# Стандарт безопасности

контейнеров NIST 800-190
в 2025 году

**Дмитрий Евдокимов**

Founder & CTO в Luntry

# Обо мне

> Я не верю в то, что систему можно сделать надежной и безопасной, не понимая того, как она устроена.

**Основатель и технический директор Luntry**

Более 15 лет опыта в ИБ

**Специализация – безопасность контейнеров и Kubernetes**

Автор ТГ-канала k8s (in) security

## Эксперт в сфере безопасности контейнерных сред

- Организатор конференции «БеКон» по БЕзопасности КОНтейнеров
- Бывший редактор рубрик в журнале «ХАКЕР», автор серии статей
- Автор курса «Cloud Native безопасность в Kubernetes»
- Член программного комитета CFP DevOpsConf и HighLoad++

## Спикер

| | | | | |
|---|---|---|---|---|
| VK Kubernetes | Confidence | ZeroNights | БеКон | HITB |
| DevOpsConf | HackInParis | KuberConf | BlackHat | PHDays |
| Kazhackstan | HighLoad++ | OFFZONE | DevOops | SAS |

# 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**

# План доклада

1. Знакомство с NIST SP 800

2. Что к чему в 2025 ?


# DISCLAIMER:

- Используется перевод и есть упрощения*
- В основном используются оригинальные формулировки - без привязки к K8s
- Анализ не предполагал оптимизации - только расширение концепции
- Данный документ не является эталоном и нужно учитывать собственную специфику каждого окружения, проекта, кластера и т.д.

* - Всегда полезно читать оригинал и тут всего 63 страницы ;)

# Знакомство с NIST SP 800

# Серия NIST SP 800

- **Это набор руководств, рекомендаций, технических спецификаций и ежегодных отчётов Национального института стандартов и технологий США (NIST) в области компьютерной безопасности.**
- **215 документов**
  - https://csrc.nist.gov/publications/sp800

# Что с Cloud Native ?

| Серия | Номер | Название | Дата |
|---|---|---|---|
| SP | 800-145 | The NIST Definition of Cloud Computing | 28/9/2011 |
| **SP** | **800-190** | **Application Container Security Guide** | **25/9/2017** |
| SP | 800-204 | Security Strategies for Microservices-based Application Systems | 07/8/2019 |
| SP | 800-204A | Building Secure Microservices-based Applications Using Service-Mesh Architecture | 27/5/2020 |
| SP | 800-210 | General Access Control Guidance for Cloud Systems | 31/7/2020 |
| SP | 800-204B | Attribute-based Access Control for Microservices-based Applications using a Service Mesh | 06/8/2021 |
| SP | 800-204C | Implementation of DevSecOps for a Microservices-based Application with Service Mesh | 08/3/2022 |
| SP | 800-207A | A Zero Trust Architecture Model for Access Control in Cloud-Native Applications in Multi-Cloud Environments | 13/9/2023 |
| SP | 800-204D | Strategies for the Integration of Software Supply Chain Security in DevSecOps CI/CD Pipelines | 12/2/2024 |
| SP | 800-201 | NIST Cloud Computing Forensic Reference Architecture | 30/7/2024 |
| SP | 800-233 | Service Mesh Proxy Models for Cloud-Native Applications | 16/10/2024 |
| SP | 800-228 | Guidelines for API Protection for Cloud-Native Systems (Draft) | 25/03/2025 |

