LUNTRY | Deckhouse ФЛАНТ

«Платформа контейнеризации под защитой: объединяя знания Deckhouse и Luntry»

Дмитрий Евдокимов
Founder&CTO Luntry

Алексей Крылов
Менеджер продукта Deckhouse Kubernetes Platform
по направлению «Информационная безопасность» (АО «Флант»)

АЛЕКСЕЙ КРЫЛОВ

> Как без фундамента не построить надежное здание, так и без безопасности не создать современную технологическую платформу

Менеджер продукта Deckhouse Kubernetes Platform по направлению «Информационная безопасность», АО «Флант»

Более 25 лет опыта в ИТ

Эксперт в области безопасной эксплуатации

- 15 лет опыта в разработке и эксплуатации промышленных систем в ФинТех секторе
- 6+ лет опыта построения конвейера SSDLC

Спикер

AnsibleConf IT Elements DevOpsConf PerformanceConf

ФЛАНТ

О компании «Флант»

ФЛАНТ

Сильная команда

Более 450 человек, из них больше половины — инженеры

Проверенный партнёр

Более 260 компаний доверяют нам обслуживание своих production-окружений

Лицензии ФСТЭК России и ФСБ России

на деятельность по технической защите КИ,
на разработку и производство средств защиты КИ
и на разработку криптографических средств

Собственная технологическая конференция

Более 500 человек посетили Deckhouse Conf в 2025 году

О компании «Флант»

ФЛАНТ

DevOps-эксперты

Российский вендор ПО и сервисная компания по построению DevOps-решений под ключ, лидер российского рынка DevOps и Kubernetes

На рынке с 2008 года

Более 17 лет опыта в Open Source, Linux, DevOps.
Работаем с Kubernetes с момента его публичного релиза.

Лидеры по контрибуциям в Open Source

Лидеры среди контрибьюторов из России в проекты CNCF

В топе вендоров ИТ-решений для банков и промышленности

Согласно рейтингам TAdviser (2024 год):
«Крупнейшие ИТ-вендоры в банках»
«Крупнейшие ИТ-вендоры в промышленности»

Deckhouse Kubernetes Platform в цифрах

8 лет

эксплуатации.
Первая российская K8s-платформа

260 +

клиентов доверяют нам
развитие своего бизнеса

200 +

сертифицированных
партнёров

1000 +

кластеров под управлением
Deckhouse

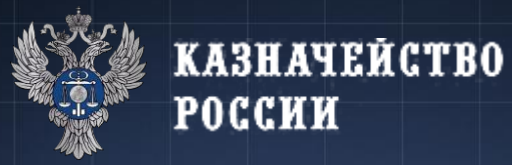
99,97 %

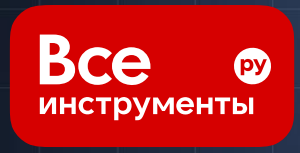
фактический SLA по кластерам
под нашим управлением

490 +

инженеров из 126 компаний
прошли обучение

Нам доверяют клиенты





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	Confidence	ZeroNights	БеКон	HITB
DevOpsConf	HackInParis	KuberConf	BlackHat	PHDays
Kazhackstan	HighLoad++	OFFZONE	DevOops	SAS

LUNTRY

Основной функционал Luntry

Luntry – это Комплексная Защита на всем жизненном цикле контейнерных приложений и средств оркестрации на базе Kubernetes

Контроль Kubernetes-ресурсов

Контроль состояния Kubernetes-кластеров

Контроль соответствия кластера стандартам

Управление безопасностью образов

Сетевая безопасность

Анализ прав доступа

Защита Runtime

План вебинара

01 Введение в безопасность Kubernetes

02 Особенности Deckhouse и Luntry

03 Заключение

LUNTRY | Deckhouse ФЛАНТ

01 ВВЕДЕНИЕ В БЕЗОПАСНОСТЬ KUBERNETES

LUNTRY | Deckhouse ФЛАНТ

Зачем и почему?

ТРЕБОВАНИЯ РЕГУЛЯТОРОВ (COMPLIANCE)

- PCI DSS
- ФЗ 152
- Приказы ФСТЭК России №118, №187
- Указ Президента РФ №166

ТЕХНИЧЕСКИЕ МЕРЫ (SECURITY)

- Атакующий смотрит по-своему, а не требования регулятора ;)
- Необходимо увеличение стоимости атаки на систему

РИСКИ И ОТВЕТСТВЕННОСТЬ

- Репутационные
- Финансовые
- Административные
- Уголовные

ВИДЫ НАРУШИТЕЛЕЙ

- Внешний атакующий (может об этом и не знать)
- Внутренний атакующий (не обязательно человек)
- Скомпрометированный разработчик
- Скомпрометированная цепочка поставок (зависимости)
- Привилегированный пользователь (администратор, DevOps, etc)

Что?

МНОЖЕСТВО ПОДХОДОВ:

- 4C Model
- Zero Trust
- CIS Kubernetes benchmark
- MITRE ATC&ACK Matrix
- OWASP Kubernetes Top Ten

ОСНОВНЫЕ ВЕКТОРА ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ K8S:

- Управление доступами и ресурсами
- Управление сертификатами и хранение секретов
- Безопасность оркестратора/платформы
- Безопасность образов контейнеров
- Безопасность Runtime
- Сетевая безопасность
- Аудит безопасности кластера

Когда?

