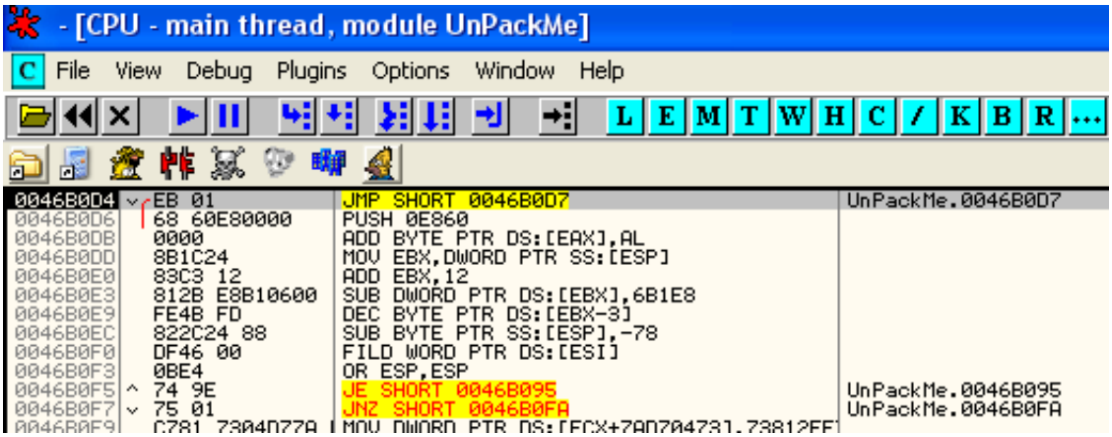
第四十八章-PeSpin V1.3.04 脱壳-Part1

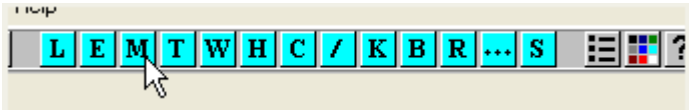
如果有的童鞋看了前两个章节,还是没有搞定那个 CrackMe 的话,可能考虑先跳过这两章从第 48 章开始看,可能确实这两章难度有点大。(PS:其实还好,基础扎实的话,看起来还是没有问题的)

