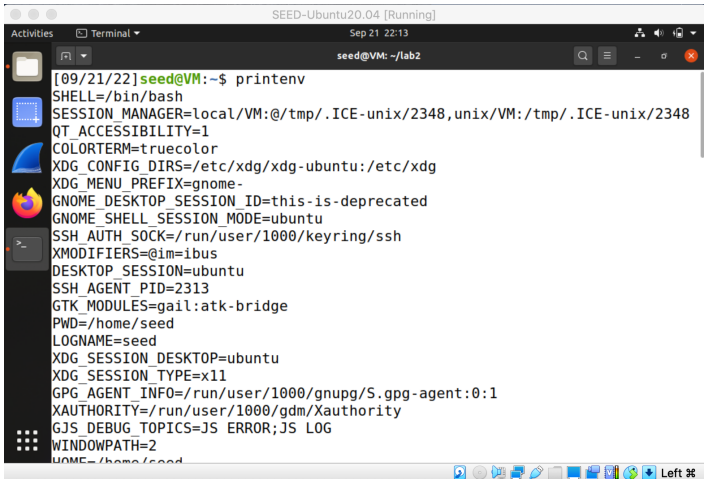
Noah Cunningham
Reese Pearsall
CSCI 476
9/21/22

Lab 2: Environment Variable and Set-UID Program Lab

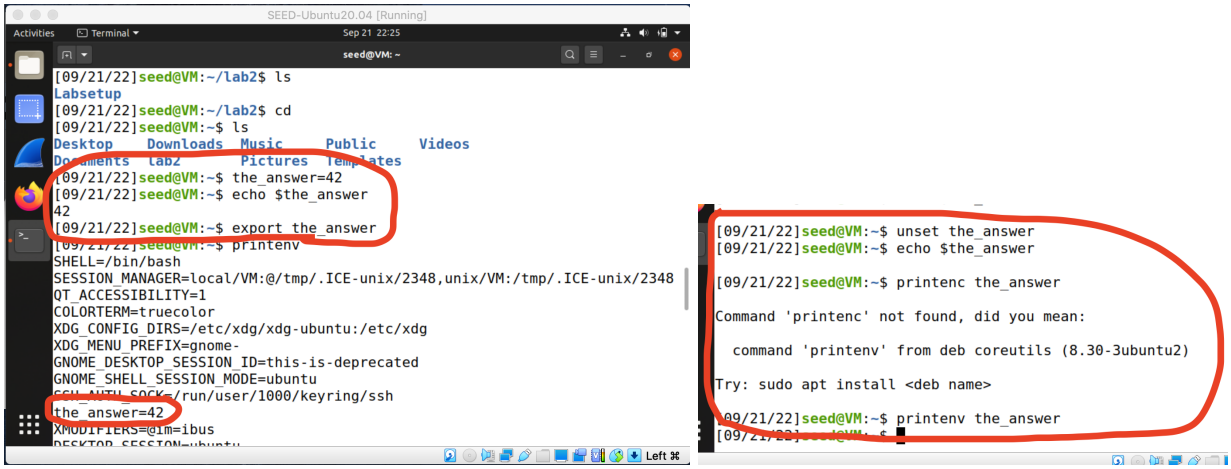
2.1)

Use printenv or env command to print out the environment variables.



I have used printenv to show the environment variables, we can see such things as the PWD and if i were to scroll down we could see the Home Directory and Username.

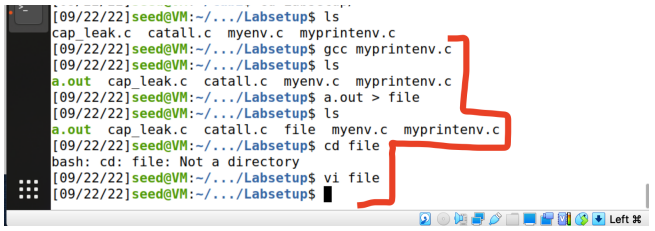
Use export and unset to set or unset environment variables.

