**《软件安全》实验报告**

姓名：胡博浩 学号：2212998 班级：信息安全

**一、实验名称：**

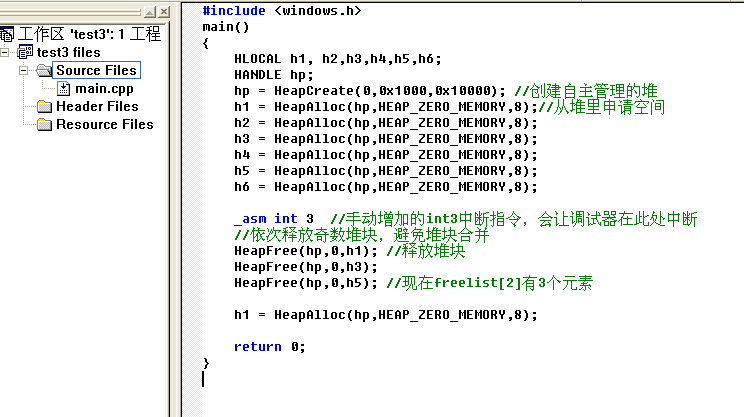
堆溢出Dword Shoot模拟实验

**二、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。

**三、实验过程：**

**1.创建文件**

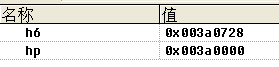


**2.代码分析**

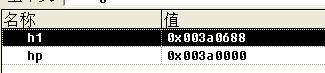
程序首先创建了一个大小为0x1000的堆区，并从中连续申请了6个块身大小为8字节的堆块，加上块首总共是16字节。然后依次释放奇数堆块，目的是为了防止堆块在释放后被合并。三次释放结束后，会形成三个16字节的空闲块表放入空表中，由于堆块大小是16字节，因此被加入freelist[2]中。再次申请8字节的堆区内存，此时会从freelist[2]中摘下第一个空闲的堆块返回给程序使用。此时，如果我们手动修改h1块首中的指针，就能够看到DWORD SHOOT发生。

**3.调试程序**

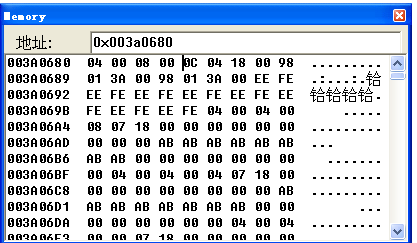
（1）运行实验代码，停止在int3断点处，此时已经完成了h1至h6的堆块初始化。hp为我们申请的堆，首地址为0x003a0000



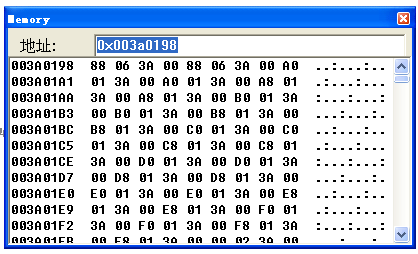
（2）执行HeapFree(hp,0,h1)前，h1块身的首地址为0x003a0688，而其对应的块首的起始地址为0x003a0680



执行HeapFree(hp,0,h1)后，块的前8个字节发生了变化，前8个字节分表示的是Flink和Blink，二者的值都是0x003a0198。这个值便是freelist[2]的地址。

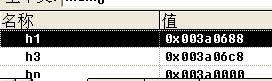


转到freelist[2]的地址处可以发现，freelist[2]的Flink和Blink都指向了0x003a0688，这就是释放的h1的块身的首地址。

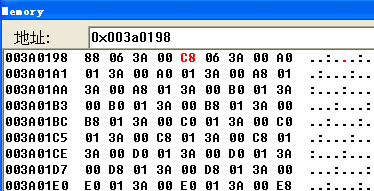


（3）之后执行HeapFree(hp,0,h3)和HeapFree(hp,0,h5)：

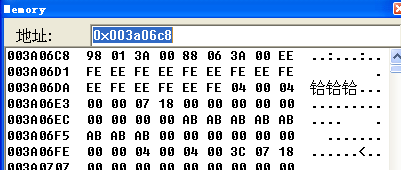
执行HeapFree(hp,0,h3)时，h3块身的首地址为0x003a06c8



执行完HeapFree(hp,0,h3)后，freelist[2]的Blink的值指向了h3块身的首地址。



而h3的Flink则指向了h1块的首地址，Blink指向了freelist[2]的地址，这说明，h3已经串入了freelist[2]中，并且是连在了最后。

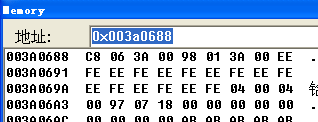
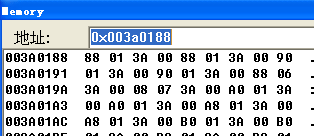


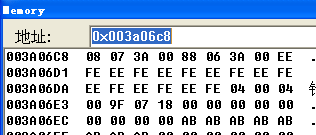
同理执行h5。最终，此时freelist[2]链表状态为：freelist[2]<=>h1<=>h3<=>h5。

（4）执行HeapAlloc(hp,HEAP\_ZERO\_MEMORY,8)语句时

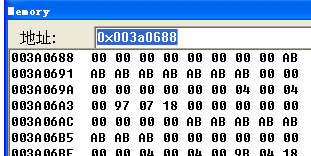
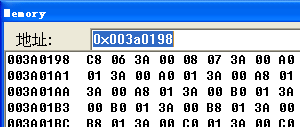
此时，当再次分配空间的时候，从freelist[2]的双向链表里摘下一块大小为16字节的堆块，首先摘得h1（地址为0x003a0688）。

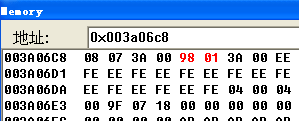
下图是执行前freelist、h1和h3的Flink和Blink的情况。



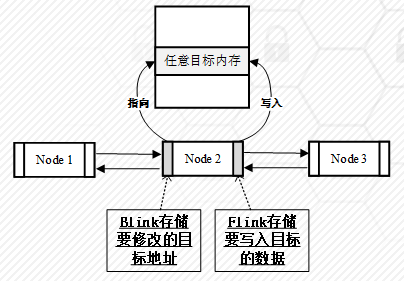


当执行完语句之后，我们可以看到，freelist[2]的Blink变成了h3块身的首地址，h3的Flink变成了freelist[2]的地址，h1块身中目前没有数据，全部为0。





我们可以注意到，在这个过程中其实是发生了



node1->Blink = node2->Blink

node3->Flink = node1->Flink

（5）Dword Shoot 攻击

假设在执行该语句之前，h1的Flink和Blink被改写为特定地址和特定数值，那么就完成一次Dword Shoot攻击。

**四、心得体会：**

在这次实验中，我学到了关于堆管理和堆溢出漏洞的重要知识。通过使用VC IDE进行调试，我深入了解了堆的内部结构，包括空闲块的管理方式以及双向空闲链表的组织形式。这让我对计算机系统内存管理的工作原理有了更清晰的认识。

另外，通过实践，我还学会了如何利用堆溢出漏洞来进行DWORD SHOOT攻击。这个过程让我意识到，即使是看似微小的内存错误也可能导致严重的系统安全问题。因此，加强对程序漏洞和安全漏洞的理解和防范显得尤为重要。

总的来说，通过这次实验，我不仅加深了对计算机系统内存管理的理解，还意识到了系统安全的重要性。这些知识和经验将对我未来的学习和工作有着积极的影响，使我能够更好地应对和防范各种安全挑战。