



Уязвимости в контейнерах и **Kubernetes**: правим, а не страдаем



**Дмитрий
Евдокимов**

Основатель
и Технический
директор Luntry

Обо мне

Дмитрий Евдокимов

” 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

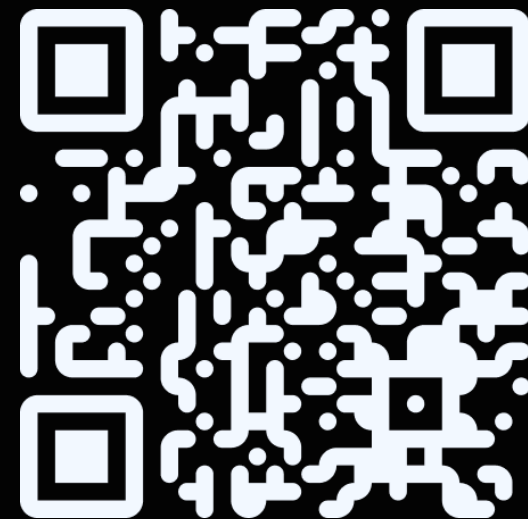
HITB
PHDays
SAS

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 во втором квартале 2026 года.



LUNTRY.RU

Функциональность Luntry

Контроль всех
Kubernetes-ресурсов

Контроль состояния Kubernetes-
кластеров

Контроль соответствия кластера
стандартам



LUNTRY

Управление безопасностью
образов контейнеров

Управление сетевой
безопасностью

Анализ прав доступа

Защита Runtime

Проактивный и реактивный подходы к безопасности



Культура киберустойчивости: «При построении культуры киберустойчивости **роль службы безопасности заключается не в предотвращении всех инцидентов, а в предотвращении негативного влияния инцидента безопасности на бизнес**».