# История

**20 марта 2013:** Начальный релиз Docker

**7 июня 2014:** Начальный релиз Kubernetes

**10 июля 2015:** Релиз Kubernetes версии 1

**21 июля 2015:** Создание Cloud Native Computing Foundation (CNCF)

**28 марта 2017:** Релиз Kubernetes версии 1.6

**4 апреля 2017:** Черновой релиз NIST SP 800-190

**30 июня 2017:** Релиз Kubernetes версии 1.7

**19 июля 2017:** Релиз Runtime и Image Specification v1.0.0 от OCI

**28 августа 2017:** Релиз Kubernetes версии 1.8

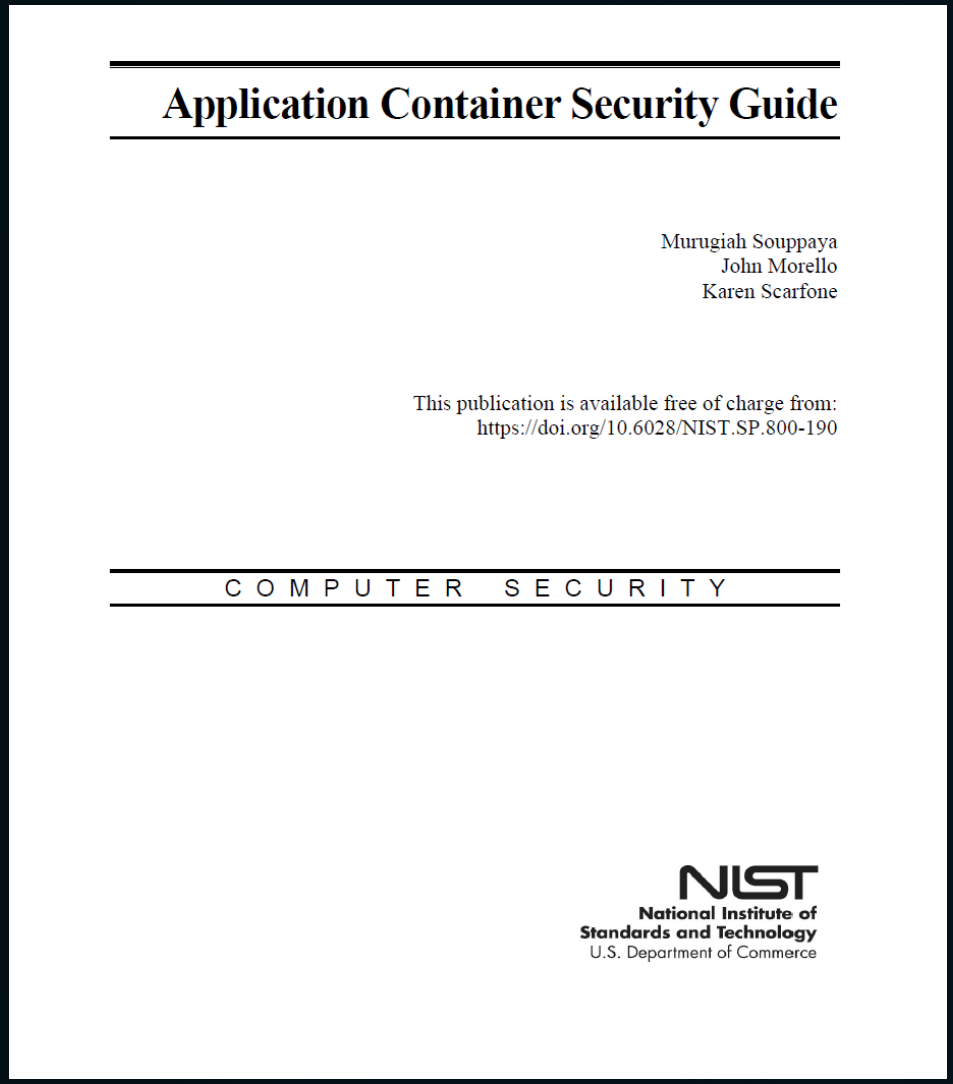
**25 сентября 2017:** Финальный релиз NIST SP 800-190

**…**

**Сегодня 28 марта 2025 года ;)**

# NIST SP 800-190

Разработан для обеспечения рекомендаций по безопасному использованию технологий контейнеризации, которые стали широко распространены в современной разработке и развертывании программного обеспечения.



