《漏洞利用及渗透测试基础》实验报告

姓名： 袁田 学号：2314022 班级： 计科三班

**实验名称：**

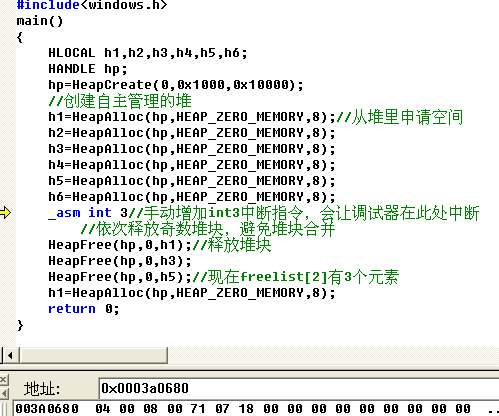
堆溢出Dword Shoot攻击示例

**实验要求：**

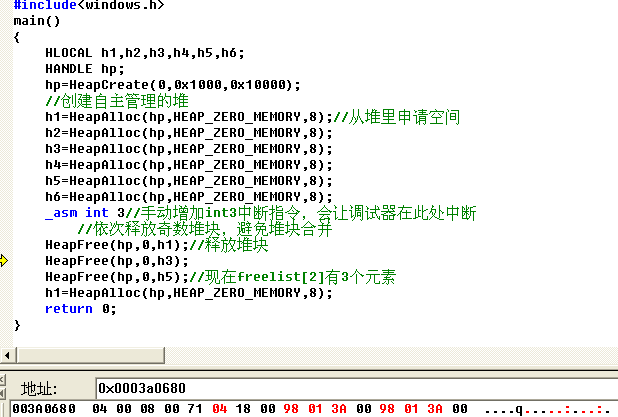
以第四章示例4-4代码为准，在VC IDE中进行调试，观察堆管理结构，记录Unlink节点时的双向空闲链表的状态变化，了解堆溢出漏洞下的Dword Shoot攻击。

**实验过程：**

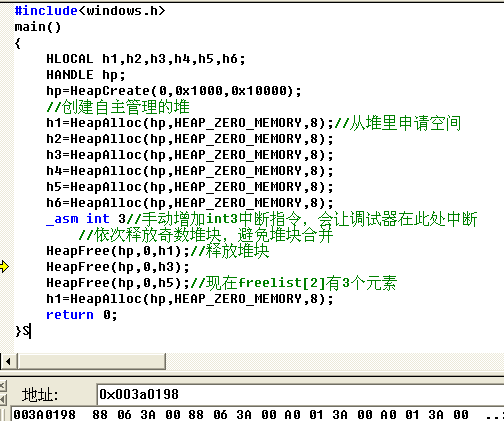
（1）进入到程序调试部分，发现在进行HeapFree操作前，h1地址(0x0003a0680)下的堆块中储存数据的位置中均为0，则说明此时堆块中无数据



（2）进行了HeapFree操作后，发现h1所指向的堆块中原本存储着数据的部分变为两个指针(Flink和Blink)，且指针数值相同(均为0x003a0198)，即指向同一个位置。由于从堆里申请空间时确定了该堆块的大小为8B,加上块首的8B，共计16B，则应储存于FreeList[2]中，说明FreeList[2]的地址为0x003a0198.



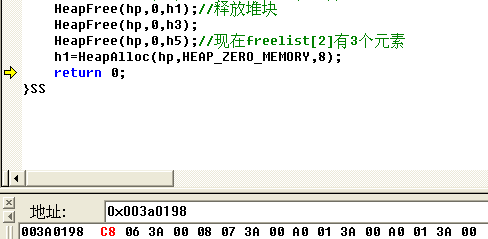
(3)此时，地址为0x003a0198时可以看到，其Flink指针和Blink指针也均连接至地址为0x003a0688,即h1所指向的块中



(4)进行了三次HeapFree操作后，FreeList[2]中有3个元素，即h1,h3,h5所指向的堆块。此时FreeList[2]的Flink指针指向h1所指向的堆块(地址为0x003a0688)，Blink指针指向h5所指向的指针(地址为0x003a0708)



(5)在执行h1=HeapAlloc(hp,HEAP\_ZERO\_MEMORY,8)后,再次调用HeapAlloc分配8字节内存。此时FreeList[2]的Flink指针指向h2所指向的堆块(地址为0x003a06C8)说明h1所指向的堆块再次被使用



(6)在执行h1=HeapAlloc(hp,HEAP\_ZERO\_MEMORY,8)前若进行操作，将h1的Flink指针和Blink指针进行修改，则实现了Dword Shoot攻击

**心得体会：**

通过本次实验，对于堆的结构(由一个堆表和大量堆块组成)，堆块的释放(将空闲的堆块链入到Free List中)，堆块的再利用(根据利用的堆块的大小，在FreeList的合适位置拿出堆块)。并且理解了该如何通过堆溢出，也就是通过溢出数据覆盖堆块中的Flink指针和Blink指针来进行Dword Shoot攻击。