本章我们来脱 PeSpin 这款壳,版本是 1.3.04。

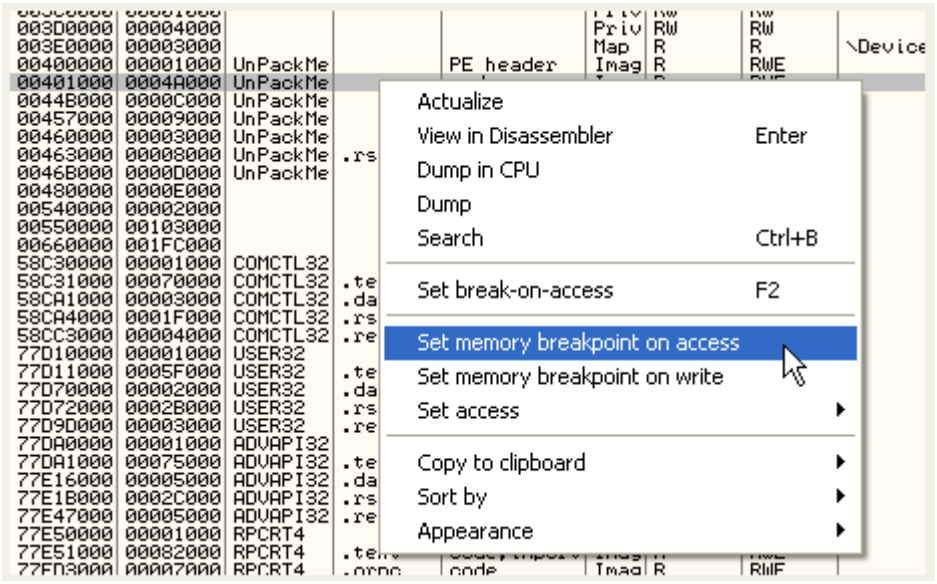
首先我们来定位该壳的 OEP。



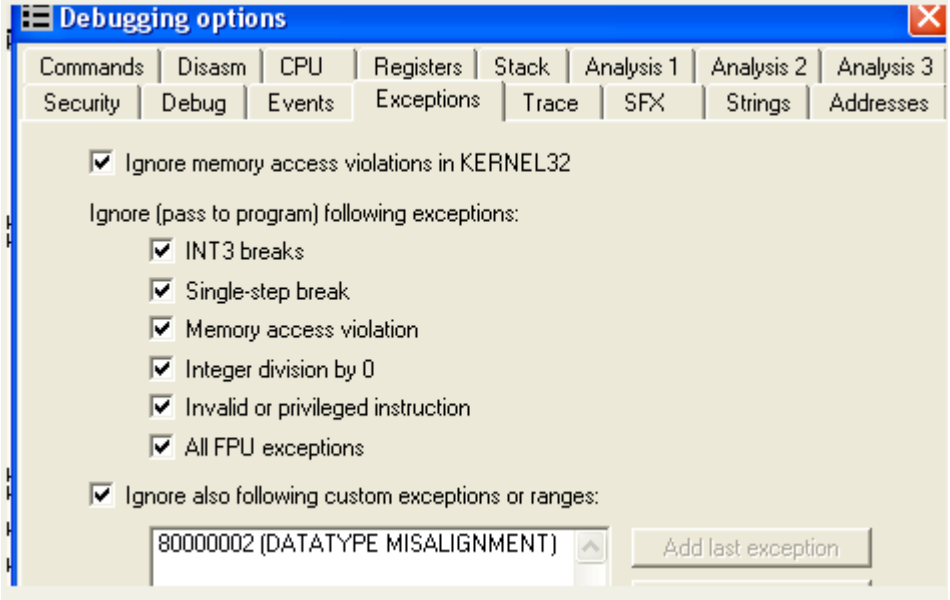
我们用专门定位 OEP 的 OD(olly_parcheado_para_vb)来加载它,断在了入口点处。



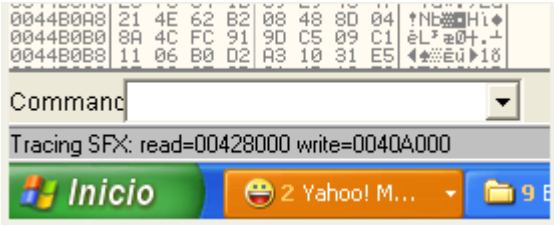
接着我们单击工具栏中的 M 按钮打开区段列表窗口,我们对主程序的代码段设置内存访问断点,大家还记得这个内存断点吧,这个 OD 的内存断点被 Patch 过,其仅仅是内存执行断点,只有执行才会断下,读取和写入并不会断下。



