LUNTRY

luntry.ru

Вебинар

# Ловим злоумышленников и собираем улики в контейнерах Kubernetes

Дмитрий Евдокимов

Founder&CTO Luntry

Сергей Канибор

R&D / Container Security, Luntry

# whoami

- Основатель и технический директор Luntry
- Опыт в ИБ более 15 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”
  (подробнее: https://slurm.io/kubernetes-security)
- 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.

LUNTRY
luntry.ru

# whoami

- R&D / Container Security в Luntry
- Специализация безопасность контейнеров и Kubernetes
- Багхантер
- Редактор telegram канала “k8s (in)security”
- Докладчик: PHDays, OFFZONE, VK Kubernetes Conf, Devoops, HackConf, CyberCamp, БеКон и др.

LUNTRY
luntry.ru

# 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 конец 2024

LUNTRY
luntry.ru

# Функциональность Luntry

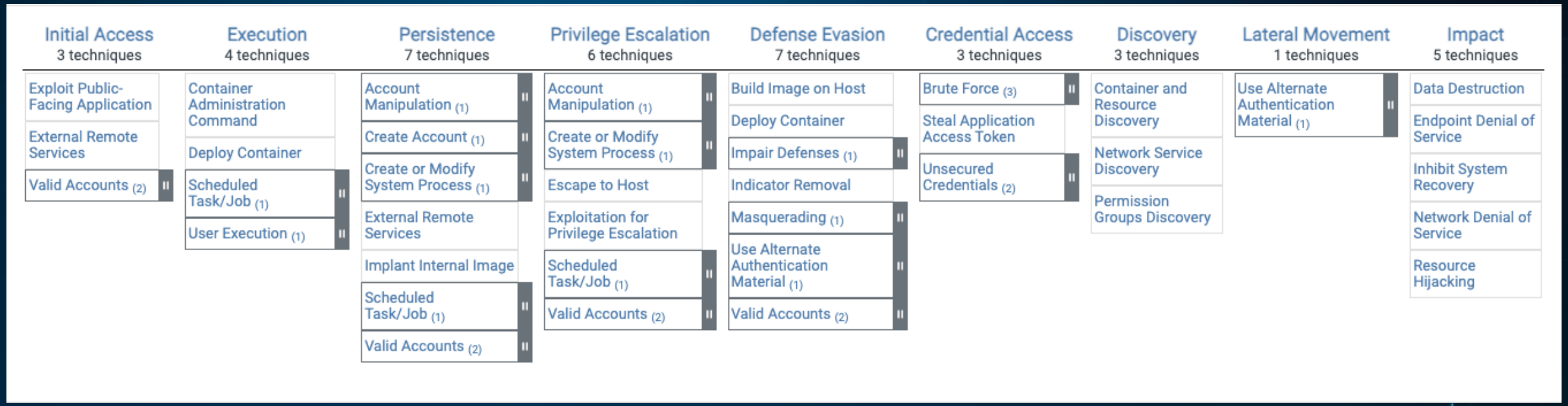
# Функциональность Luntry

# План вебинара

- Матрицы атак
- Способы обнаружения
- Концентрируемся на Runtime
- Обнаружение
  - Продвинутое обнаружение
- Реакция
- Взгляд Luntry

# Матрицы атак

# MITRE ATT&CK Container Matrix



# MITRE ATT&CK Container Matrix – минусы

- Техники довольно абстрактны и не сильно погружены в контекст Kubernetes
- Самих техники сильно меньше по сравнению с другими матрицами
- Можно расценивать как инструмент, с помощью которого можно узнать об определенных процедурах

# Threat Matrix for Kubernetes от Microsoft

| Initial Access | Execution | Persistence | Privilege Escalation | Defense Evasion | Credential Access | Discovery | Lateral Movement | Collection | Impact |
|---|---|---|---|---|---|---|---|---|---|
| Using cloud credentials | Exec into container | Backdoor container | Privileged container | Clear container logs | List K8S secrets | Access Kubernetes API server | Access cloud resources | Images from a private registry | Data destruction |
| Compromised image In registry | bash/cmd inside container | Writable hostPath mount | Cluster-admin binding | Delete K8S events | Mount service principal | Access Kubelet API | Container service account | Collecting data from pod | Resource hijacking |
| Kubeconfig file | New container | Kubernetes CronJob | hostPath mount | Pod / container name similarity | Container service account | Network mapping | Cluster internal networking | | Denial of service |
| Application vulnerability | Application exploit (RCE) | Malicious admission controller | Access cloud resources | Connect from proxy server | Application credentials in configuration files | Exposed sensitive interfaces | Application credentials in configuration files | | |
| Exposed sensitive interfaces | SSH server running inside container | Container service account | | | Access managed identity credentials | Instance Metadata API | Writable hostPath mount | | |
| | Sidecar injection | Static pods | | | Malicious admission controller | | CoreDNS poisoning | | |
| | | | | | | | ARP poisoning and IP spoofing | | |

