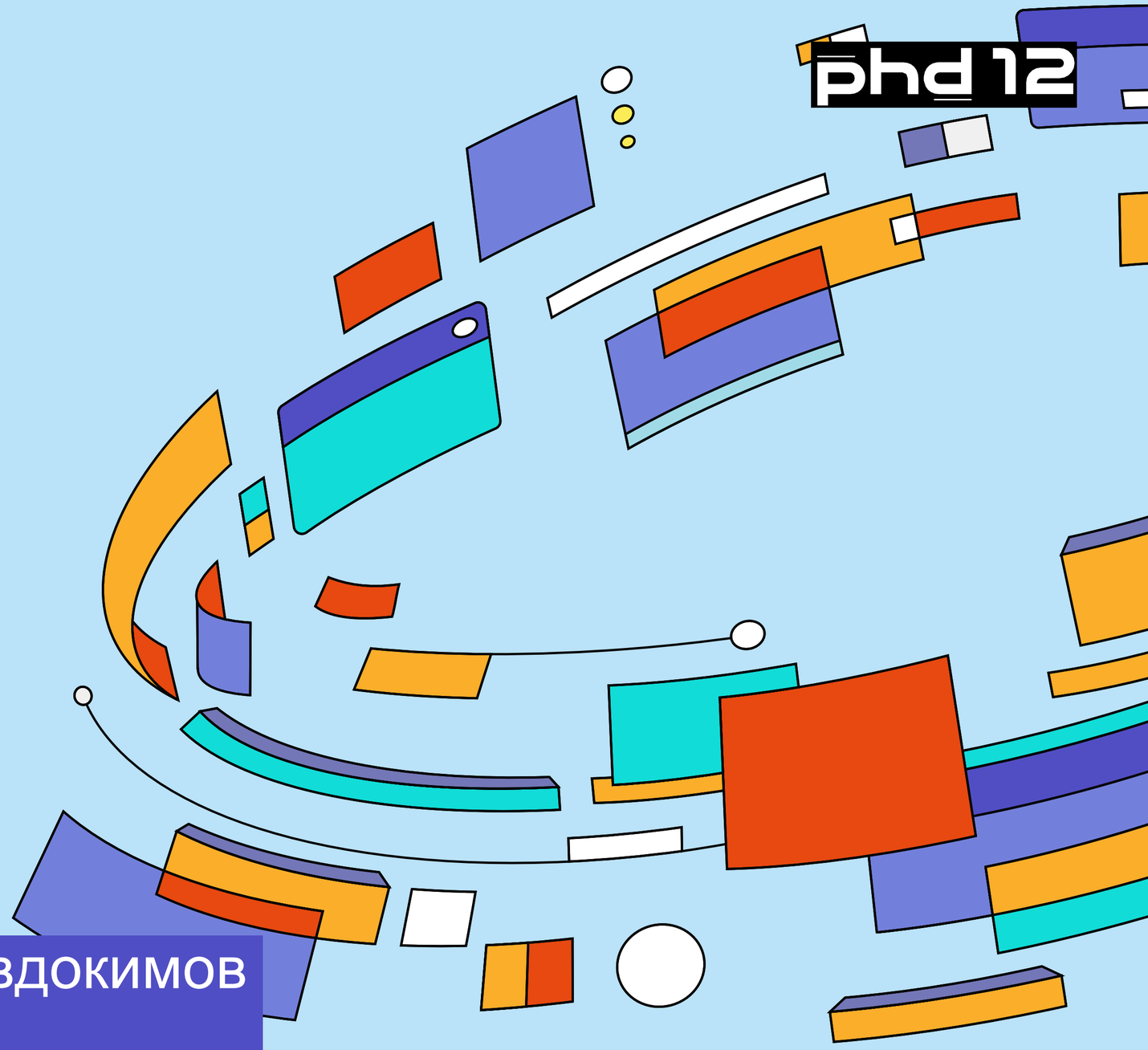
SCAзка о SCAнерах



Виктор Бобыльков
Райффайзенбанк

Дмитрий Евдокимов
Luntry

/ О нас

Руководитель Application Security Райффайзенбанк
Опыт в ИБ более 10 лет
Делаю Appsec удобным и безопасным

Founder&CTO Luntry
Опыт в ИБ более 10 лет
Автор Telegram-канала “k8s (in)security”
Автор курса “Cloud Native безопасность в Kubernetes”
Докладчик: BlackHat, HITB, ZeroNights, HackInParis, Confidence, SAS, OFFZONE, PHDays, Kazhackstan, DevOpsConf, KuberConf, VK Kubernetes Conference, HighLoad++ и др.

