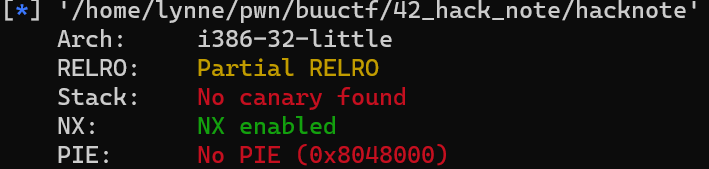
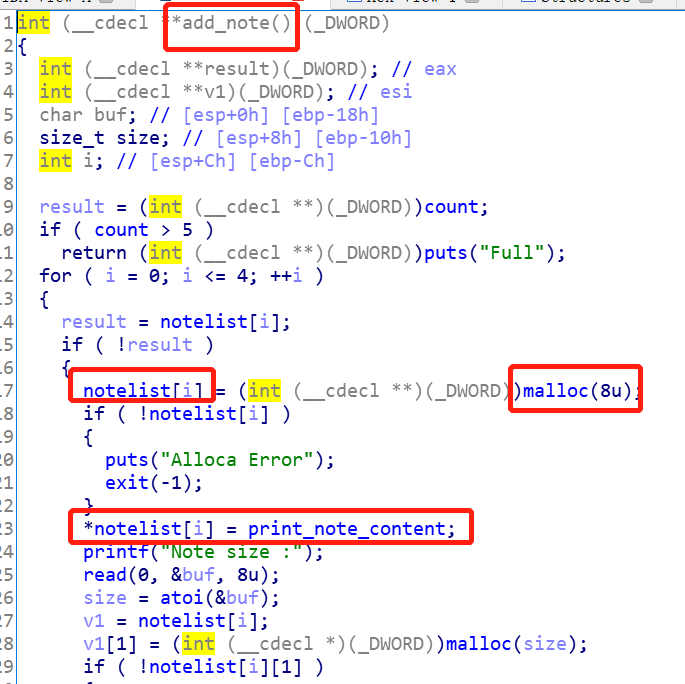
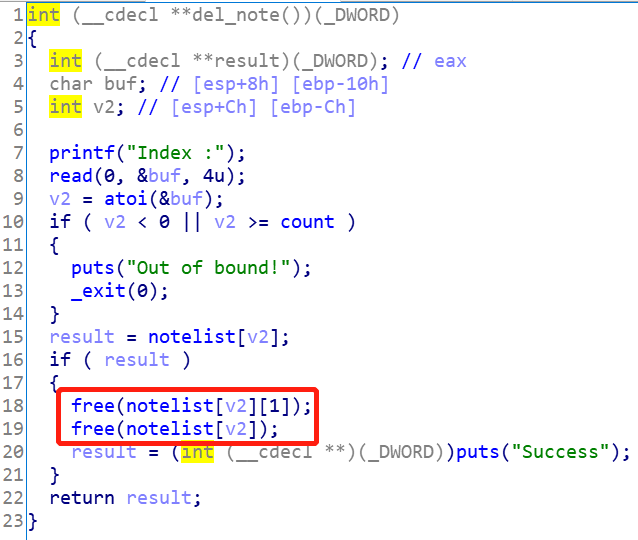
## 题目分析



32位小端。



又是一个菜单题，漏洞在del\_note函数：



Free后指针没有置空，也就是说，仍然可以print已经free掉的区域。

## 利用思路

由于每次申请8大小的内存，content存放函数指针，然后申请用户size。所以可以利用UAF：

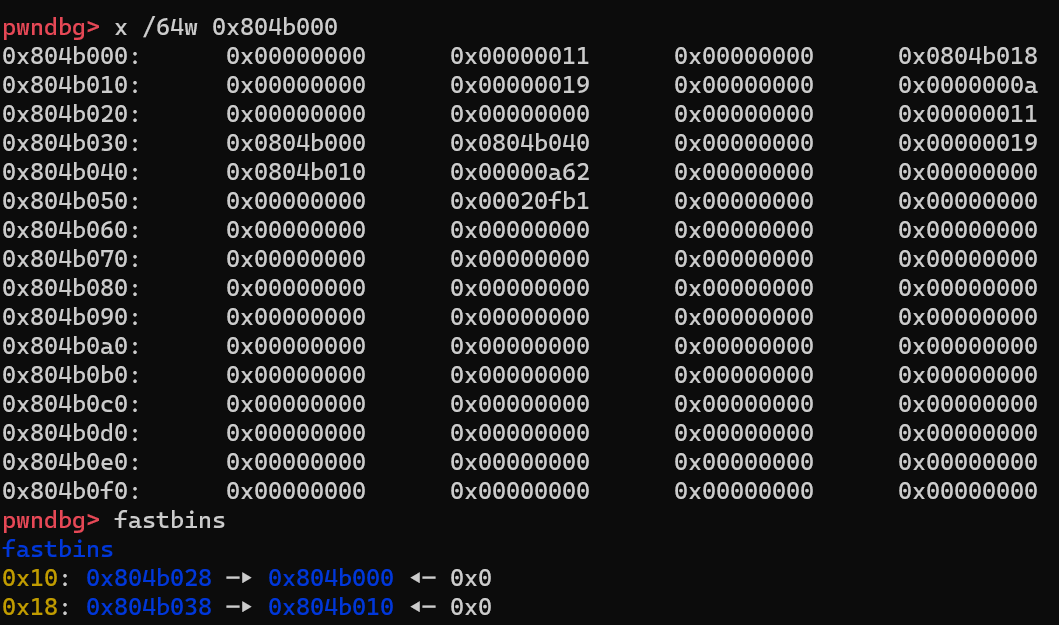
1、申请note0，大小为16，只要和8不在一个fastbins里面就行

2、申请note1，大小为16

3、释放note0

4、释放note1

此时内存布局为：



0x10和0x18处的bin存放的范围不一样噢，都是note1→note0

4、申请note2，用户大小为8，那么就会分配0x10那个bin里面的chunk，这样的话，note2的可编辑区域，就是note1的函数指针区域。

5、覆盖magic地址，然后print\_note(0)

即可以得到shell

