



SHANGHAI 2019

# 我的容器安全吗？

有关容器漏洞的研究

Cecilia Hu, Yue Guan, Zhaoyan Xu

*Palo Alto Networks*

# 议程

- 简介
- 基本统计数据
- 漏洞研究
- 集群中的镜像
- 实践建议

# 背景

- 容器已成为在云平台上配置微服务的流行方式。
- 随着越来越先进的云应用程序陆续部署，镜像的安全风险已成为 DevOps 团队最头疼的问题。
- 我们想知道具体情况到底有多糟糕，以及如何防范这些威胁。

# 动机

本次交流我们将涉及：

- 容器镜像安全性的最新现状是怎样的？
- 如何衡量您应用程序环境中容器镜像的安全性？
- 如何缓解容器镜像漏洞所造成的威胁？
- 镜像保护方面有何最佳实践？

# 日程

- 简介
- 基本统计
- 漏洞研究
- 集群中的镜像
- 实践建议

# 方法

我们通过系列方式研究这个问题：

第 1 步：针对诸如 DockerHub 和 GitHub 等不同来源，且公开可用的镜像仓库进行爬网。

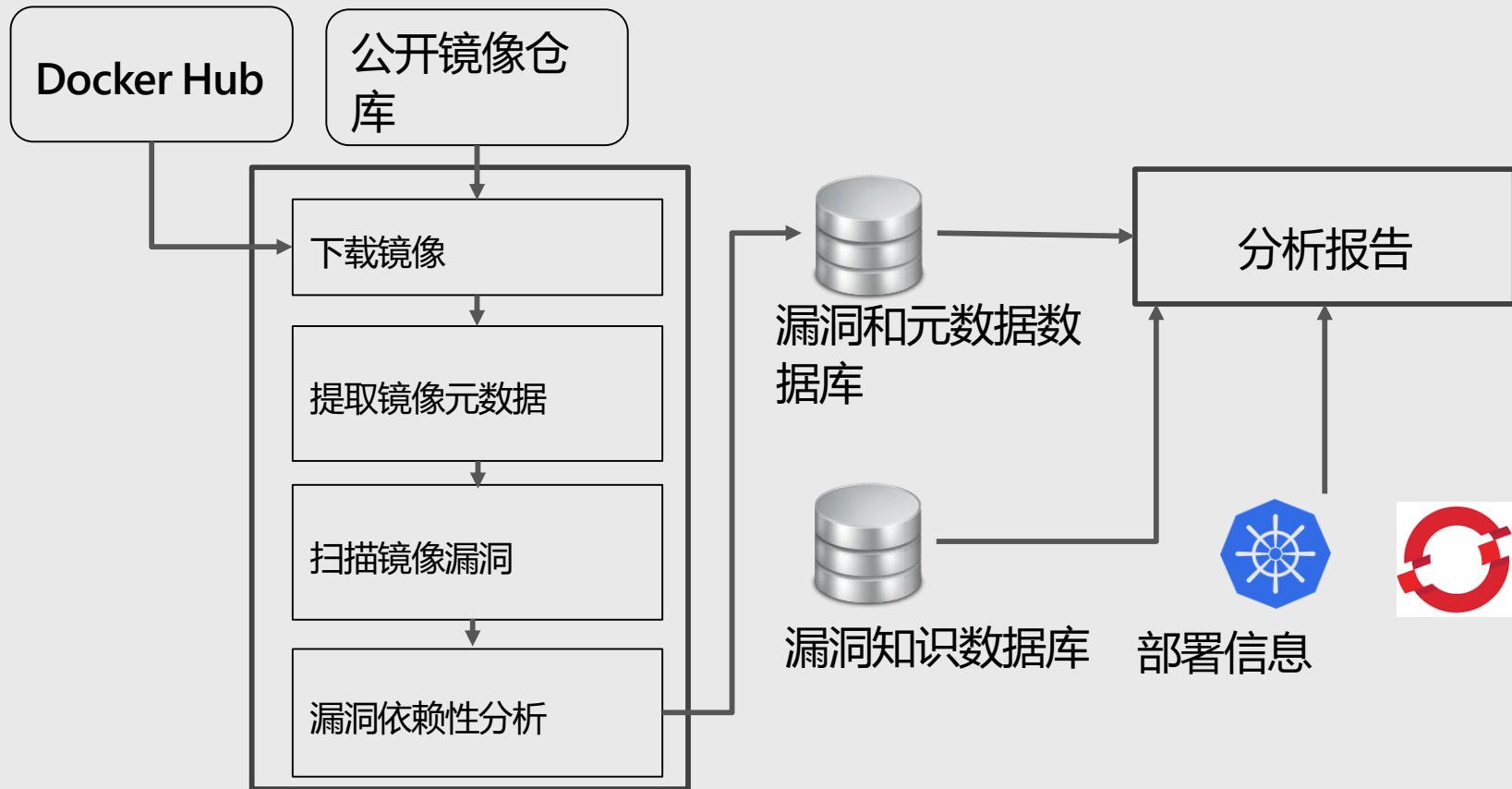
第 2 步：使用容器镜像扫描器扫描这些镜像，了解其中包含的漏洞信息。

第 3 步：使用我们的 VulnerDB 分析这些漏洞信息并公布有关这些漏洞的趋势状况。

