Классификация и систематизация средств безопасности для Kubernetes

Дмитрий Евдокимов
Founder&CTO, Luntry

Cyber Camp

Обо мне

- Основатель и технический директор Luntry
- Опыт в ИБ более 10 лет
- Бывший автор статей и редактор рубрик в журнале “ХАКЕР”

- Автор Telegram-канала “k8s (in)security”
- Автор курса “Cloud Native безопасность в Kubernetes”

- CFP ZeroNights, DevOpsConf
- Докладчик: BlackHat, HITB, ZeroNights, HackInParis, Confidence, SAS, OFFZONE, PHDays, Kazhackstan, DevOpsConf, KuberConf, VK Kubernetes Conference, HighLoad++ и др.

Disclaimer

За 40 минут невозможно рассмотреть абсолютно всё, и мы остановимся на самом важном на мой взгляд. За рамками доклада останутся практически все стоящие рядом с Kubernetes системы (CSM, CI\CD, Image Registry и т.д.).

Будем рассматривать OnPrem инсталляции Kubernetes. Хотя большая часть и будет применима к Managed Kubernetes у облачных провайдеров, но с определёнными уточнениями.

В докладе целенаправленно не упоминаются конкретные решения и реализации, чтобы быть максимально объективным.

Данный доклад подразумевает, что вы знаете, что такое контейнеры, Kubernetes и как они верхнеуровнево устроены и работают ;)

Agenda

- Введение
- Multitenancy
- Аутентификация пользователей
- Анализатор прав доступа
- Контроль Kubernetes ресурсов
- Логирование
- Контроль безопасности образов
- Управление секретами
- Безопасность Runtime
- Безопасность хоста/ноды
- Сетевая безопасность
- Контроль соответствия
- Заключение

К чему все привыкли?

- Firewall
- IPS/IDS
- WAF
- SIEM
- DLP
- Key Management
- IAM
- PAM
- SoD
- Patch Management
- Vulnerability Assessment
- Antimalware
- FIM
- DCAP
- EDR
- SOAR
- Deception Platform
- …

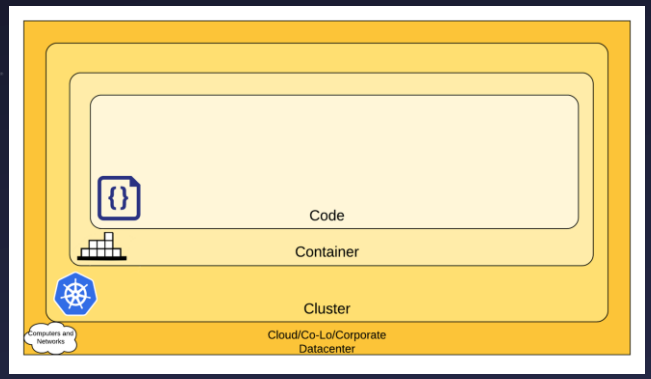
А как дела с этим в Kubernetes?

Как рассматривать?!

- Develop
- Distribute
- Deploy
- Runtime

"CNCF Cloud Native Security Whitepaper"

- Cloud/Co-Lo/Corporate Datacenter
- Cluster
- Container
- Code



"The 4C's of Cloud Native security"