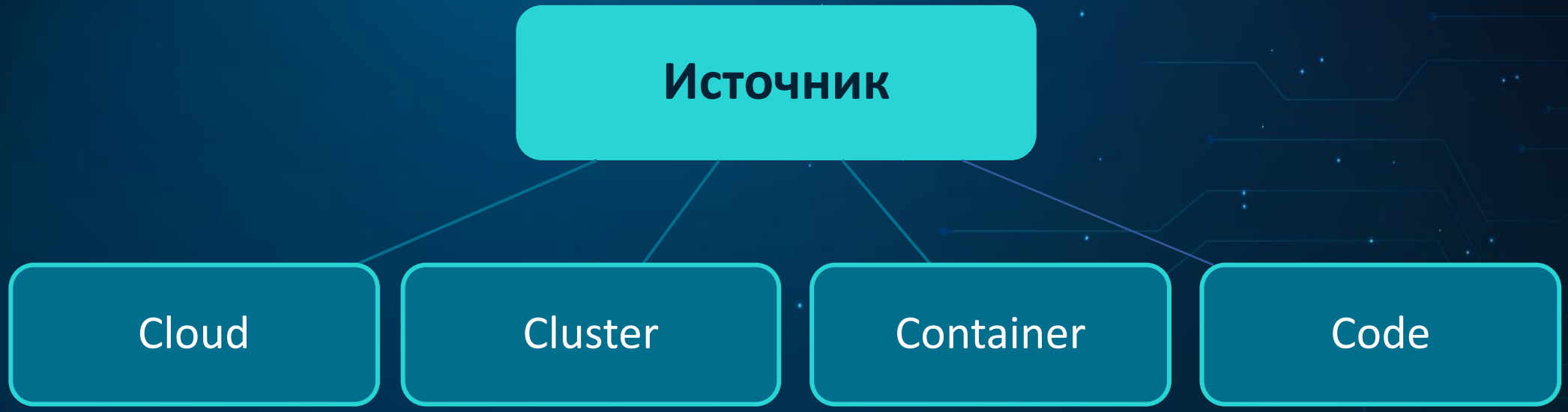
# Threat Matrix for Kubernetes от Microsoft

- Понятные и хорошо описанные техники в контексте Kubernetes
- Есть маппинг на техники, описанные в MITRE
- Небольшой акцент на Managed K8s
- Как и в любой другой матрице, **атакующий всегда на шаг впереди**

# VK Kubernetes Conf 2023. Экскурсия по матрицам угроз для контейнеров и Kubernetes (Сергей Канибор, Luntry)

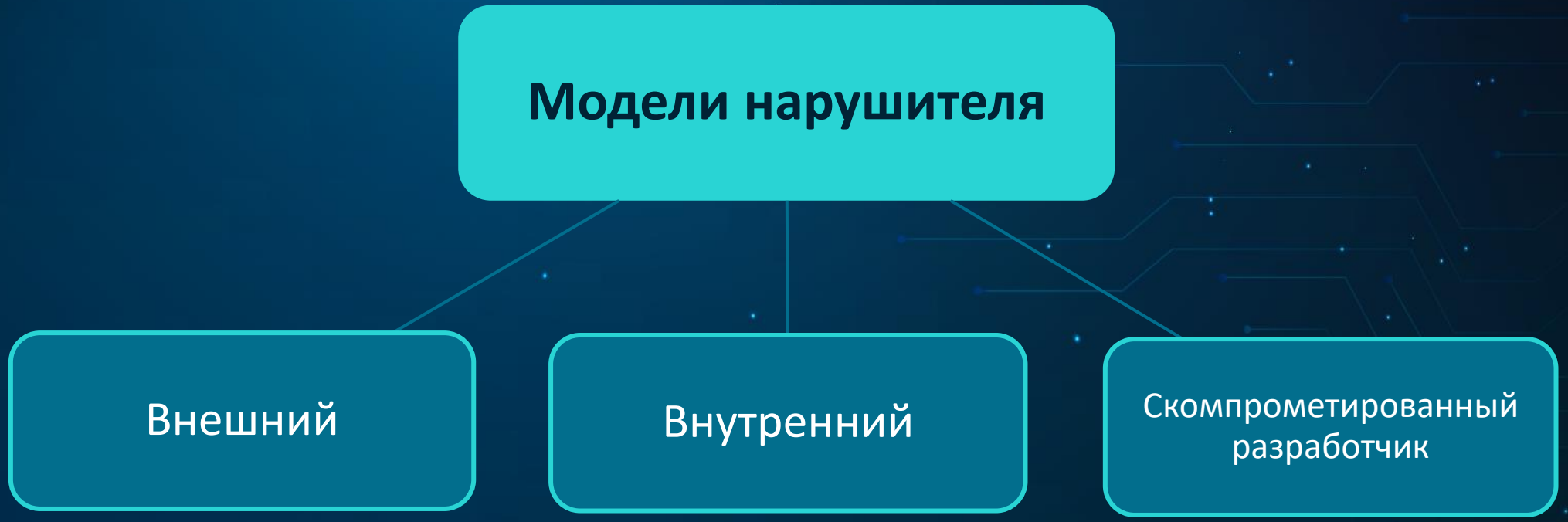
# Способы обнаружения

# Источники данных




- Логи облачного провайдера
- Kubernetes Audit Log
- Runtime события
- Логи приложения

