3 июня 2025 Москва, LOFT HALL#2

БЕКОН '25

Конференция по БЕзопасности
КОНтейнеров и контейнерных сред

Неочевидные и непонятные моменты безопасности Kubernetes

Дмитрий Евдокимов

Founder & CTO Luntry

Обо мне

БЕКОН

> 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

Disclaimer

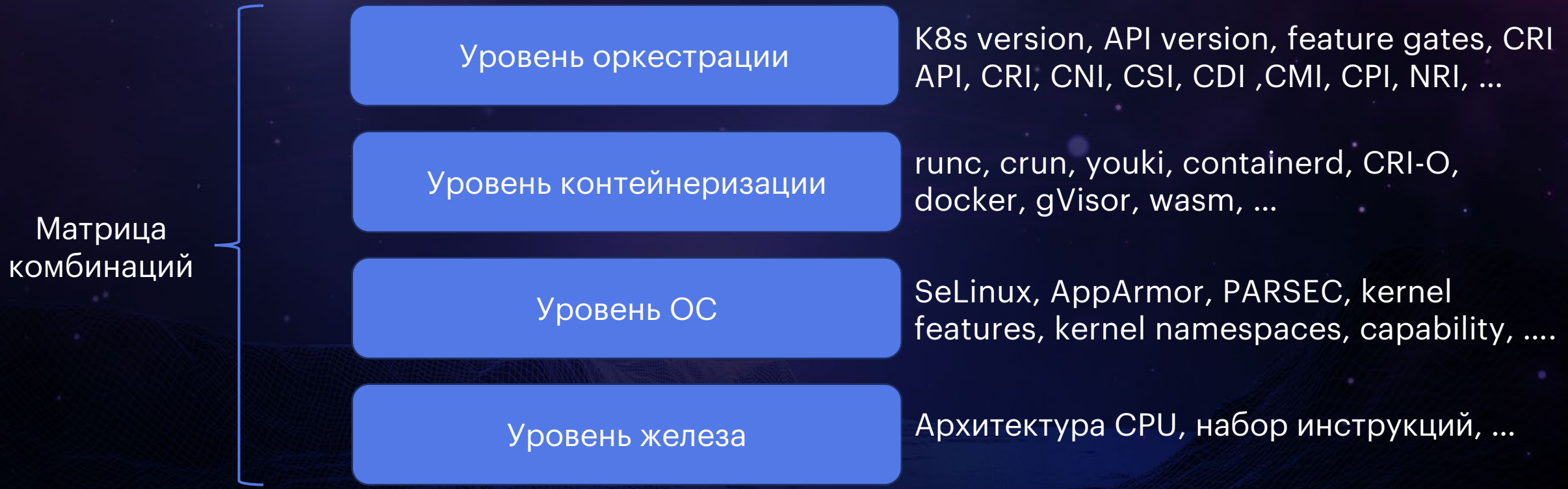
БЕКОН

От момента идеи доклада до его появления
прошло около года.

За это время в Kubernetes и контейнерных компонентах
поменялось так много, что концепция доклада
была практически полностью изменена
и дополнена новыми смыслами.

Как это вообще работает?

БЕКОН



БЕКОН

Все легко посмотреть?)

Ситуация с “kubectl get all”

БЕКОН

kubeclt -n namespace get all – не работает, как ожидается!

kubectl get all does not list all resources in a namespace #151

Closed **tback** opened this issue on Nov 28, 2017 · 89 comments

What happened:

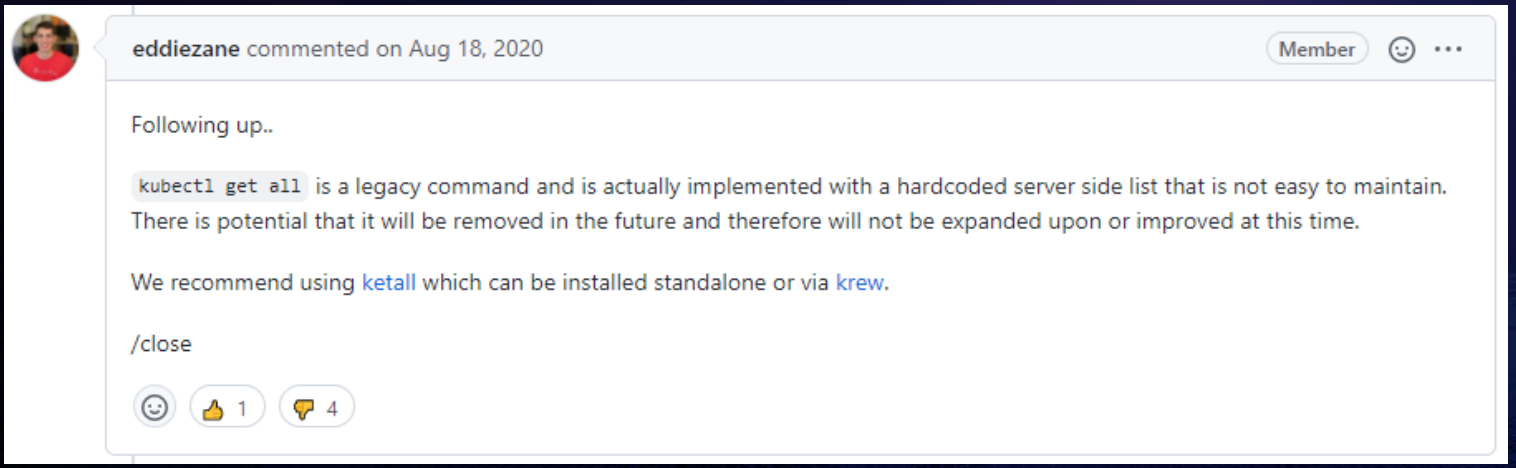
`kubectl get all` does not list all resources in a namespace.

