BLUEHAT
SHANGHAI 2019

Infiltrate Your Kubernetes Cluster

Kubernetes in Attacker View

Zhaoyan Xu
Principle Research Engineer
Palo Alto Networks

Tongbo Luo
Chief AI Secruity Scientist
JD.com

May 29, 2019

Agenda

Background

Security Features of Kubernetes

Attack Vectors

Lateral Movement Practices

Questions

Background

- **Kubernetes becomes popular worldwide**
- All mainstream cloud provider their K8s cluster, such as AKS/EKS/GKE, etc.
- According to iDatalabs[1] report, around 3,804 companies use K8s for their web application deployments
- Yearly user growth rate over 150%

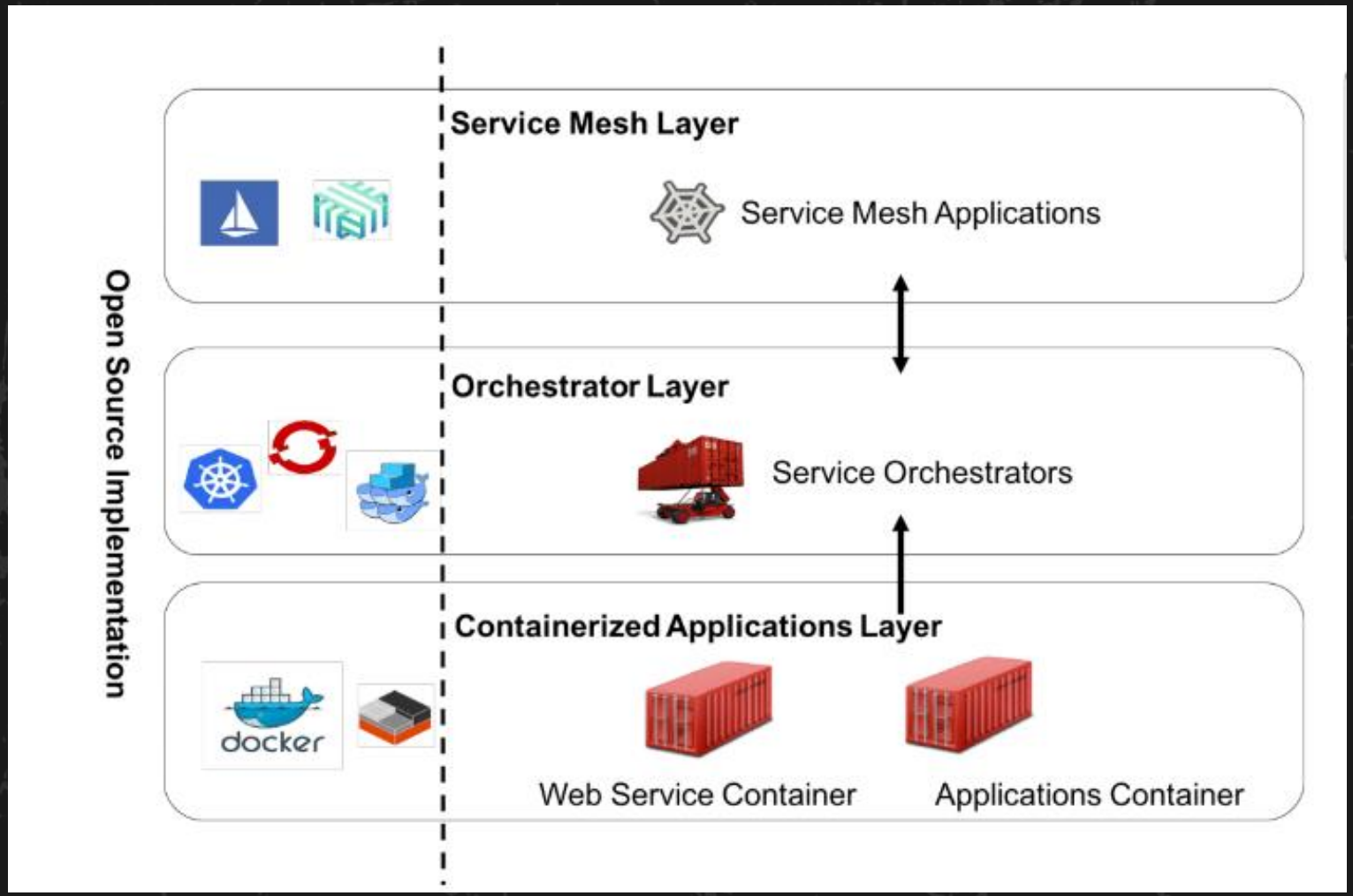
- **How secure is K8s?**
- Is K8s vulnerable to traditional attacks?
- What are new attack vectors for K8s clusters?
- How to conduct penetration test on your K8s cluster?

