What is a container escape?

A container escape is when an attacker breaks out of a container and reaches the host (or the Kubernetes node) it runs on, because the isolation boundary was weak or misconfigured. Once on the host, they control every container on that machine and can pivot further. Common causes are privileged containers, a mounted Docker socket, shared host namespaces, a host-path mount, or a dangerous capability like CAP_SYS_ADMIN. Preventing it means removing those over-permissions and keeping the kernel patched.

HOW IT WORKS

01 How escapes happen

Escapes almost always come from a misconfiguration that hands the container too much access, not an exotic kernel bug. The usual routes:

- A privileged container that can mount the host disk.
- A mounted Docker socket, which is full control of the daemon.
- Host namespace sharing (`--pid=host`, `--net=host`).
- A host-path mount exposing the host filesystem.
- A dangerous capability such as CAP_SYS_ADMIN abusing cgroups.
- An unpatched kernel vulnerability (the rarer case).

02 Why it matters in Kubernetes

In Kubernetes, escaping a pod onto its node is rarely the end. From the node the attacker can read every pod scheduled there, steal their service account tokens, and use those to talk to the API server and move across the whole cluster, then into the cloud account the cluster runs in.

That is why a single weak pod is a cluster-wide risk.

> ESCAPE = OWN THE HOST
>
> *A container escape turns one compromised app into control of the host and every container on it. In Kubernetes it is the first hop toward owning the whole cluster.*

HOW TO DEFEND

- Never run privileged containers or mount the Docker socket into workloads.
- Drop all capabilities and add back only what is needed; avoid CAP_SYS_ADMIN.
- Do not share host namespaces or mount sensitive host paths.
- Run as non-root, read-only root filesystem, with seccomp and AppArmor on.
- Enforce Pod Security Standards and admission control in Kubernetes.
- Patch the host kernel and test the cluster for real escape paths.

SOURCES

[1] MITRE ATT&CK: Containers Matrix

[2] NIST SP 800-190 Application Container Security Guide

Find the container escape paths before an attacker does.

securelayer7.net/learn/containers/what-is-a-container-escape

Open online