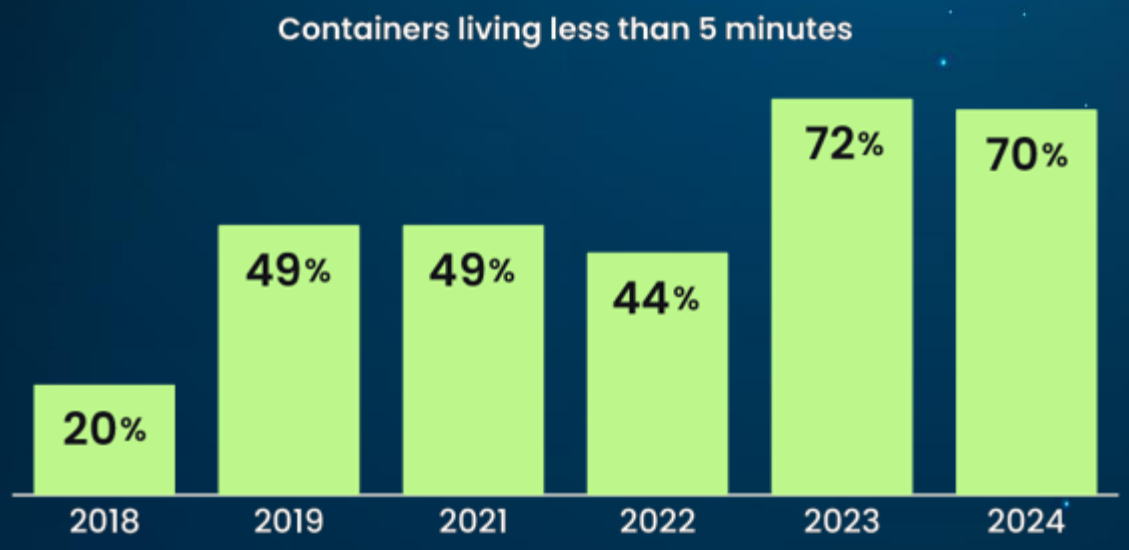
# Модели нарушителя



# KazHackStan 2022. Специфика расследования инцидентов в контейнерах(Дмитрий Евдокимов, Luntry)

# Концентрируемся на Runtime

# Динамическое окружение




Sysdig 2024 Cloud-Native Security and Usage Report

- Малый срок жизни контейнеров
- Self-healing
- Следы злоумышленника в контейнере очищаются сами собой

# Виды защиты (Linux World)

- Isolation
  - Дополнительный уровень изоляции от ядра Host ОС (WASM, Sandbox, microVM, …)
- Detection
  - Идентификация нежелательного действия
- Prevention
  - Невозможность выполнения нежелательного действия
- Mitigation
  - Смягчение последствий нежелательного действия
- Reaction
  - Ответ на нежелательное действие постфактум после нежелательного события

# Runtime Security: на вкус и цвет все фломастеры разные (Дмитрий Евдокимов, Luntry)

# Обнаружение


сигнатуры

# Сигнатуры

```
- rule: Execution from /dev/shm
  desc: >
    This rule detects file execution in the /dev/shm directory, a tactic often used by threat actors to store the
    occasionally executable files. /dev/shm acts as a link to the host or other containers, creating vulnerabilit
    as well. Notably, /dev/shm remains unchanged even after a container restart. Consider this rule alongside the
    "Drop and execute new binary in container" rule.
  condition: >
    spawned_process
    and (proc.exe startswith "/dev/shm/" or
        (proc.cwd startswith "/dev/shm/" and proc.exe startswith "./" ) or
        (shell_procs and proc.args startswith "-c /dev/shm") or
        (shell_procs and proc.args startswith "-i /dev/shm") or
        (shell_procs and proc.args startswith "/dev/shm") or
        (proc.cwd startswith "/dev/shm/" and proc.args startswith "./" ))
    and not container.image.repository in (falco_privileged_images, trusted_images)
  output: File execution detected from /dev/shm (evt_res=%evt.res file=%fd.name proc_cwd=%proc.cwd proc_pcmdline=
  priority: WARNING
  tags: [maturity_stable, host, container, mitre_execution, T1059.004]
```

