Специфика расследования инцидентов в контейнерах

Дмитрий Евдокимов
Founder&CTO Luntry

2022
ALMATY
KAZAKHSTAN

About me

- Основатель и технический директор Luntry
- Опыт в ИБ более 10 лет
- Соорганизатор конференций ZeroNights, DEFCON Russia (#7812)
- 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
- Докладчик: BlackHat, HITB, ZeroNights, HackInParis, Confidence, SAS, OFFZONE, PHDays, Kazhackstan, DevOpsConf, KuberConf, VK Kubernetes Conference, HighLoad++ и др.

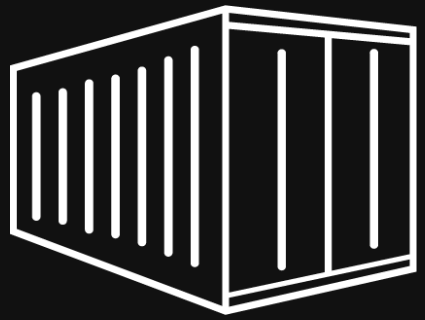
Дмитрий Евдокимов

Founder, CTO

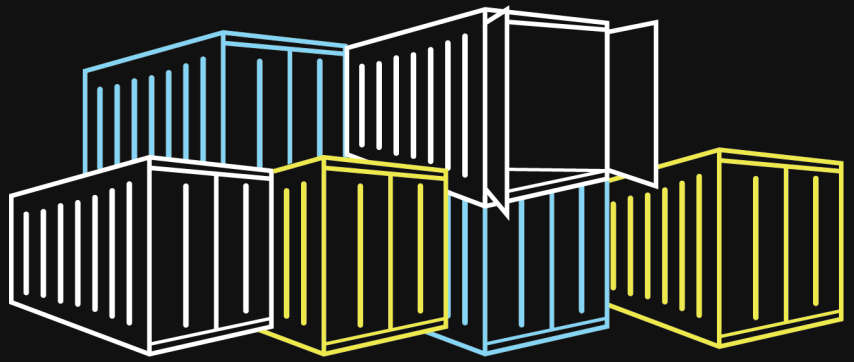
Agenda

Окружение

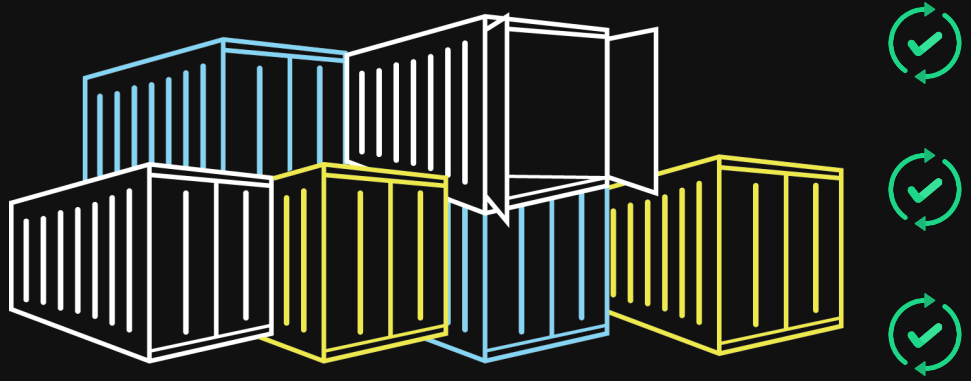
Окружение



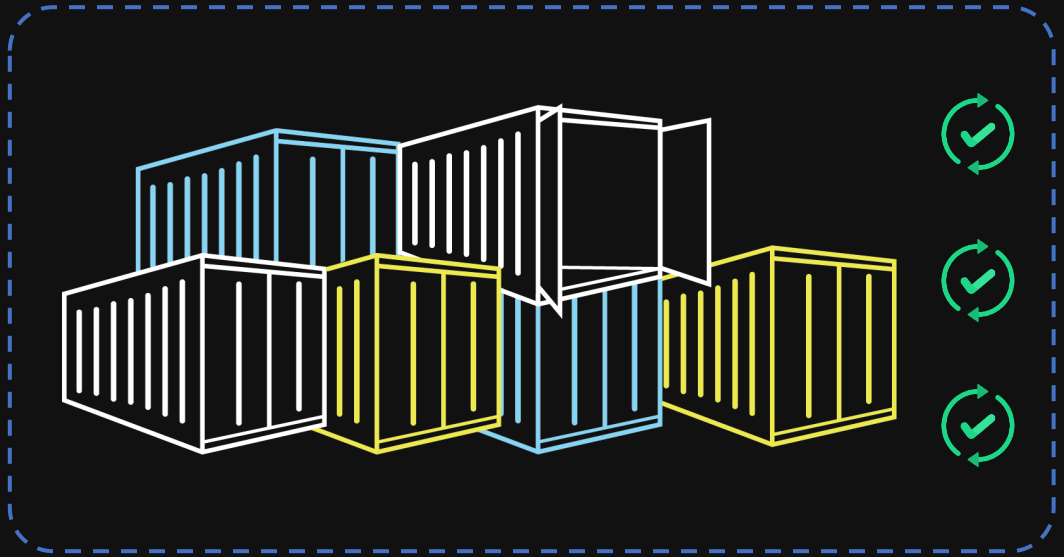
Окружение



Окружение



