AppSec на выходных

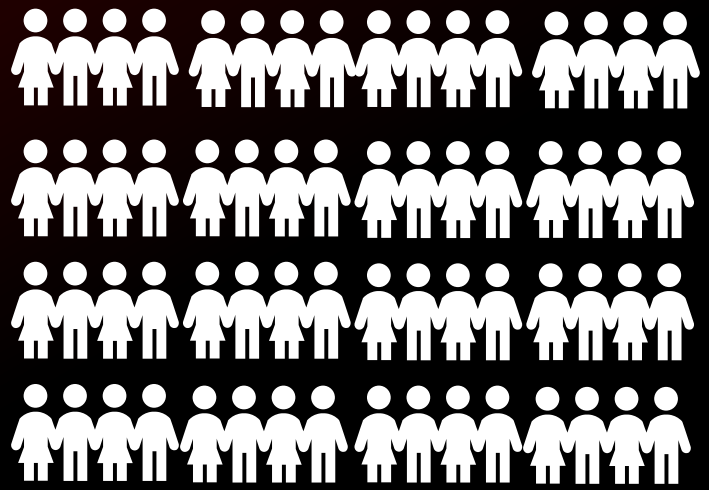
Газизова Светлана
Директор по построению процессов
безопасной разработки

Поворот не туда

Как не допустить ошибок, исправление которых будет стоить вам бизнеса?

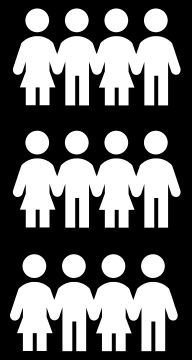
Что такое безопасность приложений?

Development



100+

Operations



10+

Cyber security

3+

Application Security

0+

Теперь понимаете, да?

Телеграм-канал
AppSec Journey

Wiki CyberOrda

О чем поговорим?

1. Где начинается и где заканчивается безопасная разработка?

2. Как дружить с бизнесом и не ссориться с разработкой?

3. С чего начать, чтобы не допустить фальстарта?

И правда, где?

Для начала...

Вас когда-нибудь взламывали?

Для начала...

Вы знаете, СКОЛЬКО
у вас приложений?

Для начала...

А сколько у вас разработчиков?

Кажется, мне даже не нужно вас убеждать WRITE TO US

С чего начнем?

С чего начнем?

> Стратегия

Почему мне нужна безопасная разработка?

Регулятор пришел?

Взломали?

Просто захотел?

Оценить масштабы бедствия!

Что вообще происходит в разработке?

Сколько команд?

Сколько репозиториев?

А сколько «живых»?

С чего начнем?

> Архитектура и план

Карта приложений

- Язык
- Численность команды
- Критичность приложения
- Количество пользователей
- Количество активных репозиториев
- Количество строк кода
- CI/CD
- Инструменты разработчика
- Бизнес-назначение приложения
- «Общий фон»

С чего начнем?

> Архитектура и план

Карта приложений

- Office Productivity
- Business Operational
- Business Critical
- Mission Critical

* Можно взять прям любую методологию, если нет своей. Главное, получить ранжированный список приложений по их важности для бизнеса

С чего начнем?

> Декомпозированная архитектура

ДПД, размещение инструментов и текущий AS IS

Разделение стейджей

Путь разработчика в репозиторий

С чего начнем?

> Документы

Политика + регламент:

Учесть

- Внедрение
- Сопровождение
- Триаж
- Управление рисками
- Хотфиксы

С чего начнем?

> Awarness / Обучение

☑ Рассказать про AppSec

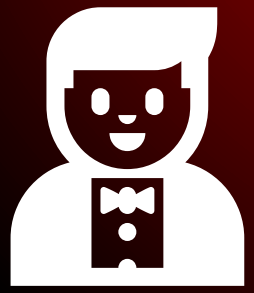
✕ Провести обучение

Уровень вовлечения	Краткое описание
1 уровень. **Базовый**	Минимальное обучение, без тренингов, без тестирования.
2 уровень. **Общая осведомленность**	Ежегодное + вводное обучение. Легкие тесты на социалку, разнообразные митапы.
3 уровень. **Безопасность в разработке**	Ежеквартальные семинары, тестирование после обучения. Упор на специфику приложений. Работа с мотивацией к безопасности.
4 уровень. **Безопасное поведение☺**	Обучение инструментам, свобода выбора решений для своих приложений. Общий уровень заинтересованности.
5 уровень. **Ценности в безопасности**	Программа на изменение культуры и тиражирование внутри команд. CTF + конкурсы с выигрышем. Транслирование ценности внутри компании.