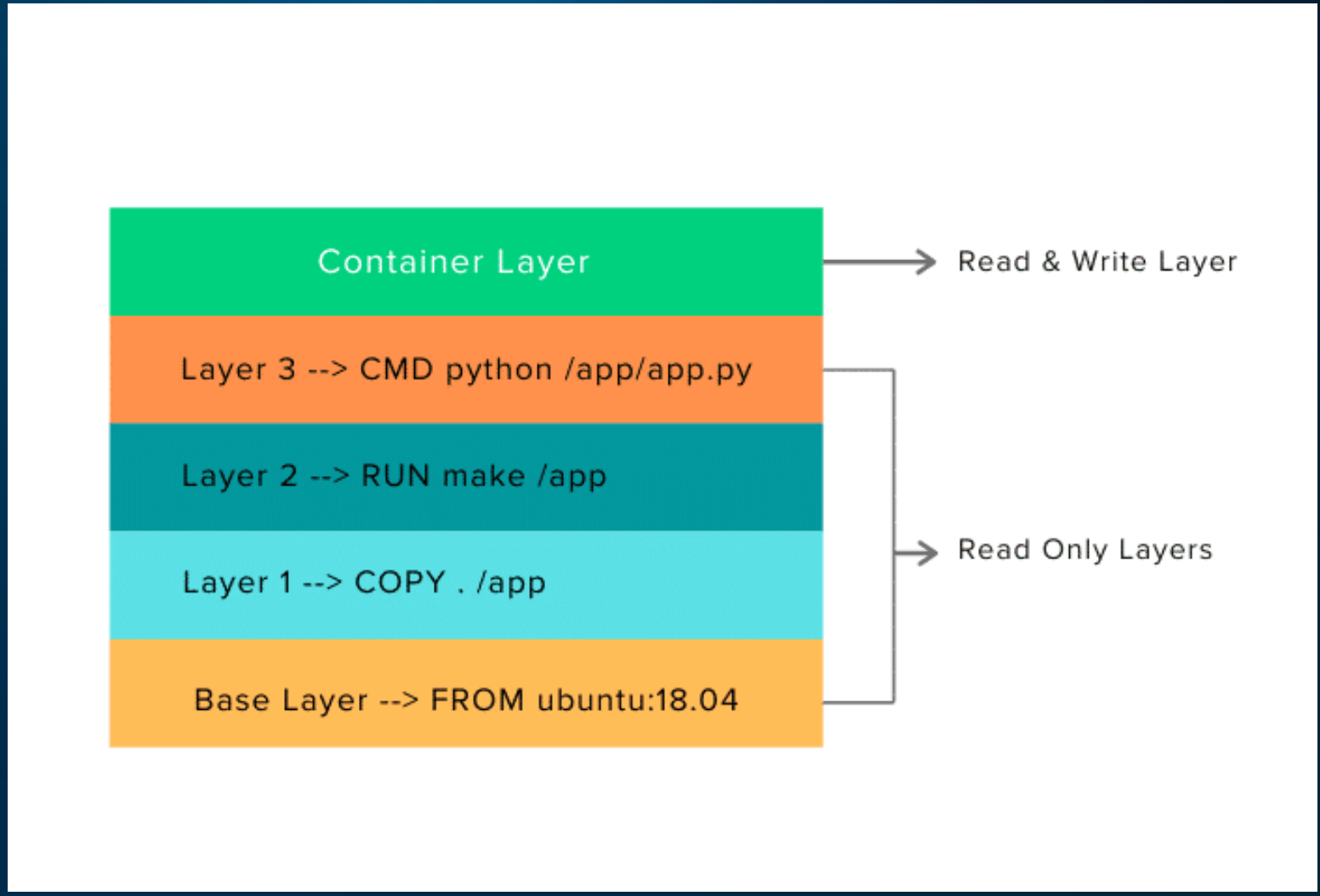
```
apiVersion: tracee.aquasec.com/v1beta1
kind: Policy
metadata:
  name: dig
  annotations:
    description: traces dns events from the dig executable
spec:
  scope:
    - executable=/usr/bin/dig
  rules:
    - event: net_packet_dns_request
    - event: net_packet_dns_response
```

```
apiVersion: cilium.io/v1alpha1
kind: TracingPolicy
metadata:
  name: "cve-2024-3094-xz-ssh"
  annotations:
    url: "https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-3094"
    description: "Detects if OpenSSH is using vulnerable XZ libraries"
    author: "Tetragon.io Team"
spec:
  kprobes:
  - call: "security_mmap_file"
    syscall: false
    return: true
    # message: "OpenSSH daemon using vulnerable XZ libraries CVE-2024-3094"
    # tags: [ "cve", "cve.2024.3094" ]
    args:
    - index: 0
      type: "file"
    - index: 1
      type: "uint32"
    - index: 2
      type: "nop"
    returnArg:
      index: 0
      type: "int"
    returnArgAction: "Post"
    selectors:
    - matchBinaries:
      - operator: "In"
        values:
        - "/usr/sbin/sshd"
      matchArgs:
      - index: 0
        operator: "Postfix"
        values:
        - "liblzma.so.5.6.0"
        - "liblzma.so.5.6.1"
      matchActions:
        - action: Post
          rateLimit: "1m"
```

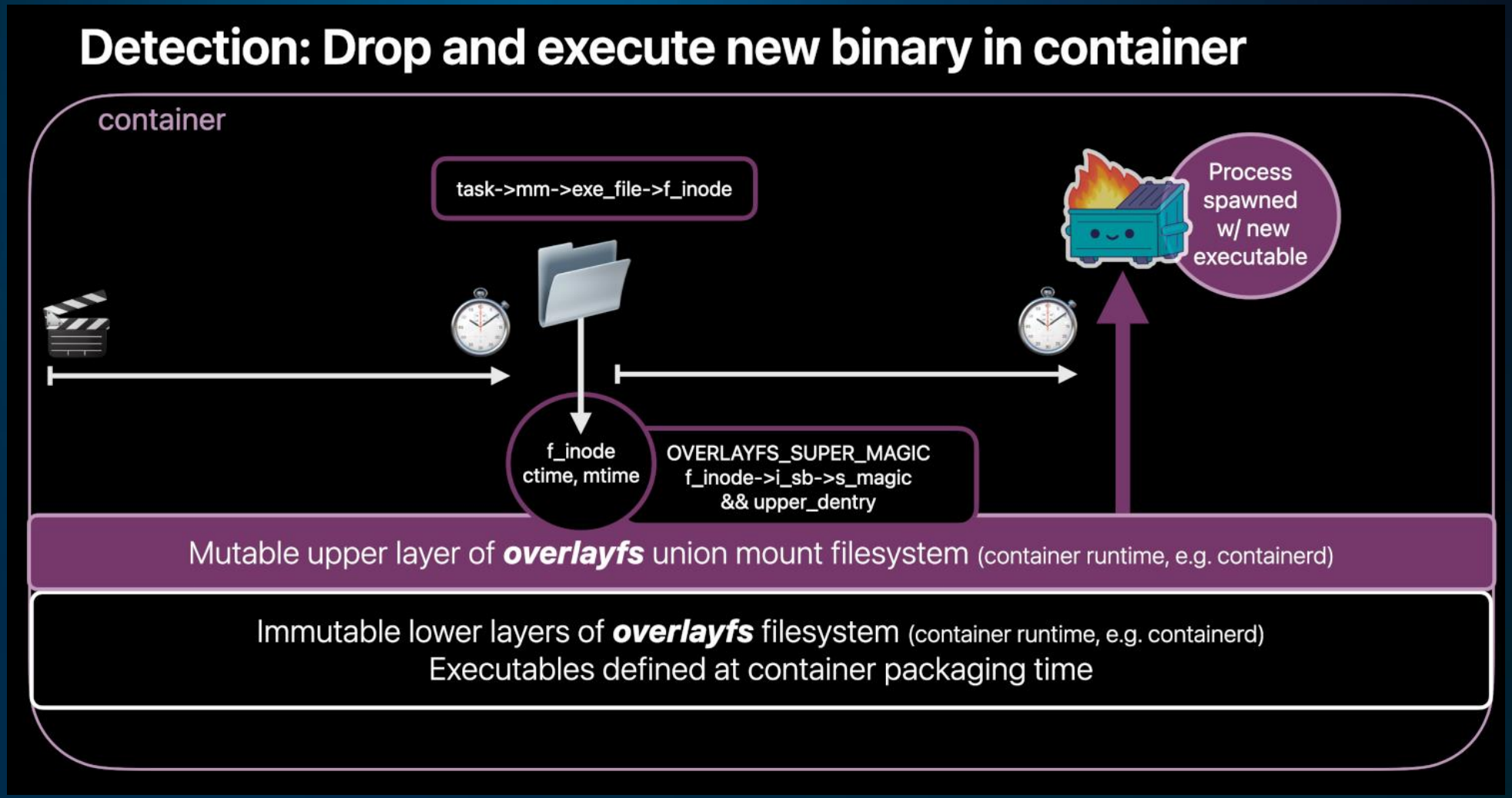
# Концентрируемся на специфике контейнеров

