گزارش آزمایش ۸

اسرا کاشانی‌نیا - 95105816

حسین قطب‌الدینی - 95109972

فاطمه باقری - 95105419

کد زیر را اجرا می‌کنیم:

sniffer.c

```c
#include <linux/init.h>
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/kallsyms.h>
#include <linux/string.h>

static int __init sniffer_init(void)
{
    printk(KERN_INFO "sniffer is initializing...");
    printk("The address of sys_call_table is: %lx\n", kallsyms_lookup_name("sys_call_table"));
    void **a;
    a = (void **) kallsyms_lookup_name("sys_call_table");
    int i  = 0;
    for(i = 0; i < 1024; i++){
        printk("%p\n", a[i]);
    }

    printk(KERN_INFO "done!\n");
    return 0;
}
static void __exit sniffer_exit(void)
{
    printk(KERN_INFO "sniffer is cleaning up...");
    printk(KERN_INFO "done!\n");
}
MODULE_AUTHOR("xyz");
MODULE_LICENSE("GPL");
module_init(sniffer_init);
module_exit(sniffer_exit);
```

توضیح کد: تابع kallsyms_lookup_name آدرس syscall را پیدا می‌کند. پس یک متغیر a می‌گیریم که به آن آدرس اشاره کند. از آنجایی که هر کدام از آدرس‌های syscall از نوع void* هستند، پس آدرس آنها باید از نوع void** باشد. بعد اعضای آن آرایه را یکی‌یکی می‌خوانیم.

در آخر کد هم باید مشخص کنیم که هنگام اضافه شدن ماژول به کرنل چه تابعی فراخوانی شود، که همین تابع sniffer_init را پاس می‌دهیم، و همینطور مشخص کنیم که هنگام حذف ماژول چه تابعی فراخوانی شود، که باز برای تست درستی کارکرد ماژول تابع sniffer_exit را نوشتیم و به module_exit پاس دادیم. همچنین چون از تابع kallsyms_lookup_name استفاده کرده‌ایم باید کتابخانه‌ی linux/kallsyms.h را وارد کنیم و در آخر ماژول هم LICENSE آن را برابر GPL قرار دهیم.

فرمت Makefile:

sniffer.c

```
obj-m += sniffer.o

all:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules

clean:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

گزارش اجرا:

به ترتیب دستورات زیر را اجرا می‌کنیم:

Make

Insmod sniffer.ko

Dmesg

که اولی فایل sniffer.ko و فایل‌های دیگر ماژول را می‌سازد، دومی آن را به کرنل اضافه می‌کند (و تابع ما را فراخوانی می‌کند) و سومی پیام‌های ماژول‌های کرنل از جمله آنهایی که هنگام فراخوانی تابع ما چاپ می‌شوند را به ما نشان می‌دهد. در ضمن در syscall table فقط آدرس syscallها هست، و اینکه ترتیب syscallها کدام است و آدرس‌ها به چه ترتیبی در جدول قرار گرفته‌اند وجود ندارد که آن را پایینتر نشان می‌دهیم.

دستور اول:

```
esra@esra-HP-Spectre-Notebook:~/Desktop/sniffer$ sudo su
[sudo] password for esra:
root@esra-HP-Spectre-Notebook:/home/esra/Desktop/sniffer# make
make -C /lib/modules/5.4.0-73-generic/build M=/home/esra/Desktop/sniffer modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-73-generic'
  CC [M]  /home/esra/Desktop/sniffer/sniffer.o
/home/esra/Desktop/sniffer/sniffer.c: In function ‘sniffer_init’:
/home/esra/Desktop/sniffer/sniffer.c:11:5: warning: ISO C90 forbids mixed declar
ations and code [-Wdeclaration-after-statement]
     void **a;
     ^~~~
/home/esra/Desktop/sniffer/sniffer.c:13:5: warning: ISO C90 forbids mixed declar
ations and code [-Wdeclaration-after-statement]
     int i  = 0;
     ^~~
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M]  /home/esra/Desktop/sniffer/sniffer.mod.o
  LD [M]  /home/esra/Desktop/sniffer/sniffer.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-73-generic'
root@esra-HP-Spectre-Notebook:/home/esra/Desktop/sniffer#
```

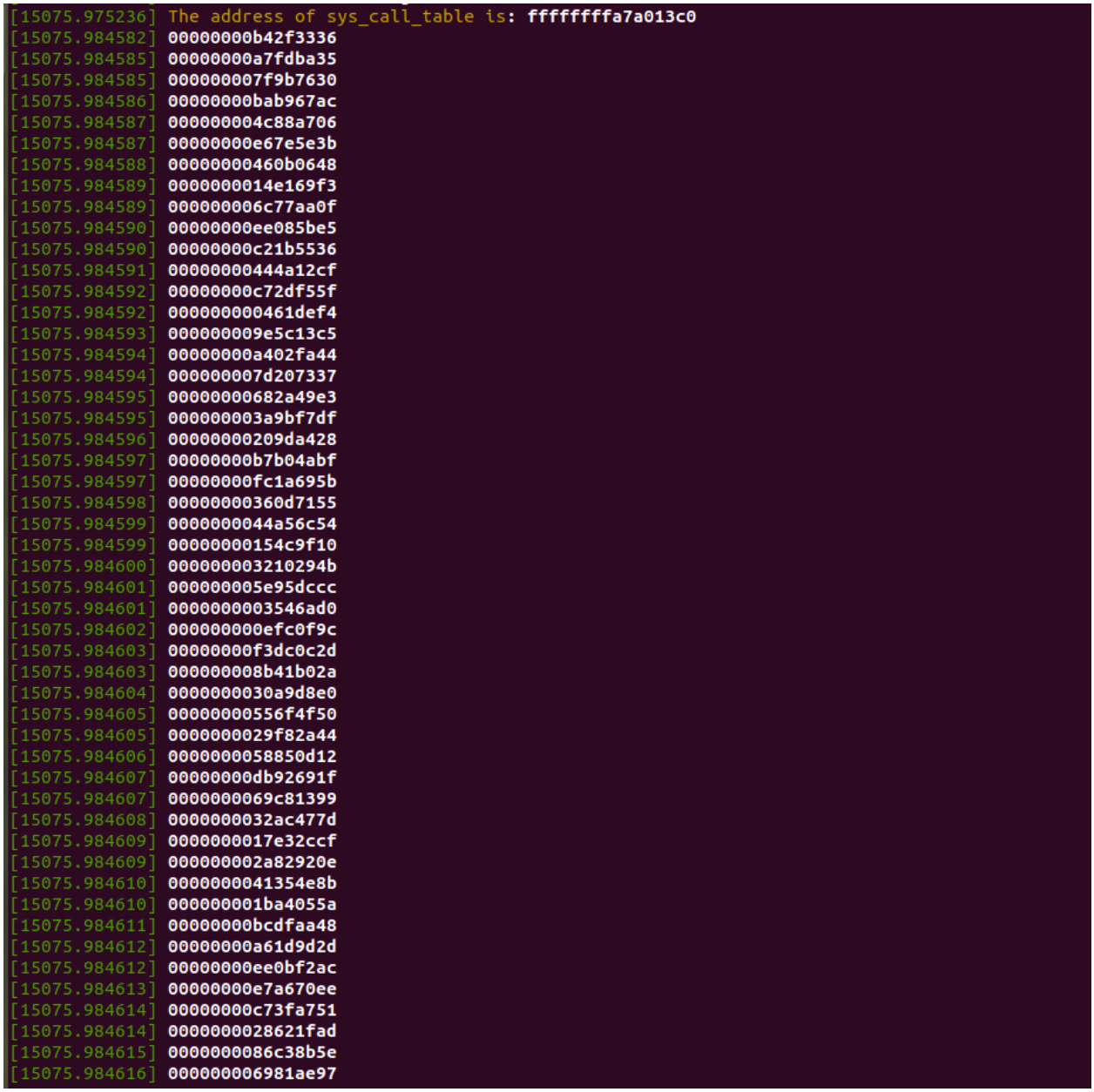
دستور دوم:

```
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-73-generic'
root@esra-HP-Spectre-Notebook:/home/esra/Desktop/sniffer# insmod sniffer.ko
root@esra-HP-Spectre-Notebook:/home/esra/Desktop/sniffer#
```

دستور سوم:

```
root@esra-HP-Spectre-Notebook:/home/esra/Desktop/sniffer# insmod sniffer.ko
root@esra-HP-Spectre-Notebook:/home/esra/Desktop/sniffer# dmesg
[    1.139725] usb usb4: Product: xHCI Host Controller
[    1.139726] usb usb4: Manufacturer: Linux 5.4.0-73-generic xhci-hcd
[    1.139726] usb usb4: SerialNumber: 0000:37:00.0
[    1.139833] hub 4-0:1.0: USB hub found
[    1.139840] hub 4-0:1.0: 2 ports detected
[    1.140000] i8042: PNP: PS/2 Controller [PNP0303:PS2K,PNP0f13:PS2M] at 0x60,0x64 irq 1,12
[    1.160331] serio: i8042 KBD port at 0x60,0x64 irq 1
[    1.160334] serio: i8042 AUX port at 0x60,0x64 irq 12
[    1.160509] mousedev: PS/2 mouse device common for all mice
[    1.160815] rtc_cmos 00:03: RTC can wake from S4
[    1.161319] rtc_cmos 00:03: registered as rtc0
[    1.161332] rtc_cmos 00:03: alarms up to one month, y3k, 242 bytes nvram, hpet irqs
[    1.161338] i2c /dev entries driver
[    1.161385] device-mapper: uevent: version 1.0.3
[    1.161434] device-mapper: ioctl: 4.41.0-ioctl (2019-09-16) initialised: dm-devel@redhat.com
[    1.161451] platform eisa.0: Probing EISA bus 0
[    1.161452] platform eisa.0: EISA: Cannot allocate resource for mainboard
[    1.161454] platform eisa.0: Cannot allocate resource for EISA slot 1
[    1.161454] platform eisa.0: Cannot allocate resource for EISA slot 2
[    1.161455] platform eisa.0: Cannot allocate resource for EISA slot 3
[    1.161456] platform eisa.0: Cannot allocate resource for EISA slot 4
[    1.161457] platform eisa.0: Cannot allocate resource for EISA slot 5
[    1.161458] platform eisa.0: Cannot allocate resource for EISA slot 6
[    1.161459] platform eisa.0: Cannot allocate resource for EISA slot 7
[    1.161460] platform eisa.0: Cannot allocate resource for EISA slot 8
[    1.161460] platform eisa.0: EISA: Detected 0 cards
[    1.161463] intel_pstate: Intel P-state driver initializing
[    1.161782] intel_pstate: HWP enabled
[    1.161855] ledtrig-cpu: registered to indicate activity on CPUs
[    1.161857] EFI Variables Facility v0.08 2004-May-17
[    1.171955] input: AT Translated Set 2 keyboard as /devices/platform/i8042/serio0/input/input3
[    1.190951] battery: ACPI: Battery Slot [BAT1] (battery present)
[    1.221106] intel_pmc_core intel_pmc_core.0:  initialized
[    1.221131] drop_monitor: Initializing network drop monitor service
[    1.221287] NET: Registered protocol family 10
[    1.226414] Segment Routing with IPv6
[    1.226450] NET: Registered protocol family 17
[    1.226579] Key type dns_resolver registered
[    1.226962] RAS: Correctable Errors collector initialized.
[    1.227025] microcode: sig=0x406e3, pf=0x80, revision=0xe2
[    1.227120] microcode: Microcode Update Driver: v2.2.
[    1.227123] IPI shorthand broadcast: enabled
[    1.227131] sched_clock: Marking stable (1226484375, 632470)->(1292801294, -65684449)
```

که اینها پرینت‌های مربوط به ماژول‌های دیگر است که تعدادشان زیاد است و پرینت‌های ما آخر کار قرار دارند:



که اینها آدرس‌های syscallها است. ترتیب syscallهای نظیر از دستور زیر بدست می‌آید:

```
root@esra-HP-Spectre-Notebook:/home/esra/Desktop/sniffer# ausyscall --dump
Using x86_64 syscall table:
0       read
1       write
2       open
3       close
4       stat
5       fstat
6       lstat
7       poll
8       lseek
9       mmap
10      mprotect
11      munmap
12      brk
13      rt_sigaction
```

در آخر هم برای اینکه ماژول در کرنل ماندگار نشود دستور rmmod sniffer.ko را اجرا می‌کنیم. خود فایل‌های ساخته شده هنگام ساخته شدن ماژول را هم می‌توانیم با دستور make clean حذف کنیم.