# 数据源

来源	描述
DockerHub	可直接下载/爬网 Docker 镜像的镜像仓库
Clair	可提供系统级漏洞信息的开源漏洞扫描器
<a href="https://hub.docker.com">hub.docker.com</a>	提供公开访问的镜像扫描结果
Container Analysis API	Google Cloud 提供的镜像分析服务
GitHub 的 Dockerfile	通过 Pull 计数我们可以了解能够作为容器部署的开源项目的流程度

# 数据收集





# 数据收集

类型	数据字段	描述
镜像信息	镜像 ID	每个唯一镜像的 Sha 256 值
镜像信息	公开下载量计数	每个镜像的总下载次数
镜像信息	更新时间	每个镜像最后一次更新的准确日期时间
镜像信息	命令/Dockerfile	构建镜像需要运行的命令

## 数据收集 (续)

类型	数据字段	描述
漏洞信息	时间	首次报告时间和最后更新时间
漏洞信息	可利用性	该漏洞可以如何利用 (如net exploitable 等)
漏洞信息	严重程度评级	CVSS 3.0/2.0、PANW 分数、以往的签名触发器
漏洞信息	相关软件包	有漏洞镜像的名称和版本
部署信息	使用情况	K8s/OpenShift 部署中与镜像有关的自定义字段

