Безопасность ML кластеров Kubernetes

Сергей Канибор

R&D/Container Security, Luntry

Whoami

- R&D/Container Security в Luntry
- Специализируюсь на безопасности контейнеров и Kubernetes
- Багхантер
- Редактор Telegram-канала “k8s (in)security”
- Спикер: PHDays, OFFZONE, VK Kubernetes Conf, Devoops, SafeCode, Kuber Conf, HackConf, CyberCamp, БеКон и др.

Функциональность Luntry

Luntry («Лантри») – это комплексная защита на всем жизненном цикле контейнерных приложений и средств оркестрации на базе K8s.

Agenda

- Реалии 2025
- Проблемы кластеров общего назначения
- Проблемы ML кластеров

Реалии 2025

AI повсюду

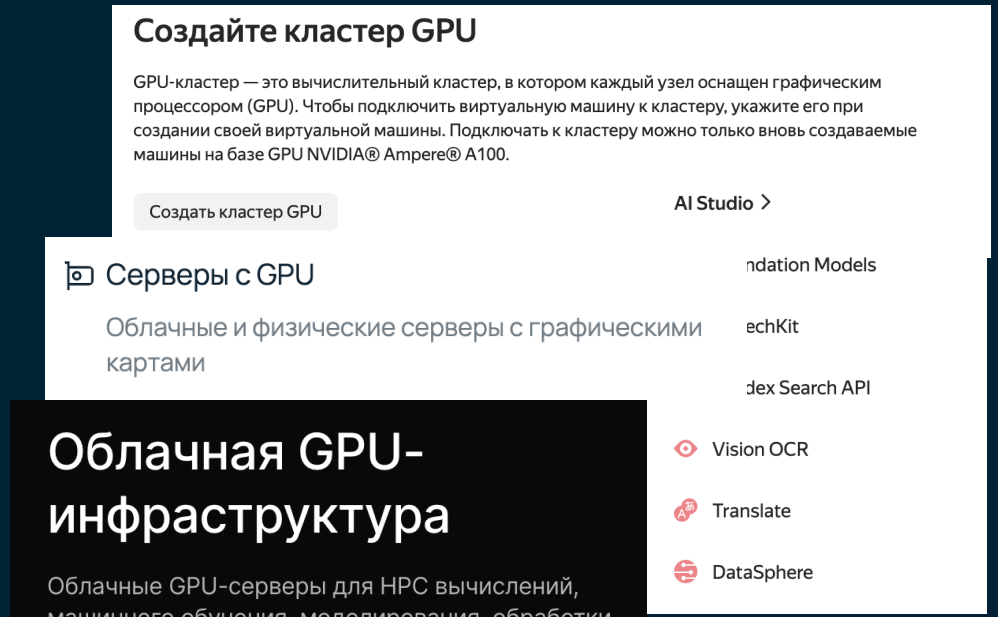
- LLM
- AI ассистенты
  - Салют
  - GigaChat
  - YandexGPT
  - Copilot
  - Cursor
  - ...
  - Приватные

Объем российского рынка LLM-продуктов для бизнеса (Large Language Model, «большая языковая модель») по итогам 2024 года составит 35 млрд руб., подсчитали в Центре искусственного интеллекта МТС (MTS AI). До 2028 года этот показатель будет расти в среднем на 25% в год, прогнозируют аналитики центра.