Essentials of Containerized Microservice

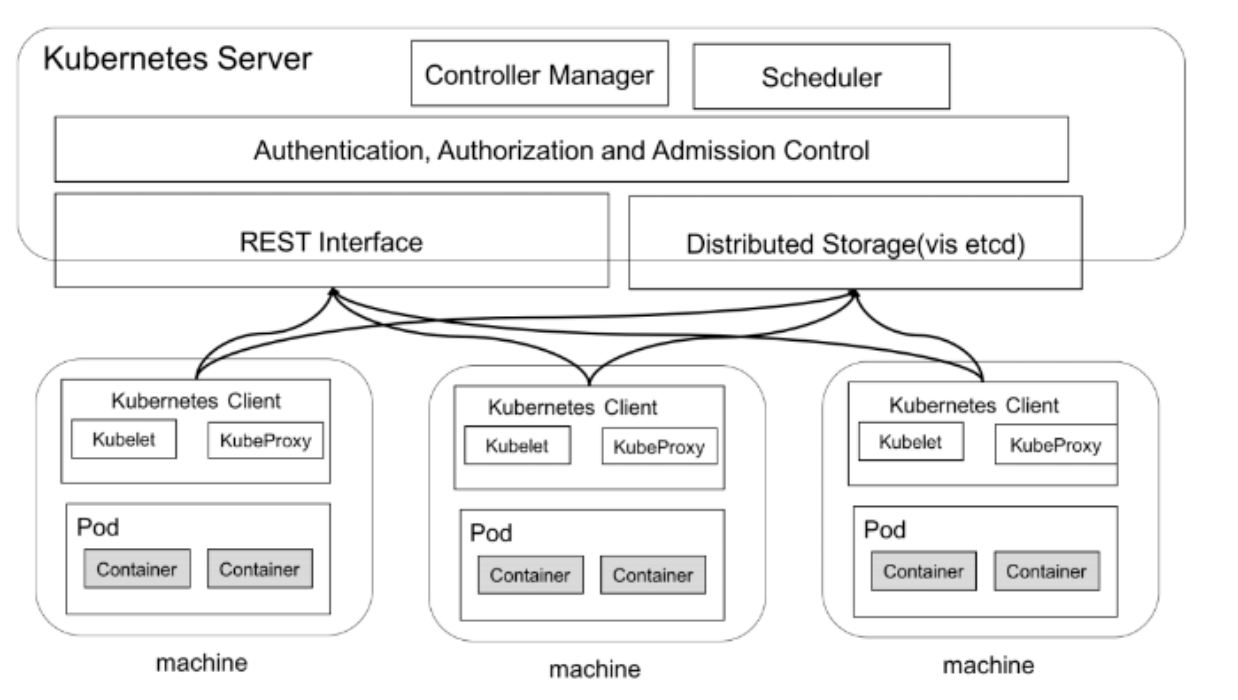
Service Mesh Layers
Istio
Linkerd

Orchestrator Layers
K8s
Openshift

Container Application Layers
Docker
Kata Container
Rkt



Essentials of K8s



Server-side Components:

api-server: central server

Controller-manger

Scheduler

Authentication/Authorization/Admission Control

etcd: kv store

Client-side Components:

kubelet: install on each host/virtual host

kubeproxy: traffic management/redirection

Terminology in K8s World

Pod: minimal unit for service schedule, containing one or more containers.