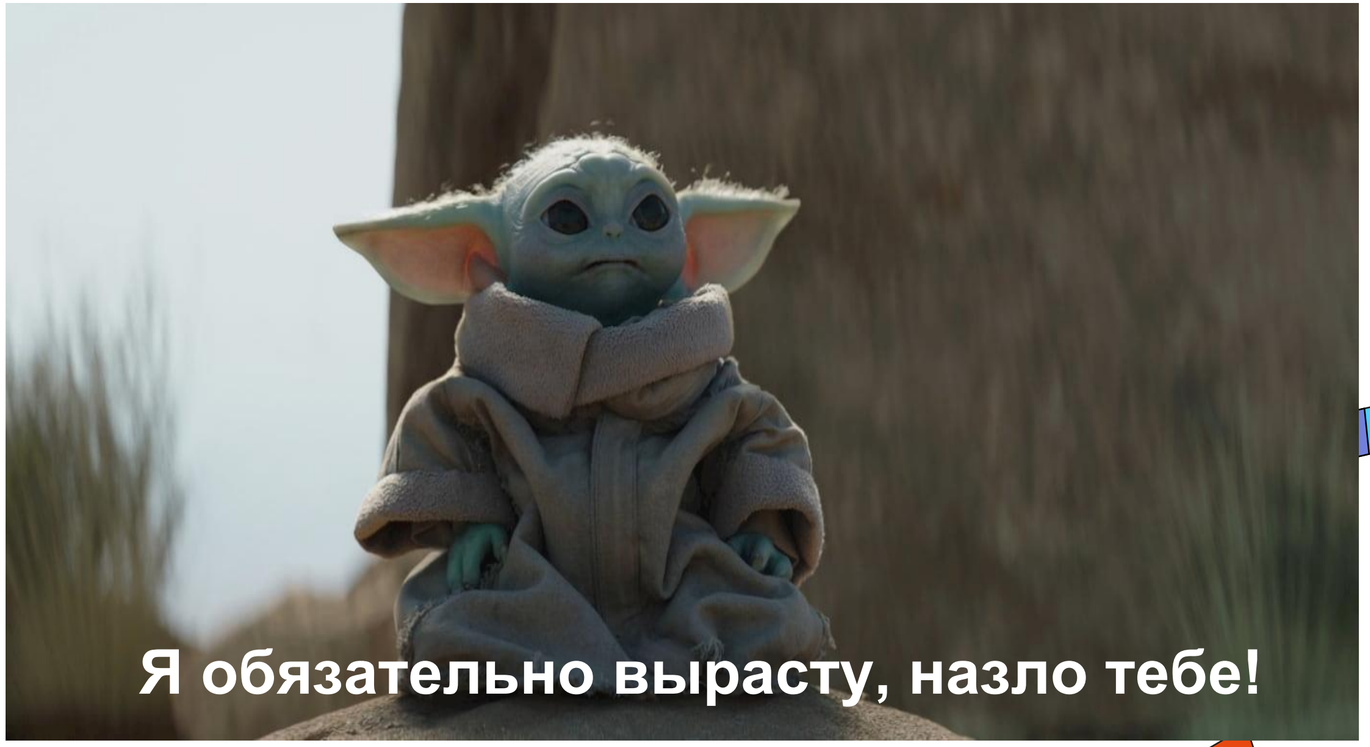
/ План доклада

1. SCA для чего? Почему

2. Методология измерения процессов CMMI

3. «Боли» на каждом уровне зрелости

4. Движение по уровням зрелости



Введение

/ Software Composition Analysis

Что смотрим

Уязвимости:

- Open-source библиотеки
- Различные зависимости
- Образы из общедоступных registry
- Финальный образ для сервиса

Вредоносная функциональность в зависимостях:

- Malware
- Protestware
- Dependency Confusion
- Supply Chain Attacks

/ Методология измерения

Capability
Maturity
Model
Integration

Модель зрелости и совершенствования процессов. Содержит набор рекомендаций в виде практик, для достижения определенного уровня зрелости.