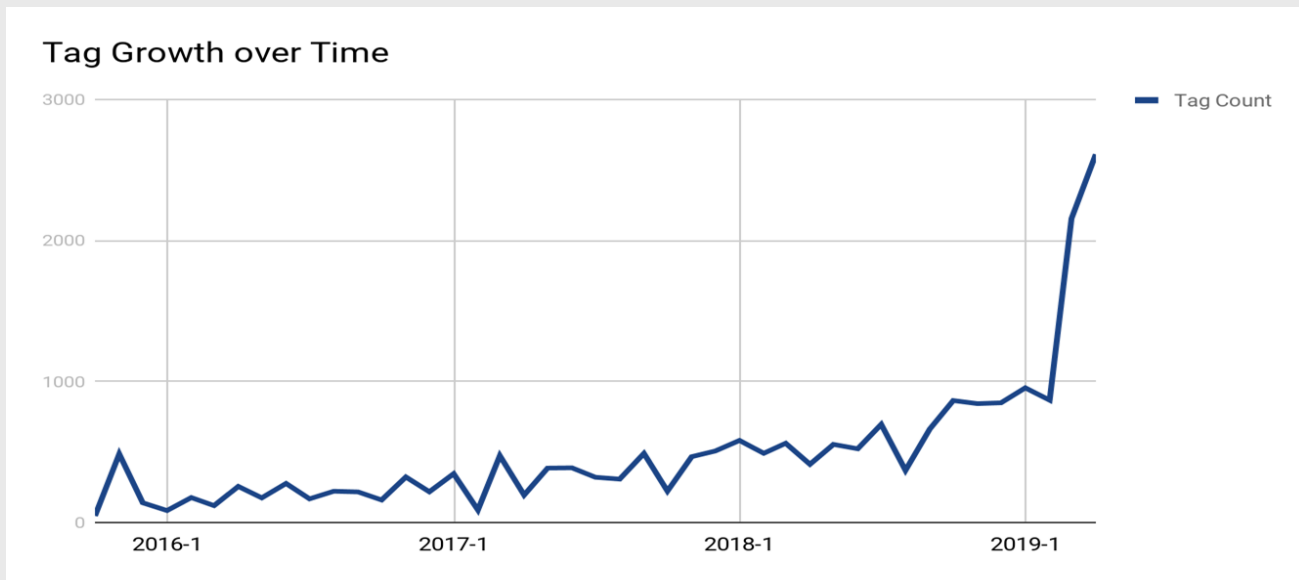
# 数据集基本状况统计

项目	统计结果	项目	统计结果
镜像仓库数量	151	镜像标签数量	22106
Dockerfile 数量	976	唯一漏洞数量	4259
发现有漏洞的镜像仓库所占比例 (最新结果)	81%	有漏洞镜像标签所占比例	69.1%



# 数据集基本状况统计 (续)

标签在长时间范围内的增长情况

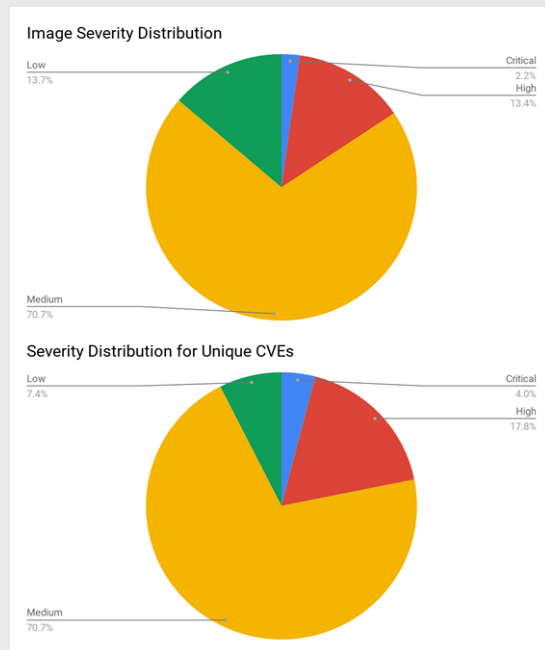
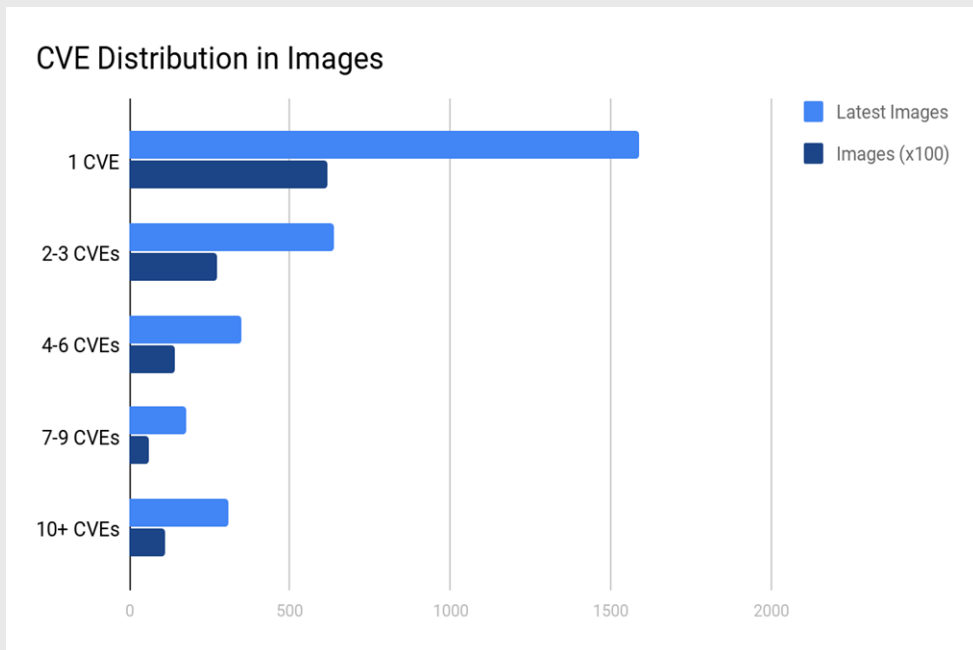


