逆向初体验

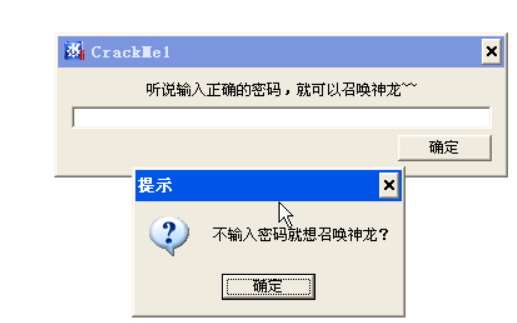
**题目描述：**

主机C:\Reverse\1目录下有一个CrackMe1.exe程序，运行这个程序的时候会提示输入一个密码，当输入正确的密码时，会弹出过关提示消息框，请对CrackMe1.exe程序进行逆向分析和调试，找到正确的过关密码。



反正搞坏了不用负责任，直接上手！

首先是空密码

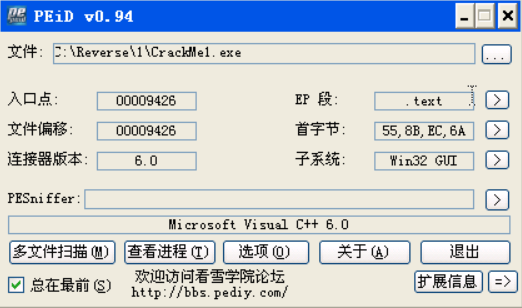


再试一下测试密码

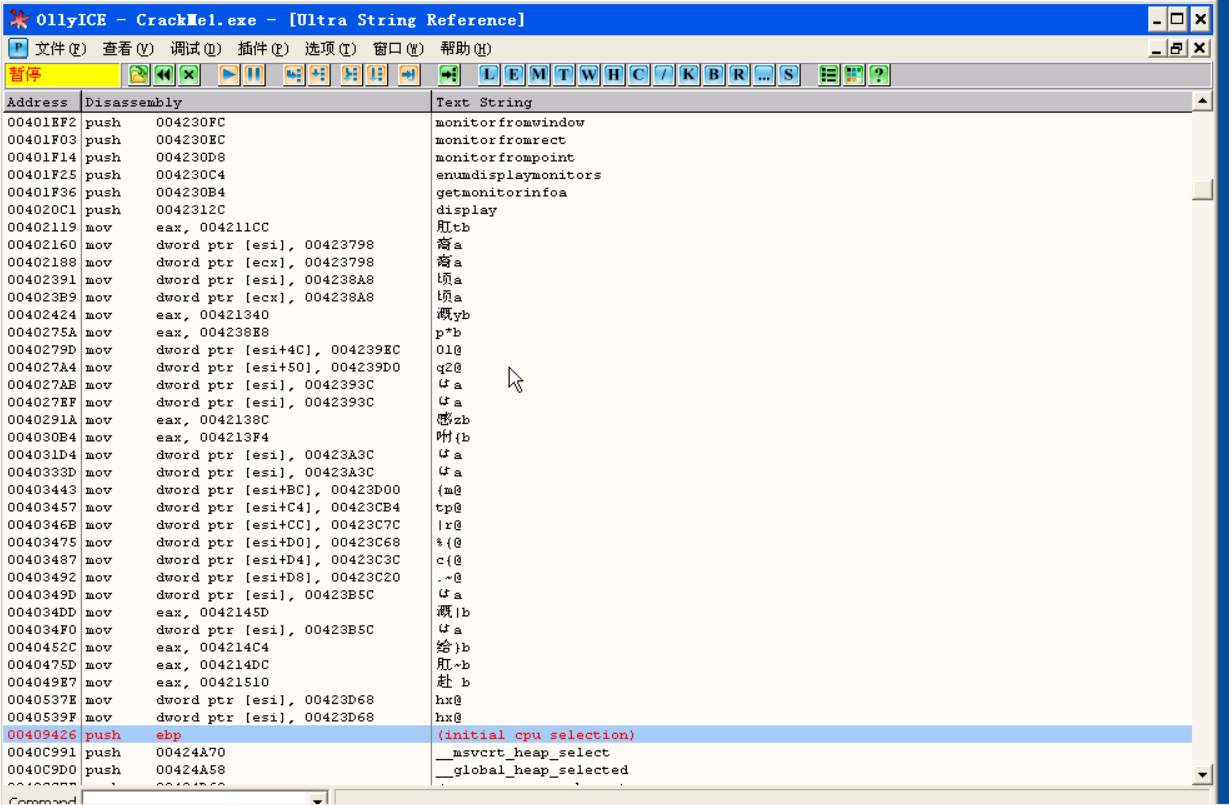


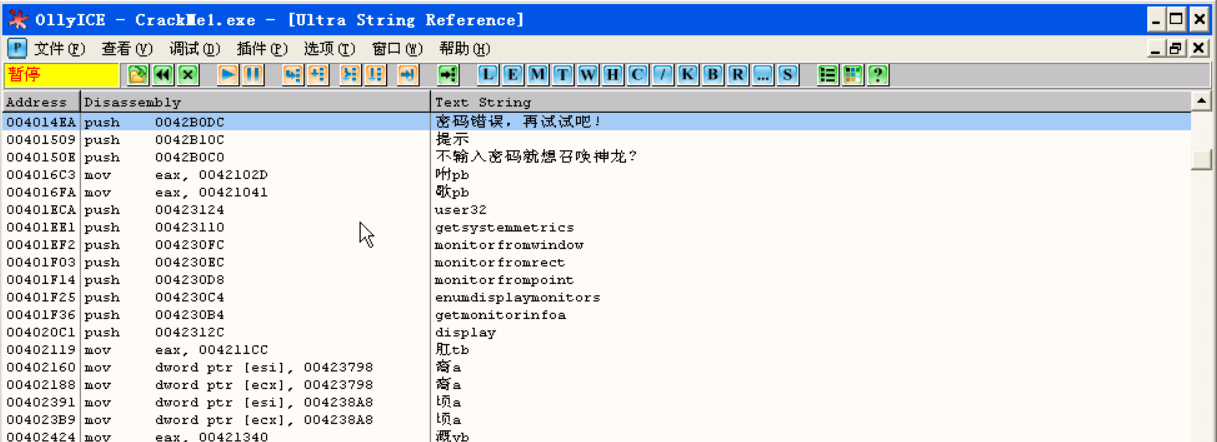
通过这些操作后，我们可以猜测程序的意图是输入一个正确的密码。在对一个程序进行逆向分析之前，除了程序的动态行为之外，查看程序是否加壳（被何种程序加壳？是什么编译器编译的？）也是一个非常关键的步骤。