Цель — улучшение процессов в компании

/ Оценка уровней зрелости (на базе CMMI)

0/1

Начальный уровень зрелости процесса
Процесс не формализован, постоянно меняется. Ему характерны хаотичность, реактивность, непредсказуемость.
Отсутствуют требования к методу исполнения шагов различных этапов. Один большой «черный» ящик.

2

Преимущественно стандартный уровень зрелости процесса
Процесс описан, его можно неоднократно использовать. Однако он все еще имеет некую долю реактивности.
Требования определены, немного контролируются. Реальное видение ситуации присутствует на промежуточных этапах.
Последовательность небольших "черных» ящиков.

3

Стандартный уровень зрелости процесса
Процесс определен, описан стандарт внутри организации. Присутствует более детальное описание процесса, лучше раскрываются все связи, знание которых позволяет улучшить управление.
Видна внутренняя сторона наших «черных» ящиков.

4

Уровень зрелости процесса выше среднего
Выбраны метрики, позволяющие количественно и качественно контролировать качество выполнения процесса.
Имеется предсказуемость эффективности процесса и возможность ею управлять.

5

Высокий уровень зрелости процесса
Имеем точные характеристики оценки эффективности, улучшаем процесс путем развития существующих практик и техник, безболезненно внедряем новые.

/ Зачем нам это?

Оценка эффективности процесса

Выбор приоритетных направлений для улучшения

Уход от парадигмы «виноваты люди»

Повышение качества разработки и безопасности продуктов

0/1 уровень

/ 0/1 уровень зрелости

Внедрили инструмент для SCA

- Не определены сценарии его использования
- Нет общего подхода к использованию
- Слабая информированность о процессе
- Подключены самые «прогрессивные» команды
- Непонимание ценности процесса для команд

0/1->2 уровень

/ 0/1 –> 2 уровень зрелости

- Описаны сценарии использования
- Определены требования
- Проинформированы команды разработки о процессе
- Контроль ИБ выборочный

/ 2 уровень зрелости

- Низкое качество триажа уязвимостей
- Нерациональное использование ресурсов AppSec
- Непонятный процент покрытия практикой
- SCA еще не включен в качестве стандарта для команд разработки
- Все еще большое число уязвимостей в типовых случаях

2->3 уровень

/ 2 –> 3 уровень зрелости

- Определение зон ответственности
- Внедрение практики базовых «золотых» образов
- Предоставление сервиса подключения к SCA
- Упрощение триажа уязвимостей
- Установление Quality Gates
- Контроль ИБ всех команд разработки

/ 3 уровень зрелости

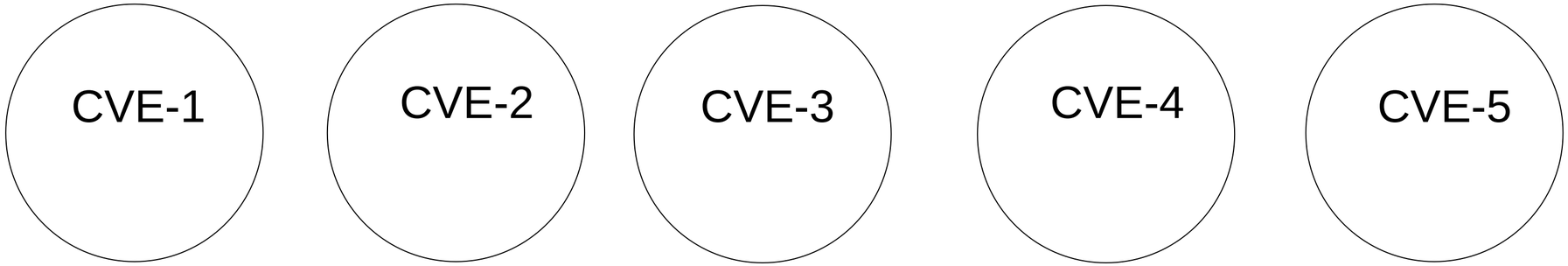
- Отсутствие метрик процесса и отчетности по метрикам
- Небольшое непонимание команд по триажу уязвимостей (false-positive, уязвимости без fix-version, транзитивные зависимости)
- Ценность практики для бизнеса

positive hack days12

Расширяем сознание

3->4 уровень

/ Проблематика: Что с начала?



