《漏洞利用及渗透测试基础》实验报告

姓名：申宗尚　学号：2213924　班级：信息安全

实验名称：

OLLYDBG 软件破解实验

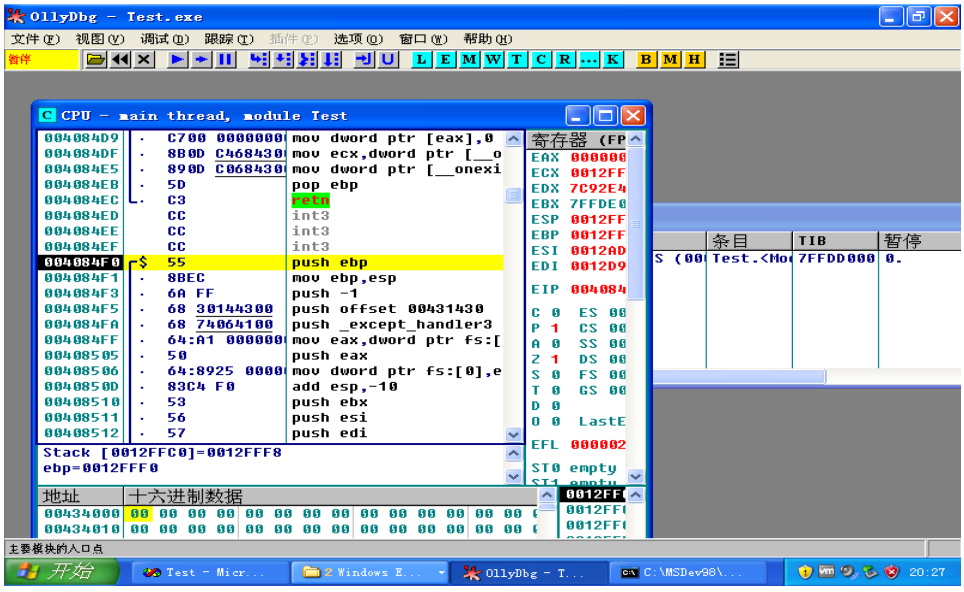
实验要求：

请在 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。

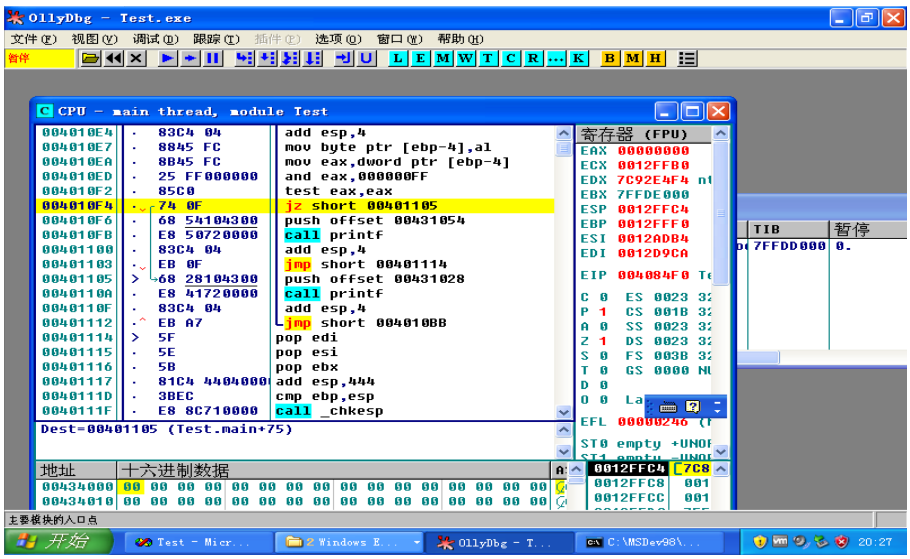
对生成的 DEBUG 程序进行破解，复现课本上提供的两种破解方法。

实验过程：

1. 进入 OllyDBG，打开提供的 DEBUG 程序。

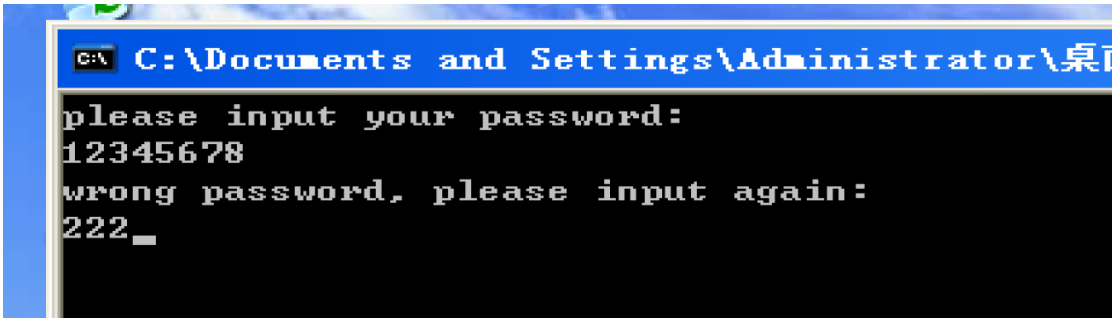