Deployment: bundle for one web application, such as combining db, frontend and backend server together.

Service: interface to expose your web application.

Service Accounts: user account in K8s.

Role/Rolebinding: role-based access control in K8s.

Agenda

Background

Security Features of Kubernetes

Attack Vectors

Lateral Movement Practices

Questions

Overview of K8s Security Features(v1.12.7)

Isolation

Pod-level Isolation

Network Security Policy for Namespace Isolation

Authentication

HTTPs for all Traffic

Token, Client certificates, third-party Authentication

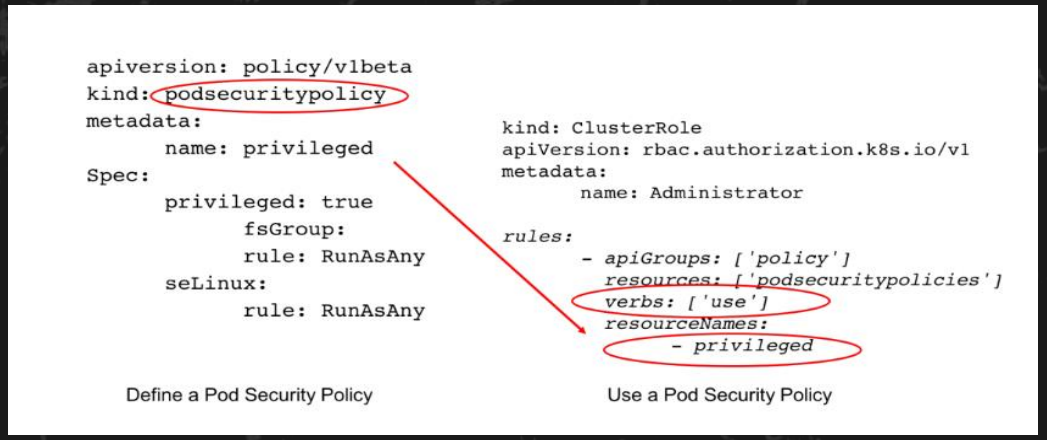
Authorization

Role-based Access Control

Admission Control (for pod, deployment, etc)