Need a real "get-all" command #527

Closed **schollii** opened this issue on Aug 29, 2018 · 29 comments

What happened:

`kubectl get all` only shows a small subset of kubernetes objects in a cluster, and there does not seem to be a command to get all objects (secrets, network policies, etc). This caused me hours of wasted time because I was trying to replicate a deployment object from another cluster into a new cluster, and didn't see that there was a networkpolicy needed for the service to expose that deployment.



eddiezane commented on Aug 18, 2020 — Member

Following up..

`kubectl get all` is a legacy command and is actually implemented with a hardcoded server side list that is not easy to maintain. There is potential that it will be removed in the future and therefore will not be expanded upon or improved at this time.

We recommend using ketall which can be installed standalone or via krew.

/close

1 4

БЕКОН

Как читается так и пишется?

PodSecurityContext & SecurityContext*

БЕКОН

В Спецификации Pod

securityContext *PodSecurityContext*	SecurityContext holds pod-level security attributes and common container settings. Optional: Defaults to empty. See type description for default values of each field.

В разделе containers, initContainers

securityContext *SecurityContext*	SecurityContext defines the security options the container should be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext. More info: https://kubernetes.io/docs/tasks/configure-pod-container/security-context/

В разделе ephemeralContainers

securityContext *SecurityContext*	Optional: SecurityContext defines the security options the ephemeral container should be run with. If set, the fields of SecurityContext override the equivalent fields of PodSecurityContext.

PodSecurityContext

SecurityContext

```
apiVersion: v1
kind: Pod
metadata:
  name: mypod
spec:
  SecurityContext:
    runAsUser: 1000
  containers:
  - name: my-container
    image: my-image
    SecurityContext:
      runAsUser: 2000
```

* - по документации для v1.33

БЕКОН

Одинаковый смысл и одинаковые значения?

Настройки PodSecurityContext & SecurityContext*

БЕКОН

PodSecurityContext	SecurityContext
appArmorProfile	appArmorProfile
runAsGroup	runAsGroup
runAsNonRoot	runAsNonRoot
runAsUser	runAsUser
seLinuxOptions	seLinuxOptions
seccompProfile	seccompProfile
windowsOptions	windowsOptions
fsGroup	-
fsGroupChangePolicy	-
seLinuxChangePolicy	-
supplementalGroups	-
supplementalGroupsPolicy	-
sysctls	-
-	allowPrivilegeEscalation
-	capabilities
-	privileged
-	procMount
-	readOnlyRootFilesystem

* - по документации для v1.33

БЕКОН

Как написал, так и будет? Часть 1

allowPrivilegeEscalation by default?!

БЕКОН

По умолчанию true

`allowPrivilegeEscalation` *boolean*

AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the no_new_privs flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1) run as Privileged 2) has CAP_SYS_ADMIN Note that this field cannot be set when spec.os.name is windows.

0xbf00 on Mar 17, 2023 · edited by 0xbf00 Edits

```
AllowPrivilegeEscalation controls whether a process can gain more
privileges than its parent process. This bool directly controls if
the no_new_privs flag will be set on the container process.
Default to false.
AllowPrivilegeEscalation is true always when the container is:
1. run as Privileged and
2. has CAP_SYS_ADMIN
Note that this field cannot be set when spec.os.name is windows.```
```

I believe this description is not (no longer?) correct, because:

1. `AllowPrivilegeEscalation` defaults to `true`, see this code. The code returns `false` both if `AllowPrivilegeEscalation` is not set and if `AllowPrivilegeEscalation` is set to `true`. This is also supported by my understanding of the code you already linked. Here, containers with `container.SecurityContext.AllowPrivilegeEscalation == nil` are flagged as bad by the script.
2. the current implementation adds an error if *either* the container is privileged **or** has the `CAP_SYS_ADMIN` capability.