Чем раньше, тем лучше.

Правила игры в начале игры,
а не переезд в последствии.

К «хорошему» быстро привыкаешь
и потом это ломать тяжело.

Как?

БЕЗОПАСНОСТЬ - ЭТО КОМПЛЕКСНАЯ И ПОСТОЯННАЯ ЗАДАЧА.

Требуется использование безопасной, надежной платформы
и навесного средства безопасности.

ОБЫЧНО ВЫДЕЛЯЮТ 3 УРОВНЯ:

- ОС
- Платформа
- Навесное средство безопасности

02 ОСОБЕННОСТИ DECKHOUSE И LUNTRY

LUNTRY | Deckhouse ФЛАНТ

2.1 DECKHOUSE:
УПРАВЛЕНИЕ ИДЕНТИФИКАЦИЕЙ И ДОСТУПОМ

Deckhouse
ФЛАНТ

Проблематика

ЛОКАЛЬНЫЕ ПОЛЬЗОВАТЕЛИ В КЛАСТЕРАХ K8s

- Неудобно управлять пользователями и их полномочиями
- Нет парольных политик
- Риски что-то забыть
- Риски дать избыточные права
- Ответственность на стороне администратора k8s

РОЛИ В K8S:

- Избыточные полномочия (Админы k8s могут управлять приложениями)
- Сложность в настройке и аудите

МУЛЬТИАРЕНДНОСТЬ:

- Namespace – не панацея, нет изоляции по ресурсам и не ограничивается сетевое взаимодействие
- Проблема «шумного соседа»
- По умолчанию не включены сборка логов, аудит и сканирование на уязвимости

Решение

DEX: OIDC-ПРОВАЙДЕР АУТЕНТИФИКАЦИИ

- Единые учетные данные для нескольких кластеров DKP и одновременная работа с несколькими провайдерами
- Поддержка внешних провайдеров и протоколов:
- Политика безопасности паролей – на стороне внешнего провайдера

РАСШИРЕНИЕ ФУНКЦИОНАЛА ЛОКАЛЬНЫХ ПОЛЬЗОВАТЕЛЕЙ НА СТОРОНЕ ПЛАТФОРМЫ:

- Парольная политика (сложность, повторяемость, срок жизни, блокировка при некорректном вводе)
- Встроенная поддержка двухфакторной аутентификации (2FA)

RBAC 2.0

- Use-роли : Viewer, User, Manager, Admin – для пользователей в проекте/конкретном пространстве имён
- Manage-роли : Viewer, Manager – для назначения прав администраторам платформы или подсистемы(модуля)

ПРОЕКТЫ И ШАБЛОНЫ ПРОЕКТОВ

- Безопасность – изоляция ресурсов и политики доступа между проектами
- Ограничения ресурсов – квоты на ресурсы для каждого проекта
- Преднастроенные шаблоны (default, secure, secure-with-dedicated-nodes)
- Шаблоны - Единообразие создания проектов и упрощение управления проектами

Вывод

DKP ПОДДЕРЖИВАЕТ ДВА ПОДХОДА К АУТЕНТИФИКАЦИИ

- Интеграция с внешним провайдером
- Локальная аутентификация

При этом интерфейс аутентификации для пользователя и способы включения аутентификации для приложения будут одинаковые.

RBAC 2.0 – РАЗДЕЛЕНИЕ ЭКСПЛУАТАЦИИ ПЛАТФОРМЫ И ПРИЛОЖЕНИЙ

- Use-роли – для продуктовых команд
- Manage-роли – для администраторов платформы или подсистем(модулей)

ПРОЕКТЫ И ШАБЛОНЫ ПРОЕКТОВ

- Для администраторов платформы: единообразие, безопасность, управление потреблением ресурсов
- Для пользователей платформы: быстрый старт и изолированные окружения

2.2 LUNTRY:

УПРАВЛЕНИЕ БЕЗОПАСНОСТЬЮ ОБРАЗОВ

Проблематика

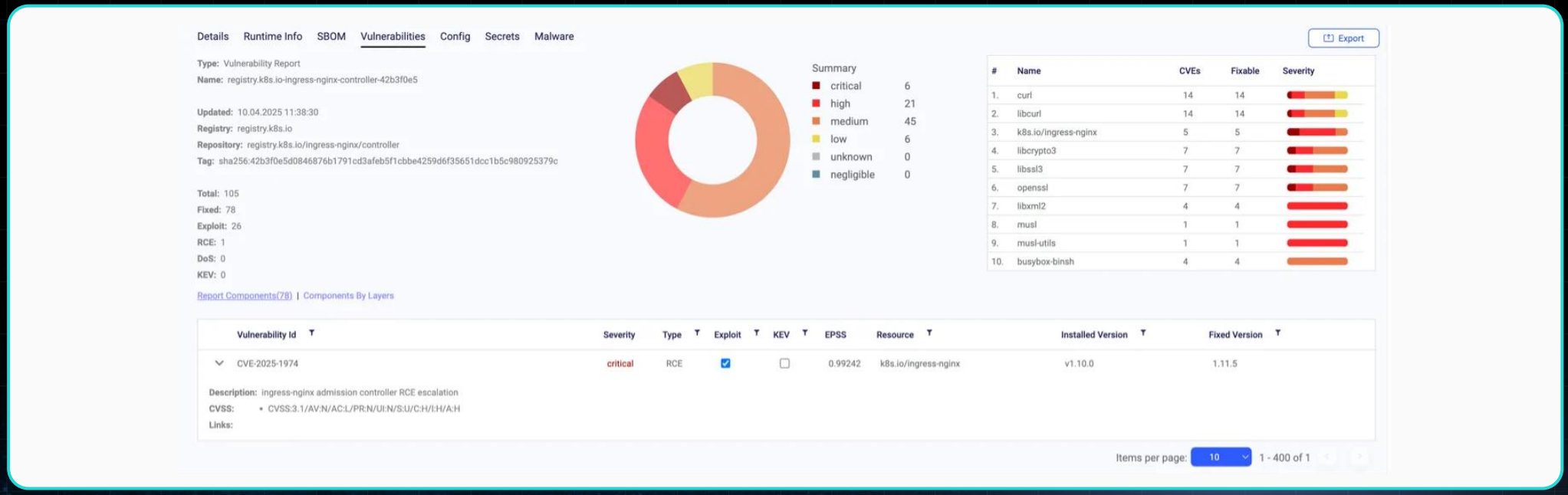
Код / приложение / микросервис – основное действующее лицо в современной компании, так как выполняет бизнес логику компании.