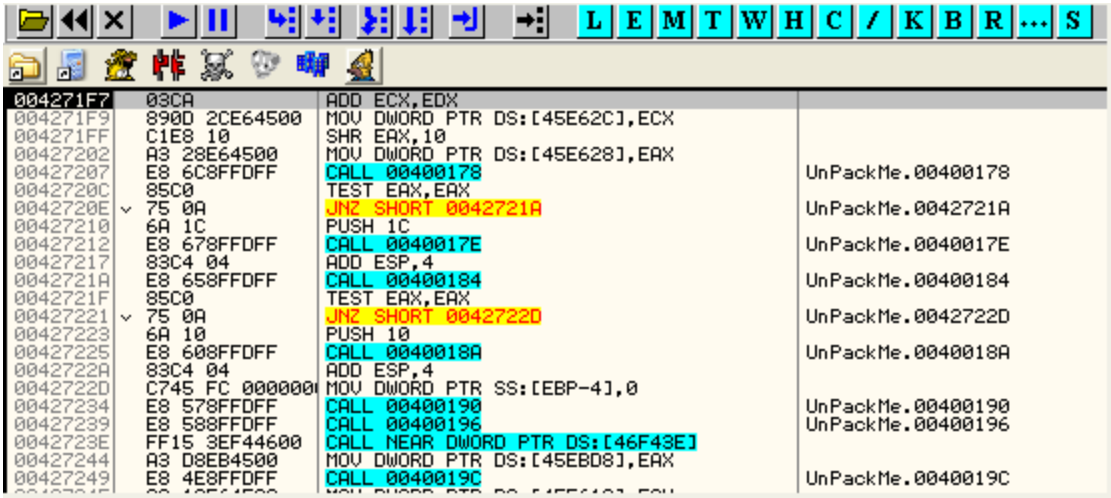
这里检查下看看要忽略的异常选项是否都勾选上了。



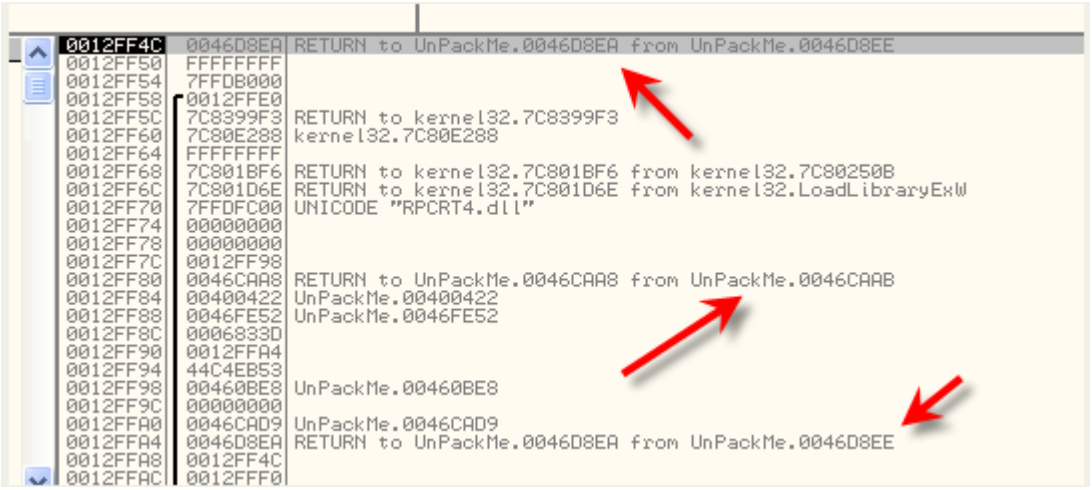
我们运行起来。



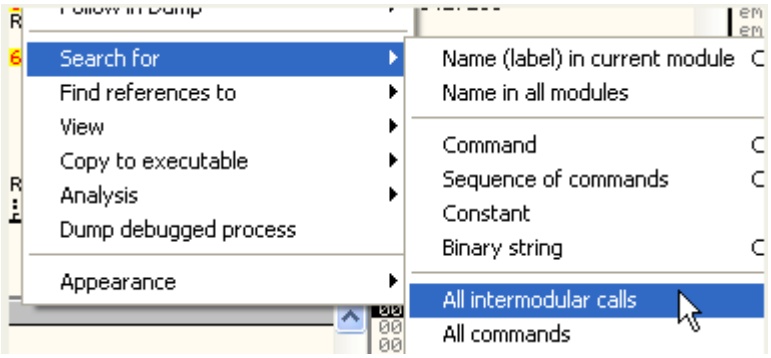
这里大家需要等待一段时间,不一会儿工夫断在了这里,应该是 OEP,但是入口点特征又不太像,应该是存在 Stolen Bytes。



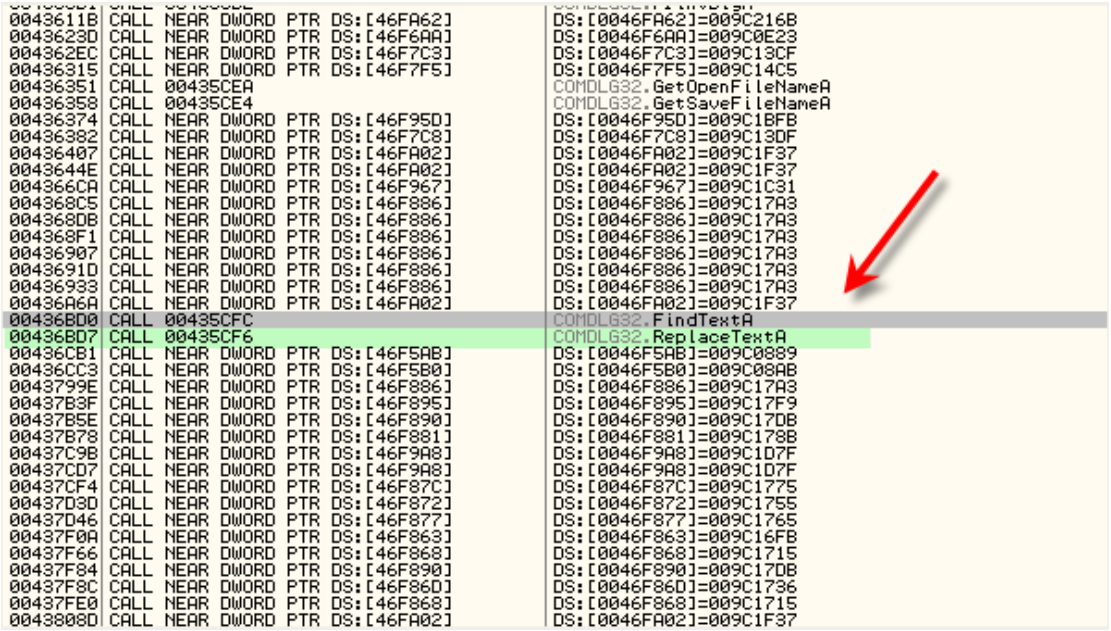
我们来看看堆栈:



