Kubernetes: Observability важная часть Security

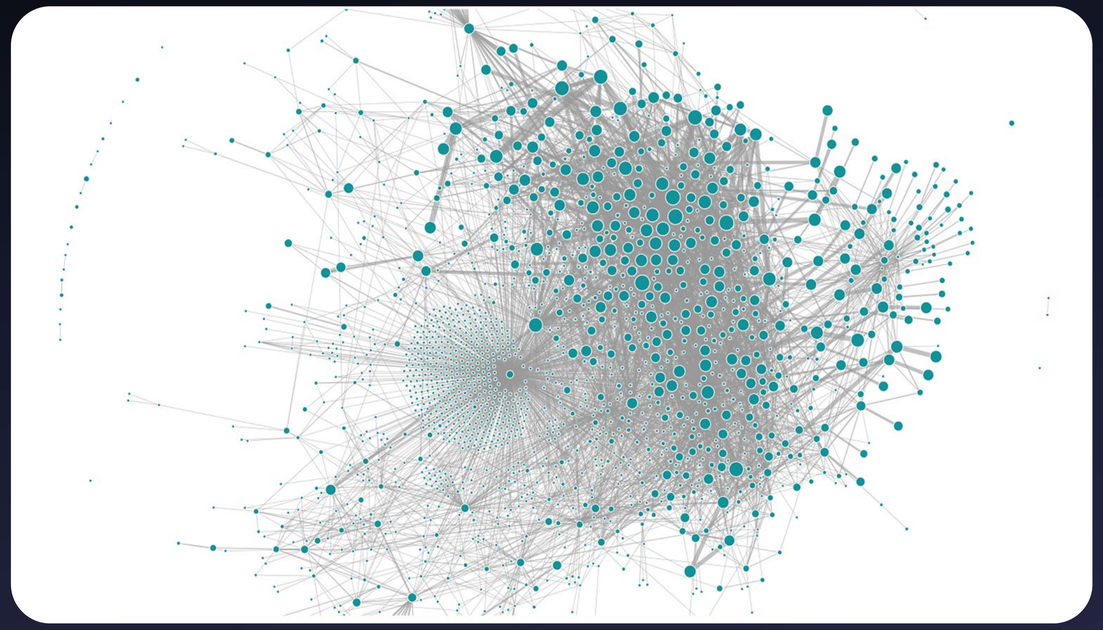
Дмитрий Евдокимов

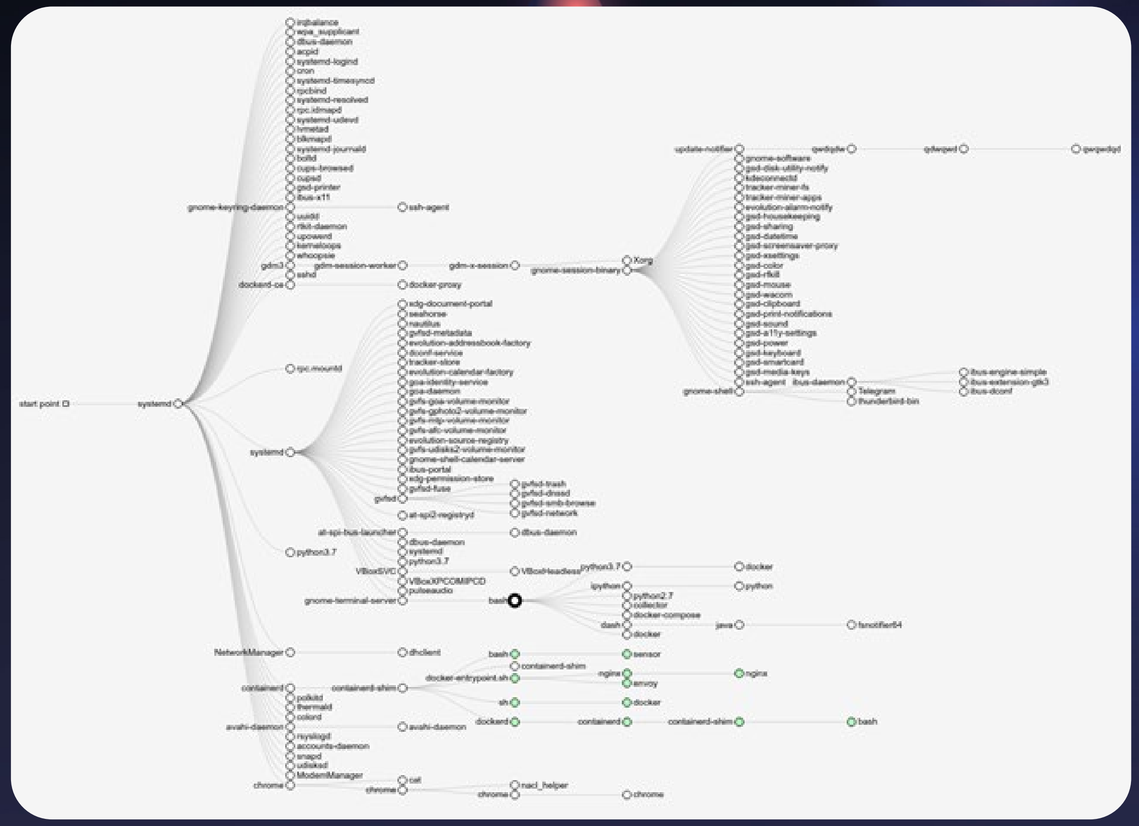
CTO, Co-Founder

Whoami

- Технический директор, исследователь ИБ
- Организатор ZeroNights, DEFCON Russia (#7812)
- В прошлом — автор статей журнала «ХАКЕР»
- Мейнтейнер проекта «Python Arsenal for Reverse Engineering»
- Автор Telegram-канала **k8s (in)security**
- Автор тренинга «Cloud-Native Security in Kubernetes»
- Спикер: BlackHat, HITB, ZeroNights, HackInParis, Confidence, SAS, PHDays и др.

Неизвестный мир





