CSC 472/583 Fall 2021 Lab 5

Dr. Si Chen

Kernel Explotation

The goals of this lab:

- Understanding the concepts of kernel exploitation
- Exploiting a UAF vulnerability in kernel land

Objectives and Targets

Target 1: Boot Up QEMU

In this lab, we're going to use QEMU – A generic and open source machine emulator and virtualizer to launch the kernel exploitation. You can use our Badger CTF system to complete this project.

Steps:

1) Create a new folder for this lab

```
mkdir lab5
cd lab5
```

2) Download the provided linux image from our course website.

```
wget https://www.cs.wcupa.edu/schen/ss2021/lab/lab5.tar
```

3) Unzip the compressed file

```
tar -xvf lab5.tar
```

4) It'll release three files: *boot.sh, bzImage and rootfs.cpio.* To boot up the linux kernel, type the following command:

```
./boot.sh
```

Please answer the following question(s):

Question1 (1 point): how many folders are there inside the root (/) folder?

Target 2: Tweaking the Default File System

To add files into the default file system, we need to unpack and then repack the *rootfs.cpio* file. Please first quit the QEMU system by typing

```
exit
```

Now we're back to the BadgerCTF (/workdir/lab5). In order to add files to the default file system, we need to figure out how to unpack/repack the rootfs.cpio.

```
mkdir fs
cd fs
cp ../rootfs.cpio ./
mv rootfs.cpio rootfs.cpio.gz
gunzip rootfs.cpio.gz
cpio -idmv < rootfs.cpio
```

Please answer the following question(s):

Question2 (2 points): Please take a screenshot and show me the output after typing the command: cpio -idmv < rootfs.cpio

You can see that the file system of rootfs.cpio is now unpacked into the **fs** folder.

Target 3: Compile and Execute Kernel Exploitation Shellcode

Let's download the source code of our kernel exploitation shellcode by typing (inside fs folder):

```
wget https://www.cs.wcupa.edu/schen/ss2021/lab/exp.c
```

Please use Vim or other editor to change the YOUR_NAME inside **puts(”get root! – hacked by YOUR_NAME”);** to your own name

```
vim exp.c
```

Then, we can compile it using gcc, by typing:

```
gcc exp.c -static -o exp
```

We need to repack the exp program into the linux file system rootfs.cpio, please typing the following command (inside the /workdir/lab5/fs folder)

```
find . | cpio -o −−format=newc > rootfs.cpio
```

We need to replace the old rootfs.cpio with our new one

```
cp rootfs.cpio ../rootfs.cpio
```

Then we need go back to the lab5 folder and boot up our linux kernel:

```
cd ..
./boot.sh
```

This time, under the root (/) folder of this virtual machine, you'll find the exp program we just compiled. *Please answer the following question(s):*

Question3 (2 points): (Inside the QEMU linux virtual machine) Please take a screenshot of the output after typing the command: ./exp

Target 4: Understand UAF

Please read **exp.c** file and answer the following questions:

Question4 (2 points): In the shellcode (exp.c) why we want to open the device (/dev/babydev) twice?
Question5 (2 points): In the shellcode (exp.c), what's the meaning of ioctl(fd1, 0x1001, 0xa8)? Why use 0xa8?
Question6 (1 point): In the shellcode (exp.c), what's the meaning of write(fd2, zeros, 28)?

More...

Please check lecture video (kernel explotation)

Submission

- The lab due date is available on our course website. Late submission will not be accepted;
- The assignment should be submitted to D2L directly.
- Your submission should include: A **detailed project report in PDF format** to describe what you have done, including screenshots and code snippets.
- **No copy or cheating is tolerated**. If your work is based on others', please give clear attribution. Otherwise, you **WILL FAIL** this course.