Kubernetes Security
How to secure your Kubernetes platform

Johannes M. Scheuermann

Karlsruhe, 09.12.2017
Johannes M. Scheuermann
IT Engineering & Operations @ inovex

› Software-Defined Datacentres
› Infrastructure as Code
› Cloud technologies
› High Availability & Scalability
› Want‘s a Z-Frame
› @johscheuer
Security
Security Kubernetes

- Security Cake
- Authentication and Authorisation
- Network Policies
- Pod Security Policies
- Admission Controller
- Vault Integration
Security Cake

- App
- Base
- Container

Platform

- OS
- HW
Secure the Pipeline

Build → Test → Ship → Run
Focus on the Platform

- Secure all components with certificates
- Only talk over TLS
- Disable the insecure port (even on localhost)
- Disable anonymous authentication
  - Or at least restrict it to uncritical resources
- Since 1.7 Kubernetes supports an Audit Log
How to build "the" platform
Authentication and Authorisation

- RBAC (Role Based Access Control)
- ABAC (Attribute Based Access Control)
- WebHook
- Certificates
- Token (JWT \(\rightarrow\) https://jwt.io)
RBAC (example)
Network Policies
Network Policies

- Since 1.6
- Ingress policies
- Egress policies (1.8)
- Network segmentation (distributed firewall)
- CNI plugin must support it
Network Policies (example)

- DENY all traffic to an application
- LIMIT traffic to an application
- DENY all non-whitelisted traffic in a namespace
- DENY all traffic from other namespaces
- ALLOW traffic from other namespaces
- ALLOW traffic from external clients
PodSecurityPolicies
Pod Security Policies

- Needs to be explicitly activated
- Let you define what’s allowed
- There must be a default policy
- Activating “runAsNonRoot” will break many things
- Only activate if needed (multi-tenant)
- Can be combined with RBAC
Pod Security Policies (example)

- Volumes
- RunAsUser
- AllowedCapabilities
- Privileged
- HostNetwork / HostPorts
- readOnlyRootFilesystem
Admission Controller(s)
Admission Controller(s)

- Intercepts request to the Kubernetes API
- (Can) Perform modifications
- Many default controllers exists
- You can also write your own
Admission Controller (example)

- DenyEscalatingExec
- ImagePolicyWebhook
- NodeRestriction
- PodSecurityPolicy
- SecurityContextDeny
- ServiceAccount
Vault Integration (CA)

1.) Auth (AppRoleID)
2.) Issue certificate
Vault Integration (secret store)

- Since 0.9 Kubernetes auth backend
- Solves only the challenge of authentication
- Secret must be fetched
  - Sidecar/init container
- Integrates with ServiceAccounts
Grafeas

- Open artifact metadata API
- Pluggable (multiple providers)
- ACL for the metadata
- Query-ability
- Integrates with Kubernetes
Istio

- Service Mesh
- Policy Enforcement (L4/L7)
- Integrated CA
- Transparent TLS
- Routing
- Language/Platform agnostic
Conclusions
Conclusions

- Many possibilities to make your cluster more secure
- Each with it’s benefits and drawbacks
- Generally → Security means (hard) work
- Depends on your Use Case what to use
- Play around with and get a feeling (in a playground)
We are hiring!

Become an inovexpert

www.inovexperts.com
Further reading

• https://kubernetes.io/docs/admin/admission-controllers
• https://kubernetes.io/docs/concepts/policy/pod-security-policy
• https://kubernetes.io/docs/concepts/services-networking/network-policies
Further reading

- https://grafeas.io
- https://istio.io
- https://ahmet.im/blog/kubernetes-network-policy
- https://github.com/kubernetes/examples/tree/master/staging/podsecuritypolicy/rbac