"Complexity is the worst enemy of security, and our systems are getting more complex all the time"

Bruce Schneier

"The only thing that ever yielded real security gains was controlling complexity"

Thomas Dullien / Halvar Flake

Cat & Mouse game или атакующий на шаг впереди

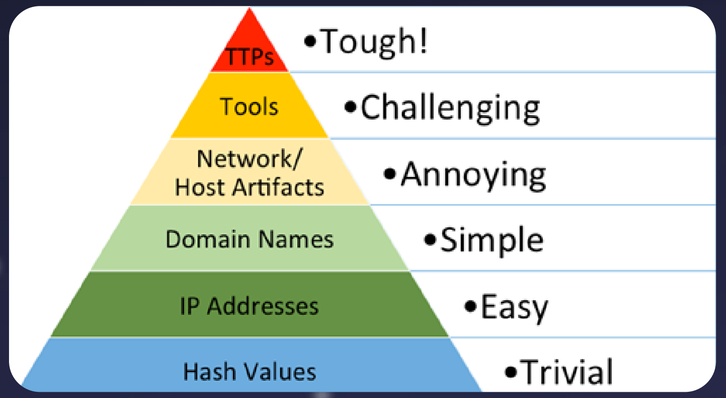
MITRE ATT&CK Matrix

- Adversarial Tactics, Techniques & Common Knowledge

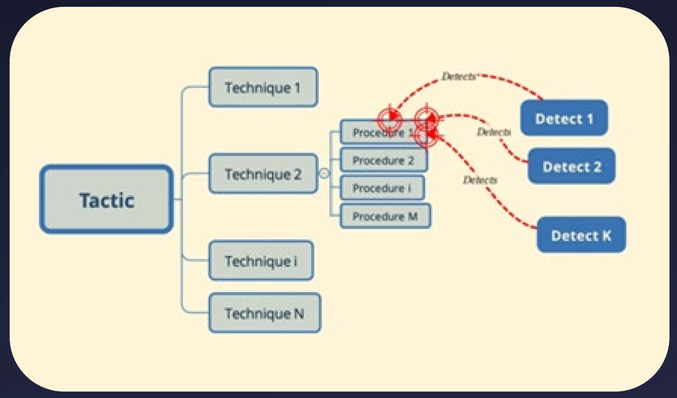
  attack.mitre.org/

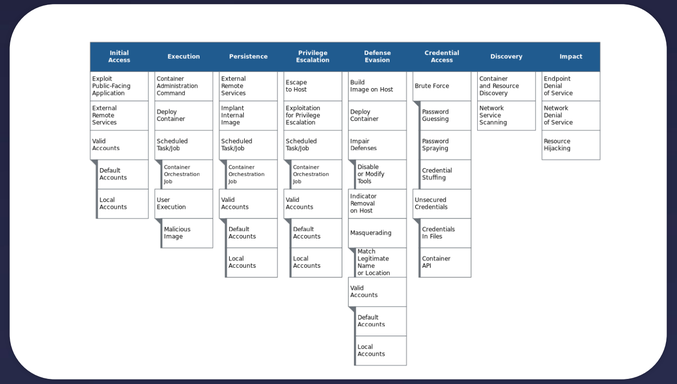
- Tactics, Techniques and Procedures

Только известные факты!



[The Pyramid of Pain]



