AVOLENS

CLOUD NATIVE COMPUTING FOUNDATION

KUBERNETES SECURITY

AVOLENS

CLOUD NATIVE COMPUTING FOUNDATION

$WHOAMI

- Benjamin Koltermann
- CEO of AVOLENS
- Cloud/Kubernetes Security Engineer
- CTF player @fluxfingers
- Regular at SIG-Security
- @p4ck3t0 on Twitter

AVOLENS

CLOUD NATIVE COMPUTING FOUNDATION

AGENDA

- Security 101
- RBAC
- Admission Controller
- Network Policies
- Runtime Security
- Isolation
- Encryption
- Sample Security Architecture

AVOLENS

CLOUD NATIVE COMPUTING FOUNDATION

SECURITY 101

PUBLIC ATTACK SURFACE

Master Ports

Protocol	Direction	Port Range	Purpose	Used By
TCP	Inbound	6443	Kubernetes API server	All
TCP	Inbound	2379-2380	etcd server client API	kube-apiserver, etcd
TCP	Inbound	10250	Kubelet API	Self, Control plane
TCP	Inbound	10259	kube-scheduler	Self
TCP	Inbound	10257	kube-controller-manager	Self

AVOLENS

CLOUD NATIVE COMPUTING FOUNDATION

SECURITY 101

PUBLIC ATTACK SURFACE

Worker Ports

Protocol	Direction	Port Range	Purpose	Used By
TCP	Inbound	10250	Kubelet API	Self, Control plane
TCP	Inbound	30000-32767	NodePort Services†	All

SECURITY 101

PUBLIC ATTACK SURFACE

- Cluster Architecture is important (Public/Private Cluster)
- Kubernetes has public Accessable https endpoints, that can be accessed without authentication
  - /version
  - /healthz
  - /livez
- Your exposed Apps

AVOLENS

CLOUD NATIVE COMPUTING FOUNDATION

SECURITY 101

CONTAINER IMAGE SCANNING

- Scan container images to find vulnerabilities

DEMO1

SECURITY 101

POD SECURITY CONTEXT

- Security Contexts are used to set the security settings for a Pod
  - runAsUser
  - runAsGroup
  - fsGroup
  - fsGroupChangePolicy
  - seLinuxOptions
  - supplementalGroups
  - runAsNonRoot
  - seccompProfile
  - sysctls
  - windowsOptions

SECURITY 101

CONTAINER SECURITY CONTEXT

- Security Contexts are used to set the security settings for a Container
  - runAsUser
  - runAsGroup
  - readOnlyRootFilesystem
  - allowPrivilegeEscalation
  - privileged
  - capabilities
  - seLinuxOptions
  - procMount
  - windowsOptions

SECURITY 101

OTHER DANGEROUS MANIFEST FIELDS

- Host Process (windowsOptions.hostProcess)
- Host Namespaces (hostNetwork, hostPID, hostIPC)
- Host Ports (hostPort)
- Host Path Volume (hostPath)
- Service externalIPs

DEMO2

SECURITY 101

SERVICE ACCOUNTS

- Every Pod has a Service Account Token mounted at `/var/run/secrets/kubernetes.io/serviceaccount/token`
- Service Account Tokens are used to authenticate against the API

LEAST PRIVILEGE PRINCIPLE!