- Это действующее лицо не идеально и содержит разные недостатки и проблемы.
- Эти недостатки и проблемы может использовать внутренние и внешние злоумышленники для нанесения ущерба как конкретному приложению, так и всей системе и всем командам в этой системе.
- Таких недостатков и проблем очень много и они постоянно появляются новые и эволюционируют, как и подходы атакующих.
- Необходимо их своевременно обнаруживать, приоритизировать, закрывать и предотвращать атаки.
- Держать такую экспертизу в каждой компании очень тяжело.

Решение

- Инвентаризация использования
- Компонентный анализ SBOM
- Сканирование на уязвимости
- Приоритизация уязвимостей
- Поддержка отечественных ОС и БДУ ФСТЭК
- Интеграция с ASOC
- Поиск секретов

- Контроль соответствия лучшим практикам
- Анализ на вредоносный код и код двойного назначения
- Поддержка внешних и собственных IoC
- Встроенная библиотека уникальных проверок
- Встраивание в CI\CD, Registry, Runtime
- Security Gates в CI\CD, Registry, Runtime
- Высокая скорость и параллелизация сканирований



LUNTRY | Deckhouse ФЛАНТ

Вывод

УНИКАЛЬНАЯ ЧЕРТА

- Поиск кода двойного назначения
- Поддержка внешних IoC
- Собственная уникальная большая база проверок с приоритетами
- Приоритизация уязвимостей
- Security Gates в Image Registry и Runtime

ЧТО ЭТО ДАЕТ

- Высокую гибкость и точность
- Правильно планировать работы по безопасности образов
- Возможность уменьшить время на работу в пустую

ЧТО БЕЗ ЭТОГО

- Гигантские отчеты без движения вперед

2.2 DECKHOUSE:
УПРАВЛЕНИЕ СЕКРЕТАМИ И СЕРТИФИКАТАМИ

Deckhouse
ФЛАНТ

Проблематика

СЕРТИФИКАТЫ

Сертификаты используются для подтверждения:

- защищённости сетевых подключений (TLS)
- подлинности приложения
- подлинности пользователей (владельцев открытого ключа)
- Взаимной аутентификации (mTLS)

Сертификатами надо управлять: выпускать, хранить, следить за сроком действия, ротировать, обновлять на стороне сервисов.

Какие Certificate Authority использовать?
Где и как хранить, как управлять?

СЕКРЕТЫ

Ресурсы типа Secret в k8s:

- предназначены для хранения и передачи конфиденциальной информации
- Данные в ресурсе Secret не шифруются, а кодируются, поэтому защищены недостаточно надёжно

Содержимое etcd – по умолчанию не шифруется.

Как хранить так, чтобы потом не было мучительно больно?

Решение

ЧАСТЬ ПЕРВАЯ : УПРАВЛЕНИЕ СЕРТИФИКАТАМИ

Встроенный модуль cert-manager поддерживает:

- заказ сертификатов во всех поддерживаемых источниках, таких как Let’s Encrypt, HashiCorp Vault, Venafi, Deckhouse Stronghold;
- интеграции с DNS провайдерами (Yandex Cloud DNS, Cloudflare, AWS Route53, Google Cloud DNS);
- подтверждение владения доменом через ACME HTTP-01 и ACME DNS0-01 Challenge Provider;
- установку cm-acme-http-solver на мастер-узлы и выделенные узлы;
- добавление сертификатов в ingress-ресурсы;
- выпуск самоподписанных сертификатов;
- автоматический перевыпуск сертификатов;
- мониторинг срока действия и статуса перевыпуска сертификатов;
- готовые роли для управления ресурсами Certificate и Issuer в namespace и кластере

Решение

ЧАСТЬ ВТОРАЯ : УПРАВЛЕНИЕ СЕКРЕТАМИ

Где и как хранить секреты:

- Kubernetes Secrets + шифрование etcd
- External Secrets Operator/CSI Secret Store Driver
- HashiCorp Vault (с инжектором или CSI-драйвером)
- Sealed Secrets / Helm Secrets (Mozilla sops).

