Dylan Aegbuniwe

CSC 472

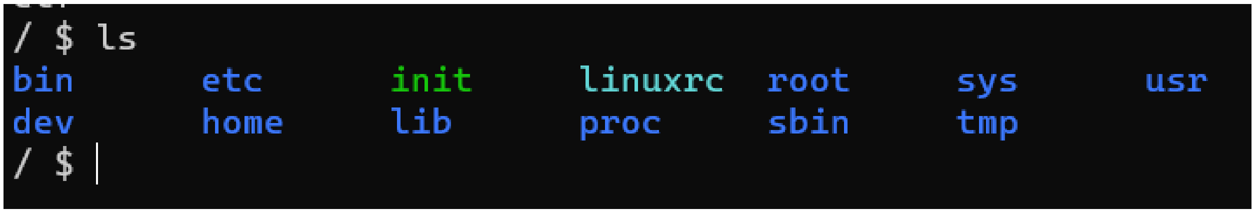
Lab 5: Kernel Exploitation

12/17/2021

Introduction

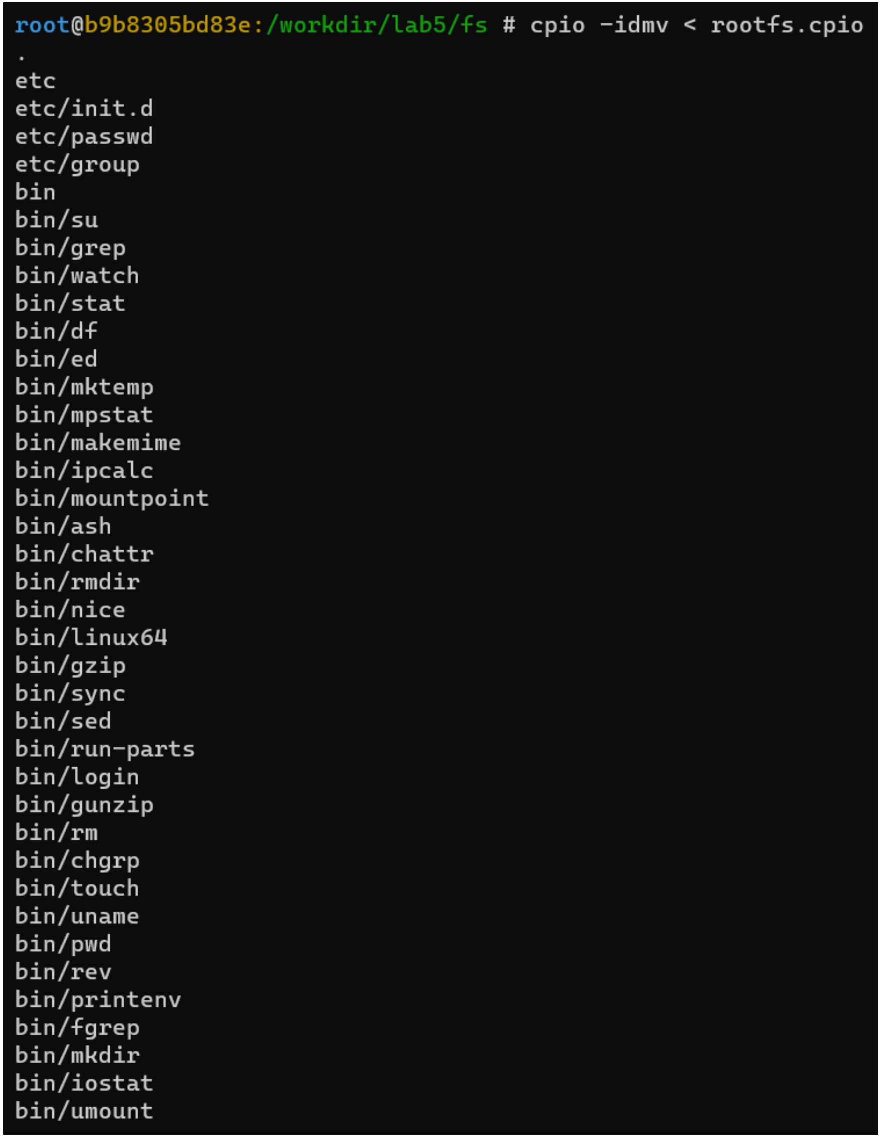
In this lab, we are using kernel exploitation to exploit a vulnerability with UAF.

Analysis and Results

1. 

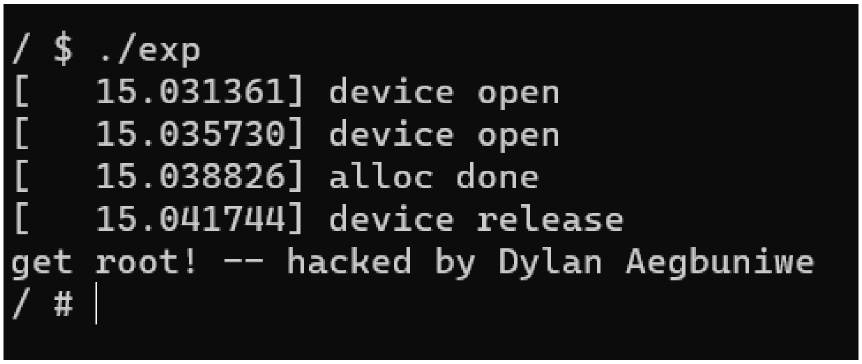
```
/ $ ls
bin       etc       init      linuxrc   root      sys       usr
dev       home      lib       proc      sbin      tmp
/ $ |
```

The directory has 12 different folders inside of it along with two other files.

2. 

```
root@b9b8305bd83e:/workdir/lab5/fs # cpio -idmv < rootfs.cpio
.
etc
etc/init.d
etc/passwd
etc/group
bin
bin/su
bin/grep
bin/watch
bin/stat
bin/df
bin/ed
bin/mktemp
bin/mpstat
bin/makemime
bin/ipcalc
bin/mountpoint
bin/ash
bin/chattr
bin/rmdir
bin/nice
bin/linux64
bin/gzip
bin/sync
bin/sed
bin/run-parts
bin/login
bin/gunzip
bin/rm
bin/chgrp
bin/touch
bin/uname
bin/pwd
bin/rev
bin/printenv
bin/fgrep
bin/mkdir
bin/iostat
bin/umount
```

Many more outputs follow these ones.

3. 

```
/ $ ./exp
[   15.031361] device open
[   15.035730] device open
[   15.038826] alloc done
[   15.041744] device release
get root! -- hacked by Dylan Aegbuniwe
/ #
```

4. The second babydev will overwrite the first allocated space, because babydev_struct is global.
5. The ioctl function passes fd1, 0x1001, and 0xa8 through by calling on babyioctl() that passes the different variables through kfree, kmalloc. 0xa8 is chosen because it is the size for kmalloc and babydev_struct.device_buf_len.
6. The data is being written after being read with babyread(), followed by babywrite().

Discussion and Conclusion

This lab satisfied the purpose of using a kernel exploitation to hack into a shell and see the results of it. Kernel is used with babydrivers inside the code to create a UAF exploit.