

CONTAINER & KUBERNETES SECURITY.



Containers ship in minutes and multiply across clusters just as fast — **a single vulnerable image or misconfigured pod can spread the exposure everywhere it's deployed** before anyone notices.

- IMAGE SCANNING
- RUNTIME PROTECTION
- ORCHESTRATION HARDENING
- SECRETS MANAGEMENT

01 THE MECHANISM

Image Scanning

Inspecting container images for **known vulnerabilities** before they reach a registry.

Runtime Protection

Monitoring running containers for **behavior that deviates from the expected baseline**.

Orchestration Hardening

Locking down cluster configuration so a **compromised pod can't reach the rest of the environment**.

02 THE FLOW

The Build Cycle

CONTINUOUS

- 1 A container image is built and **scanned for vulnerabilities and misconfigurations**.
- 2 The image is signed and pushed to **a trusted, access-controlled registry**.
- 3 The orchestrator deploys it under **policies that limit privilege and network access**.

The Breakout Cycle

LIVE EVENT

- 1 An attacker exploits a vulnerable image or **misconfigured container privilege**.
- 2 Runtime monitoring flags behavior that **deviates from the container's expected baseline**.
- 3 The pod is isolated, and the image is patched **before redeployment across the cluster**.

03 CORE PILLARS & COVERAGE

PILLAR	WHAT IT COVERS
Image Scanning	Identifies known vulnerabilities and misconfigurations before deployment.
Runtime Protection	Detects anomalous container behavior as workloads execute.
Orchestration Hardening	Restricts cluster privileges, network paths, and API access.
Secrets Management	Keeps credentials and keys out of images and environment variables.

04 CONTAINER SECURITY APPROACHES COMPARED

SIGNAL	SUFFICIENT ALONE?	WHAT IT MISSES
Base Image Trust Alone	No	Doesn't catch vulnerabilities introduced after the base layer.
Perimeter Network Firewall	No	Doesn't see pod-to-pod traffic inside the cluster.
Manual Registry Review	No	Doesn't scale and misses newly disclosed vulnerabilities.
Default Cluster Configuration	No	Ships with broad permissions that widen the attack surface.
Container Security (Scanning + Runtime + Hardening)	Yes	Secures the image, the workload, and the cluster together.

CONTAINER & KUBERNETES SECURITY.

SCAN · PROTECT · HARDEN

05 PROGRAM OPERATIONS

Image sprawl outpaces manual review.

New images ship faster than any human team can vet them.

Default configurations are rarely secure.

Clusters ship permissive by default and need explicit hardening.

Secrets end up in images more than expected.

Hardcoded credentials slip through without dedicated scanning.

Runtime drift is the norm, not the exception.

Containers change behavior across updates and need continuous baselining.

06 CONTROL MAPPING

NIST SP 800-190

Application container security guidance covering the full container lifecycle.

CIS BENCHMARKS

Configuration hardening baselines for Docker and Kubernetes.

PCI DSS

Segmentation requirements applied to containerized cardholder data environments.

ISO 27001

Technical vulnerability management extended to container images and registries.

07 WHAT THIS DOES NOT SOLVE

Application logic flaws

Container security hardens the environment; it doesn't fix flaws in the application code itself.

Supply chain trust

A scanned image is only as safe as the base layers and dependencies it was built from.

Cluster sprawl

Multiple clusters across teams create inconsistent policy enforcement without central governance.

Human misconfiguration

Tooling reduces risk but doesn't prevent a manual override that reopens the exposure.

08 MYTHS VS. REALITY

MYTH	REALITY
"Containers are isolated by default."	Shared kernels mean a container escape can reach the host.
MYTH	REALITY
"Scanning once at build time is enough."	New vulnerabilities are disclosed daily against images already deployed.
MYTH	REALITY
"Kubernetes is secure out of the box."	Default settings favor flexibility over restriction, not security.
MYTH	REALITY
"Small images are automatically safer."	Size doesn't determine risk; unpatched dependencies do.
MYTH	REALITY
"Env-var secrets are safe enough."	They're visible to anyone with pod-level access.

09 SPEAK THE PROGRAM — KEY TERMS

CONTAINER

A packaged unit of software bundled with everything it needs to run.

IMAGE SCANNING

Checking a container image for known vulnerabilities before deployment.

KUBERNETES

An orchestration platform that schedules, scales, and manages containers.

POD

The smallest deployable unit in Kubernetes, holding one or more containers.

RUNTIME PROTECTION

Monitoring running containers for anomalous or malicious behavior.

CONTAINER ESCAPE

An attack that breaks out of a container to access the underlying host.