Окружение

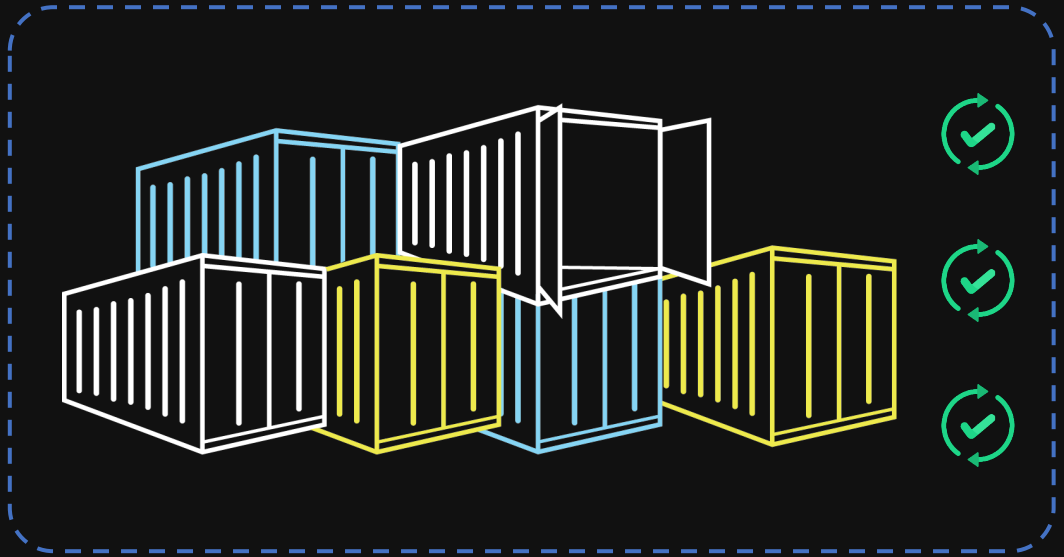


kubernetes

- Platform as a Service (PaaS)
- Configuration
- Function
- Applications
- Runtime
- Containers
- Operation Systems
- Hardware

Окружение

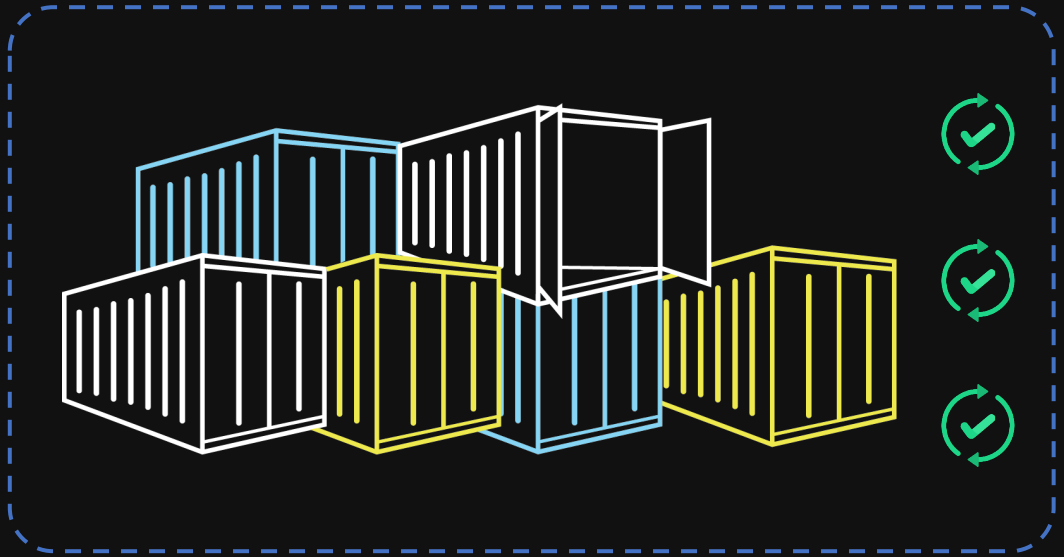
kubernetes
kubernetes
kubernetes



- Platform as a Service (PaaS)
- Configuration
- Function
- Applications
- Runtime
- Containers
- Operation Systems
- Hardware

Окружение

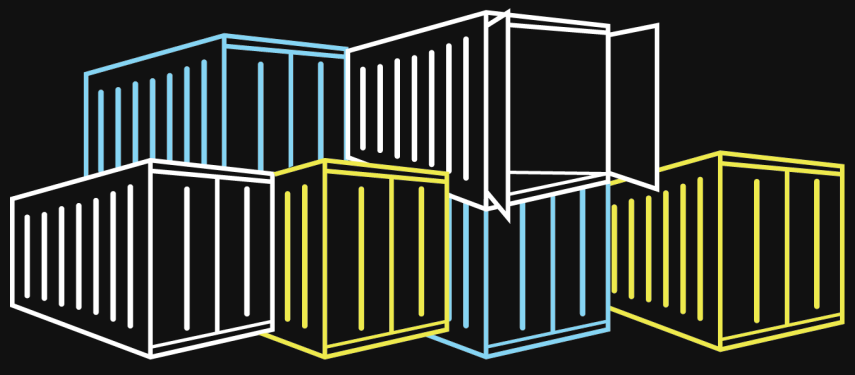
kubernetes
kubernetes
kubernetes



- Platform as a Service (PaaS)
- Configuration
- Function
- Applications
- Runtime
- Containers
- Operation Systems
- Hardware

Google Cloud
Microsoft Azure
Yandex Cloud
RED HAT OPENSHIFT
SBERCLOUD
VK Cloud Solutions
Amazon EKS