发现：流行的镜像有着数量众多的标签（发布），大部分仓库都有逐月发布计划。

# 议程

- 简介
- 基本统计
- 漏洞研究
- 集群中的镜像
- 实践建议

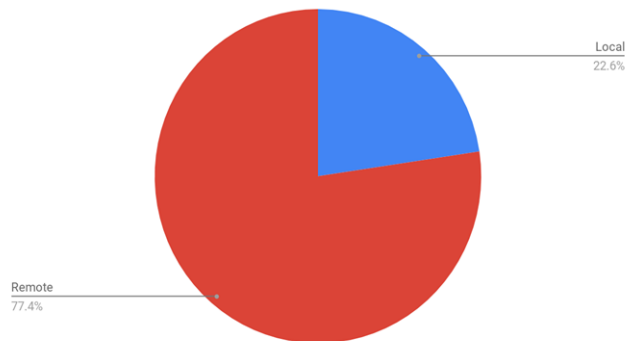
# 严重程度



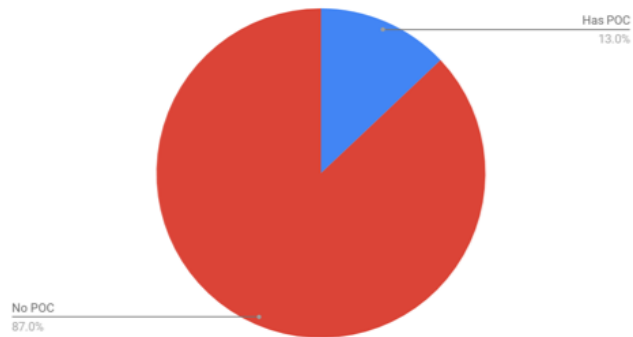
发现：大部分公开提供的镜像均存在漏洞。81% 的镜像至少包含一个漏洞。

# 可利用指数

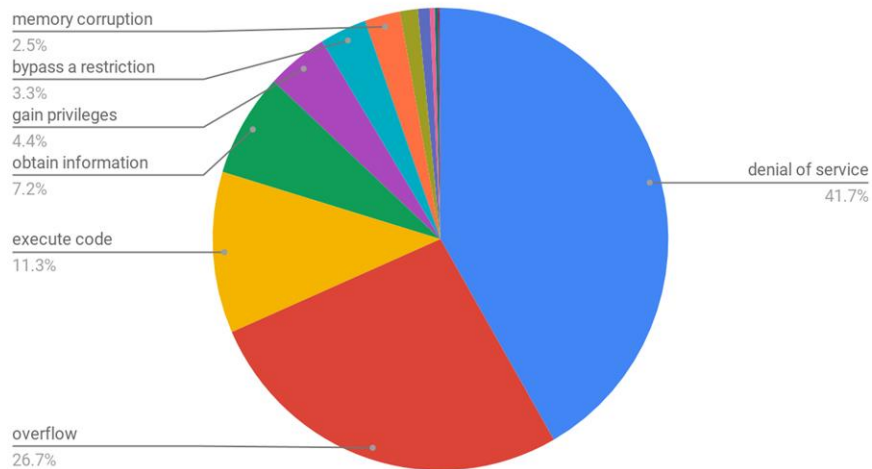
Access Vector Distribution



POC Analysis



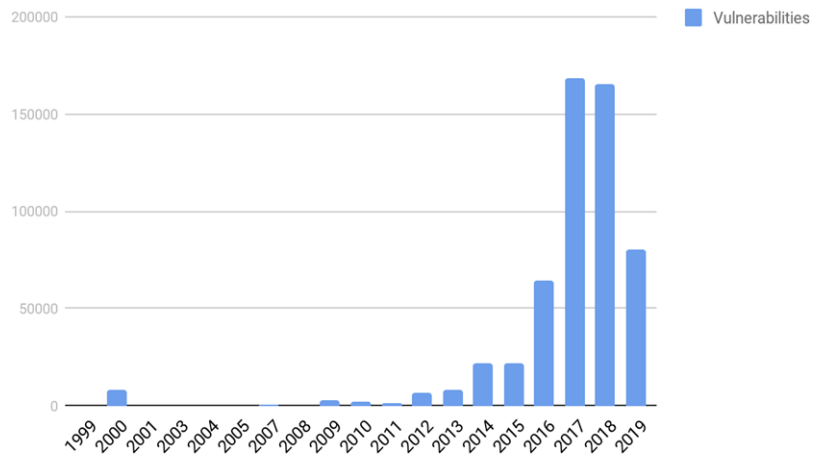
Vulnerability Type Distribution



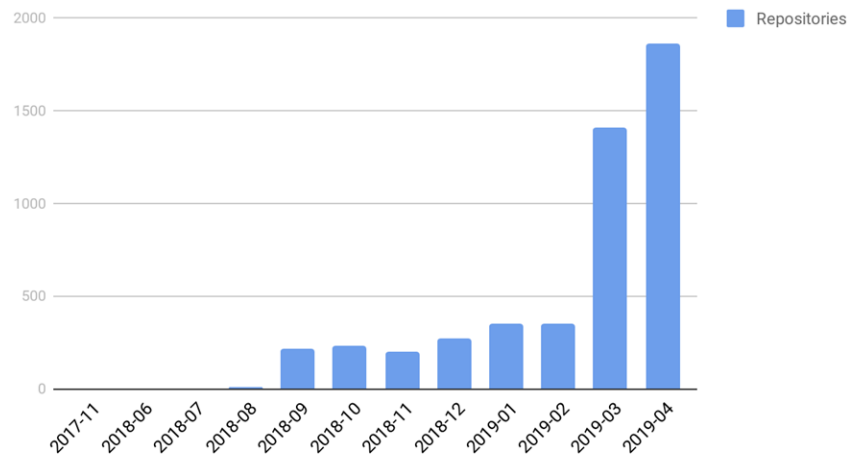


# 时间敏感度

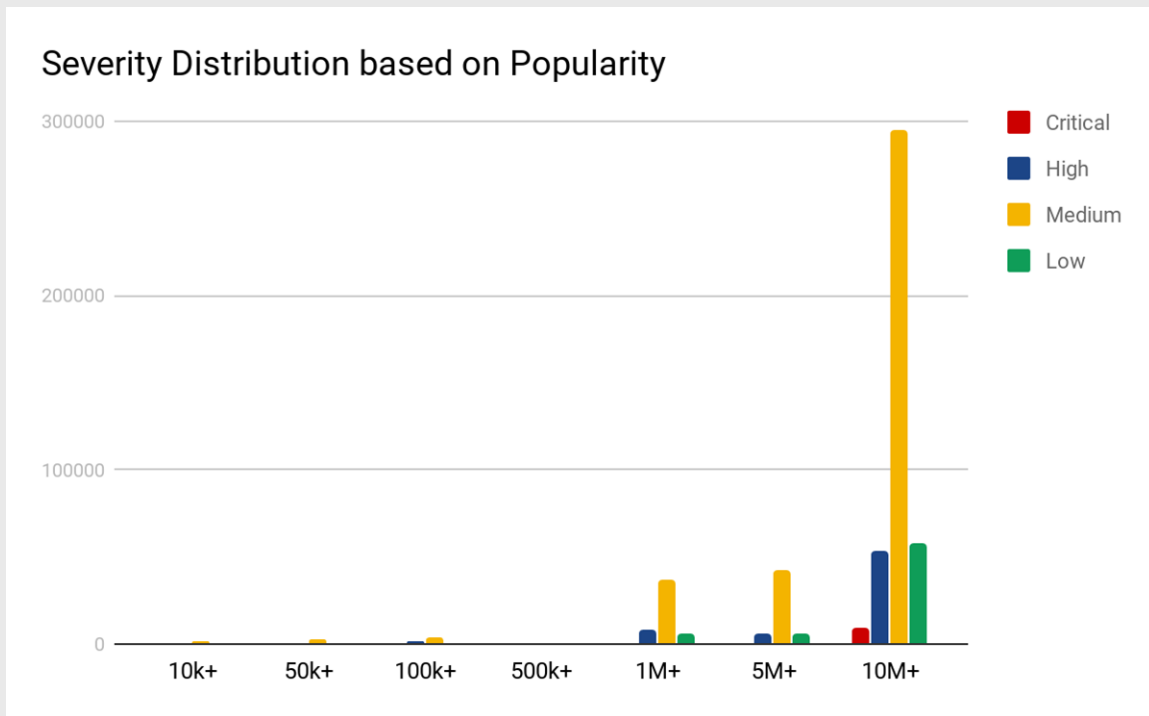
## Time Distribution for Vulnerability



## Time Distribution for Repository



# 流行度和漏洞之间的趋势



# 议程

- 简介
- 基本统计
- 漏洞研究
- 集群中的镜像
- 实践建议

# K8s 风险评估

在K8s pod 使用有漏洞的镜像时，有些注意事项需要重视。

因素	问题	危险范例
部署环境	有漏洞的镜像是否部署到生产用名称空间？	有漏洞的镜像部署到生产用名称空间，而非测试用名称空间。
Pod 特权	我们为与有漏洞镜像有关的 Pod 提供了哪些类型的特权？	使用高特权 Pod，或额外提供了非必须能力。
相关服务帐户	我们为有漏洞的 Pod/镜像关联了怎样的服务帐户？	所关联服务帐户具备 ef

# K8s 风险评估 (续)

在K8s pod 使用有漏洞的镜像时，有些注意事项需要重视。

因素	问题	危险范例
暴露的服务	有漏洞的 Pod 是否暴露出可从外部访问的服务？	Pod 被暴露至互联网上。
所连接的网络	Pod 被关联至那些虚拟/物理网络？	Pod 可访问互联网。 Pod 可连接至高特权 Pod。