/ Проблематика: Что с начала?

Критичность

CVE-1
Critical

CVE-2
Critical

CVE-3
Critical

CVE-4
High

CVE-5
High

/ Проблематика: Что с начала?

Критичность	CVE-1 Critical	CVE-2 Critical	CVE-3 Critical	CVE-4 High	CVE-5 High
Наличие патча	Да	Да	Нет	Да	Да
Уязвимость в слое ОС	Нет	Нет	Да	Нет	Нет
Удаленная эксплотация	Да	Да	Да	Да	Нет
Наличие эксплоита	Да	Нет	Нет	Да	Да
Приложение доступно извне	Да	Нет	Нет	Да	Нет
NetworkPolicy	Да	Да	Да	Нет	Да
SecurityContext	Да	Да	Да	Нет	Да

/ 3->4 уровень зрелости

- Имеем возможность наблюдать за происходящим в боевом окружении
  - От статичной картины переходим к динамичной и используем гибридный подход
  - Требуется дополнительный инструментарий

- Начинаем протезировать уязвимости проблемы
  - Ранжируем уязвимости с учетом окружения и условий
  - Proof of Concept
  - In the Wild
  - Наличие свойств безопасности

/ 4 уровень зрелости

- Понимаем какой компонент, какой версии, в каком образе- используется и где запущен и в каких условиях
  - Используется ли вообще?
    - Уже не используется вообще
  - В каком окружении?
    - Prod, Dev кластер? Тип Node? Init контейнер или основной? Deployment или Job?
  - В каких условиях?
    - Privileged контейнер? AppArmor, SeLinux, secomp есть? Internet faced app? NetworlPolicy есть? readOnlyRootFilesystem? Distroless?

- Ставим задачи на исправления на базе risk oriented подходе и меньше нагружаем бизнес/разработку/IT

4->5 уровень

/ Проблематика: Все идет по плану?

