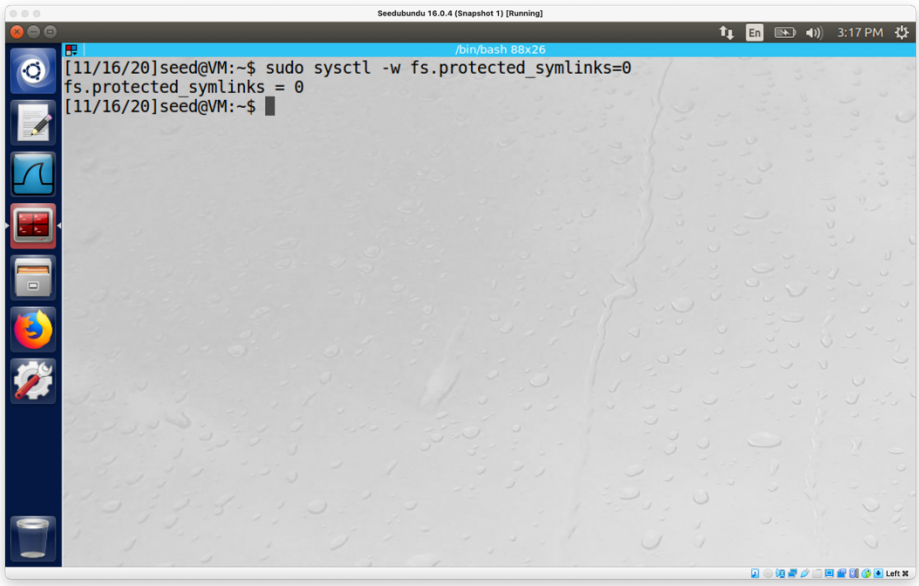
Man Tik Li
CS 377 - 001
November 16, 2020

Lab 4

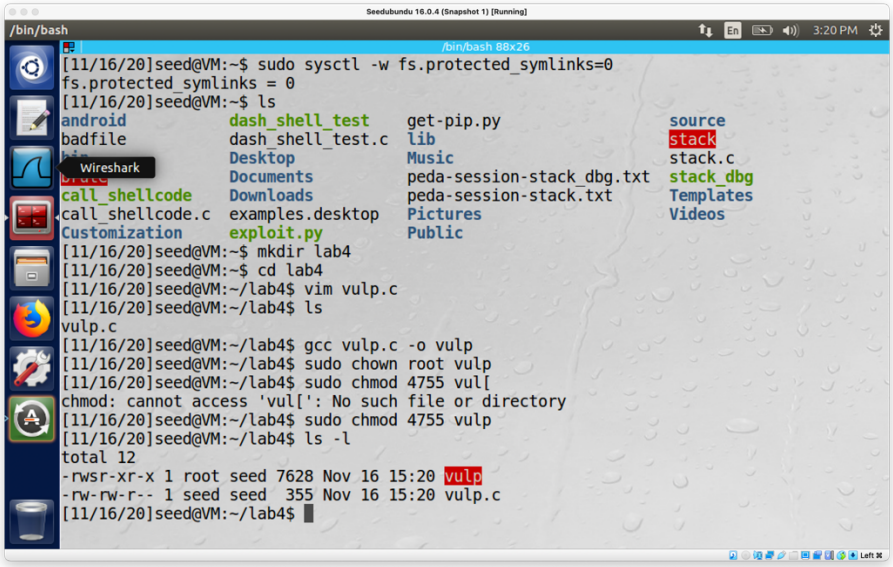
Initial Setup

On Ubuntu 16.04, use the following: $ sudo sysctl -w fs.protected_symlinks=0



Created vulp.c in "lab4" folder. Set vulp to Set-UID program:

