SAPIENT

SECURITY INSIGHTS

Security Audit Report

Files Scanned: ["sample-k8s-large-vuln.yaml"]

Date Generated: 01-10-2025

Kubernetes Security Audit (Risk Analyst)

Risk Synopsis

Risk Rating: High (CVSS Score: 9.0)

Rationale: The Kubernetes manifest contains multiple high-risk vulnerabilities, including the use of plaintext secrets, overly permissive RBAC roles, and privileged containers. These vulnerabilities can lead to unauthorized access, data breaches, and potential full control over the Kubernetes cluster. The presence of insecure configurations, such as hostPath mounts and running containers as root, significantly increases the attack surface and potential impact on business operations.

Compliance Mapping Table

Finding ID	Description	Severity	Affected Controls
1	Plaintext secret storing sensitive data	High	PCI-DSS 3.2.1, NIST AC-17
2	ConfigMap containing sensitive information	High	PCI-DSS 3.2.1, NIST AC-17
3	ServiceAccount with tokens automount enabled	High	NIST AC-3, CIS 5.1.1
4	ClusterRole granting cluster-admin privileges	High	NIST AC-6, SOC2 CC6.1
5	Privileged container with hostPath mount and running as root	High	NIST AC-5, CIS 5.4.5
6	Deployment using insecure image tag and no resource limits	Medium	NIST CM-2, PCI-DSS 3.2.1

7	DaemonSet with hostPath logs and no SELinux context	Medium	NIST AC-4, CIS 5.4.5
8	StatefulSet using emptyDir for persistent storage	Medium	NIST CP-9, PCI-DSS 3.2.1
9	Service exposing pods via NodePort to the internet	High	PCI-DSS 1.3.5, NIST AC-17
10	Ingress with insecure backend and missing TLS	High	PCI-DSS 4.1.1, NIST SC-12
11	PodDisruptionBudget with minAvailable too low	Medium	NIST CP-2, CIS 5.2.1
12	HorizontalPodAutoscaler with permissive minReplicas	Medium	NIST CM-2, SOC2 CC6.1
13	Role allowing exec/port-forward (too permissive)	High	NIST AC-6, SOC2 CC6.1
14	CronJob that mounts hostPath and runs as root periodically	High	NIST AC-5, CIS 5.4.5
15	ResourceQuota set too low without resource requests	Medium	NIST CM-2, PCI-DSS 3.2.1
16	LimitRange with permissive defaults	Medium	NIST CM-2, PCI-DSS 3.2.1
17	NetworkPolicy allowing all ingress/egress traffic	High	NIST AC-4, CIS 5.2.1
18	Pod with hostIPC true (can read IPC on host)	High	NIST AC-4, CIS 5.4.5

OWASP & Framework Alignment

- **OWASP Kubernetes Top Ten:**
    - **K8s-002:** Insecure Secrets Management (Finding ID 1, 2)
    - **K8s-003:** Insecure Network Policies (Finding ID 17)
    - **K8s-004:** Privileged Containers (Finding ID 5, 6)
    - **K8s-005:** Insecure RBAC (Finding ID 4, 13)
- **Compliance Frameworks:**
    - **CIS Kubernetes Benchmark:** Multiple findings align with CIS controls, particularly around RBAC and network policies.

  - **NIST SP 800-53:** Findings map to several controls, including AC-3 (Access Enforcement) and AC-6 (Least Privilege).

Threat Model Context

- **Attack Vector:** An attacker could exploit the plaintext secrets and overly permissive RBAC roles to gain unauthorized access to sensitive data and resources. Privileged containers can be leveraged to escalate privileges and execute arbitrary commands on the host.
- **Attacker Type:** External malicious actors or internal threat actors with limited access.
- **Potential Business Impact:** Data breaches, service disruptions, and loss of customer trust. The organization may face regulatory penalties due to non-compliance with security standards.

Residual Risk & Exceptions

- **Residual Risks:** The use of plaintext secrets and privileged containers poses significant risks that cannot be fully mitigated without substantial changes to the deployment.
- **Areas Requiring Formal Risk Exceptions:** The use of cluster-admin roles and privileged containers may require formal risk acceptance due to operational needs.
- **Mitigating Controls:** Implementing stricter RBAC policies, using encrypted secrets, and enforcing security contexts can help mitigate risks.