



БЕЗОПАСНОСТЬ КОНТЕЙНЕРОВ  
И KUBERNETES  
ДЛЯ **DevSecOps**  
СПЕЦИАЛИСТОВ



Дмитрий  
Евдокимов

Founder&CTO Luntry

# В предыдущих сериях вебинарах\*

01

## Для CISO

- Те кто формируют направление развития

02

## Для специалистов SOC

- Те кто смотрят за событиями в системе

03

## Для специалистов анализа качества

- Те кто проводят ПСИ, осуществляют сертификацию ПО, разбираются с “черными ящиками”

\*- слайды и видео доступно на <https://luntry.ru/research>

# 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  
DevOpsConf  
Kazhackstan

Confidence  
HackInParis  
HighLoad++

ZeroNights  
KuberConf  
OFFZONE

БеКон  
BlackHat  
DevOops

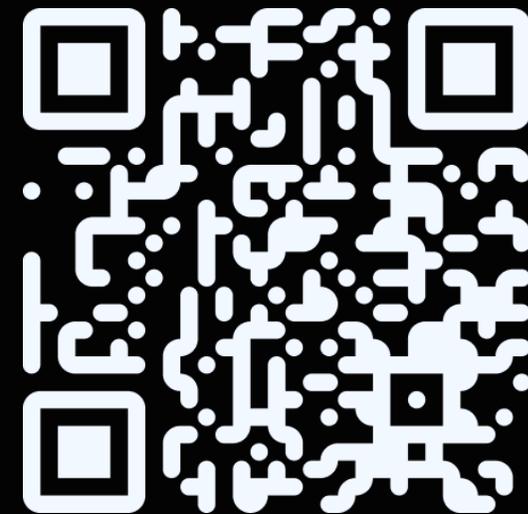
HITB  
PHDays  
SAS

# 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/>

Получение сертификата ФСТЭК планируется в четвертом квартале 2025 года.



# LUNTRY.RU

# Функциональность Luntry

---

Контроль всех  
Kubernetes-ресурсов

---

Контроль состояния Kubernetes-  
кластеров

---

Контроль соответствия кластера  
стандартам



**LUNTRY**

---

Управление уязвимостями  
образов и best practice

---

Сетевая безопасность

---

Анализ прав доступа

---

Защита Runtime

# Функциональность Luntry

---

Контроль всех  
Kubernetes-ресурсов

---

Контроль состояния Kubernetes-  
кластеров

---

Контроль соответствия кластера  
стандартам



**LUNTRY**

---

Управление уязвимостями  
образов и best practice

---

Сетевая безопасность

---

Анализ прав доступа

---

Защита Runtime

# План вебинара

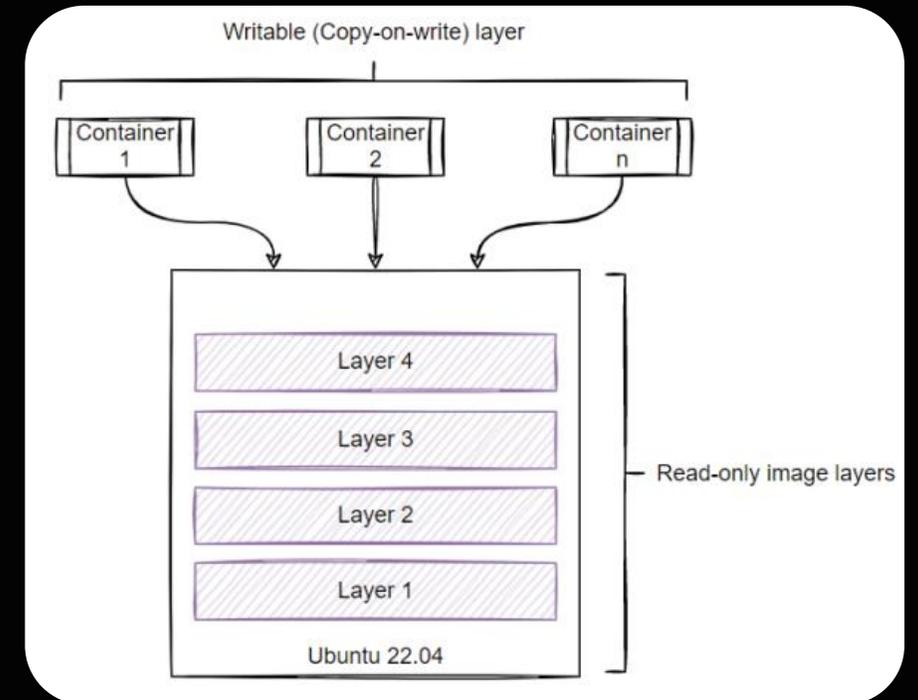
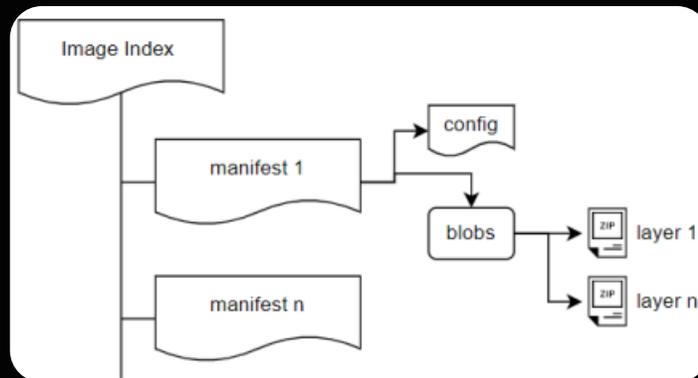
- 01 Теория про образы контейнеров
- 02 Жизненный путь образа
- 03 Анализ образа
- 04 Как Luntry помогает решить задачи, связанные с безопасностью образов
- 05 Выводы



