《漏洞利用及渗透测试基础》实验报告

姓名：赵悦蛟 学号：2313650 班级：1071

**实验名称：**

OLLYDEBG软件破解

**实验要求：**

根据第三章示例3-1，

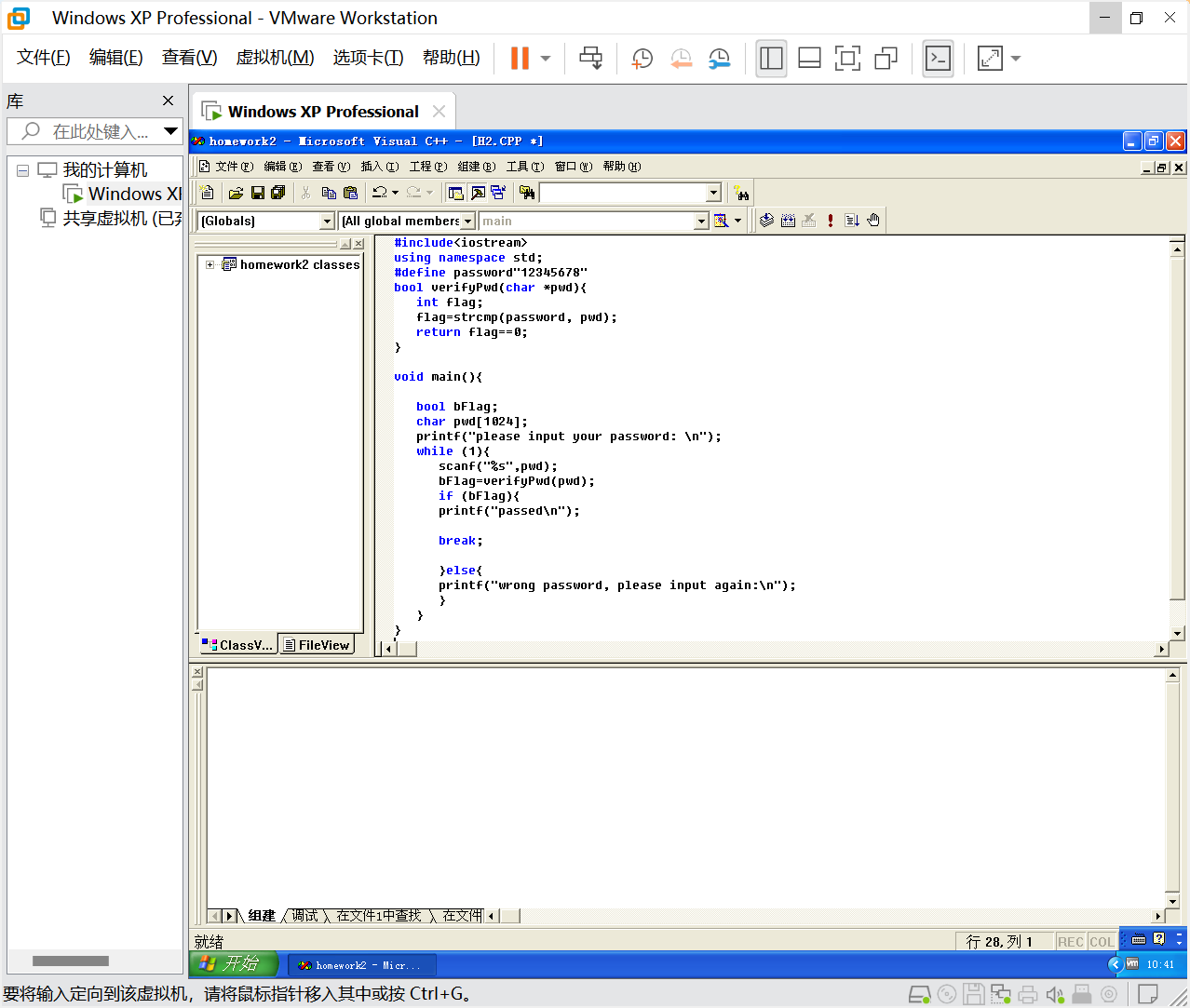
1.在XP VC6生成课本第三章软件破解的案例(DEBUG模式，示例3-1）。进而，使用OllyDBG进行单步调试，获取verifyPWD函数对应flag==0的汇编代码，并对这些汇编代码进行解释。