ПУТЬ DECKHOUSE

Модуль Deckhouse Stronghold:

- Централизованное хранение секретов
- Контроль доступа
- Шифрование данных
- Журналирование и мониторинг
- Интеграция с другими инструментами

Модуль secrets-store-integration:

- Пользовательское приложение само обращается в хранилище.
- В хранилище обращается приложение-прослойка, а ваше приложение получает доступ к секретам из файлов, созданных в контейнере.
- В хранилище обращается приложение-прослойка, и пользовательское приложение получает доступ к секретам из переменных среды.

Вывод

K8s secret – не такой уж и секрет. Base64 – это не шифрование.
Хранить секреты надо надежно.

Сертификаты – один из ключевых объектов при построении безопасной платформы и при эксплуатации решений на ее основе

Модуль cert-manager в составе DKP:

- интегрируется с различными удостоверяющими центрами и провайдерами
- управляет ЖЦ сертификатов,
- отслеживает сроки действия и статус перевыпуска сертификатов,
- поддерживает инъекцию сертификатов ingress-контроллеры
- имеет готовые роли для управления сертификатами
- Интегрируется с Deckhouse Stronghold

Хранить и передавать в приложения секреты необходимо надежно и безопасно.

Deckhouse Stronghold

- Управляет секретами, их хранением и доступом к ним
- Предоставляет API для интеграции
- Доступен в виде модуля платформы
- Доступен в виде отдельного продукта

2.4 LUNTRY:
КОНТРОЛЬ KUBERNETES-РЕСУРСОВ

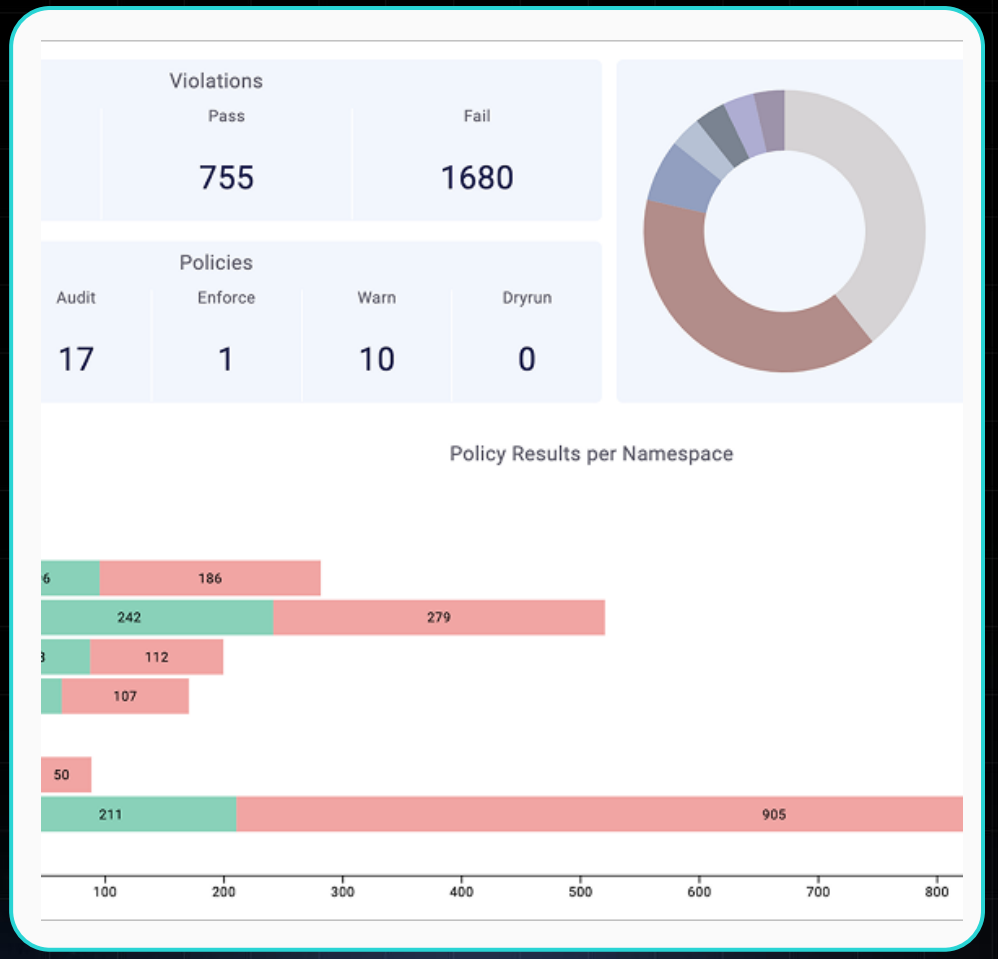
Проблематика

YAML-ресурсы Kubernetes – основное действующее лицо в кластере Kubernetes, так как система декларативная и любое действие / изменение / активность /… описывается там.

- Это действующее лицо не идеально и содержит разные недостатки и проблемы.
- Эти недостатки и проблемы может использовать внутренние и внешние злоумышленники для нанесения ущерба как конкретному приложению, так и всей системе и всем командам в этой системе.
- Таких недостатков и проблем очень много и они постоянно появляются новые и эволюционируют, как и подходы атакующих.
- Необходимо их своевременно обнаруживать, приоритизировать, закрывать и предотвращать.
- Держать такую экспертизу в каждой компании очень тяжело.