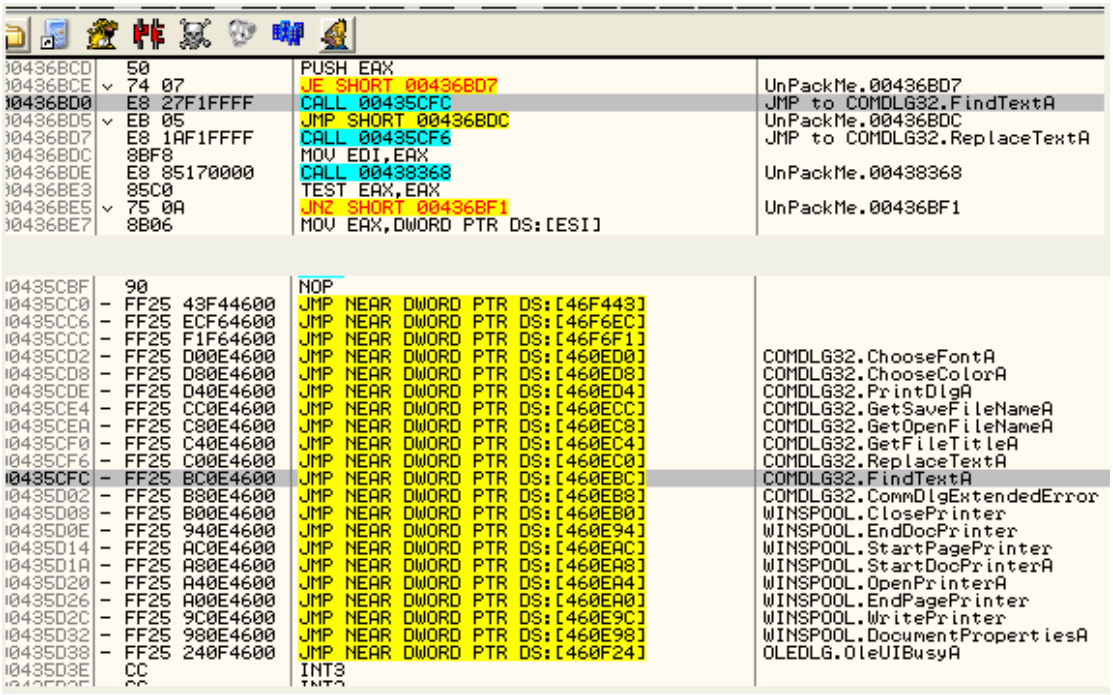
从堆栈中我们可以看到在到达伪造的 OEP 之前经过了多层函数调用,执行了很多代码,这些代码应该就是我们要找到的 stolen bytes。



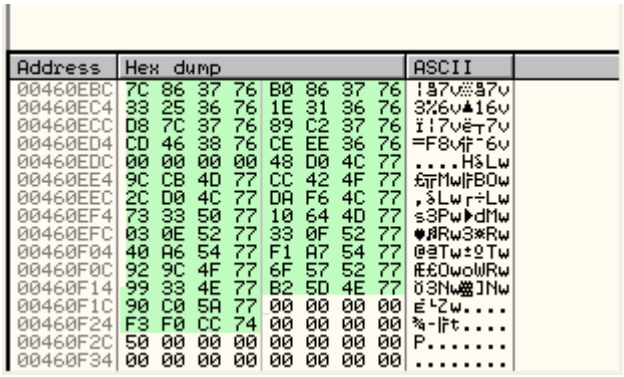
这里我们在反汇编窗口中单击鼠标右键选择 Search for-All intermodular calls,查看下主模块调用了哪些 API 函数。



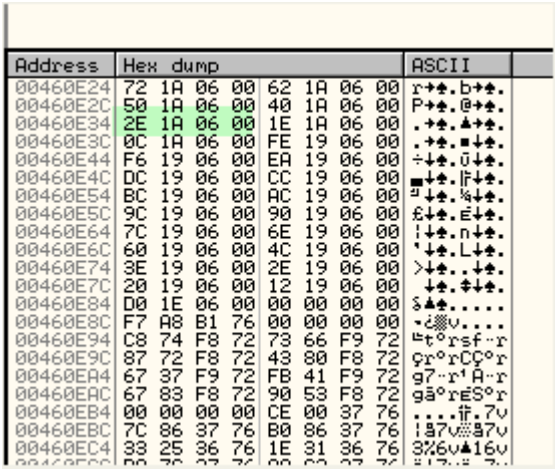
我们可以看到只有少量调用处显示了 API 函数名称,我们随便选择一个。



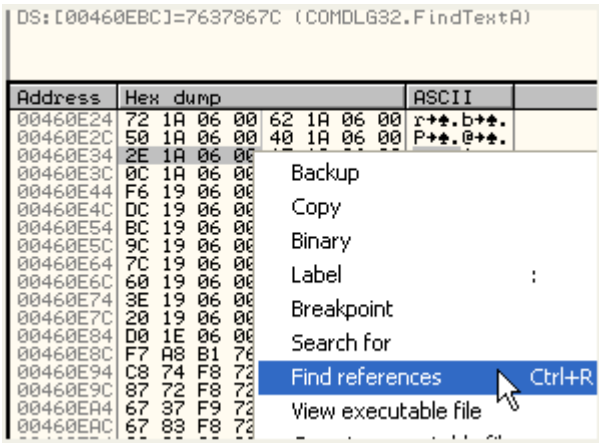
这里可以看到少量的间接调用 API 函数的指令-跳转表,我们任选一项定位到 IAT。



