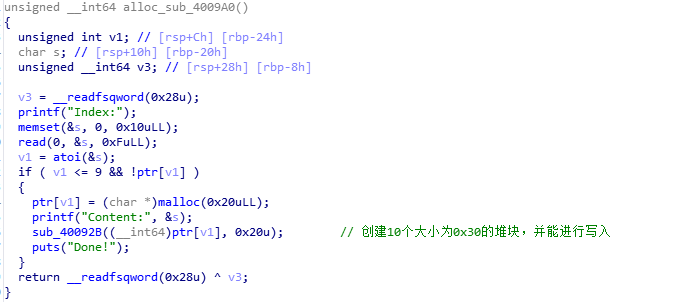
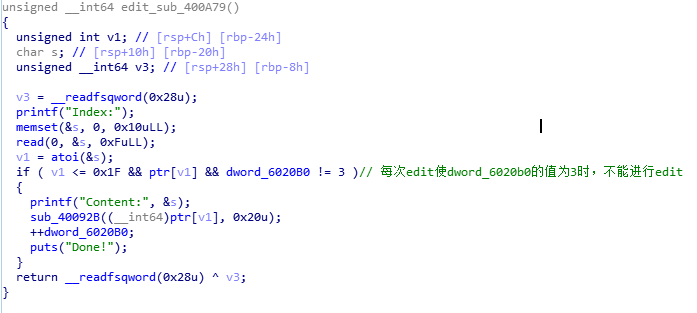
0 程序分析:

Create:



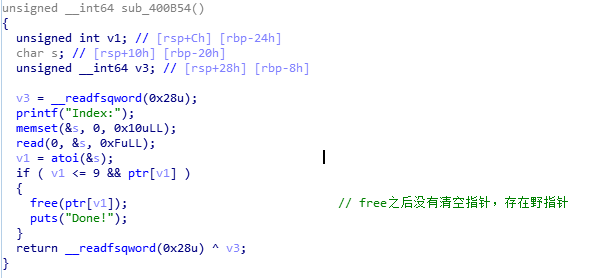
创建10个大小为0x30的chunk；

Edit：



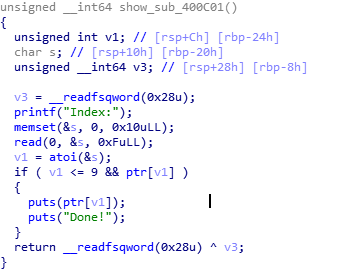
只能edit2次；

Free：



存在明显的UAF；

Show：



用于信息泄露。

二、利用思路

如何通过10次0x30大小的堆块创建及三次的edit完成内存的写入及信息泄露。

三、利用过程

1.fastbin泄露堆的地址

a=malloc（0x20）

b=malloc（0x20）

c=malloc（0x20）//防止和top chunk合并

free(b)

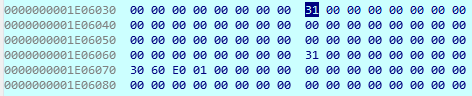
free(a)//先free a会导致最低位为0x0.输出不出来

show（b）//泄露出heap的地址。

2. UAF漏洞实现fastbin attack

即，对野指针进行edit，该野指针在fast bin chunk中，从而可以实现fast bin attack。

另外，调试发现：



原来紧挨bin的是栈顶：

Free 0x01E06030后，再free 0x01E06060时，结果是后一个free的chunk的fd放第一个free的chunk的地址，而先free的chunk的fd处还是0。

//第一次free时，第一个free的chunk块的fd指针会被清零。 按堆栈实现的阿。

此时，改的是最新被free chunk的fd指针为之前最新free chunk的地址；

及bin中的地址为最新free chunk的地址。

假若反过来，则需改之前最新free chunk的fd指针为最新chunk的地址；

及清零最新free的chunk的fd指针值为0。

3.通过fastbin attack 构造small chunk unlink，造成任意地址可写

4．不过3之后edit次数用完，通过free一个unsort bin chunk 把 dword\_6020B0的值修改为unsortbin的地址(过了edit次数限制)，同时这也泄露出了libc基址