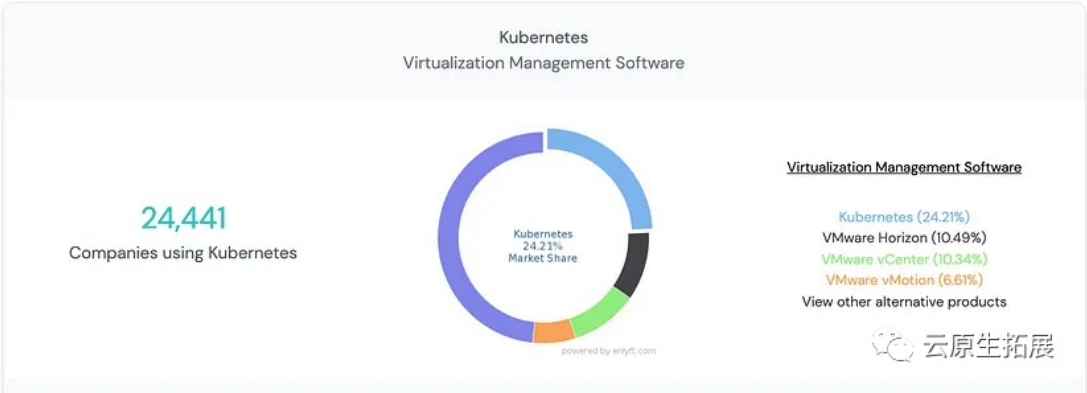
64Kubernetes 系列（五十九）12 个用于查找 Kubernetes 中的安全漏洞和错误配置的扫描器

Kubernetes 系列（五十九）12 个用于查找 Kubernetes 中的安全漏洞和错误配置的扫描器

欢迎关注我的公众号" **云原生拓展** "，原创技术文章第一时间推送。

Kubernetes 已经成为事实上的云操作系统。开发人员喜欢 K8s，因为 Kubernetes 使开发人员可以轻松地将他们的应用程序打包成可移植的微服务。

超过 90% 的处理云和微服务编排的公司都在转向 Kubernetes。有超过 24,441 家公司使用 Kubernetes。



您有没有想过您的容器是否真的安全或门是开着的?

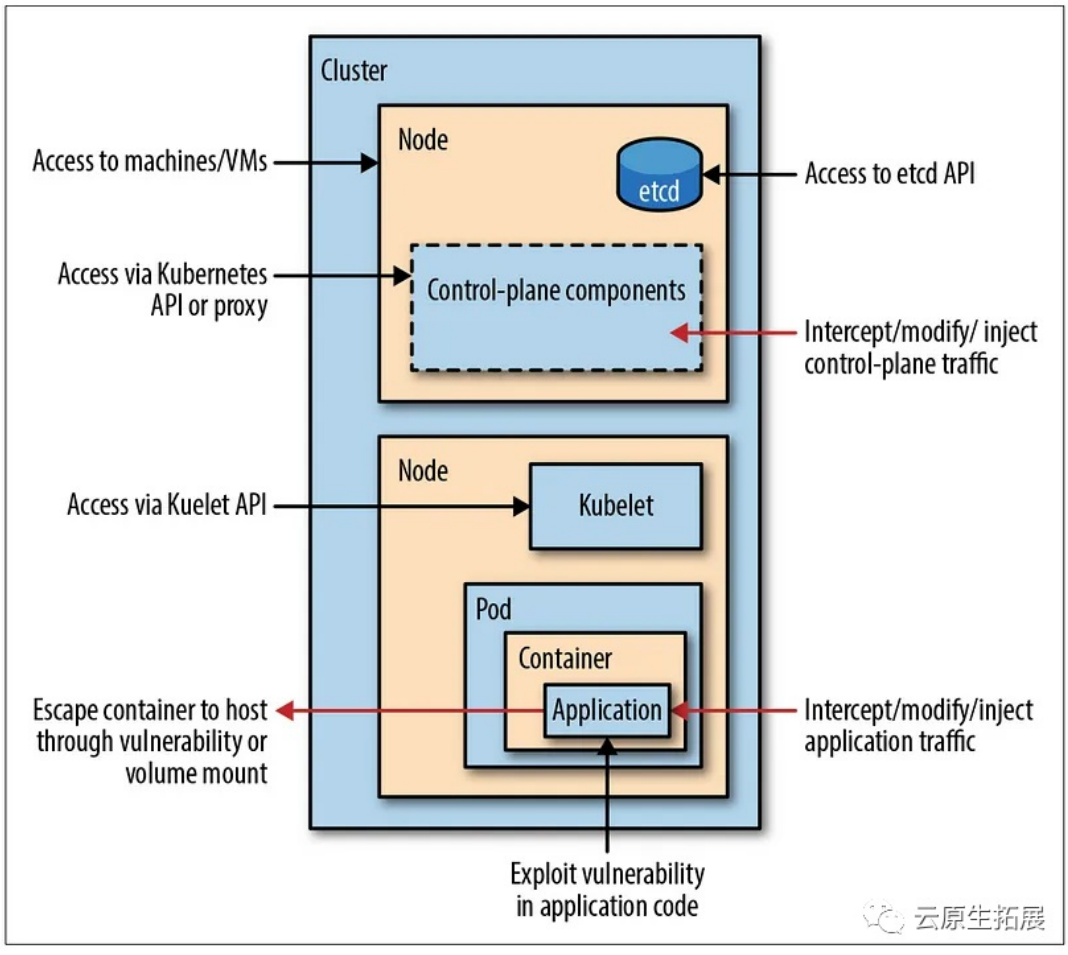
尽管 Kubernetes 提供了各种开箱即用的安全措施。通常还有其他原因会导致这些安全漏洞的出现。以下是示例：

