CSC 472 Fall 2024: Lab 5

Dr. Si Chen

Kernel Exploitation

The goals of this lab are:

- To understand the concepts of kernel exploitation
- To exploit a UAF (Use-After-Free) vulnerability in kernel space

Objectives and Targets

Target 1: Boot Up QEMU

In this lab, we will use QEMU – a generic and open-source machine emulator and virtualizer – to launch the kernel exploitation. You will use our Badger CTF system to complete this project.

Steps:

1) Create a new folder for this lab:

```
mkdir lab5
cd lab5
```

2) Copy the provided Linux image from our lab folder:

```
cp /workdir/ss2024/lab5/lab5.tar ./
```

3) Unzip the compressed file:

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

4) This will 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):

Question 1: 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 Badger CTF (/workdir/lab5). To add files to the default file system, we need to figure out how to unpack and 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):

Question 2: 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 copy the source code of our kernel exploitation shellcode by typing (inside the fs folder):

```
cp /workdir/ss2024/lab5/exp.c ./
```

Please use Vim or another editor to change YOUR_NAME inside **puts("get root! – hacked by YOUR_NAME");** to your own name:

```
vim exp.c
```

Then, 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 type 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 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):

Question 3: (Inside the QEMU Linux virtual machine) Please take a screenshot of the output after typing the command: './exp'.

Target 4: Understand UAF

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

Question 4: In the shellcode (exp.c), why do we want to open the device (/dev/babydev) twice?
Question 5: In the shellcode (exp.c), what's the purpose of ioctl(fd1, 0x1001, 0xa8)? Why use 0xa8?
Question 6: In the shellcode (exp.c), what's the meaning of write(fd2, zeros, 28)?

Additional Resources

Please check the lecture video on kernel exploitation.

Submission

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