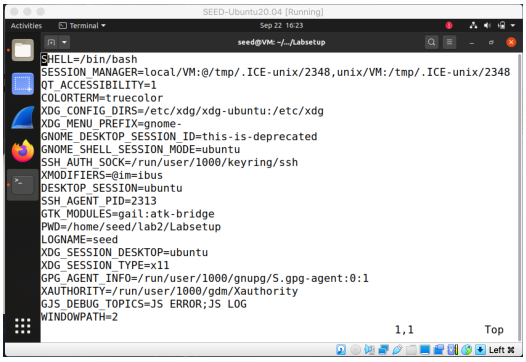


I have created a variable called the_answer and used the export command to make it an environment variable. I then unset the_answer which removes it from the environment variables.

2.2)

Step 1: On the left is me compiling myprintenv.c and saving a.out to a file called ‘file’. And on the right is me going into vi for ‘file’. Looks like the output is some environment variables

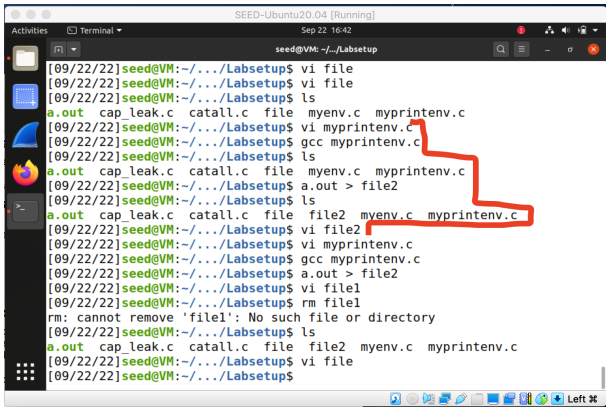


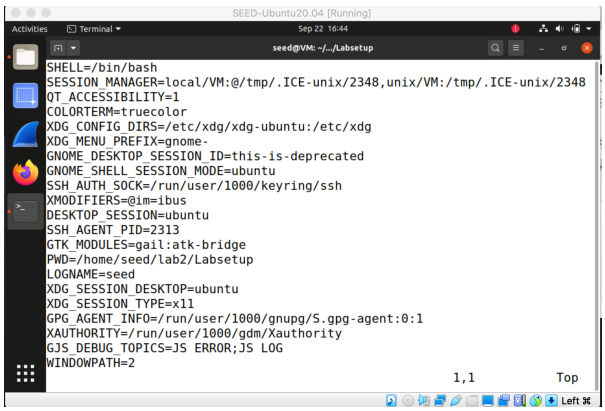



```
case 0:  /* child process */
 // printenv();
   exit(0);
default:  /* parent process */
   printenv();
   exit(0);
```

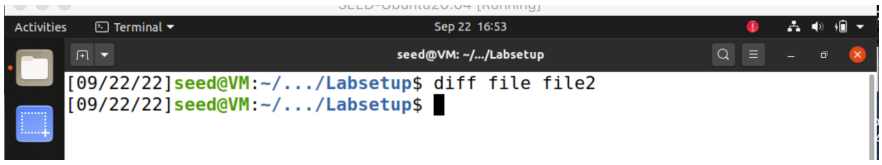
Step 2: I modified the code as seen on the right

I then compiled myprintenv.c and saved a.out to ‘file2’. When looking in ‘file2’ at first glance it seemed like nothing has changed, and everything went the exact same as the first time.



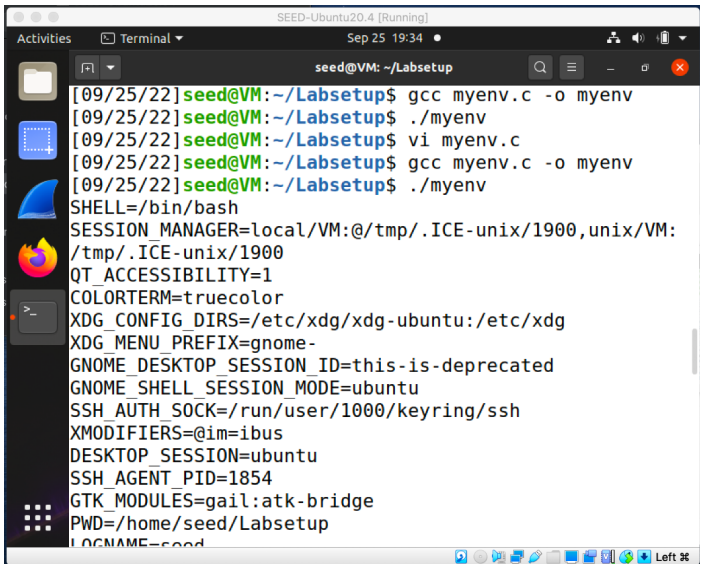


Step 3: When running diff it doesn't output anything showing that they are the same



