Одного доверенного репозитория большинству не хватит
   Разработка разнообразнее, технологий много

2. Организация должна хранить пакеты из которых собирает продукты (желательно с исходными кодами)

3. Организация может применять доп. средства контроля забирая данные из открытых источников

03

Пример: контроль артефактов

CODE SCORING

Контейнерные образы

Языки программирования

Системные зависимости

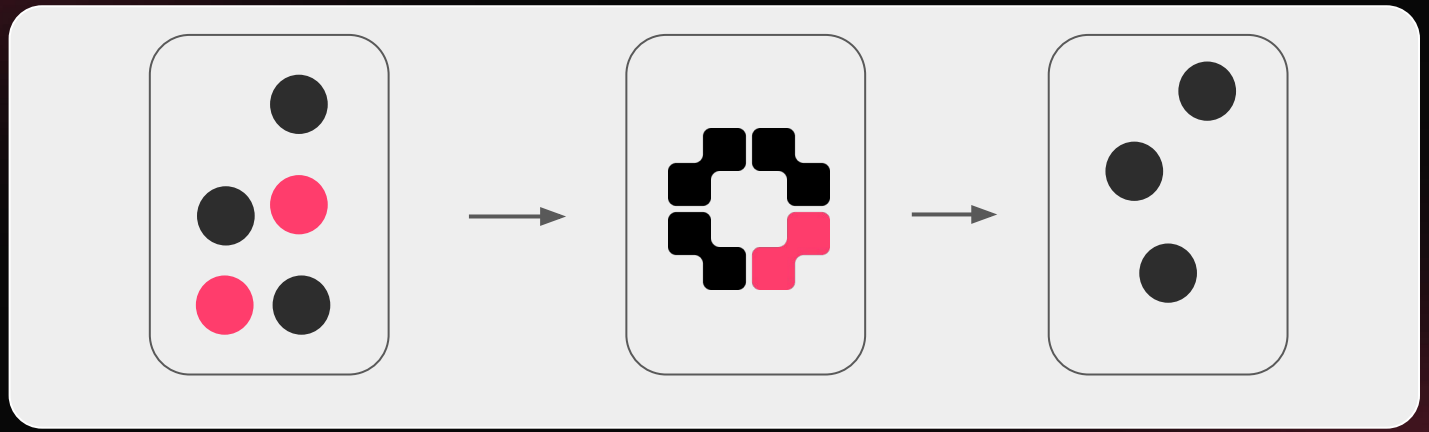
03

Пример: контроль артефактов

1. Хранение зоопарка
2. Ускорение сборок
3. Контроль безопасности со средствами защиты цепочки поставки + антивирь

03

Пример: контроль артефактов



Open Source

Проверка политик

Использование

CODE SCORING

Заключение

Комбинируем

1. Ориентируемся на доверенные репозитории
   Появляется больше гарантов
2. Организуем контролируемый репозиторий
   Включаем технологии и умения
3. Продолжаем мечтать о том, что наши репозитории будут самыми чистыми в мире

Полезные материалы

CODE SCORING

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

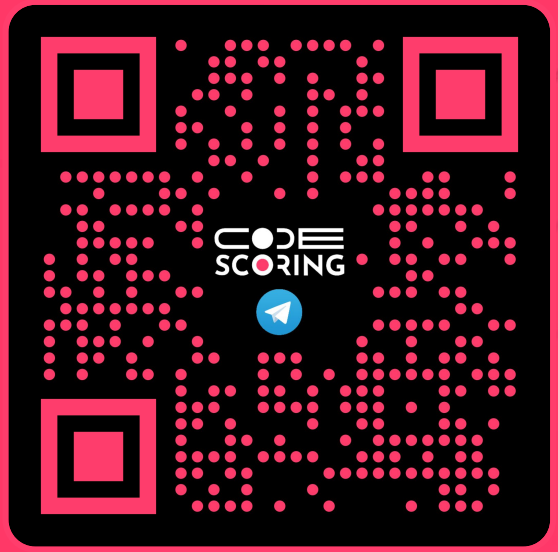
Protestware. Как много в этом слове! Devopsconf'22

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



CODE SCORING

alexey@codescoring.ru

Спасибо!