• 配置错误：最近的一项研究发现，70–75% 的组织至少有一个严重的 AWS 安全配置错误。

• Kubernetes API Service：特斯拉在未经身份验证的情况下将作为其主要 Kubernetes API 服务一部分的仪表板在线上线时发现了这一点。

然而，Kubernetes 只有最基本的安全特性。它不包括 CIS"互联网安全中心"[4] 提供的所有检查。不幸的是，当涉及到管理员的复杂安全监控和合规性执行时，Kubernetes 不提供这种级别的保护。

如下图所示，攻击者可以通过多种方式尝试破坏您的 Kubernetes 集群及其上运行的应用程序。



我们不会在本文中查看 Kubernetes 集群中可能出现安全漏洞的所有原因。这将在其他文章中考虑。

我在最近的项目中收集了几个有用的扫描器来查找 K8S 中的漏洞和错误配置，并希望在本文中向您展示它们。

1. Kube-bench

Kube-bench 是一种扫描器，它通过尽可能准确地使用 CIS Kubernetes Benchmark 实施来检查 Kubernetes 是否安全实施。 Kube-bench 荣获 2018 年 InfoWorld 基础奖。

这是用 GO 编写的，它将通过该工具运行的测试输出通过、失败或警告消息以及 CIS Kubernetes Benchmark for Lookup 的适当部分。

快速开始

你可以运行 kube-bench：

• 容器运行:

```
docker run - pid=host -v /etc:/etc:ro -v /var:/var:ro -t aquasec/kube-bench:latest - version 1.18
```

• kubernetes 中运行:

```
# 提供的yaml 文件可用于测试作为 job 运行
kubectl apply -f https://github.com/aquasecurity/kube-bench/blob/main/job.yaml
```

2. Kube-hunter

Kube-hunter 顾名思义，它将寻找 Kubernetes 集群中的安全漏洞。它旨在提高 Kubernetes 环境中安全控制的意识和可见性。从集群外部，kube-hunter 扫描域或地址范围以查找与 Kubernetes 相关的开放端口，并测试使您的集群易受攻击者攻击的配置问题。

快速开始

可以通过三种不同的方式运行 kube-hunter，每种方式都提供了一种不同的方法来发现集群中的漏洞。

你可以运行 kube-hunter：

• 在任何机器上，选择远程扫描并指定您的 Kubernetes 集群的 IP 地址或域名。这将从攻击者的角度概述您的 Kubernetes 设置。

```
pip install kube-hunter
```

• 在集群中的一台机器上

```
docker run -it — rm — network host aquasec/kube-hunter
```

• 在集群内的 Pod 中

```
kubectl create -f https://github.com/aquasecurity/kube-hunter/blob/main/job.yaml
```

