MINIO

Kubernetes (IN)Security

Lenin Alevski

About Me

Lenin Alevski

Security Software Engineer

Open Source @MinIO

Corporate & Startup world

Obsessive ❤ cybersecurity

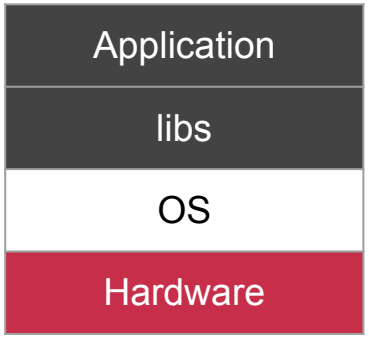
@Alevsk /in/alevsk/ lenin@alevsk.com

Agenda

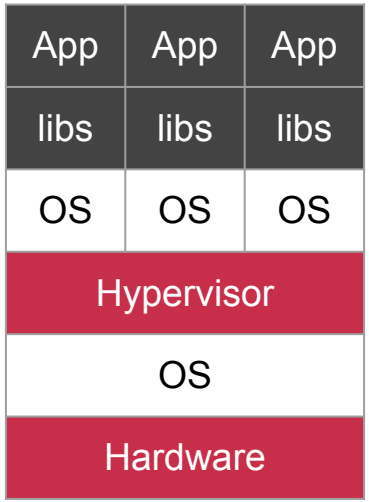
- Introduction to Containers
- Introduction to Kubernetes
- Most common attack techniques in K8S
- K8s build-in defenses
- Bonus: Kubernetes local CTF challenge

