朱 新宇
1120379029
CSDI
2013年3月16日星期六

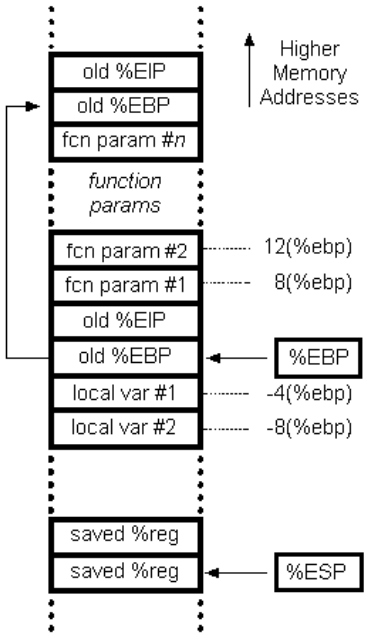
REPORT OF LAB1

1. Difficult Part

最困难的部分是Excercise 14中如何溢出到指定函数并在同时能返回到overflow me这个函数而不会到时栈结构的混乱。

2. How to Solve

如果直接通过cprintf的n%通过指针修改old EBP的值指向do_overflow的话虽然可以直接跳转过去但是无法返回，所以在实验中采取了在str buf中插入机器码，然后通过cprintf跳转找str buf所在的空间，通过push指令做好栈的结构在通过RET指令跳转。




```
str[0]=0x68;

*((int *)(str+1)) = oldeip;

str[5]=0x68;

*((int *)(str+6))=(int)do_overflow;

str[10]=0xc3;
```

以上代码构成了三条机器码，对应的汇编为：

```
push oldeip;

push do_overflow;

ret;
```

前两条指令分别将最原始的eip（overflow_me调用call start_overflow时压栈的eip）和do_overflow的地址压栈，这样在执行RET的时候会弹出栈顶的do_overflow地址作为EIP，然后跳转到do_overflow完成函数调用，在函数退出的时候，由于会调用leave指令，所以栈顶指针esp又会指回到原来的地方（oldeip）这是RET就会返回到overflow_me内正常执行。

3. Excercise Answers

Excercise 5

在BIOS进入bootloader是0x100000处是0，当从bootloader进入kernel时，则是如下

0x100000:　0x1badb002　0x00000000　0xe4524ffe　0x7205c766

0x100010:　0x34000004　0x0000b812　0x220f0011　0xc0200fd8

因为在load内核的过程中这个地址写入了内容。

Excercise 6

如果修改了连接地址那么在进入bootloader后的ljmp $PROT_MODE_CSEG, $protcseg会出错，本来应该跳到其无法跳转到mov $0x10,%ax，link address和load address不一致，BOIS将BootLoader加载到0x7c00处。

Excercise 7

在Jmp *%eax出GDT生效，因为eip已经被后移，movl $relocated, %eax会被执行。

Excercise 8：

（1）printf.c的vprintf调用了printfmt.c中的vprintfmt来处理其中的参数，在这个过程中vprintfmt又通过函数指针调用printf.c中putch输出字符串，putch又会调用console.c中的cputchar来输出char到屏幕上。

（2）在屏幕上字符满了后(crt_pos>CRT_SIZE)，通过memcopy将buf中的数据向前移动一个屏幕行，然后将后面的空间清空，这样屏幕就上一了一行。

（3）fmt指向的是第一个参数（字符串），ap是va_list也就是char *，其指向后面边长参数的第一个。

cprintf()-->vcprintf("x %d, y %x, z%d\n", 12(%ebp))-->vprintfmt()

调用x前, va_arg指向ebp+3，调用后va_arg指向ebp+4。cons_putc(49)

调用y前, va_arg指向ebp+4，调用后va_arg指向ebp+5。cons_putc(51)

调用z前, va_arg指向ebp+5，调用后va_arg指向ebp+6。cons_putc(52)

（4）He110 World，因为57616转换成16进制是e110，然后0x00646c72由于是按照char*来输出所以就变成了rld。

（5）随机值，因为实际的参数要比字符串中指定的少，所以在第二次去参数的时候指针会指向3后面的空间，其是不确定的,是cprintf调用前的栈顶。

（6）#define va_start(ap,last) ((ap=(va_list)&(last)+va_size(last)

#define var_arg(ap,type) (*(type *)((ap) -= va_size(type),(ap)+va_size(type)))

Excercise 10

在Entry.S的最后会初始化栈，栈在KSTACKTOP指定的地方并通过.space KSTKSIZE（8个PGSIZE）来预留空间，bootstacktop是高地址。

Excercise 11

每次函数调用都会执行：

```
push ebp

mov esp ebp

push ebx

...

call test_backtrace
```

所以需要2个word