# 范例

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: test sa
  automountServiceAccountToken: false
  ...
```

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
spec:
  serviceAccountName: test sa
  ...
```

```
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta
metadata:
  name: test sa binding
subjects:
  - kind: ServiceAccount
    name: test sa
    namespace: office
    roleRef:
      kind: ClusterRole
      name: deployment-manager
  ...
```

```
kind: Role
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  namespace: office
  name: deployment-manager
rules:
  - apiGroups: ["", "extensions", "apps"]
    resources: ["deployments", "replicasets", "pods"]
    verbs: ["get", "list", "watch", "create", "update"]
```

## 使用高特权服务帐户的 Pod

```
apiVersion: policy/v1beta
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']
    resourceName:
      - privileged
```

Define a Pod Security Policy

Use a Pod Security Policy

## High Privileged Pod Security Policy

# 缓解策略

解决方案	做法	优势	劣势
修补漏洞	安装补丁	从根本上解决问题	<ul style="list-style-type: none"><li>• 补丁并非总是立即可用。</li><li>• 难以修补运行中的容器。</li></ul>
替换基准镜像	更改 Dockerfile 中的基准镜像	易于实施	<ul style="list-style-type: none"><li>• 更安全的基准镜像并非总是可用。为保证稳定性，需要额外的测试。难以更改运行中的容器。</li></ul>
部署应用程序防火墙	部署应用程序级防火墙	<ul style="list-style-type: none"><li>• 防止在运行时被利用</li><li>• 易于实施</li></ul>	可能无法适用于所有漏洞。

↑ 优先级  
↓ 行动

# 议程

- 简介
- 基本统计
- 漏洞研究
- 集群中的镜像
- 实践建议



# 容器漏洞管理核查单

建议的阶段	核查项	操作
集成阶段	镜像中的漏洞	使用镜像扫描器扫描漏洞
集成阶段	可替换的基准镜像	如果基准镜像有漏洞，查找可替换并且更安全的基准镜像
集成阶段	定义风险条件	使用诸如 CVSS 分数、可利用向量等条件为漏洞定义风险级别

## 容器漏洞管理检查单（续）

建议的阶段	核查项	操作
交付阶段	风险评估	为您的名称空间/服务/Pod/部署定义风险耐受条件
交付阶段	部署策略	定义能将漏洞与部署风险要求相匹配的策略
交付阶段	策略强制实施	使用工具强制实施您的安全策略

# 容器漏洞管理核查单 (续)

建议的阶段	核查项	操作
运行阶段	扫描镜像	在运行中的容器内寻找新发现的漏洞
运行阶段	部署应用程序级防火墙	部署应用程序防火墙，获得及时更新的入侵防御能力
运行阶段	监视流量	使用服务网格策略检测 Pod 间的异常流量
运行阶段	监视主机	部署基于主机的入侵检测系统，防止基于主机的特权提升

# 问答



SHANGHAI 2019

# 我的容器安全吗？

有关容器漏洞的研究

Cecilia Hu, Yue Guan, Zhaoyan Xu

*Palo Alto Networks*