

Построение процесса безопасной разработки

от общего к частностям

Алексей Смирнов
Основатель CodeScoring



Платформа безопасной разработки

- OSA /защита цепочки поставки
- SCA /композиционный анализ
- Secrets /поиск секретов в коде с ML
- TQI /анализ качества кода

Немного про решение:

- 20+ интегрированных баз уязвимостей
- собственная база знаний про мировой open source
- интеграция на всех этапах разработки ПО
- широкий набор политик отслеживания и блокирования рисков
- защита цепочки поставки от популярных атак (+интеграция с Kaspersky)
- ключевая функциональность собственной разработки



Уже используют



- Банки и Финансовые компании
- Нефтяные и Энергетические компании
- Телеком и Медиа
- Государственные учреждения (госпорталы)
- Разработчики ПО

По итогам 22-24 гг. реализован ряд крупных проектов федерального уровня и мы непрерывно получаем обратную связь от заказчиков для регулярного улучшения продукта.



Безопасность цепочки
поставки программного
обеспечения



Проверка проектов при
приёмке заказного ПО



Анализ качества кода

Наш вклад в SDL-сообщество

- Экспертиза команды отмечена ФСТЭК России и ИСП РАН, ведется сотрудничество в части применения основной компетенции — анализ безопасности и качества сторонних компонентов.
- Членство в ТК362, руководство РГ7 по разработке проекта ГОСТа «Композиционный анализ программного обеспечения».
- Совместно с Фобос-ИТ организуем встречу SDL-сообщества в Петербурге на регулярной основе:
t.me/SDL_Community_SPb

GOOST
IN THE
SHELL

поддержка требований ГОСТ

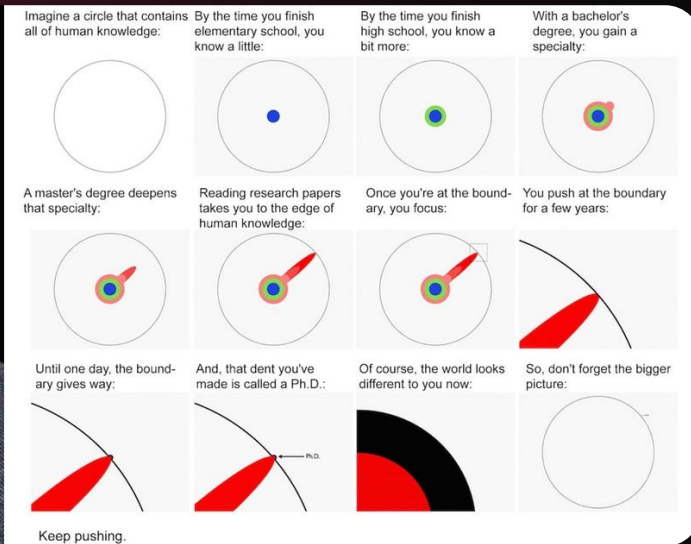
В предыдущих сериях: треугольник внедрения



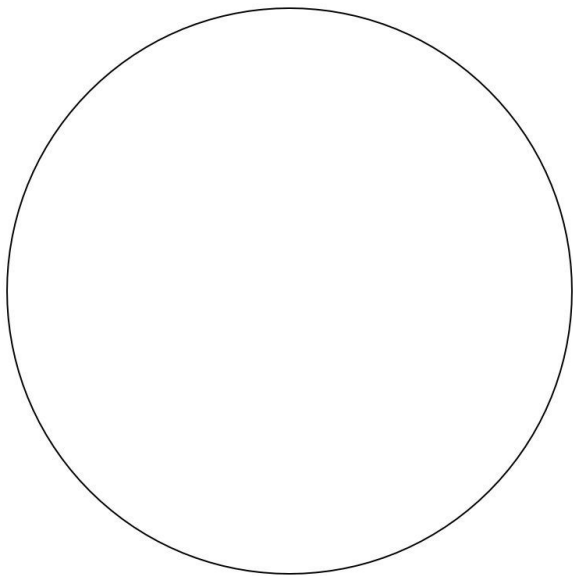
Эффективный композиционный
анализ: Путь к успешному внедрению
и упражнения на статику
ТБФорум'24

Общее и частности

Matt Might, The illustrated guide to a Ph.D.