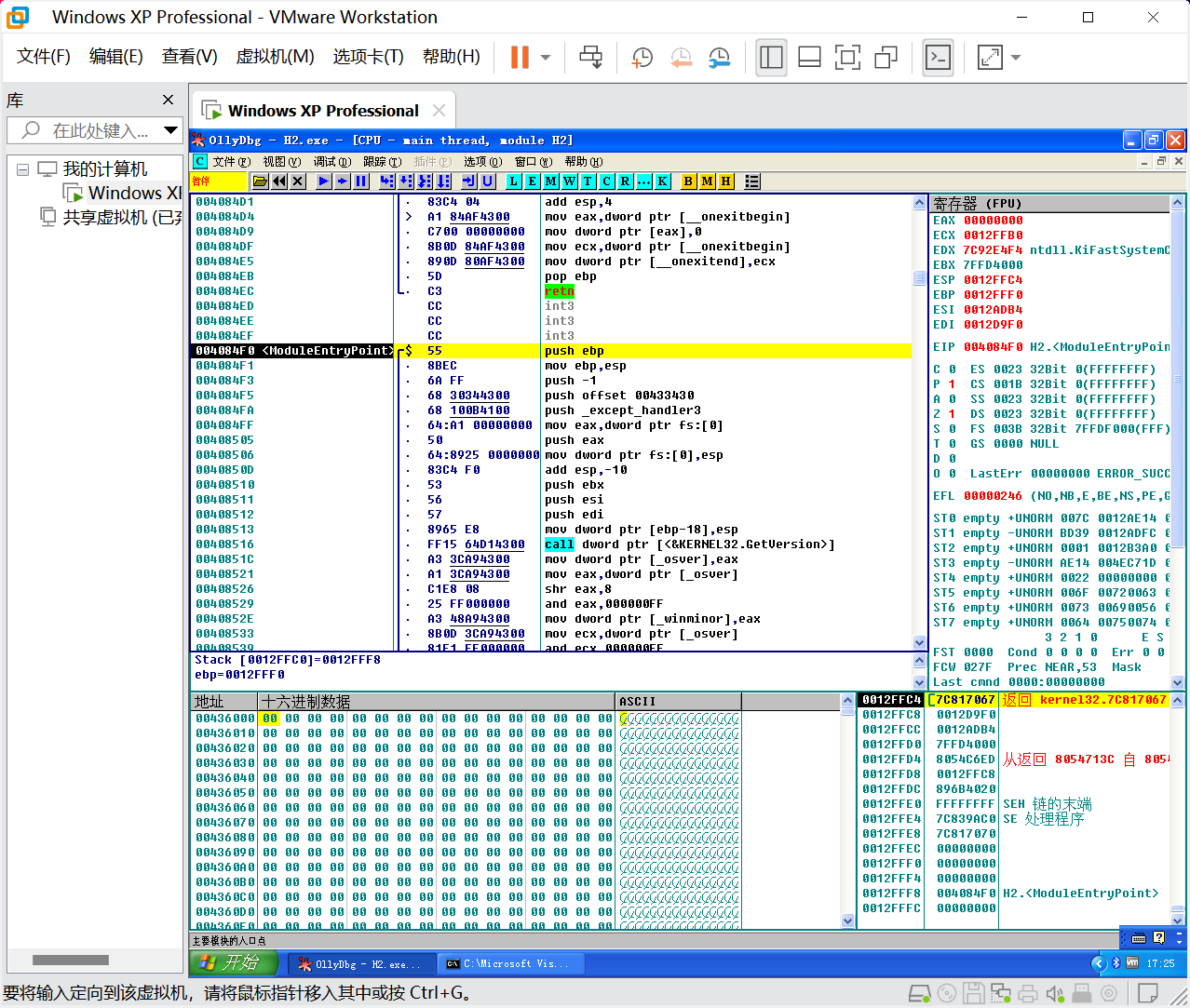
2. 对生成的DEBUG程序进行破解，复现课本上提供的两种破解方法。

**实验过程：**

1. 在虚拟机中下载ollydbg

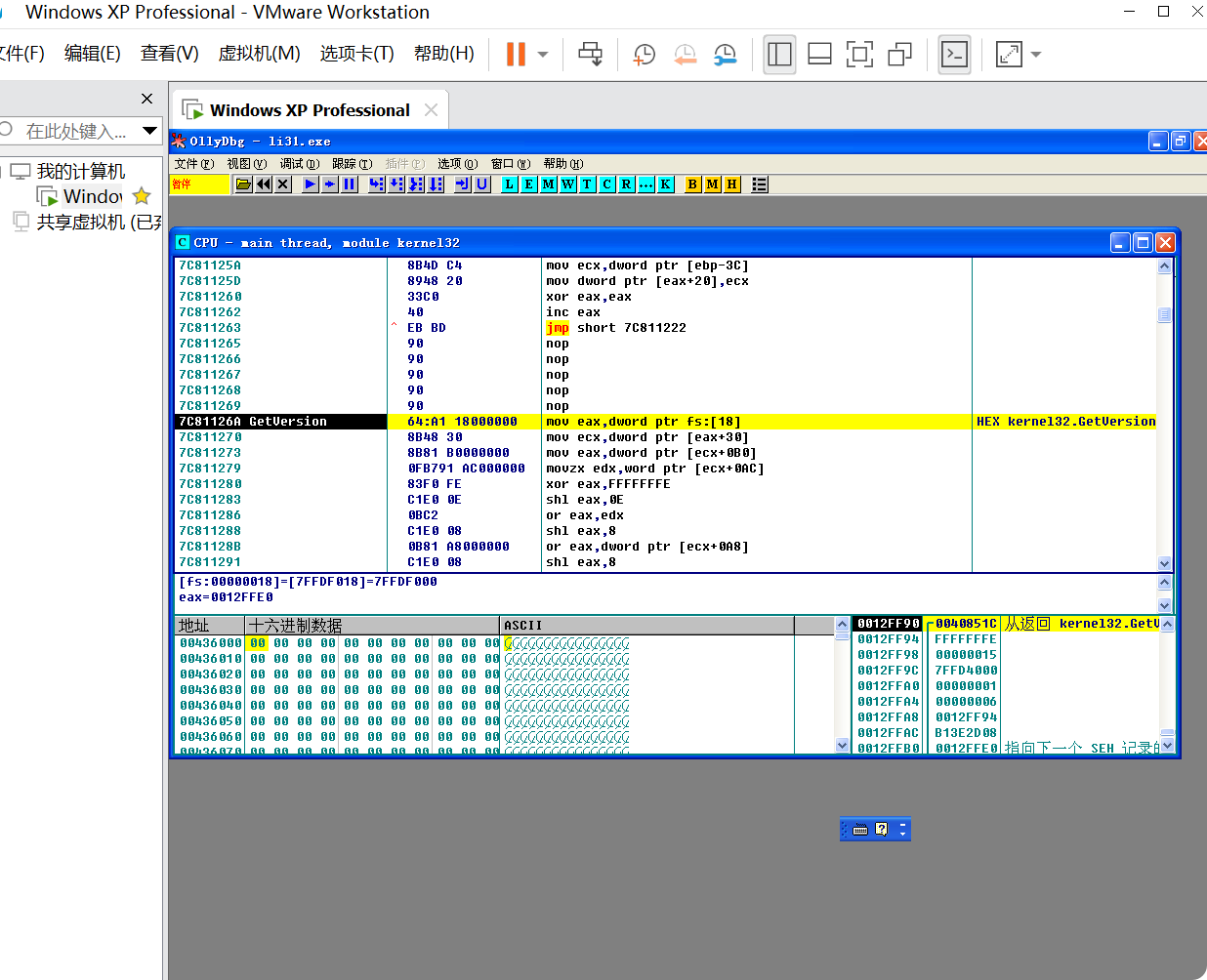
2.在vc6中编写好代码，用ollydbg打开

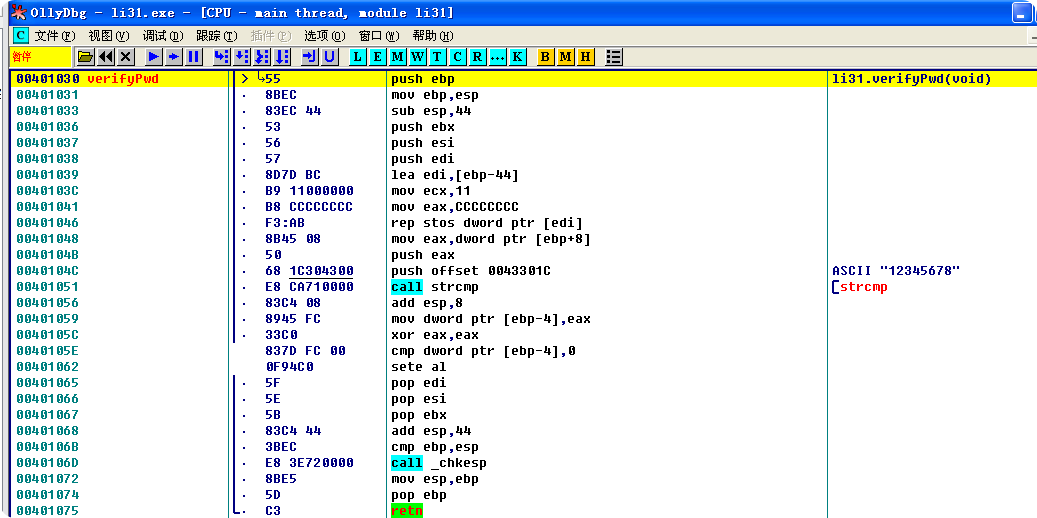




