Runtime Security: на вкус и цвет все фломастеры разные

Дмитрий Евдокимов

Founder&CTO Luntry

Обо мне

- Основатель и технический директор Luntry
- Опыт в ИБ более 10 лет
- Специализация безопасность контейнеров и Kubernetes
- CFP DevOpsConf, HighLoad++
- Бывший автор статей и редактор рубрик в журнале “ХАКЕР”
- Автор Telegram-канала “k8s (in)security”
- Автор курса “Cloud Native безопасность в Kubernetes”
- Не верит, что систему можно сделать надежной и безопасной, не понимая ее
- Организатор конференции по безопасности контейнеров - БеКон
- Докладчик: BlackHat, HITB, ZeroNights, HackInParis, Confidence, SAS, OFFZONE, PHDays, Kazhackstan, DevOpsConf, DevOops, KuberConf, VK Kubernetes Conference, HighLoad++, БеКон и др.

О компании Luntry

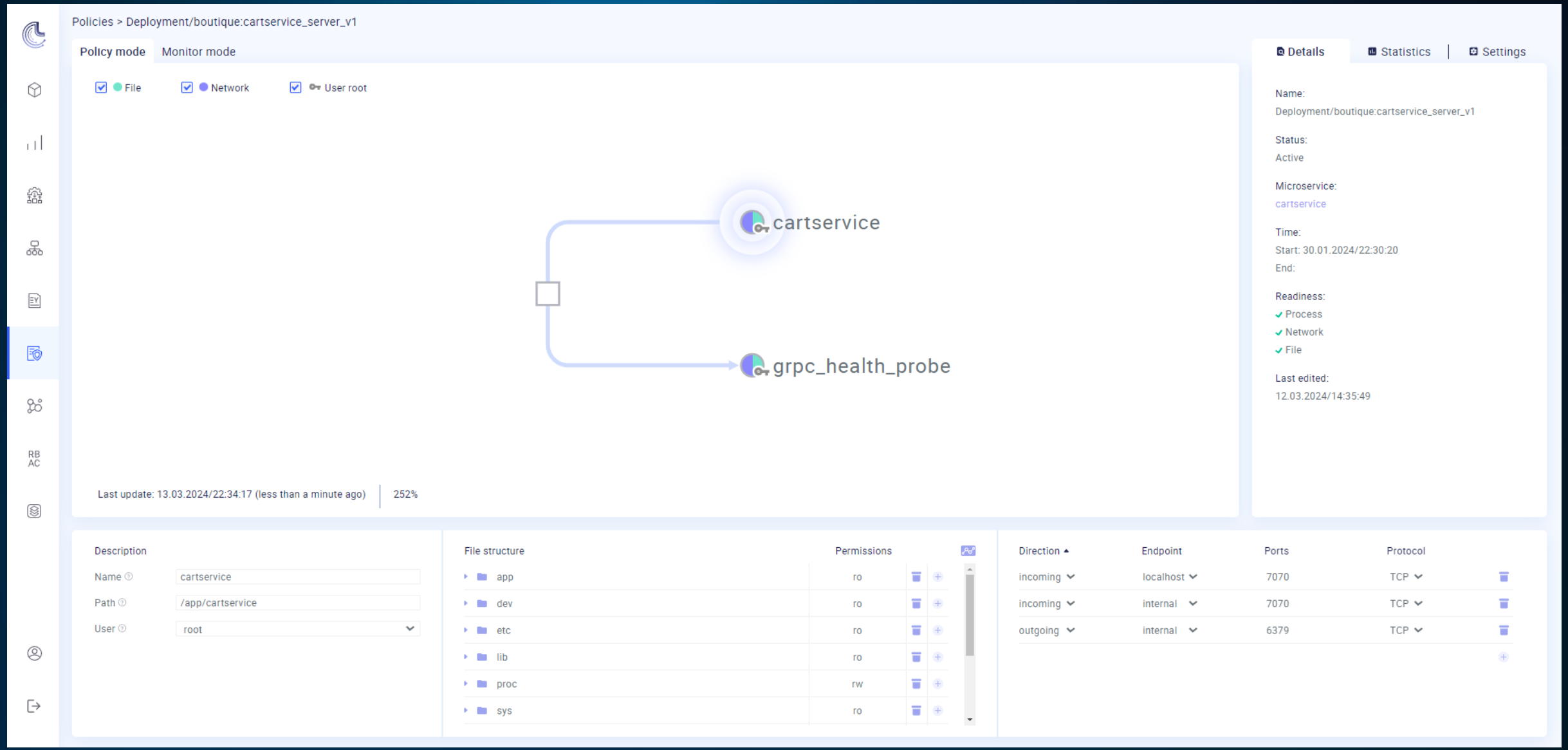
- Luntry – это Комплексная Защита на всем жизненном цикле контейнерных приложений и средств оркестрации на базе Kubernetes
- Продукт в реестре Минцифры
  - https://reestr.digital.gov.ru/reestr/1057835/
- В процессе получения сертификата ФСТЭК
  - Ориентировочно второй квартал 2024

LUNTRY
luntry.ru

Функциональность Luntry

Функциональность Luntry

Luntry – контроль runtime



План вебинара

- Введение в Runtime-защиты
- Подходы к работе, обнаружению угроз и их проблемы
- Возможности Open Source решений
- Взгляд Luntry

Почему я?