Note however that therefore and confusingly, `no_new_privs` defaults to `false`. The documentation however describes `AllowPrivilegeEscalation` and not `no_new_privs` directly.

Ссылка на issue

БЕКОН

Как написал, так и будет? Часть 2

Capability CAP_NET_RAW by default?!

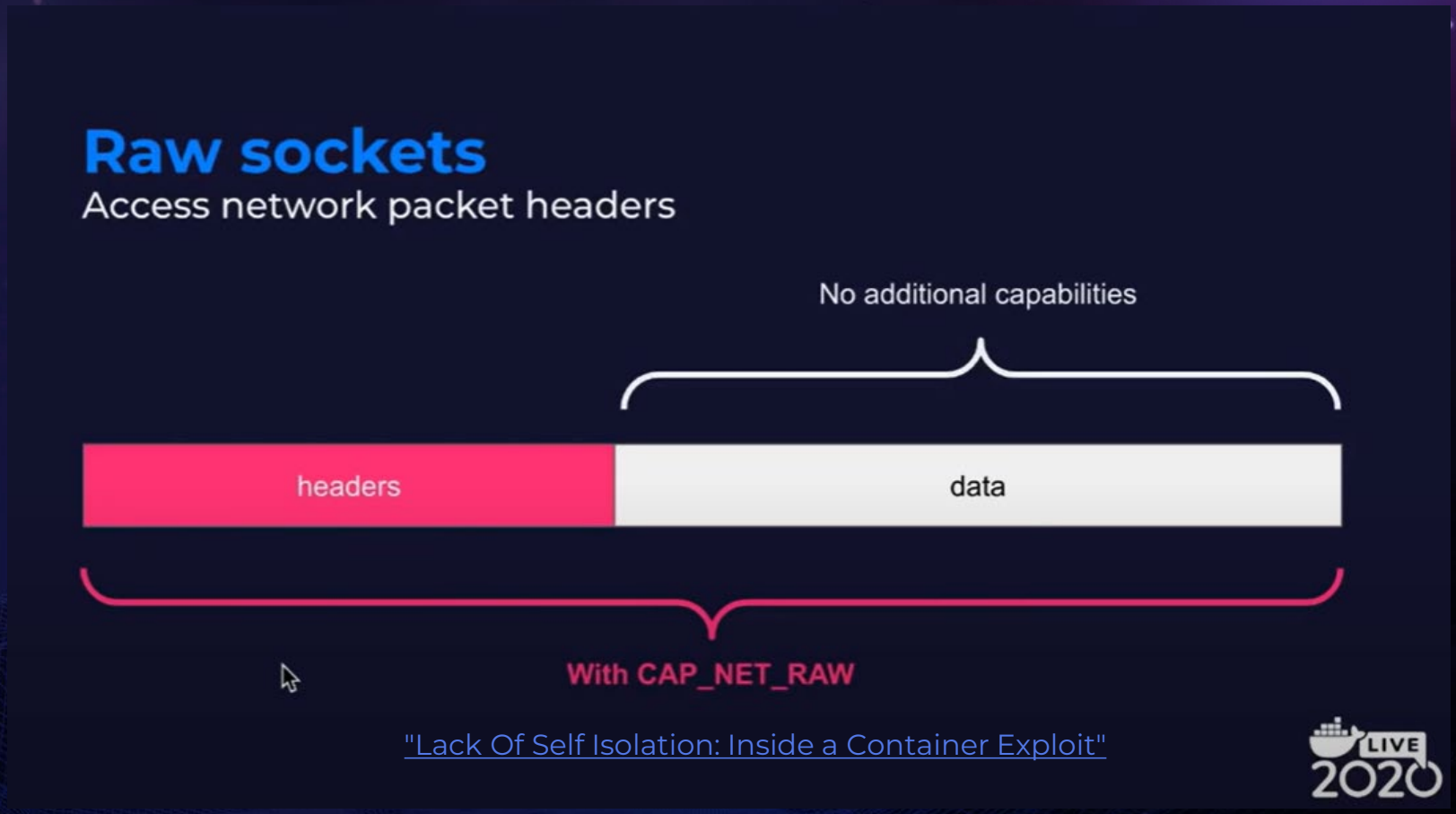
БЕКОН

cap_net_raw - Разрешает использование сырых сокетов

Можно проводить атаки
ARP и DNS spoofing.