Уязвимости != уязвимый

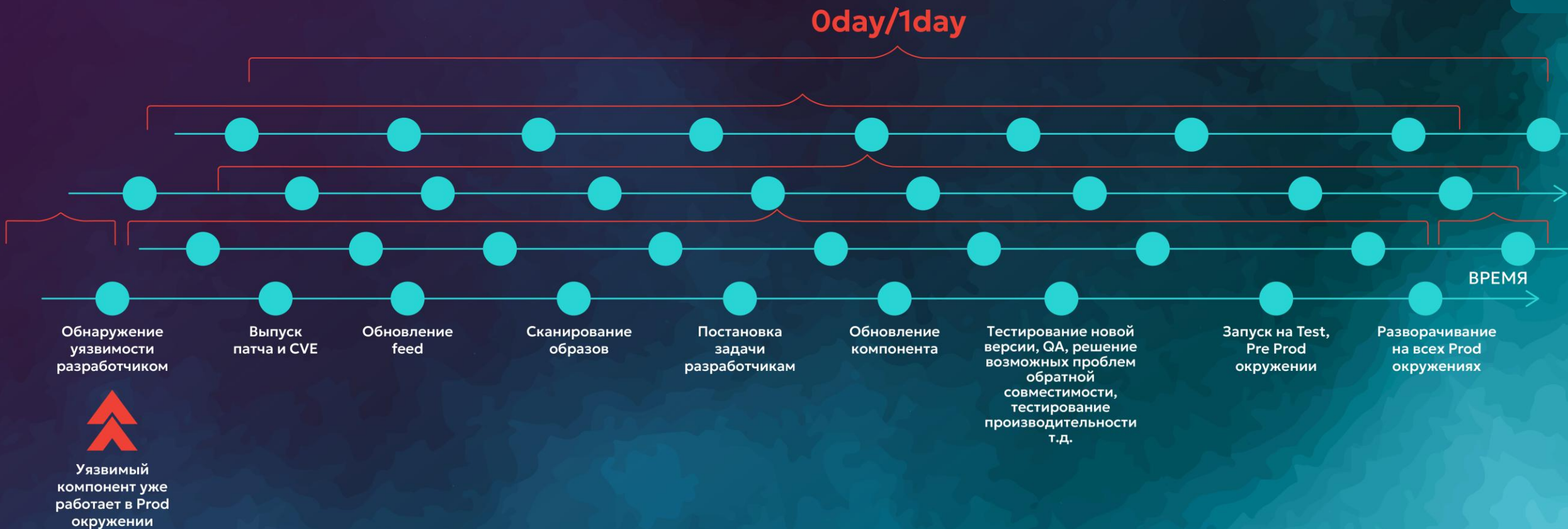


РЕАКТИВНЫЙ ПОДХОД

Работа с 1day уязвимостью



Работа с уязвимостями по факту



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

Shift Left Security



Shift Everywhere Security



Shift Right Security



Shift Down Security

Анализ образов в Luntry

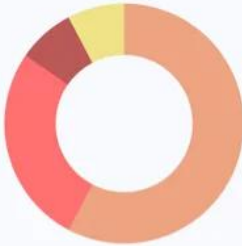
- Встраивание в CI\CD, Registry, Runtime
- Инвентаризация использования образов
- Компонентный анализ (SBOM)
- Сканирование на известные уязвимости
- Поддержка отечественных ОС и БДУ ФСТЭК
- Приоритизация уязвимостей CVSS, Severity, Type(RCE/DoS), Exploit, KEV, EPSS, Runtime
- Контроль соответствия лучшим 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

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

Vulnerability Id	Severity	Type	Exploit	KEV	EPSS	Resource	Installed Version	Fixed Version
CVS-2025-1974	critical	RCE	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0.99242	k8s.io/ingress-nginx	v1.10.0	1.11.5

Description: ingress-nginx admission controller RCE escalation
CVSS: • CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Links:

Items per page: 10 1 - 400 of 1

Приоритезация уязвимостей

CVSS

Severity

Type(RCE/DoS)

Exploit

KEV

EPSS

Runtime

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

Пример 1: В CI

Максимум экономии на “самом дешёвом” этапе:

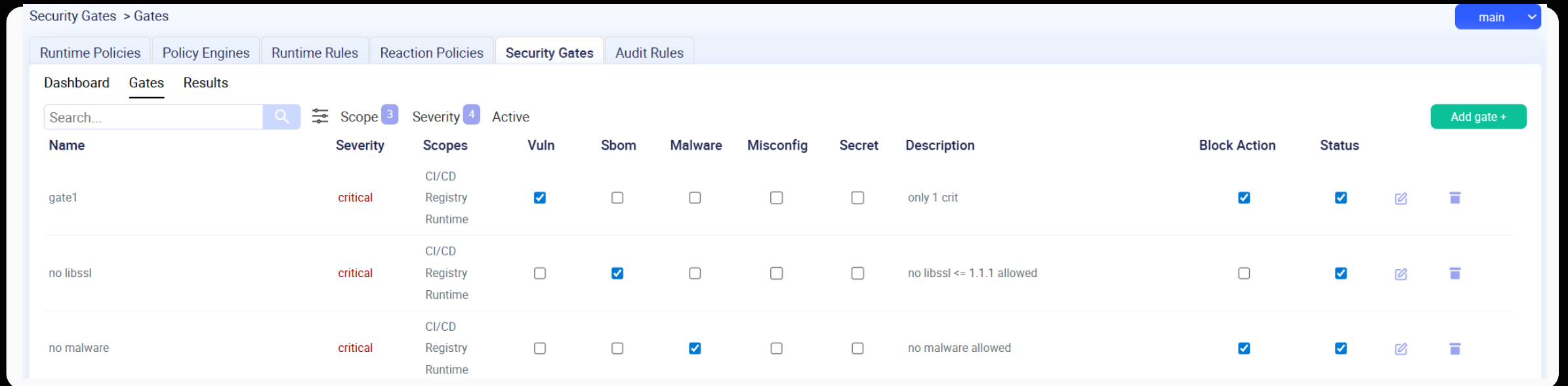
- Раннее обнаружение дефектов

Экономика:

- Исправление проблемы на этапе CI дешевле, чем на последующих этапах

Эффект на команду:

- Снижается нагрузка на QA / DevOps / SRE



The screenshot shows a web interface for "Security Gates". At the top, there are tabs for "Runtime Policies", "Policy Engines", "Runtime Rules", "Reaction Policies", "Security Gates", and "Audit Rules". Below the tabs, there are navigation links for "Dashboard", "Gates", and "Results". A search bar is present with a magnifying glass icon. To the right of the search bar, there are filters for "Scope" (3 items), "Severity" (4 items), and "Active". A green "Add gate +" button is located in the top right corner of the table area.