- Опыт обхода средств защиты
  - Включая CSP
- Опыт reverse engineering средств защиты
  - Включая CSP

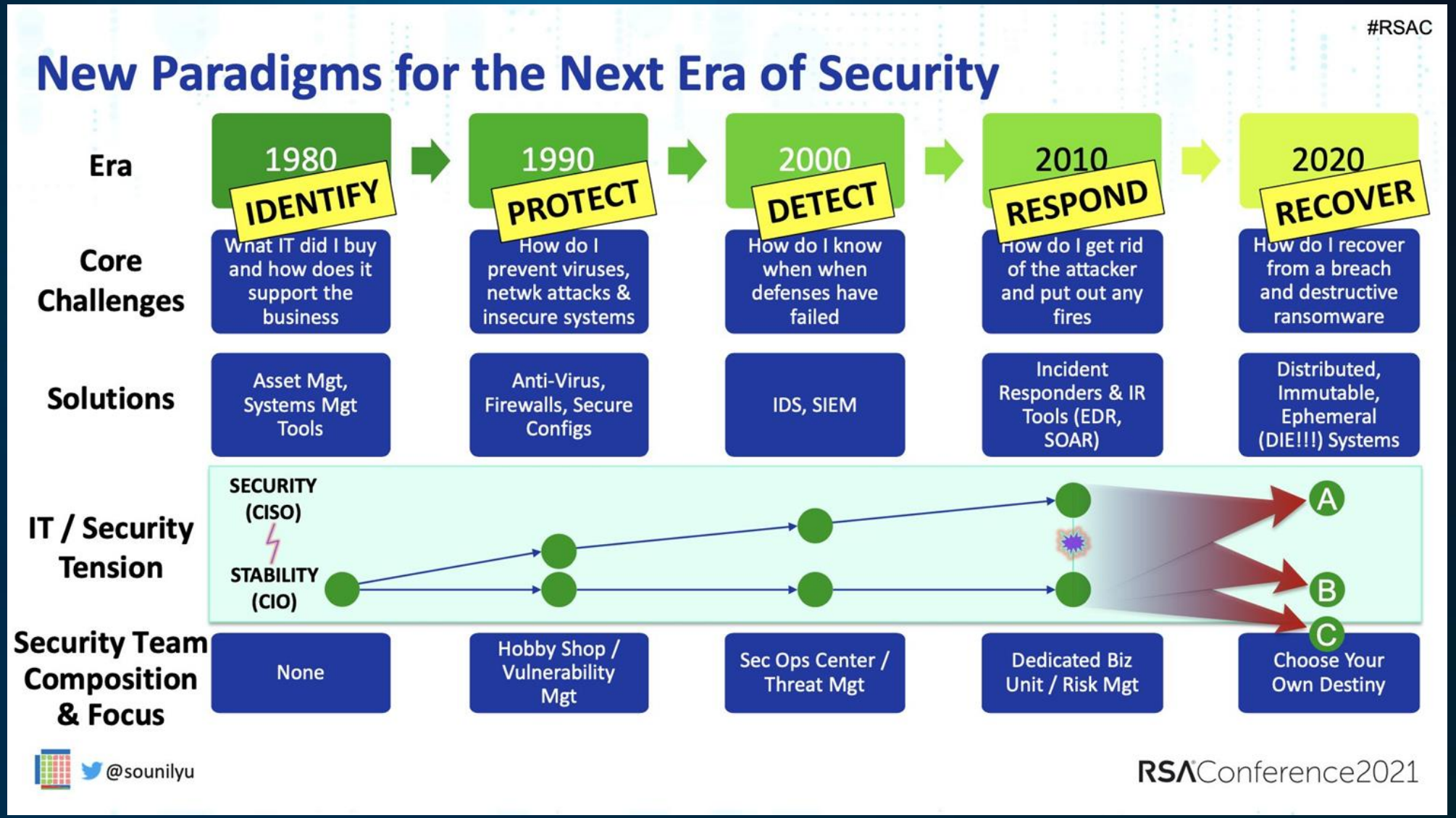
- Про обходы средств защиты говорить не будем
- Погружаться детально в код реализаций не будем

Материалы по теме

- "EDR vs Containers: актуальные проблемы", Владислав Лашкин, Дмитрий Евдокимов, SOC-форум 2023
- "SOC в контейнерах", Дмитрий Евдокимов, SOC-форум 2023
- "How to hide your actions when every step is being monitored", Иван Гаврилов, OFFZONE 2023
- "eBPF в production-условиях", Дмитрий Евдокимов, Александр Трухит, HighLoad++ 2022

