What is Kubernetes security?

Kubernetes security is the practice of protecting a Kubernetes cluster, its control plane (API server, etcd, controllers) and its workloads (pods), from attack. The cluster is controlled through the API server, gated by RBAC, and every pod carries a service account token. Attackers target weak RBAC, an exposed kubelet, unauthenticated etcd, over-permissioned tokens, and pods configured to allow a container escape to the node. Hardening means least-privilege RBAC, Pod Security Standards, and locking the control-plane components.

HOW IT WORKS

01 The attack surface

The pieces attackers go after, each with its own page:

- RBAC: the permission system; too-broad roles let a small foothold do anything.
- Service account tokens: every pod gets one; an over-permissioned token is a key to the API.
- The kubelet: the per-node agent; if its API is exposed, attackers run commands in pods.
- etcd: the cluster database; unauthenticated access leaks every secret.
- Privileged pods: pods allowed to escape to the node.

02 The typical attack path

A common cluster compromise looks like: get code execution in a pod (via an app vulnerability), read its service account token from `/var/run/secrets/...`, query the API server to see what that token can do, abuse broad RBAC or a privileged pod to escape to the node, then harvest other pods' tokens and reach the control plane, and finally pivot into the cloud account the cluster runs in.

> POD TO CLUSTER TO CLOUD
>
> *Kubernetes attacks chain: one pod 'its token 'the API 'the node 'the control plane 'the cloud account. Every weak link shortens the path.*

HOW TO DEFEND

- Least-privilege RBAC: no wildcard roles, no cluster-admin for workloads.
- Limit service-account tokens: disable automount where unused, scope tightly.
- Lock the kubelet: authenticated and authorized, never anonymous.
- Secure etcd: mutual TLS, never reachable unauthenticated, secrets encrypted at rest.
- Enforce Pod Security Standards and admission control; no privileged pods.
- Test the cluster for the real pod-to-cluster path.

SOURCES

[1] MITRE ATT&CK: Containers Matrix

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

[3] Kubernetes docs: Kubernetes security concepts

Find the container escape paths before an attacker does.

securelayer7.net/learn/containers/what-is-kubernetes-security

Open online