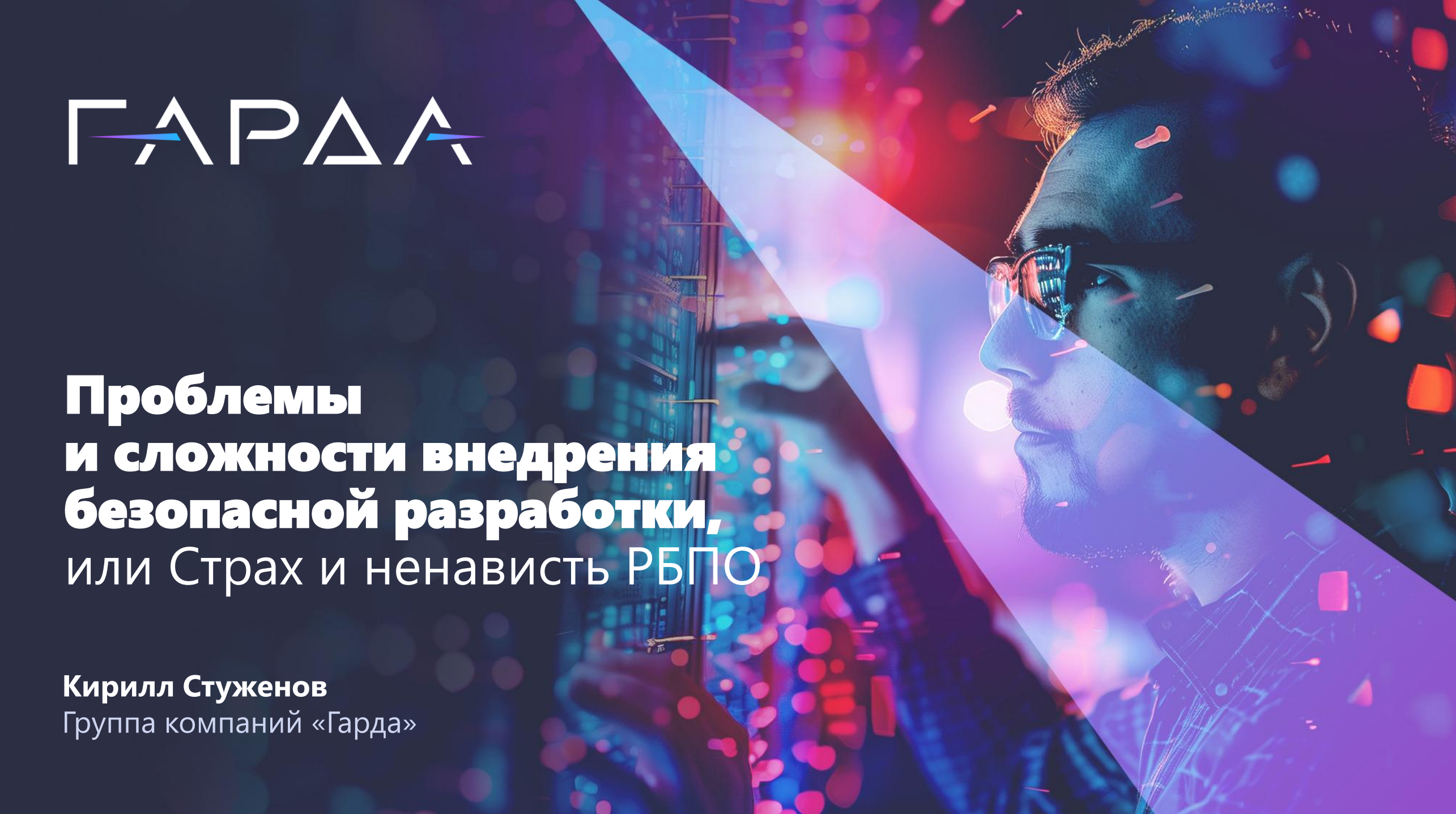

Проблемы и сложности внедрения безопасной разработки,
или Страх и ненависть РБПО

Кирилл Стуженов
Группа компаний «Гарда»

ГАРДА

Производитель решений в сфере безопасности данных и сетевой инфраструктуры

400+
клиентов

1000+
внедрений и интеграций

20+
продуктов

700+
сотрудников

5 офисов
в городах России

50% всего российского интернета защищены на базе систем «Гарда» от DDoS-атак

Решения «Гарда» обеспечивают защиту цифровых сервисов и мероприятий федерального масштаба*

500 млн персональных данных защищают решения «Гарда» в инфраструктуре заказчиков

Полностью российские решения: собственная технологическая платформа для хранения информации не требует сторонних лицензий, сертифицированы ФСТЭК

*Электронные выборы, Госуслуги, защита от подмены А-номера абонента

Системы безопасности

ГАРДА

Защита данных

- «**Гарда DBF**» — защита данных в БД.
- «**Гарда Маскирование**» — средство обезличивания статичных данных.
- «**Гарда DLP**» — защита данных от утечек, мониторинг активности сотрудников.
- «**Гарда DCAP**» — аудит файловых хранилищ и контроллеров домена.
- «**Гарда Аналитика**» — автоматизирует задачи СБ и экономической безопасности.