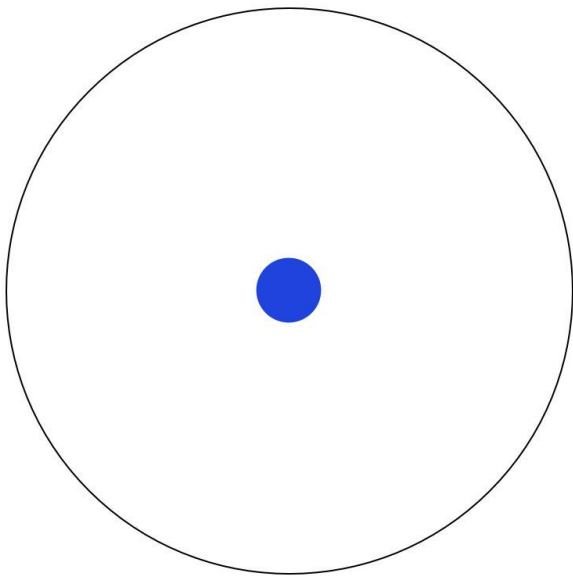
<https://matt.might.net/articles/phd-school-in-pictures>



**У вас есть ваше пространство
творчества программистов**

...и нужна безопасность разработки

добровольная



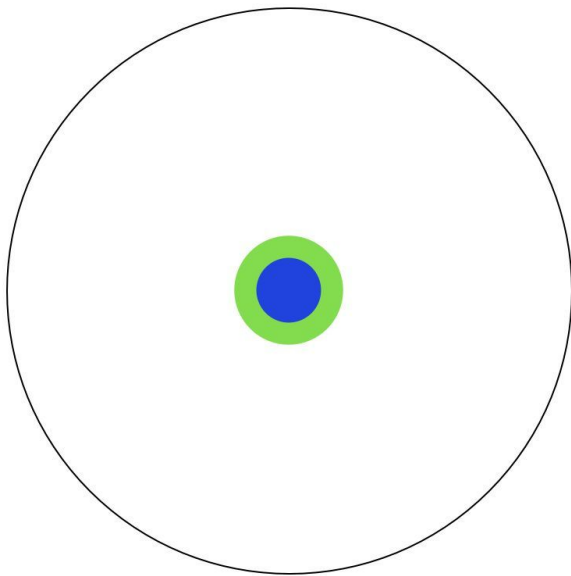
Всё начинается с людей

Ошибка здесь — проблемы потом
Особенно, на масштабах многих команд

Отнеситесь максимально внимательно
к этому шагу

Изучите вопрос, закажите
консультацию, почитайте стандарты
и практики

Сформируйте собственное мнение, кто
вам нужен (возможно у вас есть все)



