《软件安全》实验报告

姓名： 冯思程 学号： 2112213 班级： 计算机科学与技术二班

**实验名称：**

OllyDBG软件破解实验

**实验要求：**

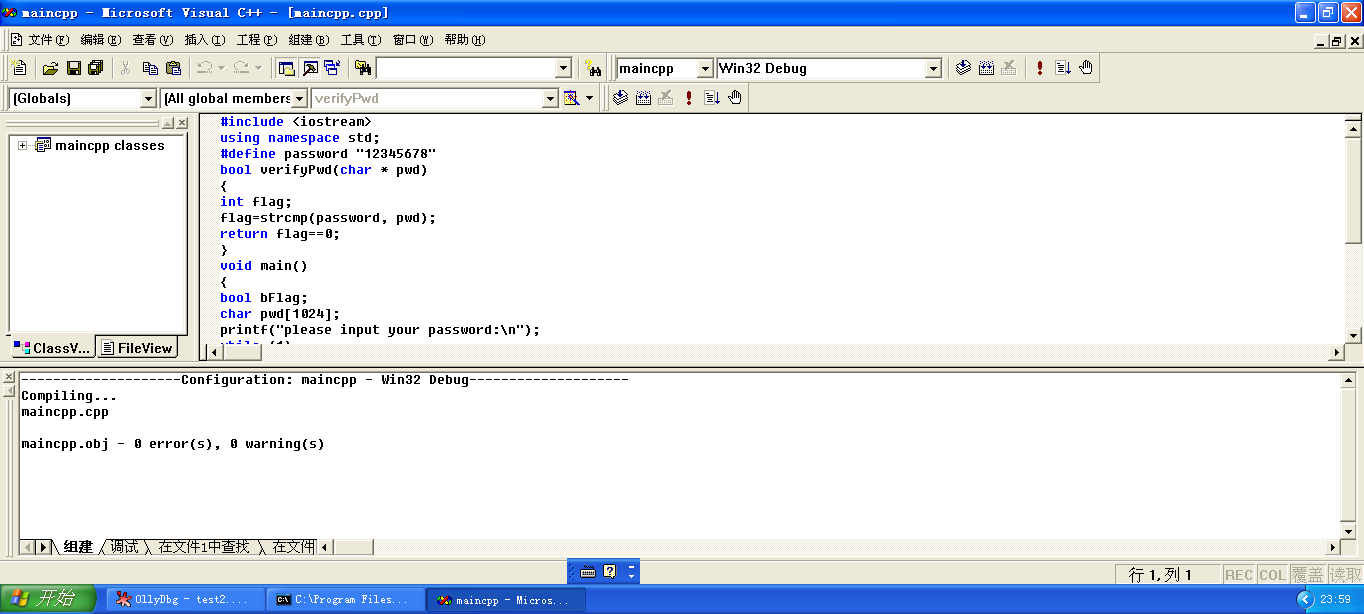
1. 请在XP VC6生成课本第三章软件破解的案例（DEBUG模式，示例3．1）。进而，使用0llyDBG进行单步调试，获取verifyPWD函数对应flag==0的汇编代码，并对这些汇

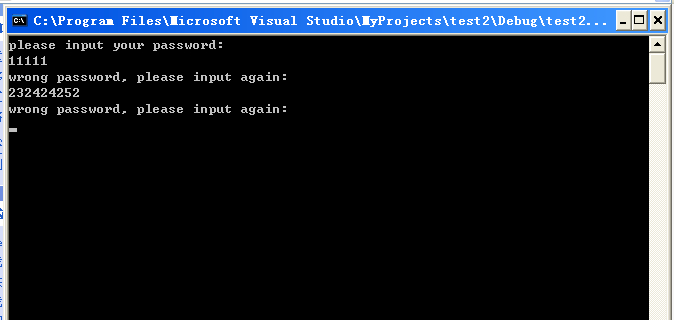
编代码进行解释。

1. 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：**

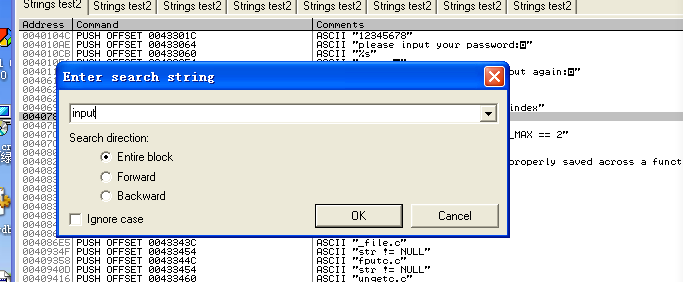
1. 在visual c++中进行编译，然后生成可执行文件（.exe），如下图。