Решение

- Контроль изменений любых ресурсов
- Анализ Kubernetes Audit Log
- Проверка любых ресурсов на лучшие практики и внутренние стандарты
- Интеграция с Policy Engine: Kyverno, OPA Gatekeeper
- Встроенная библиотека политик
- Создание собственных политик и правил



Вывод

УНИКАЛЬНАЯ ЧЕРТА

- Интеграция с Kyverno и OPA Gatekeeper
- Встроенная база политик
- Возможность создавать собственные политики
- Анализ событий Kubernetes Audit Log
- Встроенная библиотека правил детектирования
- Возможность создавать собственные правила детектирования

ЧТО ЭТО ДАЕТ

- Логика политик ограничена только вашей фантазией
- Прозрачное взаимодействие с ИТ
- Контроль любого действия с Kubernetes YAML ресурсом

ЧТО БЕЗ ЭТОГО

Отсутствие возможности полностью контролировать самый распространённый путь атаки на Kubernetes

LUNTRY | Deckhouse ФЛАНТ

2.5 DECKHOUSE:
КОНТРОЛЬ ЦЕЛОСТНОСТИ ПЛАТФОРМЫ

Deckhouse

ФЛАНТ

Проблематика

Модули платформы, их конфигурации, и компоненты уровня ОС имеют разные форматы, места хранения и способы поставки

НЕОБХОДИМО:

- Обеспечить контроль целостности платформы
- Обеспечить безопасность и работоспособность при попытках нарушения целостности

ЦЕЛОСТНОСТЬ ЧЕГО В KUBERNETES НУЖНО КОНТРОЛИРОВАТЬ

КОНТЕЙНЕРЫ

- Control plane
- Аддоны
- Полезная нагрузка

КОНФИГУРАЦИЯ

- Helm Chart
- GitOps
- Manifests
- Что-то еще

КОМПОНЕНТЫ ОС

- containerd
- kubelet
- Агенты
- Системные службы

Контроль целостности трёх К

Решение - Подготовка

ОДНОРОДНОСТЬ ПОСТАВКИ КОНТЕЙНЕРОВ, КОНФИГУРАЦИИ И КОМПОНЕНТОВ ОС

- Единый формат поставки (OCI Image Format)
- Единый контролируемый репозиторий образов
- Цифровая подпись для каждого артефакта
  - Подпись фиксирует, кто и что создал на каждом этапе
  - Это не защита сама по себе, а основа для проверки доверия
  - Open Source-решения позволяют построить собственный доверенный контур

УРОВНИ КАЧЕСТВА КОНТРОЛЯ ЦЕЛОСТНОСТИ

I – Неизменяемость: Контролируешь так, что нельзя изменить

II – Компенсация: Автоматически применяешь компенсирующие меры

III - Сигнал тревоги: Кричишь во все стороны, что происходит беда

Часть 1 – Контейнеры

РЕАЛИЗАЦИЯ DECKHOUSE:

1. За основу взяли containerd 2.0 (свежий релиз на момент начала работ)
2. Интеграция с dm-verity в delivery-kit на этапе сборки
3. Итоговая реализация

- Для гарантии иммутабельности в контейнере используется read-only файловая система EROFS.
- Delivery-kit на этапе сборки считает хеш для каждого EROFS-слоя. При скачивании контейнера проверяется его подпись встроенным внутрь сертификатом.
- При старте контейнера все снапшоты проверяются с помощью dm-verity — их нельзя поменять даже на хосте, а если и получится — контейнер не увидит этого до рестарта.
- При старте контейнера, при нарушении целостности, образ целиком перекачивается из реестра контейнеров.
- OverlayFS не используется — нельзя ничего добавить даже в запущенном контейнере.
- Периодическая проверка смотрит только на наличие dm-verity для всех слоёв EROFS.

Получился честный I уровень качества контроля целостности.

Часть 2 – Конфигурация

ПРОБЛЕМА ETCD

Доступ к etcd = режим бога в кластере

Злоумышленники или неправильно настроенная система:

- могут изменить секреты
- могут подменить конфигурацию
- могут скрыть вредоносную нагрузку.

ОСНОВНОЙ ВОПРОС

Можем ли быть уверены, что всё, что записано в etcd, записано через kube-apiserver и никак не изменено даже людьми с прямым доступом к etcd?