Application compartmentalization

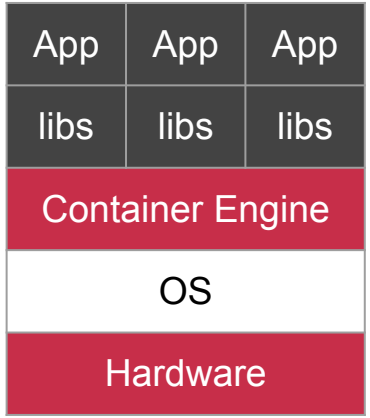
Traditional Architecture



VM Architecture

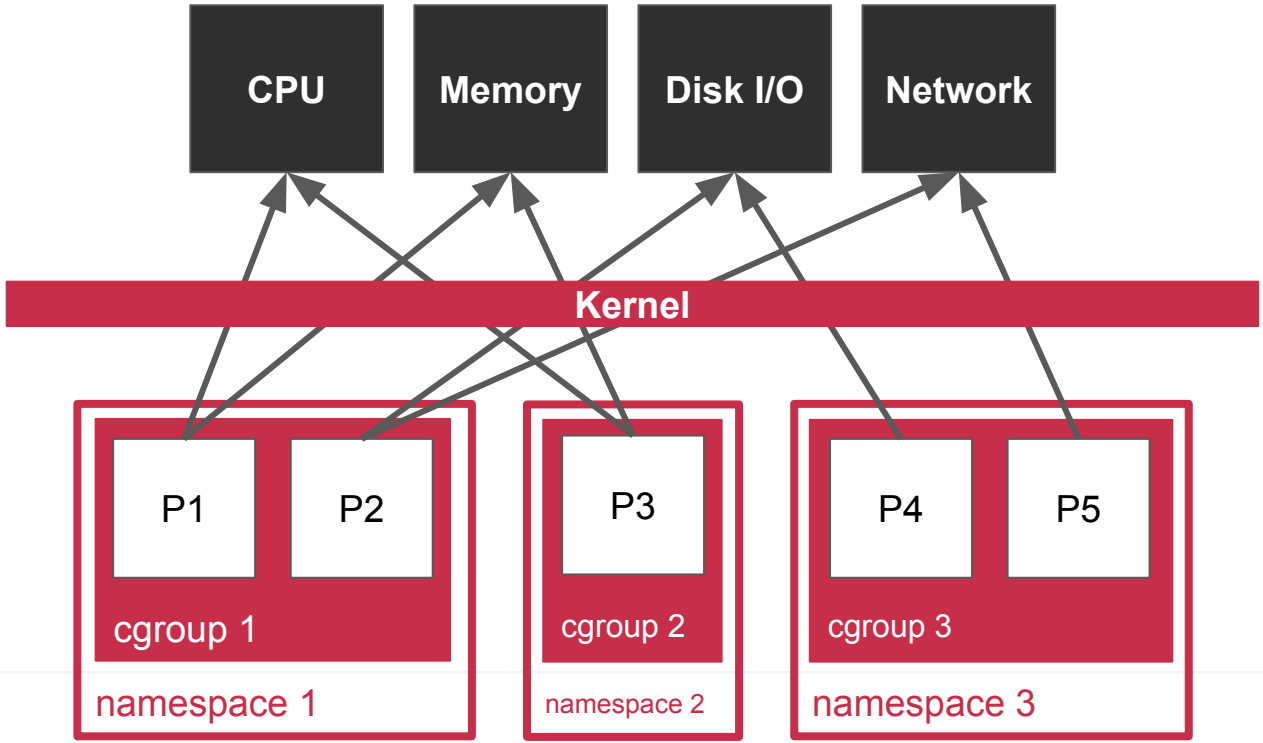


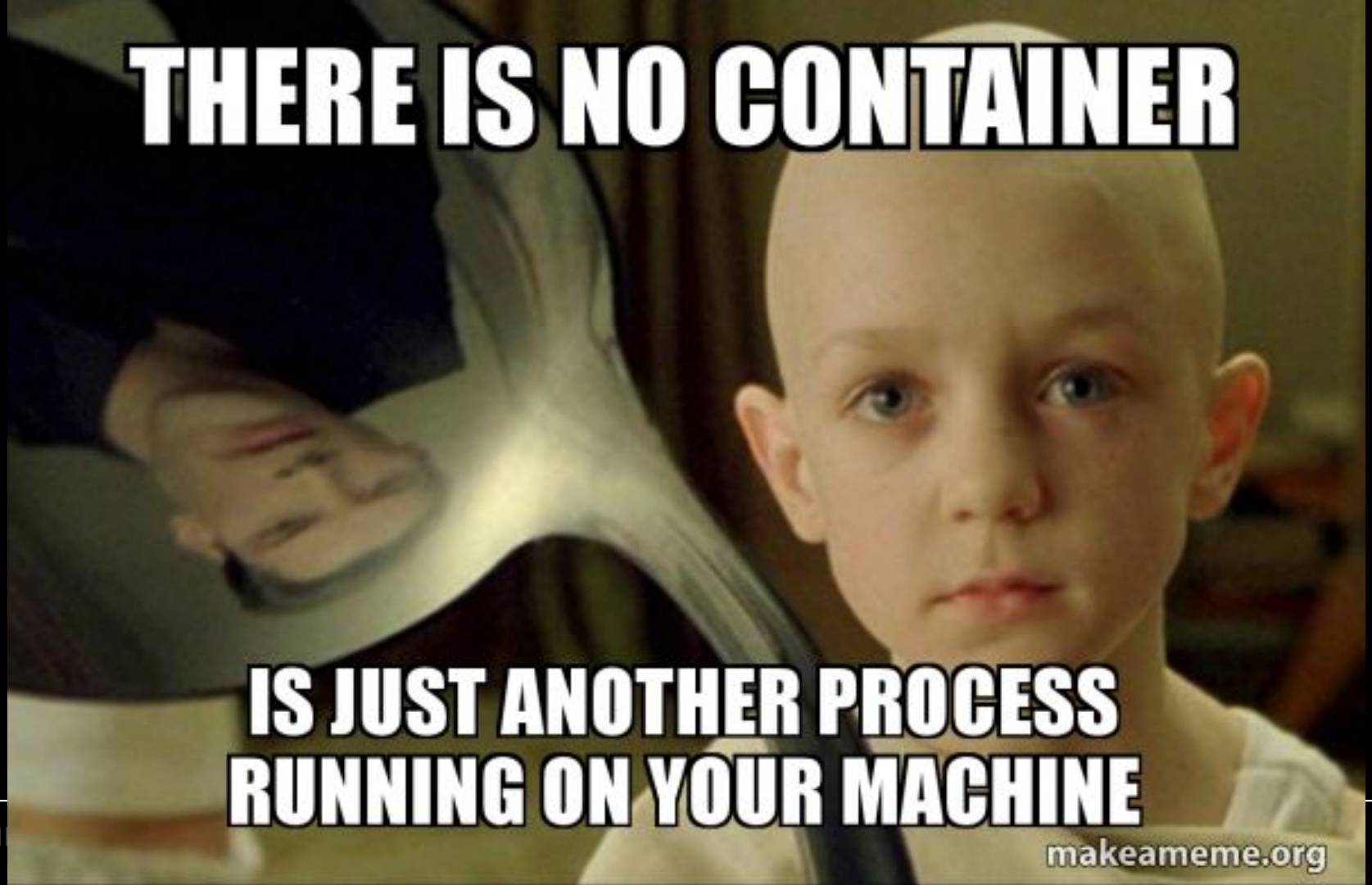
Container Architecture



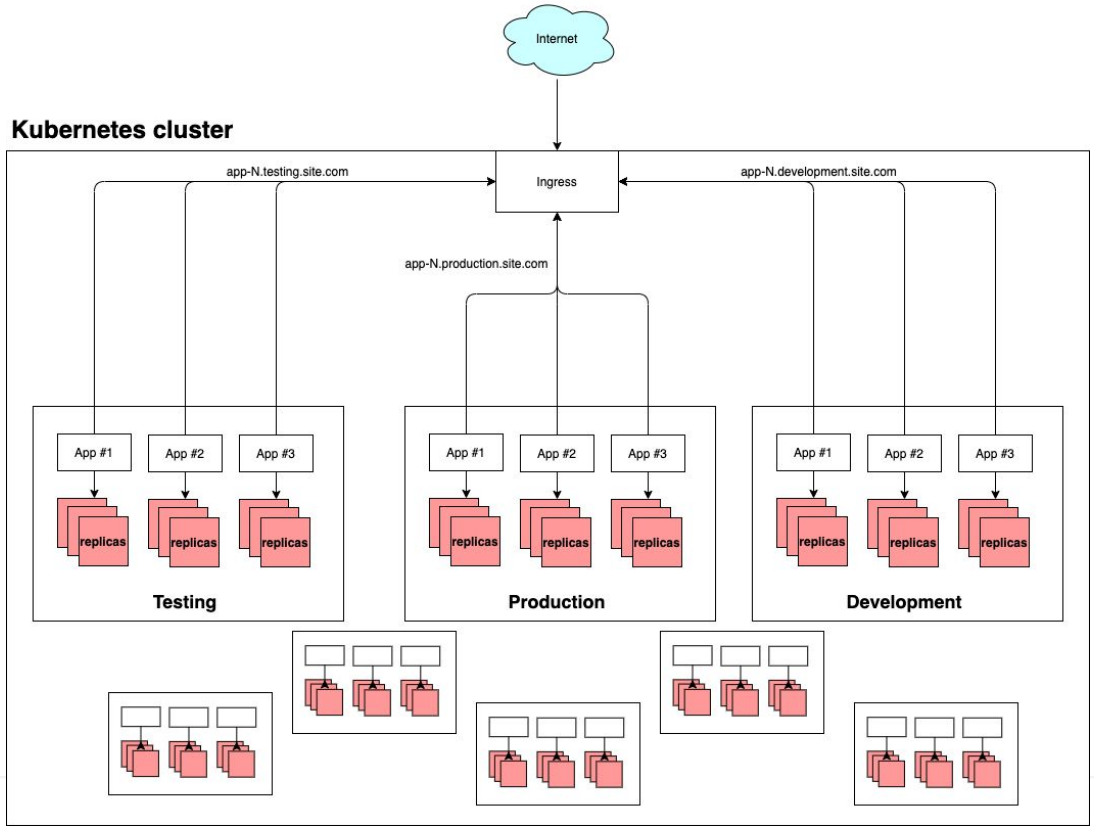
So, what containers really are?

- Namespaces
- Cgroups
- Capabilities



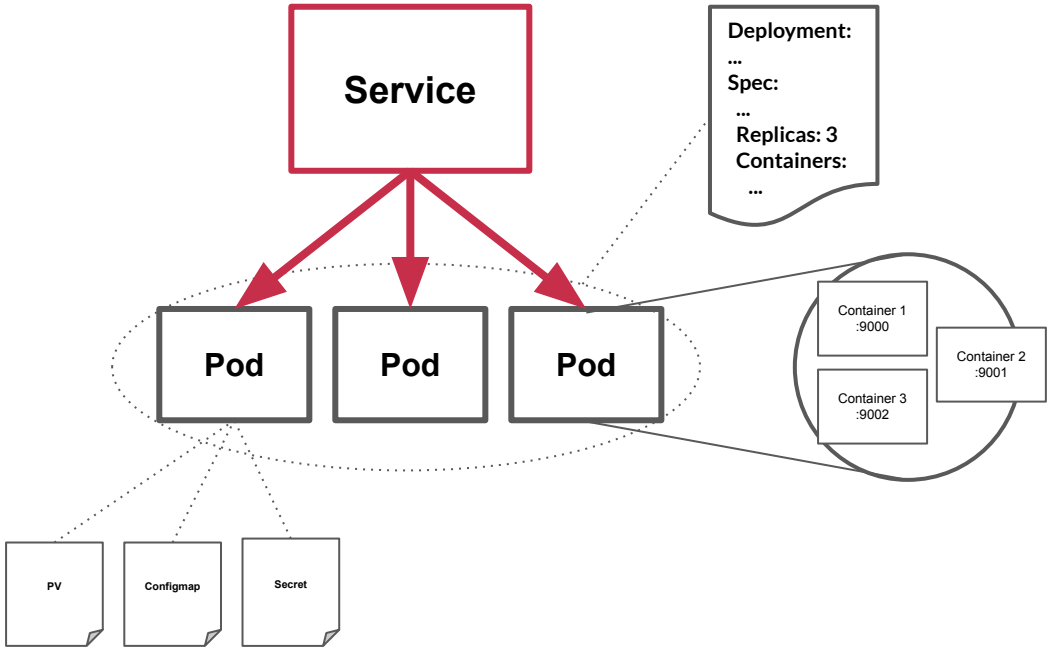

THERE IS NO CONTAINER
IS JUST ANOTHER PROCESS
RUNNING ON YOUR MACHINE
makeameme.org

What is kubernetes anyways?



