LUNTRY

Уязвимости в контейнерах и Kubernetes: правим, а не страдаем

Дмитрий Евдокимов

Основатель
и Технический
директор Luntry

Обо мне

Дмитрий Евдокимов

> Я не верю в то, что систему можно сделать надежной и безопасной, не понимая того, как она устроена.

Основатель и технический директор Luntry

Более 15 лет опыта в ИБ

Специализация — безопасность контейнеров и Kubernetes

Автор ТГ-канала k8s(in)security

Эксперт в сфере безопасности контейнерных сред

- Организатор конференции «БеКон» по БЕзопасности КОНтейнеров
- Бывший редактор рубрик в журнале «ХАКЕР», автор серии статей
- Автор курса «Cloud Native безопасность в Kubernetes»
- Член программного комитета CFP DevOpsConf и HighLoad++

Спикер

VK Kubernetes	Confidence	ZeroNights	БеКон	HITB
DevOpsConf	HackInParis	KuberConf	BlackHat	PHDays
Kazhackstan	HighLoad++	OFFZONE	DevOops	SAS

LUNTRY

О компании Luntry

Luntry — это Комплексная Защита
на всем жизненном цикле
контейнерных приложений
и средств оркестрации на базе Kubernetes

Продукт в реестре Минцифры
https://reestr.digital.gov.ru/reestr/1057835/

Получение сертификата ФСТЭК планируется во втором квартале 2026 года.

LUNTRY.RU

Функциональность Luntry

Контроль всех Kubernetes-ресурсов

Контроль состояния Kubernetes-кластеров

Контроль соответствия кластера стандартам

Управление безопасностью образов контейнеров

Управление сетевой безопасностью

Анализ прав доступа

Защита Runtime

Проактивный и реактивный подходы к безопасности

Культура киберустойчивости: «При построении культуры киберустойчивости **роль службы безопасности заключается не в предотвращении всех инцидентов, а в предотвращении негативного влияния инцидента безопасности на бизнес**».