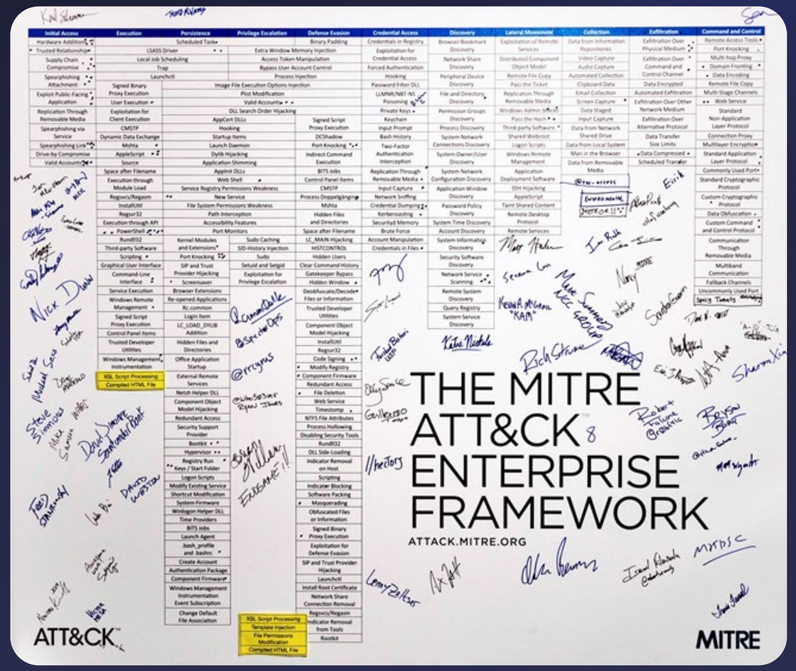


Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Impact
Exploit Public-Facing Application	Container Administration Command	External Remote Services	Escape to Host	Build Image on Host	Brute Force	Container and Resource Discovery		Endpoint Denial of Service	
External Remote Services	Deploy Container	Implant Internal Image	Exploitation for Privilege Escalation	Deploy Container	Password Guessing	Network Service Scanning		Network Denial of Service	
Valid Accounts	Scheduled Task/Job	Scheduled Task/Job	Scheduled Task/Job	Impair Defenses	Password Spraying			Resource Hijacking	
Default Accounts	Container Orchestration Job	Container Orchestration Job	Container Orchestration Job	Disable or Modify Tools	Credential Stuffing				
Local Accounts	User Execution	Valid Accounts	Valid Accounts	Indicator Removal on Host	Unsecured Credentials				
	Malicious Image	Default Accounts	Default Accounts	Masquerading	Credentials In Files				
		Local Accounts	Local Accounts	Match Legitimate Name or Location	Container API				
				Valid Accounts					
				Default Accounts					
				Local Accounts					

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



Преступная группа TeamTNT

Attackers Abusing Legitimate Cloud Monitoring Tools to Conduct Cyber Attacks

Written by **Nicole Fishbein** - 8 September 2020

[Источник]

APT

"Advanced Persistence Threats: The Future of Kubernetes Attacks",
Ian Coldwater & Brad Geesaman