Сетевая безопасность

- «**Гарда NDR**» — выявление аномалий внутри сети и реагирование на угрозы.
- «**Гарда TI**» — обогащение СЗИ с помощью фидов и сигнатур.
- «**Гарда WAF**» — комплекс для защиты веб-ресурсов от кибератак.
- «**Гарда Deception**» — распределенная система ловушек, позволяющая запутать злоумышленника.
- «**Гарда Anti-DDoS**» — защита от внешних атак.

Управление трафиком

- «**Гарда DPI**» — глубокий анализ пакетного трафика.

Введение

ГАРДА

Интеграция безопасной разработки в процесс жизненного цикла

- Работа со своими страхами
- Работа с опасениями разработки
- У каждой организации свои процессы

Исходное состояние

Анализ того, что есть, и первые сложности

Задача

- [x] Проверка актуальности собранной информации

В итоге находим

- [x] безопасность учитывалась в основном на финальных этапах;
- [x] если инструменты анализа внедрены, то с ними никто не работает;
- [x] отсутствие общих четких стандартов;
- [x] и многое другое!

Внедрение статического анализа (SAST)

ГАРДА

1 шаг

До внедрения

- Сложность выбора.
- Встраивание в CI/CD.
- «Если работает, не трогай».

2 шаг

Внедрили, и тут начинается ненависть

- Много предупреждений от анализатора.
- Не все покрыто, что нужно, так что предупреждений еще больше.
- Пока мы решаем, что с этим делать, разработка пишет дальше, и их предупреждения растут.

Решение проблем статического анализа (SAST)

Замедление конвейера

- Запуск отдельной job'ы на отдельном сервере

Проблема большого числа предупреждений

- Настройка правил анализатора до необходимого уровня чувствительности
- Проведение обучения
- Остановка роста предупреждений и уменьшение их от релиза к релизу
- Оценка предупреждений и расстановка правильной критичности

Обнаружить ошибки недостаточно

Чем больше багов, тем важнее оценивать их критичность и приоритет исправления

Фаззинг–тестирование — это просто или нет?

Выбор инструментов

- Самый простой вариант:

```
./myapp < /dev/random
```

- Лучше использовать фаззер с инструментацией:

```
CXX=afl-clang-fast++ AFL_USE_ASAN=1 make
afl-fuzz -i in -o out -- ./myapp
```

- Можно увеличить число потоков, добавить санитайзеры:

```
CXX=afl-clang-fast++ AFL_USE_ASAN=1 AFL_USE_UBSAN=1 make
afl-fuzz -i in -o out -M m1 -- ./myapp
afl-fuzz -i in -o out -S s2 -- ./myapp
afl-fuzz -i in -o out -S s3 -- ./myapp
```

Кроме того

- Работа над ошибками
- Все выходит из-под контроля
- Мыслить глобально
- Bugbane
- Оно никогда не будет стабильно
- Примите хаос и неопределенность

Возможные проблемы в процессе фаззинга и их решение

ГАРДА

Появление дополнительных взаимодействий	Security Champion
Внедрение инструментов для разных технологий отнимает время	DevSecOps и внутренняя Wiki с хорошо описанными правилами настройки инструментов
Большое количество результатов работы инструментов	AppSec показывает разработчикам только подтвержденные проблемы
Трата ресурсов на исправление несущественных ошибок	Больше внимания уделять оценке критичности

Внедрение SCA

ГАРДА

Проблематика

- Большой объем зависимостей — сложность управления
- Частое появление новых уязвимостей — необходимо постоянно отслеживать изменения
- Недостаток знаний

Внедрение SCA

ГАРДА

 Большой объем зависимостей — сложность управления	Анализ на возможность эксплуатации, ведение разметки
 Частое появление новых уязвимостей — необходимо постоянно отслеживать изменения	Регулярное отслеживание и последующий анализ
 Недостаток знаний	Ведение собственной Wiki с описанием зависимостей и как используется

GARDA

Офис в Москве
улица Новодмитровская, дом 2Б
8 800 770 70 60

garda.ai
info@garda.ai

Спасибо за внимание!