Введение в Runtime-защиты

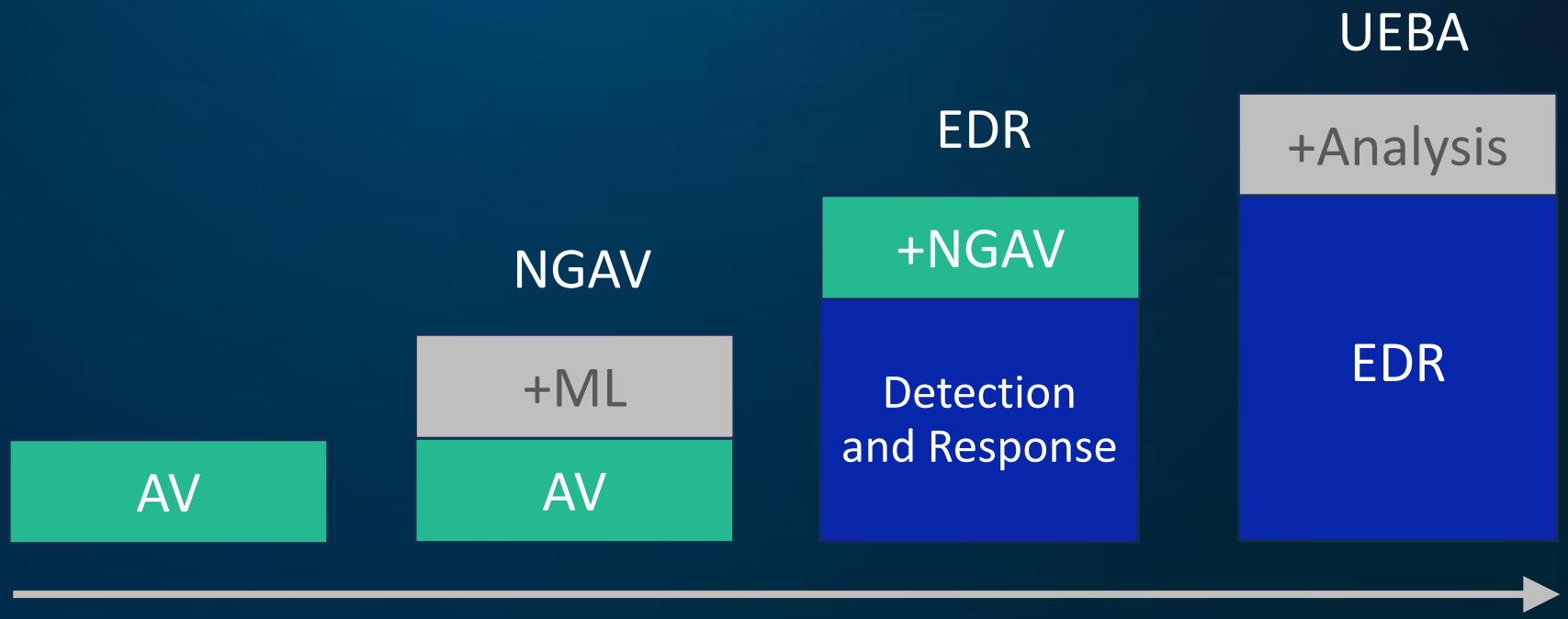
Новые парадигмы безопасности



"New Paradigms for the Next Era of Security"

Развитие средств защиты

*мое личное скромное мнение



Виды защиты (Linux world)

- Isolation
  - Дополнительный уровень изоляции от ядра хостовой ОС
  - WASM
    - WasmEdge, Wasmtime
  - Sandbox
    - gVisor
  - MicroVM
    - Kata containers

- Detection
  - Идентификация нежелательного действия
  - Сторонние решения

- Prevention
  - Невозможность выполнения нежелательного действия
  - Seccomp
  - AppArmor
  - SeLinux
  - LSM
  - iptables/eBPF

- Mitigation
  - Stack cookies
  - W^X
  - ASLR
  - SMEP
  - SMAP

- Reaction
  - Завершение процесса/контейнера/Pod постфактум после нежелательного события и/или дамп файловой системы и/или памяти
  - CRI интерфейс (stop/remove)
  - Kubelet Checkpoint API

Встроенные Runtime-защиты

- Linux capability
  - Определяет, какими правами обладает контейнер
  - Kubernetes SecurityContext

- Seccomp
  - Определяет, какие системные вызовы могут исполняться контейнером, а какие нет

- AppArmor (LSM)
  - Для Debian-based ОС
  - Определяет, какому файлу что можно, в терминах capabilities и файловых прав доступа в контейнере

- SeLinux (LSM)
  - Для RedHat-based ОС
  - Определяет, как контейнерные процессы взаимодействуют с хостовой ОС

- PARSEC (LSM)
  - AstraLinux
  - На текущий момент не отличает контейнерные процессы от хостовых