- Играем на свойствах OverlayFS ;)

# Что такое OverlayFS?



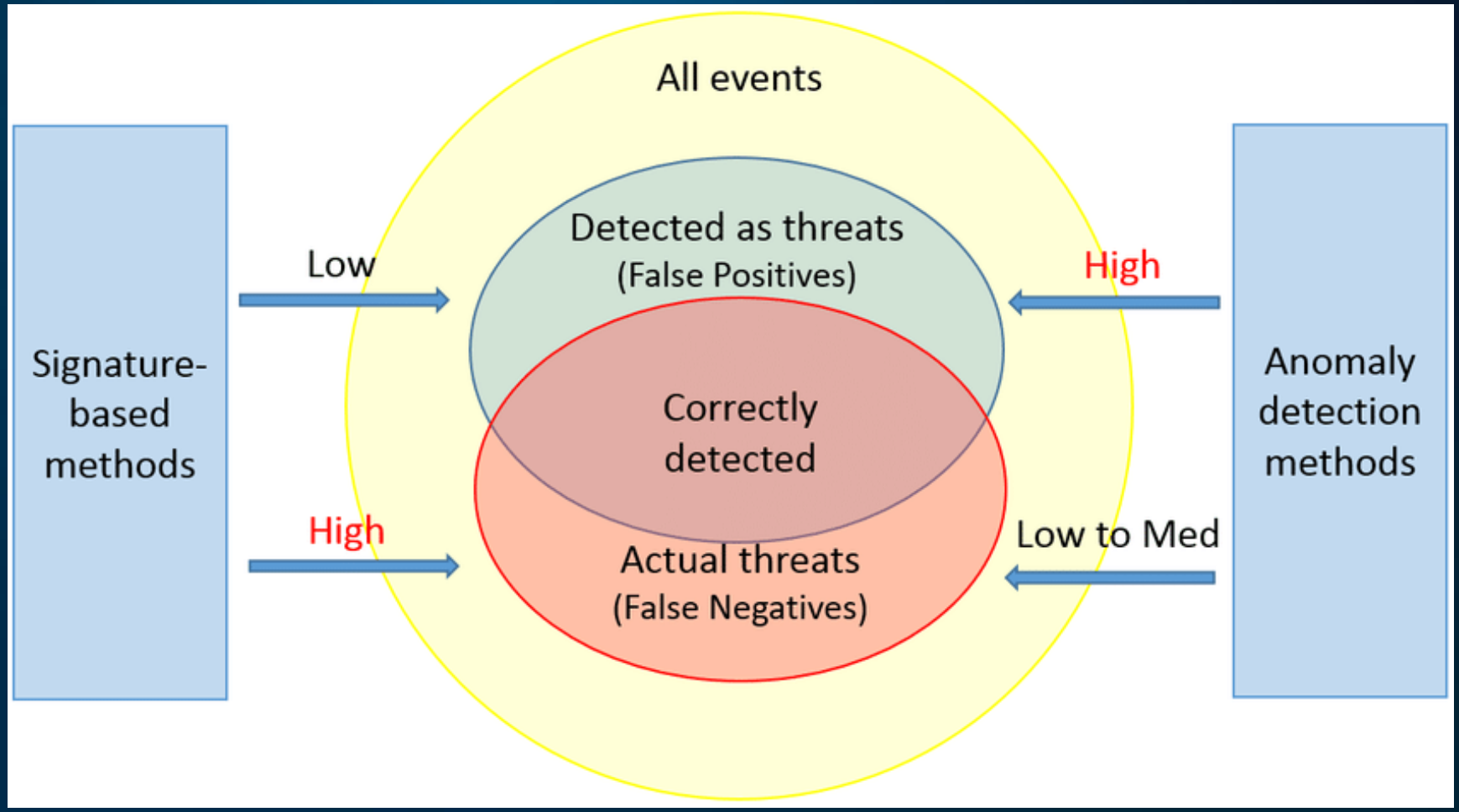
# Обнаружение новых бинарей в Upper Layer