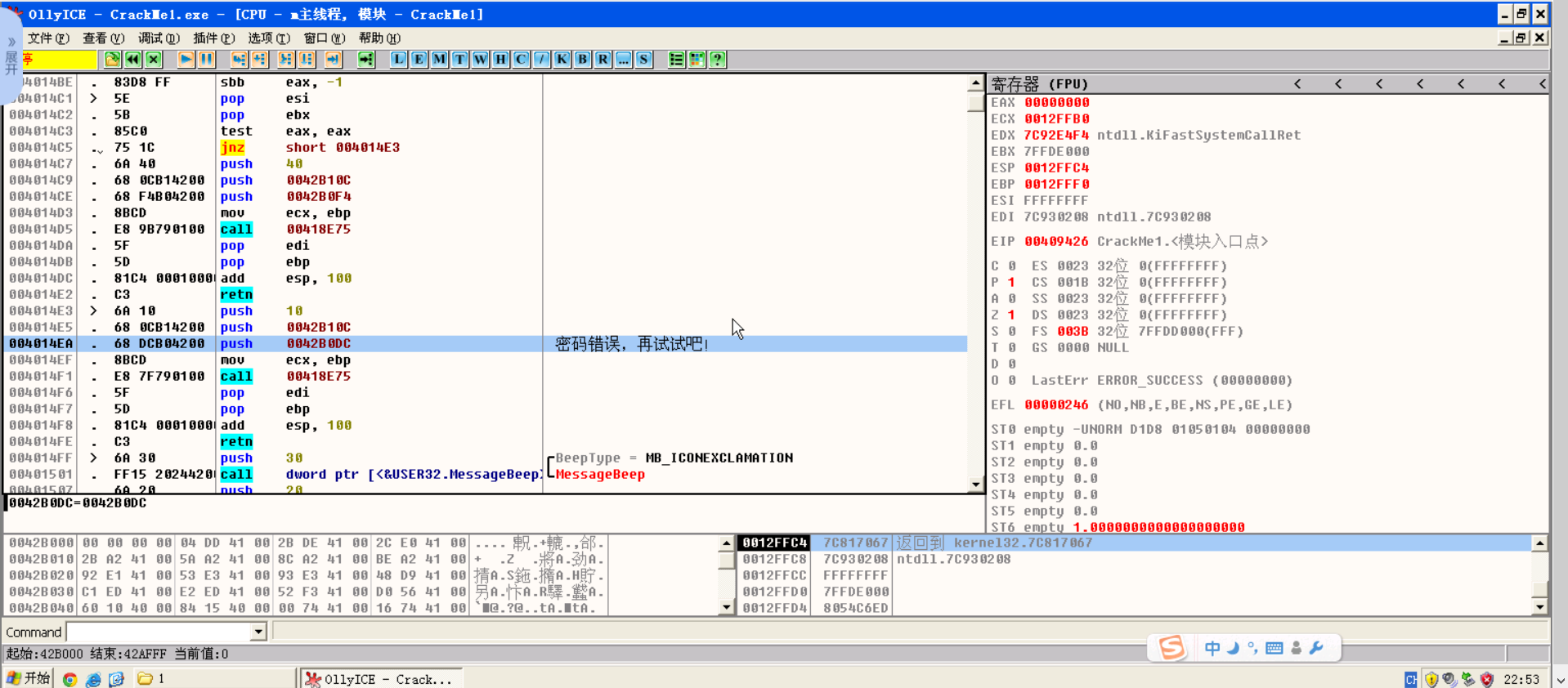
先看一下有没有壳，用peid查看一下（这一题暴力破解其实也可以，但是是逆向题，所以还是乖乖分析吧）



是没有壳的，编译器是Microsoft Visual C++ 6.0,不用脱壳，直接用OOlyICE动态调试

OD调试器的主界面，在反汇编指令列表窗口中单击右键，依次选择“Ultra String Reference”、“Find ASCII”菜单项，进入如下界面

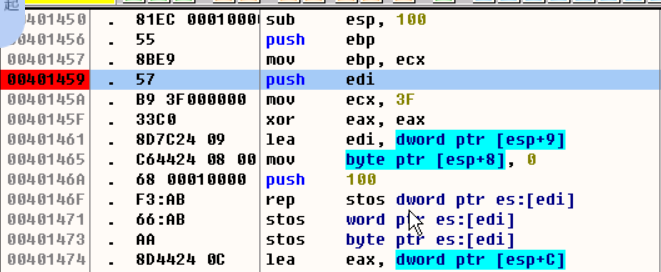
弹出字符串列表窗口，这个窗口列出了当前进程内存空间中存在的各种字符串，我们可以查看是否存在有我们感兴趣的字符串。比如前面曾经提示“密码错误”的提示，那么我们就可以按下Ctrl+F，在弹出的窗口中输入“密码错误”，然后单击确定按钮查找：

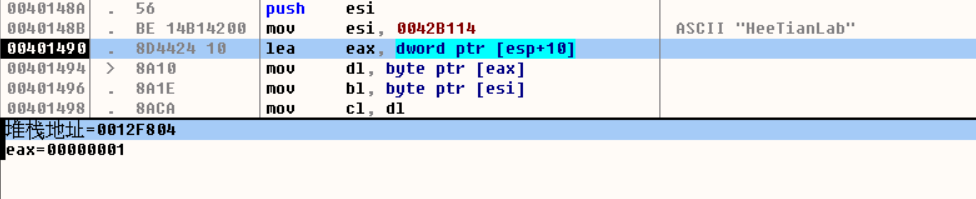
双击找到这一行，就会进入OD的反汇编窗口，就能看到代码指令中引用这个字符串的地方了。如下图所示：