Application Container Security Guide

Murugiah Souppaya
John Morello
Karen Scarfone

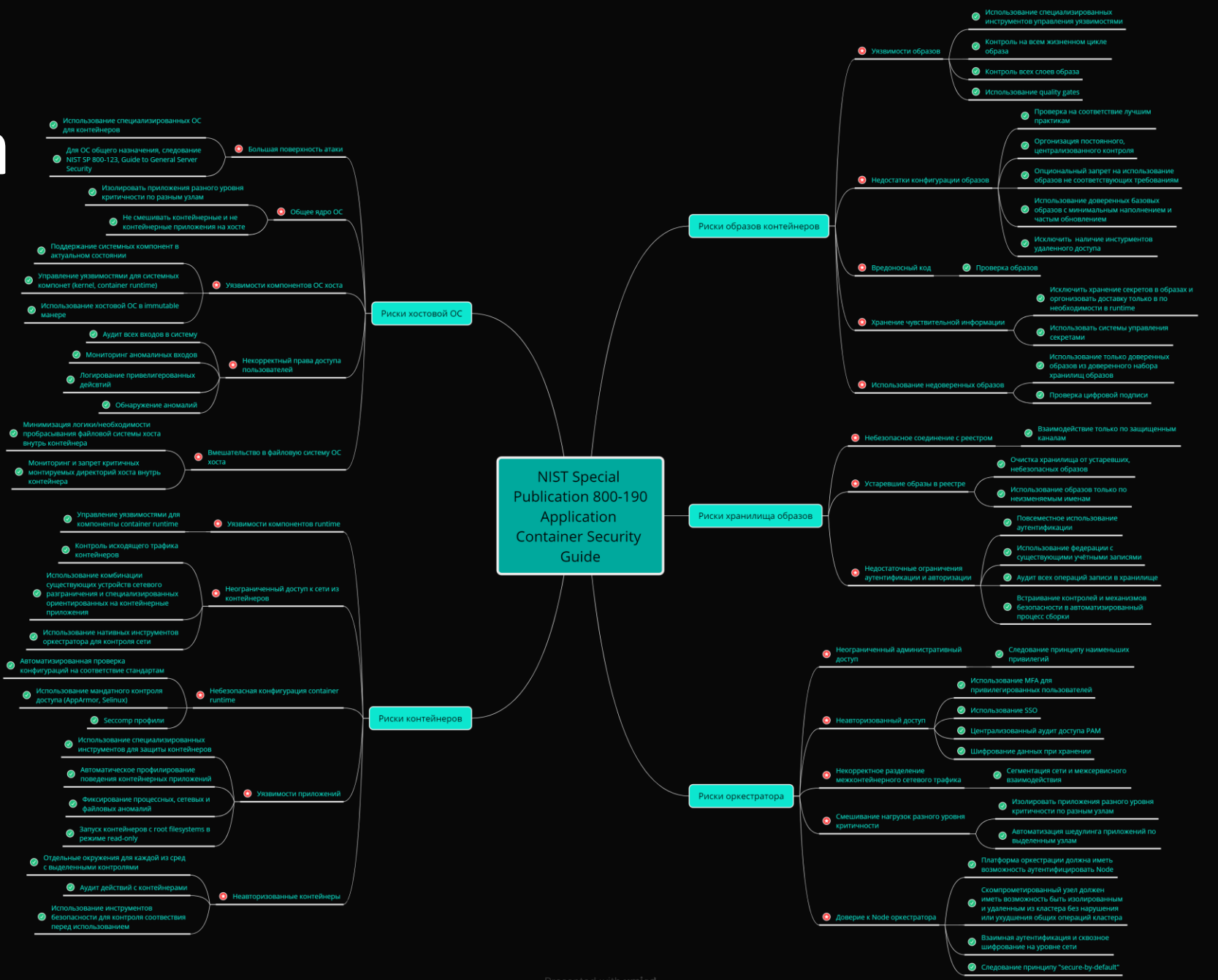
This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-190

COMPUTER SECURITY

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

https://csrc.nist.gov/pubs/sp/800/190/final

# Полная картина



# Упоминание Kubernetes

### 2.3.3 Container Deployment and Management

Tools known as *orchestrators* enable DevOps personas or automation working on their behalf to pull images from registries, deploy those images into containers, and manage the running containers. This deployment process is what actually results in a usable version of the app, running and ready to respond to requests. When an image is deployed into