/ Работа с 1day уязвимостями

Обнаружение уязвимости разработчиком

/ Работа с 1day уязвимостями

0day

Время

Обнаружение уязвимости разработчиком

Уязвимый компонент уже работает в Prod окружении

/ Работа с 1day уязвимостями

0day

Время

Обнаружение уязвимости разработчиком

Выпуск патча и CVE

Уязвимый компонент уже работает в Prod окружении

/ Работа с 1day уязвимостями

0day

Время

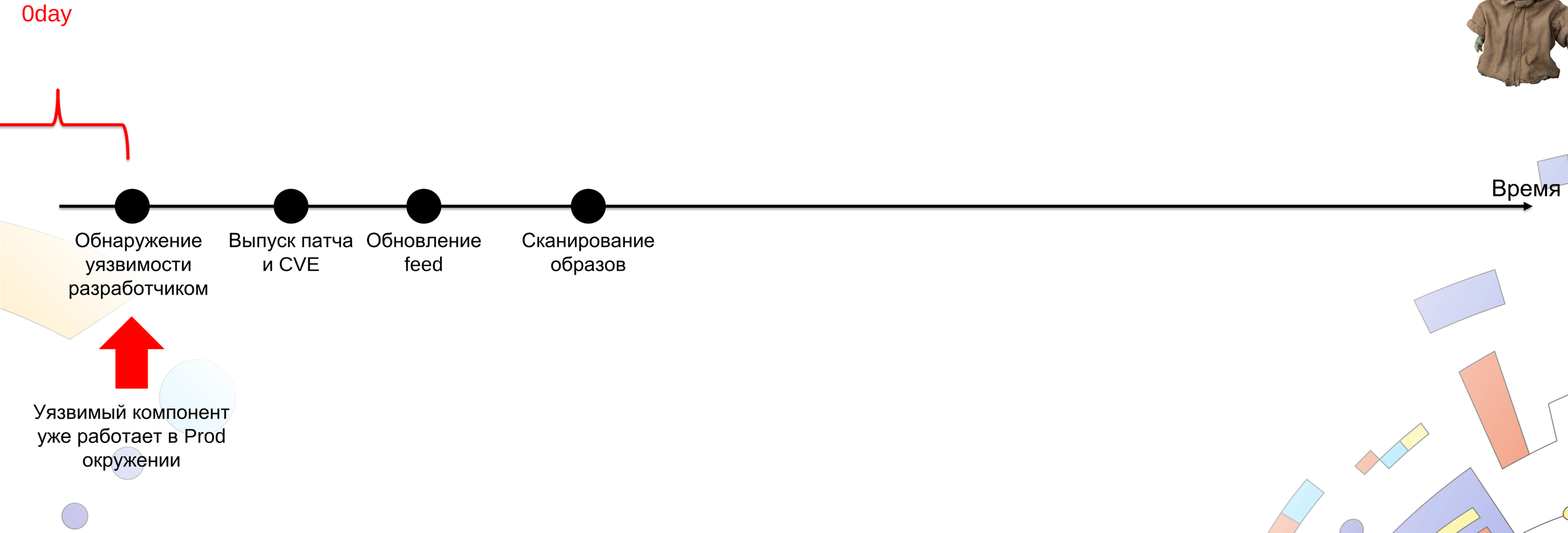
Обнаружение уязвимости разработчиком

Выпуск патча и CVE

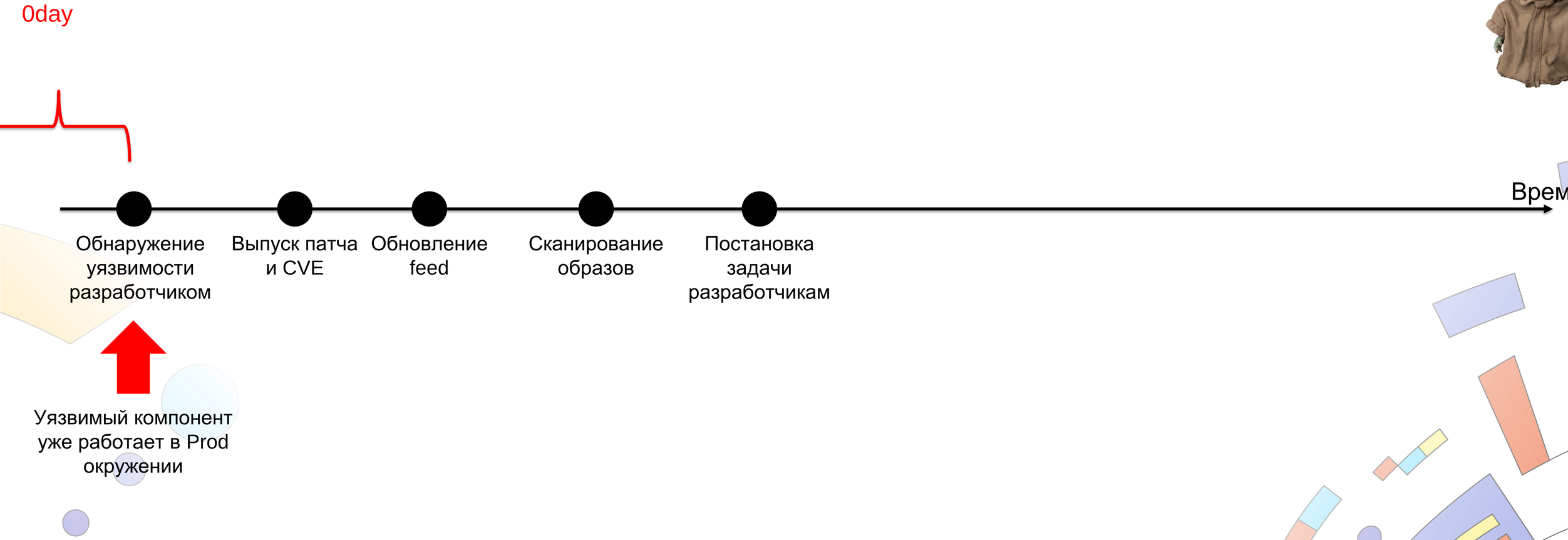
Обновление feed

Уязвимый компонент уже работает в Prod окружении

/ Работа с 1day уязвимостями



/ Работа с 1day уязвимостями



/ Работа с 1day уязвимостями

0day

Время

Обнаружение уязвимости разработчиком

Выпуск патча и CVE

Обновление feed

Сканирование образов

Постановка задачи разработчикам

Обновление компонента

Уязвимый компонент уже работает в Prod окружении

/ Работа с 1day уязвимостями

0day

Время

Обнаружение уязвимости разработчиком

Выпуск патча и CVE

Обновление feed

Сканирование образов

Постановка задачи разработчикам

Обновление компонента

Тестирование новой версии, QA, решение возможных проблем обратной совместимости, тестирование производительности т.д.

Уязвимый компонент уже работает в Prod окружении

/ Работа с 1day уязвимостями

0day

Время

Обнаружение уязвимости разработчиком

Выпуск патча и CVE

Обновление feed

Сканирование образов

Постановка задачи разработчикам

Обновление компонента

Тестирование новой версии, QA, решение возможных проблем обратной совместимости, тестирование производительности т.д.

Запуск на Test, Pre Prod окружении

Уязвимый компонент уже работает в Prod окружении

/ Работа с 1day уязвимостями

0day

Время

- Обнаружение уязвимости разработчиком
- Выпуск патча и CVE