2.3)

Step 1 and 2: Initially there was no output, but when i edited the file to include “execve("/usr/bin/env", argv, environ);”

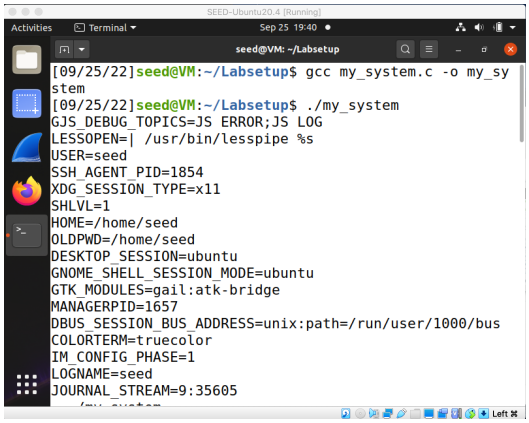


Step 3: When we call **execve(const char *_filename_, char *const _argv_[], char *const _envp_[]);** and pass in the environ variable that inherits the parent processes environment.

2.4)

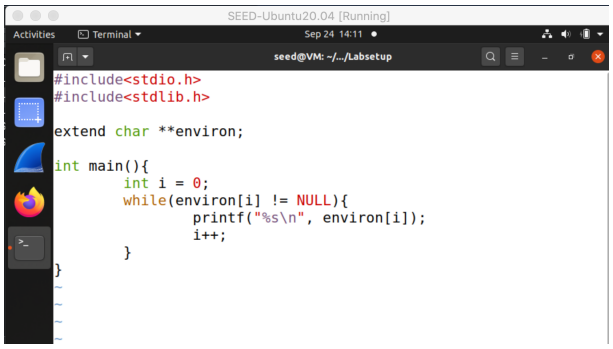
Here is my_system.c

And here's me compiling and running it

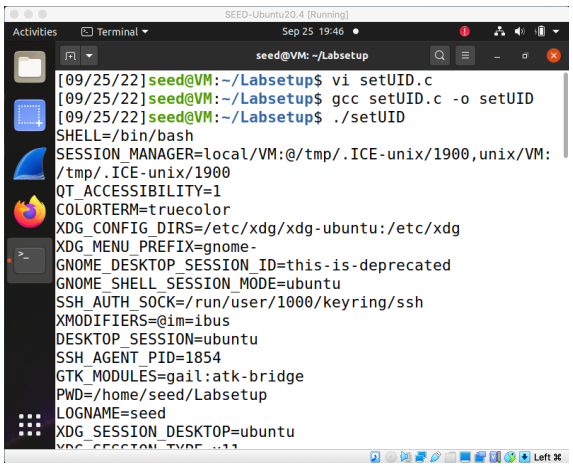


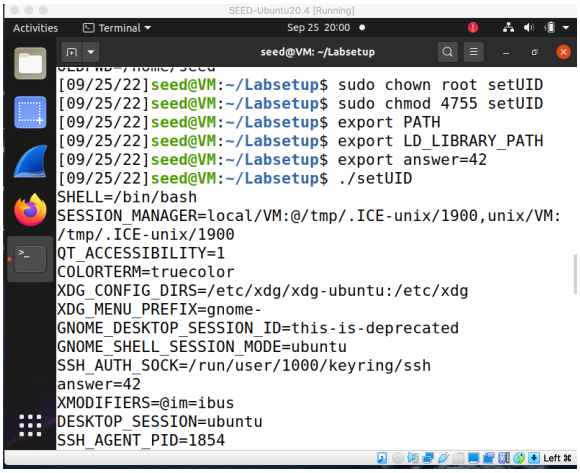
2.5)

