KUBERNETES NATIVES POLICY MANAGEMENT MIT KYVERNO

Max Körbächer | Co-Founder Liquid Reply

REPLY
LIQUID

VORSTELLUNG

Max Körbächer | Co-Founder Liquid Reply

Sr. Manager Cloud Native Engineer, mit dem Fokus auf:

- Platform Engineering
- Application Delivery
- Cloud Native Advisory

Teil des Kubernetes Release Team & Release Engineering Team

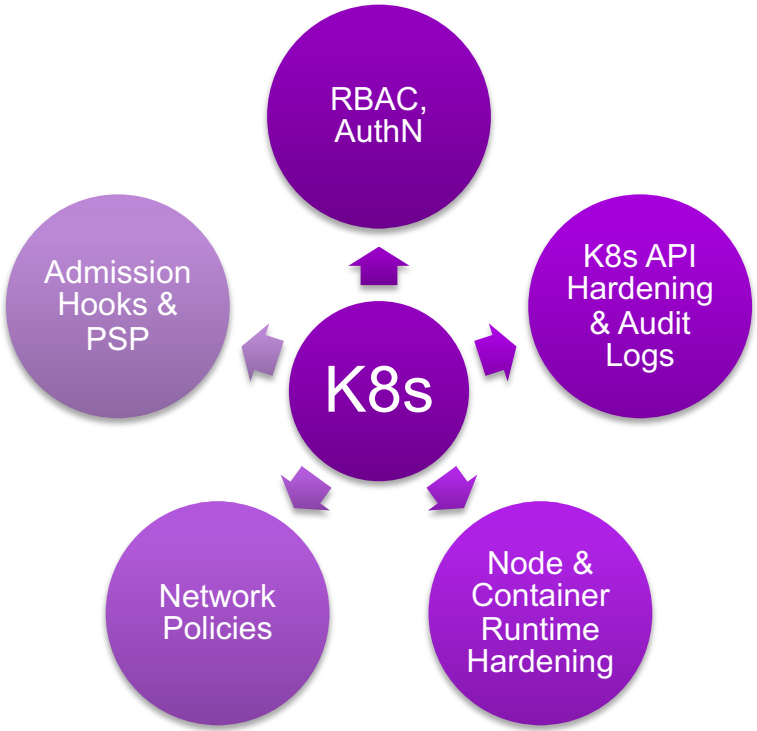
Wöchentlicher Cloud Native Newsletter: nativecloud.dev

KUBERNETES SECURITY

KERNELEMENTE ZUR ABSICHERUNG VON K8S

Security Relevant vor dem Cluster

- Code
- Container Hardening
- Build & Deployment



Security Relevant um/auf dem Cluster

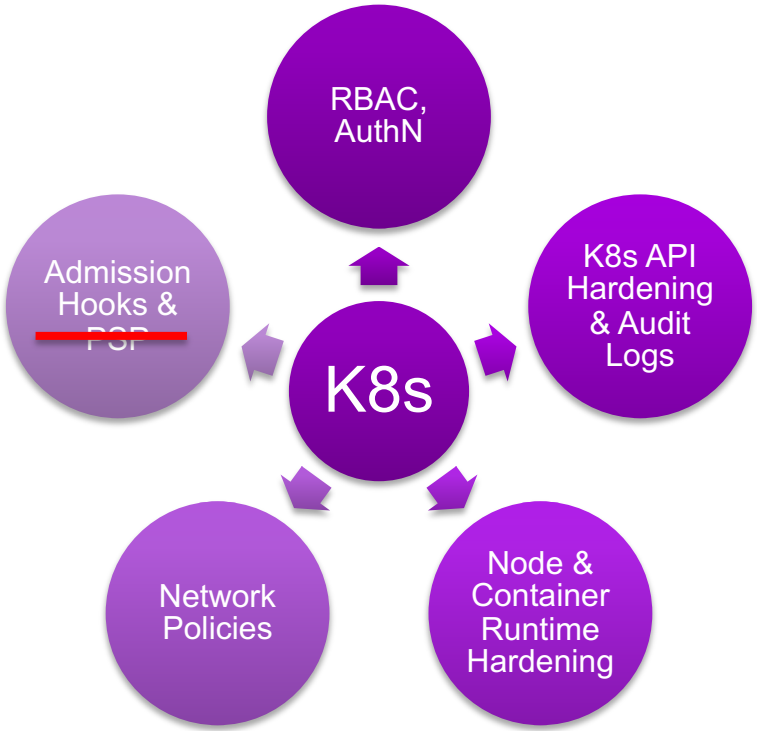
- Intrusion Detection
- Policy Engines
- CVE Scans
- Cluster Conformance