- Обновление feed
- Сканирование образов
- Постановка задачи разработчикам
- Обновление компонента
- Тестирование новой версии, QA, решение возможных проблем обратной совместимости, тестирование производительности т.д.
- Запуск на Test, Pre Prod окружении
- Разворачивание на всех Prod окружениях

Уязвимый компонент уже работает в Prod окружении

/ Работа с 1day уязвимостями

0day

1day
Все это время окружение уязвимо!

(от патча до эксплоита примерно от нескольких часов до 2-7 дней)

Время

Обнаружение уязвимости разработчиком

Выпуск патча и CVE

Обновление feed

Сканирование образов

Постановка задачи разработчикам

Обновление компонента

Тестирование новой версии, QA, решение возможных проблем обратной совместимости, тестирование производительности т.д.

Запуск на Test, Pre Prod окружении

Разворачивание на всех Prod окружениях

Уязвимый компонент уже работает в Prod окружении

/ Работа с 1day уязвимостями

0day

1day
Все это время окружение уязвимо!

(от патча до эксплоита примерно от нескольких часов до 2-7 дней)

0day*

Время

- Обнаружение уязвимости разработчиком
- Выпуск патча и CVE
- Обновление feed
- Сканирование образов
- Постановка задачи разработчикам
- Обновление компонента
- Тестирование новой версии, QA, решение возможных проблем обратной совместимости, тестирование производительности т.д.
- Запуск на Test, Pre Prod окружении
- Разворачивание на всех Prod окружениях

Уязвимый компонент уже работает в Prod окружении

/ Работа с уязвимостями по факту

0day/1day

Время

Обнаружение уязвимости разработчиком

Выпуск патча и CVE

Обновление feed

Сканирование образов

Постановка задачи разработчикам

Обновление компонента

Тестирование новой версии, QA, решение возможных проблем обратной совместимости, тестирование производительности т.д.

