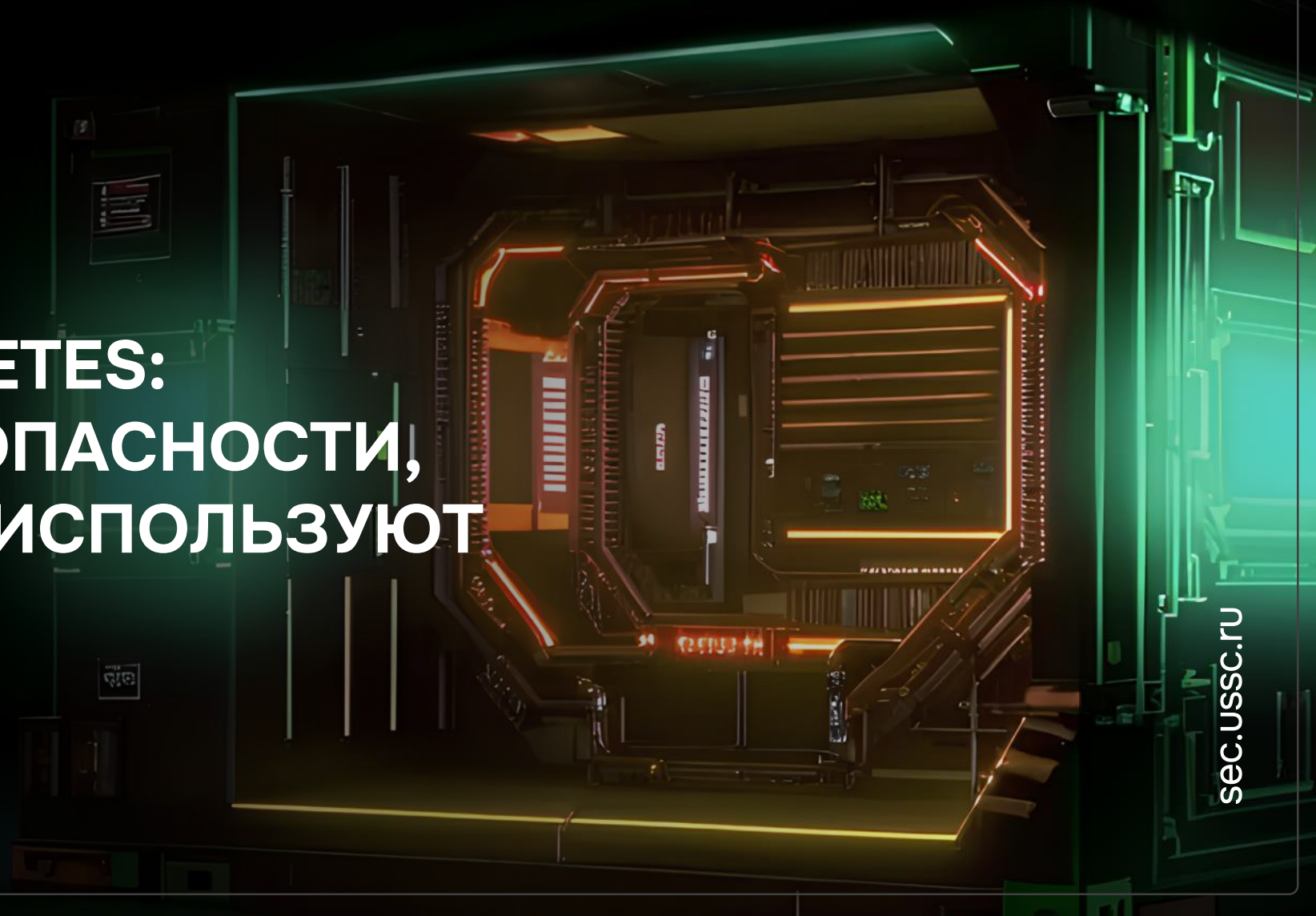
ЦЕНТР
КИБЕРБЕЗОПАСНОСТИ



ЗАЩИТА KUBERNETES: НАСТРОЙКИ БЕЗОПАСНОСТИ, КОТОРЫЕ РЕДКО ИСПОЛЬЗУЮТ

2024

sec.ussc.ru

WHOAMI

- Руководитель направления «Безопасная разработка» в Центре кибербезопасности УЦСБ
- Помогаем нашим Заказчикам создавать безопасные приложения
- Обеспечиваем безопасность облаков и микросервисов
- Запустил собственную платформу анализа защищенности
- Участвую в создании профильных мероприятий

Agenda

Расскажем

- Что такое Ephemeral Containers и почему стоит заботиться об их безопасности
