Pod Security Admission

Die Kubernetes Pod Security Admission (PSA) ist eine im Core von Kubernetes integrierte Admission-Controller-Strategie, um die Sicherheit von Pods zu gewährleisten. Sie ersetzt den früheren `PodSecurityPolicy` Mechanismus, der ab Kubernetes v1.25 als veraltet gilt. Mit Pod Security Admission lassen sich für Namespaces einfache, deklarative Sicherheitsstandards festlegen, die direkt im Namespace-Objekt hinterlegt werden.

Funktionsweise

Der Pod Security Admission Controller überprüft beim Erstellen oder Aktualisieren eines Pods, ob dessen Spezifikation den im Namespace konfigurierten Sicherheitsstufen entspricht. Pods, die restriktive Einstellungen unterlaufen, werden abgewiesen oder mit Warnungen versehen.

Es stehen drei Sicherheitsprofile zur Verfügung:

- **privileged**: minimale Einschränkungen (für vertrauenswürdige Workloads)
- **baseline**: Standard-Sicherheitsfunktionen, empfohlen für allgemeine Anwendungen
- **restricted**: strengste Einstellungen, empfohlen für Produktionsumgebungen

Konfiguration

Die Sicherheitsprofile werden auf Namespace-Ebene über Labels definiert. Jedes Profil kann drei Modi haben:

- `enforce`: Verhindert die Erstellung unzulässiger Pods.
- `audit`: Schreibt Verstöße ins Audit-Log, blockiert den Vorgang aber nicht.
- `warn`: Gibt eine Warnung zurück, aber erstellt trotzdem den Pod.

Beispiel: Namespace-Labelling

```
kubectl label namespace <namespace-name> \
    pod-security.kubernetes.io/enforce=baseline \
    pod-security.kubernetes.io/enforce-version=v1.33 \
    pod-security.kubernetes.io/warn=restricted \
    pod-security.kubernetes.io/warn-version=v1.33
```

Mit diesem Beispiel:

- Werden nur Pods akzeptiert, die das Profil `baseline` erfüllen.
- Bei Verstoß gegen `restricted` gibt es eine Warnung.

Profile im Detail

1. **Privileged**
   - Kaum Einschränkungen.
   - Erlaubt z. B. privilegierte Container, Host-Networking, HostPath Volumes.
2. **Baseline**
   - Verbietet die meisten eskalierenden Features.
   - Erlaubt, was für viele Anwendungen benötigt wird, wie z. B. einige Capabilities.
3. **Restricted**
   - Strikte Isolation des Pods.
   - Kein privilegierter Zugriff, keine systemkritischen Capabilities.

Mehr zu den genauen Profile-Einstellungen in der offiziellen Dokumentation.

Referenzen

- Kubernetes Documentation: Pod Security Admission
- Security Standards: Baseline & Restricted

Hinweis: Für eine produktive Umgebung sollte die Konfiguration regelmäßig geprüft und mit weiteren Maßnahmen (z. B. Role-based Access Control, Image Policies) kombiniert werden.

Umsetzung im produktiven Cluster / Pod-Sicherheitsstandards auf Clusterebene

Es werden die folgenden Pod-Sicherheitsstandards auf die Version latest angewendet:

- baseline standard in enforce mode
- restricted standard in warn and audit mode

Der baselinePod Security Standard bietet einen praktischen Mittelweg, der es ermöglicht, die Ausnahmeliste kurz zu halten und bekannte Rechteausweitungen zu verhindern. Um außerdem zu verhindern, dass Pods in

- kube-system
- calico-system

fehlschlagen, werden diese Namespaces von der Anwendung der Pod-Sicherheitsstandards ausgenommen.

Bei der Implementierung von Pod Security Admission in der produktiven Umgebung sollte Folgendes beachtet werden:

- Abhängig von der Risikobewertung eines Clusters könnte ein strengerer Pod-Sicherheitsstandard die bessere Wahl sein. Beispielsweise `restricted`

Für die Clusterweite Anwendung wird der Admission Controller konfiguriert

```
apiVersion: apiserver.config.k8s.io/v1
kind: AdmissionConfiguration
plugins:
- name: PodSecurity
  configuration:
    apiVersion: pod-security.admission.config.k8s.io/v1 # see compatibility
note
    kind: PodSecurityConfiguration
    # Defaults applied when a mode label is not set.
    #
    # Level label values must be one of:
    # - "privileged" (default)
    # - "baseline"
    # - "restricted"
    #
    # Version label values must be one of:
    # - "latest" (default)
    # - specific version like "v1.33"
    defaults:
      enforce: "baseline"
      enforce-version: "latest"
      audit: "restricted"
      audit-version: "latest"
      warn: "restricted"
      warn-version: "latest"
    exemptions:
      # Array of authenticated usernames to exempt.
      usernames: []
      # Array of runtime class names to exempt.
      runtimeClasses: []
      # Array of namespaces to exempt.
      namespaces: [kube-system,calico-system,longhorn-system,tigera-
operator,trident]
```

Container und Container Image Security

trivy

Trivy has scanners that look for security issues, and targets where it can find those issues.

Targets (what Trivy can scan):

- Container Image
- Filesystem
- Git Repository (remote)

- Virtual Machine Image
- Kubernetes
- AWS

Scanners (what Trivy can find there):

- OS packages and software dependencies in use (SBOM)
- Known vulnerabilities (CVEs)
- IaC issues and misconfigurations
- Sensitive information and secrets
- Software licenses

kube-bench

kube-bench is a tool that checks whether Kubernetes is deployed securely by running the checks documented in the CIS Kubernetes Benchmark.

CIS Scanning as part of Trivy and the Trivy Operator

Trivy, the all in one cloud native security scanner, can be deployed as a Kubernetes Operator inside a cluster. Both, the Trivy CLI, and the Trivy Operator support CIS Kubernetes Benchmark scanning among several other features.