《漏洞利用及渗透测试基础》实验报告

姓名：张洋　　学号：2111460　　班级：信安二班

实验名称：

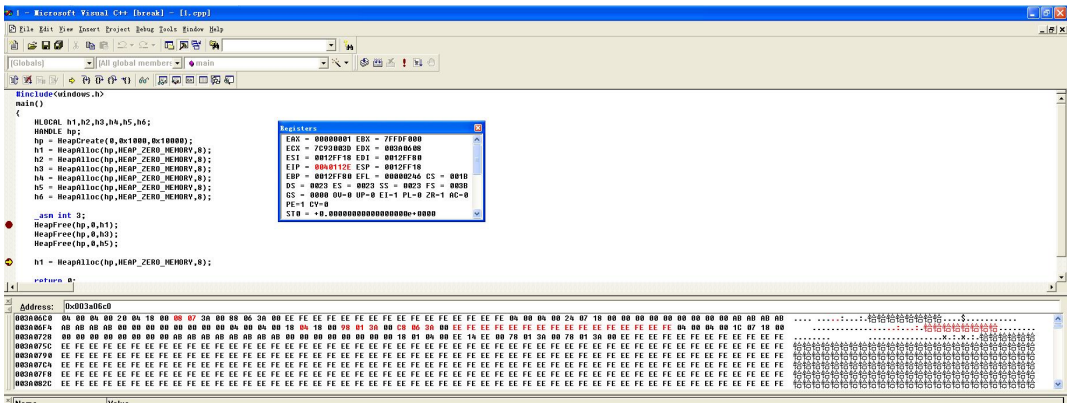
堆溢出 Dword Shoot 模拟实验

实验要求：

以第四章示例 4-4 代码为准，在 VC IDE 中进行调试，观察堆管理结构，记录 Unlink 节点时的双向空闲链表的状态变化，了解堆溢出漏洞下的 Dword Shoot 攻击。

实验过程：

1. 进入 VC IDE

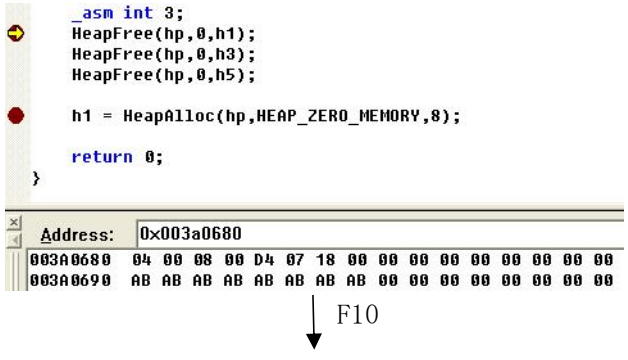


2. 获得初始地址

Name	Value
h6	0x003a0728
h5	0x003a0708
h4	0x003a06e8
h3	0x003a06c8
h2	0x003a06a8
h1	0x003a0688
hp	0x003a0000

3. 逐步调试

（1）释放 h1



```
    _asm int 3;
    HeapFree(hp,0,h1);
    HeapFree(hp,0,h3);
    HeapFree(hp,0,h5);

    h1 = HeapAlloc(hp,HEAP_ZERO_MEMORY,8);

    return 0;
}
```

```
Address: 0x003a0680
003A0680  04 00 08 00 D4 04 18 00 98 01 3A 00 98 01 3A 00
003A0690  EE FE EE FE EE FE EE FE EE FE EE FE EE FE EE FE
```

此时 h1(块身地址为 0x003a0688)的 Flink 和 Blink 均指向 0x003a0198(空闲链表 f2)。

```
Address: 0x003a0198
003A0198  88 06 3A 00 88 06 3A 00 A0 01 3A 00 A0 01 3A 00
003A01A8  A8 01 3A 00 A8 01 3A 00 B0 01 3A 00 B0 01 3A 00
```