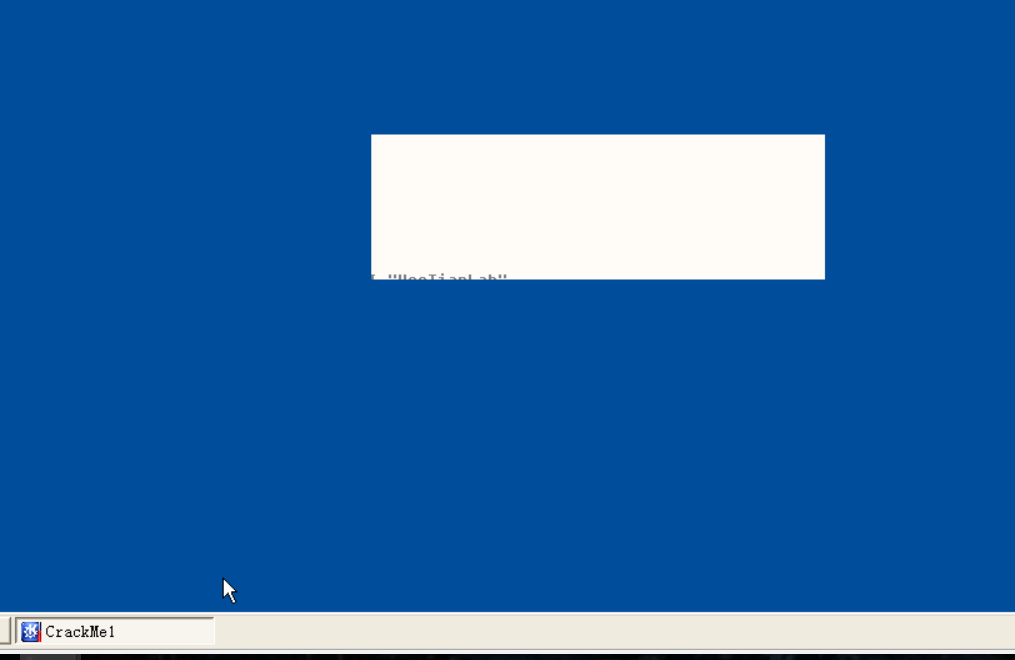
在这上面，可以看到有一个jnz跳转指令，这个指令决定着弹出成功还是弹出失败，也被称为“关键跳转”在关键跳转之上的代码往往就是关键的密码判断逻辑，因此我们可以着重分析关键跳转之上的代码。在关键跳转之上，我们在下面的位置下一个断点（鼠标单击这行代码选中，然后按下F2）：

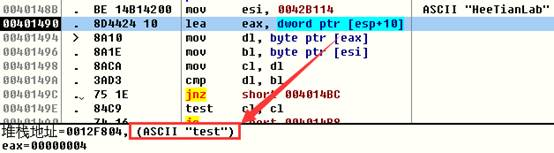
**00401456** . 55 push ebp

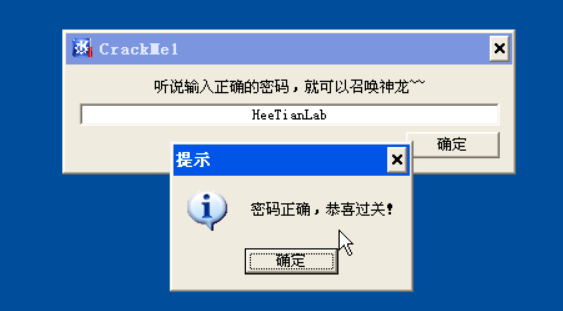
\\这里找到这一段不是爱，不是责任，只是因为这是jnz跳转所在函数的入口。

\\一般一个函数的开头都是push ebp 保存函数栈帧

下好断点之后，按下F9运行程序，随便输入一个密码（比如test）后单击按钮，程序就自动在我们的断点了，接下来按F8开始进行单步跟踪，当跟踪到00401490的时候，我们发现了我们输入的密码test，同时发现字符串HeeTianLab，如图所示：

对不起，合天的实验机内存崩了，**全剧终！！！**

开个玩笑，大家刀收一收。理论情况是这样

仔细分析上面的代码，发现是在从这两个字符串里面取出字符一个一个进行对比，只要有一个字符不一样，最终都会跳转到提示失败的地方去。那么可以猜测HeeTianLab就是正确的密码了，我们运行另一个CrackMe1进程，输入HeeTianLab，弹出成功提示：

初次入门，只是一次简单的逆向分析，如果实验过程有错误的地方请斧正。

--------------------------------------------------------------颜总的第一次分享