3. Kubeaudit

Kubeaudit 检测您的 Kubernetes 资源中的安全配置错误，并提供有关如何修复它们的提示。

Kubeaudit 附带了大量的"审核员"列表，可以测试各个方面，例如 pod 的 SecurityContext。审计员的完整列表可以在最后找到。

Kubeaudit 是一个命令行工具和一个 Go 包，用于审计 Kubernetes 集群的各种安全方面。

快速开始

您可以安装 kubeaudit：

• 通过 Homebrew

```
brew install kubeaudit
```

• 通过 Helm

```
helm upgrade — install kubeaudit secureCodeBox/kubeaudit
```

例子：

```
$ kubeaudit all -f "internal/test/fixtures/all_resources/deployment-apps-v1.yml"
---------------- Results for ---------------

  apiVersion: apps/v1
  kind: Deployment
  metadata:
    name: deployment
    namespace: deployment-apps-v1

----------------------------------------

-- [error] AppArmorAnnotationMissing
   Message: AppArmor annotation missing. The annotation 'container.apparmor.security.beta.kubernetes.io/container' should be added
   Metadata:
      Container: container
      MissingAnnotation: container.apparmor.security.beta.kubernetes.io/container

-- [error] AutomountServiceAccountTokenTrueAndDefaultSA
   Message: Default service account with token mounted. automountServiceAccountToken should be set to 'false' or a non-default ser

-- [error] CapabilityShouldDropAll
   Message: Capability not set to ALL. Ideally, you should drop ALL capabilities and add the specific ones you need to the add lis
   Metadata:
      Container: container
      Capability: AUDIT_WRITE
...
```

4. Kube-scan

使用 Kube-scan，您可以获得工作负载的风险评分。 Kube-scan 为每个工作负载给出从 0（无风险）到 10（高风险）的风险评分。 Kube-scan 旨在帮助您了解哪些工作负载面临的风险最大以及原因，并允许您优先更新 Pod 安全策略、Pod 定义和清单文件，以控制风险。

快速开始

• Kube-scan 是一个带有 YAML 文件的单 pod 部署：

```
-> kubectl apply -f https://raw.githubusercontent.com/octarinesec/kube-scan/master/kube-scan.yaml
-> kubectl port-forward — namespace kube-scan svc/kube-scan-ui 8080:80
```

• 然后你就可以开始了： http://localhost:8080

5.Kubesec

Kubesec 是一种开源工具，用于根据 YAML 配置评估 Kubernetes 工作负载的安全风险。

Kubesec 通过验证用于 Kubernetes 部署和操作的配置文件和清单文件来量化 Kubernetes 资源的风险。

快速开始

Kubesec 可用作：

• Docker 容器：

```
docker.io/kubesec/kubesec:v2 https://hub.docker.com/r/kubesec/kubesec/tags
```

• Linux/MacOS/Win 二进制文件： https://github.com/controlplaneio/kubesec/releases

• Kubernetes 准入控制器： https://github.com/controlplaneio/kubesec-webhook

• Kubectl 插件： https://github.com/controlplaneio/kubectl-kubesec

命令行用法示例：

```
kubesec scan k8s-deployment.yaml
```

6. Kube-score

Kube-score 对所有 Kubernetes 对象定义执行静态代码分析。输出是关于您可以改进哪些方面的建议列表，以使您的应用程序更安全、更有弹性。

快速开始

您可以通过以下方式轻松安装 kube-score：

• Docker: docker pull zegl/kube-score

• Homebrew： brew 安装 kube-score

• Krew 用于 kubectl 命令行工具的插件管理器：kubectl krew install score

例子：

```
kube-score score my-app/*.yaml
```

7. KubiScan

KubiScan 帮助集群管理员识别攻击者可以利用的权限来破坏集群。 KubiScan 在 Kubernetes 基于角色的访问控制 (RBAC) 权限模型中扫描 Kubernetes 集群以查找有风险的权限。 KubiScan 可以扫描包含特权服务帐户令牌的 pod，这些令牌可被滥用于特权升级攻击或破坏集群。

快速开始

您通过以下方式安装 KubiScan：

```
- alias kubiscan='python3 https://github.com/cyberark/KubiScan/blob/master/KubiScan.py to use kubiscanli
```

例子：

• 搜索具有特权帐户的 pod：

```
kubiscan -rp
```

• 验证此帐户是否出现在风险主题列表中：

```
kubiscan -rs
```

• 搜索此服务帐户具有的所有规则：

```
kubiscan -aaes "risky-sa" -ns "default" -k "ServiceAccount"
```