1. 单步调试

使用F7键对函数进行单步调试，





找到了verifyPwd函数起始的位置，其中：

Xor eax,eax ：xor是取“异或”，eax和自身取异或得到的值为0，通过这条语句将eax寄存器清零，相当于将eax寄存器初始化，为接下来储存“flag==0”返回的结果提供空间。

cmp dword ptr [ebp-A],0 ：cmp是比较指令，用于比较两数的值。dword ptr [ebp-A]表示从内存地址[ebp-A]处读取一个双字（4字节）的值，即局部变量flag。这条指令是在比较flag和0。

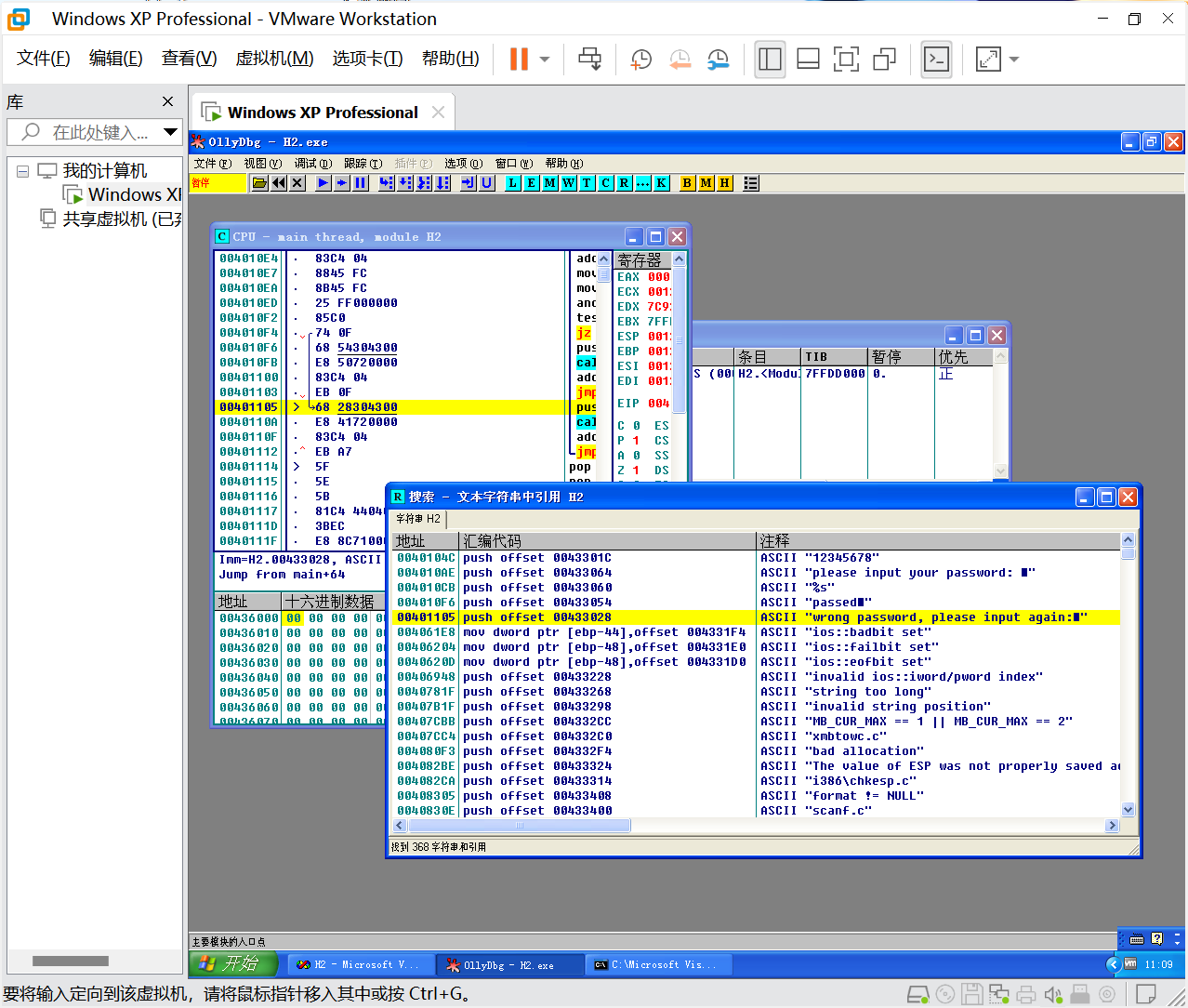
sete al ：sete是“根据相等条件设置字节”指令，al是eax寄存器的低8位，这条指令会根据 Flag的值来设置al如果flag == 0，将al设置为1，否则设置为0。