Kubernetes primitives

- Service
- Deployment
- Pod
- StatefulSet
- Configmap
- Secret
- ...
- ..
- Many more



Kubernetes Components

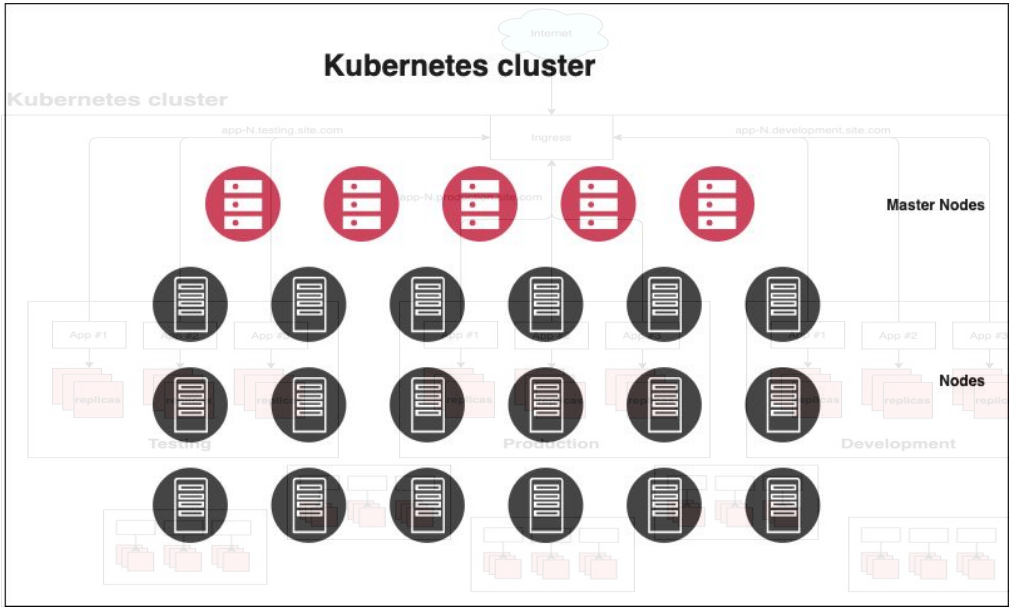
Kubernetes components

Control Plane Components

- kube-apiserver
- etcd
- kube-scheduler
- kube-controller

Node Components

- kube-proxy
- kubelet
- container runtime



Kubernetes most common attack techniques

Threat matrix for Kubernetes

https://www.microsoft.com/security/blog/2020/04/02/attack-matrix-kubernetes/

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	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	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)		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					Instance Metadata API	Writable volume mounts on the host	
							Access Kubernetes dashboard	
							Access tiller endpoint	

Updated threat matrix for Kubernetes

https://www.microsoft.com/security/blog/2021/03/23/secure-containerized-environments-with-updated-threat-matrix-for-kubernetes/

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

Initial Access

Amazon EKS

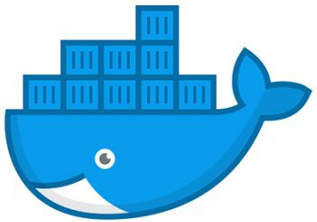
- Using cloud credentials
- Compromised images and registry
- Kubeconfig file
- Application Vulnerability
- Exposed sensitive interfaces

dockerhub

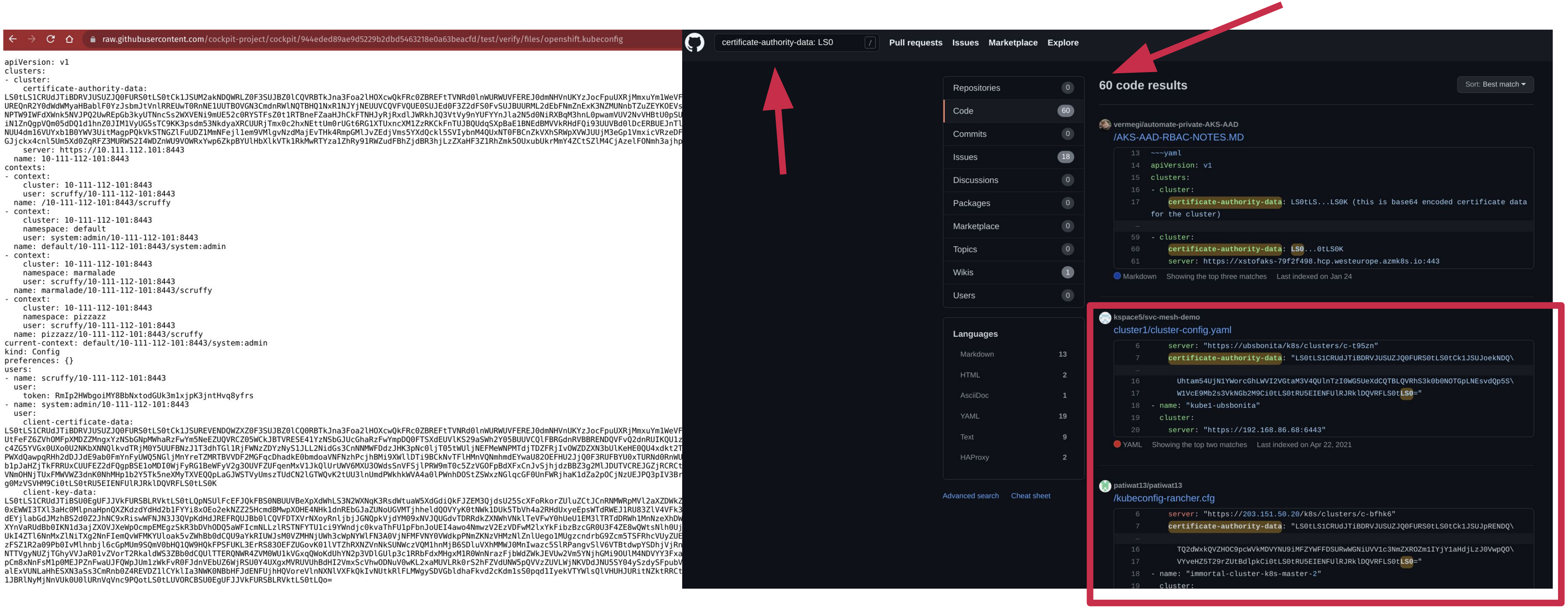
Using cloud credentials

Compromised registry and images

- Supply Chain Attacks
- Vulnerable dependencies on images



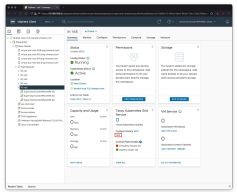
Kubeconfig file - Hunting for .kube/config

