

Kubernetes 集群渗透

攻击者视角中的Kubernetes

Zhaoyan Xu

资深研究工程师

Palo Alto Networks

Tongbo Luo

首席AI安全科学家

JD.com

2019年5月29日

日程



背景



Kubernetes的安全特性



攻击方式



横向运动实践



答疑

背景

➤ Kubernetes在全球范围内广受欢迎

- 所有主流云提供商都提供K8S集群服务，如AKS / EKS / GKE等
- 根据iDatalabs^[1]的报告，大约有3,804家公司使用K8进行Web应用程序部署
- 年度用户增长率超过150%

➤ K8的安全性如何？

- K8S是否容易受到传统攻击？
- 什么是K8S群集的新攻击方式？
- 如何在K8S集群上进行渗透测试？

容器化微服务要点

Service Mesh 层

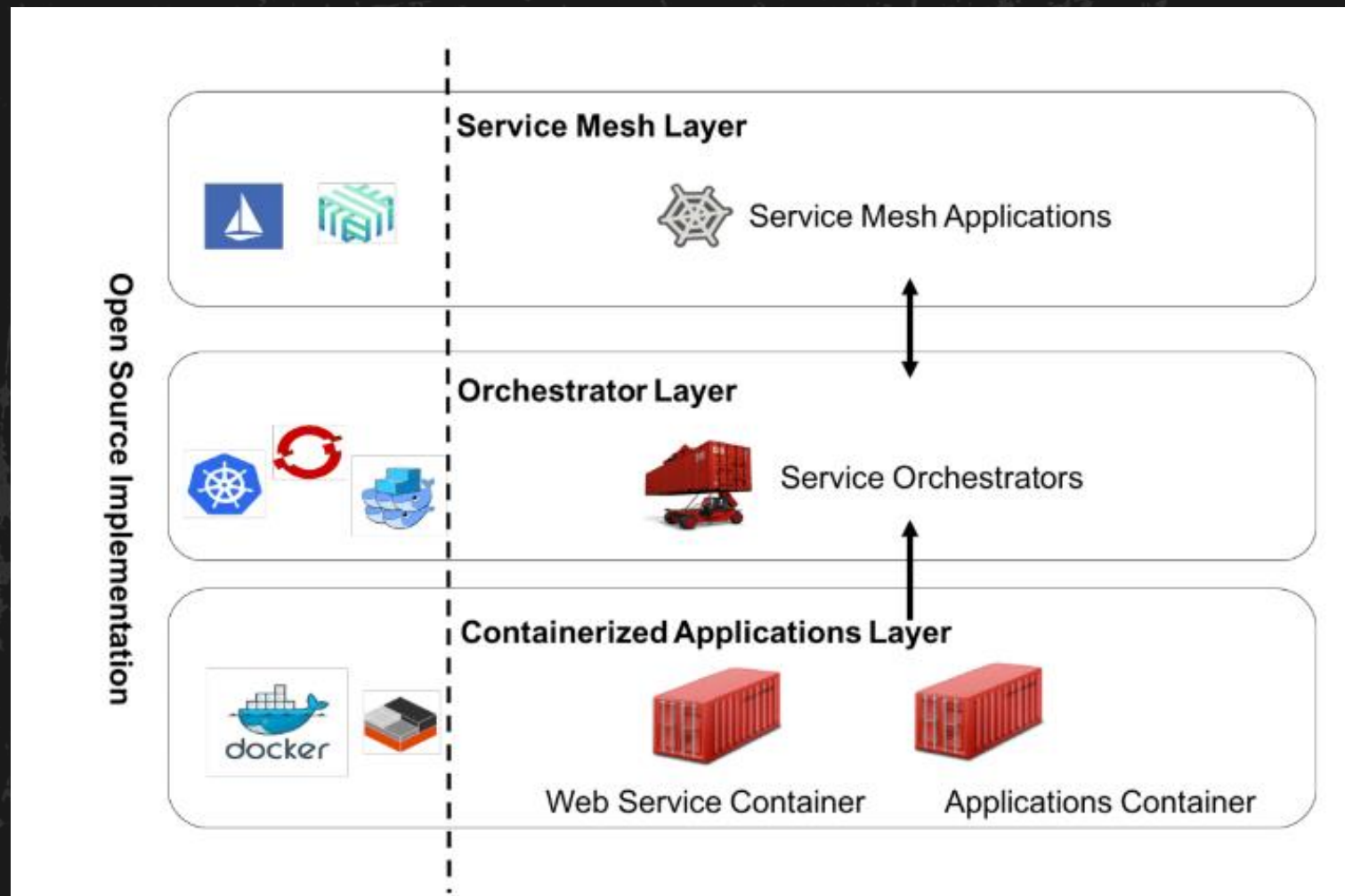
Istio
Linkerd

Orchestrator 层

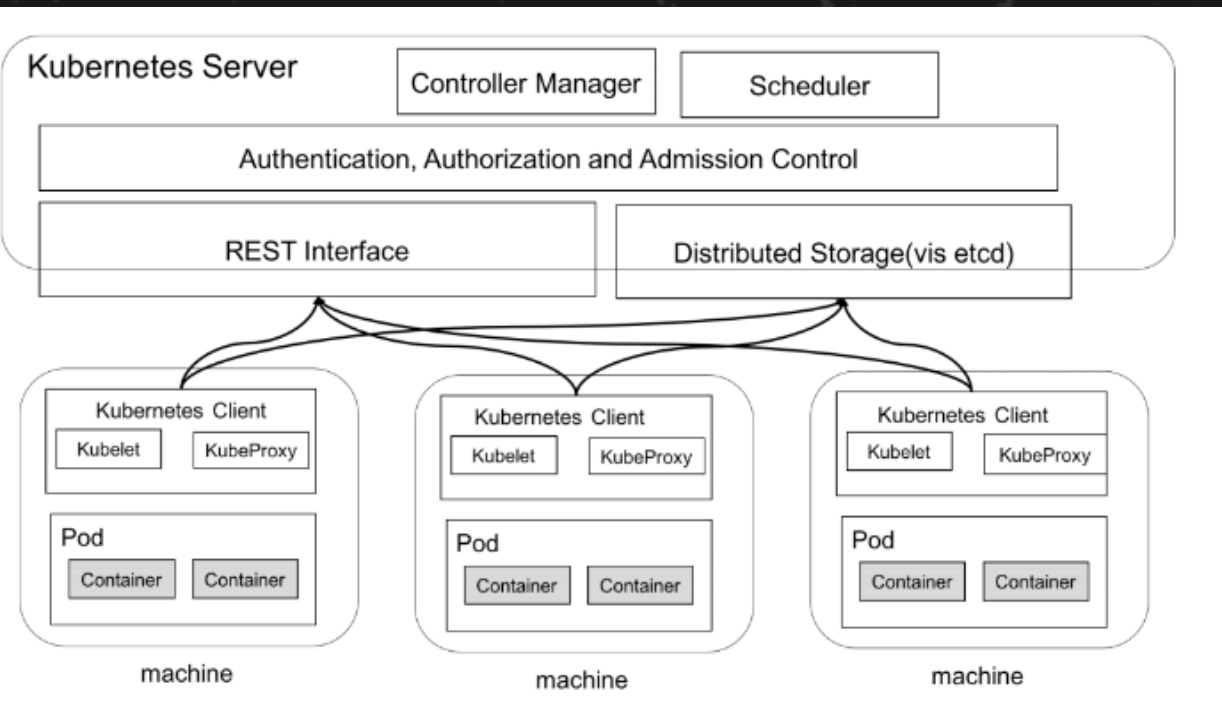
K8S
Openshift

容器应用层

Docker
Kata Container
Rkt



K8S要点



服务端组件:

api-server: central server

Controller-manger

Scheduler

Authentication/Authorization/Admission Control

etcd: kv store

客户端组件:

kubelet: 在每个主机/虚拟主机上安装

kubeproxy: 流量管理/重定向

K8S世界的术语

Pod: 服务计划的最小单位, 包含一个或多个容器。

Deployment-部署: 捆绑一个Web应用程序, 例如将db, frontend和backend服务器组合在一起。

Service-服务: 用于公开Web应用程序的界面。

Service Account-服务帐户: K8S中的用户帐户。

角色/角色绑定 Role/ Rolebinding: K8S中基于角色的访问控制。

日程



背景



Kubernetes安全特性



攻击方式



横向运动实践



答疑

K8S安全功能概述 (v1.12.7)

隔离

Pod级隔离

名称空间隔离的网络安全策略

认证

所有流量的HTTP

令牌, 客户端证书, 第三方身份验证

授权

基于角色的访问控制

准入控制 (用于pod, 部署等)

预制的管理控制

Pod安全政策

```
apiversion: policy/v1beta1
kind: podsecuritypolicy
metadata:
  name: privileged
Spec:
  privileged: true
  fsGroup:
    rule: RunAsAny
  seLinux:
    rule: RunAsAny

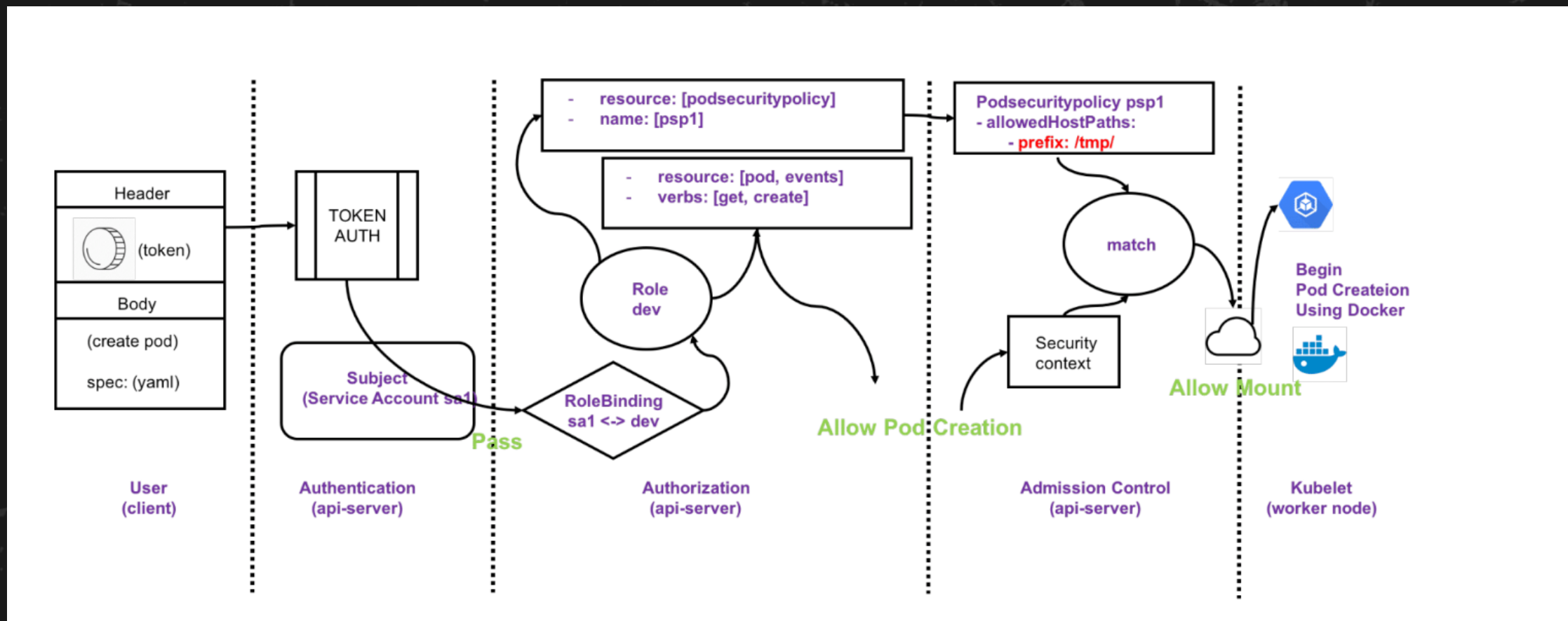
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: Administrator
rules:
- apiGroups: ['policy']
  resources: ['podsecuritypolicies']
  verbs: ['use']
  resourceNames:
    - privileged
```