a container, the image itself is not changed, but instead a copy of it is placed within the container and transitioned from being a dormant set of app code to a running instance of the app. Examples of orchestrators are Kubernetes [14], Mesos [15], and Docker Swarm [16].

### 4.1.4 Embedded clear text secrets

Secrets should be stored outside of images and provided dynamically at runtime as needed. Most orchestrators, such as Docker Swarm and Kubernetes include native management of secrets. These orchestrators not only provide secure storage of secrets and 'just in time' injection to containers, but also make it much simpler to integrate secret management into the build and

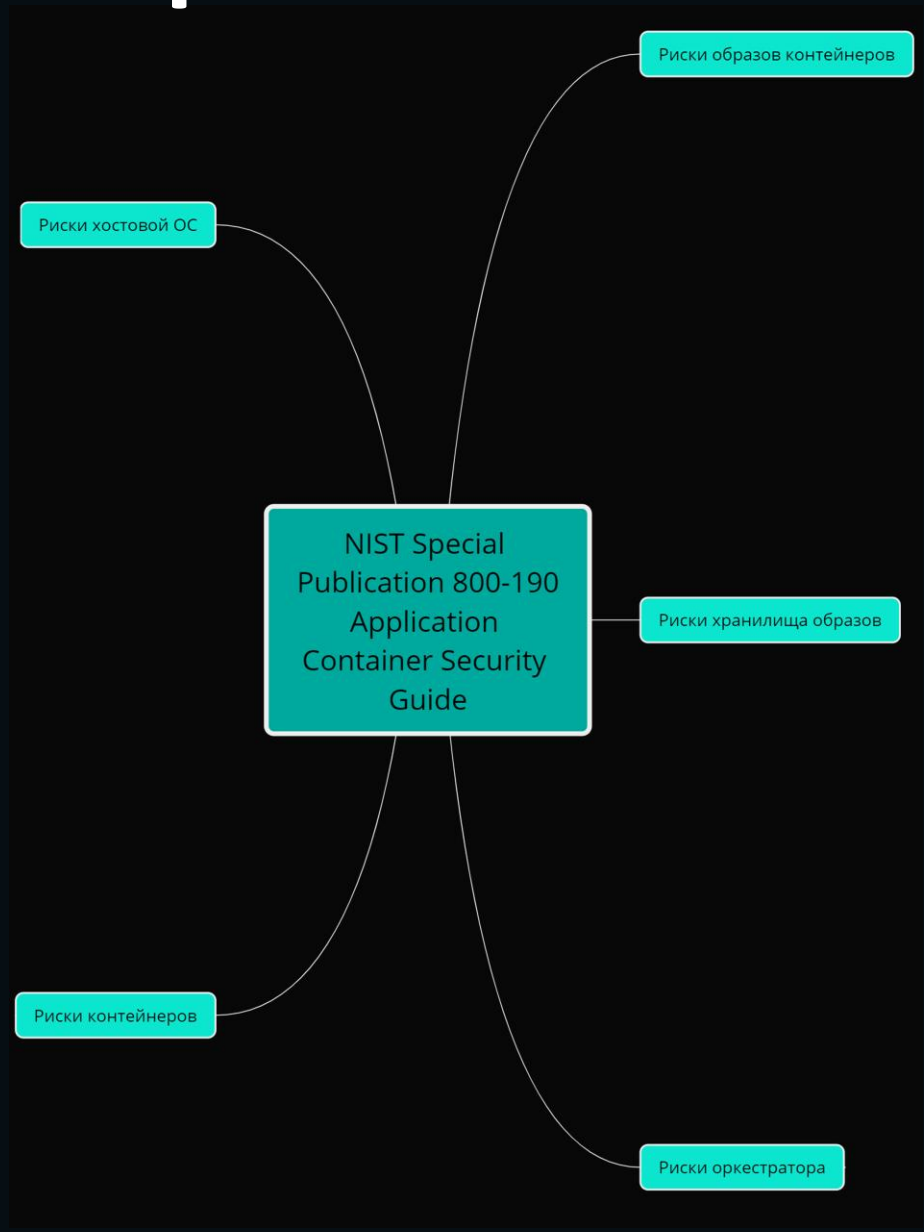
# Что к чему в 2025 ?