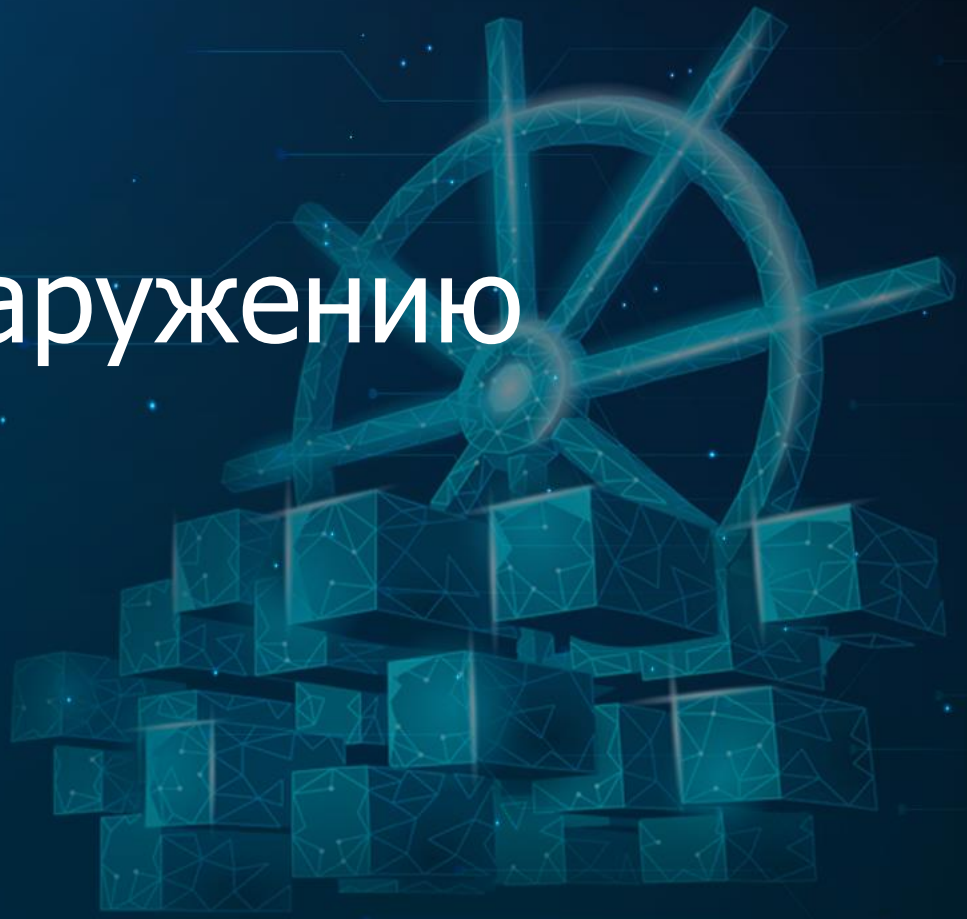
- NetworkPolicy (K8s)
  - Реализуется CNI-плагином в Kubernetes (iptables, eBPF)
  - Определяет, что можно, а что нет только на уровне сети (L3/L4,L7)

Специализация

НЕСПЕЦИАЛИЗИРОВАННЫЕ	СПЕЦИАЛИЗИРОВАННЫЕ
Auditd, Sysmon for Linux, классические EDR, …	Luntry, Falco (на его базе Sysdig, StackRox…), Aqua Security, PaloAlto Prisma, Tetragon, Tracee (на его базе KubeArmor)…

СПЕЦИАЛИЗИРОВАННЫЕ – ЭТО ТЕ, КОТОРЫЕ ПОНИМАЮТ КОНТЕКСТ КОНТЕЙНЕРОВ И КОНТЕКСТ СУЩНОСТЕЙ KUBERNETES

Подходы к работе, обнаружению угроз и их проблемы



Обзор

Базовая технология агента
Обработка данных на user space
Профилирование контейнеров
Отслеживаемые события
Обработка событий
Тип обнаружений
Создание политик/правил
Структура политики/правил
Принцип работы
Привязка политик/правил
Принцип привязки политик/правил
Режим работы
Активное воздействие (reaction)
Расследование инцидента
Предотвращение процессных событий
Предотвращение сетевых событий

В КОНТЕЙНЕРЕ

- Библиотека
  - User mode
  - Подмена low-level runtime
  - LD_PRELOAD, dlopen, патчинг GOT-таблиц и т. д.

ЗА ПРЕДЕЛАМИ КОНТЕЙНЕРА

- Sidecar-контейнер
  - User mode
- Sensor-based

  Очень привилегированный компонент
  - Kernel module
  - eBPF

Обзор

Базовая технология агента
Обработка данных на user space
Профилирование контейнеров
Отслеживаемые события
Обработка событий
Тип обнаружений
Создание политик/правил
Структура политики/правил
Принцип работы
Привязка политик/правил
Принцип привязки политик/правил
Режим работы
Активное воздействие (reaction)
Расследование инцидента
Предотвращение процессных событий
Предотвращение сетевых событий

C\C++

- Libbpf
- Сложности при многопоточности

GO

- libbpf-go
- Сложности с приоритетами для потоков

RUST

- libbpf-rs
- Сложности с порогом вхождения

Обзор

Базовая технология агента
Обработка данных на user space
Профилирование контейнеров
Отслеживаемые события
Обработка событий
Тип обнаружений
Создание политик/правил
Структура политики/правил
Принцип работы
Привязка политик/правил
Принцип привязки политик/правил
Режим работы
Активное воздействие (reaction)
Расследование инцидента
Предотвращение процессных событий
Предотвращение сетевых событий

Автоматическое

- Отображение того что и как работает в контейнере на базе анализа информации

Ручное

- Ручное определение на базе представления оператора

Нет

- На нет и суда нет

Обзор

- **Базовая технология агента**
- **Обработка данных на user space**
- **Профилирование контейнеров**
- **Отслеживаемые события**
- **Обработка событий**
- **Тип обнаружений**
- **Создание политик/правил**
- **Структура политики/правил**
- **Принцип работы**
- **Привязка политик/правил**
- **Принцип привязки политик/правил**
- **Режим работы**
- **Активное воздействие (reaction)**
- **Расследование инцидента**
- **Предотвращение процессных событий**
- **Предотвращение сетевых событий**