Окружение



Разработчики
Аналитики

DevOps
QA

SRE
Monitoring

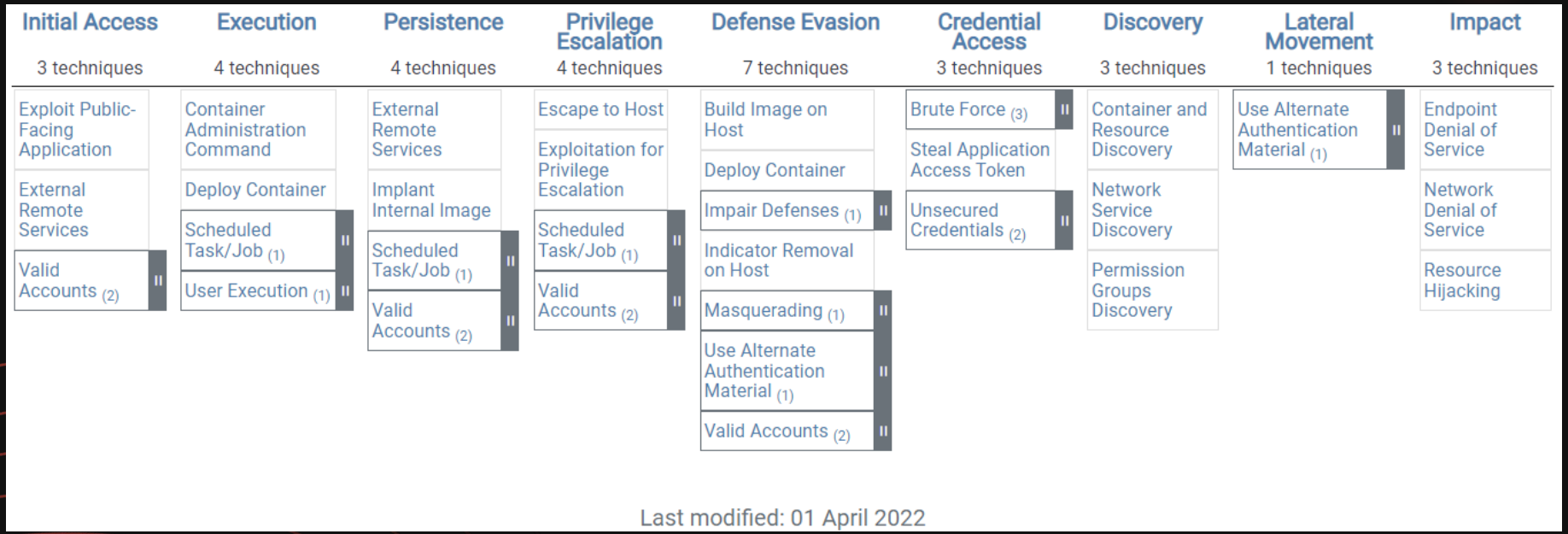
kubernetes
kubernetes
kubernetes

Platform as a Service (PaaS)
Configuration
Function
Applications
Runtime
Containers
Operation Systems
Hardware

Google Cloud
Microsoft Azure
Yandex Cloud
RED HAT OPENSHIFT
SBERCLOUD
VK Cloud Solutions
Amazon EKS

DFIR в k8s

MITRE ATT&CK®Containers Matrix



Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Impact
3 techniques	4 techniques	4 techniques	4 techniques	7 techniques	3 techniques	3 techniques	1 techniques	3 techniques
Exploit Public-Facing Application	Container Administration Command	External Remote Services	Escape to Host	Build Image on Host	Brute Force (3)	Container and Resource Discovery	Use Alternate Authentication Material (1)	Endpoint Denial of Service
External Remote Services	Deploy Container	Implant Internal Image	Exploitation for Privilege Escalation	Deploy Container	Steal Application Access Token	Network Service Discovery		Network Denial of Service
Valid Accounts (2)	Scheduled Task/Job (1)	Scheduled Task/Job (1)	Scheduled Task/Job (1)	Impair Defenses (1)	Unsecured Credentials (2)	Permission Groups Discovery		Resource Hijacking
	User Execution (1)	Valid Accounts (2)	Valid Accounts (2)	Indicator Removal on Host				
				Masquerading (1)				
				Use Alternate Authentication Material (1)				
				Valid Accounts (2)				

Last modified: 01 April 2022

[Link]

Threat matrix для 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 the K8S API server	Access cloud resources	Images from a private registry	Data Destruction
Compromised images in registry	bash/cmd inside container	Writable hostPath mount	Cluster-admin binding	Delete K8S events	Mount service principal	Access Kubelet API	Container service account		Resource Hijacking
Kubeconfig file	New container	Kubernetes CronJob	hostPath mount	Pod / container name similarity	Access 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	Applications credentials in configuration files	Access Kubernetes dashboard	Applications credentials in configuration files		
Exposed Dashboard	SSH server running inside container				Access managed identity credential	Instance Metadata API	Writable volume mounts on the host		
Exposed sensitive interfaces	Sidecar injection				Malicious admission controller		Access Kubernetes dashboard		
							Access tiller endpoint		
							CoreDNS poisoning		
							ARP poisoning and IP spoofing		

= New technique

= Deprecated technique

Link

DFIR (Digital Forensics and Incident Response)