Pre-shipped Admission Control

Pod Security Policy



Pod Security Policy

cont

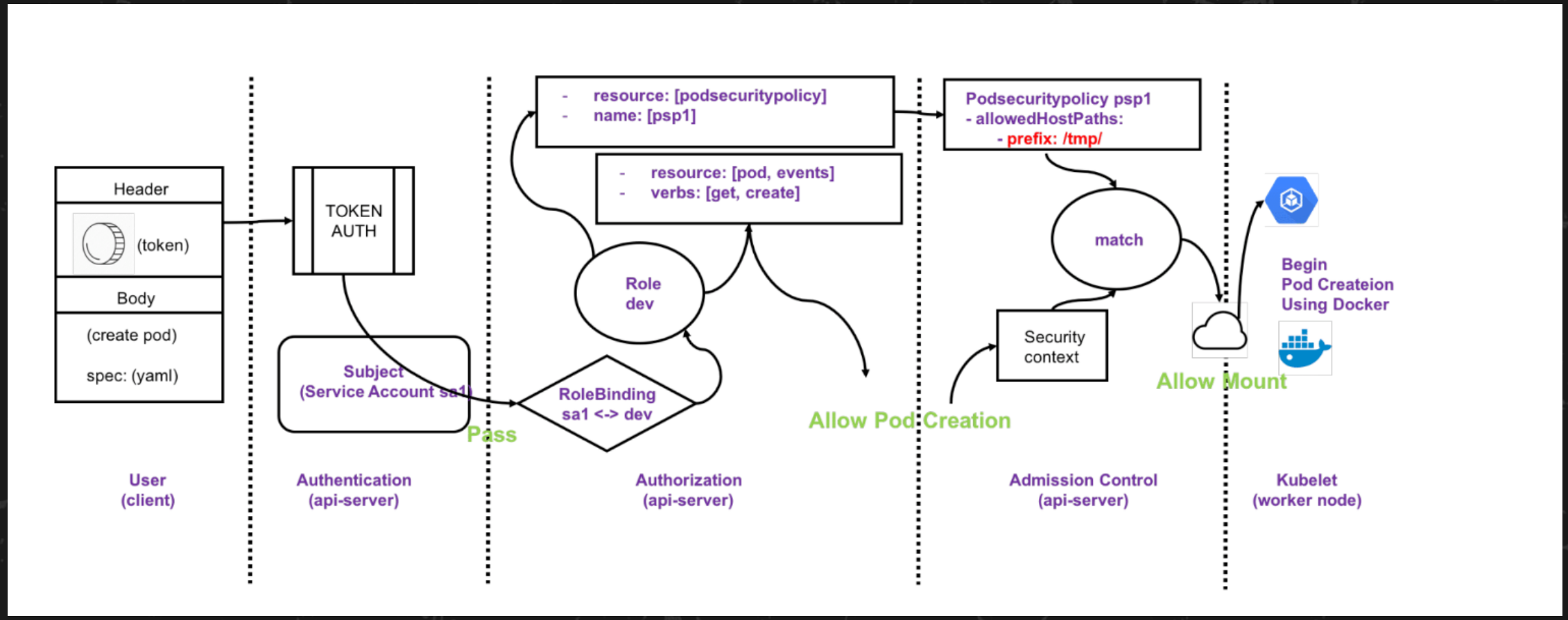


Illustration of Built-in Security Features of K8s: Creating a Pod

Agenda

Background

Security Features of Kubernetes

Attack Vectors

Lateral Movement Practices

Questions

Isolation Evasion

Network Scanning

Problem: Network Isolation is commonly enforced third-party plugins through Container Network Interface(CNI). However, most third-party plugins have vulnerabilities and some cannot enforce network security policy.

CNI plugins	Network Model	Support Network Policies	Traffic Encryption
Calico	Layer 3	Support	Encrypted
Canal	Layer2, vxlan	Support	Not Encrypted
Flannel	Layer2, vxlan	Not Support	Not Encrypted
Kopeio	Layer2, vxlan	Not Support	Not Encrypted
Kube-router	Layer2, vxlan	Support	Not Encrypted

Isolation Evasion (cont)

Network Scanning

Problem: K8s has default service pods in the namespace, kube-system, and by default, these services can be accessed by any pod in the cluster.

- One CVE Example: kube-dns pod, CVE-2017-14491

Problem: api-server can be accessed by any pod on port 6443. If the api-server allow anonymous access, it leaks your cluster's information.

- One CVE Example: CVE-2018-1002105

RBAC Evasion

Authentication Bypass

Problem: Some CNI plugins does not encrypt the traffic, so if token could be stolen if api-server does not use HTTPs.

Problem: If Role is revoked, the associated pod is not automatically killed. So it will still has the revoked role's privilege.

Authorization Abuse

Problem: Implicit Access Flow

There are multiple ways to access same resource.