# ТЕОРИЯ ПРО ОБРАЗЫ КОНТЕЙНЕРОВ

# Структура образа

- Образ собирается из слоев описанных в Dockerfile
- Container Image это неизменяемый пакет файлов операционной системы, кода приложения и любых зависимостей приложения
  - Union File System
    - OverlayFS как реализация
  - OCI спецификация
    - OCI Image Format, OCI Distribution Specification, OCI Runtime Specification



# Полезные инструменты

- [Dive](#)
- [Skiff](#)
- [skopeo](#)

The screenshot shows the Dive tool interface with the following sections:

- [Layers]**: A table listing layers with columns for Cmp, Image ID, Size, Command, Permission, UID:GID, and Size.
- [Image & Layer Details]**: A section showing the layer command and image efficiency score (99%).
- Filetree**: A tree view of the file system structure, including directories like bin, dev, etc, home, lib, media, mnt, proc, root, run, and/sbin.

Cmp	Image ID	Size	Command	Permission	UID:GID	Size
	sha256:cd7100a72410606589	4.1 MB	FROM sha256:cd7100a72410606589	drwxr-xr-x	0:0	805 kB
	sha256:f03b1ccbacace8c82e	2.1 kB	#(nop) ADD file:63d4894bd0857354b	drwxr-xr-x	0:0	0 B
	sha256:d2a26525cdac6c4358	0 B	mkdir /root/example	drwxr-xr-x	0:0	251 kB
	sha256:9f50561518484bb62a	2.1 kB	cp /somefile.txt /root/example/so	drwxr-xr-x	0:0	0 B
	sha256:edf209d09263ab03c7	2.1 kB	cp /somefile.txt /root/example/so	drwxr-xr-x	0:0	2.7 MB
	sha256:bb70c7aab83602e8a6	2.1 kB	cp /somefile.txt /root/example/so	drwxr-xr-x	0:0	0 B
	sha256:b0a712d3d00bc83c48	2.1 kB	mv /root/example/somefile3.txt /r	drwxr-xr-x	0:0	0 B
	sha256:05e8660b8f90e77b93	0 B	rm -rf /root/example/	drwxr-xr-x	0:0	0 B

**Layer Command**

```
/bin/sh -c #(nop) ADD file:093f0723fa46f6cddb6f7bd146448bb70ecce54254c35701feeceb956414622f in /
```

**Image efficiency score: 99 %**  
**Potential wasted space: 6.2 kB**

Count	Total Space	Path
2	4.2 kB	/root/example
2	2.1 kB	/root/example/somefile3.txt