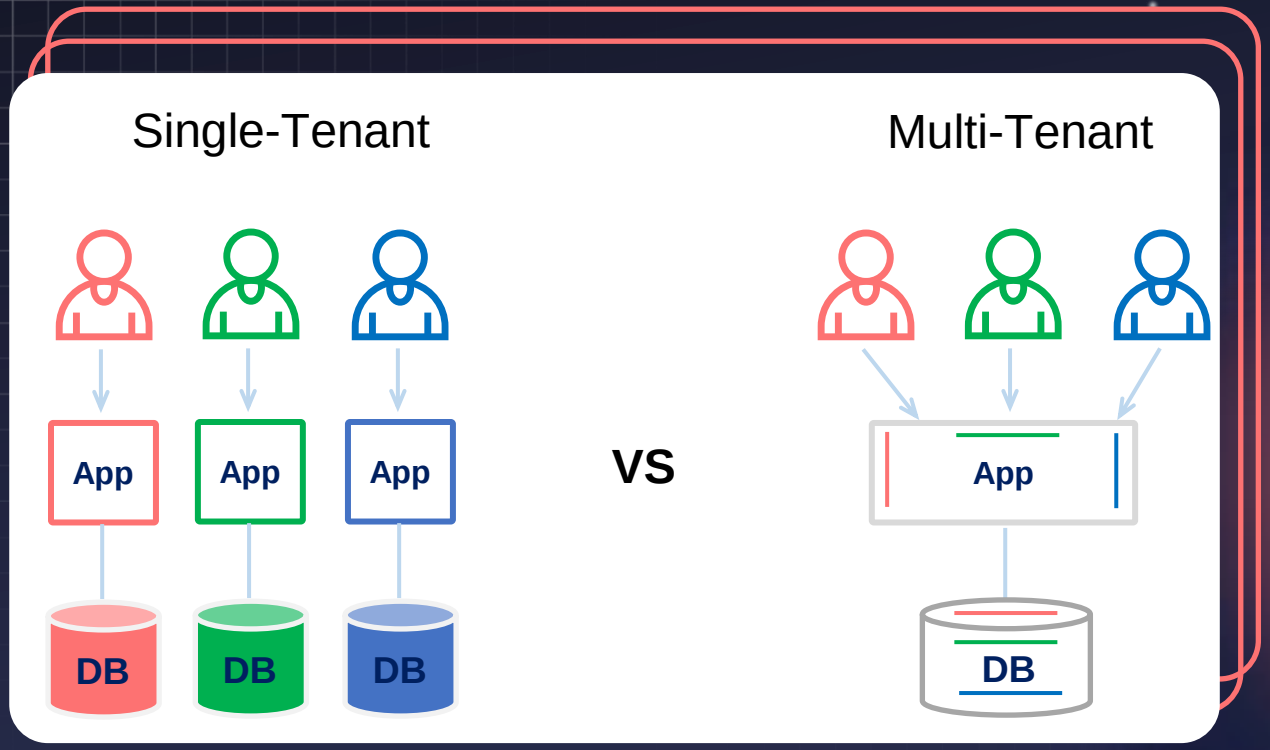
- Identify
- Protect
- Detect
- Respond
- Recover
- Deception*

"NIST CYBERSECURITY FRAMEWORK"

Multitenancy

Kubernetes из коробки не обладает мультитенантностью!

Это возможность изолированно обслуживать разные проекты, команды, клиентов, группы приложений



Multitenancy: реализации

СУЩЕСТВУЮЩИЕ СПОСОБЫ:

- Cluster-as-a-Services
- ControlPlane-as-a-Services
- Namespace-as-a-Services
- Node-based isolation

Работа с пользователями

- Identity and Access Management (IAM)
- Privileged Access Management (PAM)

All Kubernetes clusters have two categories of users: service accounts managed by Kubernetes, and normal users.

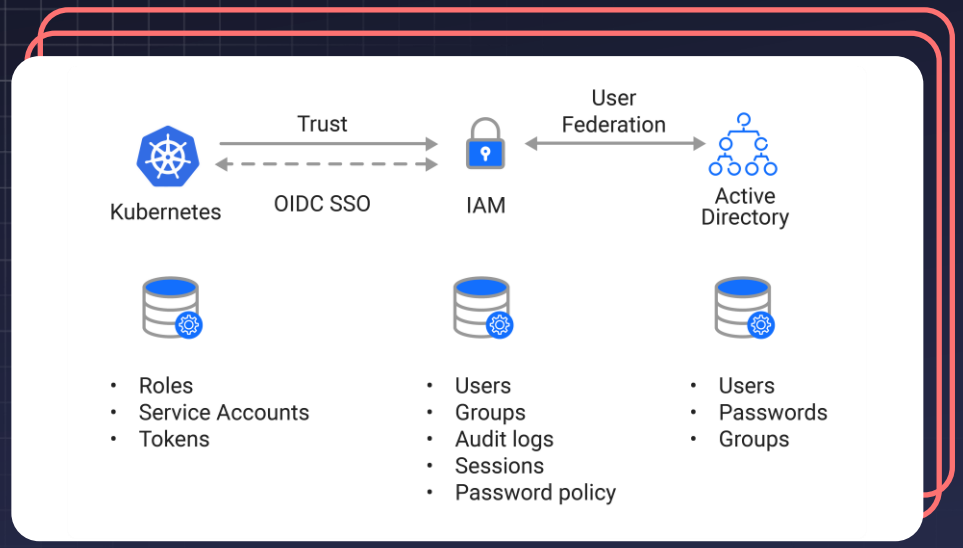
It is assumed that a cluster-independent service manages normal users in the following ways:

- an administrator distributing private keys
- a user store like Keystone or Google Accounts
- a file with a list of usernames and passwords

In this regard, *Kubernetes does not have objects which represent normal user accounts.* Normal users cannot be added to a cluster through an API call.

Аутентификация пользователей

- OpenID/SAML/LDAP/LDAPS Connect Identity Provider
- Public Identity Provider
- Own Identity Provider



Authentication: Mechanisms		
Mechanism	**Secret Source**	**Usage**
X509 Client Certs	CSR generated externally and signed with the cluster CA key	Enterprise CA / PKI
	Via Kubernetes API CertificateSigningRequest	Kubernetes cluster admin
Bearer token	Bootstrap token	Internal use
	Node authentication token	Internal use
	Static token file	Insecure
	ServiceAccount token	Pods, containers, applications, *users*
	OIDC token	Users
HTTP Basic auth	Static password file	Insecure
Auth ptoxy	N/A (trust proxy)	Integration
Impersonate	N/A (trust account)	Integration and administration

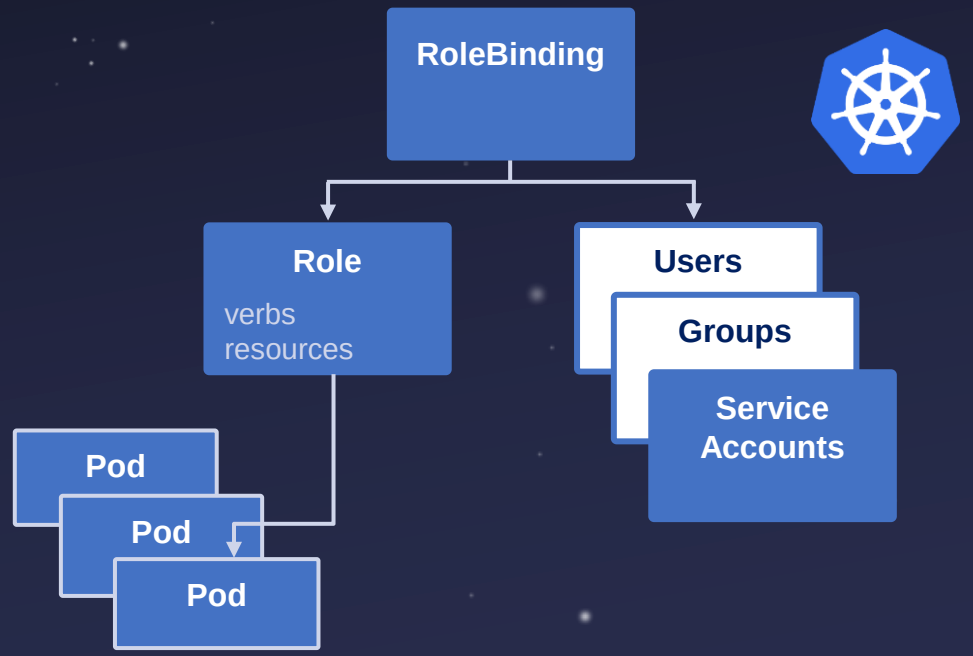
Анализатор прав доступа

Анализ прав доступа (RBAC) к Kubernetes ресурсам на соответствие принципу наименьших привилегий и на опасные права.