Example

```
kubectl create clusterrole secretadmin --verb=get --verb=list --
verb=create --verb=update --resource=secret
```

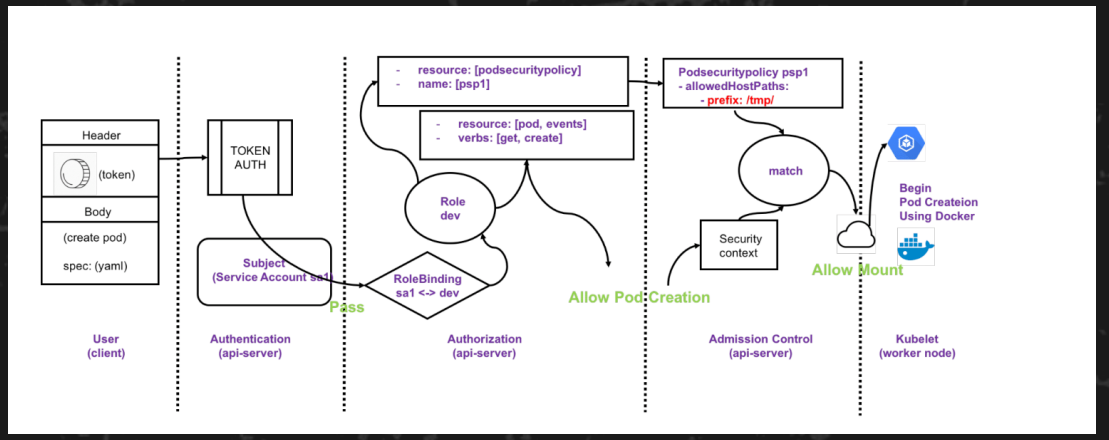
If you are not secretadmin, you cannot run, kubectl get secret, to get the secret.
However, if you have permission to create pod:

```
apiversion: v1
kind: pod
medtadata:
    - name: mypod-sa-vul
spec:
    containers:
        - name: mypod-sa
          image: alpine
          volumemounts:
            - name: foo
              mountpath: "/etc/foo"
    volumes:
        -name: foo
        secret:
            secretname: mysecret
```

Create a pod

mount a secret volume instead of getting secret

Mount the Secret through a new Pod



Potential FIX: Define a PodSecruityPolicy and define no secret is allowed to mount volumes.

RBAC Evasion (cont)

Implicit Privilege Escalation

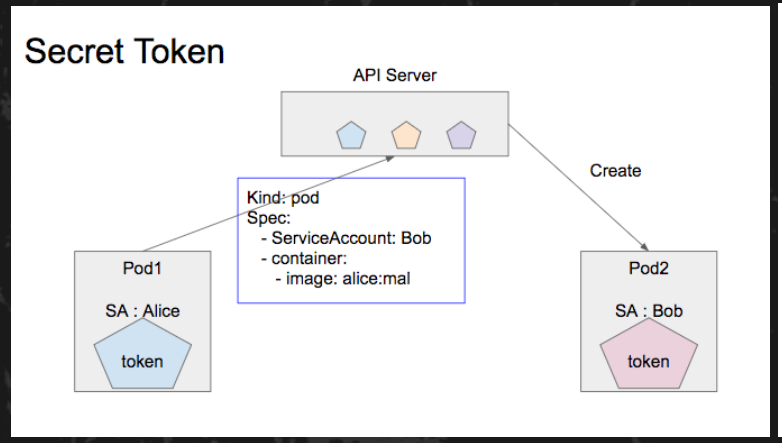
Problem: Pod can escalate it privilege by associating another service accounts.

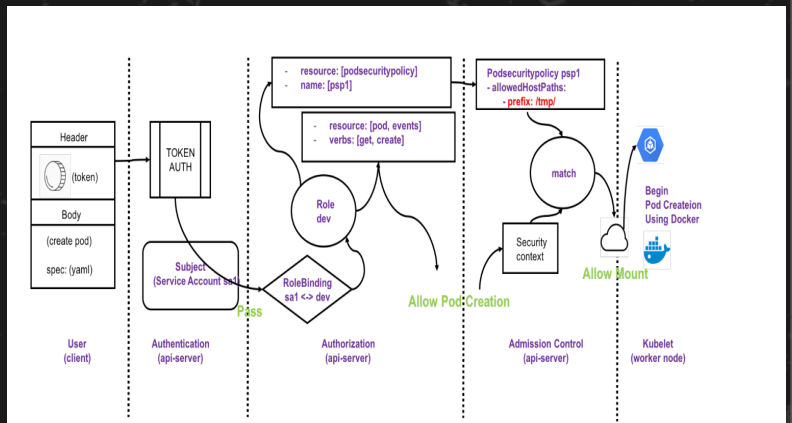
User is associated with service account sa1,