8. Krane

Krane 是一个用 Ruby 编写的命令行工具。 Krane 是一个简单的 Kubernetes RBAC 静态分析工具。它识别了 K8 的 RBAC 设计中潜在的安全风险，并就如何缓解这些风险提出了建议。 Krane 仪表板显示当前的 RBAC 安全状况，并允许您浏览定义。

快速开始

你可以安装 Krane vie：

```
gem install krane
```

进而：

```
krane deploy <app’s namespace> <kube context>
```

9. Illuminatio

Illuminatio 是一个用于自动测试 Kubernetes 网络策略的实用程序。只需运行 illuminatio clean run，illuminatio 将扫描您的 Kubernetes 集群以查找网络策略、创建适当的测试用例并执行它们以确定策略是否生效。

快速开始

您可以通过以下方式安装 illuminatio：

```
pip3 install illuminatio
```

• 或 Kubectl 插件：

```
-> ln -s $(which illuminatio) /usr/local/bin/kubectl-illuminatio
-> kubectl plugin list — name-only | grep illuminatio
```

测试您新创建的 NetworkPolicy：

```
illuminatio clean run
```

10. Checkov

Checkov 是用于基础设施即代码的静态代码分析工具。它扫描使用 Terraform、Terraform Plan、Cloudformation、AWS SAM、Kubernetes、Dockerfile、Serverless 或 ARM 模板部署的云基础设施，并通过基于图形的扫描检测安全性和合规性错误配置。

快速开始：

您可以通过以下方式轻松安装 Checkov：

• Python 包管理器：

```
pip3 install checkov
```

• Homebrew：

```
brew install checkov
```

运行：

```
checkov — directory /user/path/to/iac/code
```

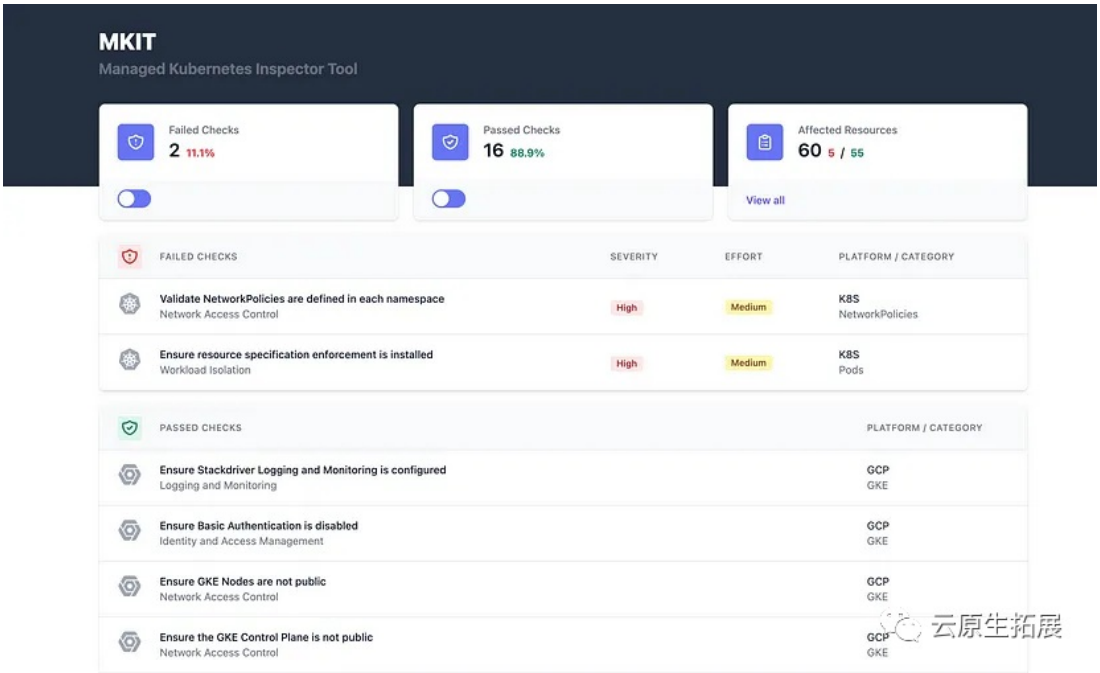
或一个或多个特定文件：

```
checkov --file /user/tf/example.tf
```