可以观察到此时 f2 的 Flink 和 Blink 均指向 0x003a0688(h1 的块身地址)。

（2）释放 h3

```
    HeapFree(hp,0,h1);
    HeapFree(hp,0,h3);
    HeapFree(hp,0,h5);

    h1 = HeapAlloc(hp,HEAP_ZERO_MEMORY,8);

    return 0;
}
```

```
Address: 0x003a0198
003A0198  88 06 3A 00 C8 06 3A 00 A0 01 3A 00 A0 01 3A 00
003A01A8  A8 01 3A 00 A8 01 3A 00 B0 01 3A 00 B0 01 3A 00
```

h3 释放后 f2 的 Blink 变为 0x003a06c8(h3 的块身地址)。

```
Address: 0x003a0680
003A0680  04 00 08 00 D4 04 18 00 C8 06 3A 00 98 01 3A 00
003A0690  EE FE EE FE EE FE EE FE EE FE EE FE EE FE EE FE
```

h1 的 Flink 变为 0x003a06c8(h3 的块身地址)。

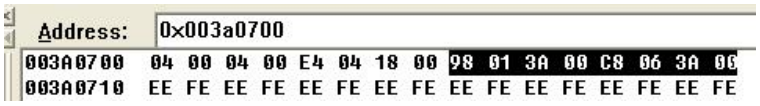
```
Address: 0x003a06c0
003A06C0  04 00 04 00 DC 04 18 00 98 01 3A 00 88 06 3A 00
003A06D0  EE FE EE FE EE FE EE FE EE FE EE FE EE FE EE FE
```

此时 h3 的 Flink 指向 0x003a0198(f2 的地址),Blink 指向 0x003a0688(h1 的地址)。

（3）释放 h5

```
Address: 0x003a0198
003A0198  88 06 3A 00 08 07 3A 00 A0 01 3A 00 A0 01 3A 00
003A01A8  A8 01 3A 00 A8 01 3A 00 B0 01 3A 00 B0 01 3A 00
```

h5 释放后 f2 的 Blink 变为 0x003a0708(h5 的块身地址)。

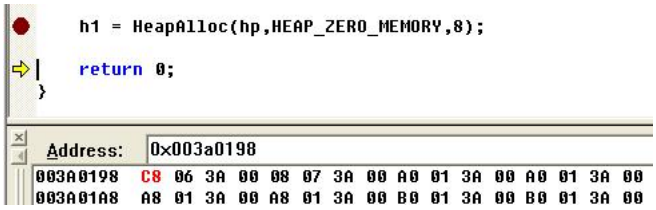


此时 h5 的 Flink 指向 0x003a0198(f2 的地址),Blink 指向 0x003a06c8(h3 的地址)。

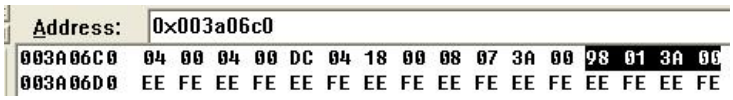
以上三步完成后，f2 的 Flink 指向 h1,Blink 指向 h5；h1 的 Flink 指向 h3,Blink 指向 f2；h3 的 Flink 指向 h5,Blink 指向 h1；h5 的 Flink 指向 f2,Blink 指向 h3。

即 f2 → h1 → h3 → h5

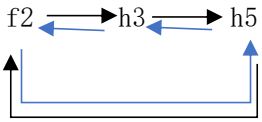
（4）再次从堆中申请空间



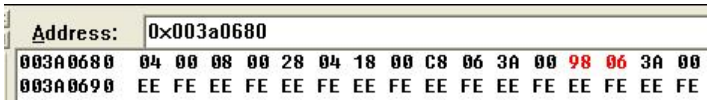
F2 指向了 003a06c8（h3 块身），说明 h1 被摘走，再去查看 h3 的 blink，变化为指向 f2：



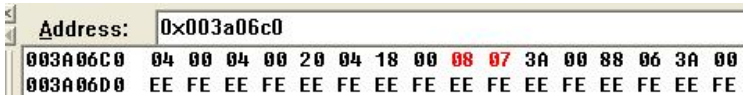
即 f2 → h3 → h5



（5）更改 h1 的 Blink 值



如果在执行第四步之前修改 h1 的 Blink 值，执行第四步后发生 Dword Shoot，再次观察 h3 中的 Blink:



h3 的 Blink 指向了 0x003a0688，即它自身，而非 f2，说明发生 Dword Shoot。

心得体会：

通过本次实验，观察到堆管理结构，通过观察 Unlink 节点的双向空闲链表的状态变化，了解堆溢出漏洞下的 Dword Shoot 攻击。Dword Shoot 漏洞是出现在双向链表表删除的时候出现的一种漏洞类型。在进行双向链表的操作过程中如果因为处置不当，比如溢出等等的情况下，构成双向链表的指向前一个和后一个节点的两个指针被恶意的改写的话，就会在链表删除的时候发生的漏洞。本次实验也通过更改一个指针观察到相关联的节点的变化从而导致的漏洞。