НАШЕ РЕШЕНИЕ

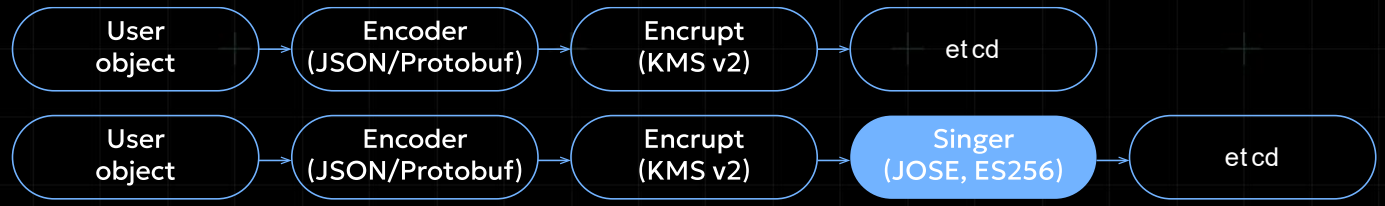
Цифровые подписи

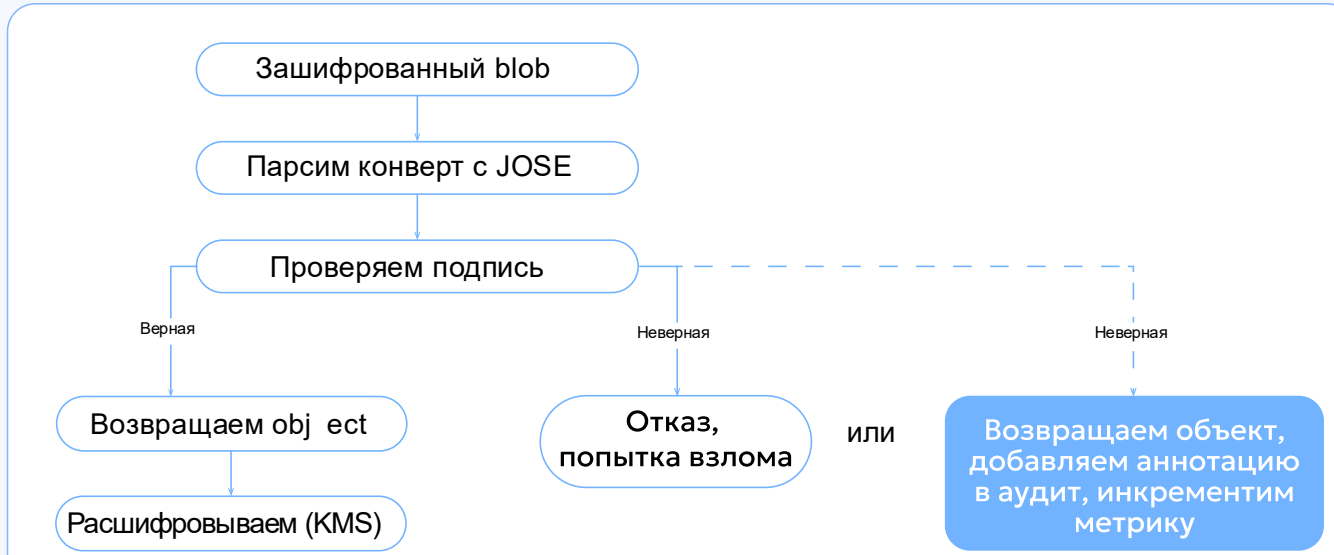
ПОЧЕМУ НЕЛЬЗЯ ПРОСТО ЗАШИФРОВАТЬ БАЗУ?

Свойство	Шифрование at rest	Цифровая подпись
Предотвращает утечку	Да	Нет
Обнаруживает подмену	Нет	Да
Данные можно восстановить при потере ключа	Нет	Да (данные всё ещё здесь)

Часть 2 – Конфигурация

Добавление цифровой подписи к объектам в etcd





JOSE = JSON Object Signing and Encryption
Популярный стандарт.
Совместим с HSMs, Vault или просто ключами в файлах.
ES256 = ECDSA с P-256 — широко распространён.

ЧТО МЫ ПОЛУЧИЛИ В ИТОГЕ

- Надежный путь записи
- Обнаружение подмены
- Аудируемая запись
- Возможна ротация ключей
- Совместимо со слоем шифрования (KMS)

Часть 3 – Компоненты ОС

Воспользуемся хорошо известной всем техникой: положим подпись в ELF-заголовки

ДЛЯ КАЖДОГО БИНАРНОГО ФАЙЛА ИЗ ПОСТАВКИ НА ПЕРВОЙ ИТЕРАЦИИ ВЫПОЛНЯЮТСЯ:

- расчет контрольной суммы;
- подпись контрольной суммы в Stronghold;
- запись метаданных в ELF-заголовок файла.

ВТОРОЙ ИТЕРАЦИЕЙ ПРОИЗВОДИТСЯ ПОДПИСАНИЕ МАНИФЕСТА OCI-ОБРАЗА, В КОТОРЫЙ УПАКОВАН ПОДПИСАННЫЙ БИНАРНИК.

Далее образ отправляется в доверенный реестр.

Для проверки подписей компонентов ОС
I уровень качества контроля целостности
достижим только средствами ОС

Для II уровня качества контроля целостности
реализован d8-bcheck
Вызывает повторный деплой компонентов
при попытке что-то в них изменить