11. MKIT

Managed Kubernetes Inspection Tool (MKIT) 是 Managed Kubernetes Inspection Tool 的首字母缩写词。

MKIT 是一种托管 Kubernetes 检查工具，它使用 FOSS 工具来查询和检查托管 Kubernetes 集群对象的各种常见安全相关配置设置以及集群中运行的工作负载/资源。



12. Kubei

在 Kubernetes 集群中，Kubei 用于分析直接威胁。 Kubei 的大部分内容都是用 Go 编程语言编写的。

Kubei 是一个漏洞扫描和 CIS Docker 基准测试工具，允许用户对其 Kubernetes 集群进行准确和即时的风险评估。

快速开始

Kubei 也像 Kube-scan 一样有一个带有 YAML 文件的 signle pod 部署：

• 运行以下命令在集群上部署Kubei：

```
kubectl apply -f https://raw.githubusercontent.com/Portshift/kubei/master/deploy/kubei.yaml
```

然后，通过以下命令将端口转发到 Kubei webapp：

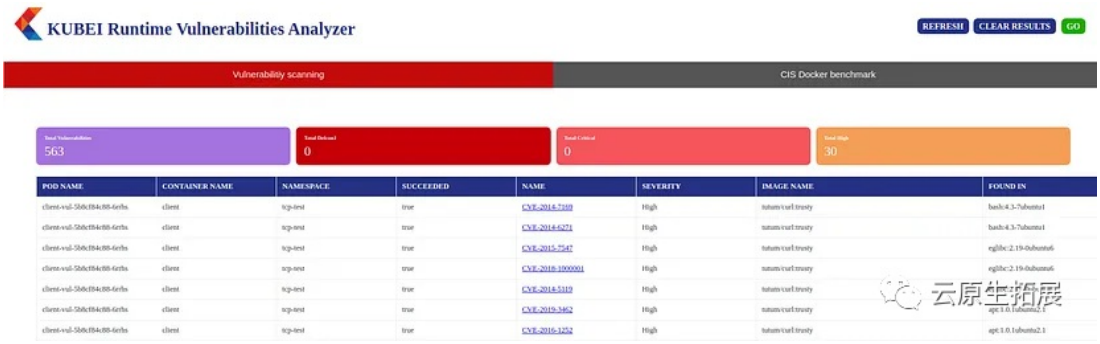
```
kubectl -n kubei port-forward $(kubectl -n kubei get pods -lapp=kubei -o jsonpath='{.items[0].metadata.name}') 8080
```

在您的浏览器中，导航至 http://localhost:8080/view/，然后单击"开始"以运行扫描。

要检查 Kubei 的状态以及正在进行的扫描的进度，请运行以下命令：

```
kubectl -n kubei logs $(kubectl -n kubei get pods -lapp=kubei -o jsonpath='{.items[0].metadata.name}')
```

刷新页面（ http://localhost:8080/view/）更新结果。



你应该使用哪个工具?

这实际上取决于用例。有一件事是肯定的，您应该深入研究这些工具并从中挑选一个或多个扫描仪来确保您现有的用例。大多数工具都提供监控服务，因此有机会将这些指标与您现有的监控服务集成在一起。这样，当对容器、pod、Ingress 和其他 Kubernetes 配置进行更改时，始终可以监控漏洞。

一般来说，你不应该依赖标准的 Kubernetes 安全性，就像 Tesla 和其他大型项目一样，它很快也会付之东流。

我希望我能为您的 Kubernetes 安全世界做出贡献。

资源

[1] 使用 Kubernetes 的公司： https://bit.ly/3DUvBtU

[2] https://bit.ly/3DKIDdz

[3] https://bit.ly/3ILrgwD

[4] https://www.cisecurity.org/benchmark/kubernetes/

[5] https://bit.ly/33gl8MI

[6] kube-bench： github.com/aquasecurity/kube-bench

[7] https://bit.ly/3DOnkYr

[8] Kube hunter： github.com/aquasecurity/kube-hunter

[9] Kubeaudit： github.com/Shopify/kubeaudit

[10] kubeaudit 审计员名单： https://bit.ly/3m1pb6a

[11] kube-scan： github.com/octarinesec/kube-scan

[12] Kubesec：kubesec.io & https://github.com/controlplaneio/kubesec