- Как использование Scheduling Kubernetes способно сократить поверхность атаки
- Как использовать Honeypot и снять дамп работающего контейнера в K8s для расследования инцидентов безопасности

Продемонстрируем

- Как Kaspersky Container Security обеспечивает безопасность современных приложений, построенных с использованием контейнеров и оркестраторов
- Какие возможности и преимущества есть у решения «Лаборатории Касперского», в каком направлении оно будет развиваться

EPHEMERAL CONTAINERS

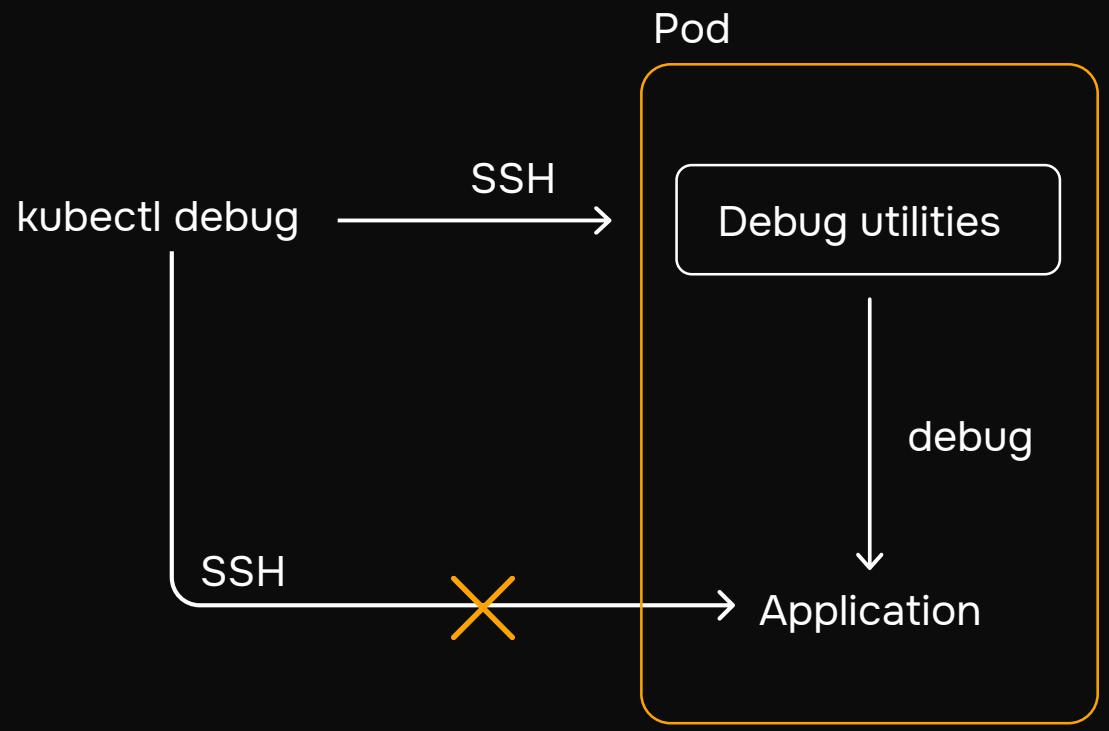
Что такое Ephemeral Containers?

