What is a privileged pod?

A privileged pod is a Kubernetes pod whose security context weakens isolation, most directly with `securityContext.privileged: true`, but also via `hostPID`/`hostNetwork`, host-path mounts, `allowPrivilegeEscalation`, or added capabilities. Such a pod can usually escape to its node, and from the node reach other pods and the control plane. Because any identity that can create pods can request a privileged one, privileged pods are both a direct escape and an RBAC escalation target. Pod Security Standards exist to block them.

HOW IT WORKS

01 How it is abused

Privileged pods are abused two ways:

- Direct escape: a workload that is already privileged is compromised, and the attacker escapes to the node (mount the host disk, abuse capabilities) then harvests other pods' tokens.
- RBAC escalation: an identity that can `create pods` but is otherwise limited requests a new privileged pod (or one mounting the node), schedules it, and uses it to break out, turning "can create pods" into "owns the node". See RBAC.

Documented techniques shown for defenders.

> CREATE-PODS CAN MEAN OWN-NODE
>
> *If RBAC lets an identity create pods without policy limits, it can launch a privileged pod and escape to the node. Pod creation plus no Pod Security Standards equals cluster escalation.*

HOW TO DEFEND

- Enforce Pod Security Standards (restricted) and admission control to reject privileged pods, host namespaces, and host mounts.
- Set `allowPrivilegeEscalation: false`, drop all capabilities, run as non-root, read-only root filesystem.
- Limit who can create pods and in which namespaces via RBAC.
- Audit running pods for privileged security contexts.
- Test the cluster for the create-pod-to-node-escape path.

SOURCES

[1] Kubernetes docs: Pod Security Standards

[2] MITRE ATT&CK: Containers Matrix

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

Find the container escape paths before an attacker does.

securelayer7.net/learn/containers/what-is-a-privileged-pod

Open online