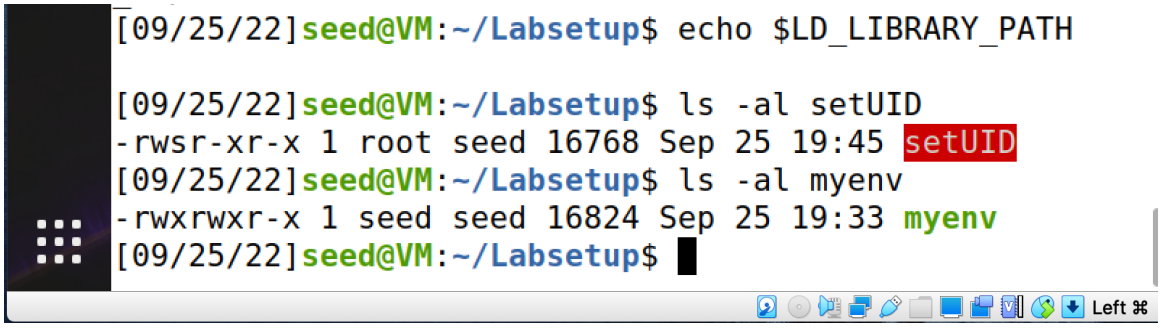
Step 1: setUID.c on



Step 2 and 3: Me compiling and running setUID.c and me changing it ownership and making it a Set-UID program. Then running it again

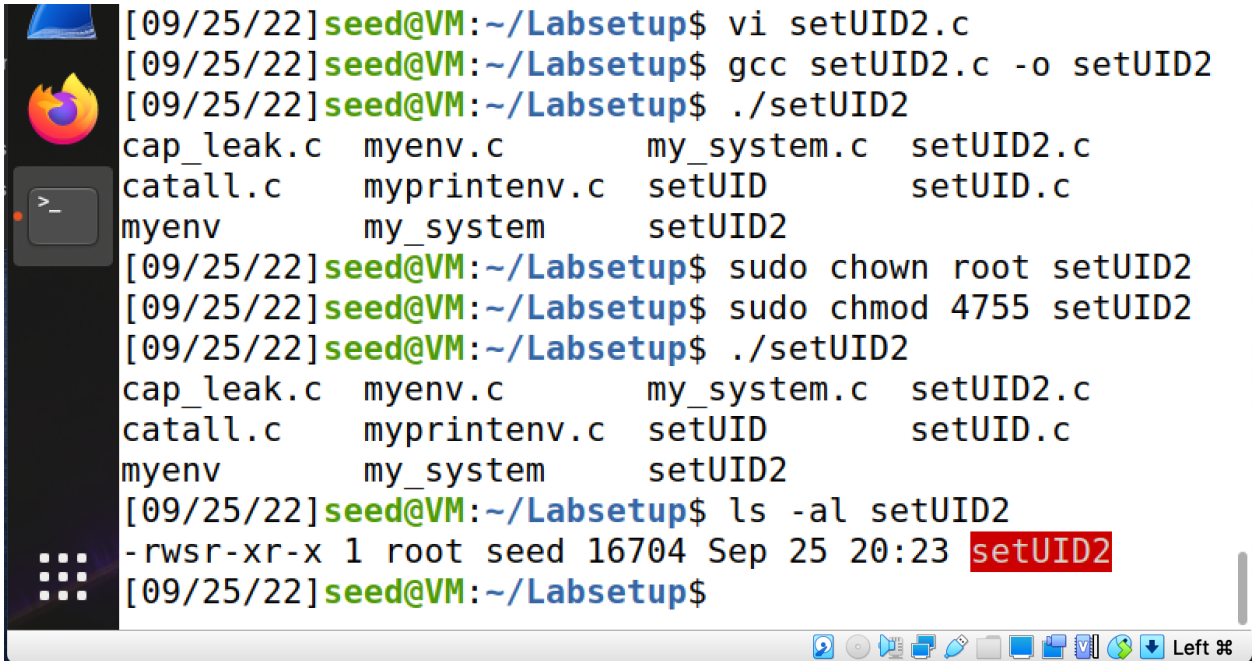






My observations are that the answer is now a part of the environment variables, but path looks the same on both iterations and I can't find the environment variable LD_LIBRARY_PATH, even when I echo there is no output. As well as i cant find any indication of it being a root and Set-UID program in the environment variables. When I do “$ ls -al setUID” it does say it is a root file and a Set-UID, also the filename is highlighted in red as well.