**Filetree**

```
bin
dev
etc
home
lib
media
  cdrom
  floppy
  usb
mnt
proc
root
run
sbin
  acpid -> /bin/busybox
  adjtimex -> /bin/busybox
  apk
  arp -> /bin/busybox
  blkid -> /bin/busybox
  blockdev -> /bin/busybox
  depmod -> /bin/busybox
  fbsplash -> /bin/busybox
  fdisk -> /bin/busybox
  findfs -> /bin/busybox
  fsck -> /bin/busybox
  fstrim -> /bin/busybox
  getty -> /bin/busybox
  halt -> /bin/busybox
  hdparm -> /bin/busybox
  hwclock -> /bin/busybox
  ifconfig -> /bin/busybox
  ifdown -> /bin/busybox
  ifenslave -> /bin/busybox
  ifup -> /bin/busybox
  init -> /bin/busybox
  inotifyd -> /bin/busybox
  insmod -> /bin/busybox
```



# ЖИЗНЕННЫЙ ПУТЬ ОБРАЗА

# Жизненный путь образа

- CI\CD Pipeline
  - Сборка образа
  - Анализ образа
  - Security/Quality gates
    - Возможна блокировка
  - Подпись образа
- Image registry
  - Хранение
  - Очистка/ротация
  - Анализ образов
  - Security/Quality gates
- Runtime
  - Запуск
  - Анализ образов
  - Security/Quality gates



# Сборка образа

- Сборка в Docker-in-Docker (DinD)
  - **Плохо! Не делайте так!**
- Используйте безопасные инструменты сборки
  - Не использующие Docker demon
- Используйте минималистичные базовые образы
  - Tiny/slim/minimal/distroless образы
- Используйте практику Golden image
  - Может сочетаться или дополнять концепцию минималистичных образов
- Используйте multi-stage сборку
  - Совместимо со всем выше сказанным

# Полезные инструменты

- [Kaniko](#)
- [Buildah](#)
- [BuildKit](#)

A presentation slide for the BEKON '25 conference. The background is a dark blue space-themed image with a wireframe landscape at the bottom. The text is centered and white. At the top, it says '3 июня 2025' with a location pin icon and 'Москва, LOFT HALL#2'. Below that is the main title 'БЕКОН '25' in a large, bold, sans-serif font. Underneath the title is the subtitle 'Конференция по БЕзопасности КОНтейнеров и контейнерных сред'. The main content of the slide is the question 'Чем собирать контейнеры параноику?' in a large font. Below the question is the speaker's name 'Кожуховский Михаил' and the company name 'Flowmaster'.

Доклад «[Чем собирать контейнеры если вы параноик?](#)», Бекон 2025

# Домены безопасности образов

5 доменов:

Анализ Dockerfile  
на лучшие практики

Поиск чувствительной информации

Анализ на известные  
уязвимости

Анализ на вредоносный код

Компонентный анализ

# Security/Quality Gates – ворота качества

- CI
  - Уведомление
  - Блокировка
- Image registry
  - Уведомление
- Runtime
  - Уведомление



Блокировка на Deploy, Runtime – опасная и плохая практика!

[«Безопасность Kubernetes кластеров: вредные советы»](#), DevOpsConf 2024

# Подпись образа

- Самый последний шаг в CI pipeline
- Только если все проверки прошли успешно и Security/Quality Gates не заблокировали CI pipeline
- Проверка подписи образа может быть
  - На Deploy в Admission Controller
  - В момент запуска на Node в CRI
    - Умеет CRI-O с версии 1.28
    - В Kubernetes версии 1.28 появилась новая image pull ошибка с кодом SignatureValidationFailed

[«Подпись и валидация образов в Kubernetes»](#), Вебинар Luntry

# Полезные инструменты

- [Sigstore](#)
- [Notary project](#)
- [Connaisseur](#)
- [In-toto](#)
- [Kyverno](#)
  - verifyImages политики
- [policy-controller](#)

```
spec:
  validationFailureAction: Enforce
  background: false
  rules:
    - name: verify-image
      match:
        any:
          - resources:
              kinds:
                - Pod
      verifyImages:
        - imageReferences:
            - "ghcr.io/kyverno/test-verify-image*"
          mutateDigest: true
        attestors:
          - entries:
              - keys:
                  publicKey: |
                    -----BEGIN PUBLIC KEY-----
                    MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE8nXRh950IZbRj8Ra/N9sbqOPZrFM
                    5/KAQN0/KjHcorm/J5yctVd7iEcnssRQjU917hmKO6JWVGHpDguIyakZA==
                    -----END PUBLIC KEY-----
```

# Registry Staging

- Доверенные источники
- Разделение сред
- Разделение доступов
- Разделение стратегий
  - Очистки
  - Анализа

**13.2** A lack of segregation between production and non-production container registries may result in insecure images deployed to the production environment.