Расширяет поверхность атаки:

- CVE-2021-23134
- CVE-2020-14386
- CVE-2018-1065
- CVE-2017-7308
- …




```
root@mtkpi-68d67d484-9tc2b:/run# capsh --print
WARNING: libcap needs an update (cap=40 should have a name).
Current: = cap_chown,cap_dac_override,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_net_bind_service,cap_net_raw,cap_sys_chroot,cap_mknod,cap_audit_write,cap_setfcap+ep
Bounding set =cap_chown,cap_dac_override,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_net_bind_service,cap_net_raw,cap_sys_chroot,cap_mknod,cap_audit_write,cap_setfcap
```

CAP_NET_RAW на Bug Bounty

БЕКОН

#899103

4

Man in the middle leading to root privilege escalation using hostNetwork=true (CAP_NET_RAW considered harmful)

Share:

SUMMARY BY CHAMPTAR

Write up with both EKS and GKE exploits and more details available at https://blog.champtar.fr/Metadata_MITM_root_EKS_GKE/

TIMELINE

champtar submitted a report to Kubernetes. June 16, 2020, 12:26am UTC

Summary:

CAP_NET_RAW capability is still included by default in K8S, leading to yet another attack.

An attacker gaining access to a hostNetwork=true container with CAP_NET_RAW capability can listen to all the traffic going through the host and inject arbitrary traffic, allowing to tamper with most unencrypted traffic (HTTP, DNS, DHCP, ...), and disrupt encrypted traffic.
In many cloud deployments the host queries the metadata service at http://169.254.169.254 to get many information including the authorized ssh keys.
This report contains a POC running on GKE, manipulating the metadata service responses to gain root privilege on the host.
The same attack should work on all clouds using similar metadata services to provision ssh keys (Amazon / Azure / OpenStack / ...)

The goal of this report is to ask the K8S team to make a breaking change by removing CAP_NET_RAW from the default capabilities,
as it allows way too many attacks.
K8S could enable `net.ipv4.ping_group_range` to still let users use ping (maybe 99% of CAP_NET_RAW usage)

Ссылка на h1

CRI-O

С версии 1.18.0 (2020 год)

БЕКОН

Apr 23, 2020

haircommander

v1.18.0

7d79f42

Compare

v1.18.0

CRI-O v1.18.0

The release notes have been generated for the commit range
v1.17.0...v1.18.0 on Thu, 23 Apr 2020 07:43:06 UTC.

Downloads

Download the static release bundle via our Google Cloud Bucket:
crio-v1.18.0.tar.gz

Changes by Kind

Dependency-Change

- Update cri-tools to v1.18.0 in static release bundle (#3520, @saschagrunert)
- Default pause image is `k8s.gcr.io/pause:3.2` (#3582, @haircommander)

Deprecation

- Drop support for golang < v1.13 (#3312, @saschagrunert)

API Change

- Removed version from default AppArmor profile name in config (#3287, @saschagrunert)
- CRI-O now runs containers without NET_RAW and SYS_CHROOT capabilities by default. This can result in permission denied errors when the container tries to do something that would require either of these capabilities. For instance, using `ping` requires NET_RAW, unless the container is given the sysctl `net.ipv4.ip_forward`. Further, if you have a container that runs buildah or configures RPMs, they may fail without SYS_CHROOT. Ultimately, the dropped capabilities are worth it, as the majority of containers don't need them. The fewer capabilities CRI-O gives out by default, the more

Commit 63b9f4e

rhatdan authored and haircommander committed on Mar 24, 2020

```
Remove NET_RAW and SYS_CHROOT capabilities

as well as add net.ipv4.ping_group_range sysctl for tests

also grabbed some changes introduced by @Chenditang on github to
update the crio.conf man page

Increase the security of containers by removing the NET_RAW and CHROOT capabilties.

Suggest adding net.ipv4.ping_group_range so that ping will still work.
```

Ссылка на release notes

```
root@mtkpi-77b59b9d6d-mlbdk:/run# capsh --print
WARNING: libcap needs an update (cap=40 should have a name).
Current: = cap_chown,cap_dac_override,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_net_bind_service+ep
Bounding set =cap_chown,cap_dac_override,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_net_bind_service
```

containerd

БЕКОН

С версии 2.0 (2025 год)

containerd / docs / containerd-2.0.md ↑ Top

Preview | Code | Blame 187 lines (108 loc) · 13.7 KB Raw

Unprivileged ports and ICMP by default for CRI

The CRI plugin now enables `net.ipv4.ip_unprivileged-port-start=0` and `net.ipv4.ping_group_range=0 2147483647` for containers that do not use the host network namespace or user namespaces. This enables containers to bind to ports below 1024 without granting `CAP_NET_BIND_SERVICE` and to run `ping` without `CAP_NET_RAW`. This default behavior change can be reverted by setting the `enable_unprivileged_ports` and `enable_unprivileged_icmp` options to `false` in the CRI plugin configuration.

