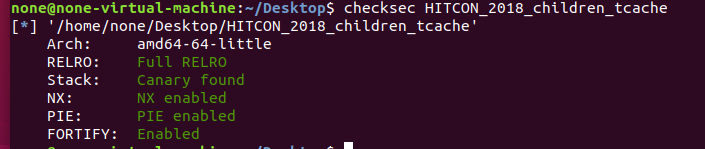
<https://blog.csdn.net/m0_51251108/article/details/121169047>

hitcon\_2018\_children\_tcache：

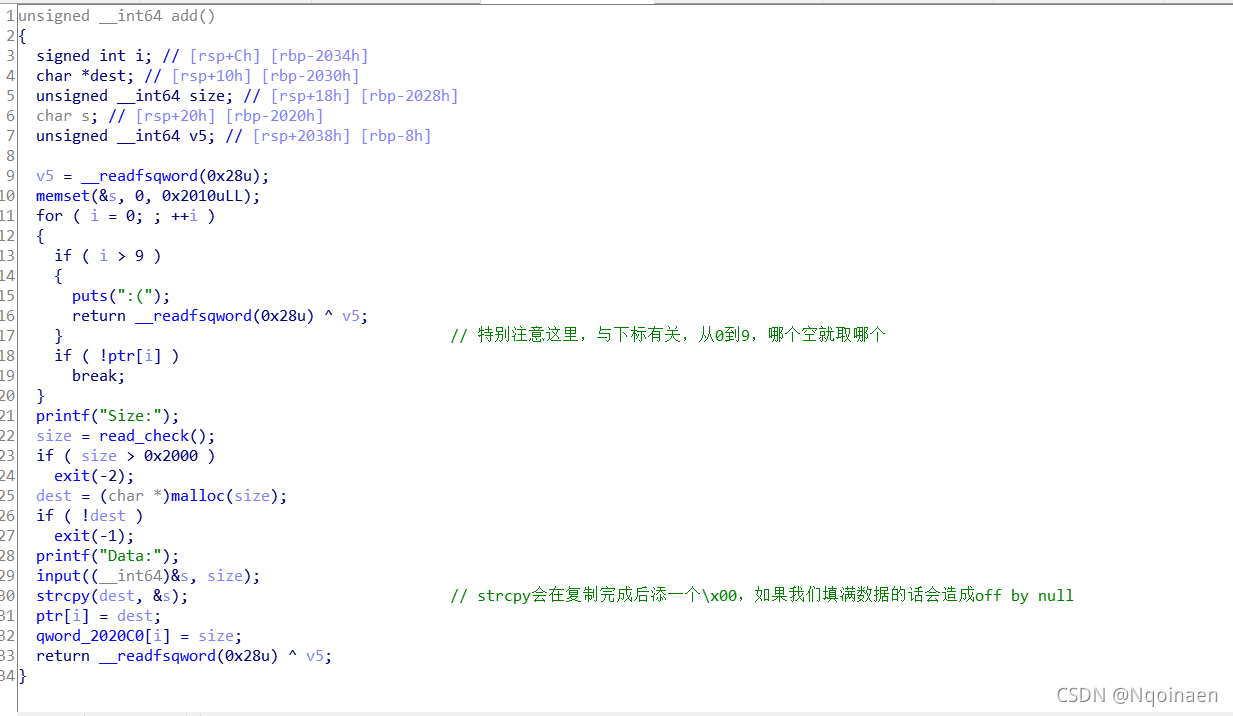
显然保护全开



逆向分析：

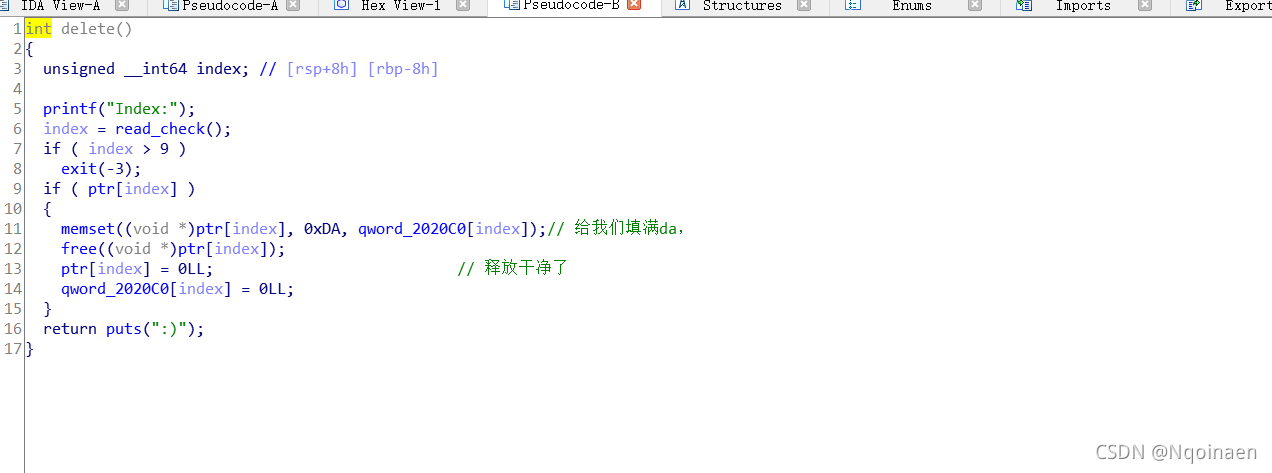
add函数：

漏洞在这个strcpy函数上



delete函数：

da是垃圾数据，阻碍我们漏洞利用，我们后面要利用strcpy的\x00循环一个一个清理prevsize所在的地方为\x00



大致思路：

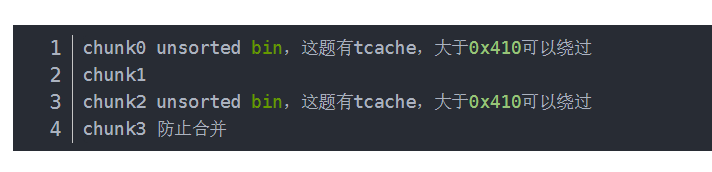
由于有off-by-null 我们可以改size的prev in use位为0，我们可以利用unssortbin合并堆块，

造成堆块复用

泄露libc地址，用doublefree，改malloc\_hook为onegadget

具体步骤：

1.先申请4个堆块

2.释放chunk0，为合并做准备

3.我们释放chunk1，再申请chunk1回来，填满数据，修改chunk2的prev in use位为0

由于free时会在prev size处填da，我们也需要控制prev size

所以循环一个一个字节一次清零，清零后我们填入prev size为chunk0+chunk1大小的和

到这里，我们已经布置好合并所需的条件了，然后用delete chunk2触发

（虽然合并了，但是chunk1里的指针还在，合并操作并没有清零指针）

4.我们申请和chunk0一样大小的chunk，就会从合并的unsortbin中割一块，然后有main\_arena+96地址就对应下移到了chunk1的fd与bk处，由于指针还在，我们就能show出来，相应计算出libc\_base,\_\_malloc\_hook,onegadget

5.我们再申请chunk1大小的size，在bss指针里就有了两个指向chunk1的指针了

我们就可以doublefree，然后我们利用doublefree改hook为og

小结：

利用offbynull，伪造了合并，释放，但bss段的指针并没有清零掉，造成了另一种uaf，doublefree

（仅为个人观点）