# Обход new binary execution [Falco]

- Для обнаружения правило матчит событие по execve/execveat
- Чтобы обойти можно использовать GTFO bin, например ld.so
- Или воспользоваться техникой fileless execution

# Signature Based VS Anomaly Based



# SOC-форум 2023. EDR vs Containers: актуальные проблемы (Владислав Лашкин, Solar; Дмитрий Евдокимов, Luntry)

# Реакция

# Forensic Container Checkpointing в k8s

- С Kubernetes v1.25 (alpha)
- Базируется на Checkpoint/Restore In Userspace (CRIU)
- Требуется включение ContainerCheckpoint feature gate на API server
- Требуется поддержка на стороне Container Runtime (В CRI-O уже есть, в containerd в процессе)
- Для создания checkpoint требуется обращение к kubelet с Node
- Цепочка вызовов kubelet -> High-level runtime - > Low-level runtime - > criu
- Результат работы сохраняется в /var/lib/kubelet/checkpoints/checkpoint-<pod-name>_<namespace-name>-<container-name>-<timestamp>.tar
- Полученный Checkpoint можно восстановить как в Kubernetes, так и за его пределами

```
curl -X POST "https://localhost:10250/checkpoint/namespace/podId/container"
```

For a container named *counter* in a pod named *counters* in a namespace named *default* the **kubelet** API endpoint is reachable at:

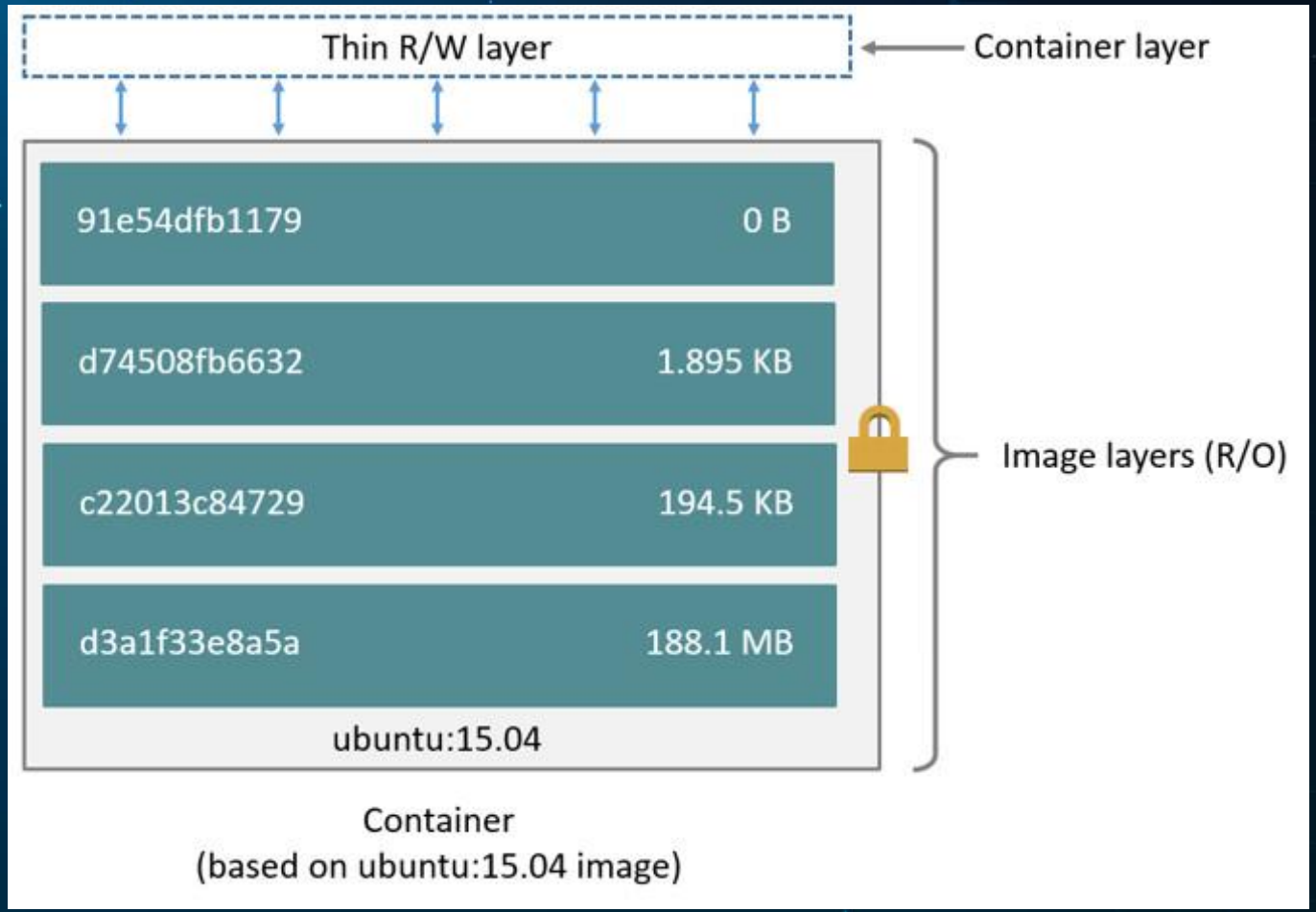
```
curl -X POST "https://localhost:10250/checkpoint/default/counters/counter"
```

