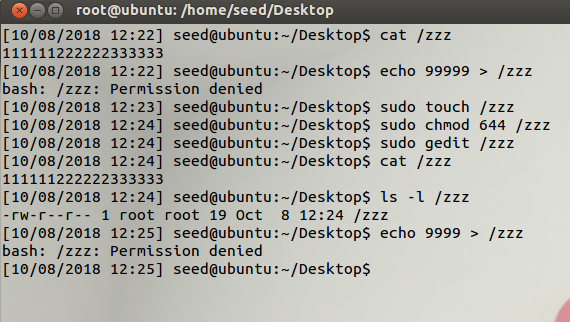
**Dirty COW Attack Lab**

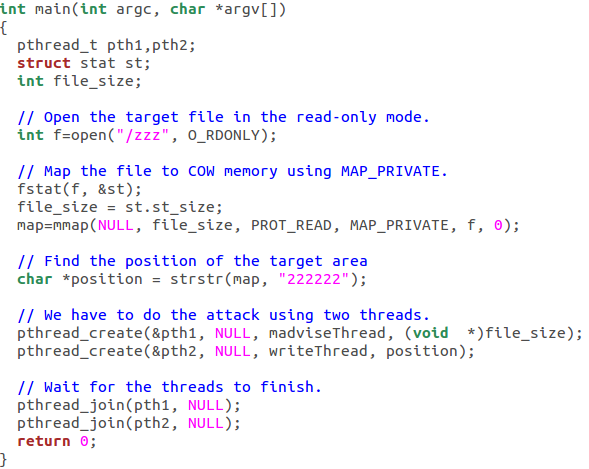
**Karan Amrutesh**

**Task 1: Modify a Dummy Read-Only File:**

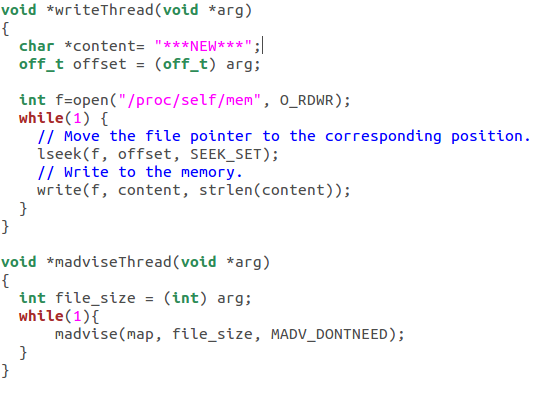
* Creating a Dummy File: We create a root owned file /zzz and change its permission to read only for normal users.
* We can see that when we try to write to this file as a normal user, it is not successful as its read only for normal users.



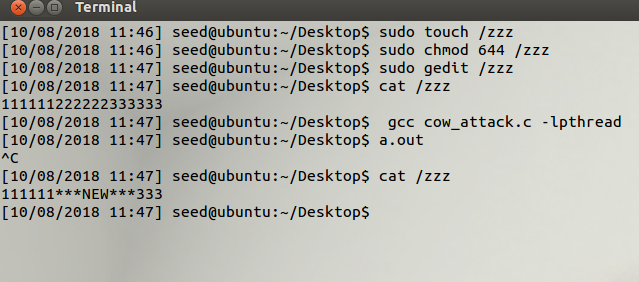
* In the main we open the file in read only mode, find the position of the characters we want to replace using the strstr() method which is “222222” here.



* Then we give the content to be replaced as \*\*\*NEW\*\*\*. The write thread will try to write into the read only file continuously.
* The madviseThread will keep discarding the private copy of the mapped memory continuously to race against the write thread so that the write happens to the original file.

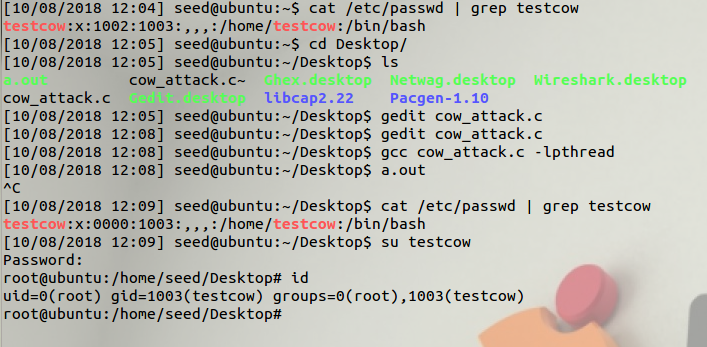


* After compiling and running the program for a few seconds, we c print out the contents of the /zzz file.
* We can see that the file has been modified even though it was read only implying that the Dirty COW attack was successful.

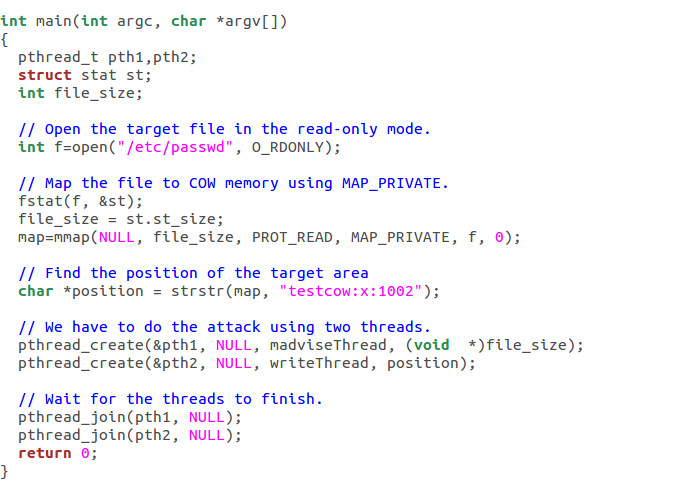


**Task 2: Modify the Password File to Gain the Root Privilege**

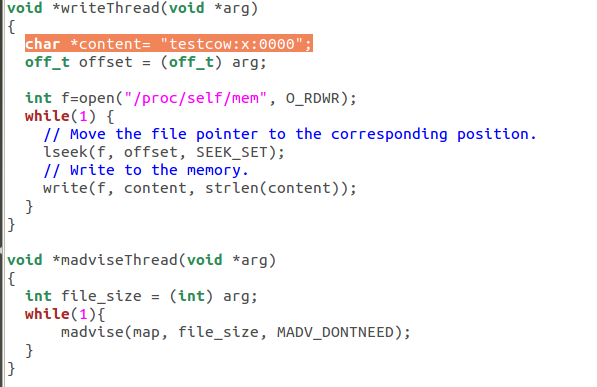
* We add a new user named testcow. We can see that the user id is 1001. We will perform the dirty COW attack shown in Task 1 on the /etc/passwd file to change this id to 0 to gain root privilege.



* We open the /etc/passwd file in read only mode and find the string “textcow:x:1002”



* We replace the user id with 0000 to indicate that it is the root’s user id.



* After running the program for a few seconds, we can see that the /etc/passwd file has been modified and logging into the user testcow will give us a root shell.
* By printing out the id, we can see that the user id is 0 which is the root user id.