Define a Pod Security Policy

Use a Pod Security Policy

Pod 安全策略

继续



K8S内置安全功能图示：创建Pod

日程



背景



Kubernetes安全特性



攻击方式



横向运动实践



答疑

隔离躲避

网络扫描

问题：网络隔离通常是通过容器网络接口（CNI）强制执行的第三方插件。但是，大多数第三方插件都存在漏洞，有些插件无法实施网络安全策略。

CNI 插件	网络模型	支持网络策略	通讯加密
Calico	Layer 3	支持	加密
Canal	Layer2, vxlan	支持	非加密
Flannel	Layer2, vxlan	不支持	非加密
Kopeio	Layer2, vxlan	不支持	非加密
Kube-router	Layer2, vxlan	支持	非加密

隔离躲避 (续)

网络扫描

问题： K8S在命名空间kube-system中有默认服务pod，默认情况下，群集中的任何pod都可以访问这些服务。

- CVE示例： kube-dns pod, [CVE-2017-14491](#)

问题： api-server可以通过端口6443上的任何pod访问。如果api-server允许匿名访问，它会泄漏您的群集信息。

- CVE示例： [CVE-2018-1002105](#)

RBAC 躲避

认证绕过

问题：某些CNI插件不会加密流量，因此如果api-server不使用HTTP，则令牌可能被盗。

问题：如果撤销角色，则不会自动终止关联的窗格。所以它仍然具有被撤销角色的特权。

认证滥用

问题：隐式访问流程

有多种方法可以访问相同的资源。

示例

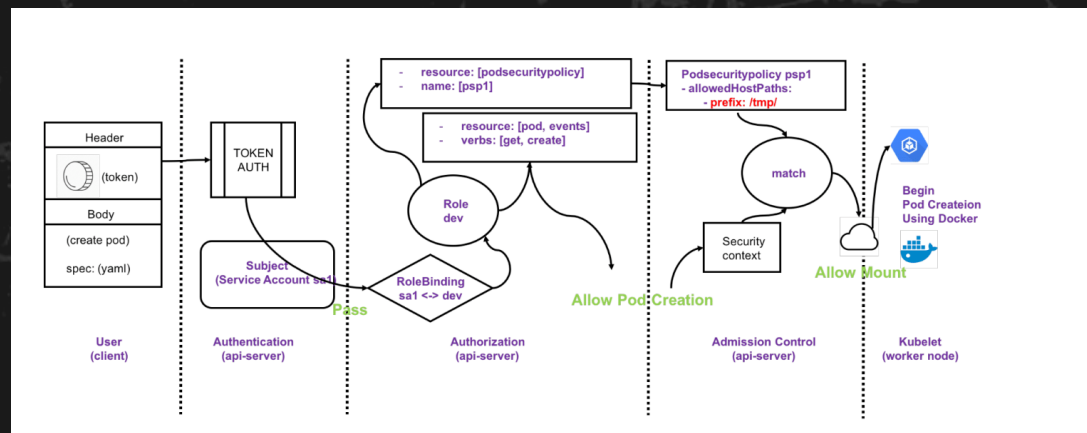
```
kubectl create clusterrole secretadmin --verb=get --verb=list --verb=create --verb=update --resource=secret
```

如果你没有密码管理权限，你就无法运行 `kubectl get secret`，以获取secret。
但是，如果您有权创建pod：

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

Annotations in the image:
- A red circle around `kind: pod` with the text "Create a pod".
- A red box around the `volumes` section with the text "mount a secret volume instead of getting secret".

