<https://mp.weixin.qq.com/s/HYZ5PLK7XaycKgohAK_d2A>

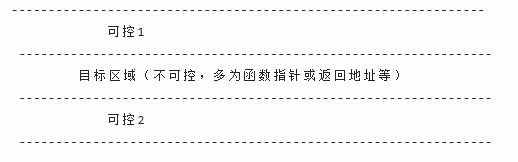
<https://blog.csdn.net/zzq487782568/article/details/123063448>

这里首先记录下exp123分别是什么，首先啊exp就是第一个链接里的exp，是最经典的house of spirit覆盖函数返回地址，达到ret2shellcdoe利用的目的。exp1是因为我在进行exp学习时产生了fake chunk addr设置在那里的疑问，随后自己重新选定一个fake chunk addr模仿exp写出的exp，同样可以成功攻击。加深了我对house of spirit的理解，而exp2则是链接2中的exp，非常简单明了地覆盖了free\_got表到的ret2shellcode利用的结果，让我陷入沉思，以后碰到house of spirit利用的时候要先看看能不能达到exp2的条件直接利用exp2的思想进行攻击。

House of Spirit（hos）是一个组合型的漏洞利用，是变量覆盖和堆管理机制的组合利用。是构造一个 fake chunk，然后释放掉它，这样再次申请的时候就会申请它。具体来说，关键在于能够覆盖一个堆指针变量，使其指向可控的区域，只要构造好数据，释放后系统会错误的将该区域作为堆块放到相应的fast bin里面，最后再分配出来的时候，就有可能改写我们目标区域。

**利用场景**

1. 想要控制的目标区域的前段空间与后段空间都是可控的内存区域。  
一般来说想要控制的目标区域多为返回地址或是一个函数指针，正常情况下，该内存区域我们是无法通过输入数据来进行控制的，想要利用hos攻击技术来改写该区域，首先需要我们可以控制那片目标区域的前面空间和后面空间。



2. 存在可将堆变量指针覆盖指向为可控区域，即上一步中的区域。

**Free要绕过的检测**

1. fake chunk的ISMMAP 位不能为 1，否则会调用munmap\_chunk函数去释放堆块。  
2. fake chunk的地址需要对齐， MALLOC\_ALIGN\_MASK。  
3. fake chunk的size 大小需要满足对应的 fastbin 的需求。  
4. 2 \* SIZE\_SZ < next chunk size < av->system\_mem 。  
5. fake chunk 对应的 fastbin 链表头部不能是该 fake chunk，即不能构成 double free 的情况。