DEMO3

RBAC

ROLES AND CLUSTERROLES

- Roles are namespaced
- ClusterRoles are cluster wide

RBAC

ROLEBINDINGS AND CLUSTERROLEBINDINGS

- Rolebindings are namespaced
- ClusterRolebindings are cluster wide

RBAC

PERMISSIONS

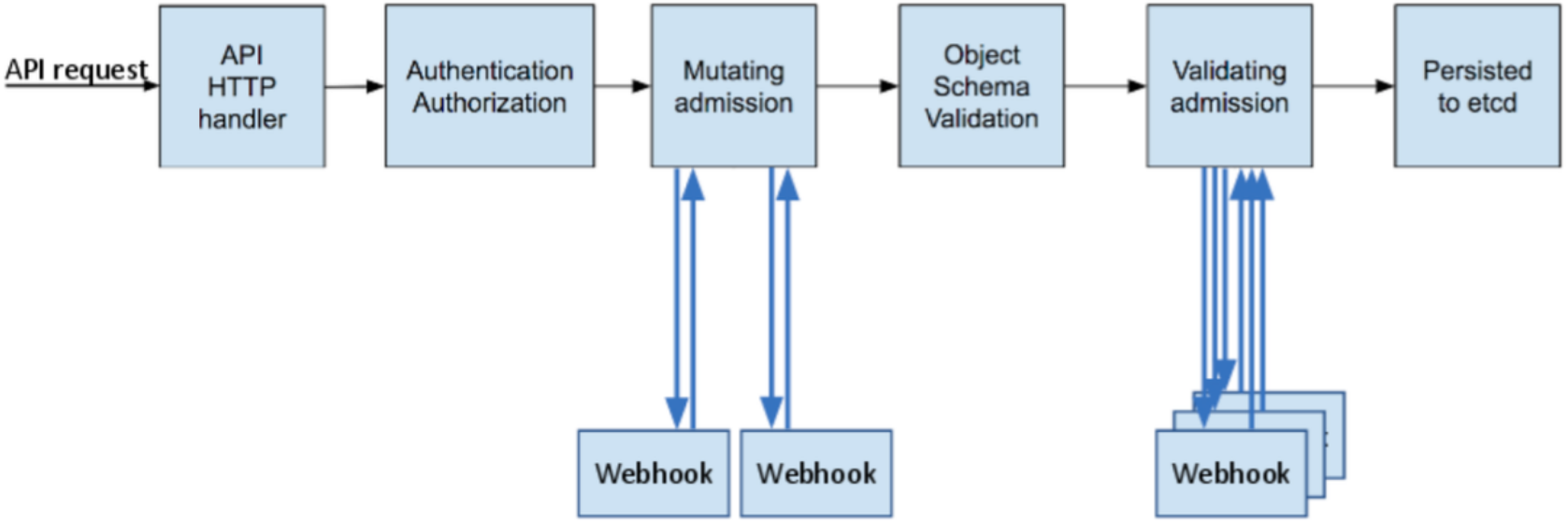
- Limit access to API resources
- Limit access to API verbs
- Limit access to API resource fields

DEMO4

ADMISSION CONTROLLERS

- Bouncer for the Kubernetes API
- Rejects dangerous manifests
- Ensures manifests conform to a certain standards or compliance
- Mutating Admission Controllers
- Validating Admission Controllers

ADMISSION CONTROLLERS



ADMISSION CONTROLLERS

- Several built-in Admission Controllers
- PodSecurity Admission (PSA) together with PodSecurity Standards (PSS)
- ValidatingAdmssionPolicies
- Third Party Admission Controllers

AVOLENS

CLOUD NATIVE COMPUTING FOUNDATION

ADMISSION CONTROLLERS

POD SECURITY STANDARDS

- Privileged
- Baseline
- Restricted

ADMISSION CONTROLLERS

POD SECURITY ADMISSION

- Warn
- Audit
- Enforce

DEMO5

NETWORK POLICIES

- Firewall in Kubernetes
- Ingress and Egress
- Namespaced
- Enforced by the CNI
- Some CNI's implement Network Policies differently

AVOLENS

CLOUD NATIVE COMPUTING FOUNDATION

NETWORK POLICIES

Network Policy Editor

https://editor.networkpolicy.io/

DEMO6

AVOLENS

CLOUD NATIVE COMPUTING FOUNDATION

RUNTIME SECURITY

- Detect and prevent malicious activity at Runtime after the Pod has started
  - `/etc/passwd`
  - Backconnecting shells
  - Maleware
  - Malicous Executables

ISOLATION

- Namespaces are no isolation! They are logical separation of resources!
- Isolation is based on Namespaces