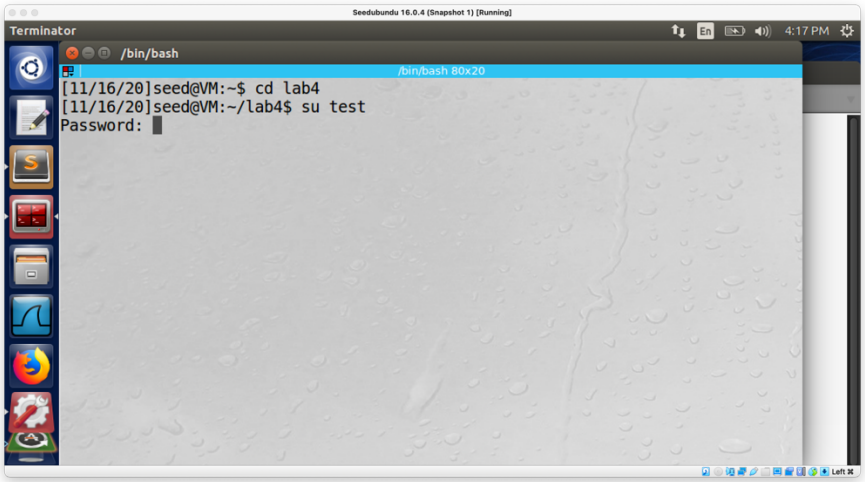
$ gcc vulp.c -o vulp
$ sudo chown root vulp
$ sudo chmod 4755 vulp



Task 1

After edited the passwd file, the test user appear in passwd.

```
root@VM:/etc# cat /etc/passwd | grep test
test:U6aMy0wojraho:0:0:test:/root:/bin/bash
root@VM:/etc#
```

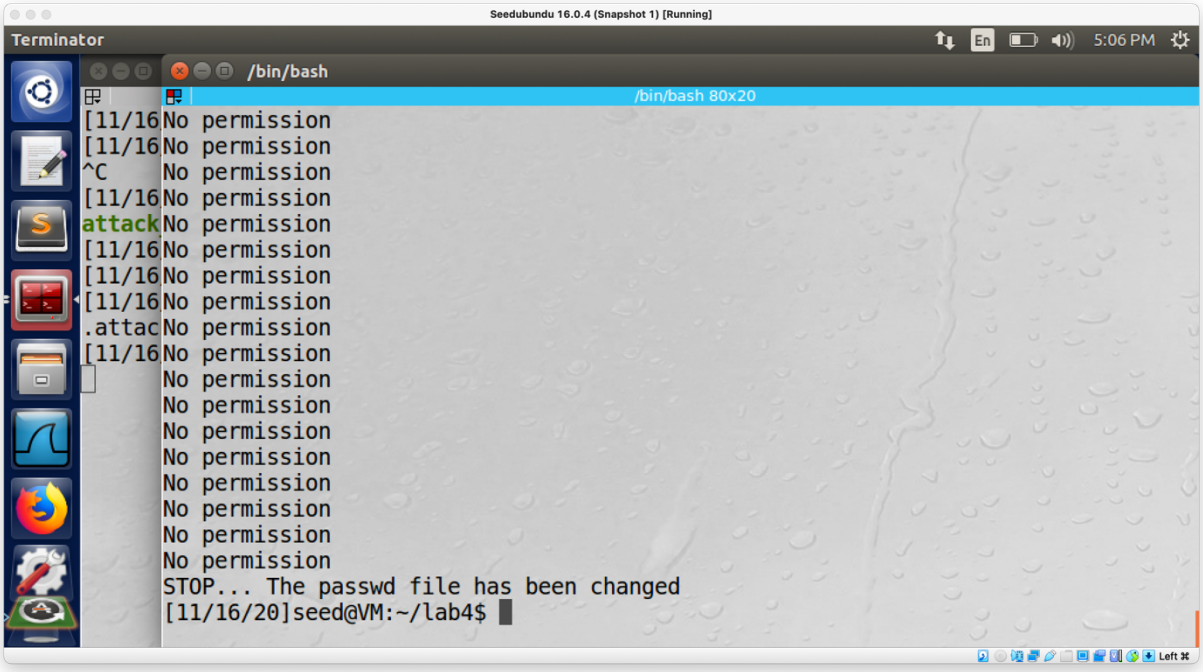


After I hit enter, I can enter to the root without password.

```
[11/16/20]seed@VM:~/lab4$ su test
Password:
root@VM:/home/seed/lab4#
```