Syscalls

- Тонкий подход к любому системному вызову
- Требуется отличное понимание системных вызовов

Высокоуровневый базовые события

- Process, Network, Files
- Понятно и близко пользователю

Собственные высокоуровневые группировки

- Среднее между двумя предыдущими вариантами

Обзор

Базовая технология агента
Обработка данных на user space
Профилирование контейнеров
Отслеживаемые события
Обработка событий
Тип обнаружений
Создание политик/правил
Структура политики/правил
Принцип работы
Привязка политик/правил
Принцип привязки политик/правил
Режим работы
Активное воздействие (reaction)
Расследование инцидента
Предотвращение процессных событий
Предотвращение сетевых событий

Client-side-логика

- Логика идентификации нежелательного действия находится на конечной точке
- С ростом числа базы знаний растет время и количество ресурсов, требуемых на обработку
- Доступна атакующему

Server-side-логика

- Логика идентификации нежелательного действия находится на удаленной точке
- С ростом числа базы знаний не растет нагрузка на хост
- Не доступна такующему

Обзор

- **Базовая технология агента**
- **Обработка данных на user space**
- **Профилирование контейнеров**
- **Отслеживаемые события**
- **Обработка событий**
- **Тип обнаружений**
- **Создание политик/правил**
- **Структура политики/правил**
- **Принцип работы**
- **Привязка политик/правил**
- **Принцип привязки политик/правил**
- **Режим работы**
- **Активное воздействие (reaction)**
- **Расследование инцидента**
- **Предотвращение процессных событий**
- **Предотвращение сетевых событий**

