Dword Shoot 攻击

作者：Yu Xiang You

学号：2312900

专业：计算机科学与技术

提交日期：2025 年 3 月 27 日

目录

1 Dword Shoot 攻击

1.1 实验名称

Dword Shoot 攻击

1.2 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。

1.3 实验过程

此次实验代码如下：

```
#include <windows.h>
main(){
        HLOCAL h1,h2,h3,h4,h5,h6;
        HANDLE hp;
        hp=HeapCreate(0,0x1000,0x10000); // my own heap
        h1=HeapAlloc(hp,HEAP_ZERO_MEMORY,8); // ask for space
            from heap:hp
        h2=HeapAlloc(hp,HEAP_ZERO_MEMORY,8);
        h3=HeapAlloc(hp,HEAP_ZERO_MEMORY,8);
        h4=HeapAlloc(hp,HEAP_ZERO_MEMORY,8);
        h5=HeapAlloc(hp,HEAP_ZERO_MEMORY,8);
        h6=HeapAlloc(hp,HEAP_ZERO_MEMORY,8);

        _asm int 3 // Struction:int 3,let debug stop here

    // release heap, prevent from merging
        HeapFree(hp,0,h1);  // release heap
        HeapFree(hp,0,h3);
        HeapFree(hp,0,h5);  // now freelsit[2] has 3 elements

        h1=HeapAlloc(hp,HEAP_ZERO_MEMORY,8);
        return 0;
}
```

然后将此代码导入到 XP 系统下的 VC6.0，编译并进入 DEBUG 页面。

表 1: 堆初始分配时节点地址

h1	h2	h3	h4	h5	h6
3a0688	3a06a8	3a06c8	3a06e8	3a0708	3a0728

接着执行内存的释放，此时有：

表 2: 释放后节点地址

	FreeList	h1	h3	h5
flink	3a0699	3a06c8	3a0708	3a0198
blink	3a0708	3a0198	3a0688	3a06c8

接着再次申请后有：

表 3: 释放后节点地址

	FreeList	h3
flink	3a06c8	3a0708
blink	3a0708	3a0198

1.4 心得体会

通过此次试验，我对内存的管理机制和内存相关的安全漏洞有了更深的理解。只听老师在课上所讲的不会像自己亲自动手一般如此让自己印象深刻。通过这次实验，我更理解了写函数时多考虑临界条件的必要性与重要性。以后自己也会慢慢培养出编码更加安全、更加严谨的习惯。