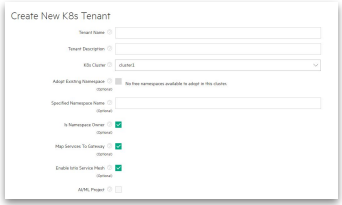


Application vulnerabilities

- OWASP Top 10
- SQLi
- RCE
- Command injection
- Etc

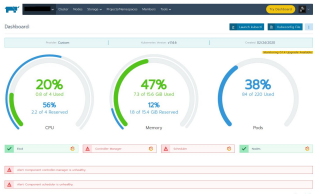
Exposed sensitive interfaces



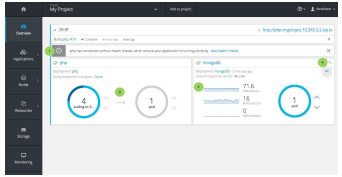


RANCHER

OPENSHIFT

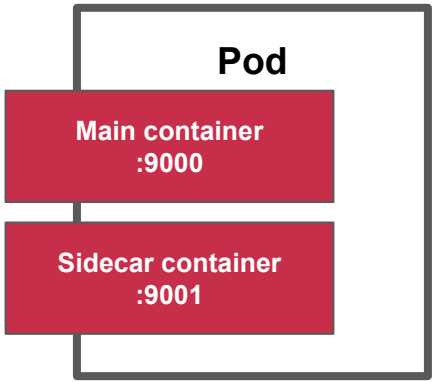


VMware Tanzu

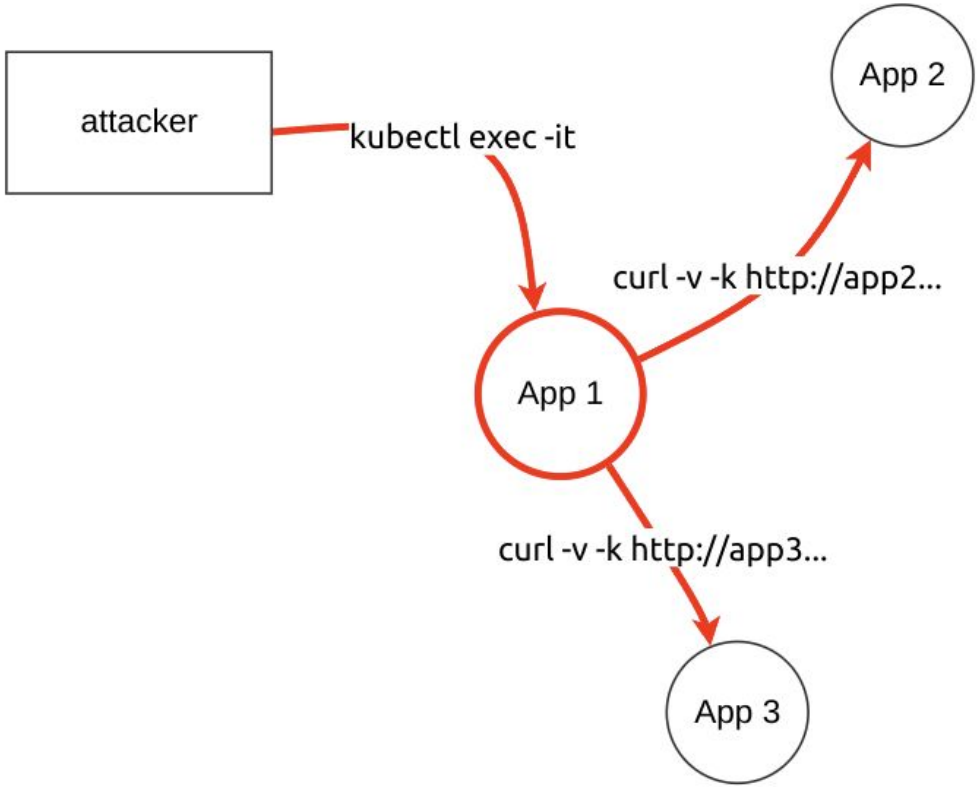


Execution - Running code inside the cluster

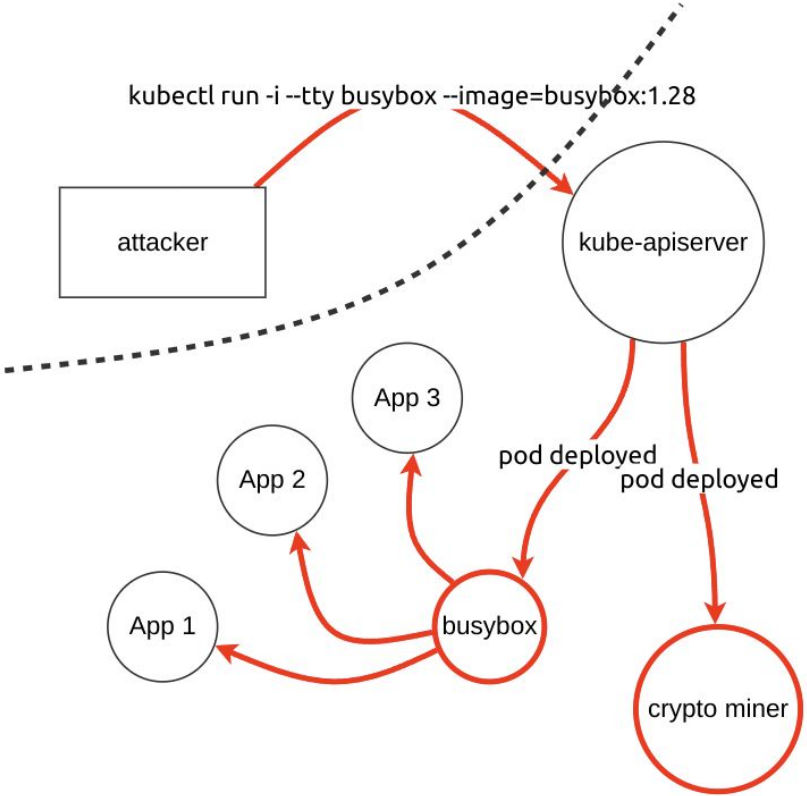
- Application exploit (RCE)
- SSH server running inside container
- Exec into container
- New container
- Sidecar Injection