Уязвимости != уязвимый

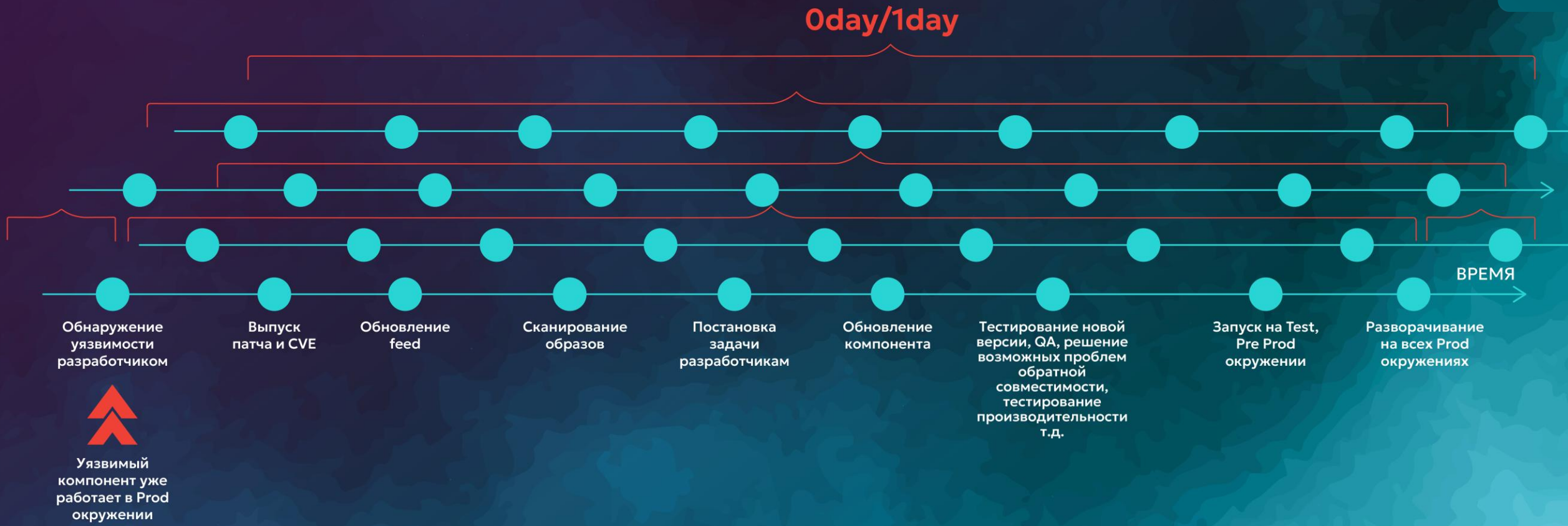
РЕАКТИВНЫЙ ПОДХОД

Работа с 1day уязвимостью

LUNTRY

Работа с уязвимостями по факту

LUNTRY



Жизненный путь образа

LUNTRY

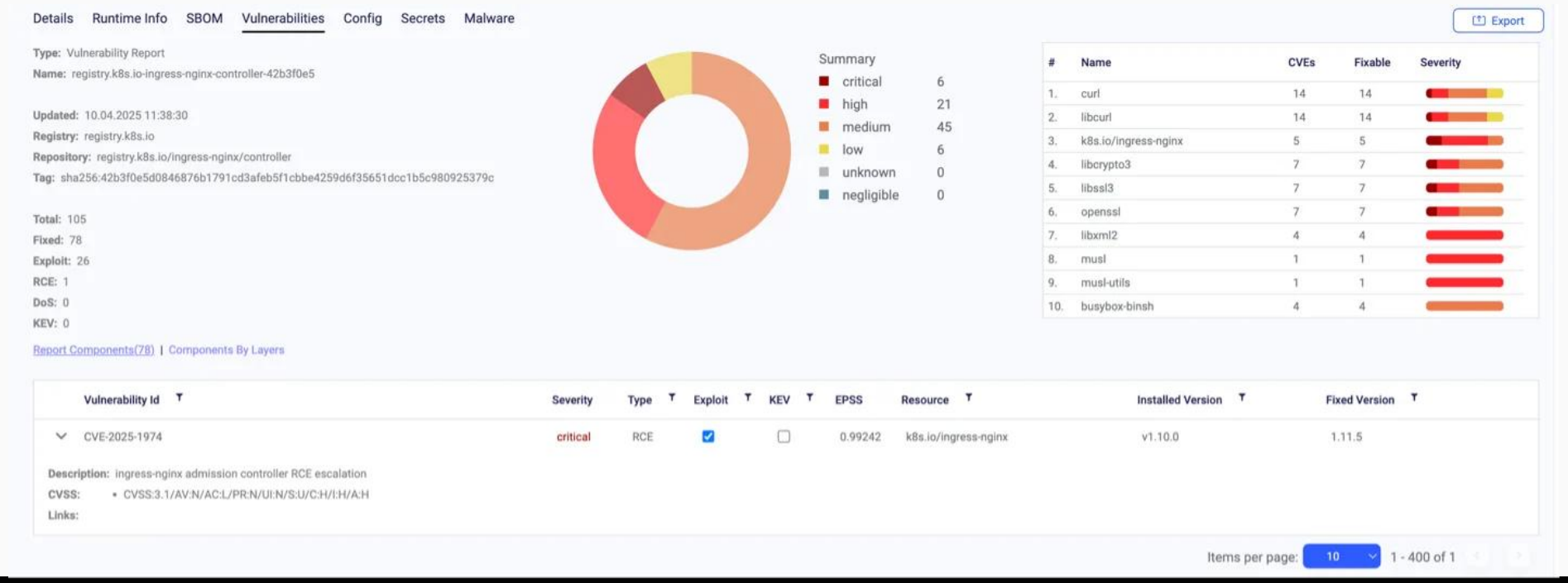
CI\CD Pipeline
- Сборка образа
- Анализ образа
- Security/Quality gates
  - Возможна блокировка
- Подпись образа

Image registry
- Хранение
- Очистка/ротация
- Анализ образов
- Security/Quality gates