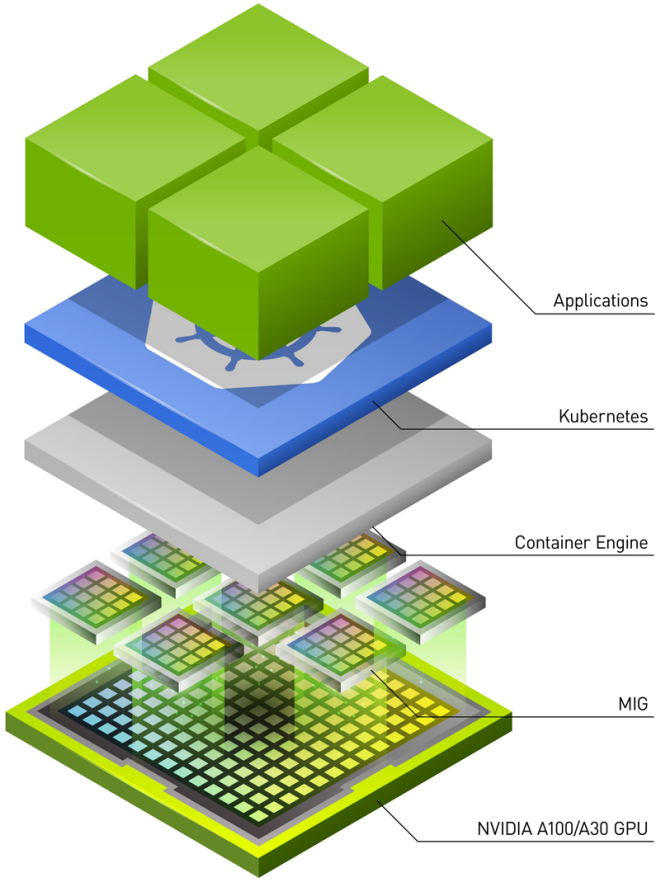
комплектующими, считает Александр Ализар из «Ситилинка». По мнению директора по закупкам направления электронной коммерции компании diHouse Алексея Мазева, рост спроса обусловлен развитием технологий искусственного интеллекта. Они теперь работают в том числе при использовании мощных графических адаптеров.

GPU используются везде

- ➔ Облачные провайдеры
  - ◆ Отечественные
  - ◆ Зарубежные
- ➔ Внутренние ML платформы



Также и в Kubernetes



Проблемы кластеров общего назначения

Основные риски контейнеризации

НА УРОВНЕ ОБРАЗОВ	НА УРОВНЕ КОНТЕЙНЕРОВ	НА УРОВНЕ ОРКЕСТРАТОРА
• Уязвимости в компонентах образа • Небезопасная конфигурация • Наличие чувствительной информации • Вредоносное ПО и ПО двойного назначения • Отсутствие подлинности образов	• Недекларированные возможности • Небезопасная конфигурация • Избыточные привилегии и возможности	• Неподдерживаемые системные компоненты • Уязвимости в системных компонентах • Отсутствие изоляции сред • Отсутствие сетевой сегментации • Небезопасная конфигурация • Остуствие контроля доступа • Избыточные привилегии сущностей

Пример: на уровне образов

Unauthorized image of Kong Ingress Controller v.3.4.0

Critical **mrwanny** published **GHSA-58mg-ww7q-xw3p** 3 weeks ago

Package	Affected versions	Patched versions
kong/kubernetes-ingress-controller	3.4.0	3.4.1

Description

Summary

On December 23, 2024, an unauthorized image of Kong Ingress Controller v.3.4.0 (hash: `sha256:a00659df0771d076fc9d0baf1f2f45e81ec9f13179f499d4cd940f57afc75d43`) was uploaded to DockerHub containing code that enabled cryptojacking in the form of calls to a crypto mining site pool.supportxmr.com.

On January 2, 2025, soon after becoming aware of the issue, we deleted version 3.4.0 and associated tags from DockerHub, rotated all affected access keys to DockerHub, and later on January 2, 2025 released version 3.4.1 which removed the unauthorized code.
We have no evidence to date to suggest that any other images (before or after the hash specified above) were affected.