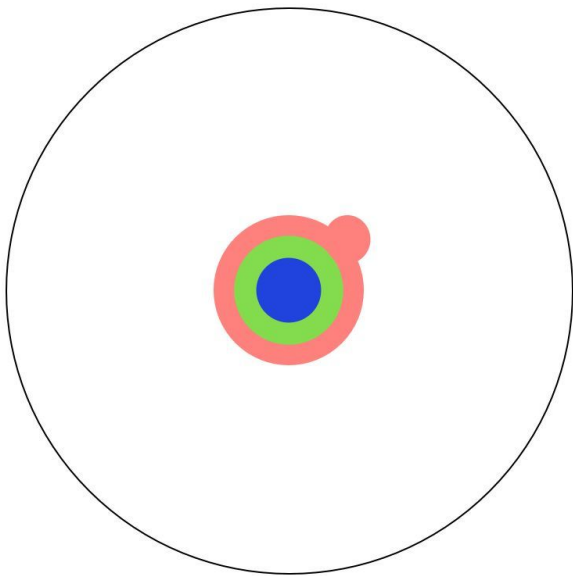
Людам нужны процессы

Построить РБПО не сложнее, чем построить хороший процесс разработки

Проблема в том, что экспертизы у вас несколько меньше

Подтягивайте экспертизу, учите людей.
Сегодня все не знают всего

Не беритесь за всё сразу, наслаивайте процессы и практики, следуйте вашим целям



Появляются инструменты

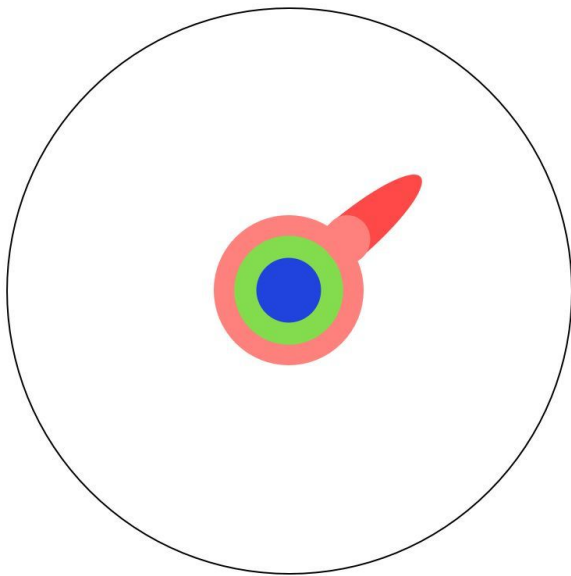
Открытые или коммерческие

Возникает **перекос экспертизы**

Появляются ответственные за практики,
разъезжаются льды

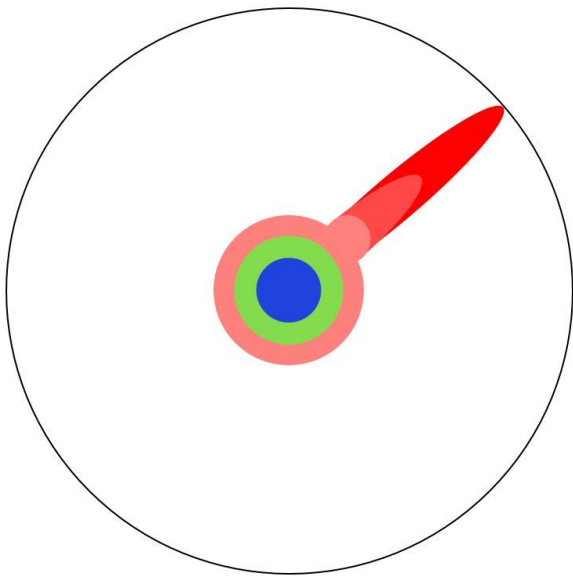
Сначала берите **понятные вам** практики:

- секреты
- статика
- композиционный анализ
- динамика



Боевое применение

Вы углубляете знания в практиках и вот уже контролируете Open Source (храните и проверяете), ищете ошибки в собственном коде, а возможно даже применяете подходы конструктивной кибербезопасности



Тотальный контроль

Практикуете изолированную сборку, собираетесь только из исходников, следите за поверхностью атаки, данными, конфигурациями и компиляторами. Контролируете рантайм и регулярно занимаетесь патч-менеджментом