# 2025 год

1. С 2017 года экосистема контейнеризации эволюционировала
2. С 2017 года угроз стало больше и некоторые эволюционировали
3. Kubernetes стандарт де-факто среди оркестраторов контейнеров
4. Текущие актуальные/поддерживаемые версия Kubernetes
   - 1.30.10
   - 1.31.6
   - 1.32.2
5. В Kubernetes появилось множество улучшений и встроенных механизмов безопасности
   - Появился и исчез PSP, на смену пришел PSA и Validating Admission Policy
   - Поддержка AppArmor в stable, Default seccomp стал GA
   - Появились Kubelet-in-UserNS (Rootless mode), Ephemeral Containers, UserNamespacesSupport, Forensic Container Checkpointing
   - И т.д.

# Основные компоненты и риски

1. Риски хостовой ОС
2. Риски образов контейнеров
3. Риски хранилища образов
4. Риски оркестратора
5. Риски контейнеров



# Возможные дополнения в компонентах

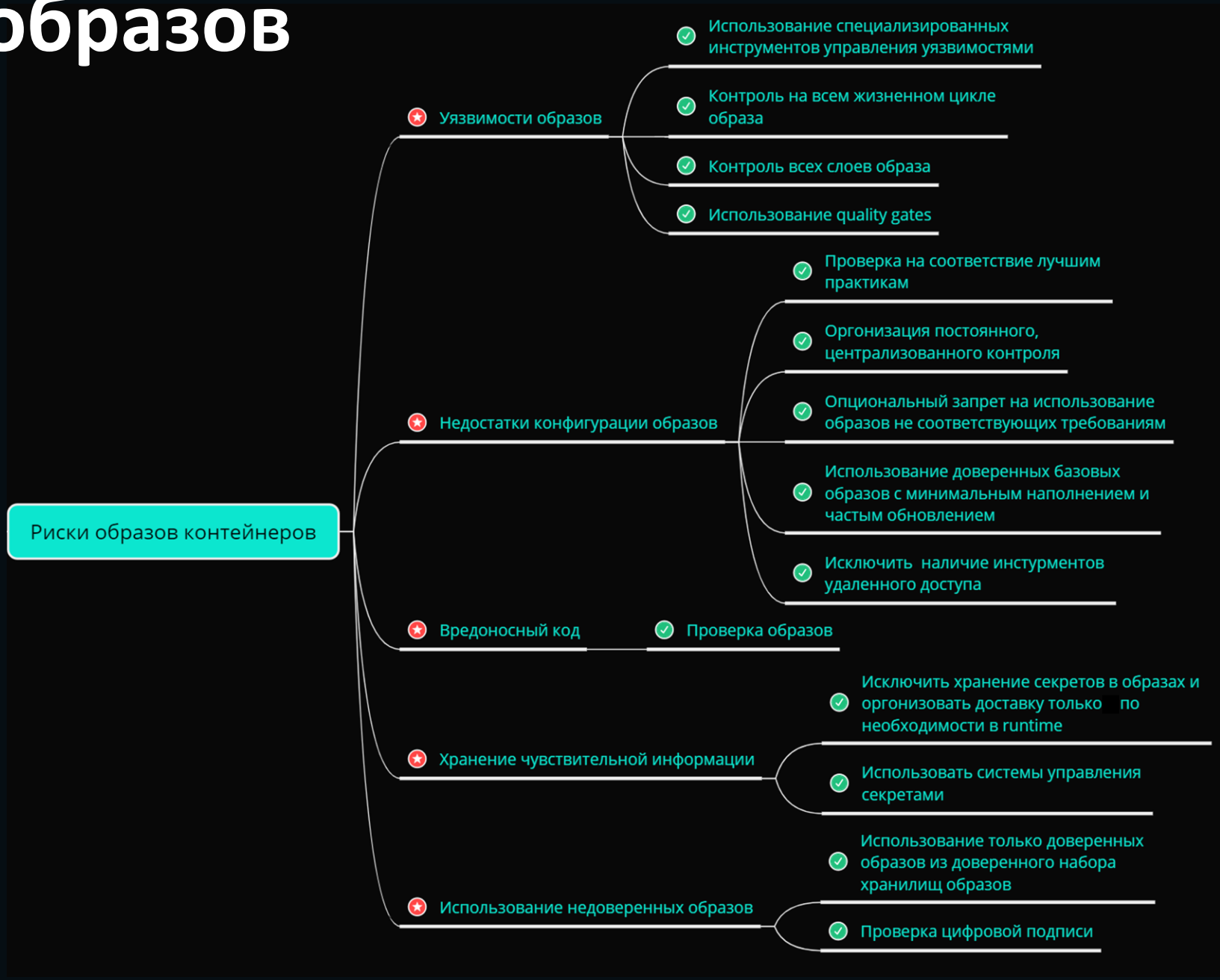

**1. Риски появления вредоносных Kubernetes-ресурсов (YAML)**

- В Kubernetes практически все есть YAML
- Контрмеры:
  - Встроенные admission controllers (LimitRange, ResourceQuota  и т.д.)
  - Policy Engine (Kyverno, OPA Gatekeeper)
  - Kubernetes Audit Log

**2. Риски проблем рядом стоящих интегрированных систем**

- CI/CD, системы управления секретами, системы хранения данных, системы мониторинга, …
- Контрмеры:
  - Аутентификация, авторизация
  - Сетевые политики
  - Настройка в соответствии с лучшими практиками для данного класса решений
  - …

# Риски образов



# Возможные дополнения

**1. Риск использования не доверенных, не разрешенных, скомпрометированных компонент**

- В современных микросервисах 60-90% это сторонний код
- Безопасность цепочки поставки (supply chain)
- Контрмеры:
  - Компонентный анализ (SBOM)



**2. Риск использования ПО двойного назначения**

- LotL-атака (Living off the Land), GTFOBins
- Контрмеры:
  - Минимизация образов


**3. Риск не безопасной сборки образа**

- DinD сборка
- Контрмеры:
  - Использовать инструменты без доступа к container runtime socket

# Риски хранилища образов

# Возможные дополнения

**1. Риск утечки и организации скрытого канала передачи**

- В registry можно хранить любые данные и из одного контура передавать информацию в другой
- Контрмеры:
  - Registry Staging
  - Запрет на push из prod кластера