并且al的值会被扩展到eax，作为函数的返回值。

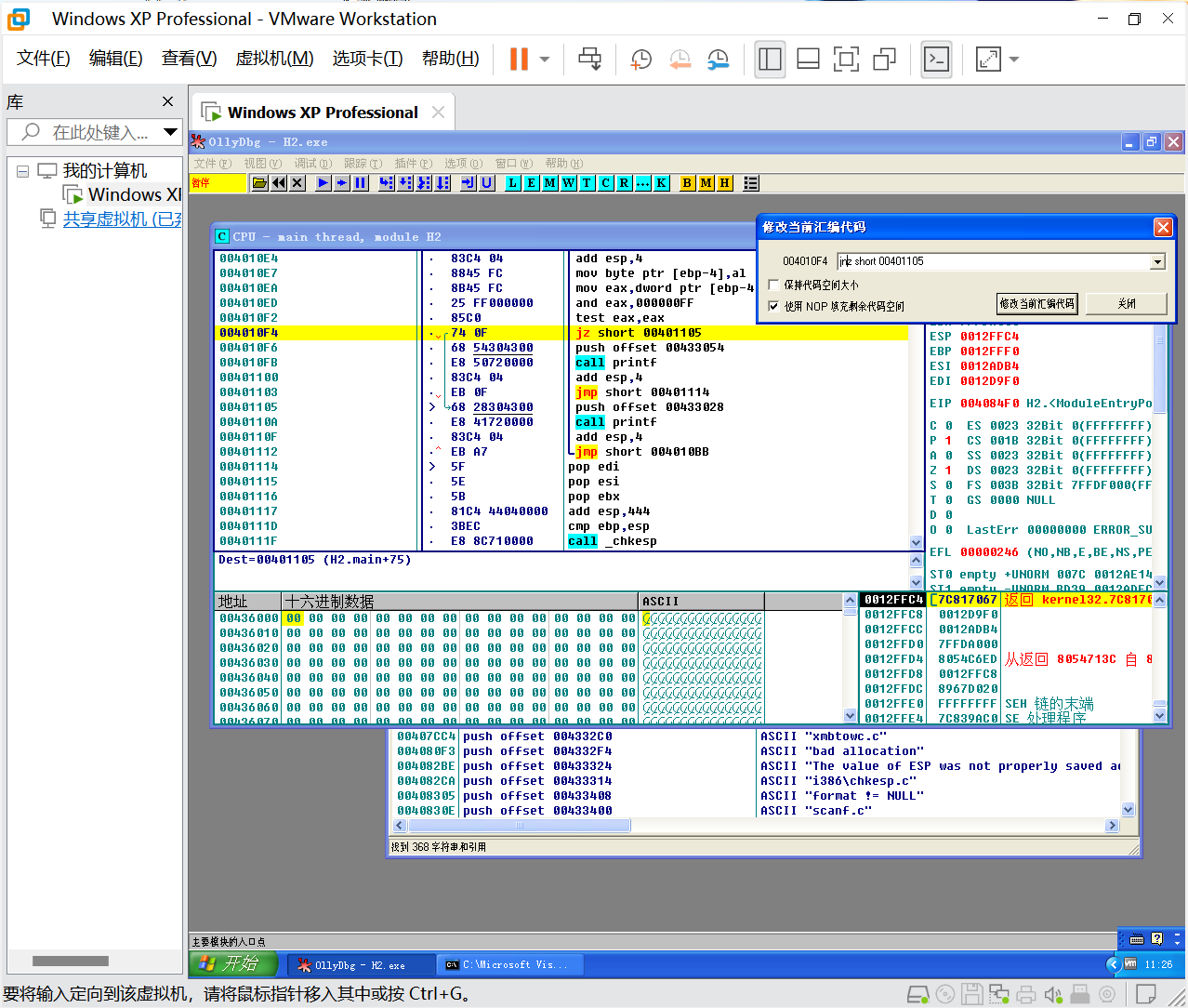
这三条语句实现了源代码中“flag==0”的判断

4.复现破解方法1：

（1）定位到wrong出现的位置



（2）找到jz short 00401105，这是如果密码正确则要跳转的位置，jz指令表示当零标志位（ZF）为1时跳转，即前一条指令的结果为零时跳转到相应的地址即输入的密码正确时跳转。而对应的jnz指令表示零标志位为0时跳转，即前一条指令的结果不为0，将jz short 00401105改为jnz short 00401105，这样一来输入错误的密码就反而跳转到验证成功的分支语句中