For completeness the following `curl` command-line options are necessary to have `curl` accept the *kubelet*'s self signed certificate and authorize the use of the *kubelet* `checkpoint` API:

```
--insecure --cert /var/run/kubernetes/client-admin.crt --key /var/run/kubernetes/client-admin
```

# Дамп ФС

- Нет смысла дампать всю ФС целиком
- Нижние слои могут очень много весить
- Злоумышленник может взаимодействовать только с upper layer



# Убийство контейнера

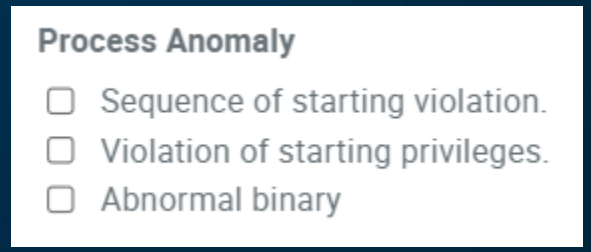
- Атакующий мог породить другие потоки, оставить для себя бэкдоры
- Такой контейнер это уже скомпрометированная среда

# Взгляд Luntry

# Обнаружение новых исполняемых файлов

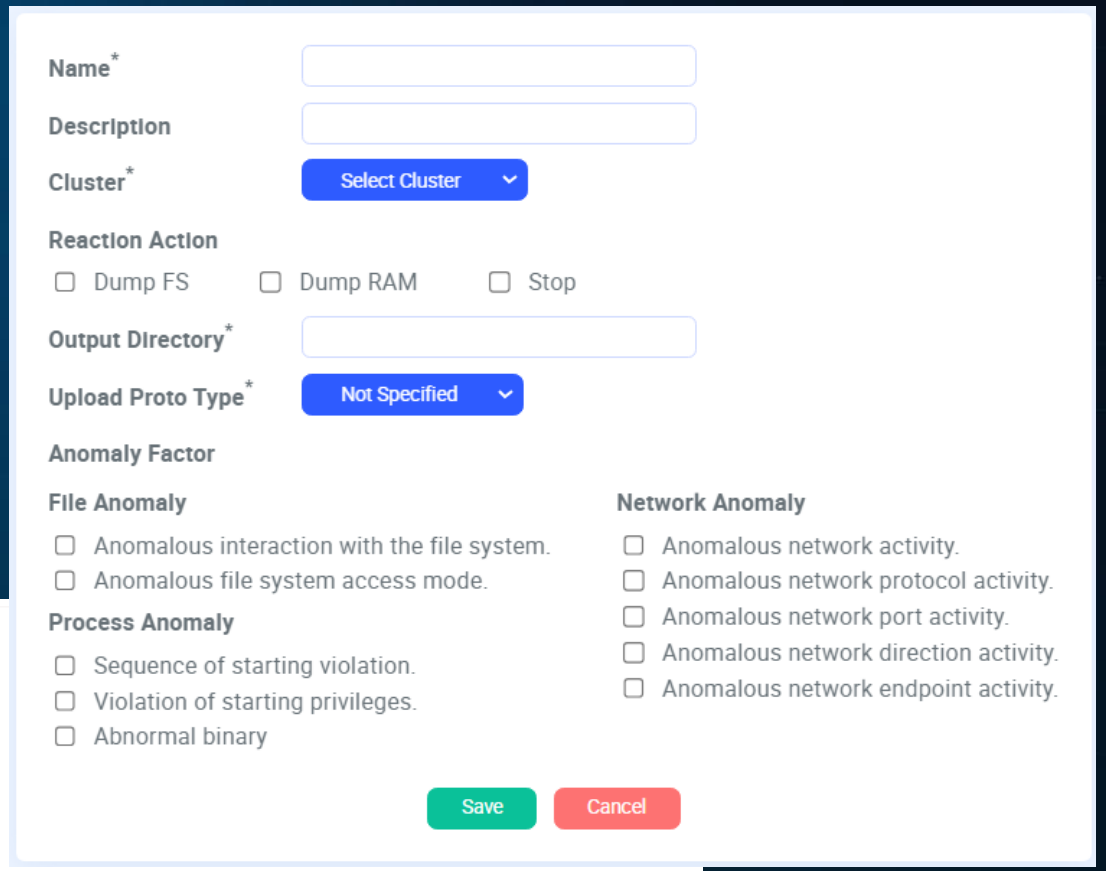
## Abnormal binary

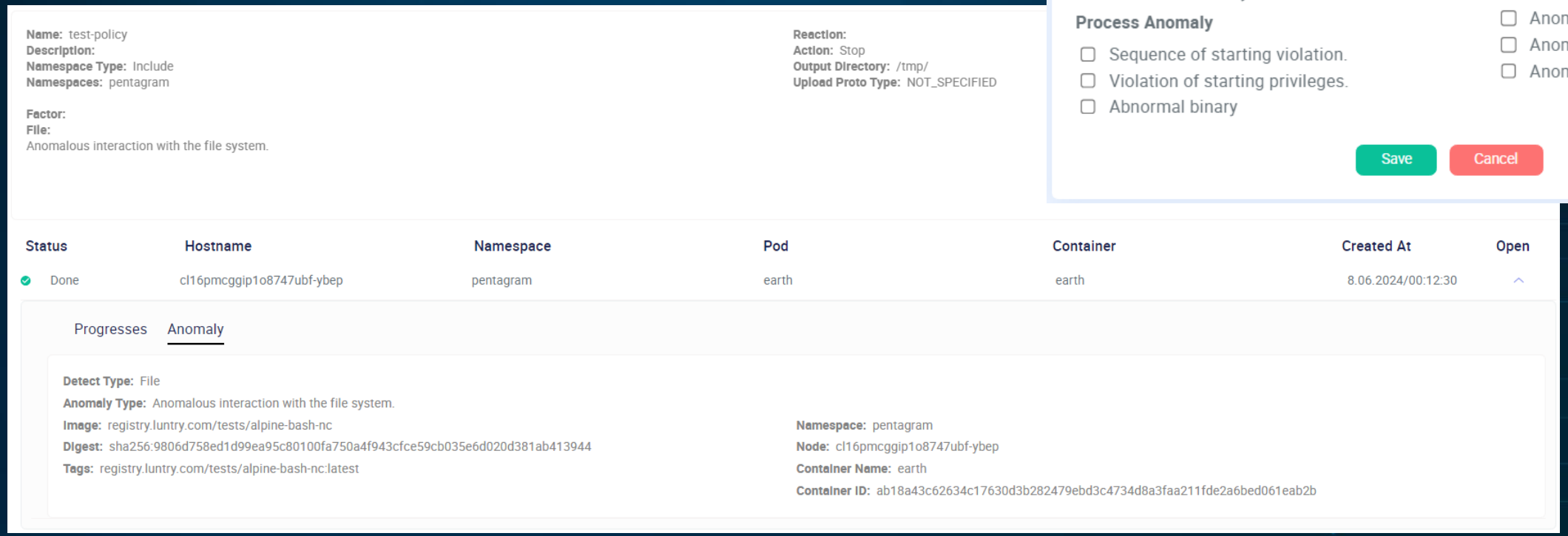
- Позволяет определить запуск в контейнере исполняемого файла, которого не было в оригинальном образе
  - Кто-то что-то до устанавливает (apt-get и т.д.) в runtime и запускает
