问：用IDA静态分析,函数结尾出现 endp ; sp-analysis failed 用F5调不出伪代码,不知道是什么原因,请问有什么解决办法没有？

答：endp ; sp-analysis failed.

一般是程序代码有一些干扰代码，让IDA的反汇编分析出现错误。比如用push + n条指令 + retn来实际跳转，而IDA会以为retn是函数要结束，结果它分析后发现调用栈不平衡，因此就提示sp analysis failed.

我还遇到过一种情况，是因为编译器优化，结果IDA无法正确识别一个函数体的结尾部分，换句话说，它找不

到C中的“大括号”应该位于哪里。例如以下代码：

int one\_function( int a,int b);

int another\_function( int a, int b)

{

if ( a == 0 || b == 0 )

return -1;

return one\_function(a,b);

}

其中return one\_function(a,b)这条语句，在某些新的编译器，可能会编译成这样的指令序列：

mov esp, ebp

pop ebp

jmp one\_funcion

而IDA是通过retn指令来识别函数的结束的，因为它不知道这里的意思，会把它当成一个函数内部 的跳转，最后就会出现sp analysis failed了。

问：有没有什么办法可以让IDA正确分析,因为函数非常长,需要伪代码来减少分析时间？

答：用alt-k调整sp来解决。

问：如何用ALT+K 调整SP 要输入什么值呢?

答：并没有一个固定值,针对每一条指令执行完后,看看栈是否正常,如果不对,则通过ALT + K来修改.重点检查虚函数的调用, 如call [esi + n] , 这里不一定非得是esi,以及跳转前后的栈是否一致.另外还需要通过ALT + P 来确认下变量起始地址,清除个数与保存个数是否正常.

1）用Option->General->Disassembly, 将选项Stack pointer打勾；

2）仔细观察每条call sub\_xxxxxx前后的堆栈指针是否平衡；

3）有时还要看被调用的sub\_xxxxxx内部的堆栈情况，主要是看入栈的参数与ret xx是否匹配；

4）注意观察jmp指令前后的堆栈是否有变化；

5）有时用Edit->Functions->Edit function...,然后点击OK，(按D再按C键)刷一下函数定义。

以上是转载。小结下：SP是堆栈指针，上面的问题就是堆栈不平衡，其实问题解决的主要思路就是修正SP，其实勾选Stack pointer选项后每行汇编代码和指令地址之间会多一列数字，这些数值就是SP实际的偏移值。call sub\_xxx前面如果显示-0x07H，则将光标停在紧接着这个call后面的一行指令之前(即刚返回的地方)，按Alt+K设置新的SP值为-0x07H，再按D按C刷新下函数定义就可以了。