Реализация:

- Периодическая проверка в Kubernetes
- На этапе деплоя

Authorization: Mechanisms		
Mechanism	**Decision Source**	**Usage**
Node	API Server built-in	Internal use (kubelets)
ABAC	Static file	Insecure, deprecated
RBAC	API Objects	User and administrators
WebHook	External services	Integration
AlwaysDeny AlwaysAllow	API Server built-in	Testing

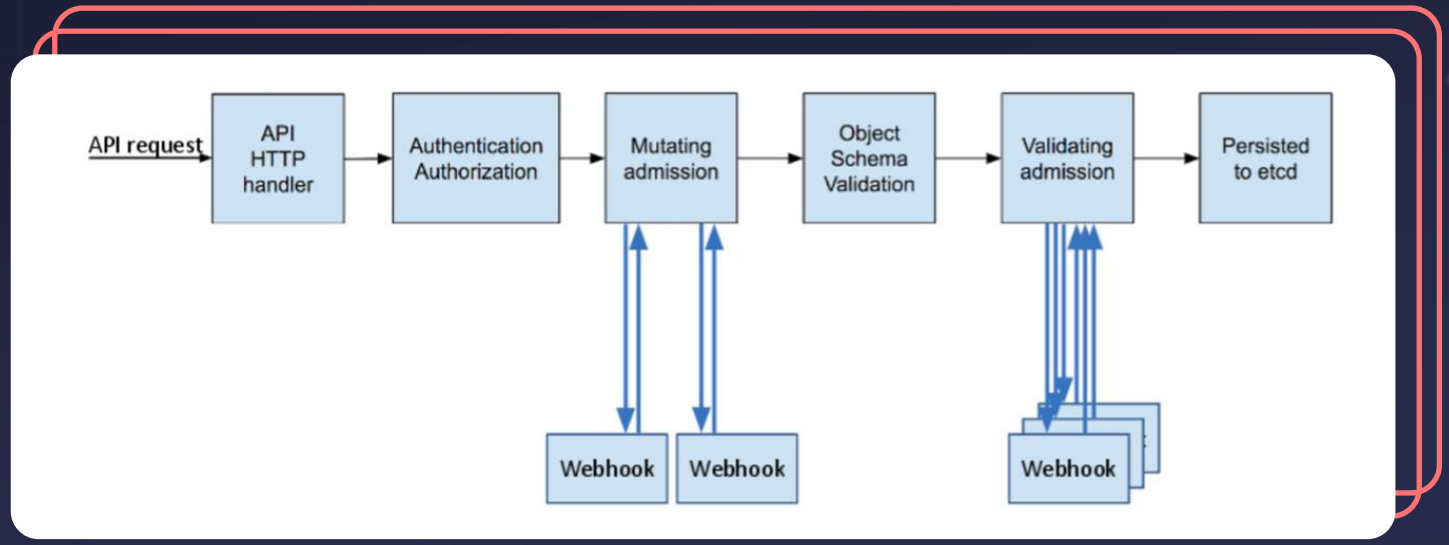


Контроль Kubernetes ресурсов

ВСЕ, ЧТО ЕСТЬ В KUBERNETES, - ЭТО YAML!

Задачи:

- Мутация – добавление, обогащение Kubernetes ресурса или исключение нежелательного
- Валидация – контроль соответствия в режимах аудита и предотвращения
- Генерация – автоматическое создание дополнительных Kubernetes ресурсов