Ссылка на документацию containerd

Docker

БЕКОН

Танцы вокруг ping

Remove CAP_NET_RAW from default capabilities #41886

Open

thaJeztah opened on Jan 14, 2021

Mostly a tracking issue as follow-up to #41030 (#41030 (comment))

Many base images now no longer require `CAP_NET_RAW` to use `ping` in a container, but there's some images that may still need it.

Questions:

1. which base images remain to be problematic? (IIRC (some versions of) ubuntu)
2. can we (and if so:) would it be acceptable to patch the upstream source when building our official base images to allow `ping` to run without `CAP_NET_RAW` ?

Answering myself; 2. is probably not an option; looks like `ping` is available by default on debian, but (no longer) installed by default on ubuntu, and requires `iputils-ping` to be installed

Busybox 1.36.1-glibc

```
docker run --rm --cap-drop CAP_NET_RAW busybox:1.36.1-glibc sh -c 'ping -c1 google.com'
ping: permission denied (are you root?)
PING google.com (142.250.179.142): 56 data bytes
```

Busybox 1.36.1-uclibc

```
docker run --rm --cap-drop CAP_NET_RAW busybox:1.36.1-uclibc sh -c 'ping -c1 google.com'
ping: permission denied (are you root?)
PING google.com (142.250.179.142): 56 data bytes
```

Busybox 1.36.1-musl

```
docker run --rm --cap-drop CAP_NET_RAW busybox:1.36.1-musl sh -c 'ping -c1 google.com'
PING google.com (142.250.179.142): 56 data bytes
ping: permission denied (are you root?)
```

Ссылка на issues

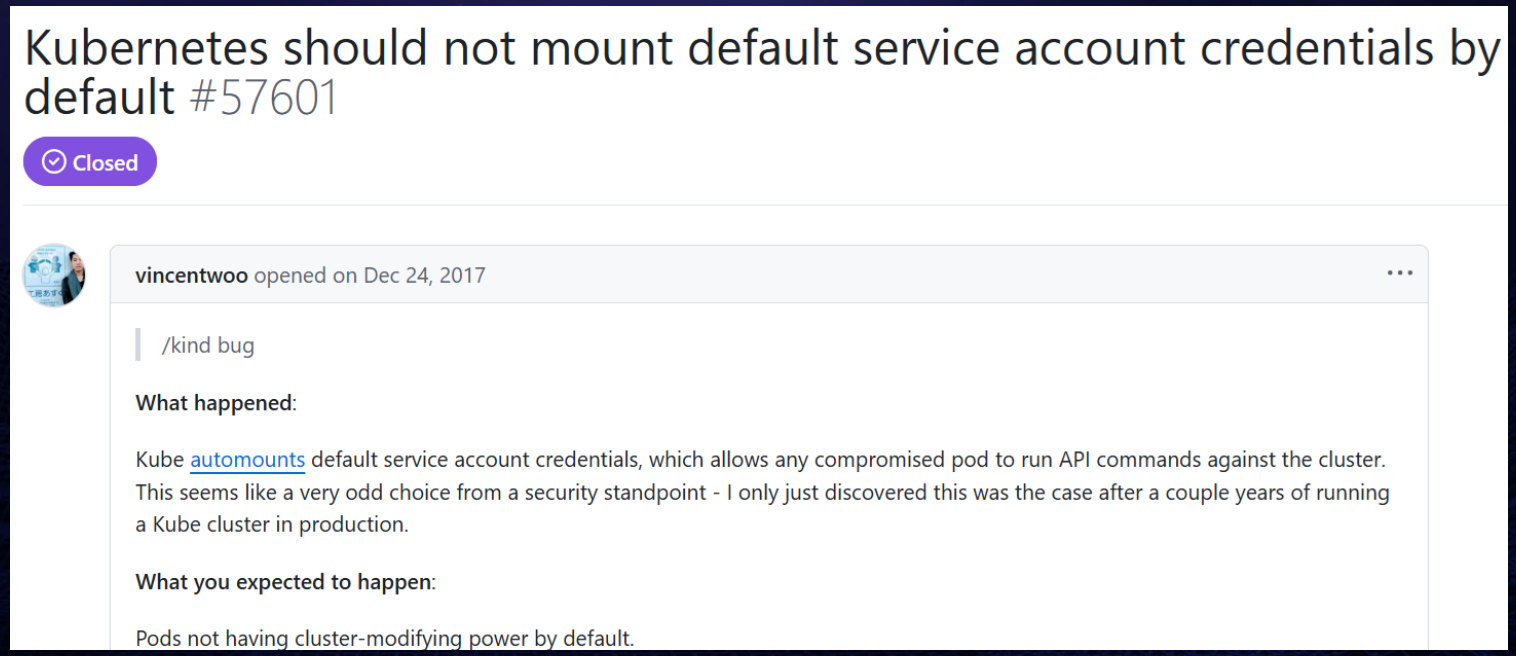
БЕКОН

Как написал, так и будет? Часть 3

Права и поведение Default Service Account

БЕКОН

- Неявно проставляется и выдаются права, если ничего не указать
  - Для 99.99% бизнес приложений не нужен
- До 1.6 (март 2017) мог иметь широкие привилегии в кластере
  - Пока RBAC не стал стандартным режимом авторизации (во времена ABAC)
- Отключать его монтирование нужно явно через automountServiceAccountToken: false