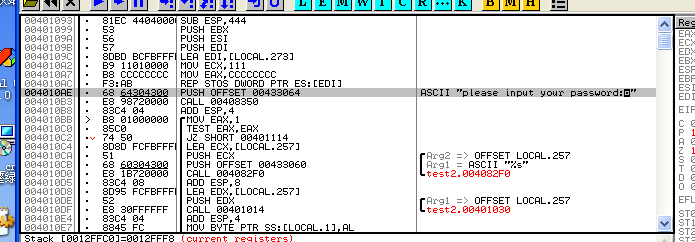


2.用ollydbg打开test2可执行文件，进行单步调试，获取verifyPWD函数对应flag==0的汇编代码。

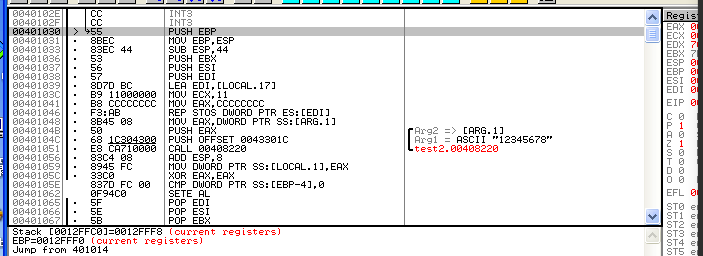
首先，打开test2可执行文件。在反汇编窗口，点右键，选择“查找 所有引用的字符串”功能。使用快捷键，Ctrl+F 打开搜索窗口查找“input”。



然后双击找到的有关“input”的字符串，定位到相关位置代码处。如下图



然后观察上下文汇编语句发现call语句调用，分别尝试几次观察哪个call对应falg==0.然后追踪到函数内，如下图。找到了目标函数的位置。



接下来对函数语句进行解释：

PUSH EBP

MOV EBP,ESP

SUB ESP,44

PUSH EBX

PUSH ESI

PUSH EDI

LEA EDI,[LOCAL.17]

MOV ECX,11

MOV EAX,CCCCCCCC

REP STOS DWORD PTR ES:[EDI]

上面这些代码是调整栈帧，同时进行新栈帧的初始化，通过11 次循环将新开辟的44h大小的空间都赋初值为0CCh

MOV EAX,DWORD PTR SS:[ARG.1]