However, he can create a pod with another service account, sa2

```
apiVersion: v1
kind: pod
medtadata:
    - name: mypod-sa1
spec:
    serviceAccountName: sa2
    containers:
        - name: mypod1
          image: malicious_alpine
```

Assign Different Service Account



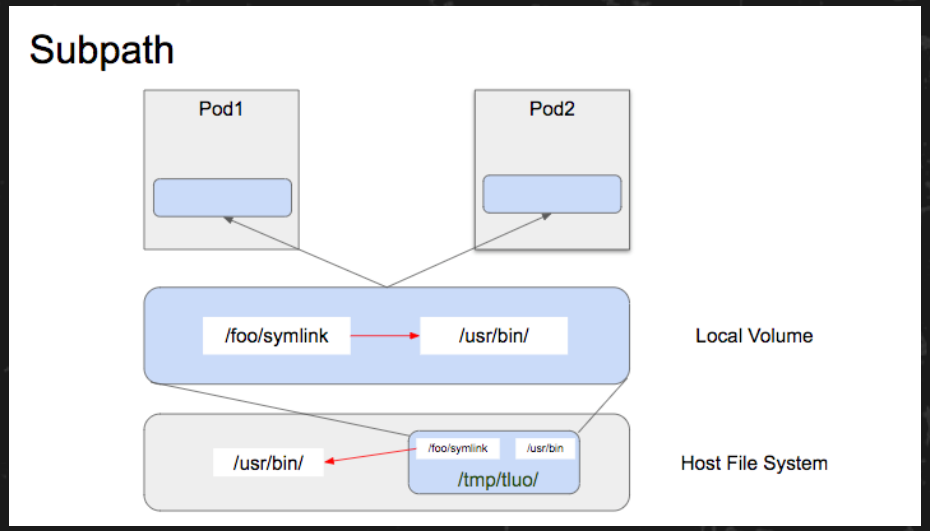


Privilege Escalation

Problem: K8s allow pod to map a hostpath, such as /tmp/, /var/log

Especially, if you mount the volume using subpath, it maps the original host file to pod's namespace.

The vulnerability: CVE-2017-1002101




```
apiVersion: v1
kind: pod
medtadata:
     - name: vuln-container3
spec:
     containers:
          - name: vuln-container3
            image: alpine
            volumeMounts:
               - mountPath: /vol
                 name: host-volume3
          - volumes:
               name:  host-volume3
               hostPath:
                    path: /tmp/test/sym   # symbolic lnk
```

Symbolic Link Points to Sensitive File

Agenda

Background

Security Features of Kubernetes

Attack Vectors

Lateral Movement Practices

Questions

Penetration in K8s

Question: From an attacker view, how to launch a lateral movement against a K8s cluster ?

Challenges: How to achieve persistence?

Hard, why?

- Transient Life-cycle of Pod
- Privilege Limited of Pod

How?

- Inject into kernel, i.e privileged container.
- Inject into host machine, i.e, privilege escalation.
- Inject into persistent storage.
- ....