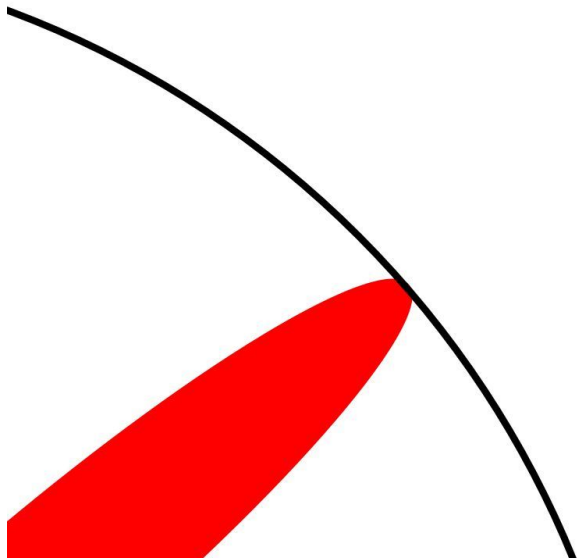
И вот вы всё сделали

Много чего поняли

Пересматриваете подходы

Оптимизируете ваши автоматизации

Думаете, как передать знания

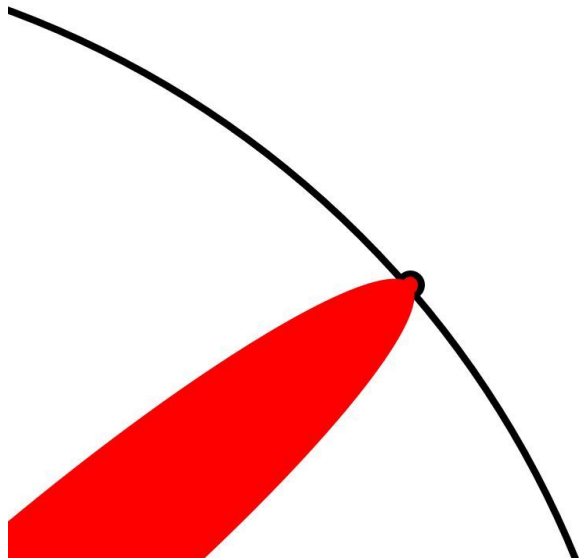


А ещё прошли сертификацию

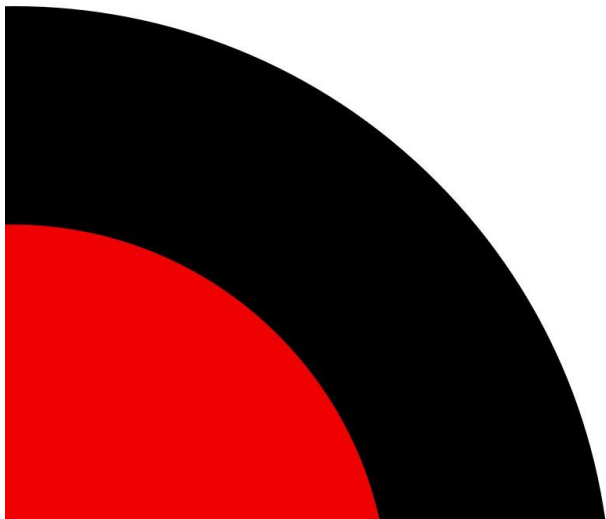
Средства или процесса

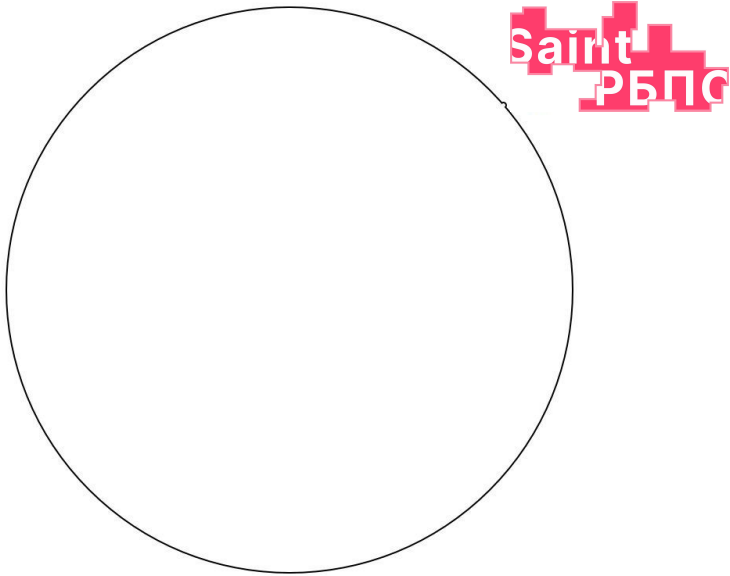
И возможно раскрыли новые
уязвимости в БДУ

И поделились лучшими практиками
с сообществом



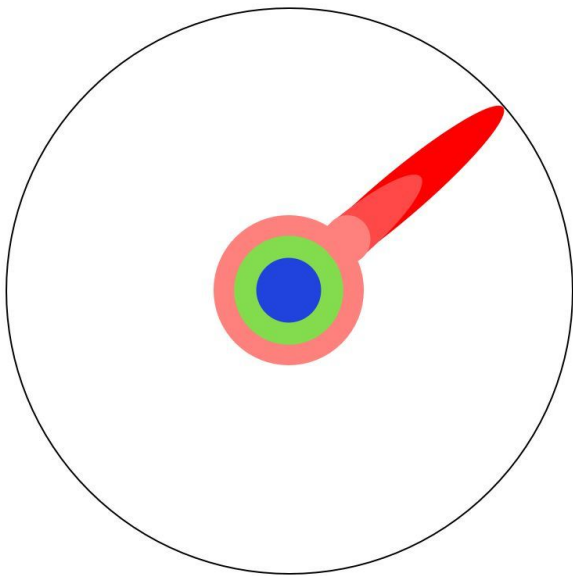
И вот вы на вершине





Но! Общая картина

Вы отодвинули границу (немного)



Кто заполнит остальные пробелы?