Сигнатурный

- Описание правил, что такое плохо
- Только известные атаки
- Точечный анализ

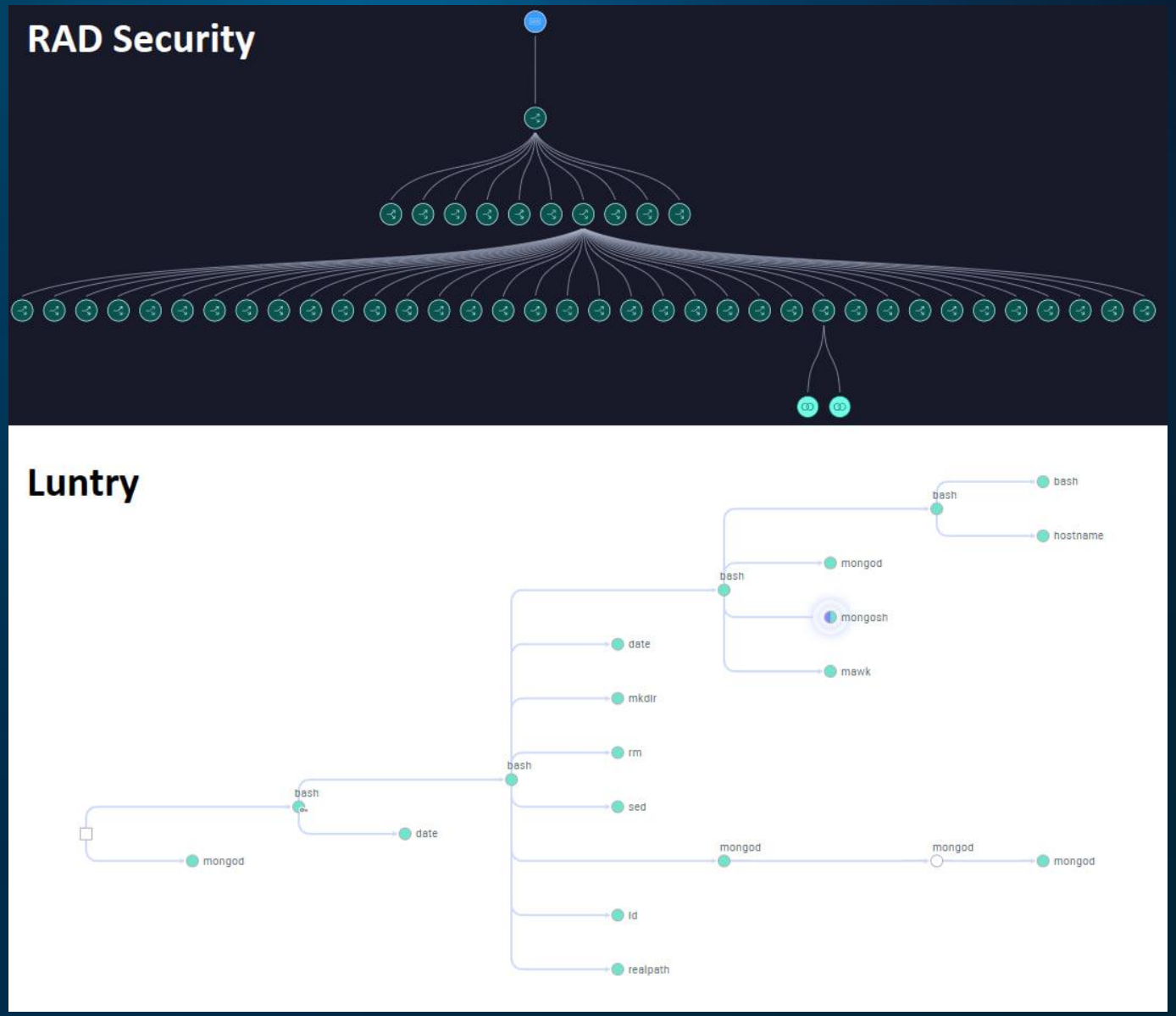
Поведенческий

- Профиль поведения отхождение от которого определяет аномалию
- Способно обнаруживать 1day и 0day

Гибридный

- Преимущества двух предыдущих подходов

Поведенческий анализ



Обзор

Базовая технология агента
Обработка данных на user space
Профилирование контейнеров
Отслеживаемые события
Обработка событий
Тип обнаружений
Создание политик/правил
Структура политики/правил
Принцип работы
Привязка политик/правил
Принцип привязки политик/правил
Режим работы
Активное воздействие (reaction)
Расследование инцидента
Предотвращение процессных событий
Предотвращение сетевых событий

Ручное

- Требует понимания, что происходит внутри микросервиса

Автоматическое

- Требует хорошего покрытия микросервиса для хорошего качества

Гибридное

- Знания оператора + автоматика
- Дообучение

Обзор

Базовая технология агента
Обработка данных на user space
Профилирование контейнеров
Отслеживаемые события
Обработка событий
Тип обнаружений
Создание политик/правил
Структура политики/правил
Принцип работы
Привязка политик/правил
Принцип привязки политик/правил
Режим работы
Активное воздействие (reaction)
Расследование инцидента
Предотвращение процессных событий
Предотвращение сетевых событий

Политики

- Список процессов
  - Слабая гранулярность
    - Легко обойти
- Дерево процессов
  - Высокая гранулярность
    - Сложно обойти

Правила

- YAML
- Go
- Rego
- …

Обзор

- **Базовая технология агента**
- **Обработка данных на user space**
- **Профилирование контейнеров**
- **Отслеживаемые события**
- **Обработка событий**
- **Тип обнаружений**
- **Создание политик/правил**
- **Структура политики/правил**
- **Принцип работы**
- **Привязка политик/правил**
- **Принцип привязки политик/правил**
- **Режим работы**
- **Активное воздействие (reaction)**
- **Расследование инцидента**
- **Предотвращение процессных событий**
- **Предотвращение сетевых событий**

Blacklist

- Описываем, что такое плохо
- Должны заранее знать, что такое плохо
- Можно обнаружить только то, что описали

Whitelist

- Описываем, что такое хорошо
- Должны заранее знать, что такое хорошо
- Можно обнаружить неизвестные атаки

Обзор

Базовая технология агента
Обработка данных на user space
Профилирование контейнеров
Отслеживаемые события
Обработка событий
Тип обнаружений
Создание политик/правил
Структура политики/правил
Принцип работы
Привязка политик/правил
Принцип привязки политик/правил
Режим работы
Активное воздействие (reaction)
Расследование инцидента
Предотвращение процессных событий
Предотвращение сетевых событий

Ручное

- Очень тяжело при большом количестве разных микросервисов

Автоматическое

- Удобно)

Обзор

Базовая технология агента
Обработка данных на user space
Профилирование контейнеров
Отслеживаемые события
Обработка событий
Тип обнаружений
Создание политик/правил
Структура политики/правил
Принцип работы
Привязка политик/правил
Принцип привязки политик/правил
Режим работы
Активное воздействие (reaction)
Расследование инцидента
Предотвращение процессных событий
Предотвращение сетевых событий

- uid
- pid
- mntns
- pidns
- uts
- process
- executable
- Namespace
- Pod label
- Non-container
- Container name
- Image name
- Microservice
- …

Обзор

Базовая технология агента
Обработка данных на user space
Профилирование контейнеров
Отслеживаемые события
Обработка событий
Тип обнаружений
Создание политик/правил
Структура политики/правил
Принцип работы
Привязка политик/правил
Принцип привязки политик/правил
Режим работы
Активное воздействие (reaction)
Расследование инцидента
Предотвращение процессных событий
Предотвращение сетевых событий

Detection

- Уведомление об срабатывании политики или правила

Reaction

- Реагирование на срабатывание политики или правила

Prevention

- Не возможность выполнить, то что не разрешено или запрещено

Обзор

Базовая технология агента
Обработка данных на user space
Профилирование контейнеров
Отслеживаемые события
Обработка событий
Тип обнаружений
Создание политик/правил
Структура политики/правил
Принцип работы
Привязка политик/правил
Принцип привязки политик/правил
Режим работы
Активное воздействие (reaction)
Расследование инцидента
Предотвращение процессных событий
Предотвращение сетевых событий