2. 由于 DEBUG 程序输入错误密码时会输入：wrong password...，通过 ollyDBG 的查找字符串功能，定位字符串“wrong password…”



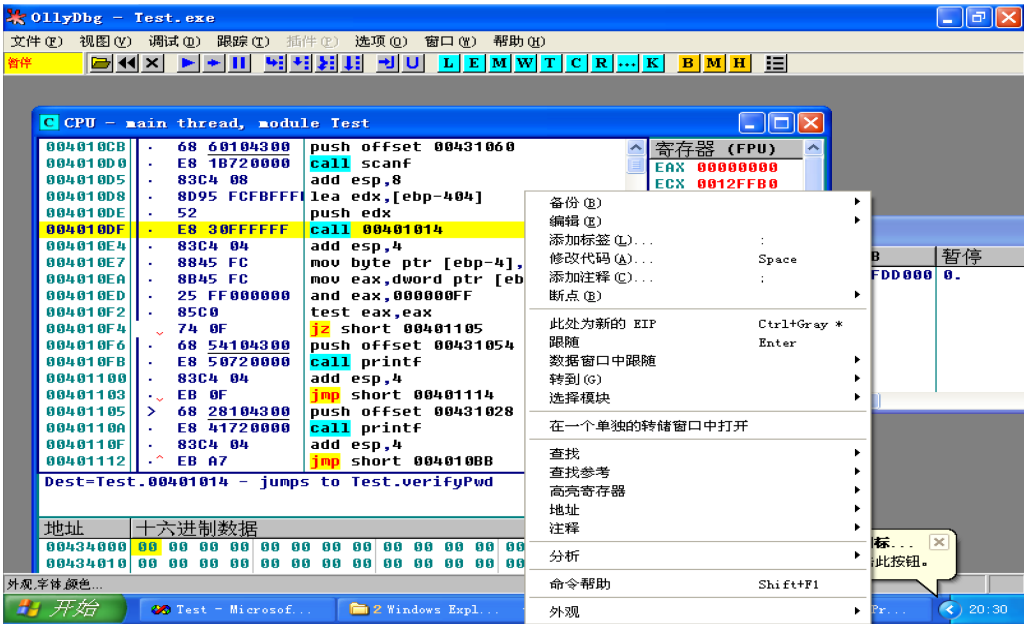
3. 通过上文汇编代码发现，该字符串在 004010F4 地址位置通过 jz 指令跳转并调用 printf 函数输出，则可以获得破解软件的第一种方式：更改跳转条件，将 jz 变为 jnz，从而在用户输入正确密码时，会输出错误提示“wrong password…”，而当用户输入错误密码时，会输出正确提示，从而破解软件。

将 jz 变为 jnz，同时编辑变化至文件，保存文件，运行如下



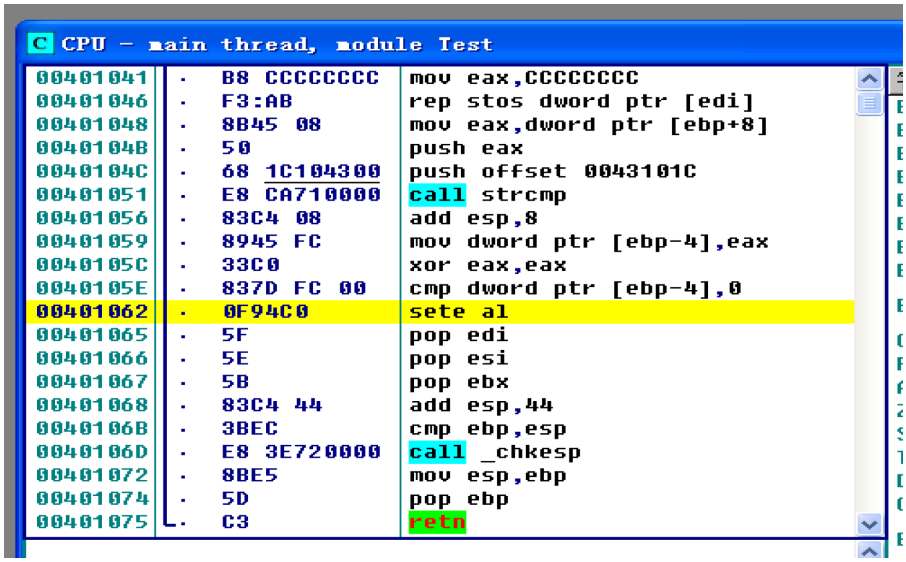
当输入正确密码，提示错误，而错误密码直接运行成功并关闭程序（没有 system（“pause”））