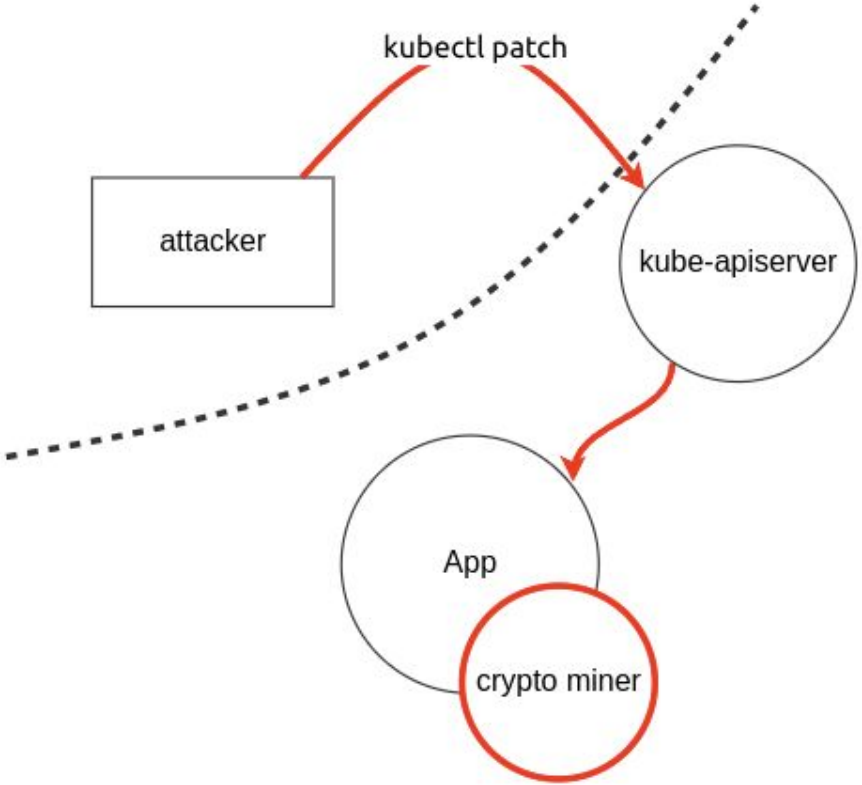
Execution - Exec into the container



Execution - Deploying new containers

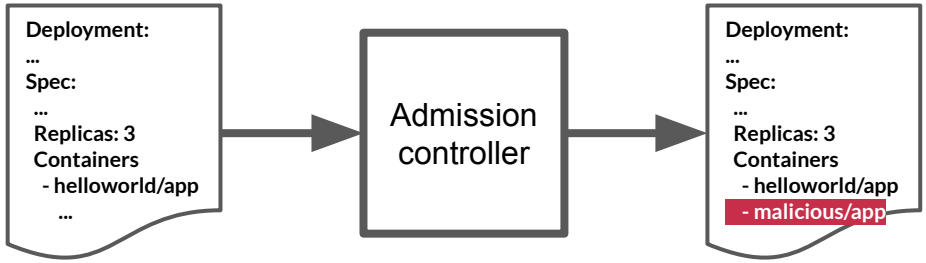


Execution - Sidecar Injection

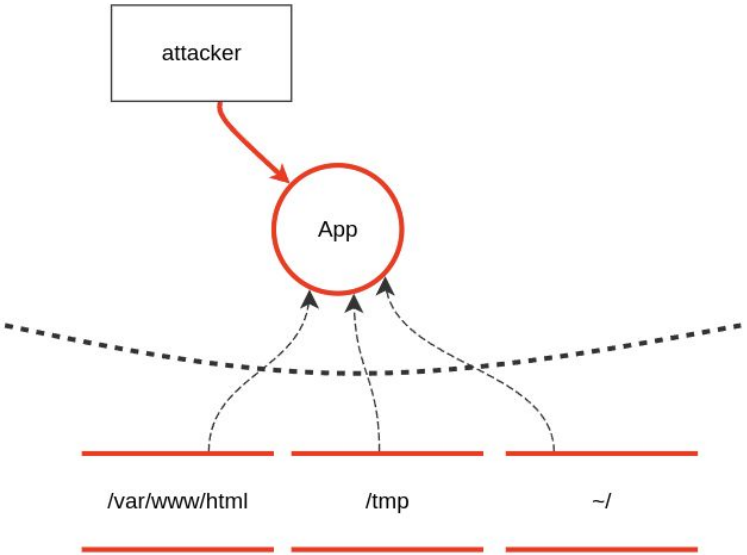


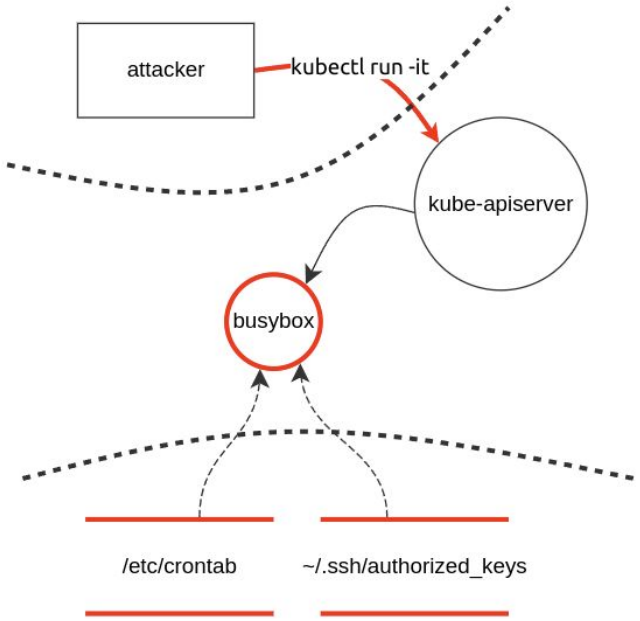
Persistence

- Backdoor container
- Kubernetes CronJob
- Writable hostPath mount
- Malicious admission controller

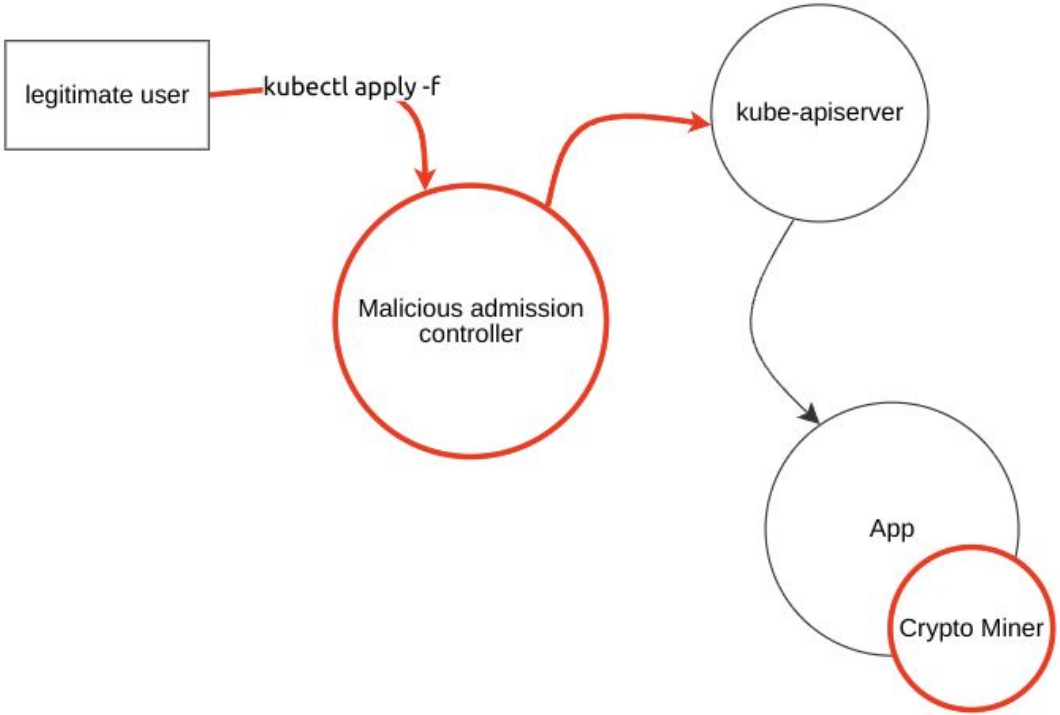


Persistence - Writable hostPath mount





Persistence - Malicious admission controller



Privilege Escalation

- HostPath mount
- Access cloud resources
- Privileged container
- Cluster-admin binding