Прерывание работы

- Завершение процесса (SIGKILL, override)
- Завершение контейнера

Обзор

Базовая технология агента
Обработка данных на user space
Профилирование контейнеров
Отслеживаемые события
Обработка событий
Тип обнаружений
Создание политик/правил
Структура политики/правил
Принцип работы
Привязка политик/правил
Принцип привязки политик/правил
Режим работы
Активное воздействие (reaction)
Расследование инцидента
Предотвращение процессных событий
Предотвращение сетевых событий

Для forensic analysis :

- Дамп RAM
- Дамп FS

Обзор

Базовая технология агента
Обработка данных на user space
Профилирование контейнеров
Отслеживаемые события
Обработка событий
Тип обнаружений
Создание политик/правил
Структура политики/правил
Принцип работы
Привязка политик/правил
Принцип привязки политик/правил
Режим работы
Активное воздействие (reaction)
Расследование инцидента
Предотвращение процессных событий
Предотвращение сетевых событий

Seccomp

- Профиль разрешенный и запрещенных syscalls

LSM

- AppArmor
  - В K8s ограничивает происходящее в контейнере
- SeLinux
  - В K8s ограничивает взаимодействие контейнеров в хостом

eBPF

- eBPF_LSM
  - Linux Kernel 5.7+

Обзор

Базовая технология агента
Обработка данных на user space
Профилирование контейнеров
Отслеживаемые события
Обработка событий
Тип обнаружений
Создание политик/правил
Структура политики/правил
Принцип работы
Привязка политик/правил
Принцип привязки политик/правил
Режим работы
Активное воздействие (reaction)
Расследование инцидента
Предотвращение процессных событий
Предотвращение сетевых событий

Iptables
- Малая гибкость
  - Оперирование CIDR

eBPF
- Малая гибкость
  - Оперирование CIDR

NetworkPolicy
- CNI
  - iptables
  - eBPF
- Декларативный подход
- Большая гибкость
- Понимание контекста окружения