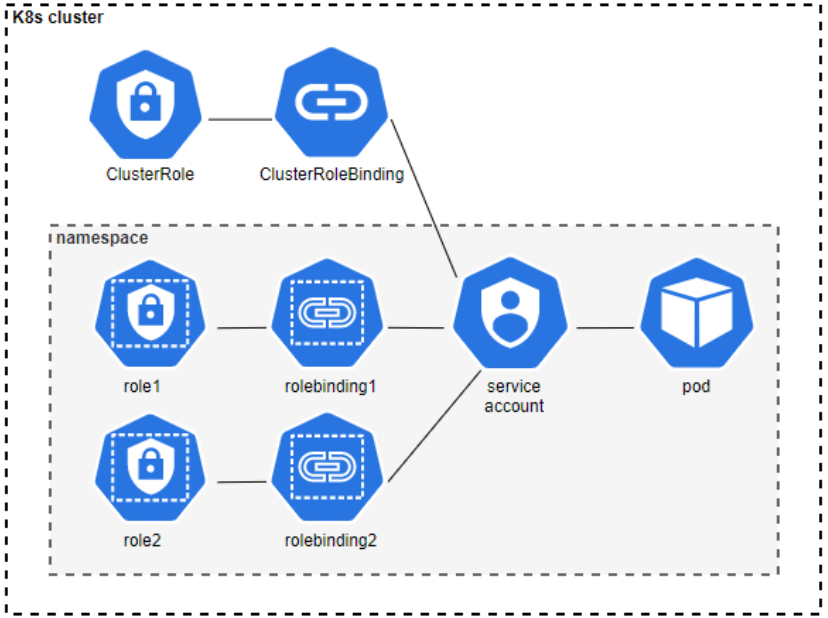
Пример: на уровне контейнеров

- ➔ Повышение привилегий
- ➔ Побег из контейнера

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: everything-allowed-exec-pod
  labels:
    app: pentest
spec:
  hostNetwork: true
  hostPID: true
  hostIPC: true
  containers:
    - name: everything-allowed-pod
      image: ubuntu
      securityContext:
        privileged: true
      volumeMounts:
        - mountPath: /host
          name: noderoot
      command: ["/bin/sh", "-c", "--"]
      args: ["while true; do sleep 30; done;"]
  volumes:
    - name: noderoot
      hostPath:
        path: /
```

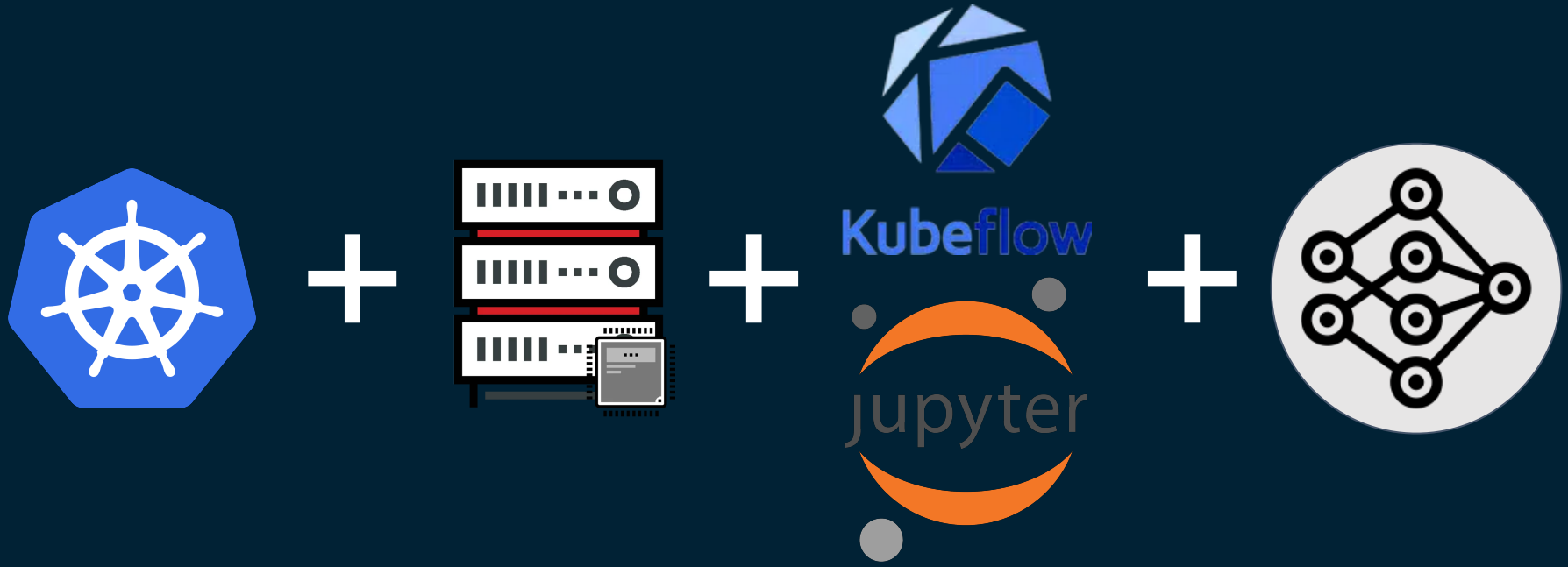
Пример: на уровне оркестратора

- ➔ Излишние RBAC привилегии
- ➔ Создание новых сущностей
- ➔ Закрепление



Проблемы ML кластеров

Рецепт ML кластера

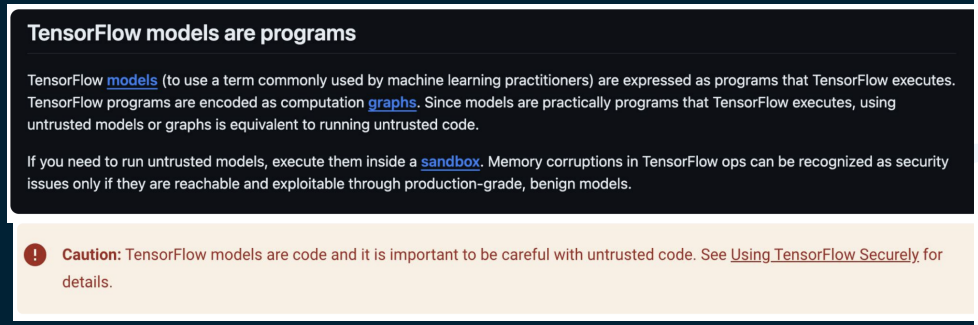


Железо

- ➔ CDI
- ➔ Проблемы драйверов
- ➔ Расширение attack surface

ML софт

- ➔ 1day и 0day уязвимости
- ➔ Insecure by design

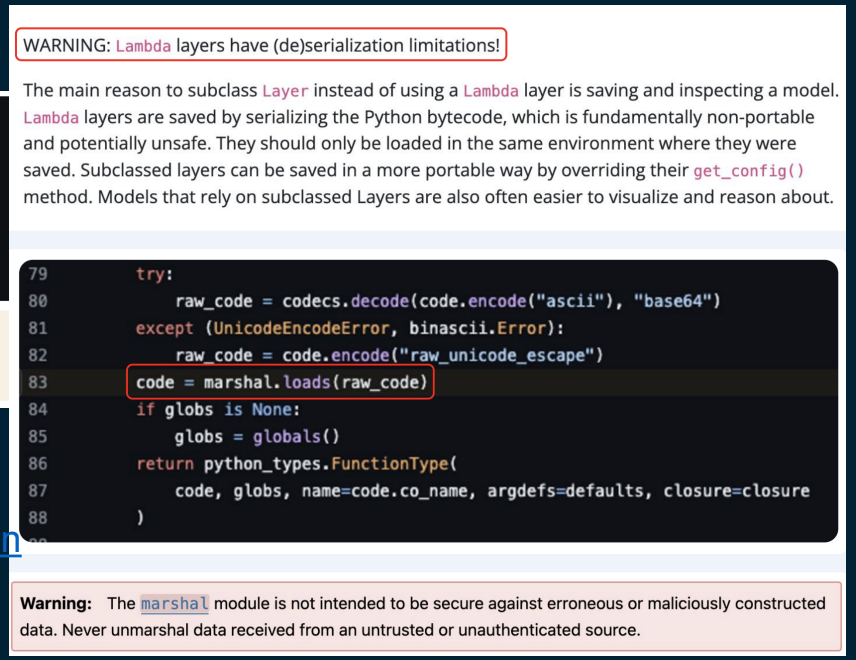


TensorFlow models are programs