KUBERNETES SECURITY

KERNELEMENTE ZUR ABSICHERUNG VON K8S

Security Relevant vor dem Cluster

- Code
- Container Hardening
- Build & Deployment



Security Relevant um/auf dem Cluster

- Intrusion Detection
- Policy Engines
- CVE Scans
- Cluster Conformance

DEPRICATION DER POD SECURITY POLICIES

K8s wird alle 4 Monate released und treibt kontinuierlich Veränderungen an

Eine alternative ist zwar in der Entwicklung, die Community schaut jetzt jedoch verstärkt auf K8s Erweiterungen – Policy Engines, wodurch deren Relevanz deutlich zugenommen hat

POLICY ENGINES/MANAGEMENT

Was ist eine Policy Engine?

- Eine Erweiterung von K8s die ein Regelwerk entgegen nimmt und auf dem Cluster durchsetzt
- Diese basieren zum Teil auf die K8s Ressource „policy“ – Limit Ranges, Resource Quotas, Pod Security Policies, Process ID Limits & Reservations und Node Resource Managers

Ein Beispiel:

- **Regel**: Kein Pod darf mit einem Root User starten
- **Auswirkung**: Das Deployment des Pods wird zwar im K8s definiert, jedoch wird kein Container gestartet der nicht explizit über einen Security Context Root User ausgeschlossen hat

EINE ALTERNATIVE ZU OPA GATEKEEPER ODER K-RAIL

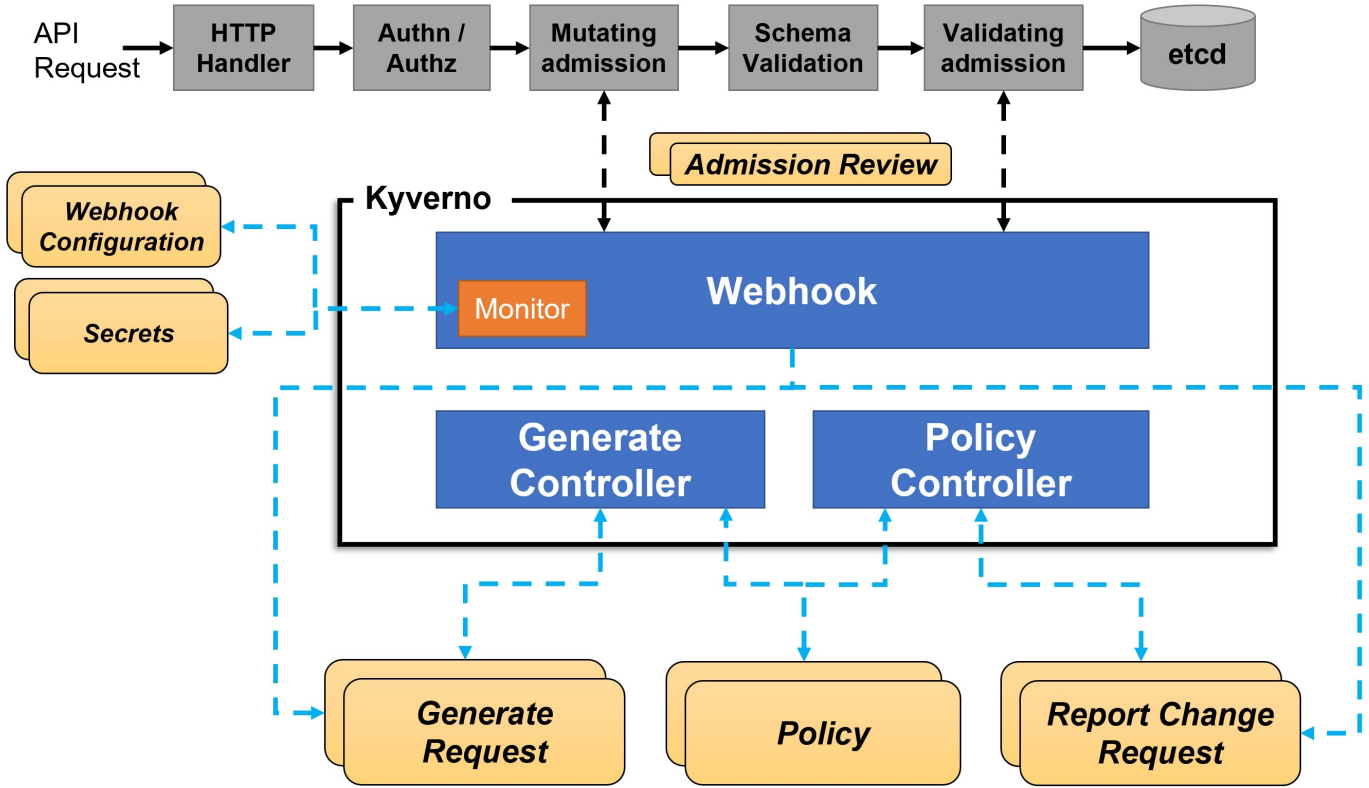
Kernfunktionalitäten:

- Validieren – Kommt der Container von einer Registry die ich kenne?
- Mutieren – Jeder Pod brauch ein label „security: super-safe“
- Generieren – Ein neuer Namespace ist angelegt, dazu brauch es noch eine neue Netzwerk Policy
- OpenAPI Validierungsschema (kubectl explain) - Was ist … ?

Was differenziert Kyverno von anderen Policy Engines?

- ➢ Kein Coding, Regeln sind Deklarativ
- ➢ Nur für Kubernetes
- ➢ Kann andere Ressourcen erzeugen
- ➢ Keine hoch Komplexe Anfragen

WIE ARBEITET SICH KYVERNO?



WO KANN EINE POLICY ENGINE INTERVENIEREN?

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Impact
Using Cloud credentials	Exec into container	Backdoor container	Privileged container	Clear container logs	List K8S secrets	Access the K8S API server	Access cloud resources	Images from a private registry	Data Destruction
Compromised images in registry	bash/cmd inside container	Writable hostPath mount	Cluster-admin binding	Delete K8S events	Mount service principal	Access Kubelet API	Container service account		Resource Hijacking
Kubeconfig file	New container	Kubernetes CronJob	hostPath mount	Pod / container name similarity	Access container service account	Network mapping	Cluster internal networking		Denial of service
Application vulnerability	Application exploit (RCE)	Malicious admission controller	Access cloud resources	Connect from Proxy server	Applications credentials in configuration files	Access Kubernetes dashboard	Applications credentials in configuration files		
Exposed Dashboard	SSH server running inside container				Access managed identity credential	Instance Metadata API	Writable volume mounts on the host		
Exposed sensitive interfaces	Sidecar injection				Malicious admission controller		Access Kubernetes dashboard		
							Access tiller endpoint		
							CoreDNS poisoning		
							ARP poisoning and IP spoofing		

= New technique

= Deprecated technique

Source: https://cloud.redhat.com/blog/2021-kubernetes-threat-matrix-updates-things-you-should-know

WO KANN EINE POLICY ENGINE INTERVENIEREN?

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Impact
Using Cloud credentials	Exec into container	Backdoor container	Privileged container	Clear container logs	List K8S secrets	Access the K8S API server	Access cloud resources	Images from a private registry	Data Destruction
Compromised images in registry	bash/cmd inside container	Writable hostPath mount	Cluster-admin binding	Delete K8S events	Mount service principal	Access Kubelet API	Container service account		Resource Hijacking
Kubeconfig file	New container	Kubernetes CronJob	hostPath mount	Pod / container name similarity	Access container service account	Network mapping	Cluster internal networking		Denial of service
Application vulnerability	Application exploit (RCE)	Malicious admission controller	Access cloud resources	Connect from Proxy server	Applications credentials in configuration files	Access Kubernetes dashboard	Applications credentials in configuration files		
Exposed Dashboard	SSH server running inside container				Access managed identity credential	Instance Metadata API	Writable volume mounts on the host		
Exposed sensitive interfaces	Sidecar injection				Malicious admission controller		Access Kubernetes dashboard		
							Access tiller endpoint		
							CoreDNS poisoning		
							ARP poisoning and IP spoofing		

= New technique

= Deprecated technique

Source: https://cloud.redhat.com/blog/2021-kubernetes-threat-matrix-updates-things-you-should-know

RESSOURCEN VALIDIEREN

DEMO

Durch Validierende Regeln überprüft Kyverno die Einhaltung und verbietet bei Bedarf das ausführen von Container.