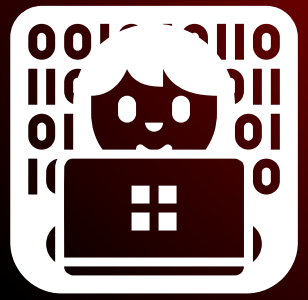
Рассказать про AppSec?

Действуем на разных уровнях!

C-level



Dev&Other



C-level

Убедить, помочь,
решить проблемы

Dev&Other

pt

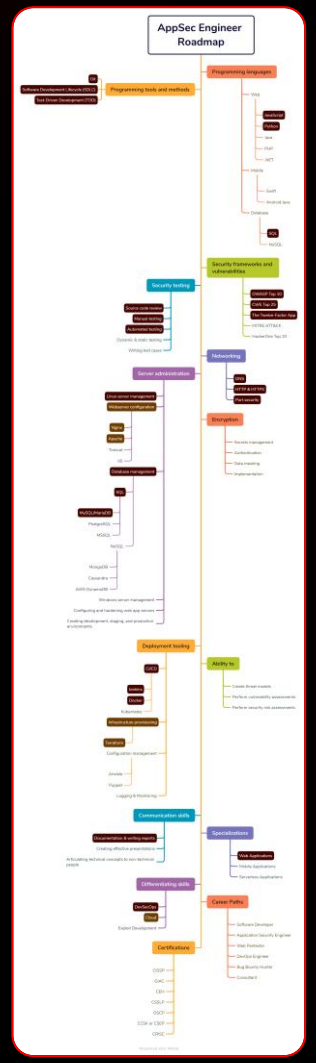
Повысить осведомленность, научить, вовлечь

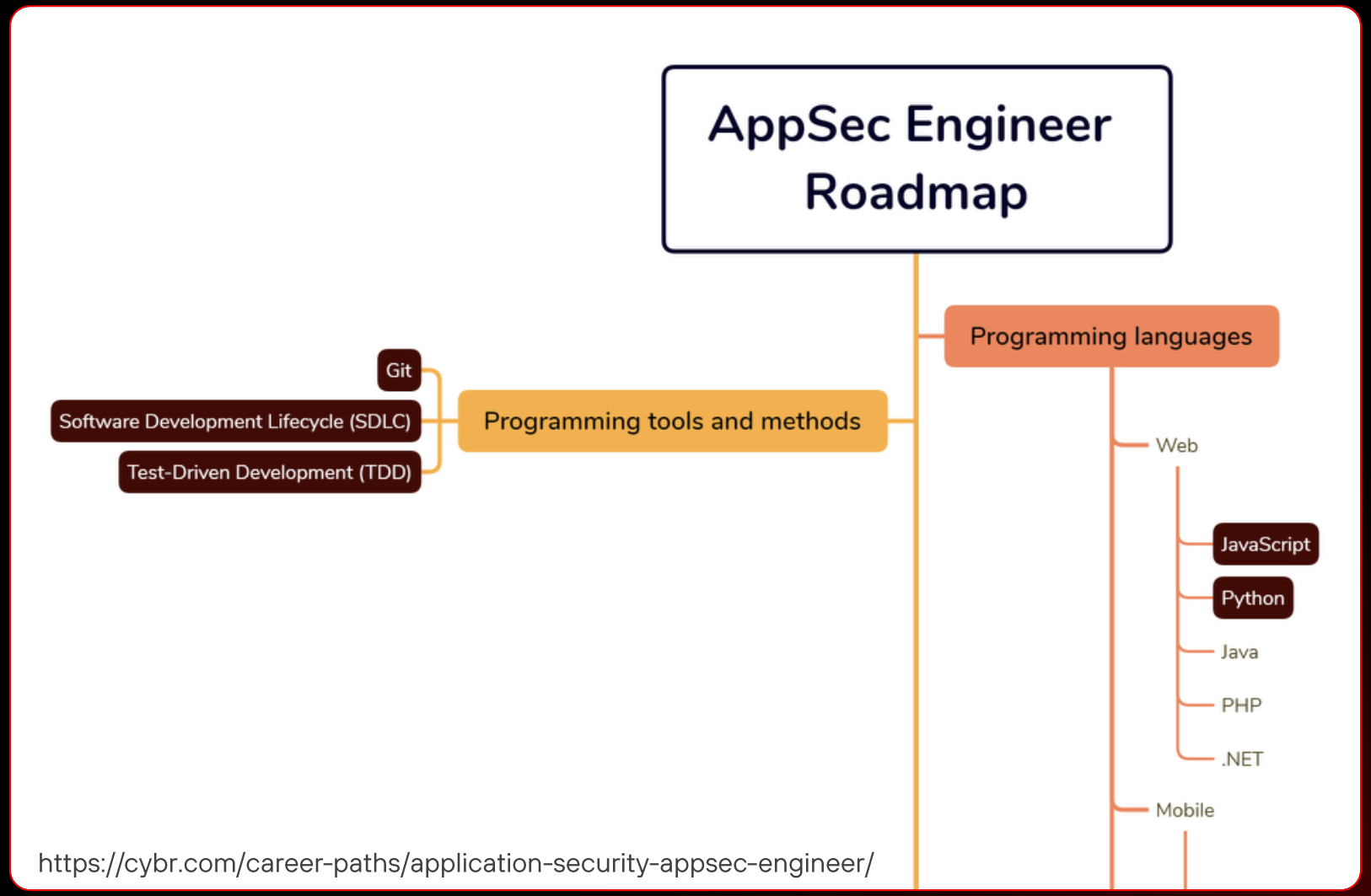
Обучение?