K8S Cluster takeover

App exposed to the Internet

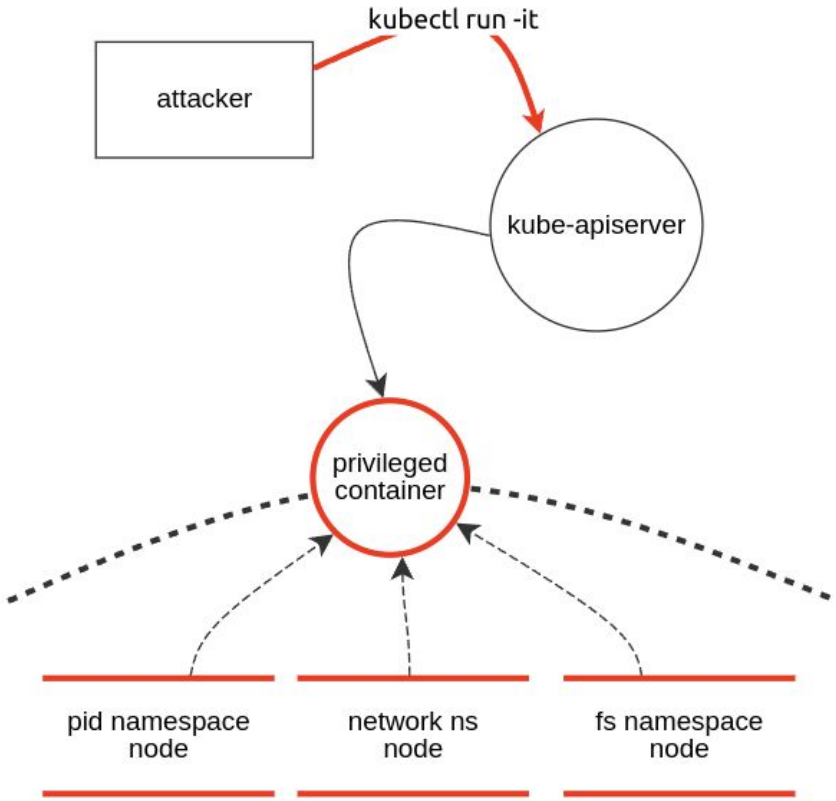
Privilege Escalation - Privileged container

Duffie Cooley 💉💉💉
@mauilion

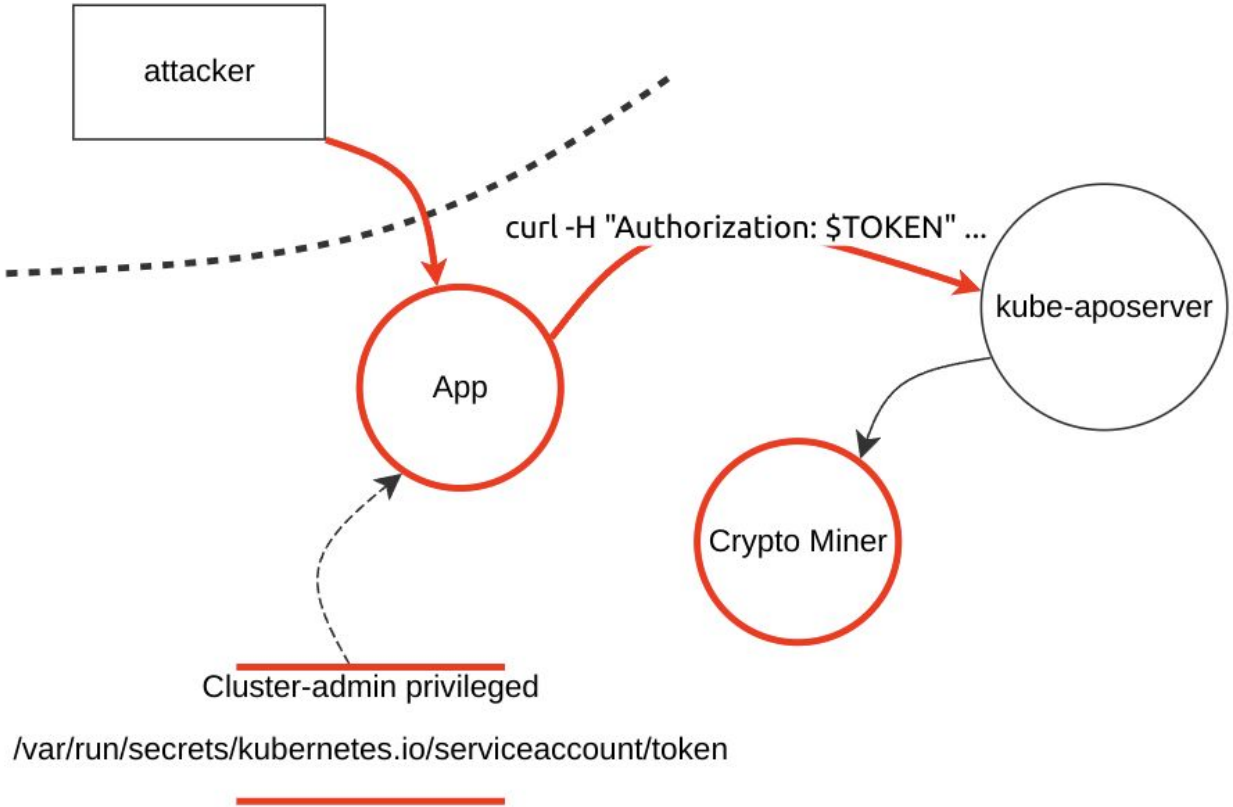
kubectl run r00t --restart=Never -ti --rm --image lol --overrides '{"spec":{"hostPID": true, "containers":[{"name":"1","image":"alpine","command":["nsenter","--mount=/proc/1/ns/mnt","--","/bin/bash"],"stdin":true,"tty":true,"securityContext":{"privileged":true}}]}}'

12:27 PM · May 17, 2019 · Twitter Web Client

Privileged container - A container that doesn't contain anything

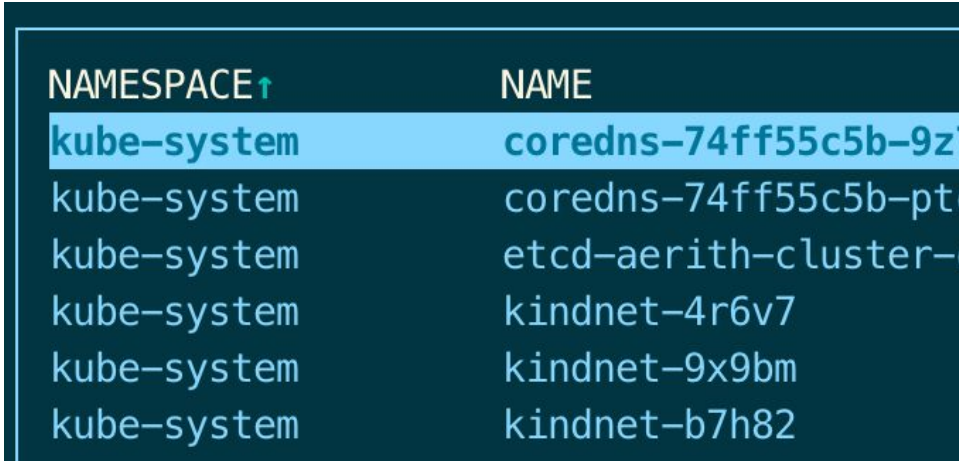


Privilege Escalation - Cluster-admin binding



Defense Evasion

- Clear container logs
- Delete Kubernetes events
- Pod / Container name similarity
- Connect from Proxy server



Credential Access

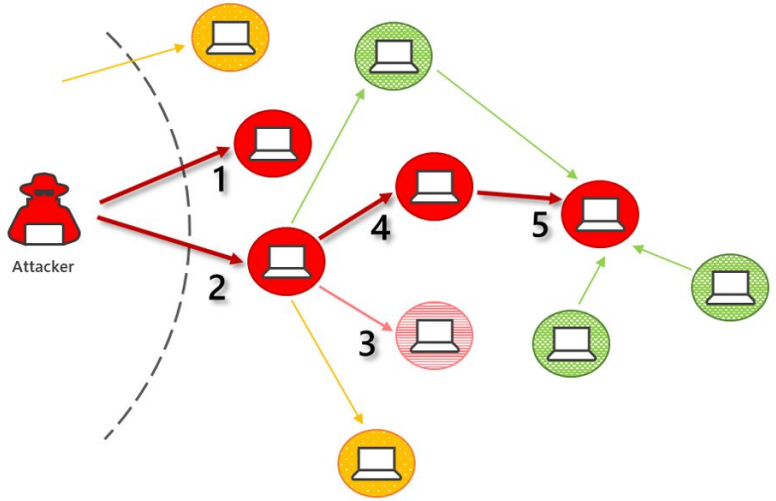
- List Kubernetes secrets
- Mount Service Principal 
- Access container service account `/var/run/secrets/kubernetes.io/serviceaccount/token`
- Applications credentials in configuration files
- Access managed identity credential  
- Malicious admission controller