- DFIR Live
  - Container
- DFIR Offline
  - Logs
  - Dumps

IMPROVEMENT
Prevent future episode

IDENTIFICATION
Fast Detection Process

INCIDENT
RESPONSE PLAN

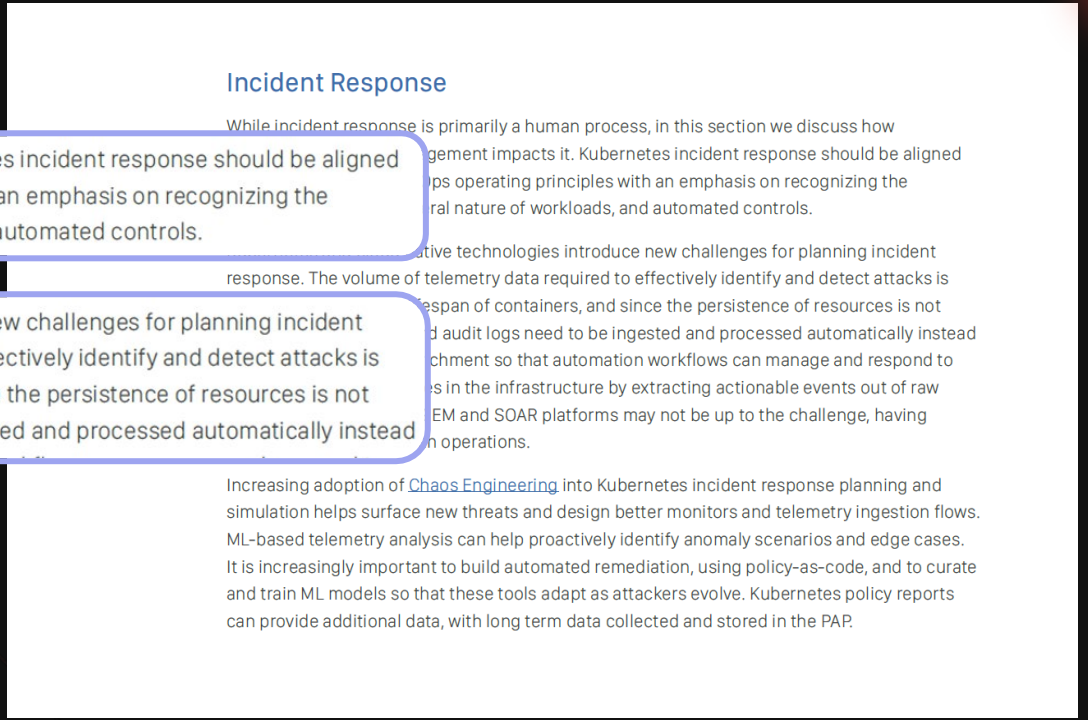
RESOLUTION
Minimize Impact

COORDINATION
Reduce Risk

IR в Kubernetes

Kuberenetes policy management impacts it. Kubernetes incident response should be aligned with established DevSecOps operating principles with an emphasis on recognizing the declarative state, ephemeral nature of workloads, and automated controls.

Kubernetes and cloud native technologies introduce new challenges for planning incident response. The volume of telemetry data required to effectively identify and detect attacks is larger due to the short lifespan of containers, and since the persistence of resources is not guaranteed. Telemetry and audit logs need to be ingested and processed automatically instead



Incident Response

While incident response is primarily a human process, in this section we discuss how gement impacts it. Kubernetes incident response should be aligned ps operating principles with an emphasis on recognizing the ral nature of workloads, and automated controls.

tive technologies introduce new challenges for planning incident response. The volume of telemetry data required to effectively identify and detect attacks is espan of containers, and since the persistence of resources is not d audit logs need to be ingested and processed automatically instead chment so that automation workflows can manage and respond to s in the infrastructure by extracting actionable events out of raw EM and SOAR platforms may not be up to the challenge, having n operations.

Increasing adoption of Chaos Engineering into Kubernetes incident response planning and simulation helps surface new threats and design better monitors and telemetry ingestion flows. ML-based telemetry analysis can help proactively identify anomaly scenarios and edge cases. It is increasingly important to build automated remediation, using policy-as-code, and to curate and train ML models so that these tools adapt as attackers evolve. Kubernetes policy reports can provide additional data, with long term data collected and stored in the PAP.

«Kubernetes Policy Management»

ПРОБЛЕМЫ DFIR в k8s

Проблема №0

Обнаружить инцидент в контейнере

- Подходы базирующиеся на правилах и сигнатурах работают очень плохо
- Происходящее в контейнерах меняется от компании к компании
- Уникальная логика у каждого микросервиса
- Сложно к чему-то привязаться

«Attackers Abusing legitimate Cloud Monitoring Tools to Conduct Cyber Attacks»
Intezer, 2020

«Bypassing Falco: How to Compromise a Cluster without Tripping the SOC»
Shay Berkovich (BlackBerry), 2022

Проблема №1

Традиционные инструменты не понимают контейнеры

Устоявший стек решений не подходит:

- Необходимо различать контейнерные процессы от хостовых
- Необходимо группировать активность по контейнерам
- Необходимо сопоставлять контейнеры с сущностями Kubernetes (Pod, Deployment, DaemonSet, ...)

Нужно использовать Cloud-native решения

Проблема №2

Высокие требования к оверхедам

Привет,
HighLoad!