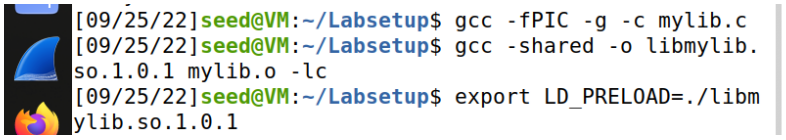
2.6)



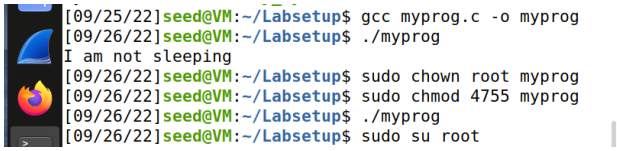
I was able to run setUID2 after changing it to a root Set-UID program and the output was the exact same as before. And when I look at the permissions of the file it says confirms that is in fact a root Set-UID program.

2.7)

Step 1 and 2:



- Condition 1 and 2



- Condition 3 and 4

```
[09/26/22]seed@VM:~/Labsetup$ sudo su root
root@VM:/home/seed/Labsetup# export LD_PRELOAD=./libmy
lib.so.1.0.1
root@VM:/home/seed/Labsetup# ./myprog
I am not sleeping
root@VM:/home/seed/Labsetup# exit
exit
[09/26/22]seed@VM:~/Labsetup$ sudo chown seed myprog
[09/26/22]seed@VM:~/Labsetup$ su otherUser
Password:
$ export LD_PRELOAD=./libmylib.so.1.0.1
$ ./myprog
I am not sleeping
$
```

I observe that it doesn't print ‘I am not Sleeping’ in the 2nd Condition.

Step 3: What do you mean by designing an experiment?

From what I can tell the reason it doesn't print in condition 2 is because since it's a Set-UID root program we can run it unless we are root.

So maybe i'm missing something so I tried to run a couple different examples but don't know if that constitutes an “experiment”.

So I set myprog to a seed account and switched to root and did LD_PRELOAD and then ran myprog. It printed ‘I am not Sleeping’.

I then unset LD_PRELOAD still in root and ran myprog and it printed ‘I am not Sleeping’. So root might be accessing the LD_PRELOAD in the seeds account because myprog is owned by seed and since root is above seed it has permission to access that.

I then exited root changed the permission of myprog to root and then re entered root and tried to run it and it didn't print. I think it did not print because it LD_PRELOAD was unset in root so and since myprog is now owned by root it does not access seeds LD_PRELOAD

I then exited root and changed the permissions of myprog back to seed and unset LD_PRELOAD and nothing printed.

I then exported LD_PRELOAD again and ran myprog and it printed.

So idk if i've figured out much more by doing this but it seems LD_PRELOAD must be set to run myprog and if its a Set-UID program and you are using an equivalent or greater access user you can run myprog if LD_PRELOAD is set somewhere where the account can access it.

Note: i can provide pictures of all these steps if necessary but i didn't want to make this doc 5 more pages and i feel like i did an accurate job at describing what i did

2.8)

Step 1:

```
[09/26/22]seed@VM:~/Labsetup$ gcc catall.c -o catall
[09/26/22]seed@VM:~/Labsetup$ sudo chown root catall
[09/26/22]seed@VM:~/Labsetup$ sudo chmod 4755 catall
```

```
[09/26/22]seed@VM:~/Labsetup$ ./catall
Please type a file name.
[09/26/22]seed@VM:~/Labsetup$ bob
```

I am not really sure what file I should try to remove. I ran catall and it printed 'please type a file name' but then it exited me back to seed. I checked the permissions on the files in Labsetup before and after and nothing seemed to change. But i theorize Bob could compromise the system because he could give a bad input to system. But catall didn't let me give a input so idk

Step 2:

```
[09/26/22]seed@VM:~/Labsetup$ vi catall.c
[09/26/22]seed@VM:~/Labsetup$ gcc catall.c -o catall
[09/26/22]seed@VM:~/Labsetup$ sudo chown root catall
[09/26/22]seed@VM:~/Labsetup$ sudo chmod 4755 catall
[09/26/22]seed@VM:~/Labsetup$ ./catall
Please type a file name.
[09/26/22]seed@VM:~/Labsetup$
```

Seems like nothing changed from running it and such but i theorize that since use use execve you can give it the same bad input or remove files. But from my observations nothing has changed so idk.