Запуск на Test/Pre Prod окружении

Разворачивание на всех Prod окружениях

Уязвимый компонент уже работает в Prod окружении

Уязвимый компонент уже работает в Prod окружении

Уязвимый компонент уже работает в Prod окружении

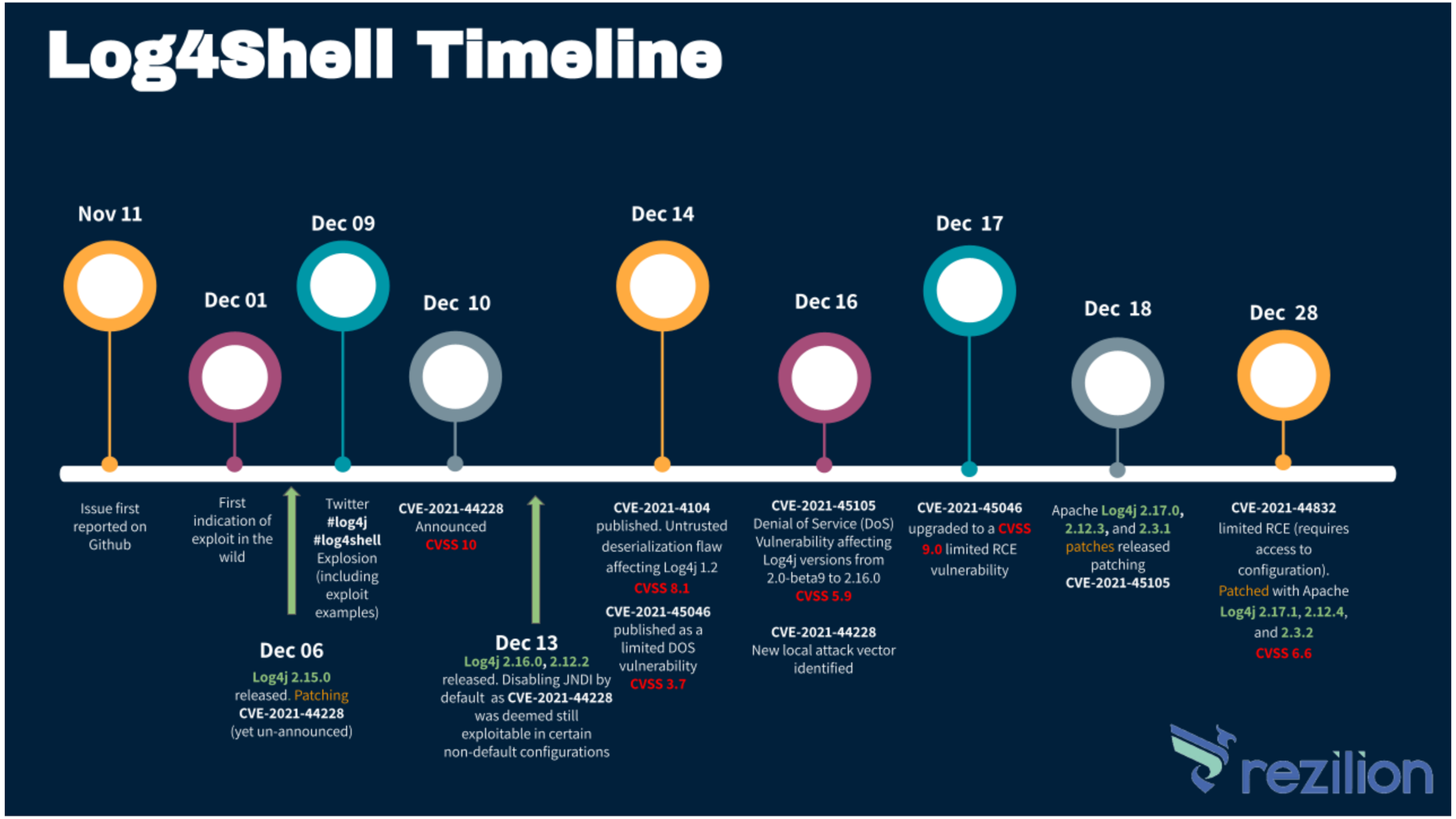
Уязвимый компонент уже работает в Prod окружении

Уязвимый компонент уже работает в Prod окружении

Уязвимый компонент уже работает в Prod окружении

Уязвимый компонент уже работает в Prod окружении

/ Пример: Log4Shell



Источник: “Making Sense of the Constantly Changing Log4Shell Landscape”

/ 4->5 уровень зрелости

- Игра на опережении
  - Меняем парадигму где атакующий имеет преимущество и находится на шаг впереди
- Переход к ZeroTrust
  - Стройте безопасность с мыслью о том что вас все равно взломают