**a.** Consider using two registries, one for production or business-critical workloads and one for development/test purposes, to assist in preventing image sprawl and the opportunity for an unmaintained or vulnerable image being accidentally pulled into a production cluster.

**"Guidance for Containers and Container Orchestration Tools"**  
от PCI Security Standards Council

## 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).

**"Cloud Native Security Whitepaper"**  
от CNCF



# АНАЛИЗ ОБРАЗА

# Анализ Dockerfile на лучшие практики

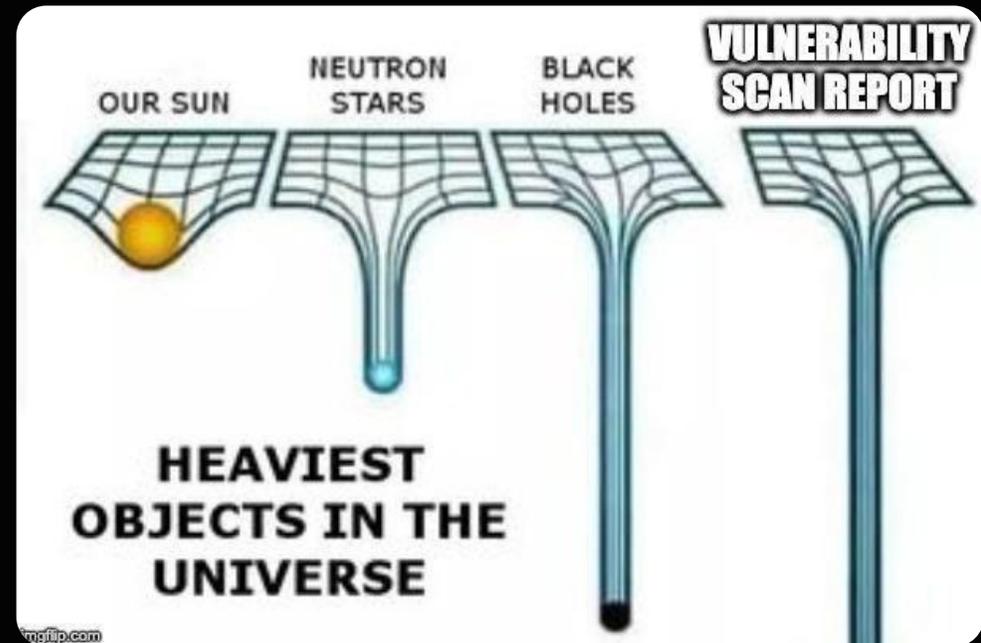
- Мысли:
  - Проверяем на лучшие практики, опасные/нежелательные конструкции, эксплоиты и что угодно в соответствии с внутренними регламентами
- Проверки:
  - Docker Building best practices
  - CIS Docker Benchmarks
  - Собственные требования
- OpenSource инструменты: [Dockle\\*](#), [Hadolint](#), [KICS](#), [Checkov](#), [Semgrep](#)

# Компонентный анализ

- Мысли:
  - Инвентаризация всего стороннего внутри образа
  - Важно для идентификации уязвимых компонент, актуальности версий, лицензионной чистоты
  - Форматы SBOM: Software Package Data Exchange (SPDX), CycloneDX (CDX), Software Identification (SWID)
- OpenSource инструменты: [syft](#), [cdxgen](#), [trivy](#)