Name	Severity	Scopes	Vuln	Sbom	Malware	Misconfig	Secret	Description	Block Action	Status		
gate1	critical	CI/CD Registry Runtime	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	only 1 crit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
no libssl	critical	CI/CD Registry Runtime	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	no libssl <= 1.1.1 allowed	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
no malware	critical	CI/CD Registry Runtime	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	no malware allowed	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Пример 2: В Image Registry

Снижение риска и затрат на масштабирование ошибок:

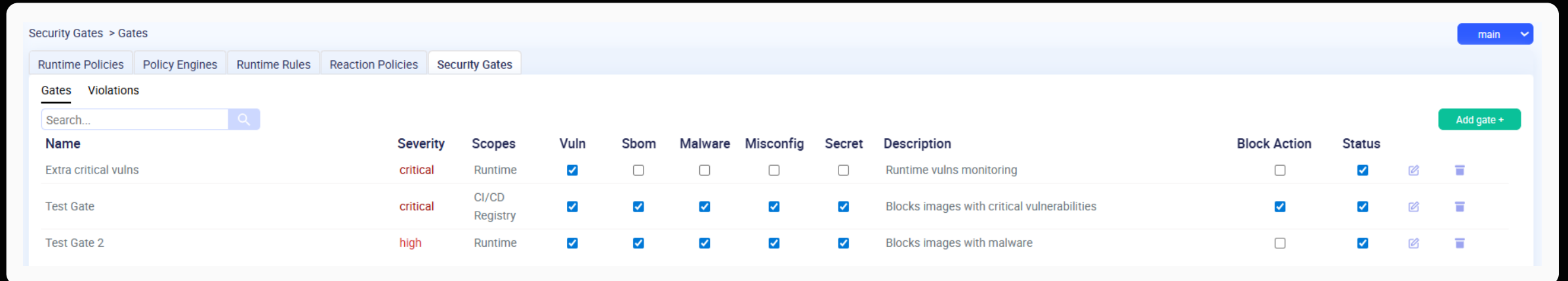
- Единая точка контроля

Экономика:

- Снижает стоимость аудита/соответствия — упрощает проверки и перепроверки, сокращает ручной труд

Эффект на команду:

- Снижение FTE на команду безопасности



The screenshot shows a web interface for 'Security Gates'. At the top, there are navigation tabs: 'Runtime Policies', 'Policy Engines', 'Runtime Rules', 'Reaction Policies', and 'Security Gates'. Below the tabs, there are two sub-tabs: 'Gates' and 'Violations'. A search bar is present. A table lists three security gates with columns for Name, Severity, Scopes, and various vulnerability checks (Vuln, S bom, Malware, Misconfig, Secret). The 'Test Gate' is highlighted in blue.

Name	Severity	Scopes	Vuln	S bom	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"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<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"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Blocks images with malware	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Пример 3: В Runtime

Фокусирование на действительно опасном и важном в текущий момент:

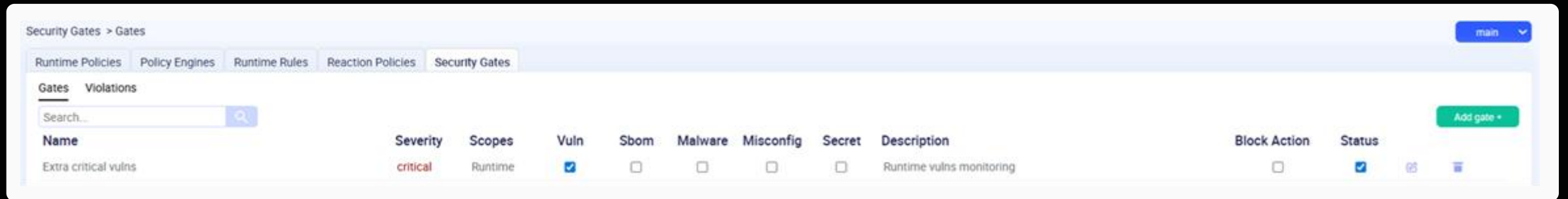
- Контроль самого актуального

Экономика:

- Экономия на инцидентах и простоях

Эффект на команду:

- Простота работы и понижение порога входа за счет приоритезации задач



Польза Security Gates

Defense in depth:

- CI: “дешево исправить”
- Registry: “не дать распространиться”
- Runtime: “не дать навредить (или минимизировать ущерб)”

Ключевые бизнес-метрики, которые обычно улучшаются

- **Lead time / time-to-release**: меньше откатов и “красных” релизов.
- **Change failure rate**: снижается за счёт отсева дефектов и небезопасных изменений.
- **MTTR (время восстановления)**: ниже, потому что меньше инцидентов и выше прозрачность.
- **Стоимость владения (TCO) платформы**: меньше ручных процессов, меньше “пожарных” часов, меньше потерь от downtime.

Но проблема усиливается ...

NIST

Search NIST



Menu

UPDATES

NIST Updates NVD Operations to Address Record CVE Growth

New risk-based model will allow NIST to manage current CVE volume while modernizing the NVD for long-term sustainability.

April 15, 2026

[Обновление политики NIST по NVD](#)



ПРОАКТИВНЫЙ ПОДХОД

Вклад AI

For the first time in bug bounty history, an autonomous penetration tester has reached the top spot on the US leaderboard.



[XBOW](#)

IDENTIFYING VULNERABILITIES AND EXPLOITS WITH CLAUDE MYTHOS PREVIEW

Over the past few weeks, we have used Claude Mythos Preview to identify thousands of zero-day vulnerabilities (that is, flaws that were previously unknown to the software's developers), many of them critical, in every major operating system and every major web browser, along with a range of other important pieces of software.

In a post on our [Frontier Red Team blog](#), we provide technical details for a subset of these vulnerabilities that have already been patched and, in some cases, the ways that Mythos Preview found to exploit them. It was able to identify nearly all of these vulnerabilities—and develop many related exploits—entirely autonomously, without any human steering. The following are three examples:

- Mythos Preview found a 27-year-old vulnerability in OpenBSD—which has a reputation as one of the most security-hardened operating systems in the world and is used to run firewalls and other critical infrastructure. The vulnerability allowed an attacker to remotely crash any machine running the operating system just by connecting to it;
- It also discovered a 16-year-old vulnerability in FFmpeg—which is used by innumerable pieces of software to encode and decode video—in a line of code that automated testing tools had hit five million times without ever catching the problem;
- The model autonomously found and chained together several vulnerabilities in the Linux kernel—the software that runs most of the world's servers—to allow an attacker to escalate from ordinary user access to complete control of the machine.

[Claude Mythos](#)

Реалии и прогнозы

Появление Claude Mythos происходит в контексте уже ускоряющегося временного окна между раскрытием уязвимости и её эксплуатацией.

Окно сокращается с 63 дней в 2018-2019 годах до всего нескольких часов к 2024-2025 годам[1][2]. Claude Mythos угрожает ещё больше сжать это окно.

С выпуском моделей, подобных Mythos, по прогнозам, время от раскрытия уязвимости к полному наступательному потенциалу может сжаться до минут или даже секунд[1][3].

Это представляет **фундаментальный сдвиг в динамике кибербезопасности**, где традиционные циклы тестирования и развёртывания патчей, которые измеряются днями или неделями, становятся неадекватными[1][4][5].