Пожалуйста, ответьте на вопросы
на вашем экране

Ephemeral Containers

- Временные контейнеры, цель которых помочь в отладке ПО
- Размещаются в том же Pod, где находится целевой контейнер
- Имеют необходимое ПО и привилегии для работы

Заблуждение – для эфемерных контейнеров нельзя установить контекст безопасности (можно!)



Обеспечение безопасности

01.

RBAC

- Права на subresource /ephemeralcontainers
- По умолчанию предоставляется в рамках привилегированных ролей
- Желательно использовать сторонний контроллер доступа

Обеспечение безопасности

01.

RBAC

- Права на subresource /ephemeralcontainers
- По умолчанию предоставляется в рамках привилегированных ролей
- Желательно использовать сторонний контроллер доступа

02.

Admission

- Поддержка Gatekeeper и Kyverno
- Необходимо обновление
- Требуется разработка отдельной политики

Обеспечение безопасности

01.

RBAC

- Права на subresource /ephemeralcontainers
- По умолчанию предоставляется в рамках привилегированных ролей
- Желательно использовать сторонний контроллер доступа

02.

Admission

- Поддержка Gatekeeper и Kyverno
- Необходимо обновление
- Требуется разработка отдельной политики

03.

Audit

- Контроль изменения RBAC
- Контроль обращения к subresource
- Аудит нарушения Admission Policy

Обеспечение безопасности

01.

RBAC

- Права на subresource /ephemeralcontainers
- По умолчанию предоставляется в рамках привилегированных ролей
- Желательно использовать сторонний контроллер доступа

02.

Admission

- Поддержка Gatekeeper и Kyverno
- Необходимо обновление
- Требуется разработка отдельной политики

03.

Audit

- Контроль изменения RBAC
- Контроль обращения к subresource
- Аудит нарушения Admission Policy

04.

PSP

- Deprecated
- Не поддерживает политики для Ephemeral Containers

SCHEDULING

Kubernetes Scheduler

- Часть Kubernetes Control Plane
- Отвечает за распределение подов по рабочим узлам
- Оперирует правилами affinity/anti-affinity, taints и tolerations

Безопасность: позволяет предотвратить горизонтальное перемещение и/или соответствовать требованиям регулятора

Механизмы планирования

01. | nodeSelector

- Указываются метки нод для планирования
- Между метками работает правило AND

Пример:

```yaml
apiVersion: v1
kind: pod
metadata:
  name: nodeSelector-pod
spec:
  containers:
  - name: nginx
    image: nginx:latest
  nodeSelector:
    label: criticalNode
```

Механизмы планирования

02. | nodeName

- Запустится только на ноде с указанным именем
- Имеет приоритет перед другими подходами планирования
- Не позволяет использовать преимущества планировщика

Пример:

```yaml
apiVersion: v1
kind: pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx:latest
  nodeName: criticalNode
```

Механизмы планирования

03. | Affinity & Anti-affinity

- Правила на основе характеристик или меток узлов
- Anti-affinity позволяет исключить узлы, например, по признаку критичности
- Для безопасности приоритетны жесткие правила

Пример:

```yaml
apiVersion: v1
kind: pod
metadata:
  name: node-affinity
spec:
  affinity:
   nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
        - matchExpressions:
          - key: net-segment
            operator: In
            values:
            -  payment-segment
  containers:
  - name: node-affinity
    image: registry.local/web:2.0
```

Механизмы планирования

04. Inter-pod Affinity & Anti-affinity

- Позволяет запускать или не запускать под в домене, если там уже запущен другой под, удовлетворяющий определенным правилам

Пример:

```
affinity:
  podAntiAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
    - labelSelector:
        matchExpressions:
        - key: app
          operator: In
          values:
          - testapp
```

Механизмы планирования

05. | Taints and Tolerations

- Equal – необходимо полное совпадение key, value, effects
- Exist – необходимо совпадение с key и effects
- NoSchedule – pods без toleration не запустится на node с taint
- NoExecute – pods, работающий без toleration, будет удален с taint Node