TensorFlow models (to use a term commonly used by machine learning practitioners) are expressed as programs that TensorFlow executes. TensorFlow programs are encoded as computation graphs. Since models are practically programs that TensorFlow executes, using untrusted models or graphs is equivalent to running untrusted code.

If you need to run untrusted models, execute them inside a sandbox. Memory corruptions in TensorFlow ops can be recognized as security issues only if they are reachable and exploitable through production-grade, benign models.

Caution: TensorFlow models are code and it is important to be careful with untrusted code. See Using TensorFlow Securely for details.



WARNING: `Lambda` layers have (de)serialization limitations!

The main reason to subclass `Layer` instead of using a `Lambda` layer is saving and inspecting a model. `Lambda` layers are saved by serializing the Python bytecode, which is fundamentally non-portable and potentially unsafe. They should only be loaded in the same environment where they were saved. Subclassed layers can be saved in a more portable way by overriding their `get_config()` method. Models that rely on subclassed Layers are also often easier to visualize and reason about.

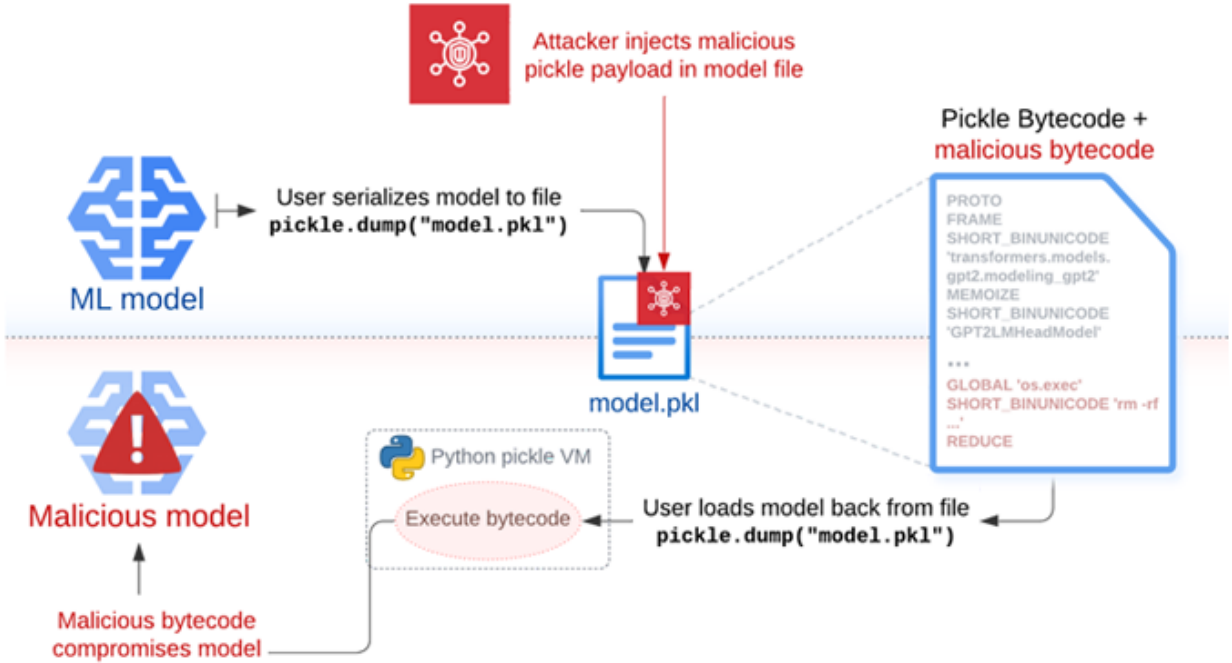
```
79          try:
80              raw_code = codecs.decode(code.encode("ascii"), "base64")
81          except (UnicodeEncodeError, binascii.Error):
82              raw_code = code.encode("raw_unicode_escape")
83          code = marshal.loads(raw_code)
84          if globs is None:
85              globs = globals()
86          return python_types.FunctionType(
87              code, globs, name=code.co_name, argdefs=defaults, closure=closure
88          )
```

Warning: The `marshal` module is not intended to be secure against erroneous or maliciously constructed data. Never unmarshal data received from an untrusted or unauthenticated source.

[Shadow Vulnerabilities in AI/ML Data Stacks - What You Don't Know CAN Hurt You - Avi Lumelsky & Nitzan Mousseri, Oligo Security]

Специфичные данные

➔ Небезопасная десериализация



Специфичные модели нарушителя

➔ Скомпрометированный разработчик,
с возможностью изменять ML модели (Training Data Poisoning)

Выводы

1. Каждый кластер уникален
2. К каждому нужен особый подход
3. Еще больше особенностей и пользователей

Спасибо!

Сергей Канибор
R&D / Container Security

Email: sk@luntry.ru
Channel: @k8security
Site: www.luntry.ru