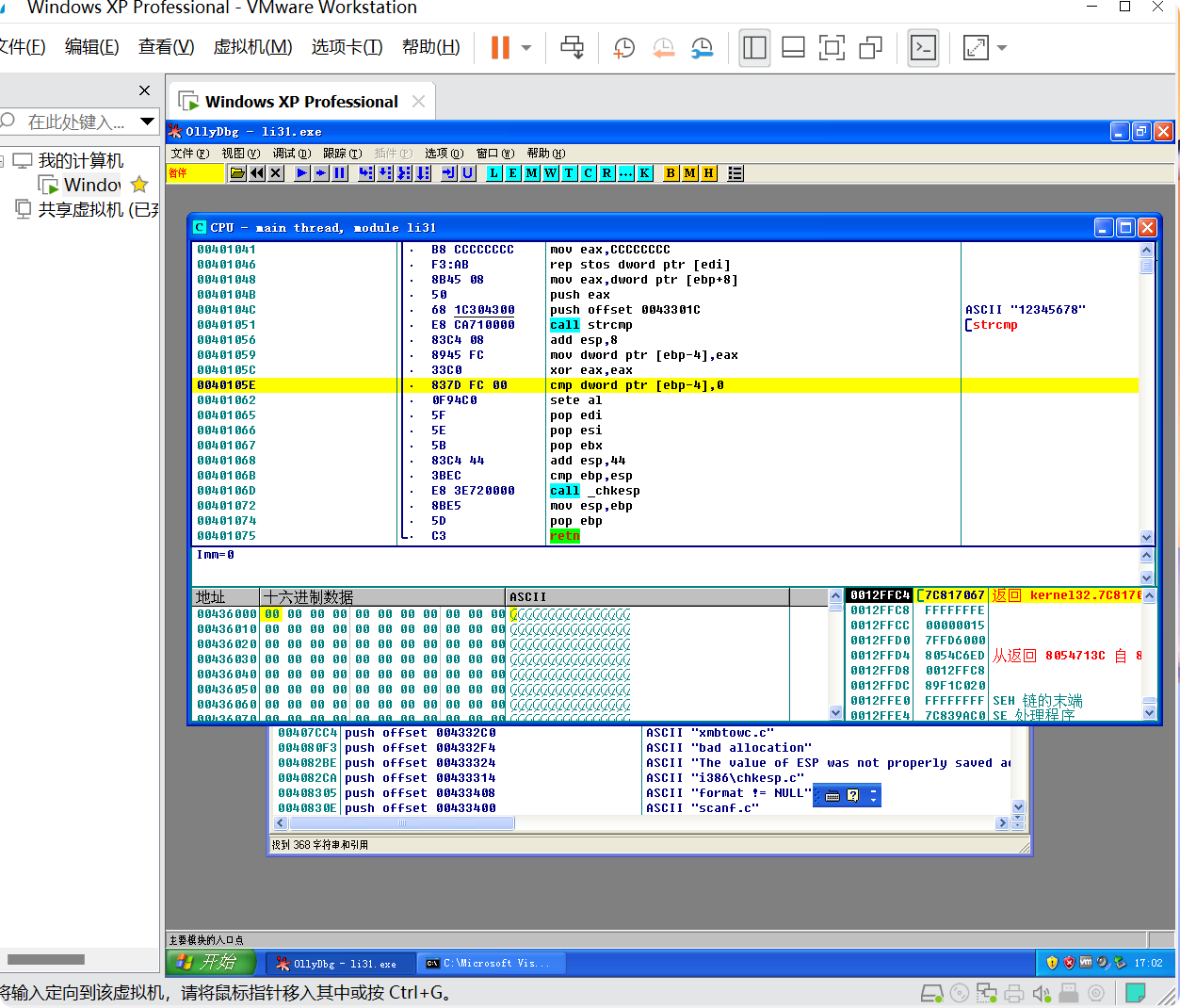


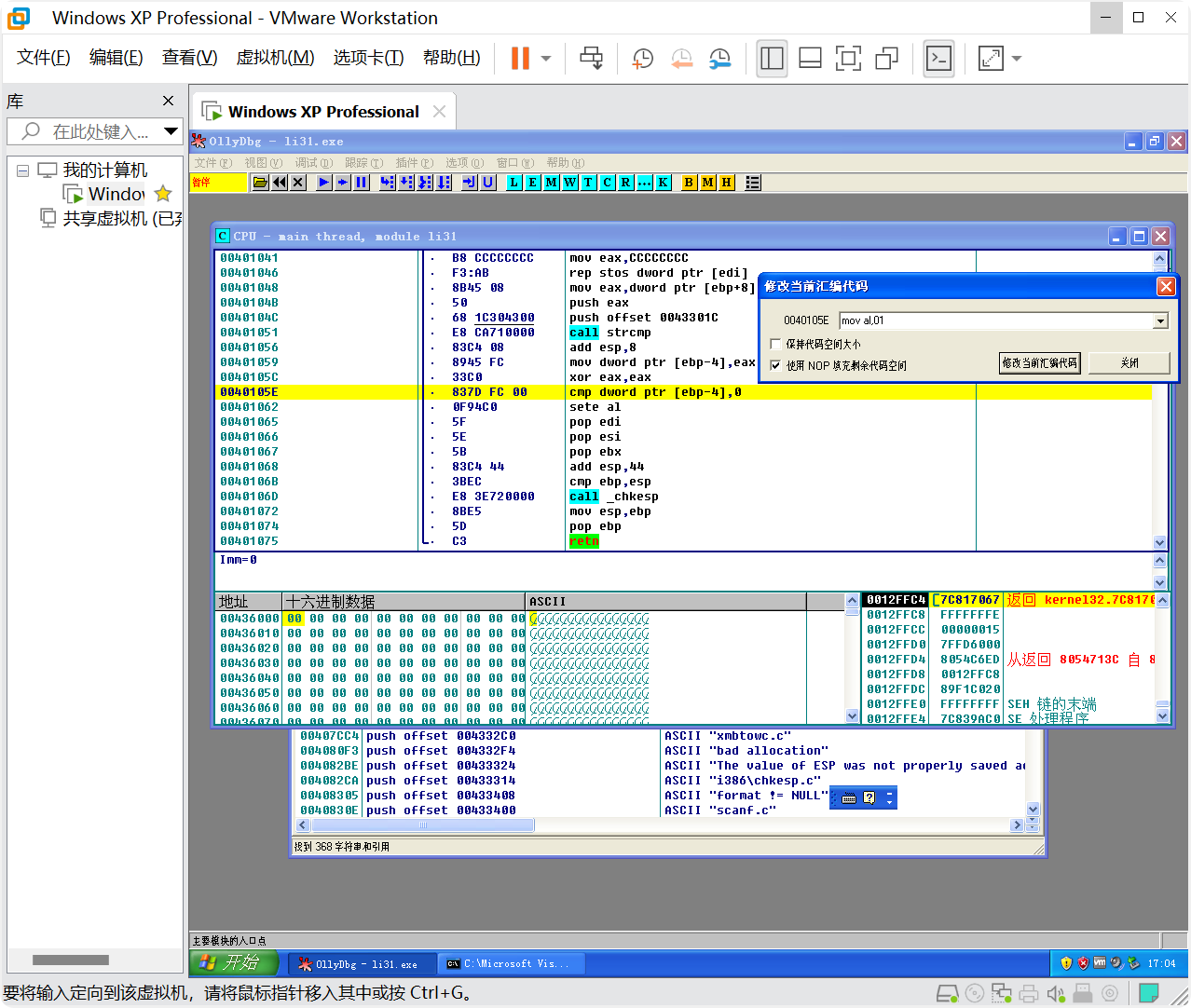
5.复现破解方式2：

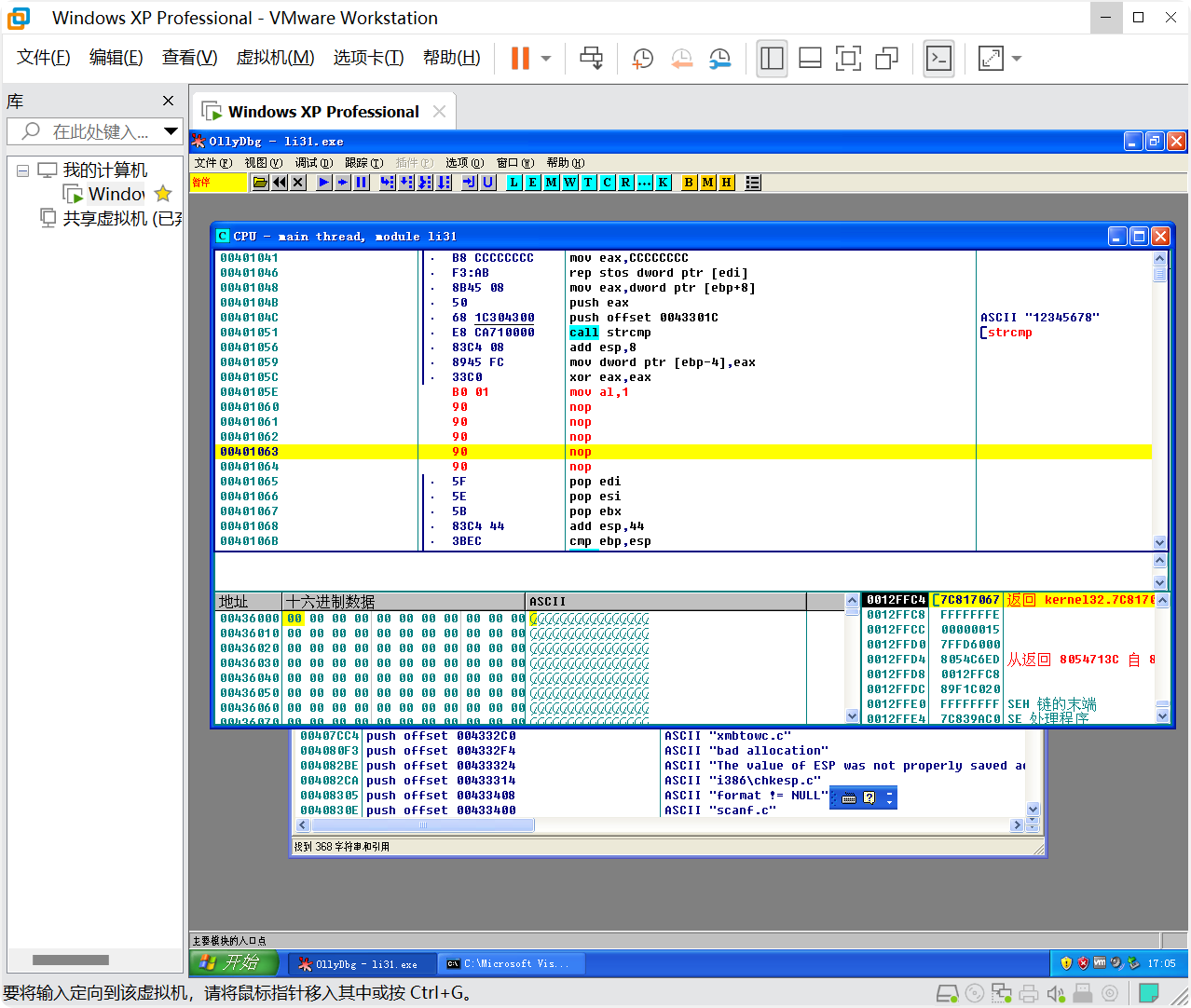
（1）选择“跟随”命令逐步进入该函数

（2）通过修改函数的方式进行破解。cmp 指令用于比较 [ebp-4] 的值与 0，如果相等，则设置零标志位（ZF）为1。sete al 会根据ZF的值将 al 寄存器设置为1（相等）或0（不相等）通过将 cmp dword ptr [ebp-4], 0 修改为 mov al, 01，程序不再进行比较，而是直接将 al 设置为1，表示条件成立。同时，将 sete al 修改为 nop（空操作），使其不再根据ZF修改 al 的值。

这样修改后，程序跳过了密码验证的逻辑，直接认为条件成立（al 始终为1），从而绕过验证，实现破解。无论输入什么密码，程序都会认为验证成功。从而完成破解。







**心得体会：**

通过本次实验，我学到了如何使用OllyDBG，学习了如何用ollydbg检查和简单的破解程序。还学会了ollydbg内部许多快捷键的使用方式等。在调试过程中，我熟悉了如何定位关键函数（如verifyPWD），并分析其汇编代码，进一步理解了条件判断（如flag==0）在汇编层面的实现方式。

对汇编语言的分析让我对其有了更好的了解与认识，记住了jz、jnz、xor、sete等指令的含义和作用

此外，通过复现课本上的两种破解方法，我掌握了如何通过修改汇编指令或内存数据来改变程序的行为，进而实现破解。这不仅加深了我对软件破解原理的理解，也让我更加熟悉了OllyDBG等调试工具的使用。