Amazon EKS

Discovery

- Access the Kubernetes API server
- Access Kubelet API
- Network mapping
- Access Kubernetes dashboard
- Instance Metadata API

Lateral Movement



- Access cloud resources
- Container service account
- Cluster internal networking
- Applications credentials in configuration files
- Writable volume mounts on the host
- CoreDNS poisoning

Collection

- Images from private registry


```yaml
apiVersion: v1
kind: Secret
metadata:
  ...
  name: regcred
  ...
data:
  .dockerconfigjson: eyJodHRwczovL2luZGV4L ... J0QUl6RTIifX0=
type: kubernetes.io/dockerconfigjson
---
apiVersion: v1
kind: Pod
metadata:
  name: private-reg
spec:
  containers:
  - name: private-reg-container
    image: <your-private-image>
  imagePullSecrets:
  - name: regcred
```

Impact

- Data destruction
- Resource Hijacking
- Denial of service

Kubernetes build-in defences

Security Context for Pods

Set the security context for a Pod

- allowPrivilegeEscalation
- capabilities
- privileged
- runAsGroup
- runAsNonRoot
- runAsUser
- seLinuxOptions
- seccompProfile
- etc

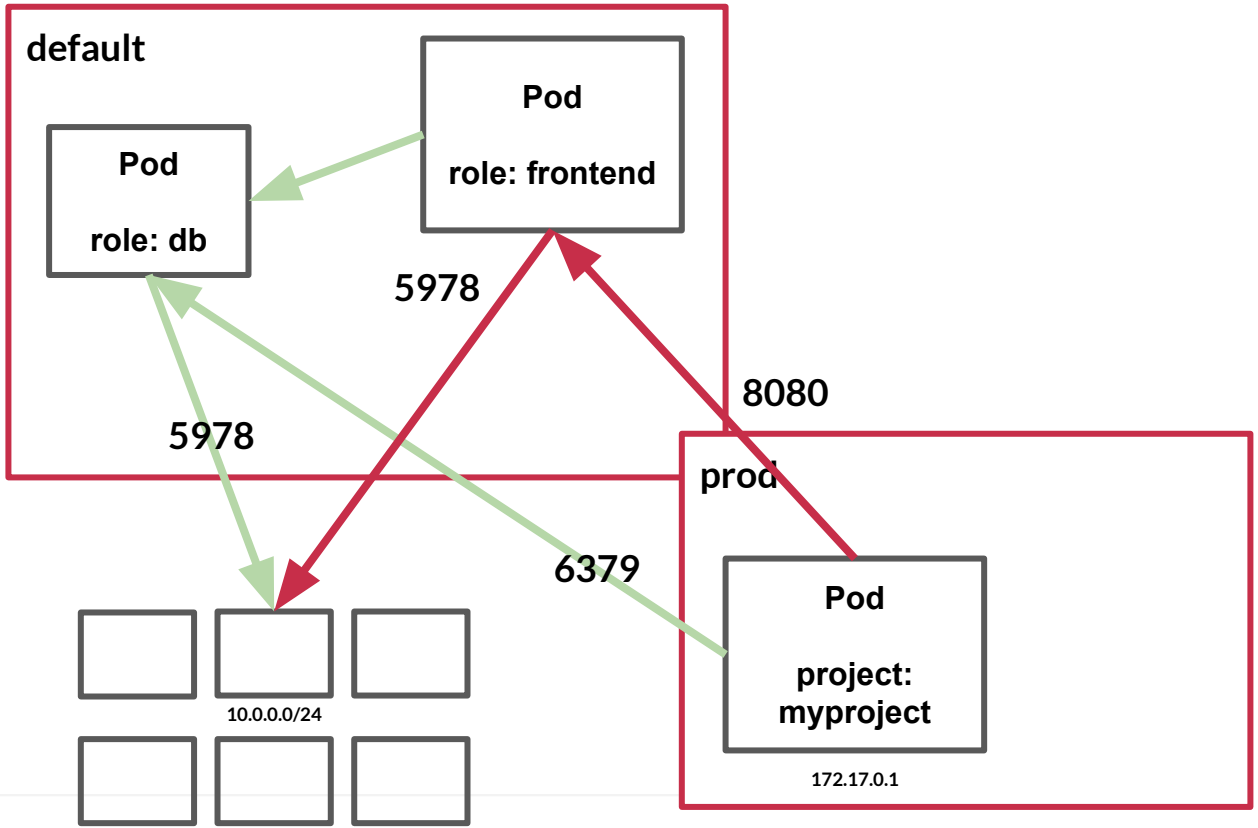
```yaml
apiVersion: v1
kind: Pod
metadata:
  name: security-context-demo
spec:
  securityContext:
    runAsUser: 1000
    runAsGroup: 3000
    fsGroup: 2000
  volumes:
  - name: sec-ctx-vol
    emptyDir: {}
  containers:
  - name: sec-ctx-demo
    image: busybox
    command: [ "sh", "-c", "sleep 1h" ]
    volumeMounts:
    - name: sec-ctx-vol
      mountPath: /data/demo
    securityContext:
      allowPrivilegeEscalation: false
```

Network Policies

Network Policies

Network Policies

```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: test-network-policy
  namespace: default
spec:
  podSelector:
    matchLabels:
      role: db
  policyTypes:
  - Ingress
  - Egress
  ingress:
  - from:
    - ipBlock:
        cidr: 172.17.0.0/16
        except:
        - 172.17.1.0/24
    - namespaceSelector:
        matchLabels:
          project: myproject
    - podSelector:
        matchLabels:
          role: frontend
    ports:
    - protocol: TCP
      port: 6379
  egress:
  - to:
    - ipBlock:
        cidr: 10.0.0.0/24
    ports:
    - protocol: TCP
      port: 5978
```



Pod Security Policies

Pod Security Policies + RBAC

- PodSecurityPolicy
- ClusterRole
- ClusterRoleBinding
- Role
- RoleBinding

Protect cluster against malicious pods

PodSecurityPolicy

```yaml
apiVersion: policy/v1beta1
kind: PodSecurityPolicy
metadata:
  name: restrictive
spec:
  privileged: false
  hostNetwork: false
  allowPrivilegeEscalation: false
  defaultAllowPrivilegeEscalation: false
  hostPID: false
  hostIPC: false
  runAsUser:
    rule: RunAsAny
  fsGroup:
    rule: RunAsAny
  seLinux:
    rule: RunAsAny
  supplementalGroups:
    rule: RunAsAny
  volumes:
  - 'configMap'
  - 'downwardAPI'
  - 'emptyDir'
  - 'persistentVolumeClaim'
  - 'secret'
  - 'projected'
```