通过一个新的 Pod 嵌入密码



可能的修复措施：定义 PodSecurityPolicy，并定义不允许嵌入密码。

RBAC 躲避 (续)

隐性权限提升

问题：Pod可以通过关联其他服务帐户来升级其权限。

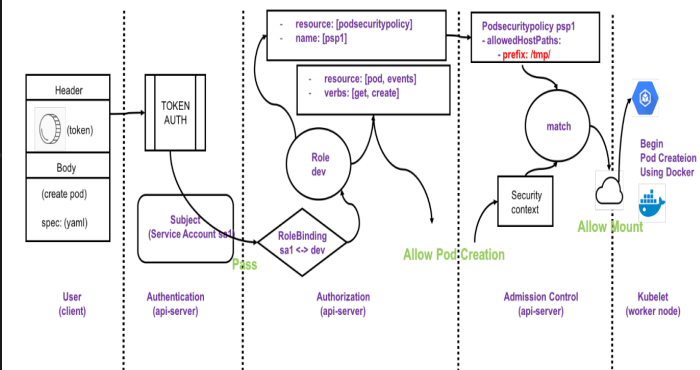
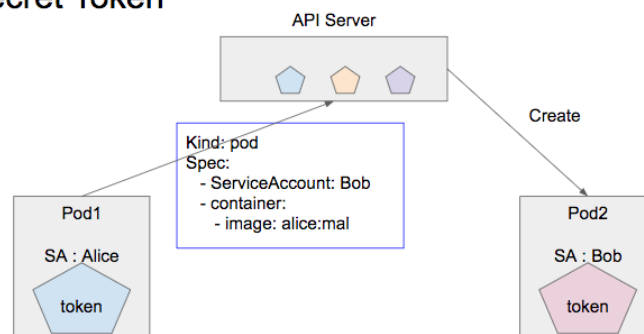
用户与服务帐户sa1相关联，

但是，他可以使用另一个服务帐户sa2创建一个pod

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

Assign Different Service Account

Secret Token



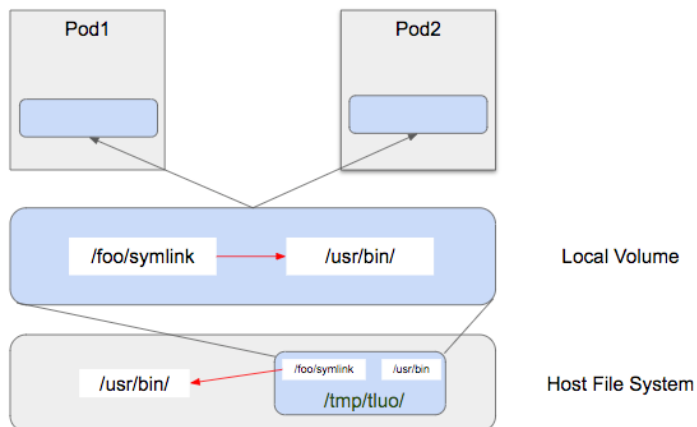
特权提升

问题: K8S允许pod映射主机路径, 例如/tmp/, /var/log

特别是, 如果使用子路径装入卷, 它会将原始主机文件映射到pod的命名空间。

漏洞: [CVE-2017-1002101](#)