Пример:

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    env: test
spec:
  containers:
  - name: nginx
    image: nginx
    imagePullPolicy: IfNotPresent
  tolerations:
  - key: "example-key"
    operator: "Exists"
    effect: "NoSchedule"
```

Рекомендации по безопасности

- **Контролируйте права, позволяющие изменять метаданные.** Без этого появляется риск манипуляции метками и увеличения поверхности атаки
- **Проверяйте политику планирования.** В случае некорректных настроек возможно размещение критичных и некритичных компонентов на одном узле
- **Запретите Kubelet управлять метками ноды,** используя Admission plugin NodeRestriction

Занимаетесь ли вы планированием нагрузок?

Пожалуйста, ответьте на вопросы
на вашем экране

ОБНАРУЖЕНИЕ И РАССЛЕДОВАНИЕ ИНЦИДЕНТОВ

Аудит K8s: Request Stage and Policy Audit

Ведение журнала аудита используется для следующих стадий

- RequestReceived
- ResponseStarted
- ResponseComplete
- Panic

Уровни аудита для настройки политик

None – не регистрировать события

Metadata – регистрировать метаданные запроса

Request – metadata + тело запроса

RequestResponse – request + тело ответа

Анализируем действия с ресурсами (Verbs)

- Namespace
- User / userGroup
- Resource
- Group
- nonResourceURLs

Расширение аудита с использованием HoneyPot

Неиспользуемые ресурсы

- Ingress
- Secrets / ConfigMap
- Service Acount
- Pod
- etc

Внесение изменений в текущие нагрузки

- С помощью MutatingAdmissionWebhook можно добавлять в переменные окружения информацию и отслеживать обращение к ней
- Или подкладывать специальные файлы и также смотреть за обращением к ним

Разворачивать уязвимые сервисы по аналогии с классическим HoneyPot

Forensic container analysis

Что это