Как в процессе сбора, обработки информации, так и при ее хранении

CPU

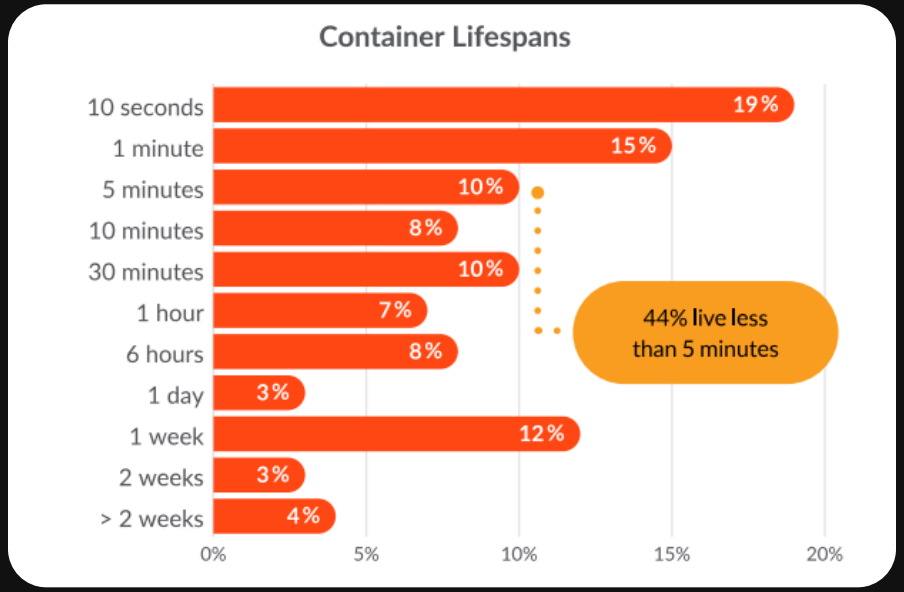
Memory

Никто не хочет замедлять свои микросервисы

Агенты генерируют большой объем данных

Никто не хочет отдавать много процессорных ядер под агенты на Nodes

Проблема №3
Динамическое окружение



“Sysdig 2022 Cloud-Native Security and Usage Report”

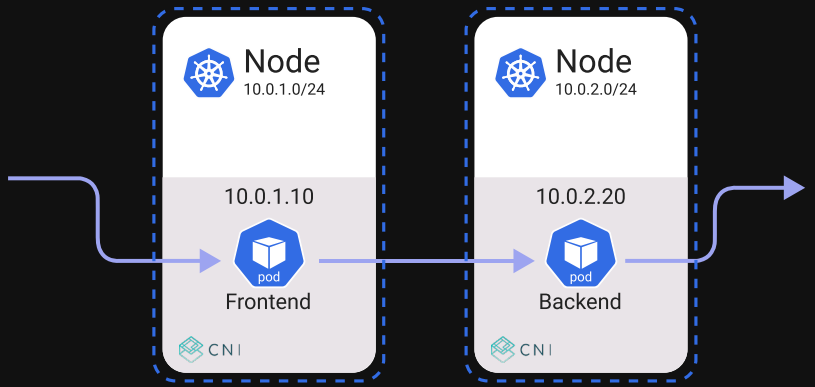
Малый срок жизни контейнеров

- Большое количество обновлений
- Self-healing
- Переезд контейнеров с Node на Node
- Появление новых Nodes и копий контейнеров при автоскейлинге

Следы злоумышленника в контейнере очищаются сами собой!

Проблема №4

Специфика сетевого взаимодействия в k8s

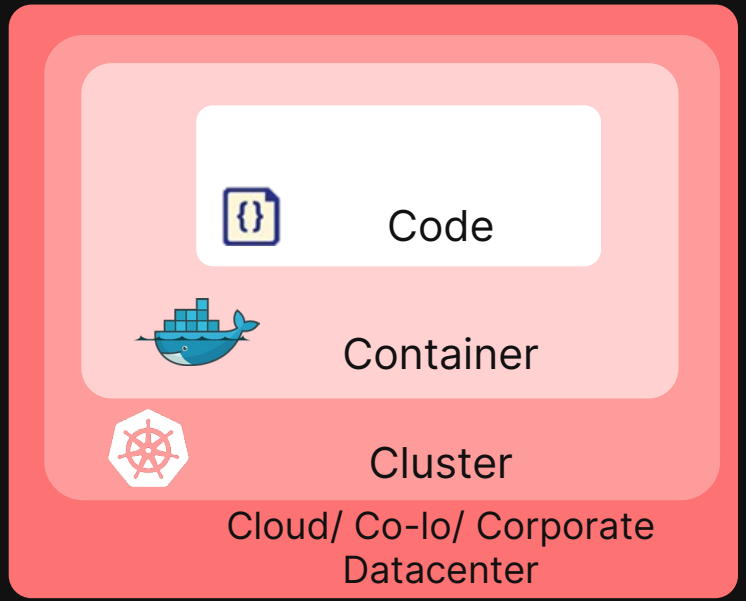


- All Pods have IPs
- All Pods can talk
- PodCIDR[s] per node

- Services for load-balancing
- DNS for service-discovery
- Network Policy for segmentation