Контейнеры на страже ИБ

Микросервисы (контейнеры)

- Дополнительная изоляция
- Дополнительный слой защиты

Простота

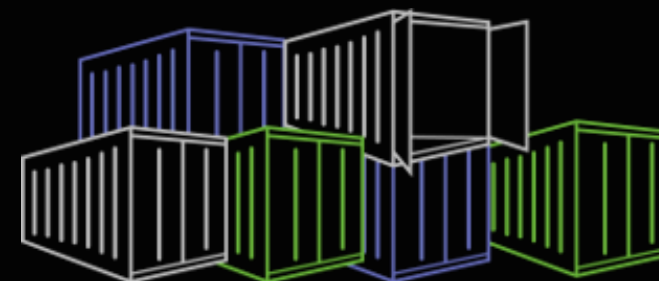
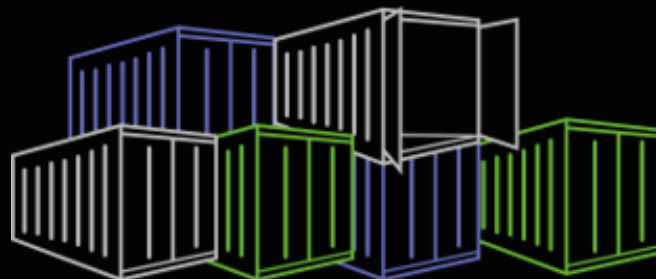
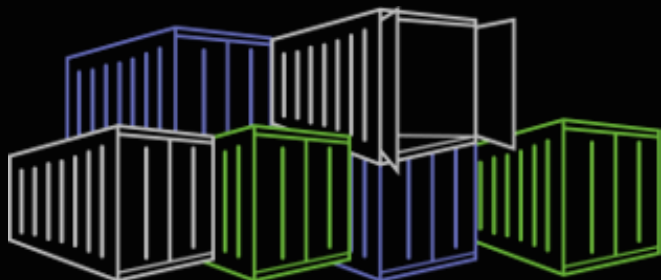
- Микросервис проще ОС и монолитов
 - Иммутабельность образа контейнера

Предсказуемость поведения

- Распределенность
 - Работа микросервисов в несколько копий

Эфемерность

- Высокая скорость модификации без простоя системы



Kubernetes на страже ИБ

KUBERNETES:

- Декларативность
 - Security/Policy-as-Code
 - PolicyEngines
 - Прозрачное взаимодействие с ИТ
- ZeroTrust
 - Whitelisting
 - Micro segmentation
- ShiftLeftSecurity
 - SecDevSecOpsSec



kubernetes

Platform as a Service
(PaaS)

Configuration

Function

Applications

Runtime

Containers

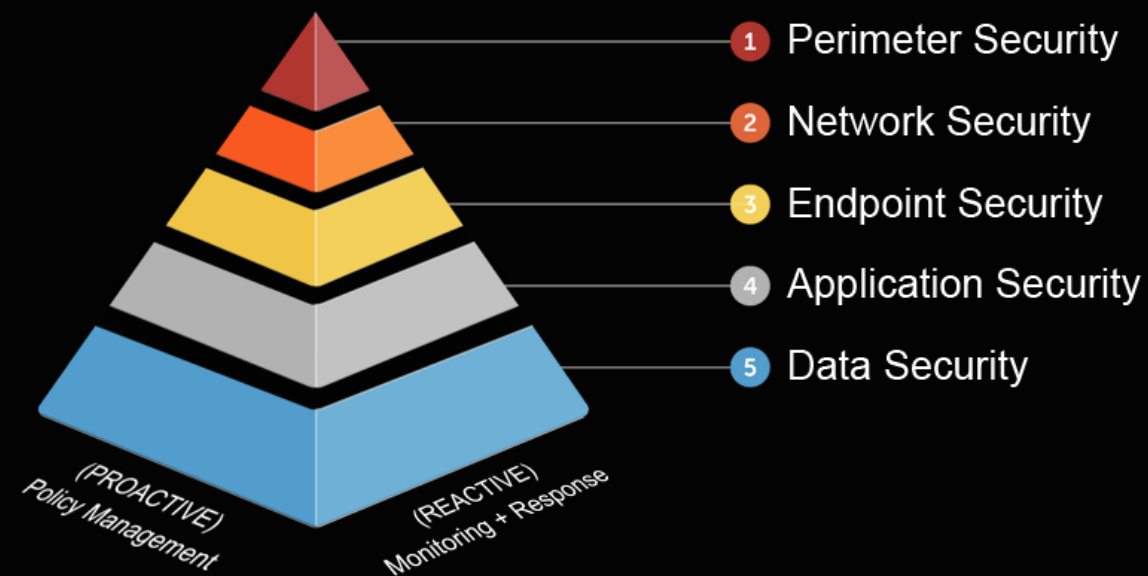
Operation Systems

Hardware

Пример проактивной защиты

МИКРОСЕРВИС:

- На базе минимального образа
 - distroless
- С файловой системой, доступной только на чтение
 - `SecurityContext.readOnlyRootFilesystem.true`
- Без лишних возможностей
 - `SecurityContext.capabilities.drop.all`
- Со строго определённым набором исполняемых файлов
 - AppArmor profile
- С ограниченной активностью по сети
 - NetworkPolicy



Luntry для проактивного подхода

Контроль hardening
механизмов для
микросервисов

Автоматическое создание
сетевых политик
NetworkPolicy

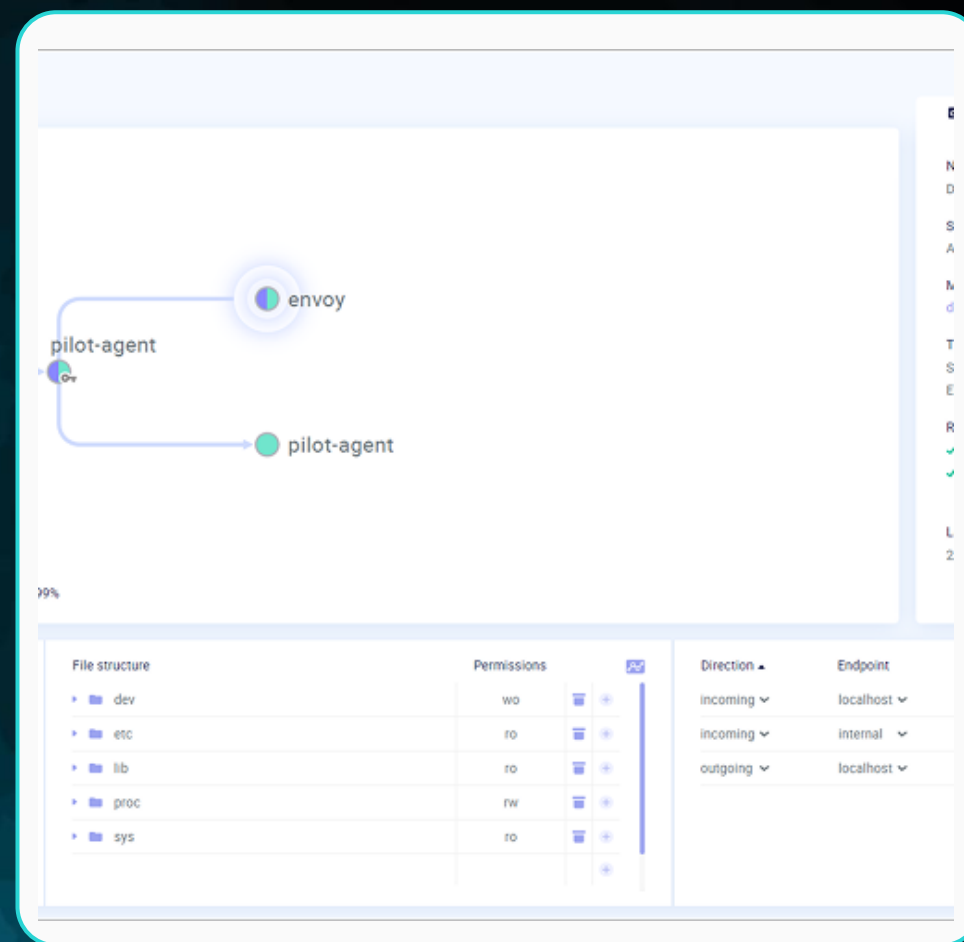
Автоматическое создание
профилей поведения