## Artifacts & Images

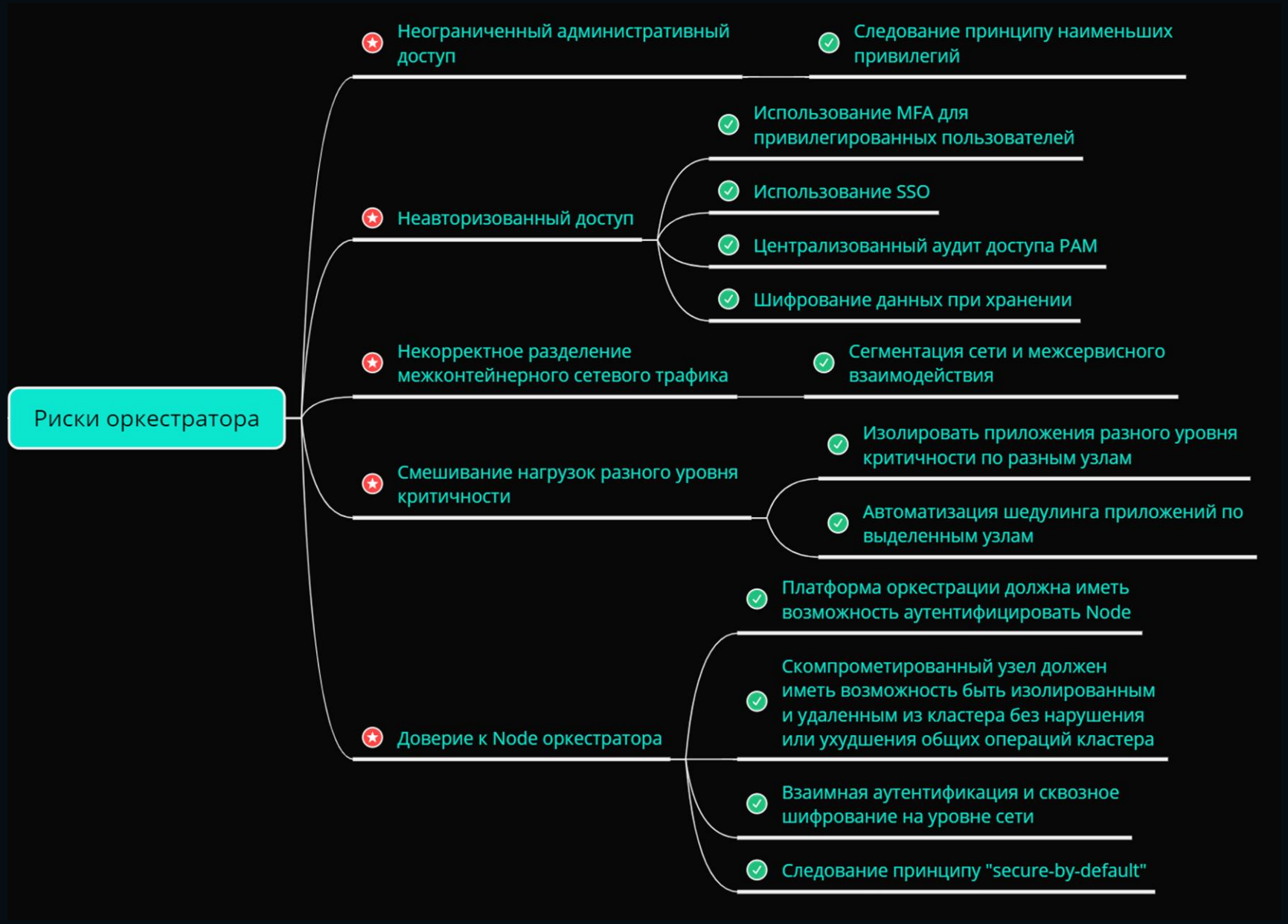
### Registry Staging

Due to the use of open source components that are often pulled from public sources, organizations should create several stages of registries in their pipelines. Only authorized developers should be able to pull base images from public registries and store them in an internal registry for wide consumption within the organization. It is also advised to have separate private registries for keeping development artifacts per team or group, and finally a staging or pre-production registry for images ready for production. This enables tighter control over the provenance and security of open source components, while enabling different types of testing for stages in the CI/CD chain.

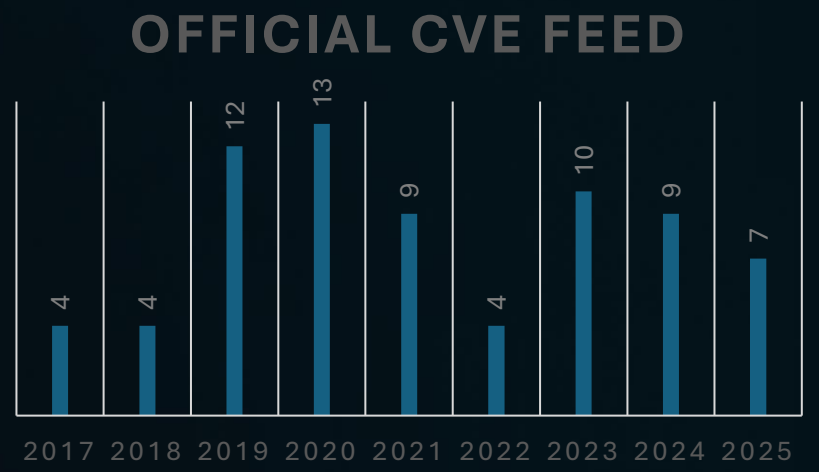
For any registry used, access control through a dedicated authentication and permission model must be implemented. Use mutually authenticated TLS for all registry connections (among other interactions within the architecture).

CNCF Cloud Native Security Whitepaper

# Риски оркестратора



# Возможные дополнения




## 1. Риск уязвимых системных компонент

- В компонентах Kubernetes находятся уязвимости
- Контрмеры:
  - Обновлять системные компоненты
  - Использовать Policy Engine для митигации