IP адрес меняется/переходит от запуска к запуску контейнеров

Проблема №5
Множество уровней абстракций

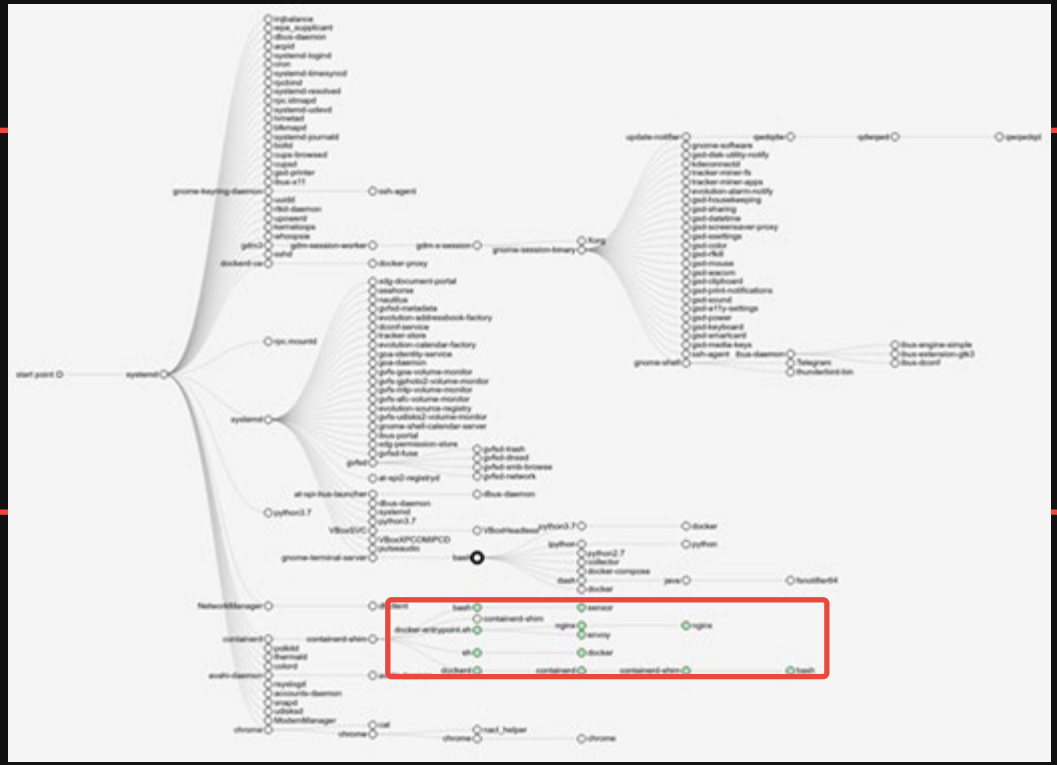


- Kubernetes это PaaS и каждый нижележащий уровень абстрагирован от другого
  - Четкое разделение на Control plane и workloads
- OS
  - OS Log
- Application
  - App Log
- Containers
  - Runtime
  - FS dump
- Kubernetes cluster
  - Kubernetes Audit Log
- Cloud/ Datacenter
  - Log