- Позволяет создавать копии работающего контейнера с отслеживанием состояния без его остановки и оказания влияния
- Работает на основе CRIU (Checkpoint/Restore In Userspace)
- Потенциальный злоумышленник не заметит снятие копии контейнера

Как использовать

- Необходимо установить *CRIU*, запустить кластер с *ContainerCheckpoint*
- Для снятия копии необходимо на ноде выполнить

curl -X POST "https://localhost:10250/checkpoint/namespace/podId/container"

- Результат будет расположен

/var/lib/kubelet/checkpoints/checkpoint-<pod-name>_<namespace-name>-<container-name>-<timestamp>.tar

- В завершении необходимо запустить контейнер в песочнице (на примере CRI-O)

crictl runp pod-config.json

crictl create <POD_ID> container-config.json pod-config.json

crictl start <CONTAINER_ID>

- Возможно исследовать полученный архив с помощью утилиты *Checkpointctl*

Программа вебинаров

23.04 Container Security: комплексный подход к безопасности K8s. Совместно с Orion soft

14.05 Защита Kubernetes: настройки безопасности, которые редко используют. Совместно с «Лабораторией Касперского»

Вебинар совместно с Positive Technologies

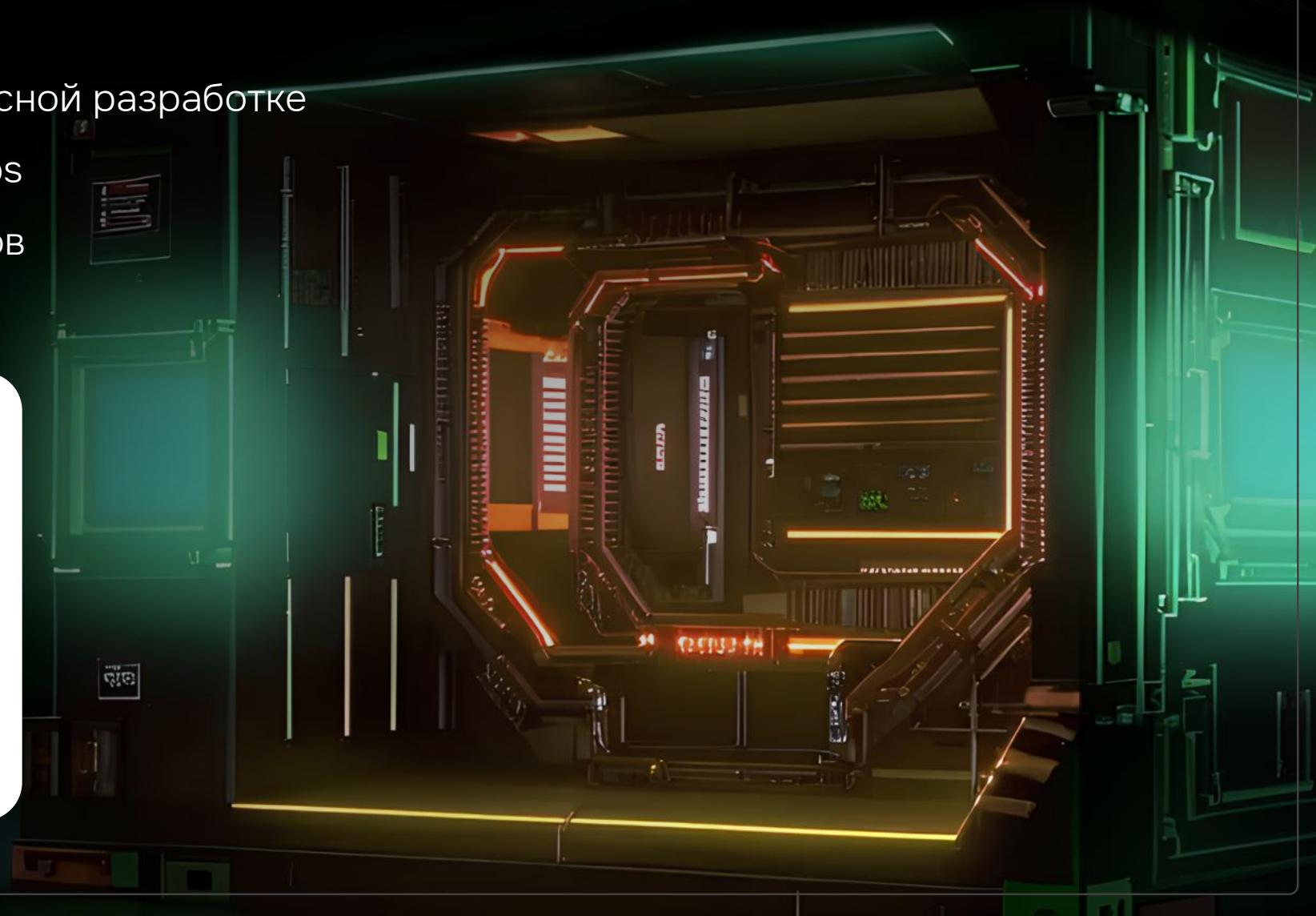
Автоматизация процессов защиты Kubernetes

Вебинар совместно с Luntry

- Демонстрация решений
- Атаки на Kubernetes и защита кластера
- Обнаружение атак, реагирование и блокирование действий злоумышленника
- Автоматизированное реагирование на инциденты в среде Kubernetes с использованием SOAR

ПОДПИСЫВАЙТЕСЬ НА НАШ КАНАЛ В ТЕЛЕГРАМЕ

- Практические кейсы по безопасной разработке
- Экспертные статьи о DevSecOps
- Анонсы тематических вебинаров



ЦЕНТР КИБЕРБЕЗОПАСНОСТИ

sec.ussc.ru

cybersec@ussc.ru

Евгений Тодышев

etodishev@ussc.ru +7 (950) 555-68-90 @mr.appsec