Контроль Kubernetes ресурсов: реализация

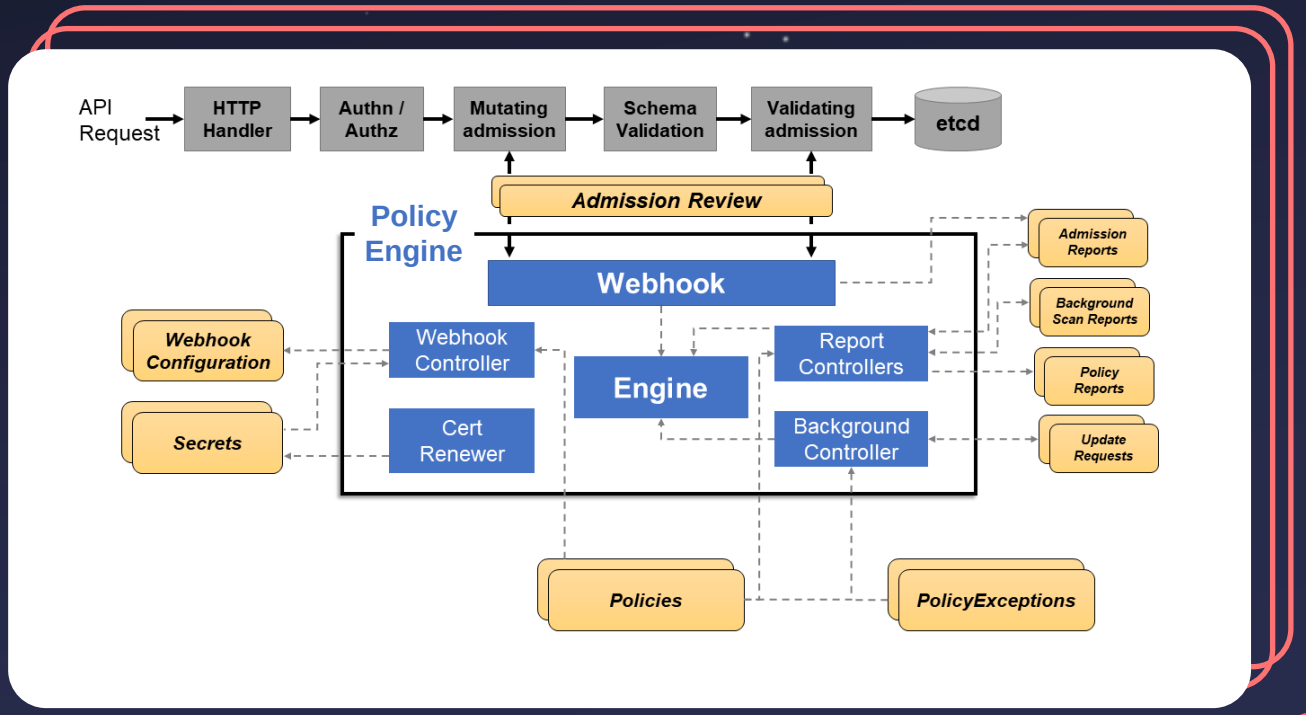
PODSECURITYPOLICY (PSP)
- В deprecated статусе с 1.21 и удален в 1.25

POD SECURITY ADMISSION (PSA)
- В alpha стадии в 1.22

VALIDATINGADMISSIONPOLICY
- В alpha стадии в 1.26

СОБСТВЕННАЯ РЕАЛИЗАЦИЯ
- Свой Admission Controller

POLICY ENGINES
- Must have!!!



Логирование

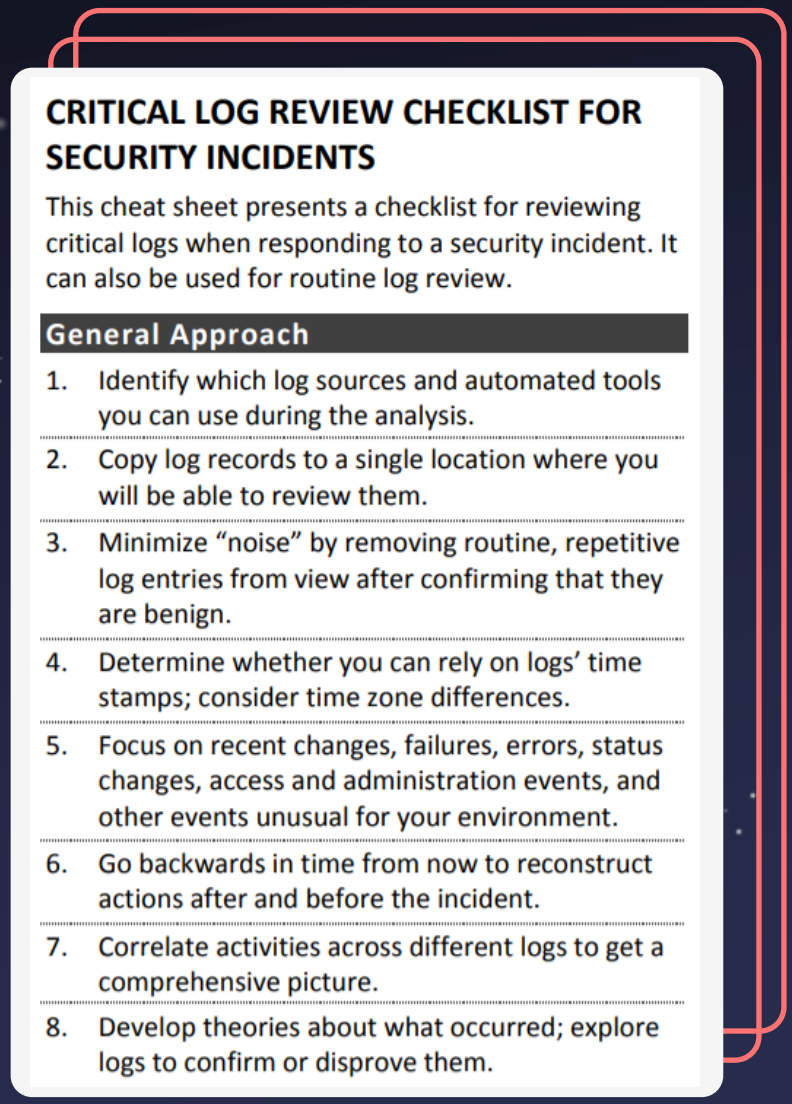
ОТВЕЧАЕМ НА ВОПРОСЫ:

- Что произошло?
- Кто сделал?
- Когда?
- Где?

ОТДЕЛЬНОЕ ХРАНИЛИЩЕ

БАЗОВЫЕ ПРАВИЛА КОРРЕЛЯЦИИ

ПЛАН РЕАГИРОВАНИ



CRITICAL LOG REVIEW CHECKLIST FOR SECURITY INCIDENTS

This cheat sheet presents a checklist for reviewing critical logs when responding to a security incident. It can also be used for routine log review.

General Approach

1. Identify which log sources and automated tools you can use during the analysis.
2. Copy log records to a single location where you will be able to review them.
3. Minimize “noise” by removing routine, repetitive log entries from view after confirming that they are benign.
4. Determine whether you can rely on logs’ time stamps; consider time zone differences.
5. Focus on recent changes, failures, errors, status changes, access and administration events, and other events unusual for your environment.
6. Go backwards in time from now to reconstruct actions after and before the incident.
7. Correlate activities across different logs to get a comprehensive picture.
8. Develop theories about what occurred; explore logs to confirm or disprove them.

Kubernetes Audit Log

KUBERNETES AUDIT LOG:
- По умолчанию выключен
- Требует настройки Audit Policy
- Не видит интерактивных сессий

ПОДДЕРЖИВАЕМЫЕ BACKENDS:
- Log Backend
- Webhook Backend

СОБЫТИЯ МОЖНО ОТДАВАТЬ НА:
- Стороннее средство защиты
- Отправлять в SIEM, syslog и т.д.


```
  "kind": "Event",
  "apiVersion": "audit.k8s.io/v1",
  "level": "Request",
  "auditID": "bd93fded-1f5a-4046-a37c-82d8909b2a80",
  "stage": "ResponseComplete",
  "requestURI": "/api/v1/namespaces/default/pods/nginx-
deployment-75ffc6d4d-nt8j4/exec?
command=%2Fbin%2Fbash&container=nginx&stdin=true&stdout=true&tty
=true",
  "verb": "create",
  "user": {
    "username": "kubernetes-admin",
    "groups": [
      "system:masters",
      "system:authenticated"
    ]
  },
  "sourceIPs": [
    "<removed>"
  ],
  "userAgent": "kubectl/v1.21.2 (darwin/amd64)
kubernetes/092fbfb",
```

Контроль безопасности образов

МОЖНО ПРОВЕРЯТЬ НА:

- Соответствие лучшим практикам
- Известные уязвимости
- Вредоносный код
- Наличие чувствительной информации
- Корректность подписи образа
- Соответствие содержимому SBOM
- Соответствие используемого image registry

Контроль безопасности образов: реализация

Специальный исполняемый файл или контейнер, получающий на вход образ контейнера для анализа. Может быть обернут в Kubernetes operator.