Генерация
prevention policy

Luntry для проактивного подхода

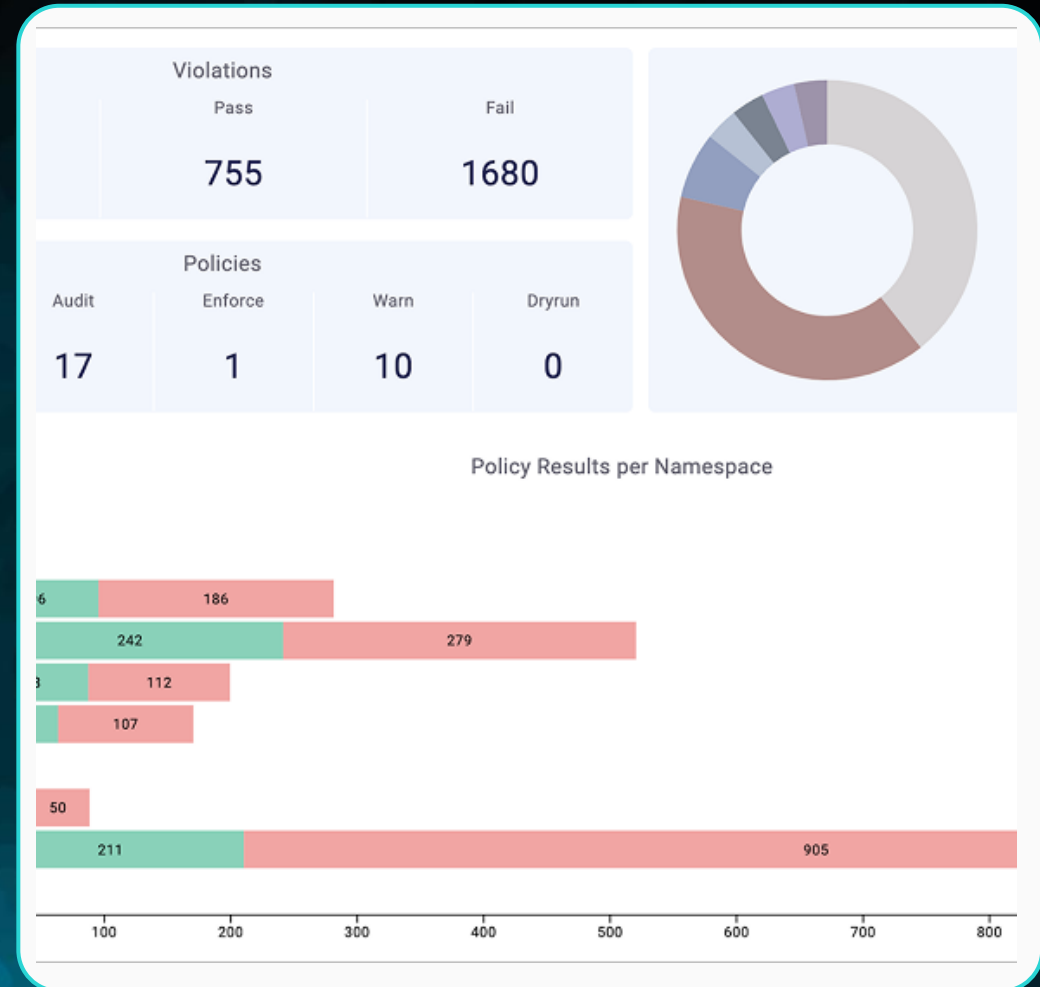
Генерация
prevention policy

Автоматическое создание
профилей поведения



Luntry для проактивного подхода

Контроль hardening
механизмов для
микросервисов



Выводы

LUNTRY

01

Разный уровень зрелости, требований, критичности позволяют использовать разные подходы

02

Ситуация с уязвимостями с каждым днем становится все хуже

03

Специфика контейнеров и Kubernetes позволяет более прозрачно смотреть на вопрос управления уязвимостями

04

Luntry позволяет построить комплексный подход с использованием реактивных и проактивных практик



LUNTRY



Telegram [luntry_official](#)

Website [luntry.ru](#)

Website [luntrysolution](#)

Email info@luntry.ru

YouTube [luntrysolution](#)

ДМИТРИЙ ЕВДОКИМОВ

Founder & CTO Luntry

Email de@luntry.ru

Telegram [Qu3b3c](#)

Telegram [k8security](#)

СПАСИБО ЗА ВНИМАНИЕ!