DEMO7

ISOLATION

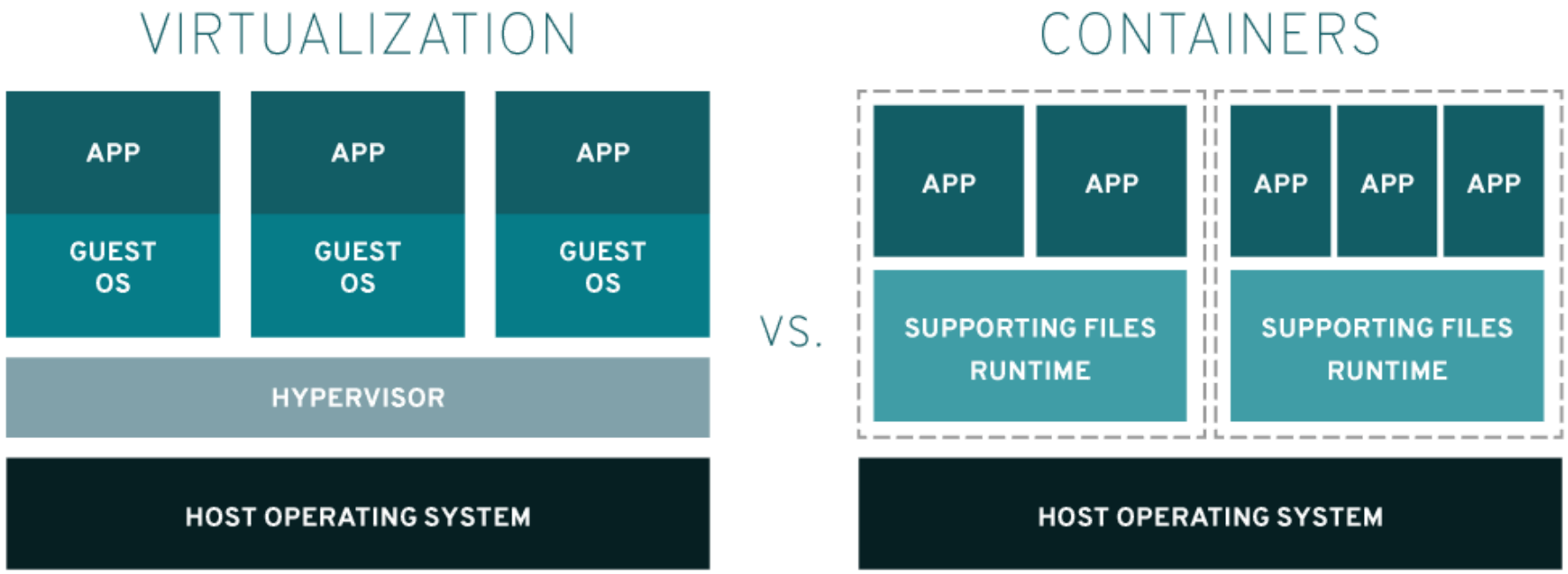
- Network Isolation => Network Policies
- Ressource Limitation => Resource Quotas
- Ressource/Process Isolation => Container Runtime
- Pod Isolation => Label/Taints that Pods are scheduled on different Nodes

AVOLENS

CLOUD NATIVE COMPUTING FOUNDATION

ISOLATION

CONTAINER VS VMS



ISOLATION

CONTAINER RUNTIME

- Linux Namespaces
- Cgroups
- Capabilities

CONTAINERD, CRI-O, DOCKER, ...

But what if thats not enough?

ISOLATION

CONTAINER RUNTIME

- VM based Container Runtime
  - Kata Containers
  - Firecracker
- Confidential Computing
  - Confidential Containers

ENCRYPTION

What can be encrypted in Kubernetes?

- Secrets
- Persistent Volumes
- Etcd
- Network Communication

ENCRYPTION

SECRETS

- By default Base64 encoded
- Stored in etcd
- Secret Vault should be used

DEMO8

ENCRYPTION

PERSISTENT VOLUMES

- Container Storage Interface (CSI) needs to support encryption

ENCRYPTION

ETCD

- Etcd is unencrypted by default
- Encryptionconfiguration is used to encrypt etcd

DEMO9

ENCRYPTION

NETWORK COMMUNICATION

- mTLS
- Service Mesh should be used

MAYBE DEMO 10, IF WE HAVE TIME

SAMPLE SECURITY ARCHITECTURE

- Private Cluster, API only Accessible via VPN or through a Bastion Host
- Container Images are regulary scanned
- One Service Account per Deployment/DaemonSet/StatefulSet
- RBAC to limit access to the API, least privilege principle
- Secret Vault for Secrets
- Network Policies to isolate Workloads
- Encrypted Persistent Volumes for Workloads
- Service Mesh to encrypt Network Communication
- Encryptionconfiguration to encrypt etcd
- Admission Controllers to enforce compliance
- Runtime Security to detect malicious activity

AVOLENS

CLOUD NATIVE COMPUTING FOUNDATION

THANKS FOR LISTENING