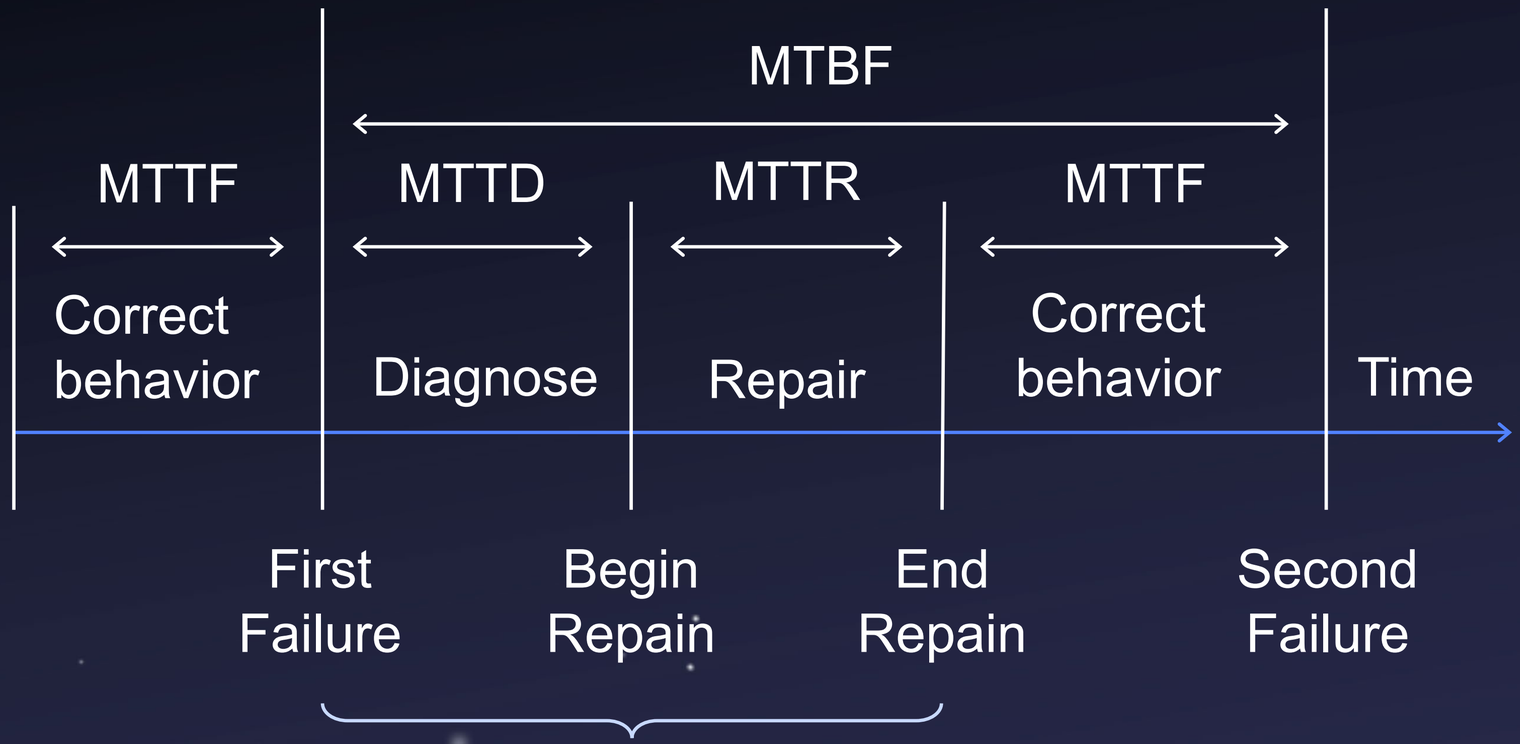
Микросервисы меняют подход к ИБ

The Defender’s Advantage at Scale

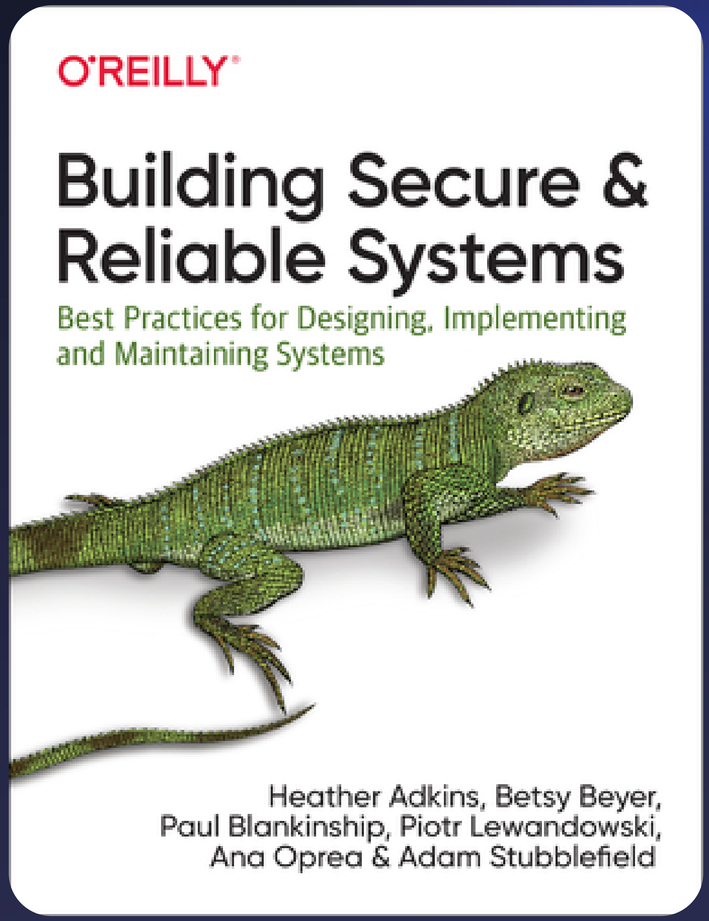
- Attackers can’t attack all targets at once, any successful attack with side-effects will be an outlier
- Side-effects aren’t always obvious to attackers
- The more replicas, the more anomalous outliers look
- Applies to production servers and endpoint software
- Anomaly detect all the things and find the outliers!

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

Безопасные и надежные системы



MTTD + MTTR = это общая продолжительность инцидента ИТ или ИБ



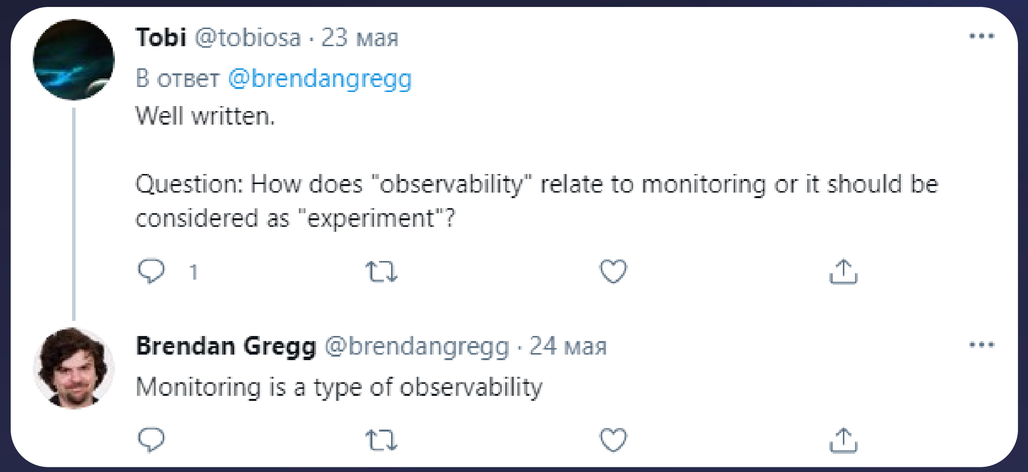
Что такое Observability?