Task 2

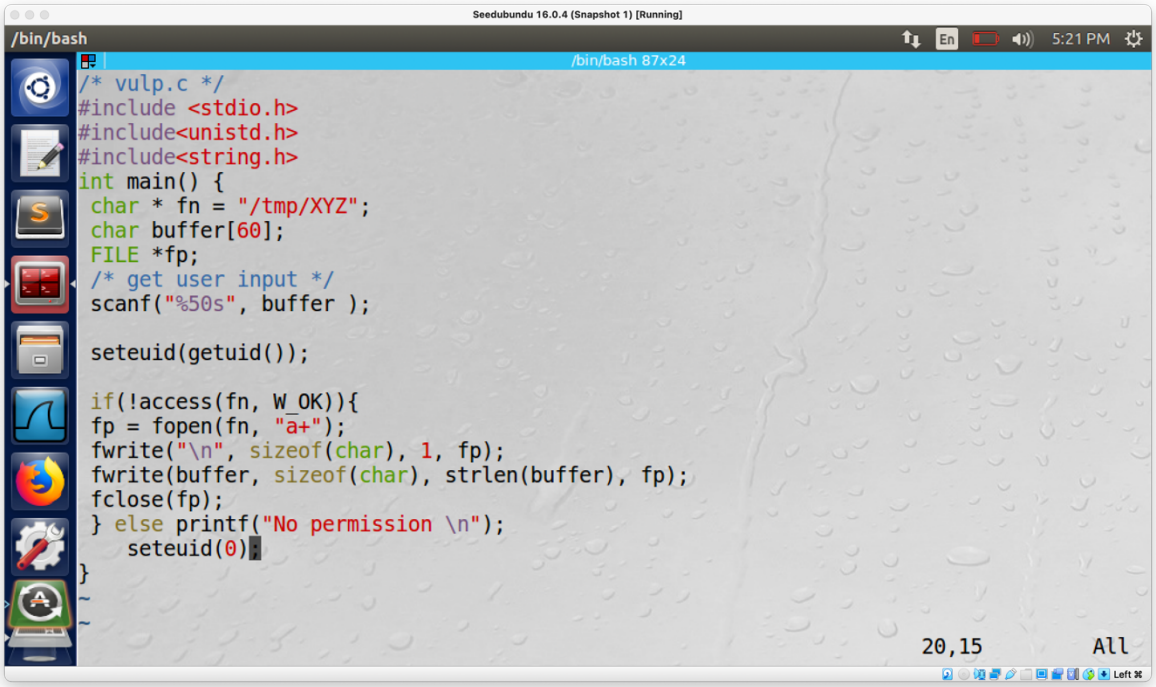
```
[11/16/20]seed@VM:~/lab4$ vim attack_process.c
[11/16/20]seed@VM:~/lab4$ gcc -o attack_process attack_process.c
[11/16/20]seed@VM:~/lab4$ ./atack_process
bash: ./atack_process: No such file or directory
[11/16/20]seed@VM:~/lab4$ ./attack_process
```



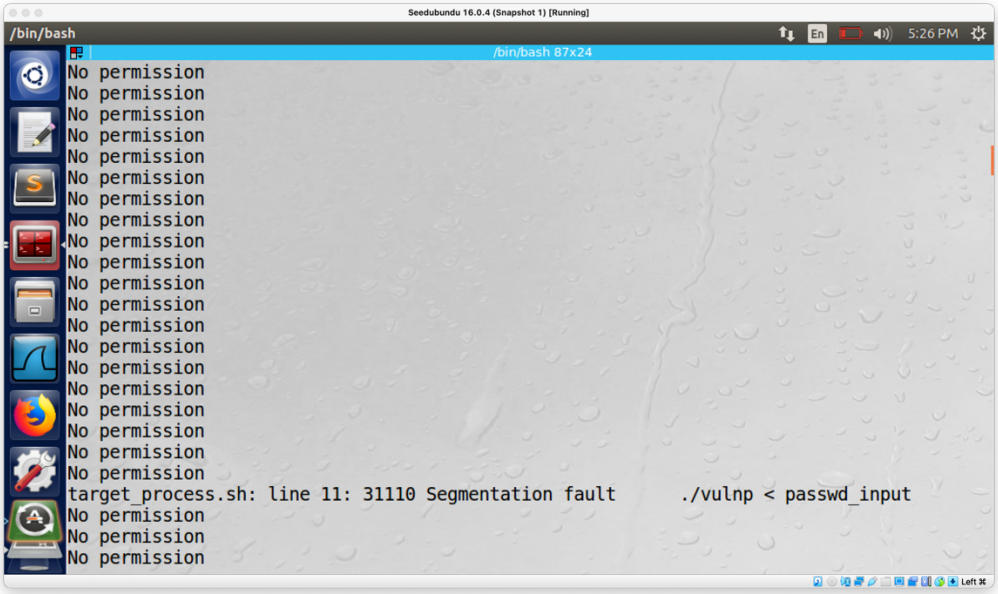
After running the attack_process and target_process.sh. The program successfully changed the paswd file.

```
[11/16/20]seed@VM:~/lab4$ cat  /etc/passwd | grep test
rambo:U6aMy0wojraho:0:0:test:/root:/bin/bash
```