/ 5 уровень зрелости

- От реактивной к проактивной ИБ
  - Защищаемся митигируем риски эксплуатации 1day, 0day уязвимостей и вредоносного кода

- Бизнес получает больше времени на исправлении уязвимостей, исправления становятся планируемыми, а не авральными

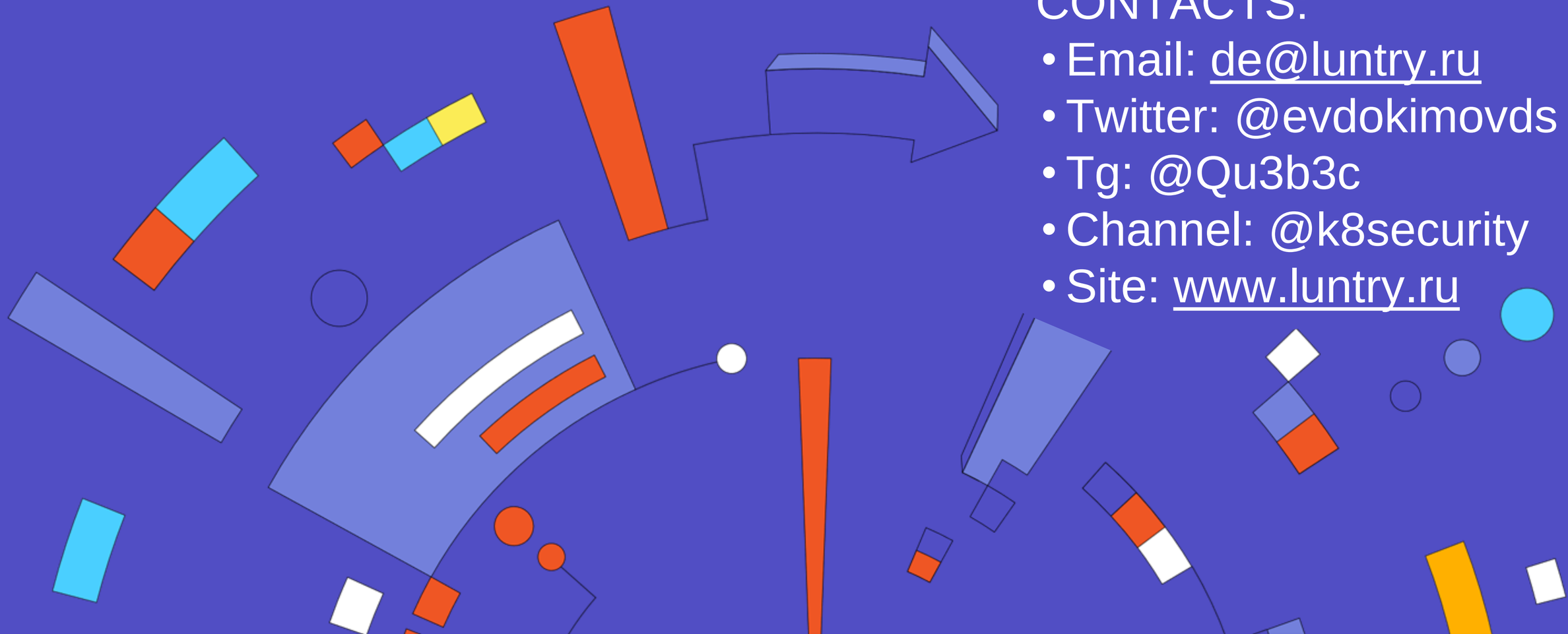
Заключение

/ Выводы

- Просто включение SCA в pipeline не дает ничего
- Командам разработки необходимо помочь в работе с инструментом
- Увеличиваем контроль за счет удобства
- Приоритезация проблем чрезвычайно важна
- SCA не серебряная пуля и не гарантирует отсутствие инцидентов

positive hack days 12

спасибо!



CONTACTS:

- Email: de@luntry.ru
- Twitter: @evdokimovds
- Tg: @Qu3b3c
- Channel: @k8security
- Site: www.luntry.ru