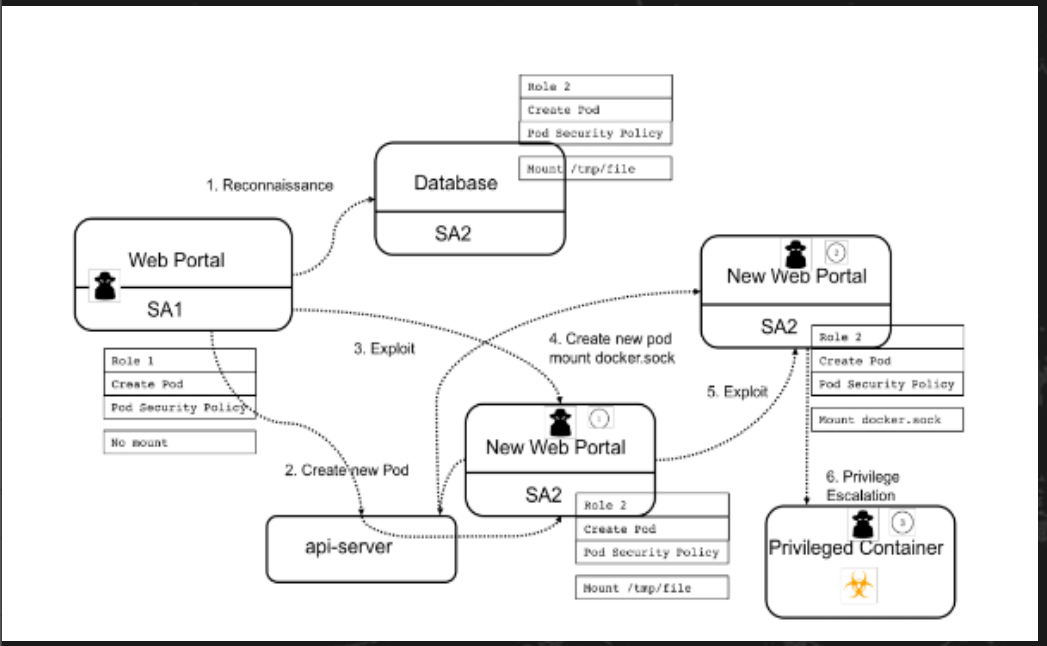
Attacker's Arsenal

Potential Way	Difficulty	Persistence	Succeed Condition	Problems
Compromise one Pod (full control)	Medium	Depends	• Pod expose its service to external • Pod's image has vulnerabilities	• Pod's transient lifecycle • Pod's limited privilege
Compromise api-server from a compromised pod	Hard	Yes	• Pod has access to the api-server • Api-server has vulnerabilities	• Pod's limited privilege to api-server • Hard to find vulnerabilities in api-server
Scan Network	Easy	No	• Flat network	
Cluster Reconnaissance	Easy	No	• Flat network or • Access to api-server	
DDoS attacks from compromised pod(s)	Easy	No	• Pod has access to network • Pod has create pod permission	• Easy to detect

Attacker's Arsenal (cont)

Potential Way	Difficulty	Persistence	Succeed Condition	Problems
Bypass RBAC	Easy	Depends	• Compromised Pod has create pod permission	• Need to know a high privileged service account
Enter Kernel	Easy	Yes	• Compromised Pod is a privileged pod	
	Hard	Yes	• Exploits container-runtime vulnerabilities	
Host Executable Replacement	Medium	Yes	• Hostpath Mount permission	
Map docker.sock	Medium	Yes	• Hostpath Mount permission	
Download malware to persistence storage	Easy	Yes	• Pod has access to persistence storage	• Hard to execute malware (need create pod privilege)

One Lateral Movement Example



Step I: Exploit Web Portal Pod which has a remote execution vulnerability

Step II: Download kubectl and query api-server

Findings: (1) exploited pod has create pod permission with service account SA1
(2) there is another db pod has mounted "/tmp/" hostpath
(3) db pod service account is SA2