# Анализ на известные уязвимости

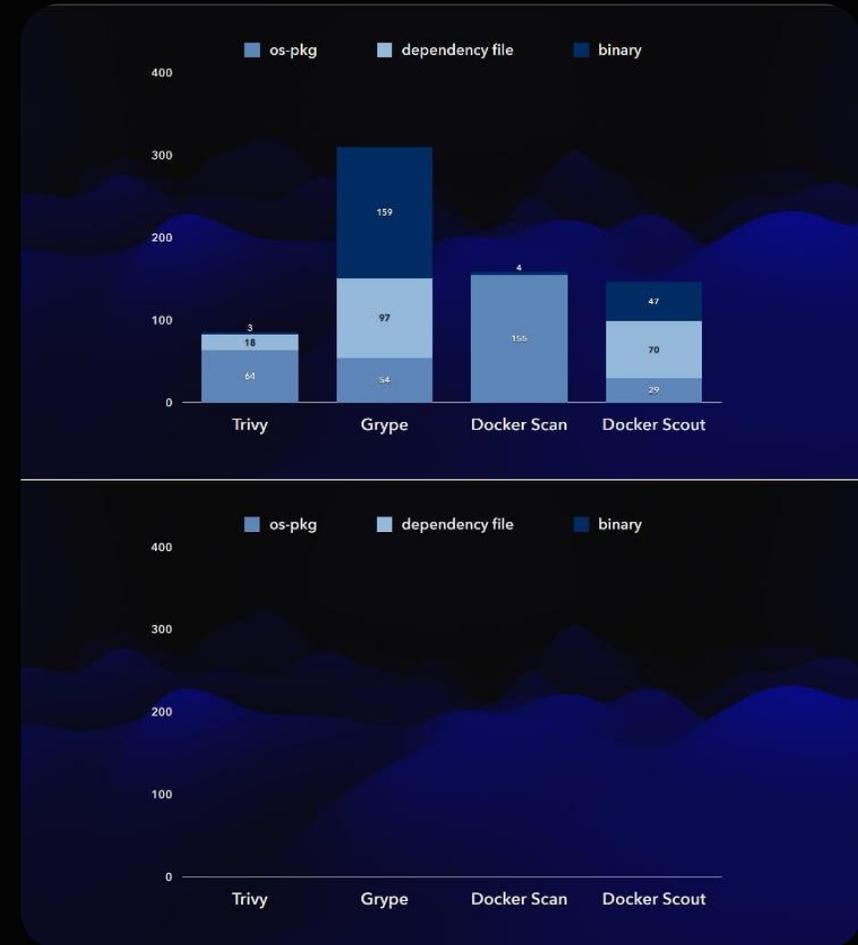
- Мысли:
  - Важная часть процесса управления уязвимостями
  - В любой момент времени всегда есть какая-то незакрытая уязвимость
  - Окружений без уязвимостей не бывает
  - Важно уметь приоритезировать уязвимости
- OpenSource инструменты:  
[Trivy](#), [Grype](#), [Clair](#)



# Проблемы поиска известных уязвимостей



Все подходы на базе статического анализа позволяют защититься только лишь от легитимного пользователя, но не являются никакой помехой для вредоносного!



# Работа с уязвимостями

- Application Security Orchestration and Correlation (ASOC)
- Экспорт результатов в CycloneDX формате в ASOC платформы
- ASOC - инструмент для работы с найденными уязвимостями
  - Агрегирует данные из разных сканеров (DAST, SAST, SCA, ...)
  - Помогает делать триаж уязвимостей (работа с дубликатами тд)
  - Назначение задач на исправление, приоритезация, контроль исправлений
  - Интеграция с VCS, tasktracker, registry, сканерами
  - Формирование отчетов по уязвимостям
- Инструменты: [DefectDojo](#), [Hexway Vampy](#)



# Поиск чувствительной информации

- Мысли:
  - Важно сканировать в CI и блокировать при обнаружении
    - При попадании образа с секретом в registry нужно заводить инцидент и ротировать секрет!
- Подходы:
  - На основе сигнатур
  - На основе энтропии
  - Гибридный подход
- OpenSource инструменты: [SecretScanner](#), [truffleHog](#)

# Анализ на вредоносный код

- Мысли:
  - Важно обнаруживать не только вредоносный код, но и код двойного назначения (GTFOBins)
  - Не тащите и не запускайте непонятные образы извне!
  - Комбинируйте с динамикой – Runtime Security
- OpenSource инструменты: [Package Hunter](#), [Malcontent](#), [PISC](#)

# Проблемы OpenSource

- Отсутствие профессиональной поддержки и обслуживания
- Отсутствие российской специфики (ОС, feeds, сертификации)
- Необходимость высокой внутренней экспертизы для работы с данными проектами
- Зависимость от внутренних носителей знаний, которые могут уйти из компании
- Как правило отсутствие энтерпрайзных фич в бесплатных версиях
- Сложность интеграции и совместимости
- Для решения данной задачи нужен целый разнородный набор инструментов, над которым придётся писать еще обвязку и поддерживать ее
- Скрытые затраты на “бесплатное”