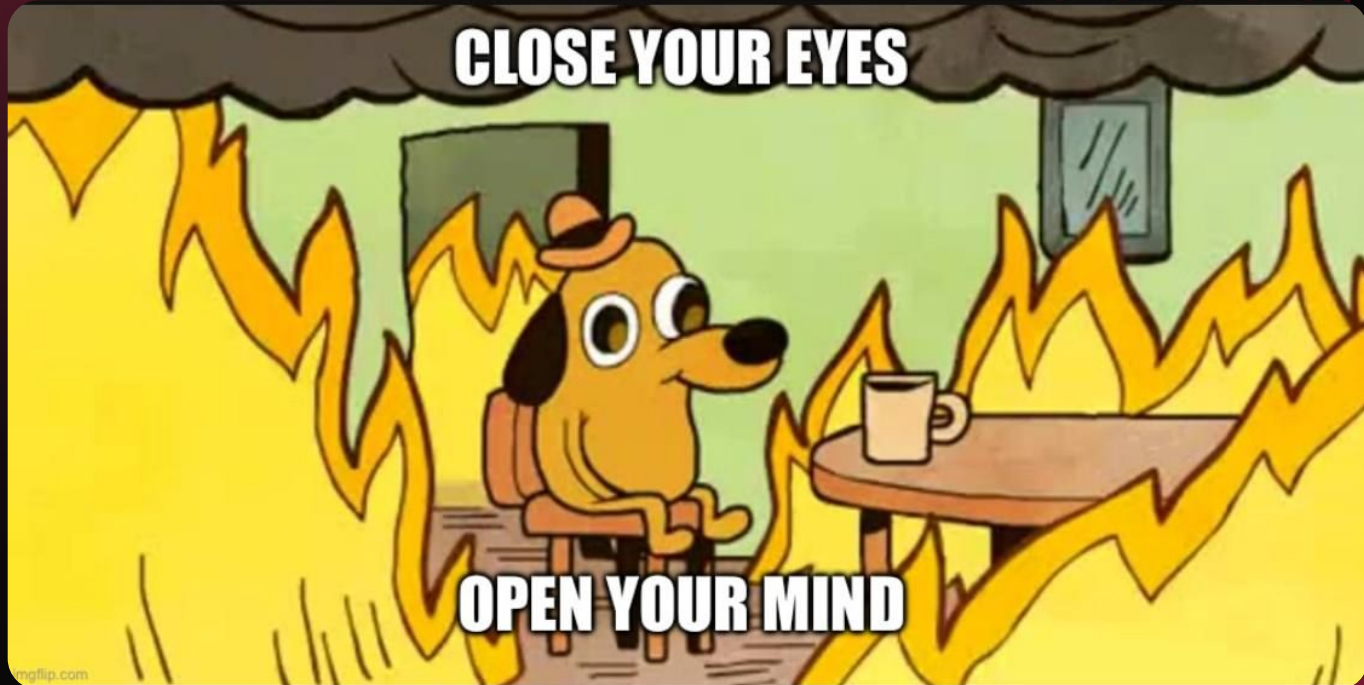
РБПО-служба крутая, но не всеильная,
важно распространять практику
в команды разработки

Да, security champions, инструменты
для разработчиков и иные подходы

Корпускулярно-волновой дуализм РБПО

Погнавшись за технологической крутостью — можно упустить главное — всеобщую грамотность в процессах.

Смотрите — программисты уже пишут код с роботами. А вы?





Спонсор медитационной паузы: наш фаззингист Никита Б.

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

Композиционный анализ

Поиск секретов в коде



Защита цепочки поставок

Анализ качества кода

alexey@codescoring.ru — контакт

[@codescoring](#) — новости продукта

[@codemining](#) — анализ кода

Образование:

youtube.com/@codescoring

vkvideo.ru/@codescoring