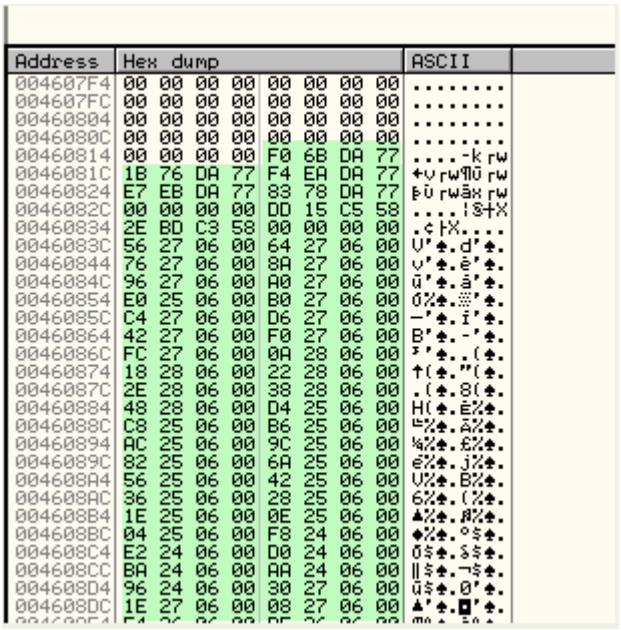
这里我们可以看到 IAT 的结束地址为 460F28,我们往上拉。



我们可以看到这些 IAT 项好像是经过重定向的,我们随便选中一项查看一下参考引用。

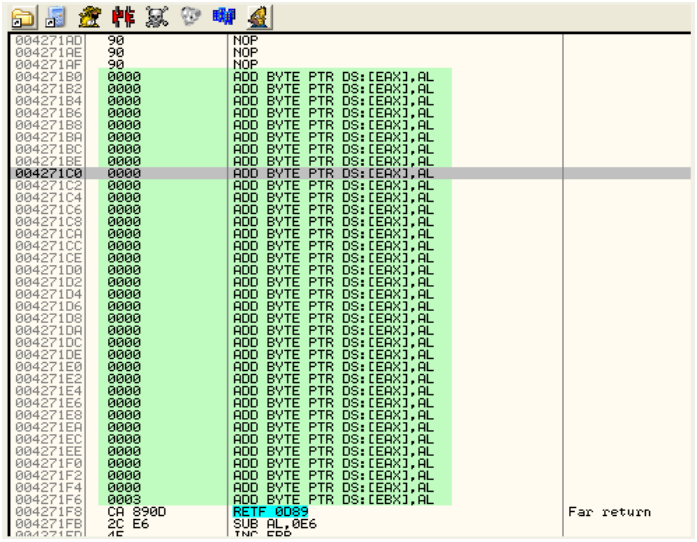


我们会发现这些项并没有参考引用处,我们继续往上翻,马上就可以看到 IAT 的起始地址了,所以说明刚刚这些查不到参考引用的项的确是 IAT 项,这些项被该壳重定向过了而已。



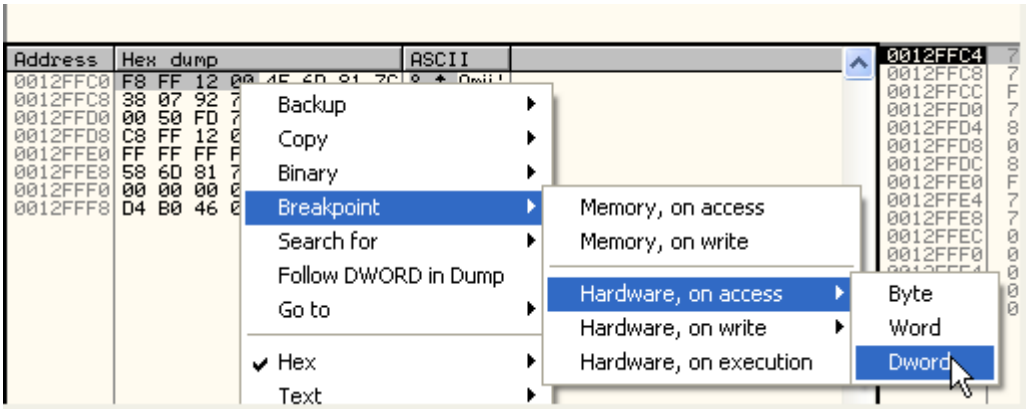
现在我们 IAT 的起始地址和结束位置都知道了,起始地址为 460818,结束地址为 460F28,下面我们的任务就是要修改 IAT 以及 stolen bytes 了。