ПРЕИМУЩЕСТВА DFIR В K8S

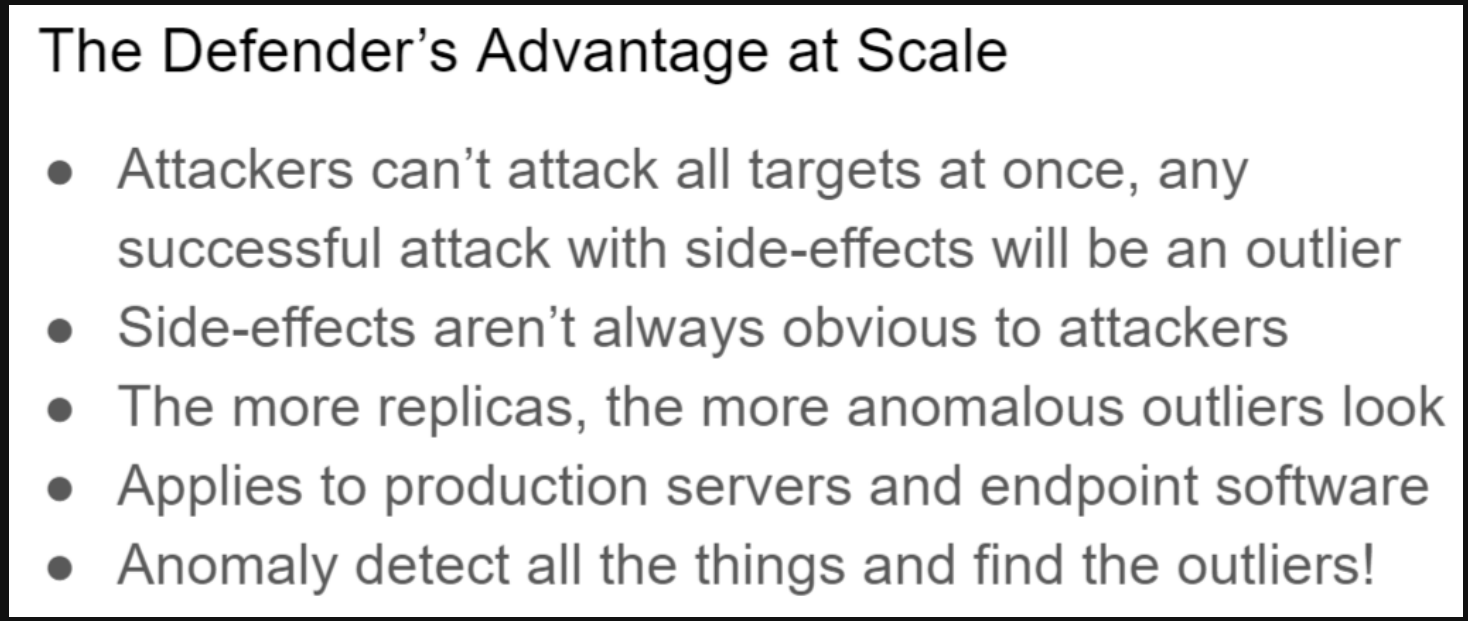
Преимущество #1
Простота микросервисов

Микросервис проще
ОС и монолитов



Преимущество #2

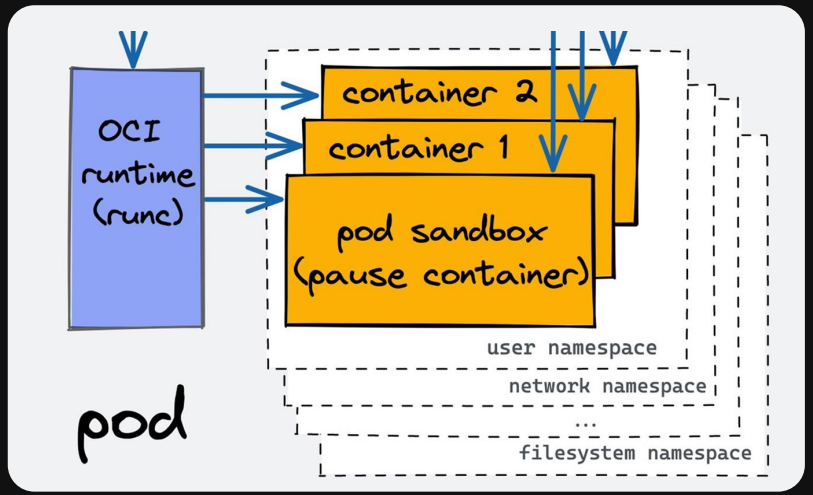
Работа микросервисов в несколько копий/реплик



"Thinking Outside the Box: Or, How I Learned to Stop Worrying and Love the Cloud", Dino A. Dai Zovi

Преимущество #3

Контейнеры это не rocket science 😉



Классический Container = Linux process + cgroup + namespaces (pid, user, uts, ipc, net, mnt, ...) + pivot_root + image

- Изоляция
- Образ

Можно работать как с обычными процессами и частью файловой системы хоста

```
root      598309  0.0  0.0 110128  6224 ?        Sl   Nov20   0:07  \_ containerd-shim -namespace moby -workdir /var/lib/containerd/io.containerd.runtime.v1
root      598334  1.4  5.5 7236340 1832196 pts/0 Ssl+ Nov20  39:39      \_ /docker-java-home/bin/java -Djava.util.logging.config.file=/opt/atlassian/conflue
root      599854  1.0  1.3 7007820 427956 pts/0  Sl+  Nov20  28:11      |   \_ /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java -classpath /opt/atlassian/conf
root      701694  0.0  0.0   4288   764 ?        Ss+  Nov20   0:00      \_ /bin/sh
```

Преимущество #4
Kubernetes это декларативная система

Все есть YAML

Все что касается Kubernetes Resources идет через Kubernetes API server

- Все попадает в Kubernetes Audit Log
- Исключение интерактивные команды и runtime
  - kubectl exec – установленное соединение по WebSocket
  - Нужен Runtime агент

Компании всё чаще приходят к концепции Everything-as-Code

- IaC
- Security-as-Code и т.д.

ИНСТРУМЕНТЫ

Классический Linux way

Работаем как с частью Linux системы

- Исключение Container Runtime на базе sandbox, microVM

Для оперативной памяти:

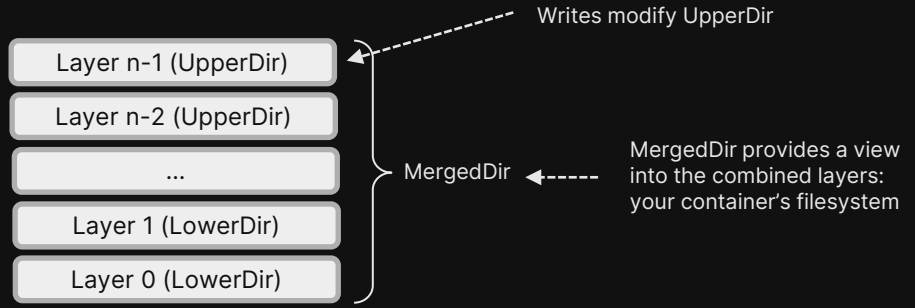
- /proc/<pid>/maps
- /proc/<pid>/mem
- ptrace
- gdb dump memory

Для ФС:

- /proc/<pid>/root
- /proc/<pid>/mountinfo
- LowerDir/UpperDir/MergedDir

Анализ памяти:

- AVML
- Volatility
- Rekall



От контейнера к хосту

Pod -> Container -> Process

- kubectl
- docker
- crictl
- ctr
- nerdctl

```
containerStatuses:
- name:  "server"
  image:  "gcr.io/google-samples/microservices-demo/paymentservice:v0.3.7"
  ready:  true
  state:
    running:
      startedAt:  "2022-08-16T09:14:21Z"
  imageID:  "gcr.io/google-samples/microservices-demo/paymentservice@sha256:59ed17..."
  started:  true
  lastState:  {}
  containerID:  "containerd://985a4b9154ecc190103517e3387b5a99b85c356433ef88a97e2f369f0eeb6856"
  restartCount:  0
```

```
➜ ~ docker run --rm -it debian:bullseye-slim bash
root@b474cf96a4eb:/#
```

```
➜ ~
➜ ~ docker ps --no-trunc
CONTAINER ID                                                       IMAGE                 COMMAND   CREATED         STATUS         PORTS     NAMES
b474cf96a4eb259edbffad4f352559f7c6f9dee12ce5d06d738d1583ee53d1de   debian:bullseye-slim   "bash"    9 minutes ago   Up 9 minutes             sharp_jennings
➜ ~ find /sys/fs/cgroup/ -name '*b474cf96a4eb259edbffad4f352559f7c6f9dee12ce5d06d738d1583ee53d1de*'
/sys/fs/cgroup/system.slice/docker-b474cf96a4eb259edbffad4f352559f7c6f9dee12ce5d06d738d1583ee53d1de.scope
➜ ~ cat /sys/fs/cgroup/system.slice/docker-b474cf96a4eb259edbffad4f352559f7c6f9dee12ce5d06d738d1583ee53d1de.scope/cgroup.procs
1149770
➜ ~ ps -p 1149770
    PID TTY          TIME CMD
1149770 pts/0    00:00:00 bash
➜ ~
```

Docker Checkpoint

Экспериментальная feature

CRIU под капотом

Можно запускать контейнер с checkpoint

Using checkpoint and restore

A new top level command `docker checkpoint` is introduced, with three subcommands:

- `docker checkpoint create` (creates a new checkpoint)
- `docker checkpoint ls` (lists existing checkpoints)
- `docker checkpoint rm` (deletes an existing checkpoint)

Additionally, a `--checkpoint` flag is added to the `docker container start` command.

Use cases for checkpoint and restore

This feature is currently focused on single-host use cases for checkpoint and restore. Here are a few:

- Restarting the host machine without stopping/starting containers
- Speeding up the start time of slow start applications
- "Rewinding" processes to an earlier point in time
- "Forensic debugging" of running processes

Kube-forensics

Kubernetes operator

- Custom Resource
- Декларативный подход

Под капотом набор команд:

- docker inspect
- docker diff
- docker export

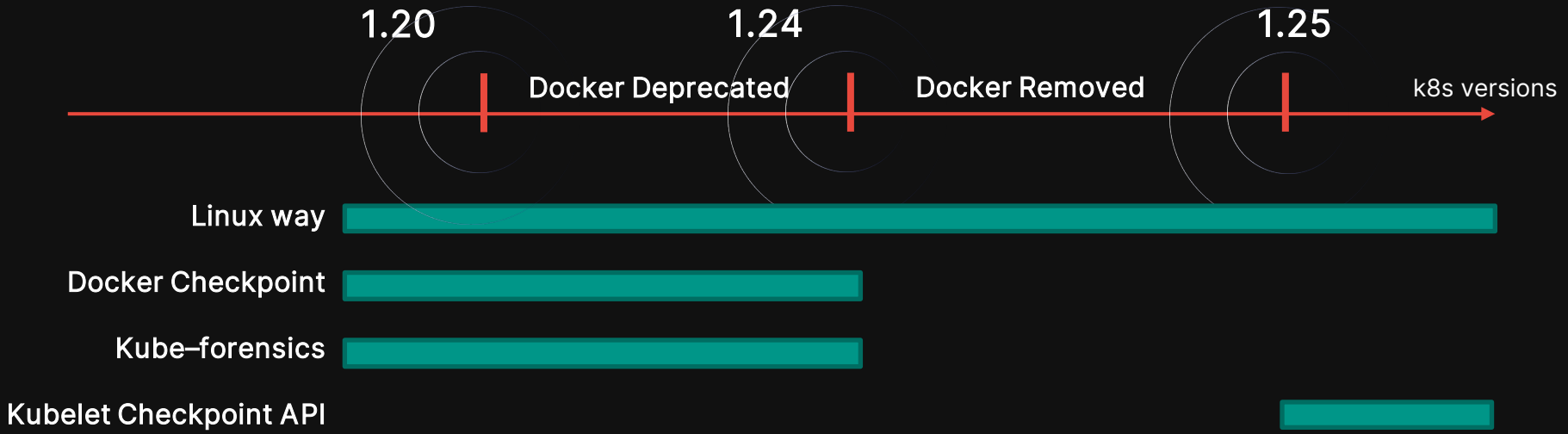
```
apiVersion: forensics.keikoproj.io/v1alpha1
kind: PodCheckpoint
metadata:
  name: podcheckpoint-sample
  namespace: forensics-system
spec:
  destination: s3://my-bucket-123456789
  subpath: forensics
  pod: bad-pod-1234567890-dead1
  namespace: default
```

Kubelet Checkpoint API

- Доступно с Kubernetes версии 1.25
- CRIU под капотом
- `POST /checkpoint/{namespace}/{pod}/{container}`

checkpointctl

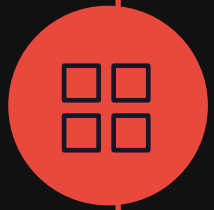
Итоги по инструментам



Заключение

Выводы

Необходимо фиксировать инциденты в контейнерах максимально быстро пока они еще существуют



Необходимо использовать иммутабельность, распределенность и эфемерность контейнерных инфраструктур в свою пользу

Лучше (проще) не доводить до инцидентов в контейнерах, но это другая история ;)

LUNTRY
luntry.ru

СПАСИБО ЗА ВНИМАНИЕ!

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

TOLTARYS
KAZHACKSTAN