Wikipedia:

""Observability" is being able to fully understand our systems. In control theory, observability is a measure of how well internal states of a system can be inferred from knowledge of its external outputs. The observability and controllability of a system are mathematical duals".

Помогает:

- в прошлом: Root Cause Analysis (RCA)
- в настоящем: Оперативно и правильно принимать решения
- в будущем: Планировать улучшения, изменения и т. д.



Источник

CNCF SIG Observability

CNCF TAG Observability

Technical Advisory Group for Observability under the umbrella of CNCF.

Mission statement

The TAG Observability focuses on topics pertaining to the observation of cloud native workloads. Additionally, it produces supporting material and best practices for end-users and provides guidance and coordination for CNCF projects working within the TAG's scope.

Создают документ "Observability Whitepaper" (WIP)

Из документа “Observability Whitepaper”

Sysadmins, developers, and software operators must know the state of an application in production as well as the underlying infrastructure health where that application is running.

… that wish to deliver software that is both observable, and which integrates with their customers’ existing observability systems while reaching a demonstrable level of reliability, security, and transparency.

… Observability is a multidisciplinary topic.

Observability can be used on literally all phases of the development lifecycle of a system.

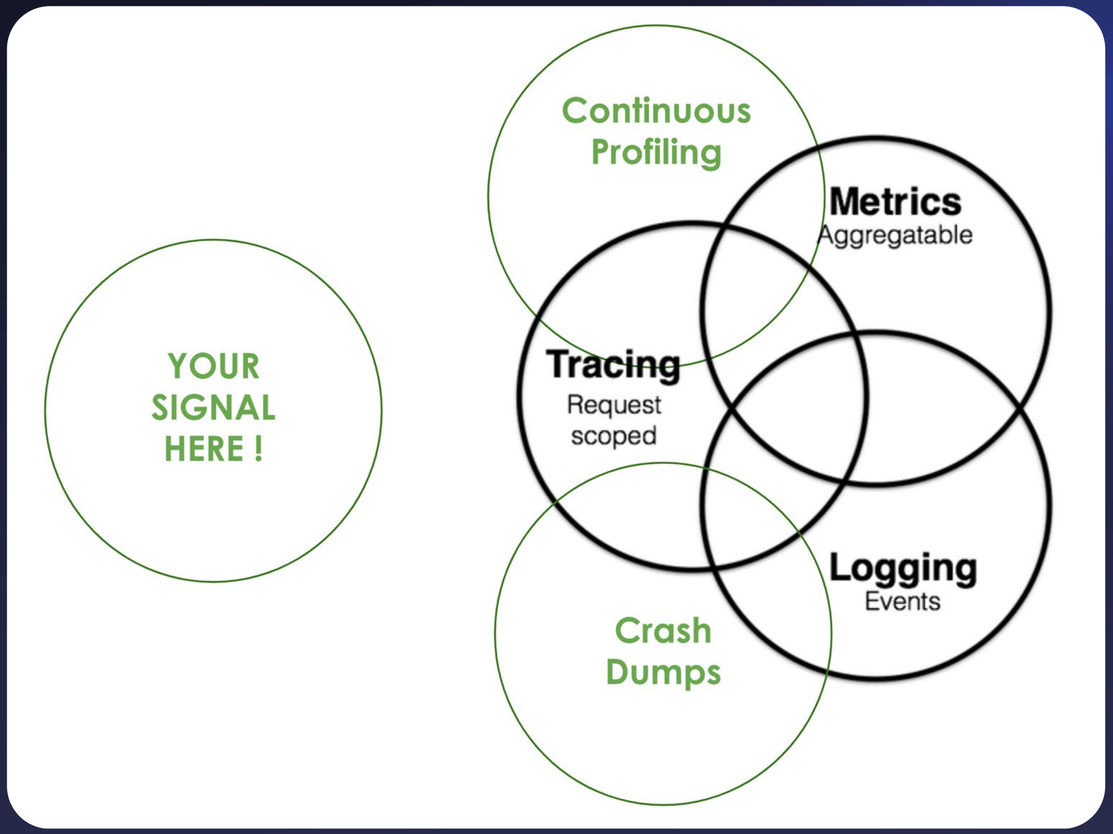
Once you have your goal in mind, that’s when you’ll start thinking about the outputs,
or what we like to call them, the signals.

Observability сигналы

Signals are the outputs that a system produces that a human, or machine, can infer knowledge from. Those signals will vary from system to system, and also depending on the objective you want to accomplish.