Runtime
- Запуск
- Анализ образов
- Security/Quality gates

Анализ образов в Luntry

- Встраивание в CI\CD, Registry, Runtime
- Инвентаризация использования образов
- Компонентный анализ (SBOM)
- Сканирование на известные уязвимости
- Поддержка отечественных ОС и БДУ ФСТЭК
- Приоритизация уязвимостей CVSS, Severity, Type(RCE/DoS), Exploit, KEV, EPSS, Runtime
- Контроль соответствия лучшим практикам
- Поиск секретов
- Анализ на вредоносный код и код двойного назначения
- Поддержка внешних и собственных IoC
- Встроенная библиотека уникальных проверок
- Скорость и парализация сканирований
- Security Gates в CI\CD, Registry, Runtime



Приоритезация уязвимостей

LUNTRY

CVSS

Severity

Type(RCE/DoS)

Exploit

KEV

EPSS

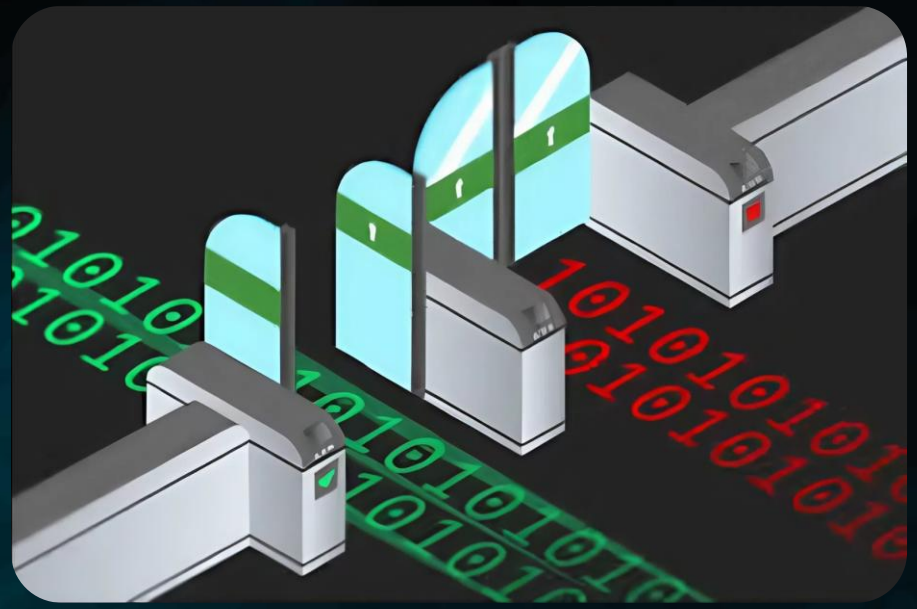
Runtime

Security/Quality Gates – ворота качества

CI
- Уведомление
- Блокировка

Image registry
- Уведомление

Runtime
- Уведомление



Блокировка на Deploy, Runtime — опасная и плохая практика!

«Безопасность Kubernetes кластеров: вредные советы», DevOpsConf 2024

Пример 1: В CI

Максимум экономии на “самом дешёвом” этапе:

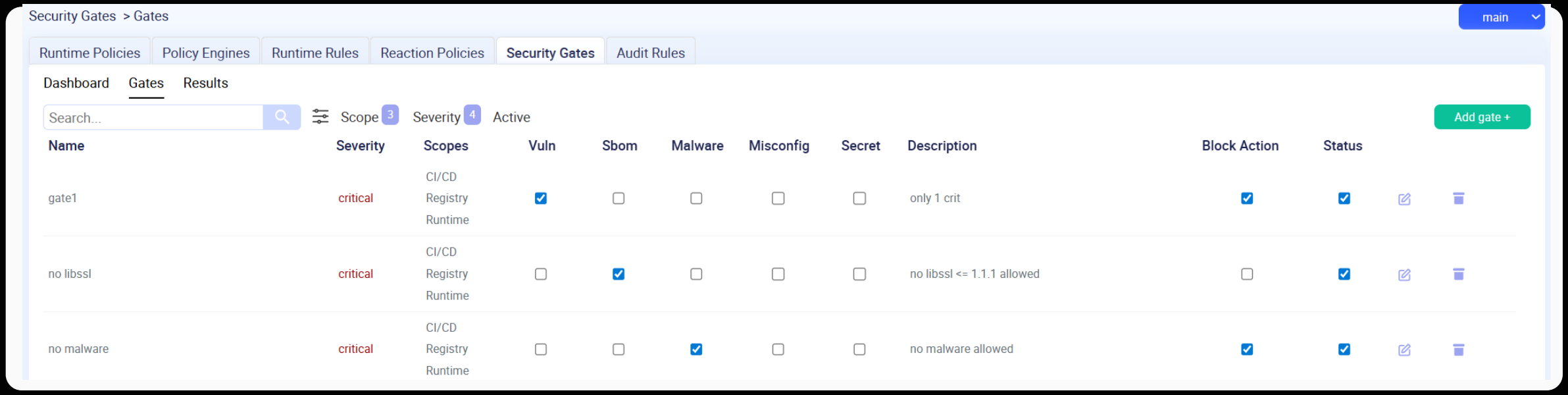
- Раннее обнаружение дефектов

Экономика:

- Исправление проблемы на этапе CI дешевле, чем на последующих этапах

Эффект на команду:

- Снижается нагрузка на QA / DevOps / SRE



Пример 2: В Image Registry

Снижение риска и затрат на масштабирование ошибок:

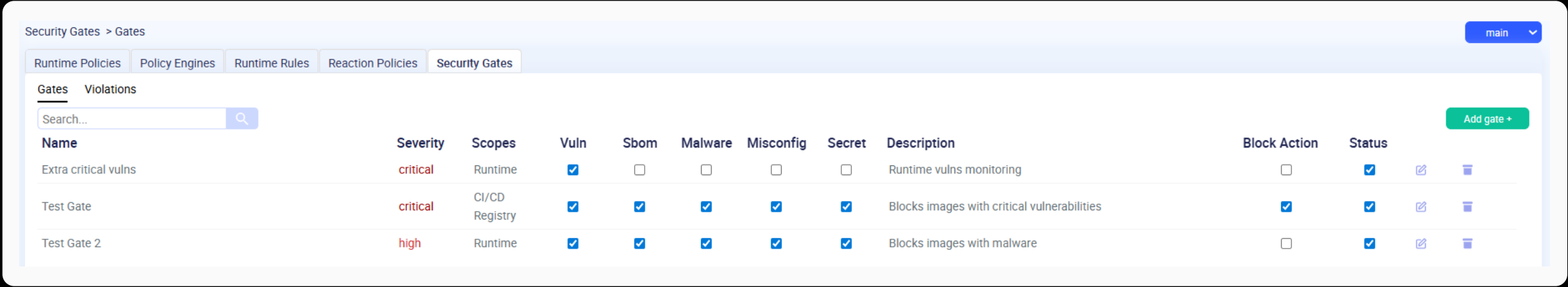
- Единая точка контроля

Экономика:

- Снижает стоимость аудита/соответствия — упрощает проверки и перепроверки, сокращает ручной труд

Эффект на команду:

- Снижение FTE на команду безопасности



Пример 3: В Runtime

Фокусирование на действительно опасном и важном в текущий момент:

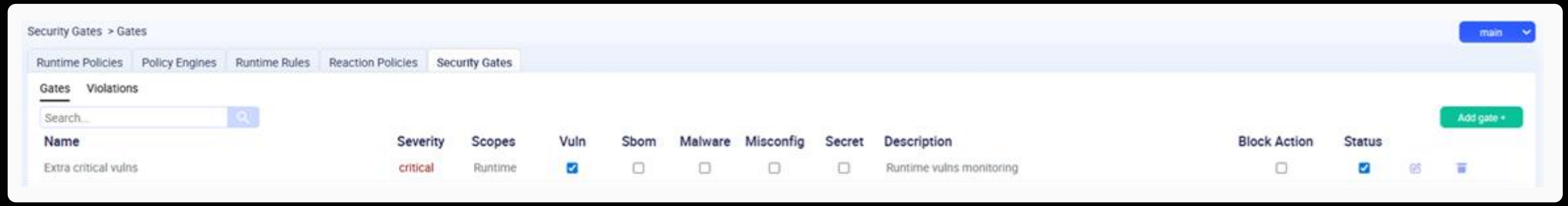
- Контроль самого актуального

Экономика:

- Экономия на инцидентах и простоях

Эффект на команду:

- Простота работы и понижение порога входа за счет приоритезации задач



Польза Security Gates

LUNTRY

Defense in depth:

- CI: “дешево исправить”
- Registry: “не дать распространиться”
- Runtime: “не дать навредить (или минимизировать ущерб)”

Ключевые бизнес-метрики, которые обычно улучшаются

- Lead time / time-to-release: меньше откатов и “красных” релизов.
- Change failure rate: снижается за счёт отсева дефектов и небезопасных изменений.
- MTTR (время восстановления): ниже, потому что меньше инцидентов и выше прозрачность.
- Стоимость владения (TCO) платформы: меньше ручных процессов, меньше “пожарных” часов, меньше потерь от downtime.

Но проблема усиливается …

Обновление политики NIST по NVD

LUNTRY

ПРОАКТИВНЫЙ ПОДХОД

Вклад AI

For the first time in bug bounty history, an autonomous penetration tester has reached the top spot on the US leaderboard.

XBOW

IDENTIFYING VULNERABILITIES AND EXPLOITS WITH CLAUDE MYTHOS PREVIEW

Over the past few weeks, we have used Claude Mythos Preview to identify thousands of zero-day vulnerabilities (that is, flaws that were previously unknown to the software's developers), many of them critical, in every major operating system and every major web browser, along with a range of other important pieces of software.

In a post on our Frontier Red Team blog, we provide technical details for a subset of these vulnerabilities that have already been patched and, in some cases, the ways that Mythos Preview found to exploit them. It was able to identify nearly all of these vulnerabilities—and develop many related exploits—entirely autonomously, without any human steering. The following are three examples:

- Mythos Preview found a 27-year-old vulnerability in OpenBSD—which has a reputation as one of the most security-hardened operating systems in the world and is used to run firewalls and other critical infrastructure. The vulnerability allowed an attacker to remotely crash any machine running the operating system just by connecting to it;
- It also discovered a 16-year-old vulnerability in FFmpeg—which is used by innumerable pieces of software to encode and decode video—in a line of code that automated testing tools had hit five million times without ever catching the problem;
- The model autonomously found and chained together several vulnerabilities in the Linux kernel—the software that runs most of the world's servers—to allow an attacker to escalate from ordinary user access to complete control of the machine.

Claude Mythos

LUNTRY

Реалии и прогнозы

Появление Claude Mythos происходит в контексте уже ускоряющегося временного окна между раскрытием уязвимости и её эксплуатацией.

Окно сокращается с 63 дней в 2018-2019 годах до всего нескольких часов к 2024-2025 годам[1][2]. Claude Mythos угрожает ещё больше сжать это окно.

С выпуском моделей, подобных Mythos, по прогнозам, время от раскрытия уязвимости к полному наступательному потенциалу может сжаться до минут или даже секунд[1][3].

Это представляет **фундаментальный сдвиг в динамике кибербезопасности**, где традиционные циклы тестирования и развёртывания патчей, которые измеряются днями или неделями, становятся неадекватными[1][4][5].

Контейнеры на страже ИБ

LUNTRY

Микросервисы (контейнеры)

- Дополнительная изоляция
  - Дополнительный слой защиты

Простота

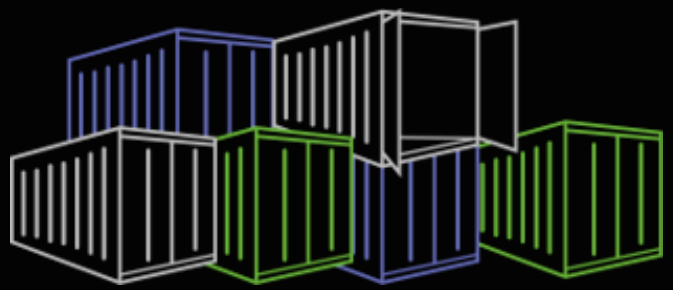
- Микросервис проще ОС и монолитов
  - Иммутабельность образа контейнера