Task 3

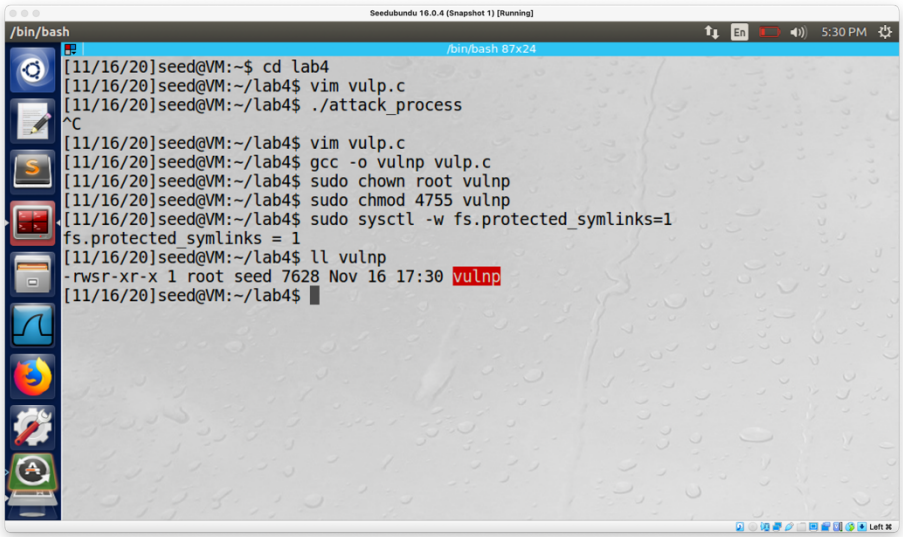


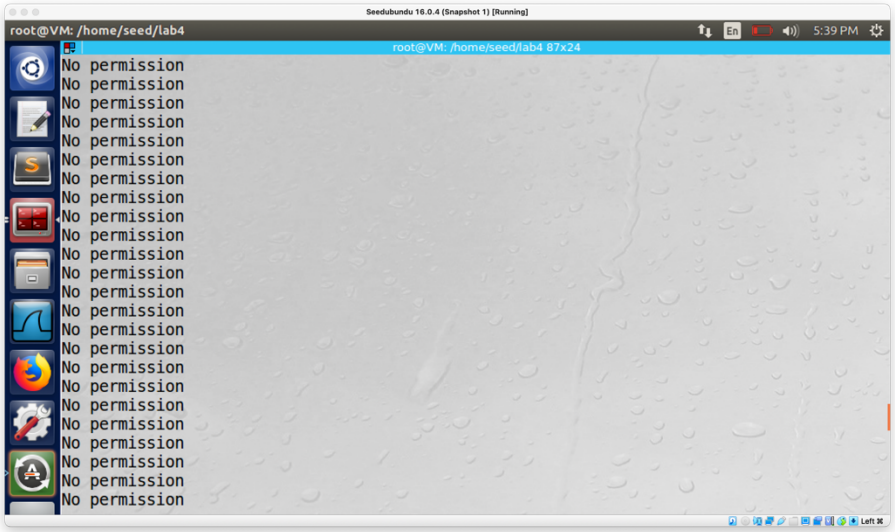
Edited the vulnerable program by use the real UID and effective UID of the program. We set the effective UID to be the same as real UID. Then it drops the privileges and at the end of the program we change the effective UID to be the same as before to gain the privileges back.



Result: The attack is not successful. I received segmentation fault, because the seteuid to currently limit the privilege. I can't use normal user to edit root permission files, which reults in segment fault.

Task 4





The attack is not successful, I received no permission in the output and the program keeps on running in a loop for a while.

1. Here does this protection scheme work?
   The race condition vulnerability exploited involved symbolic link inside the '/tmp' directory. Therefore, preventing programs from following symbolic links under specific conditions might overcome this vulnerability. In this lab, since the program ran with root privilege and the /tmp directory is also owned by the root, the program will not be allowed to follow any symbolic link that is not created by the root. Also, the symbolic link was created by the attacker which was actually the seed account. Hence, this link will not be followed, and that lead to a crash.

2. What are the limitations of this scheme?
   The scheme does not stop the race condiiotn from taking place but hinders it from causing damages. Therefore, it is a good mechanism for access control that allowed only the intended users to access the file but did not really pause the race condition from taking place.