# КАК LUNTRY ПОМОГАЕТ РЕШИТЬ ЗАДАЧИ, СВЯЗАННЫЕ С БЕЗОПАСНОСТЬЮ ОБРАЗОВ

# Возможности Luntry

- Встраивание в CI\CD, Registry, Runtime
- Инвентаризация использования образов
- Компонентный анализ (SBOM)
- Сканирование на известные уязвимости
- Поддержка отечественных ОС и БДУ ФСТЭК
- Приоритизация уязвимостей CVSS, Severity, Type(RCE/DoS), Exploit, KEV, EPSS, Runtime
- Контроль соответствия лучшим практикам
- Поиск секретов
- Анализ на вредоносный код и код двойного назначения
- Поддержка внешних и собственных IoC
- Встроенная библиотека уникальных проверок
- Скорость и парализация сканирований
- Security Gates в CI\CD, Registry, Runtime

Details Runtime Info SBOM **Vulnerabilities** Config Secrets Malware

Type: Vulnerability Report

Name: registry.k8s.io-ingress-nginx-controller-42b3f0e5

Updated: 10.04.2025 11:38:30

Registry: registry.k8s.io

Repository: registry.k8s.io/ingress-nginx/controller

Tag: sha256:42b3f0e5d0846876b1791cd3afeb5f1cbb4259d6f35651dcc1b5c980925379c

Total: 105

Fixed: 78

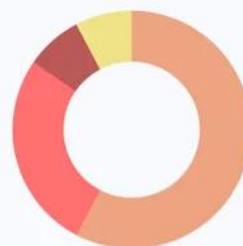
Exploit: 26

RCE: 1

DoS: 0

KEV: 0

[Report Components\(78\)](#) | [Components By Layers](#)



Summary

critical	6
high	21
medium	45
low	6
unknown	0
negligible	0

Export

#	Name	CVEs	Fixable	Severity
1.	curl	14	14	
2.	libcurl	14	14	
3.	k8s.io/ingress-nginx	5	5	
4.	libcrypto3	7	7	
5.	libssl3	7	7	
6.	openssl	7	7	
7.	libxml2	4	4	
8.	musl	1	1	
9.	musl-utils	1	1	
10.	busybox-binsh	4	4	

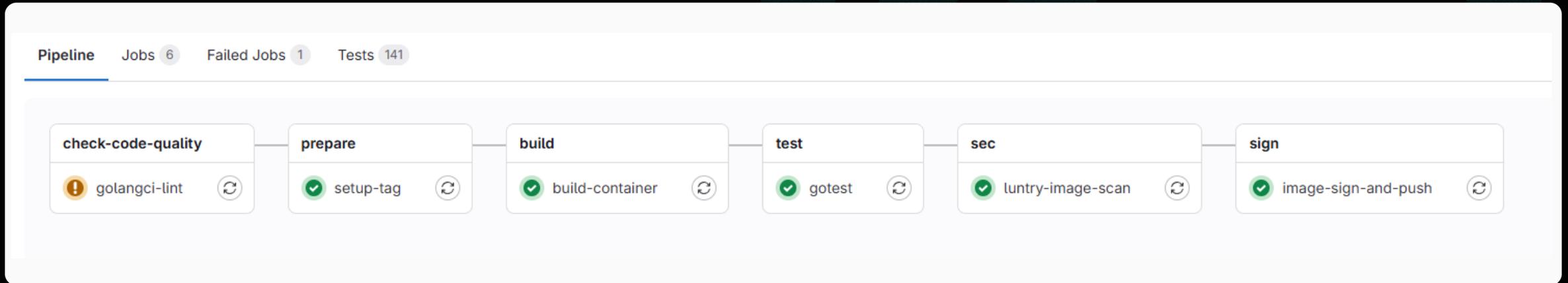
Vulnerability Id	Severity	Type	Exploit	KEV	EPSS	Resource	Installed Version	Fixed Version
▼ CVE-2025-1974	critical	RCE	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0.99242	k8s.io/ingress-nginx	v1.10.0	1.11.5
<p>Description: ingress-nginx admission controller RCE escalation</p> <p>CVSS: <ul style="list-style-type: none"><li>CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H</li></ul></p> <p>Links:</p>								

Items per page: 10 1 - 400 of 1

# Что за сканер внутри Luntry?

- Собственная разработка
  - Для анализа Dockerfile на лучшие практики используется движок YARA
  - Для компонентного анализа используется библиотека Syft
  - Для анализа на известные уязвимости используется библиотека Gypre
  - Для поиска чувствительной информации используется движок YARA
  - Для поиска вредоносного кода и кода двойного назначения используется движок YARA
- Оптимизации
- Умный параллелизм сканирований
- Высокая скорость сканирования
- Работа в закрытом контуре

# Пример встраивания в CI



# Интеграция с CI GitLab

- Нативная интеграция
  - Очень полезно для разработчиков

CVE-2018-6551

X

## Description

The malloc implementation in the GNU C Library (aka glibc or libc6), from version 2.24 to 2.26 on powerpc, and only in version 2.26 on i386, did not properly handle malloc calls with arguments close to SIZE\_MAX and could return a pointer to a heap region that is smaller than requested, eventually leading to heap corruption.