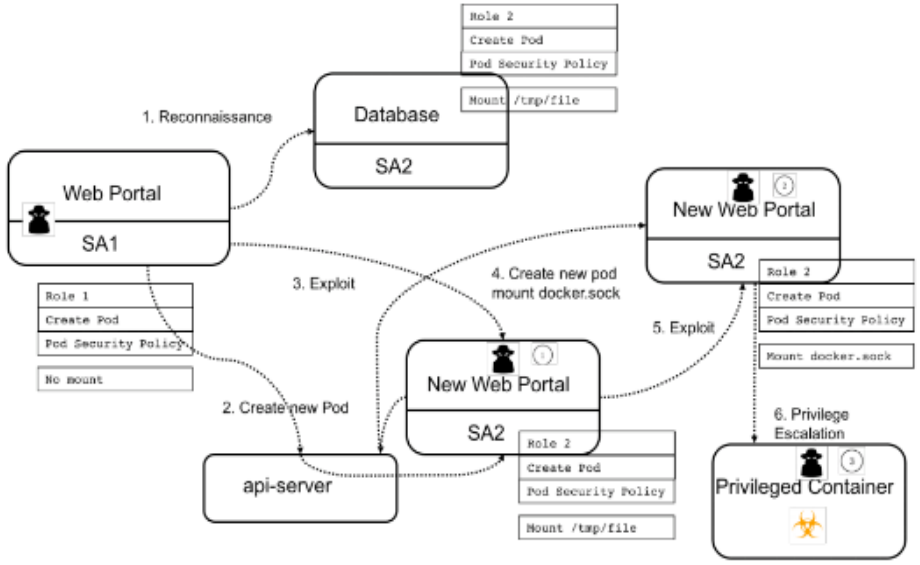
Step III: Create a new pod

(1) The pod has the vulnerable web portal image
(2) The pod uses service account SA2 and mount /tmp/ folder

Step IV: Exploit the new pod

(1) Create /tmp/sym
(2) Point the /tmp/sym at /var/run/docker.sock, which is the host docker

One Lateral Movement Example



Step V: Create another new Pod

(1) Use service account SA2

(2) Mount subpath /tmp/sym, /tmp/sym points to host /var/run/docker.run

Step VI: Send create privileged docker container command to /tmp/sym

(1) The new container is privileged and have access to kernel

Notes:

(1) Subpath vulnerability has partially fixed by google, current solution is make the subpath file read only.
However, we still think it will cause some problem like information leak, if attacker points the file to the password file.

(2) There are two root causes making the attack succeed:

a. The Pod has vulnerability to be compromised

b. The associated service account has **create pod permssion**

Agenda

Background

Security Features of Kubernetes

Attack Vectors

Lateral Movement Practices

Summary

Take-aways from Kubernetes Protection

We review the security features of Kubernetes, which includes:

Network Isolation

- Use isolation-supported CNI plugins

Authentication

- Disable Anonymous Access and Use third-party authentication service for external visit

Authorization and Access Control: Role based Access Control

- Enable RBAC
- Carefully grant pod creation/execution privilege to service accounts

Admission Control – Pod Security Policy

- Apply least privilege principle to each pod
- Understand the potential impact of privileged pod

Take-aways from Lateral Movement

Learn how an attacker could perform lateral movement within a cloud-native environment:

- Prevent an external accessible and high privileged pod in our cluster
- Grant least privilege to service accounts and pod
- Prevent/Detect Scanning Traffic in your cluster and set proper resource limit for each pod
- Use network security policy and pod security policy to manage your K8s cluster
- Upgrade/patch vulnerabilities for Kubernetes

Tools we can use to protect us

Image Vulnerability Scanner

https://github.com/coreos/clair

https://github.com/aquasecurity/kube-hunter

Kubernetes Security/Compliance Check

https://www.cisecurity.org/benchmark/kubernetes/

https://www.cisecurity.org/benchmark/docker/

Pod Security Auditing Tools

https://github.com/sysdiglabs/kube-psp-advisor

Run time Kubernetes Monitoring

https://github.com/falcosecurity/falco

Q & A?