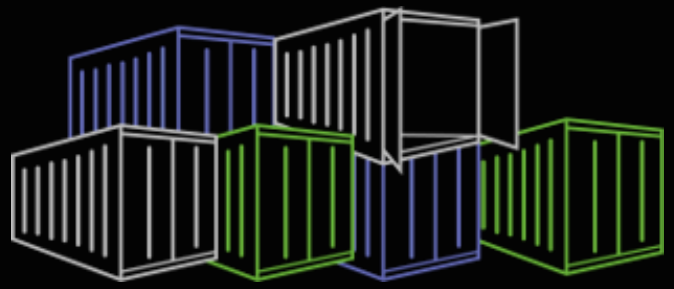
Предсказуемость поведения

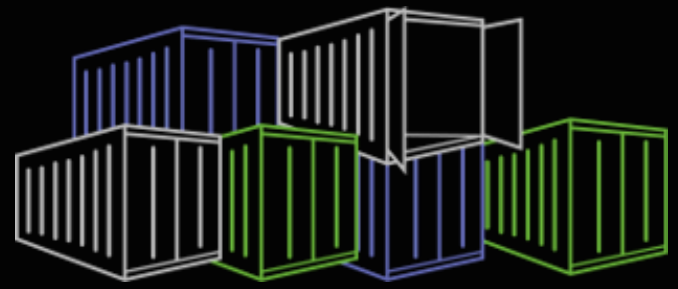
- Распределенность
  - Работа микросервисов в несколько копий

Эфемерность

- Высокая скорость модификации без простоя системы



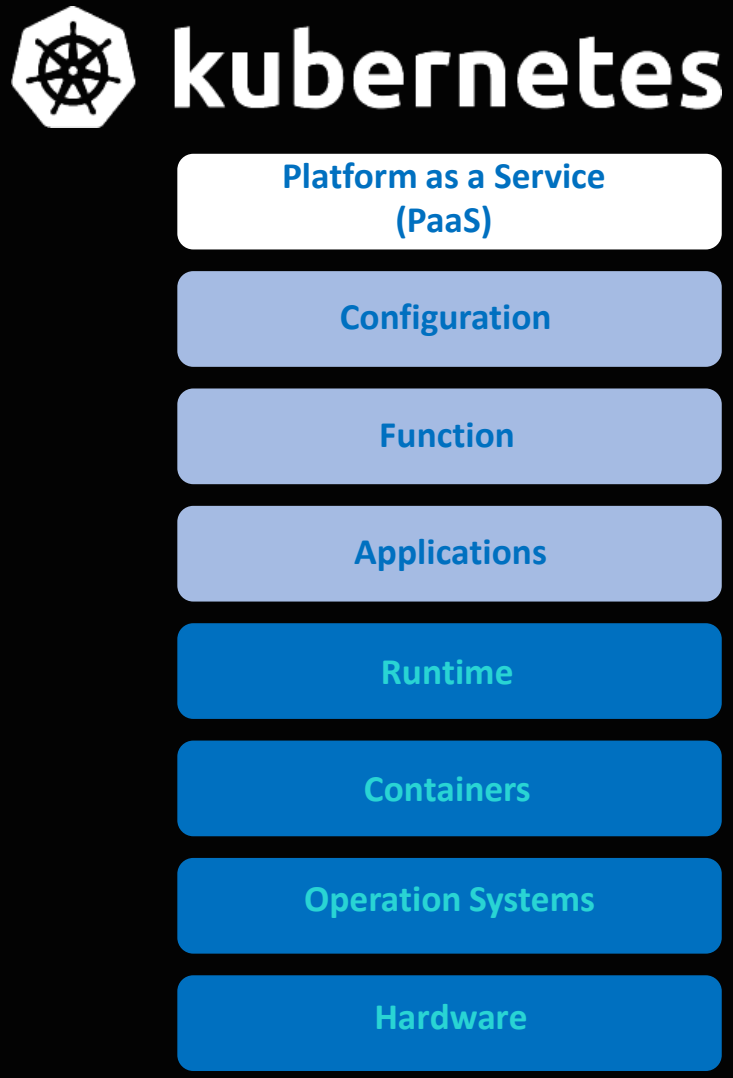




Kubernetes на страже ИБ

KUBERNETES:

- Декларативность
  - Security/Policy-as-Code
    - PolicyEngines
  - Прозрачное взаимодействие с ИТ
- ZeroTrust
  - Whitelisting
  - Micro segmentation
- ShiftLeftSecurity
  - SecDevSecOpsSec



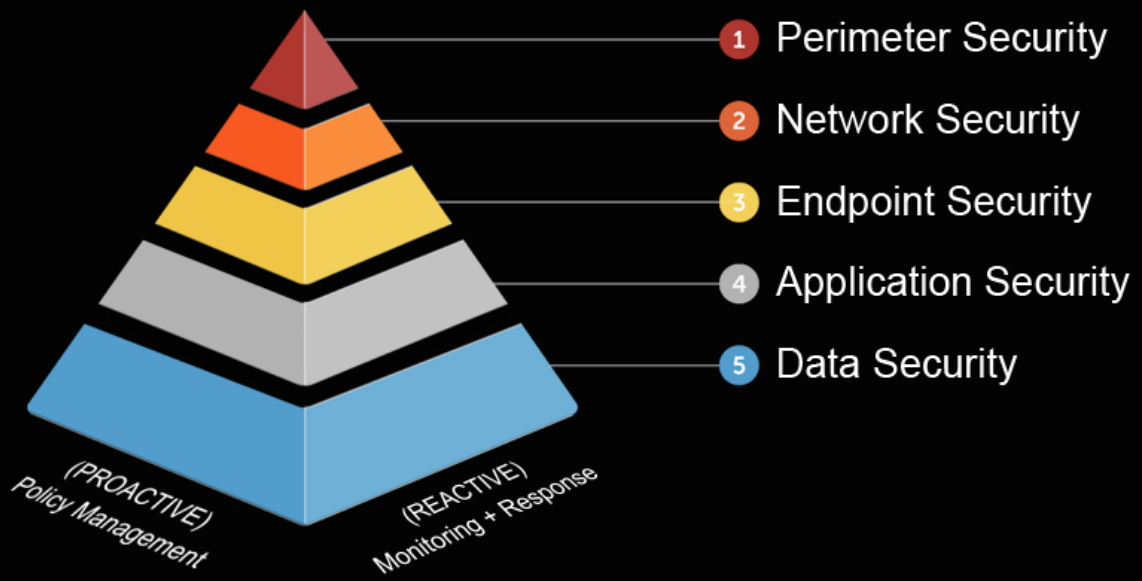
LUNTRY

Пример проактивной защиты

LUNTRY

МИКРОСЕРВИС:

- На базе минимального образа
  - distroless
- С файловой системой, доступной только на чтение
  - SecurityContext.readOnlyRootFilesystem.true
- Без лишних возможностей
  - SecurityContext.capabilities.drop.all
- Со строго определённым набором исполняемых файлов
  - AppArmor profile
- С ограниченной активностью по сети
  - NetworkPolicy