Subpath



```
apiVersion: v1
kind: pod
metadata:
  - 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

日程



背景



Kubernetes安全特性



攻击方式



横向运动实践



答疑

K8S的渗透攻击

问：从攻击者的角度来看，如何针对K8S群集发起横向移动？

挑战：如何实现持久性？

很难，为什么？

- Pod的瞬态生命周期
- Pod的有限特权

如何？

- 注入内核，如：特权容器。
- 注入主机，如：特权提升。
- 注入持久存储。
-

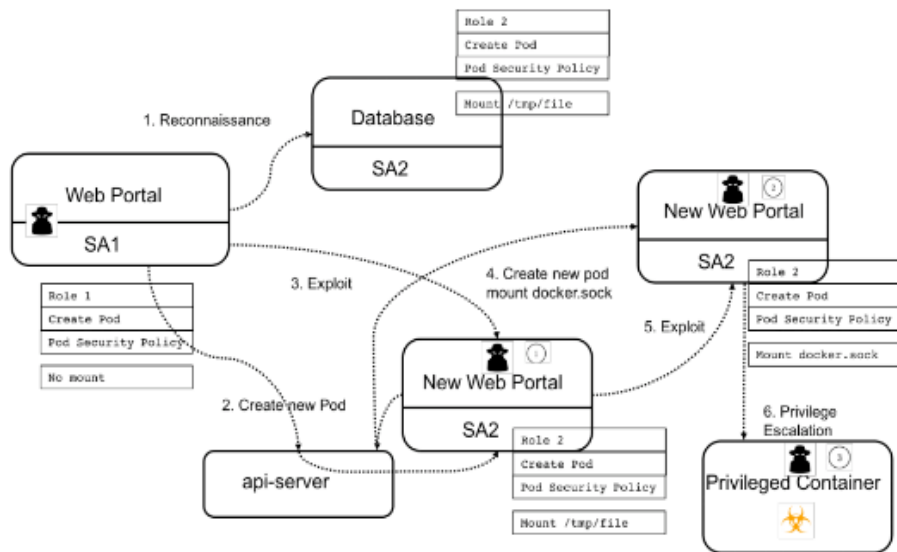
攻击者的军火库

潜在方法	难度	持续性	前置条件	问题
入侵一个Pod (完全控制)	中等	取决于实际情况	<ul style="list-style-type: none">Pod将其服务暴露给外部Pod的映像存在漏洞	<ul style="list-style-type: none">Pod的瞬态生命周期Pod的有限特权
从受损的pod中入侵api-server	困难	是	<ul style="list-style-type: none">Pod可以访问api-serverApi-server存在漏洞	<ul style="list-style-type: none">Pod只对api-server的有限权限很难在api-server中找到漏洞
扫描网络	简单	否	<ul style="list-style-type: none">Flat 网络	
集群侦察	简单	否	<ul style="list-style-type: none">Flat network or能访问到api-server	
来自被入侵的pod的DDoS攻击	简单	否	<ul style="list-style-type: none">Pod可以访问网络Pod已创建pod权限	<ul style="list-style-type: none">容易被检测到

攻击者的军火库(继续)

潜在方法	难度	持久性	前置条件	问题
绕过RBAC	简单	取决于实际情况	<ul style="list-style-type: none">被控制的Pod具有创建pod权限	<ul style="list-style-type: none">需要了解高特权服务帐户
进入内核	简单	是	<ul style="list-style-type: none">被控制的Pod是一个特权Pod	
	困难	是	<ul style="list-style-type: none">利用容器运行时漏洞	
替换主机可执行文件	中等	是	<ul style="list-style-type: none">Hostpath Mount 权限	
映射 docker.sock	中等	是	<ul style="list-style-type: none">Hostpath Mount 权限	
将恶意软件下载到持久性存储	简单	是	<ul style="list-style-type: none">Pod可以访问持久性存储	<ul style="list-style-type: none">难以执行恶意软件（需要创建pod权限）

一个横向运动的例子