ПО ТИПУ АНАЛИЗА:

- Статические
- Динамические

СТАДИИ СКАНИРОВАНИЯ:

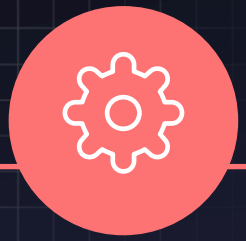
- CI\CD
- Image Registry
- Deploy
- Runtime

Управление секретами

МЕТОДЫ ЗАЩИТЫ:

- Шифрование сторонним Key Management Store (KMS)
- Шифрование, управляемое оркестратором
- Без шифрования

Управление секретами: реализации



МЕСТО ДОСТАВКИ СЕКРЕТА:

- Переменная окружения
- Файл
- В память программы

МОДЕЛИ РАБОТЫ С СЕКРЕТАМИ ПРИЛОЖЕНИЙ:

- Push модель
- Pull модель

РЕАЛИЗАЦИИ:

- Direct API
- Kubernetes Controller/Operator
- Sidecar + MutatingWebhook
- Secrets Store CSI Driver

Безопасность Runtime

МОЖНО РАЗДЕЛИТЬ НА 2 ПОДКЛАССА:

- Host Runtime Security
- Container Runtime Security

ПО ВОЗМОЖНОСТЯМ МОЖНО РАЗДЕЛИТЬ НА:

- Уменьшение поверхности атаки
- Дополнительную изоляцию
- Обнаружение
- Предотвращение
- Реакцию

Безопасность Runtime: реализации

- Механизмы на уровне Pod и системные механизмы Linux
  - Linux capabilities, seccomp, AppArmor, SeLinux, PodSecurityContext & SecurityContext
- На уровне образа
  - distroless images, golden images, rootless containers
- На уровне CRI и Container Runtime
  - Остановка, пауза, слепок контейнера
  - Дополнительная изоляция (Sandbox/MicroVM)
- Агент как DaemonSet
  - Работа через eBPF

- На уровне user space компонента
  - Происходит хук и инжект компонента во все процессы контейнера

Безопасность хоста/ноды

ЗАДАЧИ

- Vulnerability Management
- Logging
- CIS Benchmarks for “Operating Systems”
- Antimalware
- OS host machine hardening
- File Integrity Monitoring (FIM)

Безопасность хоста/ноды: реализации

ОПЕРАЦИОННАЯ СИСТЕМА НА NODE:

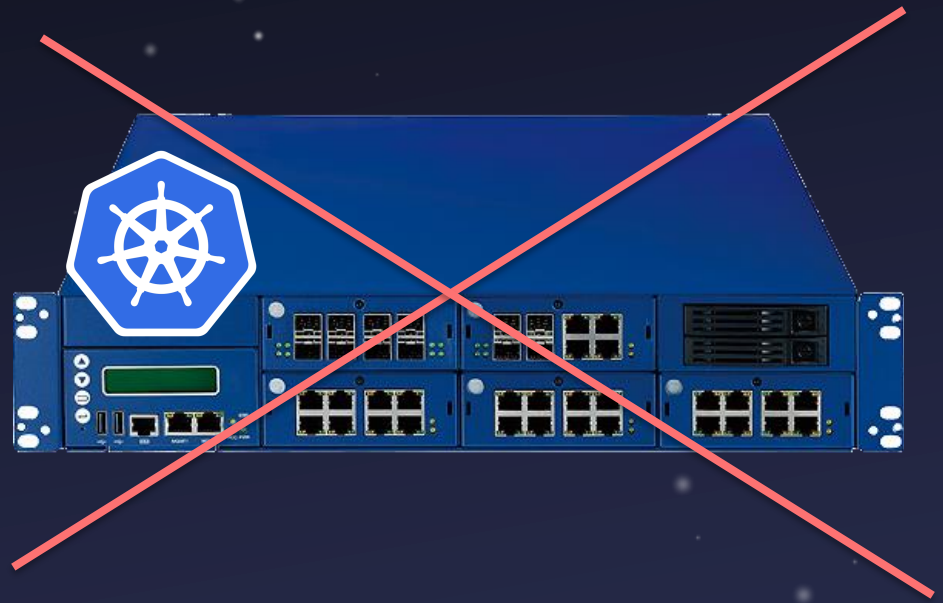
- ОС общего назначения
  - Разный цикл обновлений
  - Большая поверхность атаки
  - Много возможностей для атакующего
  - Много шума от сканеров уязвимостей
  - Много compliance требований и контролей (доступ, целостность, …)
  - Configuration drift
  - Требуется реализовать и использовать все ранее описанное
  - Приводит к неэффективному использованию вычислительных ресурсов
  - Приводит к замедлению работы системы в целом