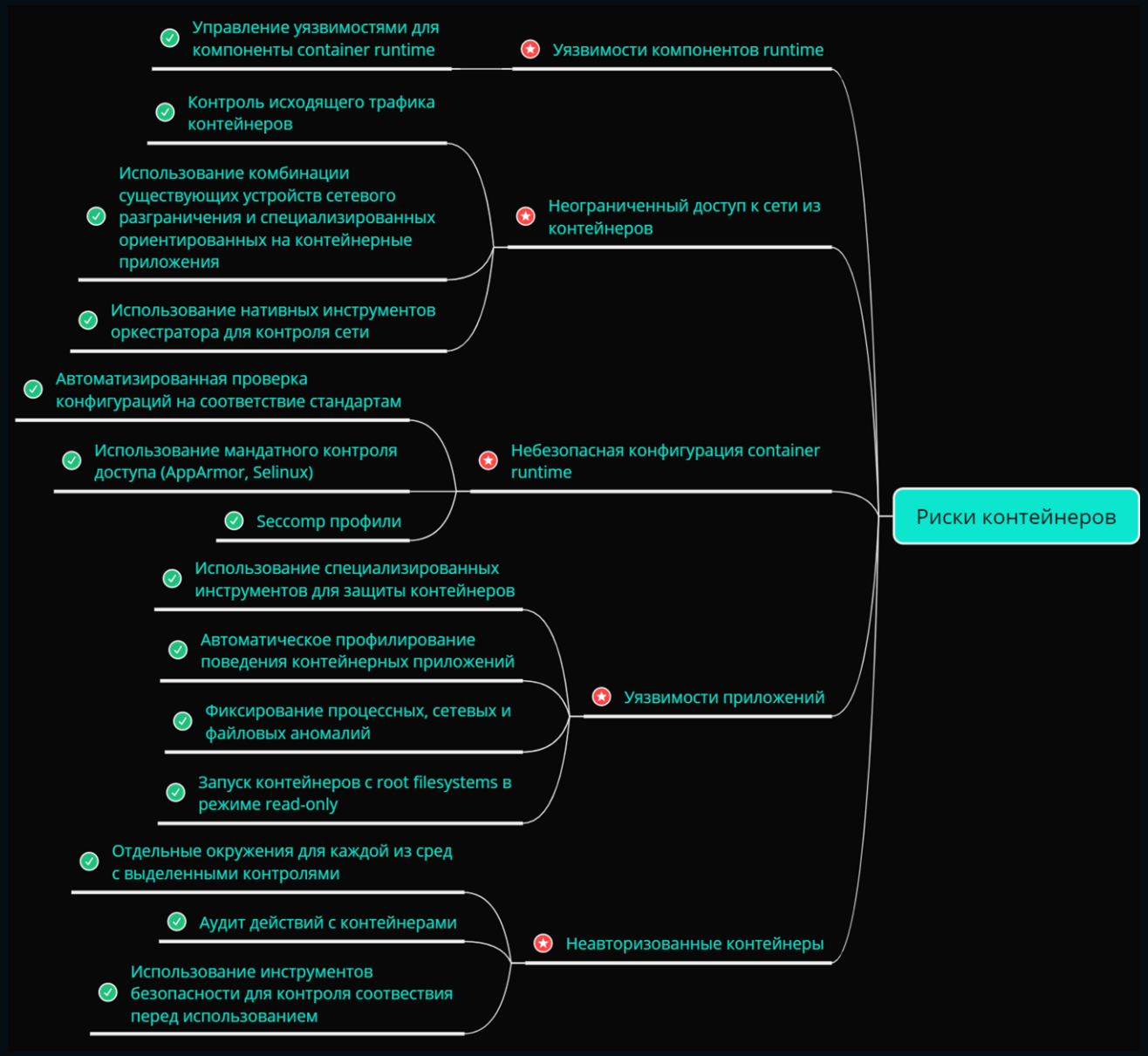
## 2. Риск небезопасной конфигурации системных компонент

- Оркестратор можно развернуть без учета рекомендаций по безопасности, что откроет определённые вектора атак и/или не позволят использовать механизмы безопасности
- Контрмеры:
  - Проверять по CIS Kubernetes benchmark
  - Проверять по внутренним требованиям

## 3. Риск избыточных привилегий

- Права доступа у Users и Groups через Roles и ClusterRoles
- Контрмеры:
  - Контроль RBAC - использовать Policy Engine

# Риски контейнеров



# Возможные дополнения

**1. Риск «Неограниченный доступ к сети из контейнеров»**

- Контрмеры:
  - Network Policy
  - Service Mesh
  - Ingress, Egress gateway

**2. Риск «Небезопасная конфигурация container runtime»**

- Контрмеры:
  - PodSecurityContext, SecurityContext
  - Policy Engine, PSA, Validating Admission Policy

