Software Security – Exam: Buffer Overflow

ftp sourcecodegenerator.top
用户名：softwaresecurity
密码：njussftp

```c
int main(){
  int choice = 0;
  printf("Welcome the eaxm of Software Security!\n");
  printf("current file is exam1.c\n");
  printf("Please choose the exam: 1, 2 or 3\n");
  printf("1: buffer overflow exam\n");
  printf("2: return to libc exam\n");
  printf("3: rop exam\n");
  scanf("%d", &choice);

  switch (choice){
    case 1:
      exam1_bof();
      break;
    case 2:
      exam2_ret2libc();
      break;
    case 3:
      exam3_rop();
      break;
    default:
      printf("You type the wrong number! Please select between 1-3!\n");
  }
}
```

这里有三个实验，输入相应的数字：1，2，3 来进入各自的实验。

确保自己的实验环境里面将ASLR(地址空间随机化)关闭:

sudo sysctl -w kernel . randomize_va_space =0

第一题

目标：将check 覆盖为0xdeadbeef

```
void exam1_bof(){
    printf("<=================Welcome to the exam1: Simple-BOF(10 Points)\n");
    int check = 0x04030201;
    // this is random length
      char buf[40];

      printf("Please input the buf:");
      scanf("%s", buf);
      printf("\n[buf]: %s\n", buf);
      printf("[check] %p\n", check);

    if ((check != 0x04030201) && (check != 0xdeadbeef)){
      printf ("\nYou are on the right way! Please try again!\n");
      exit(-1);
    }

    if (check == 0xdeadbeef)
     {
       printf("Yeah dude! You win!\nOpening your shell...\n");
       system("/bin/dash");
       printf("Shell closed! Bye.\n");
       printf("================Congratuations! You got 10 points!==============");
     }
  }
```

需要注意的是 buf 的长度每个人的都是 random 生成的，也就是每个人的 buf 长度都是不一样的，需要大家通过 objdump 或者 gdb 来手动确认下。

可以将输入文件写到一个文件里面，通过文件重定位来进行 exploit。比如，重定位的文件为 attack_input1: 它的内容布局为

```
1
xxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

第一行为 1，2 或 3，用来选择相应的实验，第二行为覆盖的 buf 的字符串。进行攻击时，需要：

```
./exam < attack_input1
```

实验 2，3 的输入同理。

第二题

目标1：通过return2libc攻击，调用puts函数输出字符串
"My-user-id-is-" + 学号+ ",-I-got-X-points!“
目标2：调用exit函数，使程序正常返回，不会造成程序崩溃或错误

```
// Scores are 40 points
void exam2_ret2libc(){
  printf("<==================Welcome to the exam2: Ret2libc(40 Points)\n");
  // this is random length
  char buf[40];

  scanf("%s", buf);
}
```

第三题

前置条件：创建一个名为data的文件（可以使用touch指令）

目标1：使用系统调用chmod，将data文件的权限设置为777;
目标2：使用系统调用rename，将文件名重命名为你的学号;
目标3：调用exit 正确退出。

```
const char* output="./data";
void exam3_rop(){
  printf("<==================Welcome to the exam3: ROP(50 Points)\n");
  printf("output is %s\n", output);
  // random length
  char buf[40];

  scanf("%s", buf);
}

void rop_gadget(){
    __asm__( /* Assembly function body */
  "int $0x80 \n"
  "push %edi \n"
  "ret \n"
  "int $0x80 \n"
  "pop %esi \n"
  "inc %eax \n"
  "dec %ebx \n"
  "ret \n"
  );
}
```

第三题

提示1：scanf 遇到空白符号会截断输入，在处理rop 地址的时候避免有空白字符的出现(如0x20, 0x0a, 0x00 等)。
提示2：该实验的buf 和第二个实验的buf 长度一致。
提示3：chmod和rename的系统调用号，和各个寄存器需要的值，可在下面链接进行查询：
https://chromium.googlesource.com/chromiumos/docs/+/master/constants/syscalls.md#x86-32_bit

提示4：chmod设置文件权限为777，需要的参数umode_t会被程序理解为8进制。
以权限0640为例，需要将其转化为十进制值416放入对应寄存器

NR	syscall name	references	%eax	arg0 (%ebx)	arg1 (%ecx)	arg2 (%edx)	arg3 (%esi)	arg4 (%edi)	arg5 (%ebp)
1	exit	man/ cs/	0x01	int error_code					
15	chmod	man/ cs/	0x0f	const char *filename	umode_t mode	-	-	-	-
38	rename	man/ cs/	0x26	const char *oldname	const char *newname	-	-	-	-