- Container specific OS
  - Практически полная противоположность ОС общего назначения

Сетевая безопасность

МОЖНО РАЗДЕЛИТЬ НА 2 УРОВНЯ:

- Host Network Security
- Pod Network Security



ЗАДАЧИ:

- Сегментация
  - Микросегментация
- Взаимная аутентификация
- Шифрование
- Контроль входящего и исходящего трафика

Подходы:

- Whitelist
- Blacklist

IP-адрес меняется/переходит от запуска к запуску микросервиса!

Сетевая безопасность: сегментация

3 СПОСОБА

CNI NetworkPolicy – родной межсетевой экран Kubernetes

- На основе iptables или eBPF
- Native – формат политик от Kubernetes
- Custom – расширенный формат политик от разработчиков CNI

Собственная реализация

- User space хуки
- DPI
- …

Service Mesh

- Sidecar proxy (service proxy)
- Shared proxy per node
- Shared proxy per service account (per node)
- Shared remote proxy with micro proxy
- eBPF Accelerated Per-Node Proxy
- Hybrid

Сетевая безопасность: контроль входящего и исходящего трафика

ДЛЯ ВХОДЯЩЕГО ТРАФИКА:

- Host-Based Firewall
- Ingress Gateway
- API Gateway

ДЛЯ ИСХОДЯЩЕГО ТРАФИКА:

- Host-Based Firewall
- CNI NetworkPolicy (Native, Custom)
- Service Mesh
- Конфигурацией NAT
- С помощью Egress Gateways

Сетевая безопасность: шифрование

3 СПОСОБА

На уровне CNI плагинов с помощью:

- IPSec
- Wireguard

На уровне Service Mesh с помощью:

- mTLS

На уровне ваших приложений с помощью:

- SSL/TLS протоколов

Контроль соответствия

COMPLIANCE CHECKS:

- Control Plane
- Data/Workload Plane

Any orchestration system has a number of threats that impact the overall security of the deployment and continued security at runtime. Malicious access to an orchestrator's API, unauthorized access and changes to the key-value store, orchestrator dashboard to control clusters, intercept control plane traffic, API misuse, intercepting application traffic, and so on are all potential threat areas. It is important to use best practices and configuration hardening for any orchestrator to prevent exposure to these threats, several exist. It is also important to monitor and detect any changes to the initial configurations made in runtime to ensure the continued security posture of the cluster. Other security best practices such as minimizing administrative access to the control plane, segregation of duties and principle of least privilege should be enforced.

СУЩЕСТВУЕТ БОЛЬШОЕ КОЛИЧЕСТВО КАК ТЕХНОЛОГИЧЕСКИХ, ТАК И ИНДУСТРИАЛЬНЫХ СТАНДАРТОВ:

- Cis Kubernetes Benchmark
- Mitre Att&CK® Framework
- PCI DSS
- HIPAA
- NIST Application Container Security Framework
- NSA/CISA Kubernetes Hardening Guide

Заключение

- Почти любую функциональность можно реализовать через собственный Kubernetes Operator
- Есть как пересечения с привычными средствами защиты информации, так и уникальные
- Огромные возможности для построения ZeroTrust
- Уменьшение поверхность атаки
- Ориентир на DevSecOps

Полезные материалы

Для глубокого погружения в тему:

1. “Kubernetes: трансформация к SecDevSecOpsSec”
2. “NetworkPolicy — родной межсетевой экран Kubernetes”
3. “SOAR в Kubernetes малой кровью”
4. “Безопасность Kubernetes: Фаза Deception”
5. “Специфика расследования инцидентов в контейнерах”
6. “Сочетание несочетаемого в Kubernetes: удобство, производительность, безопасность”

СПАСИБО ЗА ВНИМАНИЕ!

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