**3. Риск «Неавторизованные контейнеры»**

- Контрмеры:
  - Policy Engine
  - Проверка подписи
  - GitOps

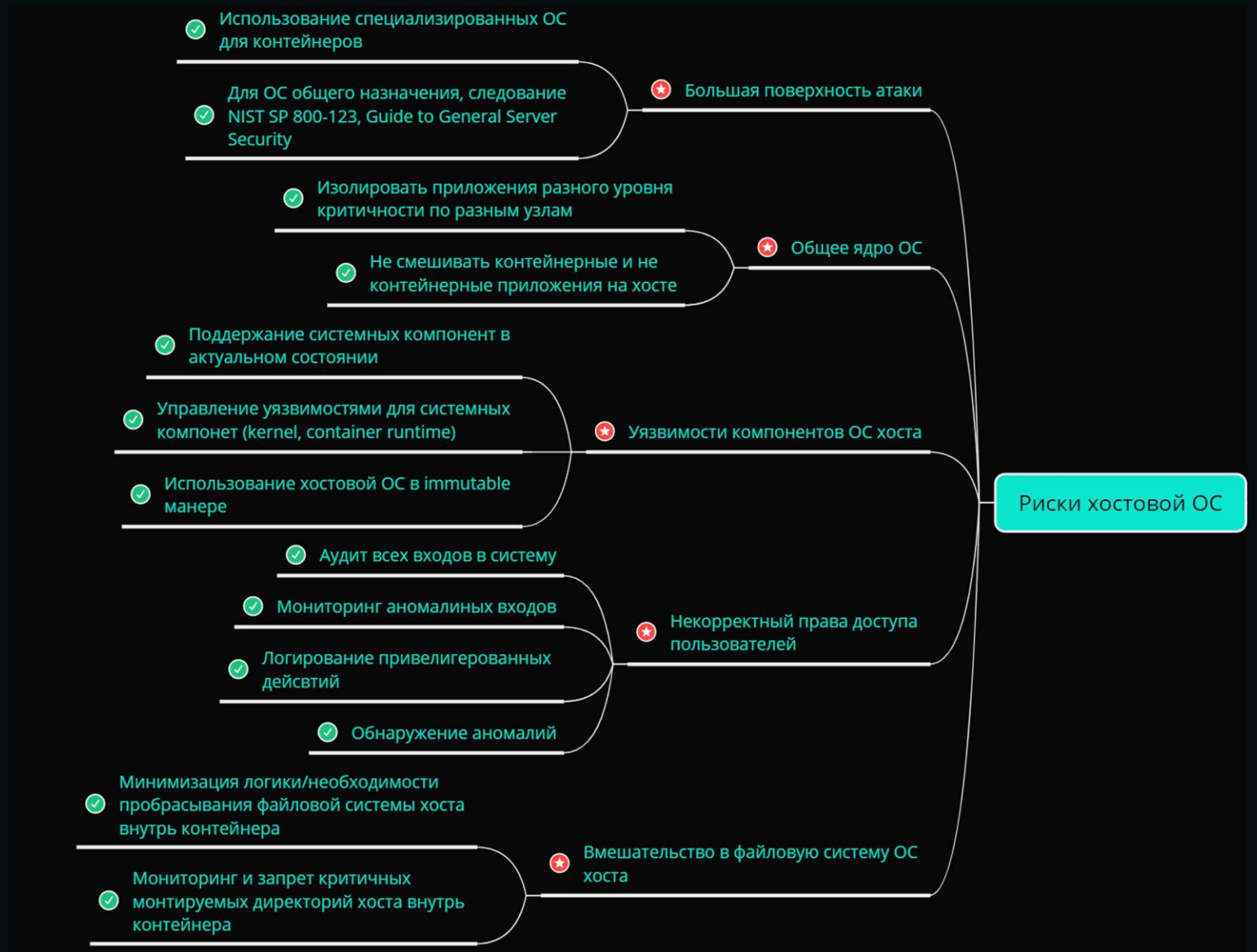
**4. Риск избыточных привилегий**

- Привилегированный Service Account
- Контрмеры:
  - Контроль RBAC

**5. Риск отказа в обслуживании**

- Контрмеры:
  - ingressClassName
  - Запросы и лимиты

# Риски хостовой ОС



# Возможные дополнения

**1. Риск «Общее ядро ОС»**

- Контрмеры:
  - Node-based multi tenancy
  - Seccomp профили
  - AppArmor, SeLinux политики
  - Альтернативные container runtimes с высокой изоляцией

**2. Риск «Некорректные права доступа пользователей»**

- Контрмеры:
  - Выполнение действий на Node только через специальный operator
  - Доступ на Node через специальный временный привилегированный контейнер
  - Фиксировать все интерактивные сессии на Node

# Заключение

# Итоги

1. Не учитывается специфика современных Kubernetes кластеров

2. Можно добавить 2 категории ключевых компонентов

3. Можно добавить 9 новых рисков

4. Можно добавить 31 новую контрмеру

# Спасибо за внимание!


Дмитрий Евдокимов

Founder&CTO Luntry

Email: de@luntry.ru

Twitter: @evdokimovds

@Qu3b3c

Channel: @k8security

Site: www.luntry.ru


luntry.ru

k8security

luntrysolution

ХАРДКОРНЫЕ ДОКЛАДЫ Новости продуктов Безопасность NIST 800-190

# ОЦЕНИТЕ ДОКЛАД

Дмитрия Евдокимова

ХАРДКОРНЫЕ ДОКЛАДЫ Новости продуктов