4. 第二种破解方法，对于汇编代码分析得知，程序输出正确/错误提示取决于 verifyPwd 函数的调用，故通过 ollyDBG 的跟随功能，进入函数内部分析代码。

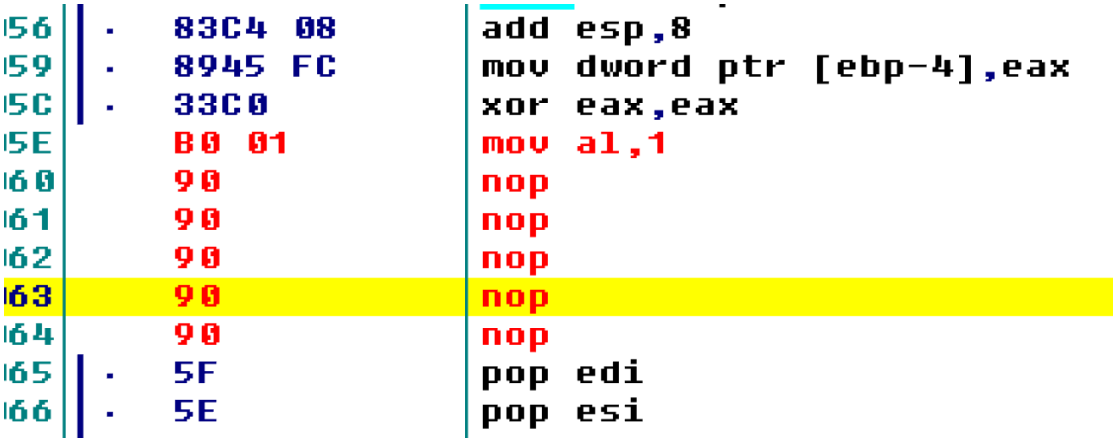


5. 在函数内部，可以看出在 0040105E 处有一个 cmp 指令，而后通过 sete 指令进行 al 位的设置，从而对后续 eax 寄存器的值产生改变，影响程序输出，从而可以知道，cmp 指令对于 al 设置这一步代码是破解的关键。

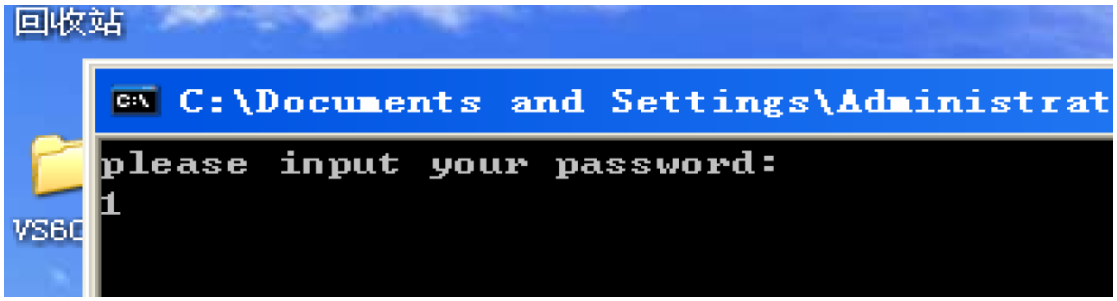
由于我们希望输入任何密码都可以成功运行软件，从而我们可以把 cmp 和 sete 指令直接变成一条简单的 mov al,01，从而直接将 al 设置为 01，而无视状态判定，值得指出的是，由于指令长度不同，这样的改变会需要我们使用 nop 空指令进行空位的填充。



6. 如图，进行代码的修改和指令的填充，同时保存文件。



7. 运行程序如下，当我们输入密码 1，文件正常运行并关闭，即破解成功



心得体会：

通过实验，掌握了使用 ollyDBG 软件进行 pe 文件的反汇编分析，通过程序文件的汇编指令推断其运行流程，并且运用了 ollyDBG 的查找、修改、跟随等功能，进行了简单 DEBUG 程序的破解，是逆向技术的入门。