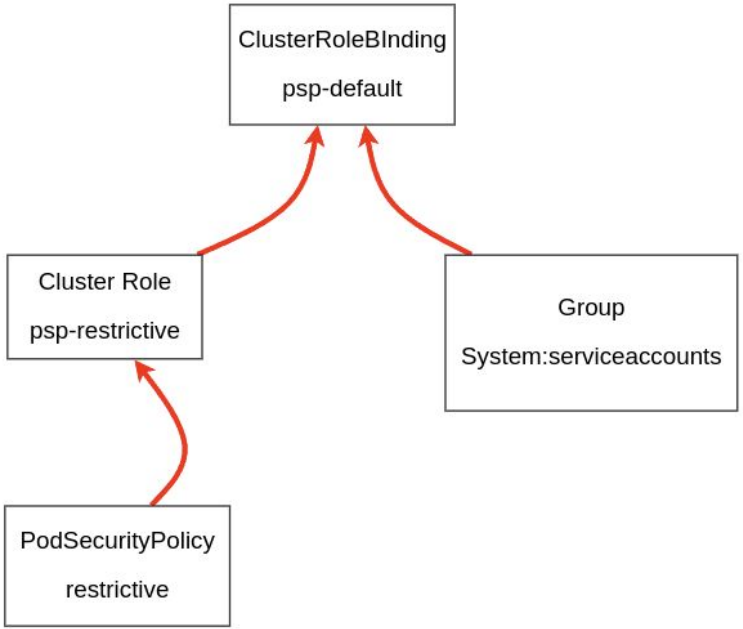
ClusterRole

```yaml
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: psp-restrictive
rules:
- apiGroups:
  - extensions
  resources:
  - podsecuritypolicies
  resourceNames:
  - restrictive
  verbs:
  - use
```

ClusterRoleBinding

```yaml
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: psp-default
subjects:
- kind: Group
  name: system:serviceaccounts
  namespace: kube-system
roleRef:
  kind: ClusterRole
  name: psp-restrictive
  apiGroup: rbac.authorization.k8s.io
```

Protect cluster against malicious pods

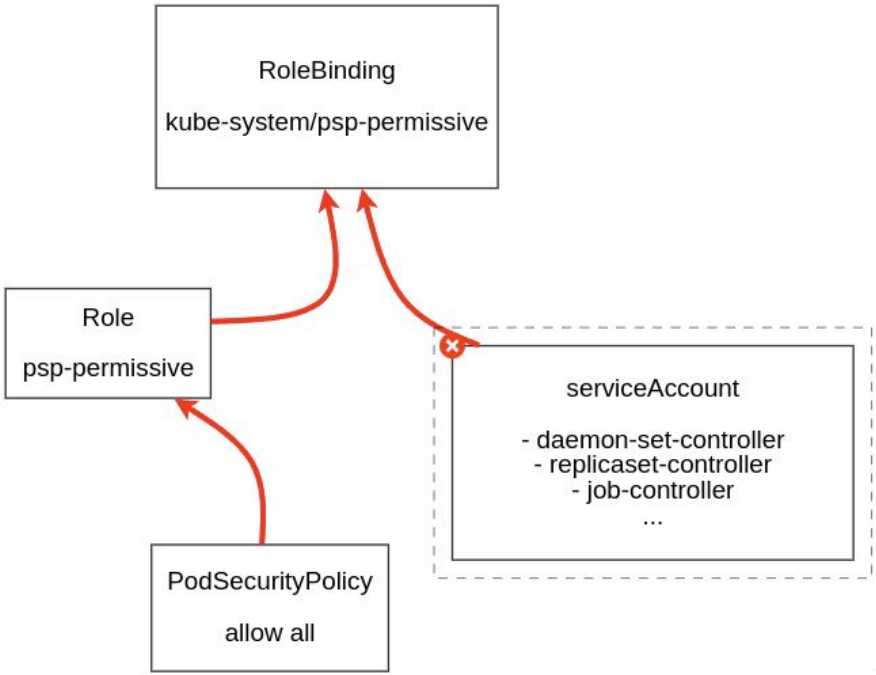


Protect cluster against malicious pods

```yaml
kind: Deployment
metadata:
  name: nginx-hostnetwork-deployment
  namespace: default
  labels:
    app: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.15.4
      hostNetwork: true
```

```
Controlled By:  Deployment/nginx-hostnetwork-deployment
Replicas:       0 current / 1 desired
Pods Status:    0 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=nginx
           pod-template-hash=597c4cff45
  Containers:
   nginx:
    Image:        nginx:1.15.4
    Port:         <none>
    Host Port:    <none>
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type             Status  Reason
  ----             ------  ------
  ReplicaFailure   True    FailedCreate
Events:
  Type     Reason        Age                From                   Message
  ----     ------        ----               ----                   -------
  Warning  FailedCreate  13s (x13 over 33s)  replicaset-controller  Error creating: pods "nginx-hostnetwork-deployment-597c4cff45-" is forbidden: unable to validate against any pod security policy: [spec.securityContext.hostNetwork: Invalid value: true: Host network is not allowed to be used]
```

Policy for pods in kube-system namespace with RoleBinding



Open Policy Agent

Open Policy Agent

- OPA Gatekeeper: Policy and Governance for Kubernetes
- https://www.openpolicyagent.org/
- Admission controller
- Validation
- Mutation (to be implemented)

Audit Log

Audit Log

- What happened?
- When did it happened?
- Who initiated it?
- On what did it happen?
- Where was it observed?
- From where was it initiated?
- ...

Logstash

Summary

- Don’t deploy images from untrusted sources in your cluster
- Try not to store sensitive data on K8S secrets, instead use something like kube-seal or and **external KMS** to manage secrets
- Choose **CNI plugin** that supports **network policies** and limit communication between your **pod/services**
- Never use **cluster-admin** the **ClusterRole** sa in your applications
- ~~Define strong **Pod Security Policies** for each of your services, **principle of least privilege** ~~Use OPA to enforce policies in your cluster
- Enable Audit Log for you cluster

Bonus: Kubernetes local CTF Challenge

Bonus: Kubernetes local CTF Challenge

Requirements

- https://docs.docker.com/get-docker/ (docker)
- https://kubernetes.io/docs/tasks/tools/ (kubectl)
- https://kind.sigs.k8s.io/ (kind)
- https://kustomize.io/ (kustomize)

Bonus: Kubernetes local CTF Challenge

Create local Kubernetes cluster

- kind create cluster

Deploy the challenge

- git clone https://github.com/Alevsk/dvka.git
- cd ./dvka/lab-1
- kustomize build k8s/base | kubectl apply -f -
- kubectl port-forward svc/nft-store 8080:8080 -n lab-1

Lab 1: NFT Museum

localhost:8080

Incognito

Unfold

HOME NFTS RESET $0

WELCOME TO THE FIRST

NFT MUSEUM

We sell and buy your precious NFTs

BUY NFT

MINIO

Resources used during this presentation

- https://www.redhat.com/en/blog/openshift-and-kubernetes-whats-difference
- https://www.microsoft.com/security/blog/2021/03/23/secure-containerized-environments-with-updated-threat-matrix-for-kubernetes/
- https://www.microsoft.com/security/blog/2020/04/02/attack-matrix-kubernetes/
- https://blog.aquasec.com/kubernetes-security-pod-escape-log-mounts
- https://www.parsons.com/2020/08/kubernetes-security-embracing-built-in-primitives-for-more-secure-environments/
- https://kubernetes.io/docs/tasks/configure-pod-container/security-context/

Thanks

 @Alevsk

 /in/alevsk/

 lenin@alevsk.com