Решение – Компоненты ОС

ВЫВОДЫ ПРО КОМПОНЕНТЫ ОС

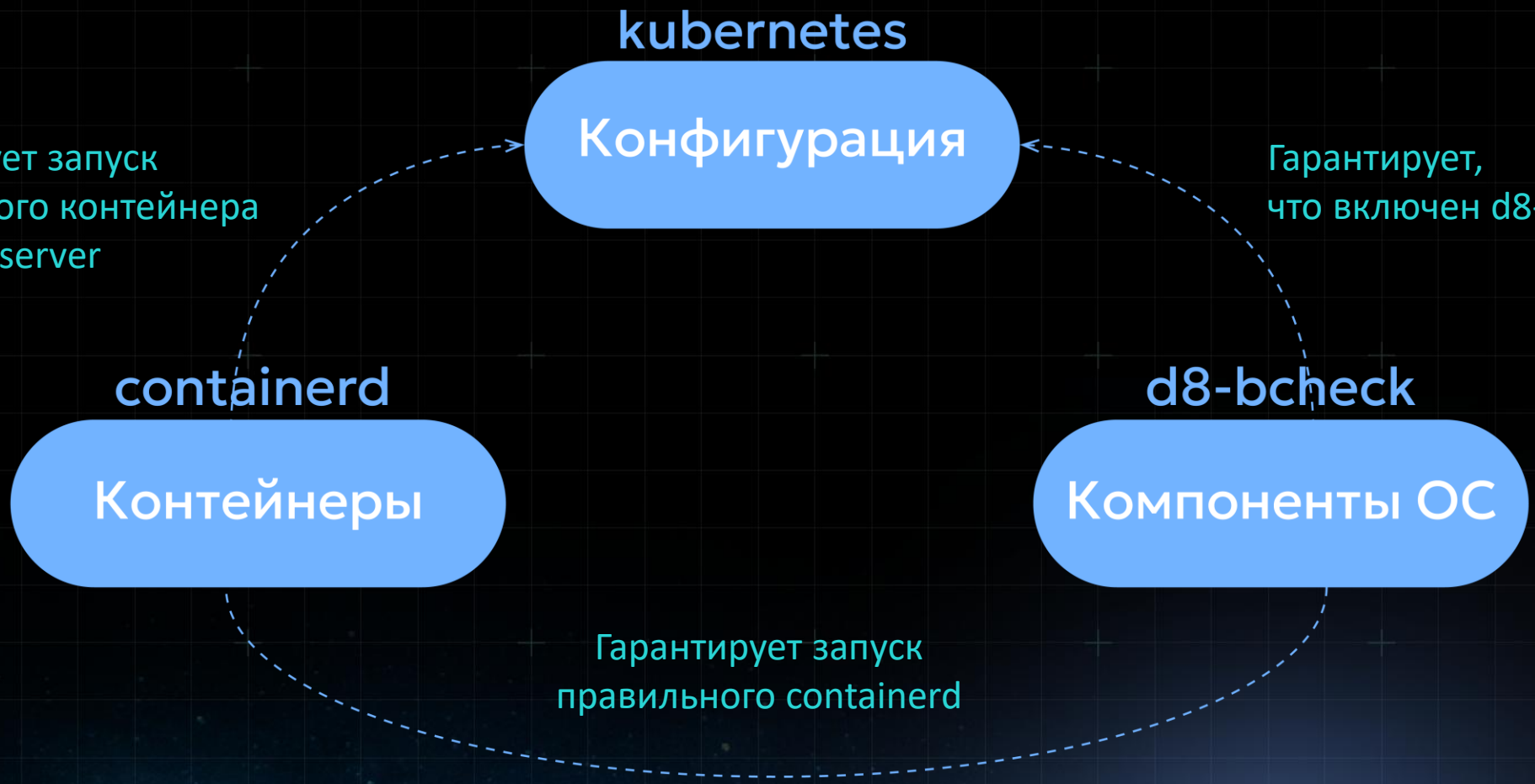
Подпись превращает бинарный файл в контролируемый объект.

Удобно иметь единый формат подписи для компонентов ОС и контейнеров.

Компоненты ОС доставляются в контейнерах, поэтому дополнительно проверяется ещё и подпись контейнера.

