第三次实验

1.Linux 内核源码的编译与安装（可以在包括 Ubuntu 等任何 Linux 操作系统上完成）

(1) 通过命令查看所用系统当前内核版本号;

```
wxt@wxt-virtual-machine:~$ uname -r
5.4.0-26-generic
wxt@wxt-virtual-machine:~$
```

(2) 从 https://www.kernel.org/下载 4.18.16 内核源码， 手动编译;

```
wxt@wxt-virtual-machine:~$ wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.9.14.tar.xz
--2020-12-13 23:02:30--  https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.9.14.tar.xz
Resolving cdn.kernel.org (cdn.kernel.org)... 151.101.109.176, 2a04:4e42:1a::432
Connecting to cdn.kernel.org (cdn.kernel.org)|151.101.109.176|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 115556784 (110M) [application/x-xz]
Saving to: ‘linux-5.9.14.tar.xz.1’
```

下的最新的5.9.14...

```
su
tar -xvf linux-5.9.14.tar.xz
cd linux-5.9.14
make menuconfig
make -j8
make modules_install
make install
update-grub2
```

(3) 安装新内核映像

见（2）

(4)重启系统， 加载新内核， 通过命令查看内核版本号， 验证内核更新是否成功。

```
wxt@wxt-virtual-machine:~$ uname -r
5.9.14
```

2.系统调用的实现与添加

(1) 在下载的 4.18.16 内核中添加一个 hellosys 系统调用， 其功能为创建一个特定的文件，并在其中打印一条由调用者传入的一行字符串;

修改系统调用表

修改目录下`arch/x86/entry/syscalls/syscall_64.tbl`文件，为`hellosys`分配一个新的系统调用号（用来唯一标识每一个系统调用的编号，服务例程则是内核具体实现系统调用功能的函数，以`sys_`的格式命名），每个系统调用在该系统调用表中占一个表项，具体格式为：

<系统调用号><commmon/x32/64><系统调用名><服务例程入口地址>

```
333     common  io_pgetevents           sys_io_pgetevents
334     common  rseq                    sys_rseq
335     64      hellosys                sys_hellosys
```

声明系统调用服务例程原型

修改`include/linux/syscalls.h`文件，服务例程的原型声明格式为：

```
asmlinkage long sys_系统调用名(参数)
```

```
asmlinkage long sys_hellosys(const char __user *str, unsigned int len);
```

实现系统调用服务例程

修改目录下文件`kernel/sys.c`，实现系统调用的服务例程，新版本的内核中引入了宏`SYSCALL_DEFINEN(sname)`对服务例程的原型进行了封装（为了防止利用漏洞入侵），其中N是系统调用所需参数的个数，sname则是系统调用名+系统调用各参数，中间以`,`分割

```
SYSCALL_DEFINE2(hellosys, char __user *, str, unsigned int, len)
{
        struct file *f=NULL;
        loff_t offset=0;
        unsigned int ret=0;
        f=filp_open("/tmp/hellosys", O_CREAT | O_RDWR, 0644);
        if(f){
                ret = vfs_write(f,str,len,&offset);
                filp_close(f,NULL);
        }
        if(ret==0) printk("open file error!!!");
        return ret;
}
```

(2) 重新编译、 安装内核;

同1

(3) 编写用户测试程序， 测试 hellosys 系统调用。

```c
#include <sys/syscall.h>
#include <stdio.h>
#include <unistd.h>

int main(){
        const char s[]="test hellosys\n";
        long ret = syscall(335,s,sizeof(s));
        printf("ret:%ld\n",ret);
        return 0;
}
```

```
wxt@wxt-virtual-machine:~$ vim test.c
wxt@wxt-virtual-machine:~$ gcc -Wall -ggdb -o test test.c
wxt@wxt-virtual-machine:~$ ./test
ret:15
wxt@wxt-virtual-machine:~$ cat /tmp/hellosys
test hellosys
```