Наденьте маскусначала на себя...

Наденьте маскусначала на себя...

Хочу все знать!





Dev&

Security

App

DevOps

Code

PM

DevTeam

All

Dev&

Security

App

DevOps

Code

PM

DevTeam

Secure coding

Security

Ops

All

Dev&

Security

App

DevOps

Code

PM

DevTeam

Secure coding

Security

Ops

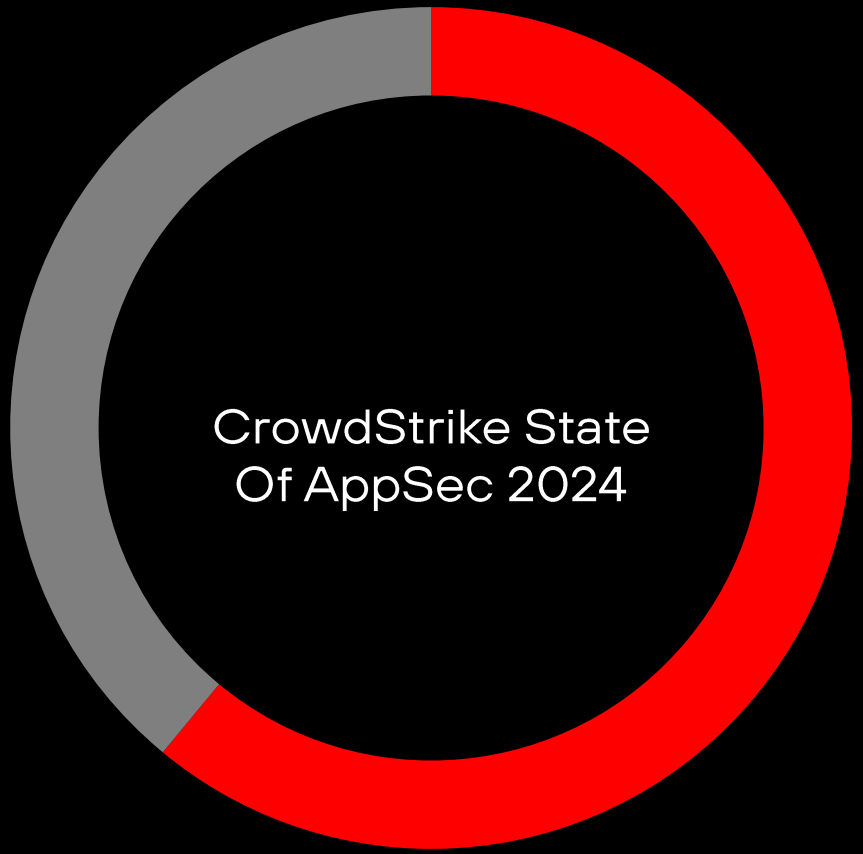
All

Cyber Hygiene

PR

Как внедрять контроли?

Application Security



61%

специалистов выделяют приоритезированное управление уязвимостями своей основной задачей

Какие контроли-то внедрять?

Какие контроли-то внедрять?

Организационно-распорядительные 😉

Политика, регламент, стандарт...

Какие контроли-то внедрять?