BDU:2018-00364: Уязвимость реализации функции malloc библиотеки, обеспечивающей системные вызовы и основные функции glibc, позволяющая нарушителю вызвать отказ в обслуживании

- Уязвимость реализации функции malloc библиотеки, обеспечивающей системные вызовы и основные функции glibc вызвана выходом операции за границы буфера в памяти. Эксплуатация уязвимости может позволить нарушителю, действующему удалённо, вызвать отказ в обслуживании

Status: Needs triage

Severity: Critical

Project: image-scan-test

Tool: Container Scanning

Scanner: Luntry image scanner

## Location

Image: docker.io/library/debian@sha256:c5c5200ff1e9c73ffbf188b4a67eb1c91531b644856b4aefe86a58d2f0cb05be

Namespace: unknown

## Links

- <https://security-tracker.debian.org/tracker/CVE-2018-6551>

## Identifiers

- CVE-2018-6551

## Evidence

DOS false

KEV false

RCE false

EPSS 0.61087

Type deb

Exploit false

Cancel

Dismiss vulnerability

Create issue

# Поддерживаемые feeds уязвимостей

- Alpine Linux SecDB
- Amazon Linux ALAS
- Chainguard SecDB
- Debian Linux CVE Tracker
- Echo Security Advisories
- GitHub Security Advisories (GHSAs)
- MinimOS SecDB
- National Vulnerability Database (NVD)
- Oracle Linux OVAL
- RedHat Linux Security Data
- RedHat RHSAs
- SUSE Linux OVAL
- Ubuntu Linux Security
- Wolfi SecDB

- БДУ ФСТЭК
- Alt Linux
- РЕД ОС
- Astra Linux
- Sber Linux

# Live Demo

- Security/Quality Gates in Luntry

Security Gates > Gates main

Runtime Policies | Policy Engines | Runtime Rules | Reaction Policies | **Security Gates**

Gates | Violations

Search... 🔍 Add gate +

Name	Severity	Scopes	Vuln	Sbom	Malware	Misconfig	Secret	Description	Block Action	Status		
Extra critical vulns	critical	Runtime	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Runtime vulns monitoring	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
Test Gate	critical	CI/CD Registry	<input checked="" type="checkbox"/>	Blocks images with critical vulnerabilities	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						
Test Gate 2	high	Runtime	<input checked="" type="checkbox"/>	Blocks images with malware	<input type="checkbox"/>	<input checked="" type="checkbox"/>						

# RoadMap Luntry

- Новый UI
- Новые режимы сканирования
- Пользовательские правила на YARA для всего
- Продвинутое приоритизация уязвимостей
- Гибкие фильтры для уязвимостей с учетом пакетов, слоев образа и т.д.
- Поддержка из коробки коммерческих TI feeds
- ...

Мы всегда рады вашей обратной связи и features requests =)



**ВЫВОДЫ**

# Выводы

# LUNTRY

01

Безопасность образов это не только их анализ, но и сборка и хранение

02

Важно анализировать образы по всем 5 доменам безопасности на всем жизненном цикле образа – CI\CD, Registry, Runtime

03

Security/Quality Gates на всем жизненном цикле с разными политиками облегчат работу и контроль

04

Luntry позволяет закрыть и работать со всеми аспектами безопасности образов контейнеров на всех стадиях

# Полезные ссылки

- Все выступления Анатолия Карпенко ;)
  - <https://luntry.ru/research>





# LUNTRY



Telegram [luntry\\_official](#)

Website [luntry.ru](#)

Website [luntrysolution](#)

Email [info@luntry.ru](mailto:info@luntry.ru)

YouTube [luntrysolution](#)

**ДМИТРИЙ ЕВДОКИМОВ**  
Founder & CTO Luntry

Email [de@luntry.ru](mailto:de@luntry.ru)

Telegram [Qu3b3c](#)

Telegram [k8security](#)

# СПАСИБО ЗА ВНИМАНИЕ!