Kubernetes specific:

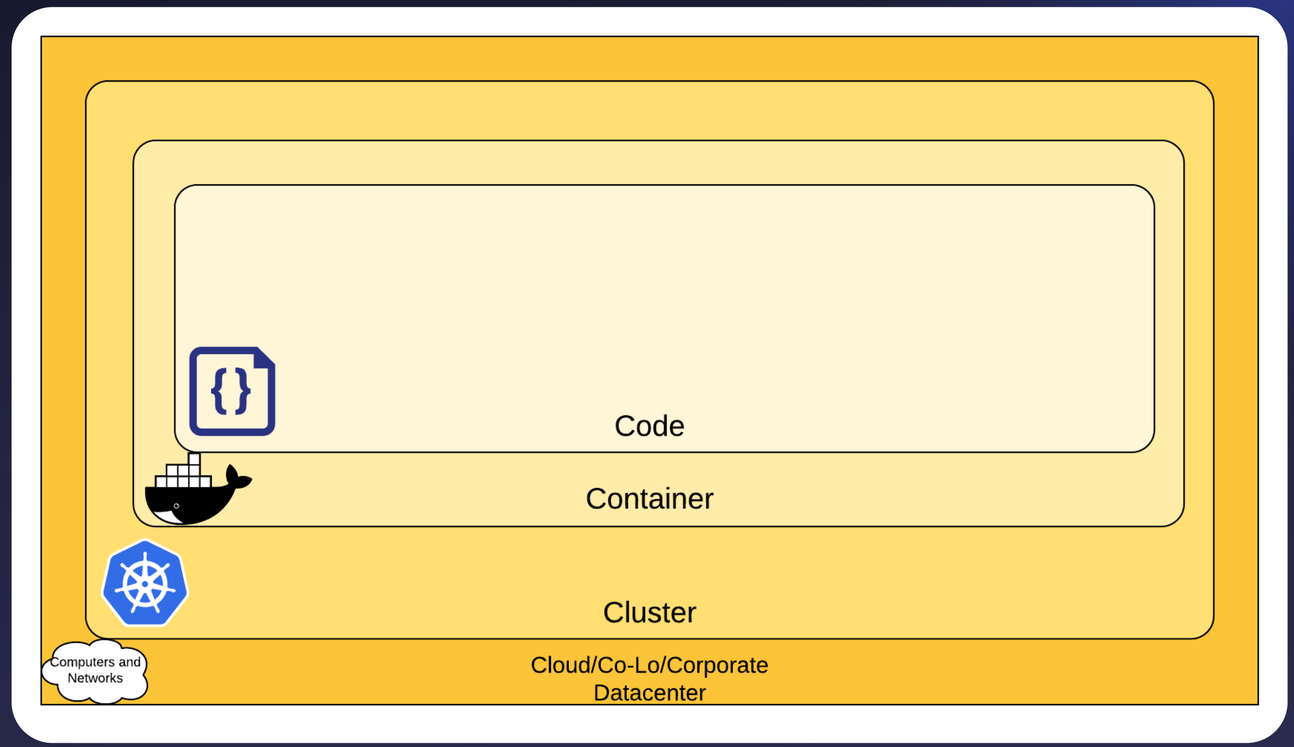
- Изменение ресурсов
- Events ресурс
- События внутри workloads
- ...



Сигналы для безопасности

Будем исходить из следующего:
"The 4C's of Cloud Native security":

- Cloud/Co-Lo/Corporate Datacenter
  - Управление активами
- Cluster
  - Компоненты Kubernetes
  - Kubernetes ресурсы
- Container
  - Процессы
  - Взаимодействие с файлами
  - Взаимодействие с сетью
- Code
  - Images, SBOM



С точки зрения моделей нарушителя

Скомпрометированная Supply Chain

- Уязвимый вредоносный:
1. Библиотека, Open Source код
2. Образ контейнера
3. HELM chart

Внутренний злоумышленник

- «Отладка» в Prod окружении
- Bad-practices

Внешний злоумышленник

- Server-side уязвимости в приложениях
- Misconfigurations

Новые/забытые активы

Изменения Kubernetes ресурсов и компонент

Поведенческие аномалии

Изменения SBOM, репозиториев

Почему вы можете не знать, что злоумышленник проявляет вредоносную активность в вашем Kubernetes?

- Злоумышленник проэксплуатировал неизвестную 0day уязвимость
- Злоумышленник успел проэксплуатировать незакрытую 1day уязвимость и залил неизвестный вредоносный код
- Злоумышленник встроил backdoor в библиотеку, что используется в образе
- Непонятно, что происходит внутри микросервисов

Observability для Continuous security

Continuous inventory

- Всегда знаем, что используется и когда меняется

Continuous security profiling

- Всегда понимаем поведение и когда оно меняется

Нельзя сделать надежным и безопасным то, что скрыто от ваших глаз!