首先我们来定位 stolen bytes,定位到伪造的 OEP 处,往上拉,可以看到一片零区域。

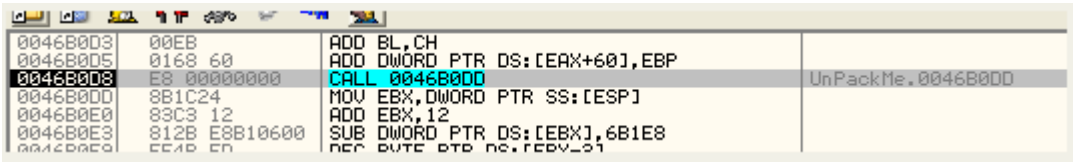


下面我们重启 OD,看下堆栈。

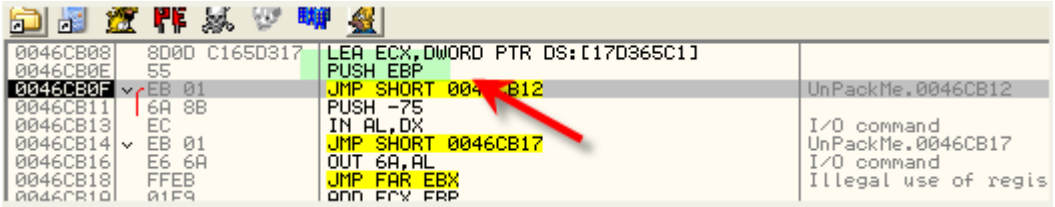
这里我们可以首次断在壳的入口点处时,栈顶指针指向了 12FFC4,也就是说,基于堆栈平衡的原理,当壳解密区段完毕,跳往真实的 OEP 处时,此时的栈顶指针也应该是指向的 12FFC4。我们知道通过情况下 OEP 处的第一个指令为 PUSH EBP,执行了这条指令后,栈顶指针应该指向的是 12FFC0,所以在数据窗口中定位到 12FFC0,给该内存单元设置硬件写入断点。如果该壳对硬件断点没有检测,我们这么做就没有问题,如果该壳对硬件断点有检测的话,就另当别论了。



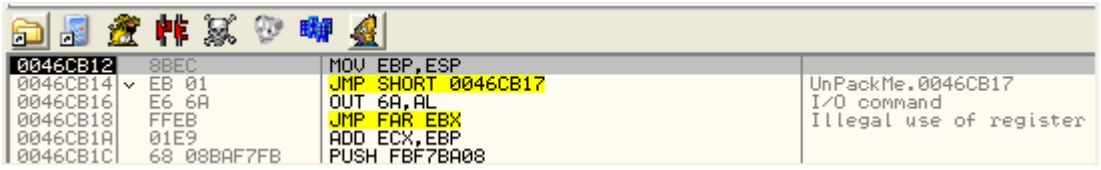
我们运行起来。

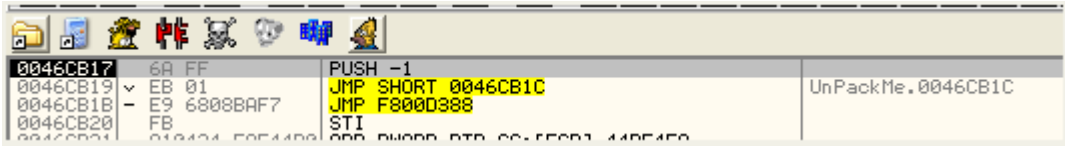


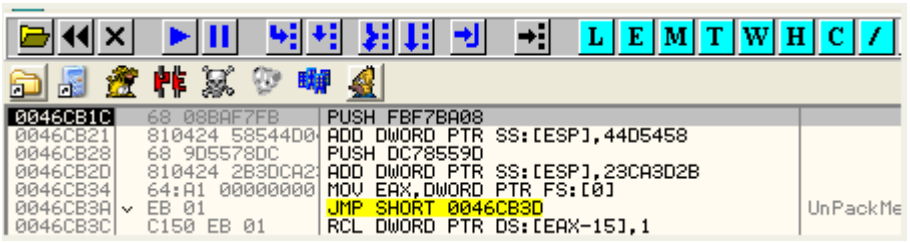
断在了这里,这里壳还没有解密完毕,我们继续按 F9 键运行。



断在了这里,这里有条 PUSH EBP 指令,有点像 OEP 处的指令,嘿嘿,我们继续往下单步跟踪,验证一下看看到底是不是 OEP 处的指令,对于无 JMP 指令我们并不会感兴趣,因为其并不影响寄存器组以及堆栈的状态。



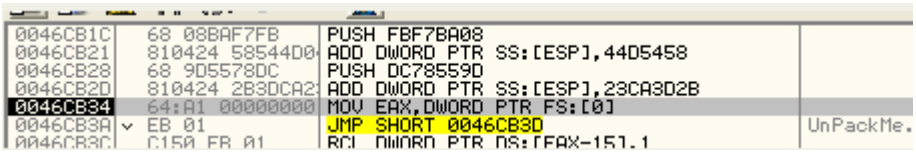




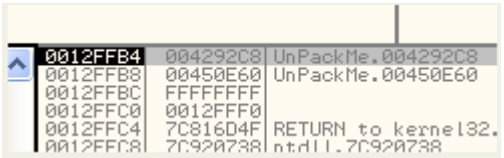
这里我们可以看到 PUSH FBF7BA08 这条指令,FBF7BA08 被压入堆栈以后,下面又要加上 44D5458 这个值。

