1.realloc的几个特殊用法（摘自官方WP）

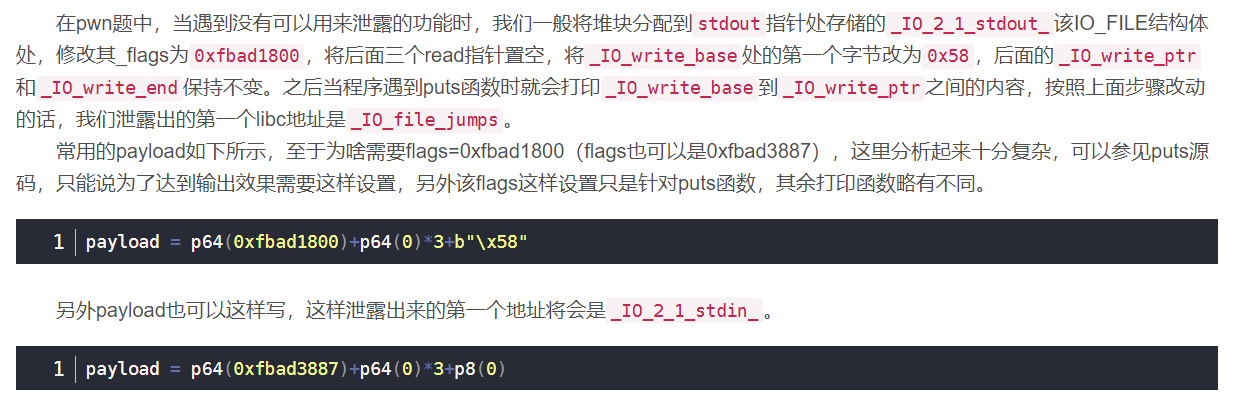
size == 0 ，这个时候等同于free

realloc\_ptr == 0 && size > 0 ， 这个时候等同于malloc

malloc\_usable\_size(realloc\_ptr) >= size， 这个时候等同于edit

malloc\_usable\_size(realloc\_ptr) < szie， 这个时候才是malloc一块更大的内存，将原来的内容复制过去，再将原来的chunk给free掉

2.IO\_file leak：



漏洞利用：

先利用doublefree释放到unsortbin中，unsortbin\_chunk fd，与bk就有了main\_arena+96

然后要把fd处main\_arena+96的地址改到\_IO\_2\_1\_stdout处，后4为要改为x870，x需要爆破（爆破说白了就是猜，1/16），我们就能把堆分配到\_IO\_2\_1\_stdout处，然后我们改它结构里的flag=0xfbad1800，三个IO\_read为0，\_IO\_write\_base处的第一个字节改为0x58，这样改然后泄露出的第一个libc地址是\_IO\_file\_jumps，接着我们算出libc\_base

选项666指针清零后，再次利用doublefree改fd为free\_hook，申请到hook处,，改为system，-8处改为/bin/sh\x00，（也可以写one\_gadget,我看另一个师傅的博客里就是用one\_gadget的，其实常理来说利用free\_hook就是one\_gadget，但free\_hook的参数其实就是free的参数，当可以写入binsh的时候也可以利用上面hook处,改为system，-8处改为/bin/sh\x00的写法，当然写入chunk时的地址要设置成hook-8

更详细一点的wp：

<https://blog.csdn.net/weixin_44145820/article/details/105585889>

由于本题没有show功能，因此我们要想泄露libc的地址就需要通过修改\_IO\_2\_1\_stdout\_来实现

因此第一步我们需要想办法让\_IO\_2\_1\_stdout\_留在tcache的fd处，这样就能分配到这个地方进行修改

过程如下

我们需要分配三个chunk0,1,2，申请大小分别为0x70，0x100，0xa0，这里只要realloc之后又realloc(0)就行（如果没有realloc(0)就接着realloc下一个是不能进入tcache的），因为chunk会进入tcache而不会和top chunk合并

realloc(0x100)把chunk1申请出来，然后删除7次填满tcache，再进行一次删除就进入unsorted bin，这样main\_arena+0x60就留在了fd处

realloc(0x70)把chunk0申请出来，然后realloc(0x180)，这样会把chunk0和chunk1合并分配给你，然后把chunk1的size改为0x41，fd的最低两个字节填充为0x?760（即\_IO\_2\_1\_stdout\_的地址），这里需要爆破一位。当然，我们在调试的时候可以强行修改内存，如下

set (long long)0x55c1fee5d2e0=0x7fae9fb1d760

完成这一步之后，我们就需要把\_IO\_2\_1\_stdout\_申请出来了，从而泄露libc地址，使用tcache poisoning就能分配出来，过程如下

realloc(0x100)把chunk1申请出来，然后realloc(0)释放，此时因为size被我们修改为0x41导致它不会回到原来的tcache队列中，然后再次realloc(0x100)就申请到了\_IO\_2\_1\_stdout\_，我们填入p64(0xfbad1887)+p64(0)\*3+p8(0x58)

这里0xfbad1887是flag位我们不用管原样填上，然后把\_IO\_read\_xxx的部分用0填充，并把\_IO\_write\_base的最低一个byte置为0x58，这样他就指向了\_IO\_2\_1\_stderr\_+216，其中存储着 \_IO\_file\_jumps的地址，根据它我们就能计算出libc地址

————————————————