Red Hat

Trustee + Sandboxed Containers

Attack Surface Analysis

Trust Boundaries

In the Confidential Containers model, the trust boundary is fundamentally different from traditional Kubernetes deployments:

Entity	Trust Status
Workload Owner	**TRUSTED** Controls the workload, holds encryption keys
TEE Hardware	**TRUSTED** Hardware root of trust (Intel TDX, AMD SEV-SNP)
Guest VM (CVM)	**TRUSTED** Attested enclave running kata-agent, CDH, AA
Trustee/KBS	**TRUSTED** Key broker, runs in IBM SE/hyperprotect
Host OS/Hypervisor	**UNTRUSTED** Can observe/modify control plane, but not TEE memory
OpenShift Admin	**UNTRUSTED** Has cluster access but should not access workload secrets
Kata Runtime/Shim	**UNTRUSTED** Runs on host, communicates with kata-agent via ttRPC

Attack Vectors & Mitigations

The following attack vectors are relevant to confidential container deployments.

(a) oc exec / oc rsh / terminal access

Threat: A bad actor with OpenShift administrator privileges uses `oc exec` or `oc rsh` to gain shell access inside a confidential pod, potentially exfiltrating secrets.

Layer	Mitigation
Kata Agent Policy	Block `ExecProcessRequest` in agent policy. Set `default ExecProcessRequest := false`
OCP RBAC	Remove pods/exec verb from roles.
Attestation	Policy hash is bound to TEE evidence. Trustee verifies before releasing secrets.

(b) Container Log Access

Threat: Sensitive data accidentally logged by the application could be viewed via oc logs.

Layer	Mitigation
Kata Agent Policy	Block `ReadStreamRequest` and `WriteStreamRequest` for stdout/stderr
Application	Never log secrets. Use structured logging with redaction.

(c) Unauthorized Secret Retrieval via CDH

Threat: If exec were allowed, an attacker could curl the Confidential Data Hub to retrieve secrets: curl http://127.0.0.1:8006/cdh/resource/...

Layer	Mitigation
Kata Agent Policy	Block `ExecProcessRequest` entirely
Trustee Policy	Use resource policies to restrict which secrets are released based on attestation claims
InitData Binding	Bind secrets to specific `InitData` hashes so that only expected workloads receive keys

(d) Container Image Tampering

Threat: Malicious container image substitution or layer injection.

Layer	Mitigation
Kata Agent Policy	Enforce `allowed_images` list in `CreateContainerRequest` policy
Image Signature	Use Sigstore/cosign with signature verification in image-rs
Encrypted Images	Encrypt container images; decryption key released only after attestation