Kubernetes should not mount default service account credentials by default #57601

Closed

vincentwoo opened on Dec 24, 2017

/kind bug

What happened:

Kube automounts default service account credentials, which allows any compromised pod to run API commands against the cluster. This seems like a very odd choice from a security standpoint - I only just discovered this was the case after a couple years of running a Kube cluster in production.

What you expected to happen:

Pods not having cluster-modifying power by default.

* - обсуждение там идет и по сей день

Ссылка на issue *

БЕКОН

Выбор это всегда хорошо?

«Просто» проверить YAML?

БЕКОН

Как контролировать содержимое Pod?

- Pod Security Policies (PSP)
  - Признана устаревшей в v1.21 и удалена в v1.25
- Pod Security Admission (PSA)
  - Статус Alpha в v1.22, Beta в v1.23, GA в v1.25
- Pod Security Standards (PSS)
  - Набор рекомендаций, который может быть привязан к PSA
- Validating Admission Policy (VAP)
  - Статус Alpha в v1.26, Beta в v1.28, GA в v1.30
- Policy Engines
  - Сторонние решения: Kyverno, OPA Gatekeeper, JSPolicy, Kubewarden, ...
  - OPA Gatekeeper поддерживает VAP с 1.13
  - Kyverno поддерживает VAP с 1.11

БЕКОН

Меняются ли правила игры?

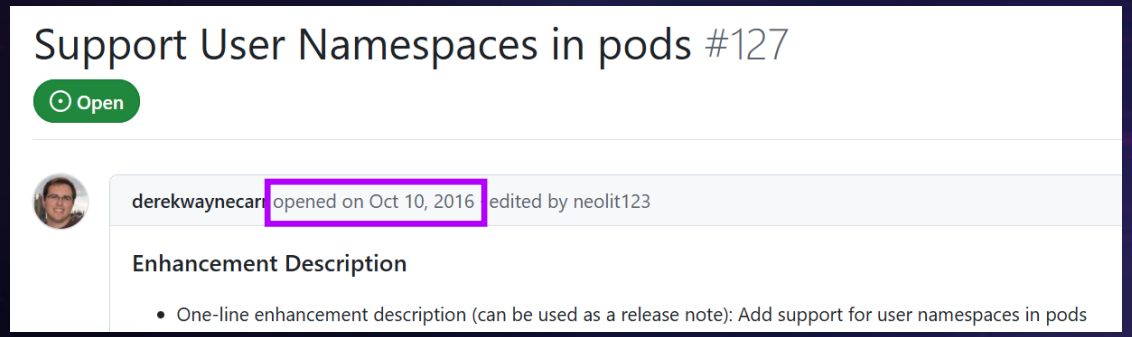
Новая эра hostUsers: false

БЕКОН

Root в контейнере, это root на Host?!

> Without using a user namespace a container running as root, in the case of a container breakout, has root privileges on the node. And if some capability were granted to the container, the capabilities are valid on the host too. None of this is true when we use user namespaces.

Ссылка на документацию



KEP-127

"Linux user namespace в чертогах Kubernetes", БеКон 2024

Kubernetes v1.33: User Namespaces enabled by default!

By **Rodrigo Campos Catelin (Microsoft), Giuseppe Scrivano (Red Hat), Sascha Grunert (Red Hat)** | Friday, April 25, 2025

In Kubernetes v1.33 support for user namespaces is enabled by default. This means that, when the stack requirements are met, pods can opt-in to use user namespaces. To use the feature there is no need to enable any Kubernetes feature flag anymore!

Ссылка на статью

БЕКОН

Меняются ли стандарты игры?

UserNamespacesPodSecurityStandards feature gate

БЕКОН

Не все так просто…

С v1.29 в статусе Alpha

For Linux Pods that enable user namespaces, Kubernetes relaxes the application of Pod Security Standards in a controlled way. This behavior can be controlled by the feature gate `UserNamespacesPodSecurityStandards`, which allows an early opt-in for end users. Admins have to ensure that user namespaces are enabled by all nodes within the cluster if using the feature gate.