Организационно-распорядительные 😉

Политика, регламент, стандарт...

Какие контроли-то внедрять?

Организационно-распорядительные?

Политика, регламент, стандарт...

НЕТ!

Какие контроли-то внедрять?

- Адекватные рекомендации
- Архитектурные чек-листы
- Процессные карты

Какие контроли-то внедрять?

- [x] Адекватные рекомендации
- [x] Архитектурные чек-листы
- [x] Процессные карты

Какие контроли-то внедрять?

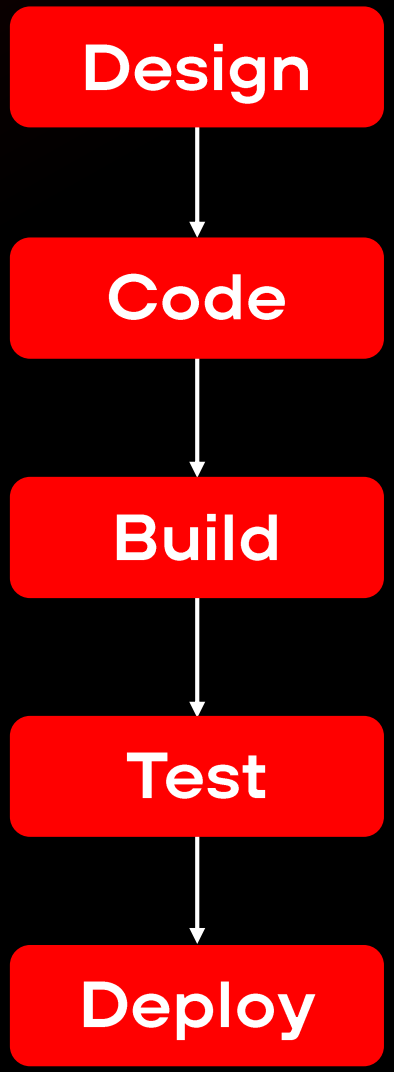
Инструментально-технические 😉

Какие контроли-то внедрять?

- [x] Quality Gates
- [x] Метрики и KPI
- [x] SLA и тайм-менеджмент

Итак, Quality Gates...

Стартовый вариант



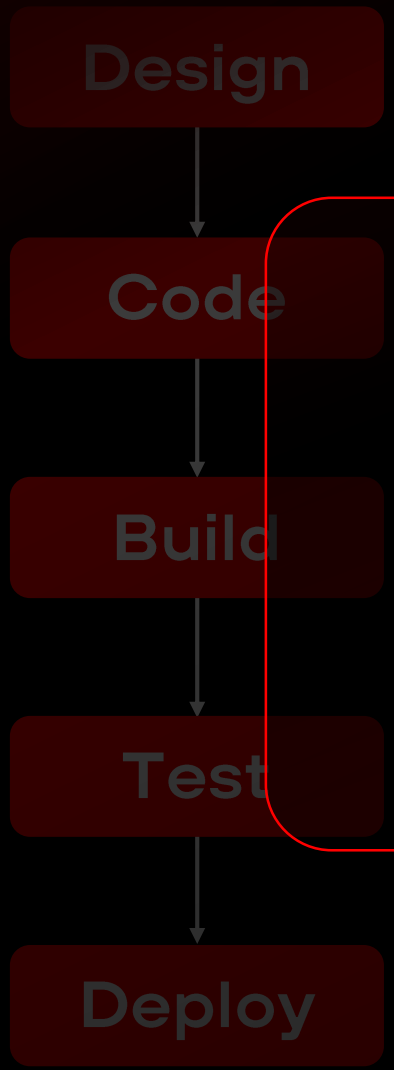
Смотрим архитектуру, советуем

Пропускаем все, что нашел SAST, оповещаем разработку и AppSec, храним централизованно

Компонентный анализ, накидываем экстренных патчей, если что-то совсем ненадежно

Все, что нашел DAST (и не FP) –исправляем.
Все, что нашел пентестер – исправляем.

