Secrets
ABAP
Apex
C
C++
CloudFormation
COBOL
C#
CSS
Flex
Go
HTML
Java
JavaScript
Kotlin
Kubernetes
Objective C
PHP
PL/I
PL/SQL
Python
RPG
Ruby
Scala
Swift
Terraform
Text
TypeScript
T-SQL
VB.NET
VB6
XML

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

Exposing Docker sockets is security-sensitive

Analyze your code

Security Hotspot Major cwe

Exposing Docker sockets can lead to compromise of the host systems.

The Docker daemon provides an API to access its functionality, for example through a UNIX domain socket. Mounting the Docker socket into a container allows the container to control the Docker daemon of the host system, resulting in full access over the whole system. A compromised or rogue container with access to the Docker socket could endanger the integrity of the whole Kubernetes cluster.

Ask Yourself Whether

- 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 never add a Docker socket as a volume to a Pod.

Sensitive Code Example

```
apiVersion: v1
kind: Pod
metadata:
  name: test
spec:
  containers:
  - image: k8s.gcr.io/test-webserver
    name: test-container
    volumeMounts:
    - mountPath: /var/run/docker.sock
      name: test-volume
  volumes:
  - name: test-volume
    hostPath:
      path: /var/run/docker.sock # Sensitive
      type: Socket
```

Compliant Solution

```
apiVersion: v1
kind: Pod
metadata:
  name: test
spec:
  containers:
  - image: k8s.gcr.io/test-webserver
    name: test-container
```

See

- Kubernetes Documentation - Volumes
- Docker Documention - Daemon socket option
- MITRE, CWE-284 - Improper Access Control

Available In:

sonarcloud | sonarqube