Решение

ЦЕЛОСТНОСТЬ КОНТРОЛЯ ЦЕЛОСТНОСТИ



Вывод

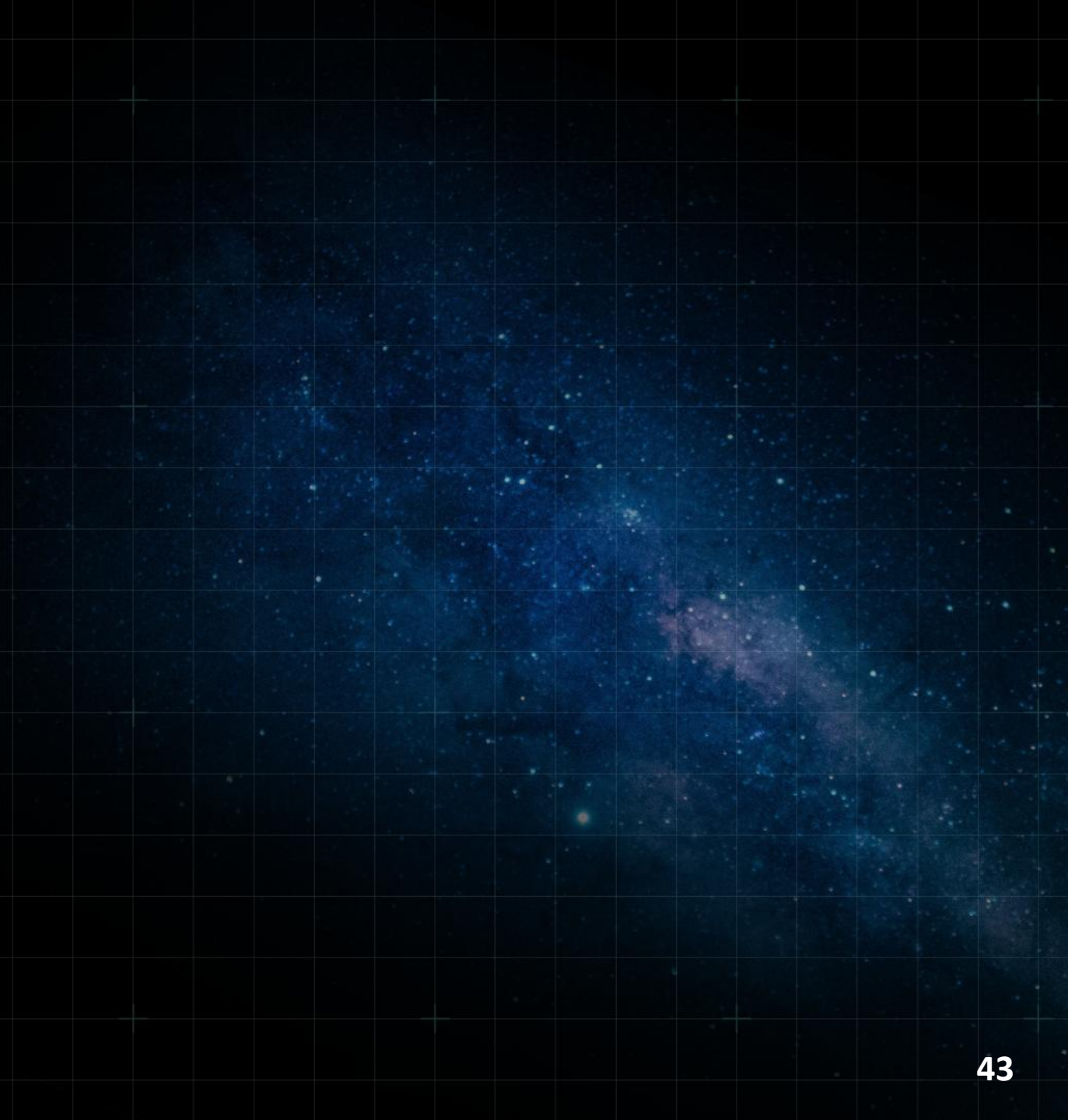
Контроль целостности — не опция, а основа доверия

В Kubernetes целостность — это системная задача, которая обеспечивается на уровне трёх К

Целостность должна быть сквозной

Нельзя забывать про уровни качества контроля

2.6 LUNTRY:
ЗАЩИТА RUNTIME



LUNTRY

Проблематика

Runtime или среда выполнения – это там, где код / приложение / микросервис реально запущены, выполняются и работают с данными.

ИМЕННО ЗДЕСЬ ПРИЛОЖЕНИЕ:

- доступно для внешнего злоумышленника
- работает с реальными / критичными / персональными /… данными
- подвергается большему количеству угроз и сценариев атак

Успешная атака может нанести ущерб как конкретному приложению, так и всей системе и всем командам в этой системе или компании в целом.

Своевременно закрыть все недостатки и проблемы невозможно, тем более в случае использования атакующим новых и неизвестных векторов (0day), что неминуемо делает невозможным создание идеальных систем.

Необходимо своевременно митигировать, обнаруживать, реагировать и предотвращать атаки.
Держать такую экспертизу в каждой компании очень тяжело.

Решение

- Построение модели поведения микросервисов
- Гибридное обнаружение атак по аномалиям и правилам
- Встроенная библиотека правил / детектов
- File Integrity Monitoring (FIM) для процессов
- Генерирование политик предотвращения
- Реагирование на инциденты
- Сбор данных для расследования инцидентов
- Интеграция с SIEM и другими внешними системами

LUNTRY | Deckhouse ФЛАНТ

Вывод

УНИКАЛЬНАЯ ЧЕРТА

- Гибридный подход (поведение + правила)
- File Integrity Monitoring (FIM) для процессов
- Сбор артефактов для расследования
- Prevention политики
- Анализ на server-side

ЧТО ЭТО ДАЕТ

- Минимальное количество ложных срабатываний
- Возможность проводить форензику
- Минимальная нагрузка на наблюдаемые Nodes

ЧТО БЕЗ ЭТОГО

Очень сложное внедрение
и поддержка в процессе работы

03 ЗАКЛЮЧЕНИЕ

LUNTRY | Deckhouse ФЛАНТ

ЗАКЛЮЧЕНИЕ

01 В каждом домене безопасности Kubernetes есть несколько уровней защищённости.

02 Функциональность функциональности рознь – она должна решать проблему или задачу, а не просто быть.

03 Комплексный подход, объединяющий контроль целостности, управление секретами и мониторинг поведения, позволяет создавать надёжные и устойчивые к атакам платформы.

04 Уникальные черты Luntry и Deckhouse дополняют друг друга и дают преимущества для безопасности.

luntry_official
luntrysolution
luntrysolution
luntry.ru
info@luntry.ru

deckhouse_news
Хабр Флант
flant
deckhouse.ru
contact@deckhouse.ru

ДМИТРИЙ ЕВДОКИМОВ

Founder & CTO Luntry

de@luntry.ru
Qu3b3c
k8security

АЛЕКСЕЙ КРЫЛОВ

Менеджер продукта Deckhouse Kubernetes Platform
по направлению «Информационная безопасность» (АО «Флант»)

aleksey.krylov@flant.ru
krylov_av

LUNTRY | Deckhouse ФЛАНТ