  - Кто-то что-то переименовывает для обхода сигнатурного подхода (bash в nginx)
  - Кто-то что-то докачивает в контейнер через reverse shell или другим способом



# Политики для реакций

- На cluster
- На namespace
- На аномалию



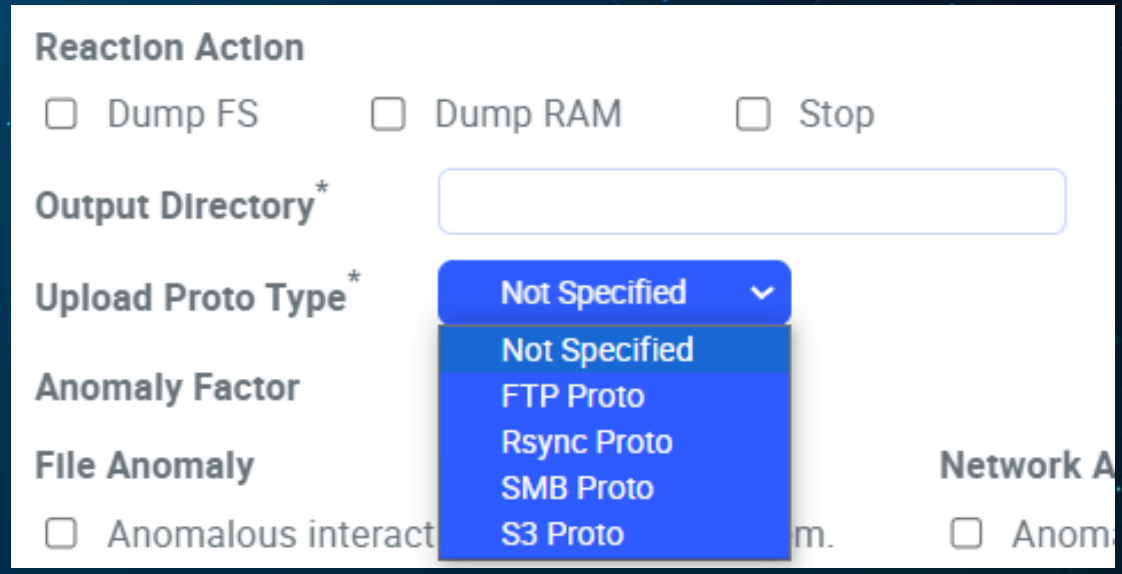
# Артефакты для расследования

Артефакт

- Дамп верхнего слоя ФС
- Дамп оперативной памяти контейнера

Отправка артефакта

- FTP
- Rsync
- SMB
- S3



# Дорожная карта развития для Runtime

- 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. Классические подходы в контейнерах не эффективны
2. Нужно использовать специфику контейнеров
3. Будущее за сочетанием правил и аномалий
4. Высокая важность сбора артефактов для расследования

# Спасибо за внимание!


Дмитрий Евдокимов
Founder&CTO


Email: de@luntry.ru


Twitter: @evdokimovds

@Qu3b3c


Channel: @k8security


Site: www.luntry.ru

Сергей Канибор
R&D / Container Security

Email: sk@luntry.ru


k8security luntrysolution