我们来看看相加的结果等于多少。

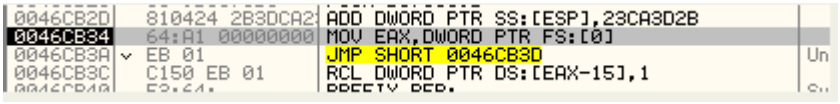
这里我们可以看到相加的结果等于 450E60,以上两条指令的作用等价于 PUSH 450E60。



接下来两条指令是类似的,等价于 PUSH 4292C8。



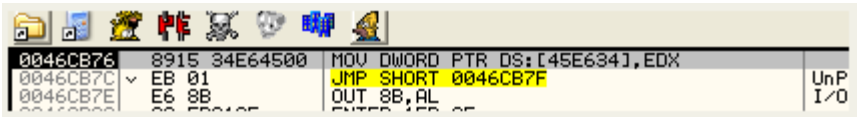
我们继续。

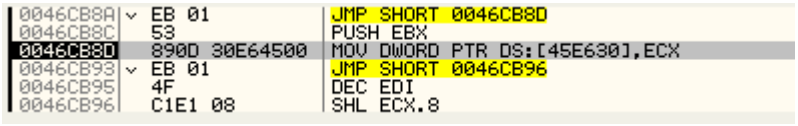


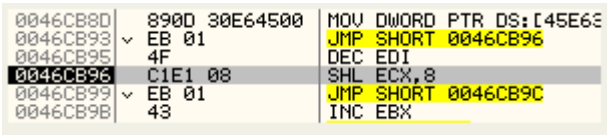
这里又是一条语句,我们继续。

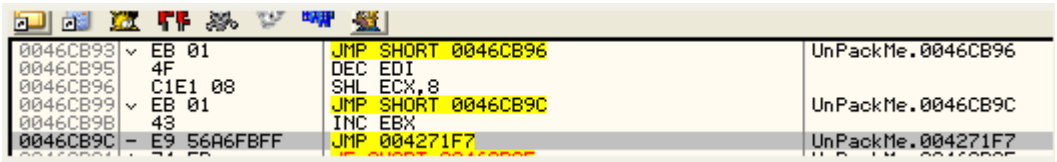
继续:

这里应该是调用一个 API 函数,等我们修改了 IAT 就可以看出来了。

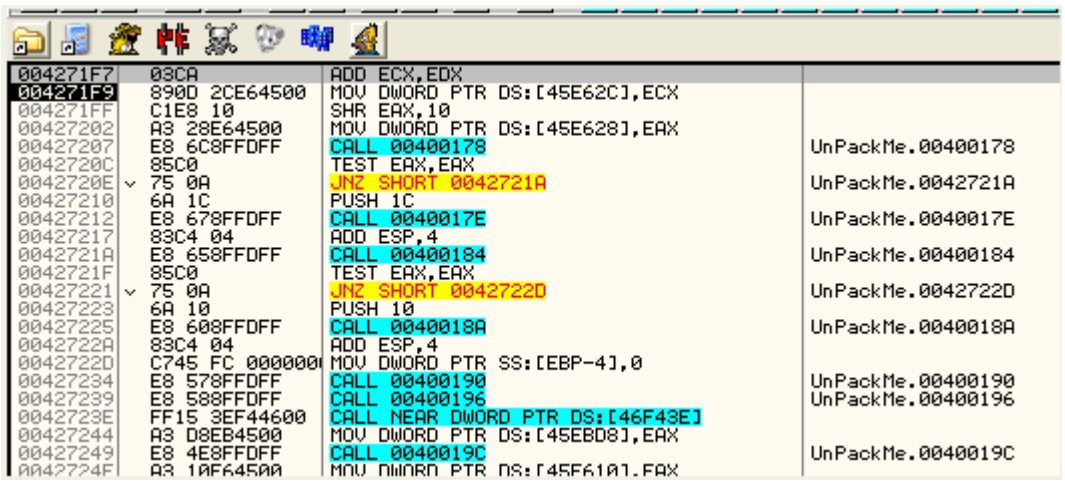






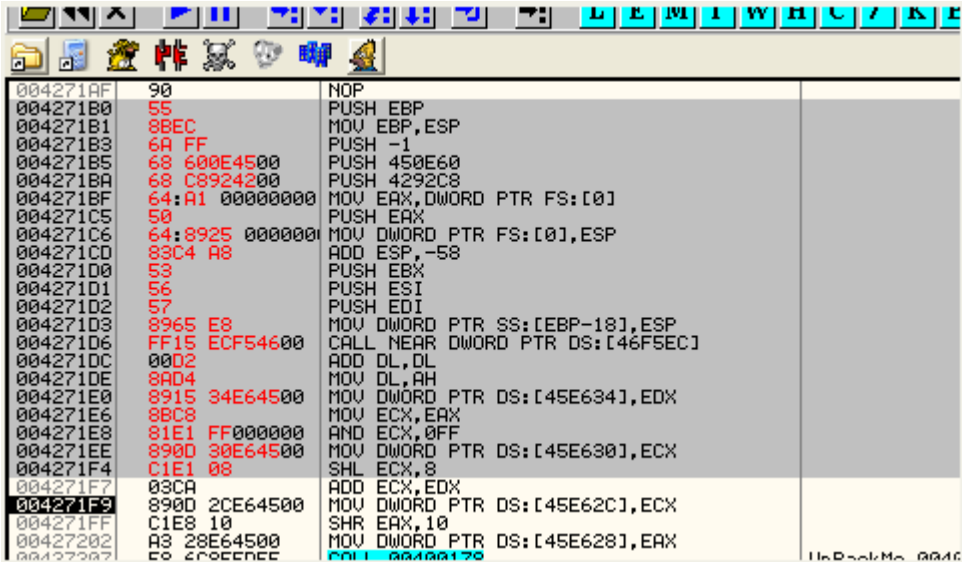


跟到了这里,可以看到这里将要跳往伪造的 OEP 了,嘿嘿,stolen bytes 都知道了。



```
004271F7    03CA            ADD ECX,EDX
004271F9    890D 2CE64500   MOV DWORD PTR DS:[45E62C],ECX
004271FF    C1E8 10         SHR EAX,10
00427202    A3 28E64500     MOV DWORD PTR DS:[45E628],EAX
00427207    E8 6C8FFDFF     CALL 00400178                   UnPackMe.00400178
0042720C    85C0            TEST EAX,EAX
0042720E    75 0A           JNZ SHORT 0042721A              UnPackMe.0042721A
00427210    6A 1C           PUSH 1C
00427212    E8 678FFDFF     CALL 0040017E                   UnPackMe.0040017E
00427217    83C4 04         ADD ESP,4
0042721A    E8 658FFDFF     CALL 00400184                   UnPackMe.00400184
0042721F    85C0            TEST EAX,EAX
00427221    75 0A           JNZ SHORT 0042722D              UnPackMe.0042722D
00427223    6A 10           PUSH 10
00427225    E8 608FFDFF     CALL 0040018A                   UnPackMe.0040018A
0042722A    83C4 04         ADD ESP,4
0042722D    C745 FC 000000( MOV DWORD PTR SS:[EBP-4],0
00427234    E8 578FFDFF     CALL 00400190                   UnPackMe.00400190
00427239    E8 588FFDFF     CALL 00400196                   UnPackMe.00400196
0042723E    FF15 3EF44600   CALL NEAR DWORD PTR DS:[46F43E]
00427244    A3 D8EB4500     MOV DWORD PTR DS:[45EBD8],EAX
00427249    E8 4E8FFDFF     CALL 0040019C                   UnPackMe.0040019C
0042724E    A3 10E64500     MOV DWORD PTR DS:[45E610],EAX
```

我们将 stolen bytes 都拷贝到 OEP 处。



```
004271AF    90              NOP
004271B0    55              PUSH EBP
004271B1    8BEC            MOV EBP,ESP
004271B3    6A FF           PUSH -1
004271B5    68 600E4500     PUSH 450E60
004271BA    68 C8924200     PUSH 4292C8
004271BF    64:A1 00000000  MOV EAX,DWORD PTR FS:[0]
004271C5    50              PUSH EAX
004271C6    64:8925 000000( MOV DWORD PTR FS:[0],ESP
004271CD    83C4 A8         ADD ESP,-58
004271D0    53              PUSH EBX
004271D1    56              PUSH ESI
004271D2    57              PUSH EDI
004271D3    8965 E8         MOV DWORD PTR SS:[EBP-18],ESP
004271D6    FF15 ECF54600   CALL NEAR DWORD PTR DS:[46F5EC]
004271DC    00D2            ADD DL,DL
004271DE    8AD4            MOV DL,AH
004271E0    8915 34E64500   MOV DWORD PTR DS:[45E634],EDX
004271E6    8BC8            MOV ECX,EAX
004271E8    81E1 FF000000   AND ECX,0FF
004271EE    890D 30E64500   MOV DWORD PTR DS:[45E630],ECX
004271F4    C1E1 08         SHL ECX,8
004271F7    03CA            ADD ECX,EDX
004271F9    890D 2CE64500   MOV DWORD PTR DS:[45E62C],ECX
004271FF    C1E8 10         SHR EAX,10
00427202    A3 28E64500     MOV DWORD PTR DS:[45E628],EAX
```

好了,我们可以看到空间刚刚好,这里 stolen bytes 我们就修复完毕了,下一章节我们来修改 IAT。