第一步：利用具有远程执行漏洞的Web Portal Pod

第二步：下载kubectl并查询api-server

嗅探结果： (1) 被利用的pod已经与服务帐户SA1创建了pod权限
(2) 还有另一个db pod已经安装了"/ tmp /"主机路径
(3) db pod服务帐号为SA2

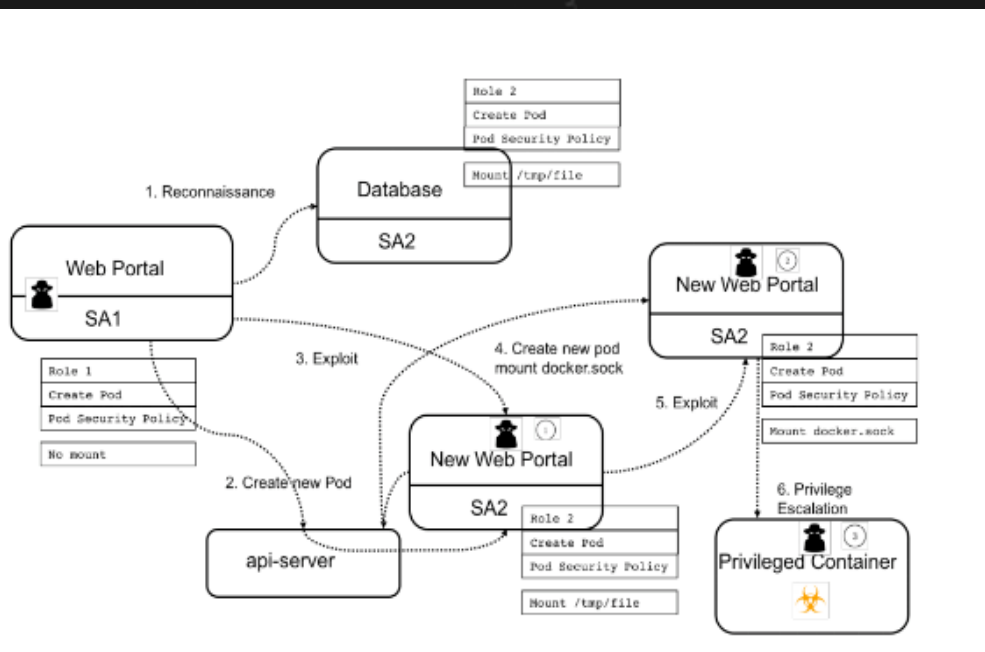
第三步：创建一个新的pod

- 该pod具有易受攻击的Web门户图像
- 该pod使用服务帐户SA2和mount / tmp /文件夹

第四步：利用新的pod

- (1) 创建/ tmp / sym
- (2) 将/ tmp / sym指向/var/run/docker.sock, 它是docker主及

一个横向运动的例子



第五步：创建另一个新Pod

- (1) 使用服务帐户SA2
- (2) 挂载子路径/ tmp/sym, /tmp/sym指向主机 /var/run/docker.run

第六步：将创建特权容器的命令发送到/tmp/ sym

- (1) 新容器具有特权并且可以访问内核

备注：

1) 谷歌部分修复了子路径漏洞，目前的解决方案是使子路径文件只读。

但是，如果攻击者将文件指向密码文件，我们仍然认为它会导致信息泄漏等问题。

2) 攻击成功有两个根本原因：

- a. Pod容易受到攻击
- b. 关联的服务帐户具有**创建 pod 权限**

日程



背景



Kubernetes安全特性



攻击方式



横向运动实践



总结

Kubernetes 安全防护总结

我们回顾了Kubernetes的安全功能，其中包括：

网络隔离

- 使用支持隔离的CNI插件

认证

- 禁用匿名访问并使用第三方身份验证服务进行外部访问

授权和访问控制：基于角色的访问控制

- 启用RBAC
- 小心地将创建Pod/执行权限授予服务帐户

权限控制 - Pod安全策略

- 对每个Pod应用最小特权原则
- 了解特权Pod的潜在影响

横向移动总结

了解攻击者如何在云原生环境中执行横向移动：

- **防止群集中的外部可访问和高权限Pod**
- **授予服务帐户和Pod的最小权限**
- **阻止/检测群集中的扫描流量并为每个Pod设置适当的资源限制**
- **使用网络安全策略和Pod安全策略来管理K8S群集**
- **升级/修补Kubernetes的漏洞**

推荐的防护工具

映像漏洞扫描工具

<https://github.com/coreos/clair>

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

Kubernetes 安全性/合规性 检查工具

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

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

Pod 安全审计工具

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

Run time Kubernetes Monitoring

<https://github.com/falcosecurity/falco>

答疑