Kubernetes static code analysis

Unique rules to find Security Hotspots in your KUBERNETES code

All rules 7 Security Hotspot 6 Code Smell 1

Tags

Search by name...

Mounting sensitive file system paths is security-sensitive
Security Hotspot

Using host operating system namespaces is security-sensitive
Security Hotspot

Allowing process privilege escalations is security-sensitive
Security Hotspot

Exposing Docker sockets is security-sensitive
Security Hotspot

Running containers in privileged mode is security-sensitive
Security Hotspot

Setting capabilities is security-sensitive
Security Hotspot

Kubernetes parsing failure
Code Smell

Mounting sensitive file system paths is security-sensitive

Analyze your code

Security Hotspot Major cwe

Mounting sensitive file system paths can lead to information disclosure and compromise of the host systems.

System paths can contain sensitive information like configuration files or cache files. Those might be used by attackers to expand permissions or to collect information for further attacks. System paths can also contain binaries and scripts that might be executed by the host system periodically. A compromised or rogue container with access to sensitive files could endanger the integrity of the whole Kubernetes cluster.

Ask Yourself Whether

- The mounted file path contains sensitive information.
- The mounted file path contains configuration files or executables that are writable.
- The Pod is untrusted or might contain vulnerabilities.

There is a risk if you answered yes to any of those questions.

Recommended Secure Coding Practices

It is recommended to avoid mounting sensitive system file paths into containers. If it is necessary to mount such a path due to the architecture, the least privileges should be given, for instance by making the mount read-only to prevent unwanted modifications.

Sensitive Code Example

```
apiVersion: v1
kind: Pod
metadata:
  name: test
spec:
  containers:
  - image: k8s.gcr.io/test-webserver
    name: test-container
    volumeMounts:
    - mountPath: /data
      name: test-volume
  volumes:
  - name: test-volume
    hostPath:
      path: /etc # Sensitive
```

Compliant Solution

```
apiVersion: v1
kind: Pod
metadata:
  name: test
spec:
  containers:
  - image: k8s.gcr.io/test-webserver
    name: test-container
    volumeMounts:
      - mountPath: /data
        name: test-volume
  volumes:
  - name: test-volume
    hostPath:
      path: /mnt/nfs
```

See

- Kubernetes Documentation - Volumes
- MITRE, CWE-668 - Exposure of Resource to Wrong Sphere

Available In:

sonarcloud | sonarqube