If you enable the associated feature gate and create a Pod that uses user namespaces, the following fields won't be constrained even in contexts that enforce the *Baseline* or *Restricted* pod security standard. This behavior does not present a security concern because `root` inside a Pod with user namespaces actually refers to the user inside the container, that is never mapped to a privileged user on the host. Here's the list of fields that are **not** checks for Pods in those circumstances:

- `spec.securityContext.runAsNonRoot`
- `spec.containers[*].securityContext.runAsNonRoot`
- `spec.initContainers[*].securityContext.runAsNonRoot`
- `spec.ephemeralContainers[*].securityContext.runAsNonRoot`
- `spec.securityContext.runAsUser`
- `spec.containers[*].securityContext.runAsUser`
- `spec.initContainers[*].securityContext.runAsUser`
- `spec.ephemeralContainers[*].securityContext.runAsUser`

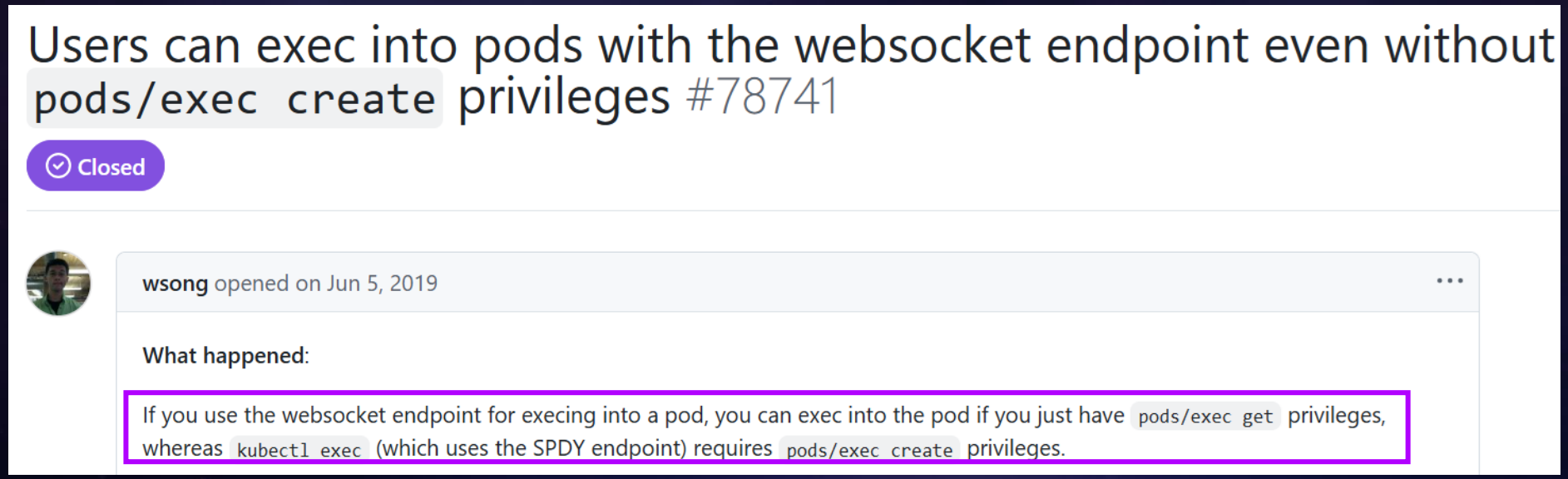
БЕКОН

Когда без бутылки уже не разобраться

Потемки RBAC

БЕКОН

На примере, exec (из свежего)



Users can exec into pods with the websocket endpoint even without `pods/exec create` privileges #78741

Closed

wsong opened on Jun 5, 2019

What happened:

If you use the websocket endpoint for execing into a pod, you can exec into the pod if you just have `pods/exec get` privileges, whereas `kubectl exec` (which uses the SPDY endpoint) requires `pods/exec create` privileges.

«Не такой очевидный RBAC Kubernetes», БеКон 2023

С версии Kubernetes 1.31 для операций attach, exec и port-forward теперь по умолчанию используется Websockets!

Подробнее в заметке "When is read-only not read-only?" от Rory McCune

БЕКОН

E – единообразие

«Просто» все залогируем

БЕКОН

Какие сложности с Kubernetes Audit Log?