"Scientia potentia est", — Latin aphorism

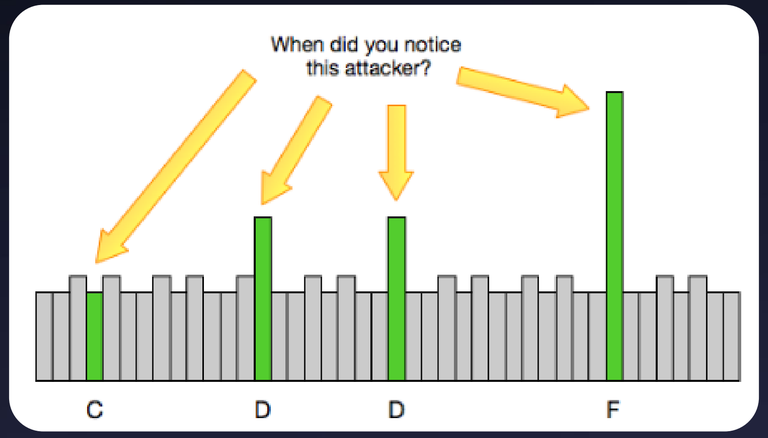
Ответьте себе на следующие вопросы:

1. Знаете/видите/понимаете ли вы, что происходит внутри ваших контейнеров?
2. Какие образы и где используются в инфраструктуре?
3. Какой стек технологий и ПО используется в инфраструктуре?
4. Видите ли вы все изменения инфраструктуры?
5. На что способны и что делают пользователи и сервис-аккаунты?

При этом вы должны быть способны ответить на любой из вопросов
в любой момент времени.

Материал для старта: "Risk8s Business: Risk Analysis of Kubernetes Clusters"

Observability & Metrics



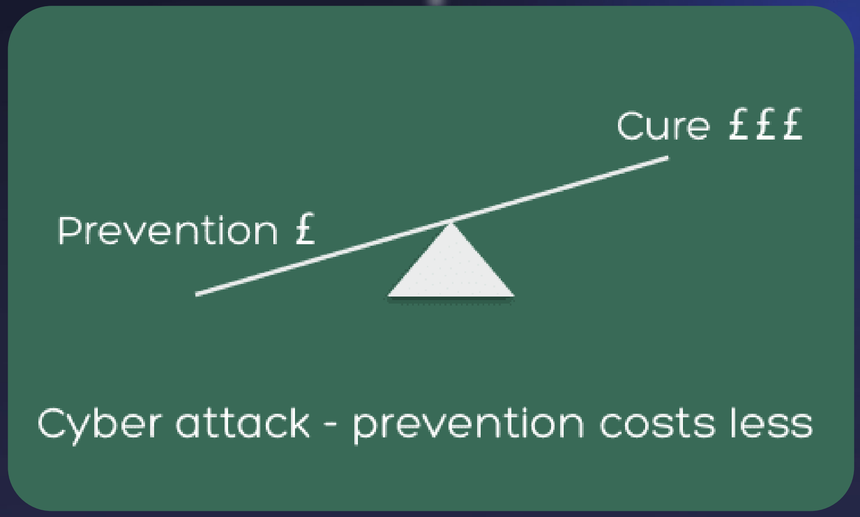
Instituting observability and metrics into cloud native architectures delivers security insights so appropriate stakeholders can resolve and mitigate anomalies appearing in reporting; tools in this area can help collect and visualize this information. Through the use of behavioral and heuristic analysis teams can detect and escalate outliers, suspicious events, and unexplained calls to appropriate stakeholders. Artificial intelligence (AI), machine learning (ML), or statistical modeling are all mechanisms that are encouraged to assist in behavioral and heuristic analysis development.

CNCF Cloud Native Security Whitepaper

Безопасность не через понимание угрозы, а через понимание себя

По нормальному поведению можно создавать:

- NetworkPolicy
  - Hubble/Weave Scope/Service Mesh/Distributed tracing/...
  - Inspektor Gadget/Network Policy Editor/…
- Политики для policy engines
- OPA/Gatekeeper/Kyverno/MagTape/k-rail/Kubewarden/…
- AppArmor или SELinux профили
  - AppArmor complain mode/SELinux permissive mode
  - kube-apparmor-manager/bane/…
- seccomp профиль
  - SCMP_ACT_LOG action - complain mode
  - Kubernetes Security Profiles Operator/oci-seccomp-bpf-hook/…



"Building "security" into a system involves limiting what can happen in the system". — Thomas Dullien/Halvar Flake

Ломать — не строить!

Второй важный момент: не сломать!

- Заблокировать, запретить и ограничить — это не проблема
- Проблема — сделать так, чтобы все продолжало работать как надо:
  - нужно знать, как работает
  - нужно знать всем, кто с этим работает

Не пренебрегайте:

- созданием Pre-prod окружений
- тестами
  - Dry run тестированием

"Forewarned is forearmed", — proverb

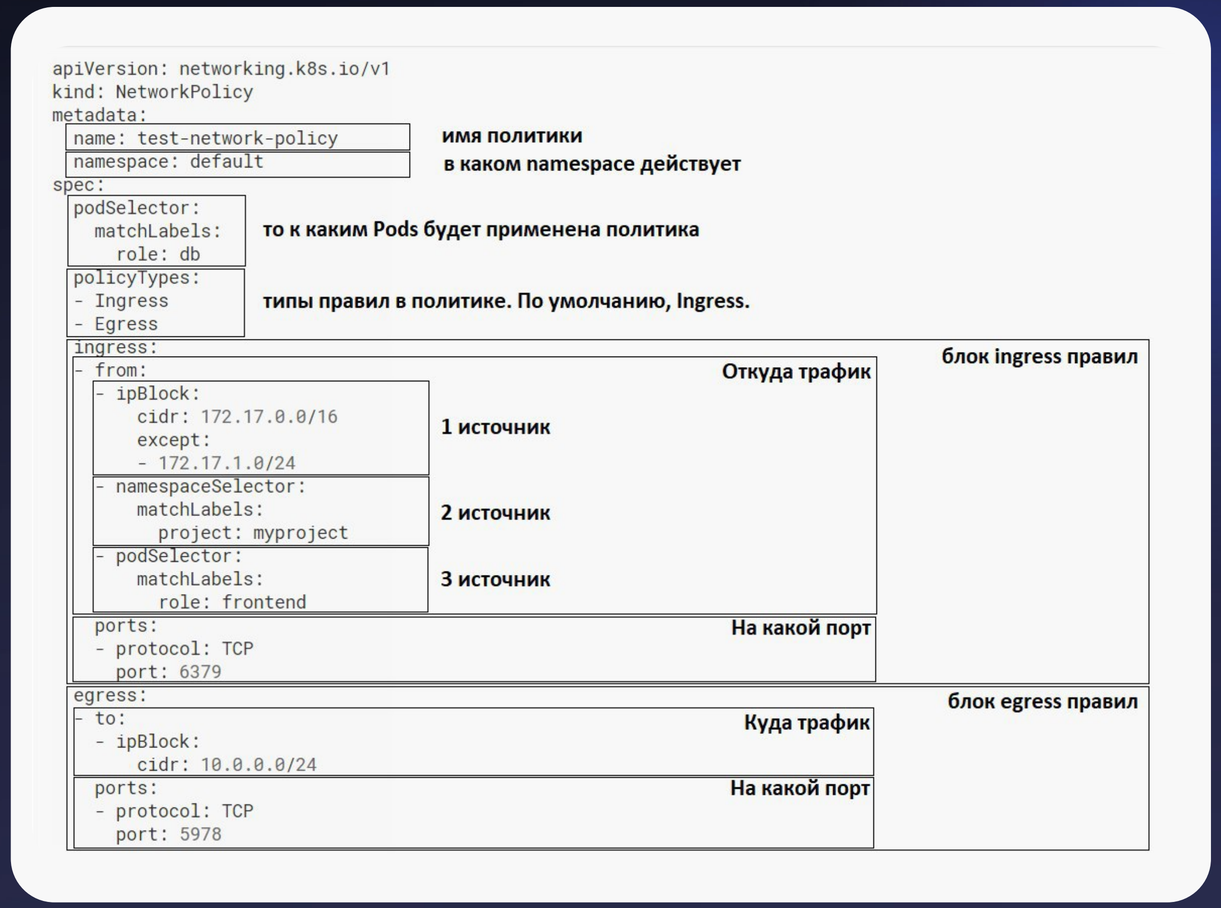
Пример: Создание NetworkPolicy

Условия:

- GitOps
- Приложение:
  - 2 Deployments, 1 Ingress, 2 Service

Что нужно знать и учитывать:

- Знание labels у взаимодействующих namespaces
- Используется ли Service Mesh?
- Какая реализация Ingress используется и где она находится?
- Используется ли отдача Logs, metrics, traces в коде приложения сторонним сервисам?
- …



Эшелонированная оборона

Threat modeling

Code	Images	k8s resources	Authentication Webhook	Authorization Webhook	Admission controllers	Audit Log Webhook	Container/VM	Observability
SAST	Immutable	Labels, annotations	RBAC	RBAC	LimitRanger		Isolation	Asset management
DAST	Distroless images	IaC			ResourceQuota		Rootless containers, Capabilities	Security monitoring
IAST		Security as Code			PodSecurityPolicy		seccomp, AppArmor, Selinux	Application monitoring
RASP		Configuration check			ImagePolicyWebhook		Limiting the blast radius	Anomaly detection
SCA					NetworkPolicy		Segregation of duties (Secrets, ServiceAccounts token)	
					PodSecurityPolicy			
					MutatingAdmissionWebhook			
					Init containers + sidecars containers injection			
					ValidatingAdmissionWebhook			
+ Multi-tenancy*					Custom Resource + operator (policy engines)			

Заключение

Observability/знание/понимание происходящего позволяет:

- Обнаруживать аномалии
- Строить безопасность от себя
- Не ломать то, что работает

Безопасность без слепых зон:

- Защита от незнания

Reliability и security идут рука об руку.

"The behavior of a piece of software does not determine whether it is malicious or not. The true defining line between malicious and non-malicious software is whether software does what the user expects it to do. The question of malicious / non-malicious software is a question of alignment between user expectations and software behavior",

— Thomas Dullien/Halvar Flake

Q&A

Спасибо за внимание!

Дмитрий Евдокимов

CTO, Co-Founder

Email: de@luntry.ru

Twitter: @evdokimovds

Telegram: @Qu3b3c