Стартовый вариант



Смотрим архитектуру, советуем

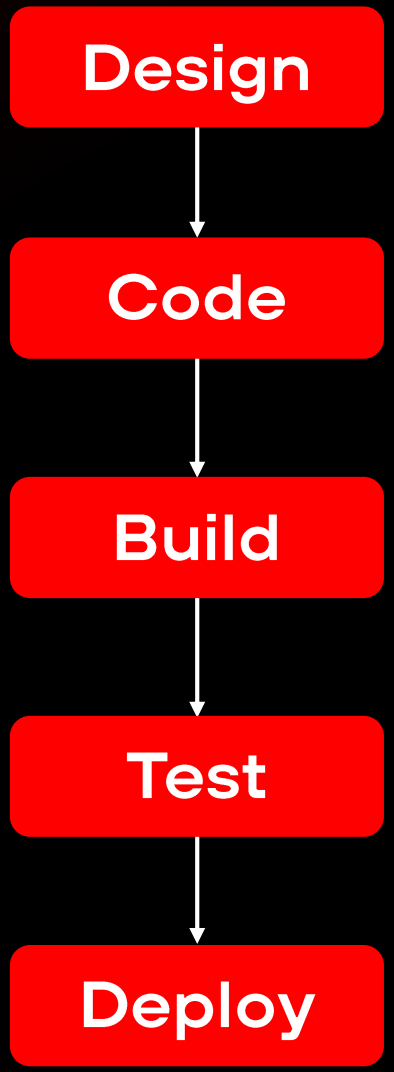
Пропускаем все, что нашел SAST, оповещаем разработку и AppSec, храним централизованно

Компонентный анализ, выявление уязвимых патчей, если что-то совсем ненадежно

Все, что нашел DAST (и не FP) –исправляем.
Все, что нашел пентестер – исправляем.

1-6 месяцев
+ обучение
разработчиков

Промежуточный вариант

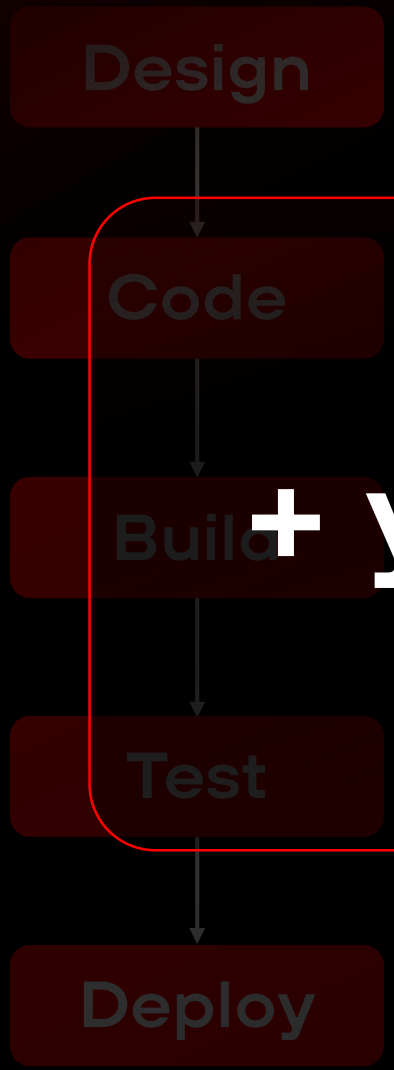


Даем базовую архитектуру сами: требования, чеклисты

Блокируем HIGH/CRITICAL,
потом разбираемся.
Дальше не идем!

Все, что нашел DAST (и не FP) – исправляем.
Все, что нашел пентестер – исправляем.

Стартовый вариант



Даем базовую архитектуру сами: требования, чеклисты

Блокируем HIGH/CRITICAL
потом разбираемся.
Дальше на идем!

Все, что нашел DAST (и не FP) – исправляем.
Все, что нашел пентестер – исправляем.

6-12 месяцев
+ углубленное обучение разработчиков

Цифры

	1-6m	6-12m	12-18m
OSA	0% - low 0% - medium 100% - high 100% - critical	10% - low 10% - medium 50% - high 100% - critical	100% - low 100% - medium 100% - high 100% - critical
SAST	0% - low 0% - medium 30% - high 70% - critical	20% - low 30% - medium 50% - high 100% - critical	100% - low 100% - medium 100% - high 100% - critical
SCA	0% - low 0% - medium 100% - high 100% - critical	10% - low 10% - medium 50% - high 100% - critical	100% - low 100% - medium 100% - high 100% - critical
DAST	100% - low 100% - medium 100% - high 100% - critical	100% - low 100% - medium 100% - high 100% - critical	100% - low 100% - medium 100% - high 100% - critical

Заранее подумать о:

- Управление критичностью приложений – правильный выбор
- Экономика и целесообразность – бизнес-ориентированность
- Команда и обучение – гарантированные соратники

Выводы?

Спасибо!