Luntry для проактивного подхода

LUNTRY

Контроль hardening механизмов для микросервисов

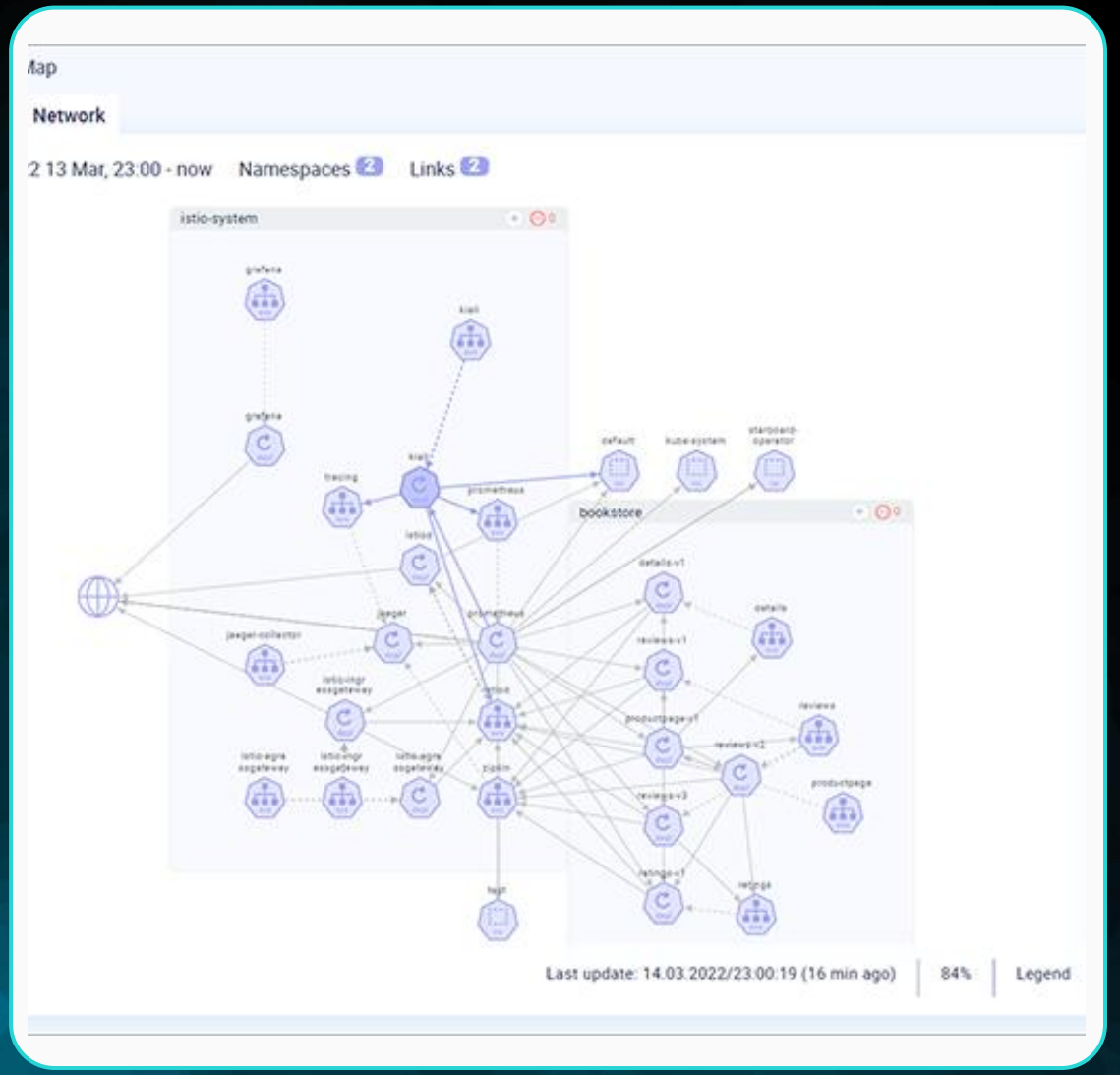
Автоматическое создание сетевых политик NetworkPolicy

Автоматическое создание профилей поведений

Генерация prevention policy

Luntry для проактивного подхода

Автоматическое создание сетевых политик NetworkPolicy



LUNTRY

Luntry для проактивного подхода

Генерация
prevention policy

Автоматическое создание
профилей поведений

LUNTRY

Luntry для проактивного подхода

Контроль hardening механизмов для микросервисов

LUNTRY

Выводы

LUNTRY

01 Разный уровень зрелости, требований, критичности позволяют использовать разные подходы

02 Ситуация с уязвимостями с каждым днем становиться все хуже

03 Специфика контейнеров и Kubernetes позволяет более прозрачно смотреть на вопрос управления уязвимостями

04 Luntry позволяет построить комплексный подход с использованием реактивных и проактивных практик

LUNTRY

luntry_official
luntrysolution
luntrysolution

luntry.ru
info@luntry.ru

ДМИТРИЙ ЕВДОКИМОВ
Founder & CTO Luntry

de@luntry.ru
Qu3b3c
k8security

СПАСИБО ЗА ВНИМАНИЕ!