Возможности Open Source решений

Популярные Open Source решения

- Falco
  - Sysdig
  - StackRox*
- Tracee
  - KebeArmor
- Tetragon

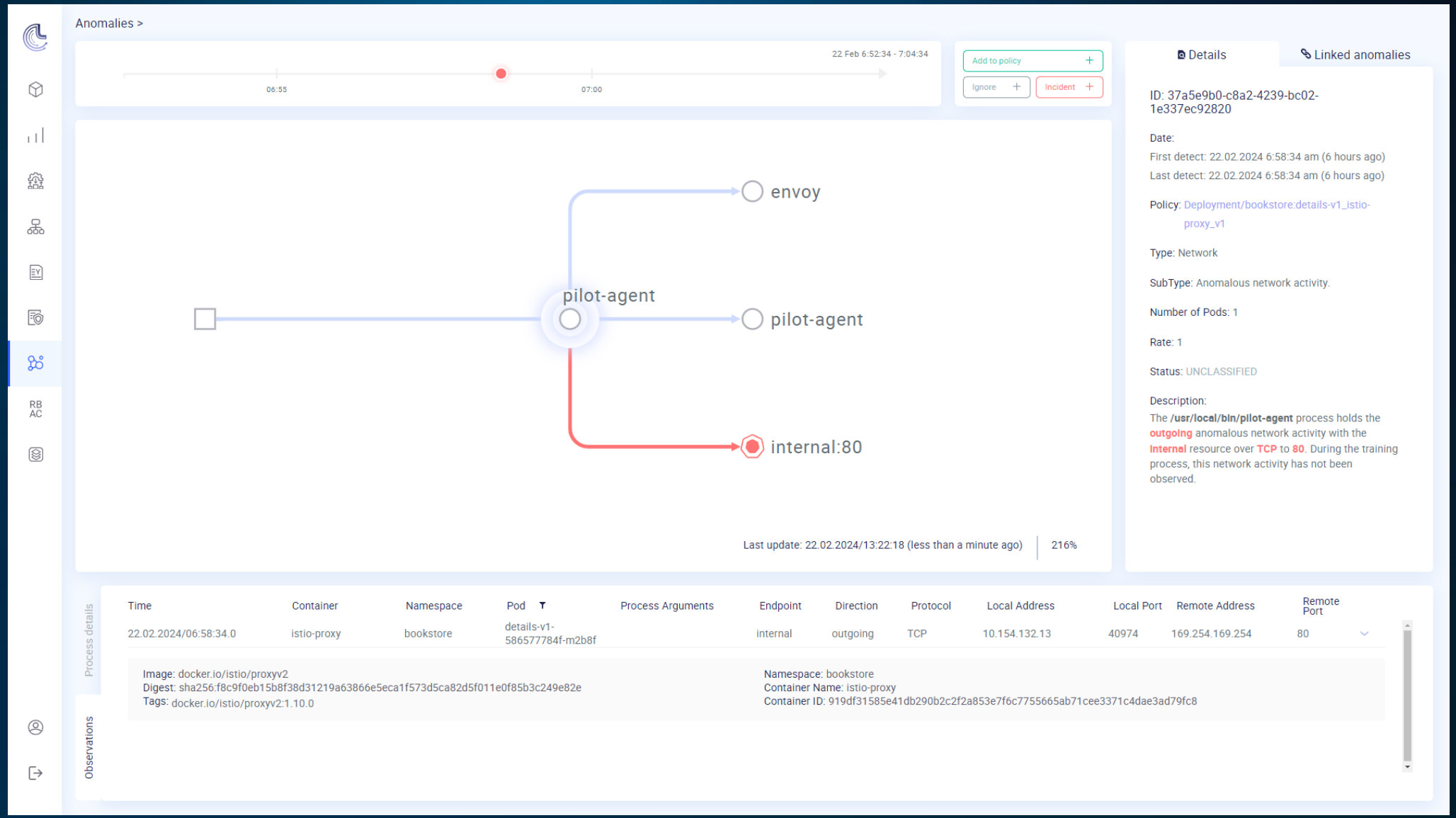
Сравнение OpenSource решений

	Falco	Tracee	Tetragon
Базовая технология агента	eBPF, Kernel module	eBPF	eBPF
Обработка данных на user space	C\C++	Go	Go
Профилирование контейнеров	Нет	Нет	Нет
Отслеживаемые события	syscalls	syscalls, network, security, lsm, containers, misc	syscalls
Обработка событий	Client-side	Client-side	Client-side
Тип обнаружений	Сигнатурный (правила)	Сигнатурный (правила)	Сигнатурный (правила)
Создание политик/правил	Правила в ручную	Правила в ручную	Правила в ручную
Структура политики/правил	YAML	Rego/Go + YAML	YAML
Принцип работы	Blacklist	Blacklist	Blacklist
Привязка политик/правил	Правила в ручную	Правила в ручную	Правила в ручную
Принцип привязки политик/правил	Очень мощный фильтр	Scope	Namespace and pod label filtering (beta)
Режим работы	Detection	Detection	Detection, Reaction
Активное воздействие (reaction)	Нет	Нет	Завершение процесса
Расследование инцидента	Нет	Нет	Нет
Предотвращение процессных событий	Нет	Нет	Да!?*
Предотвращение сетевых событий	Нет	Нет	Да!?*

* - “Tetragone: A Lesson in Security Fundamentals”, grsecurity

Взгляд Luntry

Live Demo


Anomalies >
22 Feb 6:52:34 - 7:04:34
06:55
07:00
Add to policy
Ignore
Incident
Details
Linked anomalies
ID: 37a5e9b0-c8a2-4239-bc02-1e337ec92820
Date:
First detect: 22.02.2024 6:58:34 am (6 hours ago)
Last detect: 22.02.2024 6:58:34 am (6 hours ago)
Policy: Deployment/bookstore:details-v1_istio-proxy_v1
Type: Network
SubType: Anomalous network activity.
Number of Pods: 1
Rate: 1
Status: UNCLASSIFIED
Description:
The /usr/local/bin/pilot-agent process holds the outgoing anomalous network activity with the internal resource over TCP to 80. During the training process, this network activity has not been observed.
envoy
pilot-agent
pilot-agent
internal:80
Last update: 22.02.2024/13:22:18 (less than a minute ago)
216%
RB AC
Process details
Observations
Time
Container
Namespace
Pod
Process Arguments
Endpoint
Direction
Protocol
Local Address
Local Port
Remote Address
Remote Port
22.02.2024/06:58:34.0
istio-proxy
bookstore
details-v1-586577784f-m2b8f
internal
outgoing
TCP
10.154.132.13
40974
169.254.169.254
80
Image: docker.io/istio/proxyv2
Digest: sha256:f8c9f0eb15b8f38d31219a63866e5eca1f573d5ca82d5f011e0f85b3c249e82e
Tags: docker.io/istio/proxyv2:1.10.0
Namespace: bookstore
Container Name: istio-proxy
Container ID: 919df31585e41db290b2c2f2a853e7f6c7755665ab71cee3371c4dae3ad79fc8

Характеристики Luntry Runtime Security

	Luntry
Базовая технология агента	eBPF
Обработка данных на user space	Rust
Профилирование контейнеров	Автоматическое
Отслеживаемые события	Process, Network, Files
Обработка событий	Server-side
Тип обнаружений	Поведенческий
Создание политик/правил	Гибридное (автоматическое + ручное)
Структура политики/правил	Дерево процессов
Принцип работы	Whitelist
Привязка политик/правил	Автоматическое (политик)
Принцип привязки политик/правил	К микросервису
Режим работы	Detection, Reaction, Prevention
Активное воздействие (reaction)	Остановка контейнера
Расследование инцидента	Дамп FS и RAM контейнера
Предотвращение процессных событий	Генерация AppArmor профиля
Предотвращение сетевых событий	Генерация NetworkPolicy (Native, Cilium, Calico)

Дорожная карта развития

- Обнаружение на основе правил для процессных, файловых и сетевых событий
  - С пред заготовленной библиотекой правил
- Уровни критичности событий
  - Для приоритезации
- Гибридный подход для обнаружения угроз
  - Поведение + правила
- Реализация блокирующих политик
  - На eBPF
- …

Итог

1. Runtime Security очень многогранна
2. Kubernetes вносит свою специфику
3. Важно понимать кто и как это будет использовать и обсуживать в компании
4. Для защиты Runtime есть и другие подходы на уровне образов, контейнеров и Kubernetes ;)

Спасибо
за внимание!

Дмитрий Евдокимов
Founder&CTO

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

k8security luntrysolution