Die sogenannte Baseline definiert ein Minimum an Security relevanten Konfigurationen.

- Keine Host Namespaces, Privilegierten Container, Host Path, Host Ports
- Verbieten von Kernal Capabilities wie NET_ADMIN
- Erlaubt nur Non Root, Non Root Groups und ohne privilegierte Eskalation

RESSOURCEN MUTIEREN

DEMO

Mutierende Regeln können die gegebenen Ressource anpassen.

Da ich nicht jedes einzelne Chart anpassen möchte um den Security Context umzusetzen kann das Kyverno gleich mit machen.

Im Enterprise Kontext sind mutierende Regeln neben dem Umsetzen von Security Context auch extrem nützlich um die Adresse der Container Registry anzupassen.

Dadurch kann zusätzlich verhindert werden, dass Images aus dem Internet oder einer unbekannten CR bezogen werden.

RESSOURCEN GENERIEREN

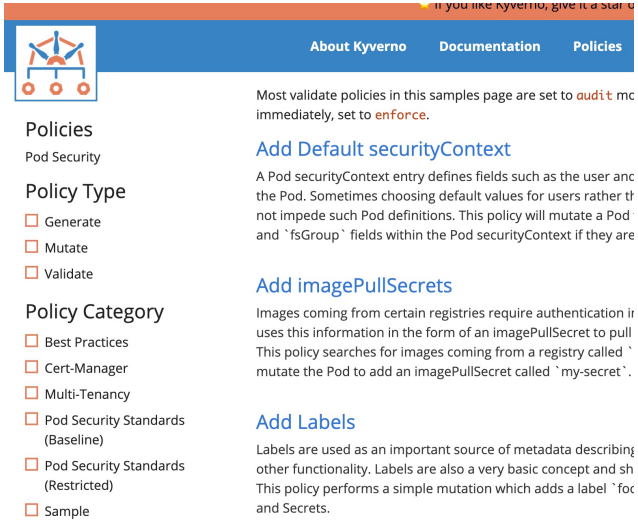
DEMO

Die Regeln reagieren auf neue Ressourcen und erstellen dazu bspw. neue Network Policies

Bei großen, komplexen Kubernetes Umgebungen hilft dies auch neue Ressourcen abzudecken ohne das komplizierte CICD Pipelines benötigt werden.

KYVERNO POLICY LIBRARY

EINE SAMMLUNG AN BEST PRACTICES UND BEISPIELEN



https://kyverno.io/policies

KYVERNO & POLICY ENGINES

SIND NUR SO GUT WIE DIE GESAMTE KONFIGURATION

Policy Engines wirken sich direkt oder indirekt auf fast alle Bedrohungen aus.

Wichtig ist aber auch ein gutes RBAC, feine Netzwerk Policies und ein gehärtetes Grundsystem (Nodes, OCI, Infrastruktur et al)

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Impact
Using Cloud credentials	Exec into container	Backdoor container	Privileged container	Clear container logs	List K8S secrets	Access the K8S API server	Access cloud resources	Images from a private registry	Data Destruction
Compromised images in registry	bash/cmd inside container	Writable hostPath mount	Cluster-admin binding	Delete K8S events	Mount service principal	Access Kubelet API	Container service account		Resource Hijacking
Kubeconfig file	New container	Kubernetes CronJob	hostPath mount	Pod / container name similarity	Access container service account	Network mapping	Cluster internal networking		Denial of service
Application vulnerability	Application exploit (RCE)	Malicious admission controller	Access cloud resources	Connect from Proxy server	Applications credentials in configuration files	Access Kubernetes dashboard	Applications credentials in configuration files		
Exposed Dashboard	SSH server running inside container				Access managed identity credential	Instance Metadata API	Writable volume mounts on the host		
Exposed sensitive interfaces	Sidecar injection				Malicious admission controller		Access Kubernetes dashboard		
							Access tiller endpoint		
							CoreDNS poisoning		
							ARP poisoning and IP spoofing		

= New technique

= Deprecated technique

DEMO

Alle gezeigten Beispiele finden sich unter

https://github.com/Liquid-Reply/talk-ressources