Принцип написания

SOC FORUM 2024

Политику можно разбить на 3 блока:

1. Детализируем важное
2. Исключаем лишнее
3. Остальное в общем режиме

```yaml
apiVersion: audit.k8s.io/v1
kind: Policy
omitStages:
  - "RequestReceived"
rules:
  - level: RequestResponse
    resources:
    - group: ""
      resources: ["pods"]
  ...
  - level: None
    users: ["system:kube-proxy"]
    verbs: ["watch"]
    resources:
      - group: ""
        resources: ["endpoints", "services", "services/status"]
  ...
  - level: Metadate
```

Высокий уровень детализации
Отказ от логирования
Все остальное

"Kubernetes Audit Log в арсенале SOC", SOC Forum 2024

Ну и в придачу еще разное применение wildcards для ValidatingWebhookConfiguration, Roles, AuditPolicy

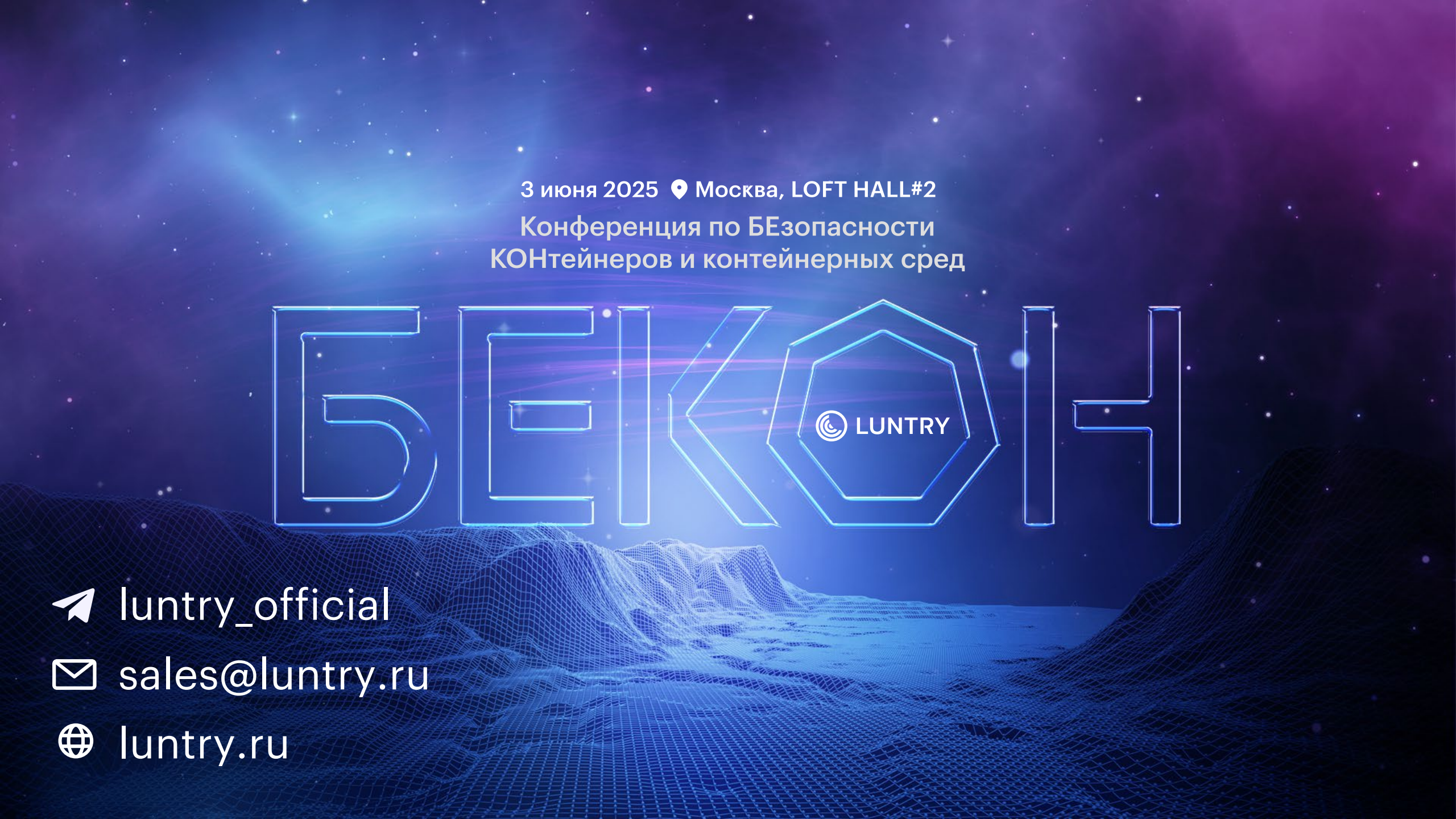
Заключение

БЕКОН

1. В Kubernetes инфраструктуре, как в любой многоуровневой системе, много неочевидных, неожиданных, непонятных, необъяснимых моментов

2. Контейнерные компоненты развиваются чрезвычайно быстрыми темпами, что приводит к большому разбросу и вариаций итоговых систем

3. Следить за всеми изменениями и, тем более, держать все в голове становится все сложнее


3 июня 2025 Москва, LOFT HALL#2
Конференция по БЕзопасности
КОНтейнеров и контейнерных сред
БЕКОН
LUNTRY
luntry_official
sales@luntry.ru
luntry.ru