PUSH EAX

PUSH OFFSET 0043301C

CALL 00408220

这部分代码是调用了strcmp函数，比较PWD和password是否一致（其中eax装的是输入的密码，0043301C地址存的是标准密码“12345678”。）

ADD ESP,8//恢复esp的原始位置

MOV DWORD PTR SS:[LOCAL.1],EAX//将函数调用后的返回值赋给局部变量flag

XOR EAX,EAX//eax清零

CMP DWORD PTR SS:[EBP-4],0//将flag的值和0比较

SETE AL //如果上一句是相等，则设置al为1

POP EDI

POP ESI

POP EBX

ADD ESP,44

CMP EBP,ESP

CALL 00408280

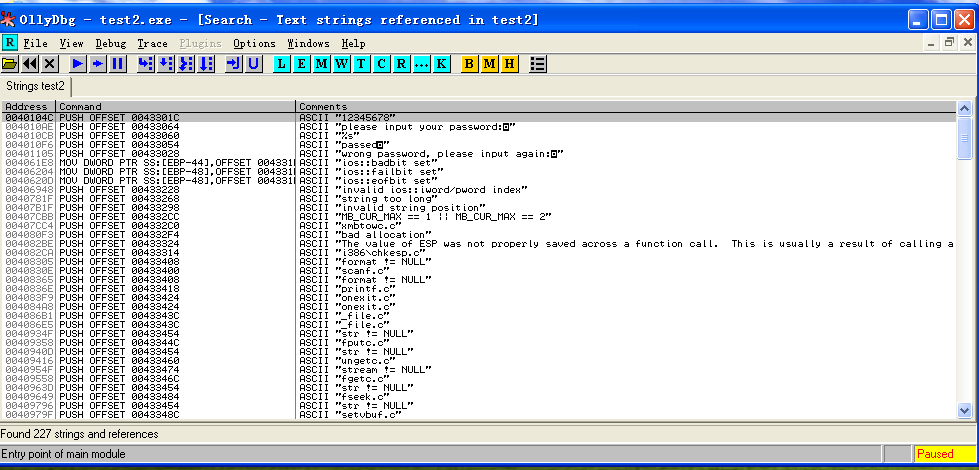
MOV ESP,EBP

POP EBP

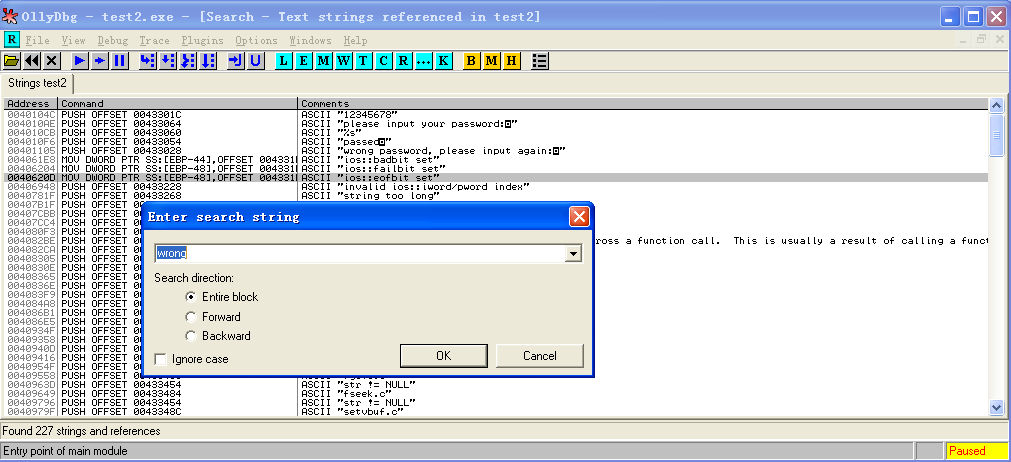
RETN

这部分是调整栈帧，结束函数调用。

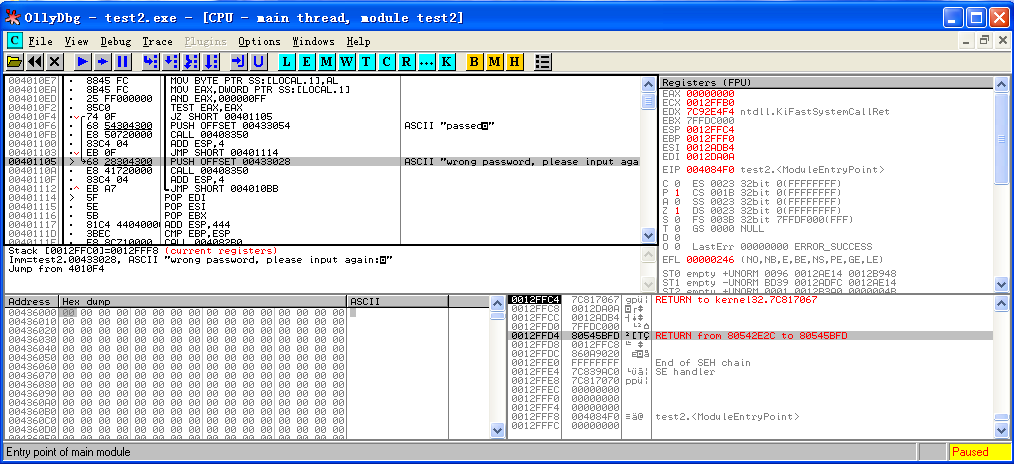
3. 用ollydbg打开test2可执行文件，为了尽快定位到分支语句处，在反汇编窗口，点右键，选择“查找 所有引用的字符串”功能。如下图。



然后，使用快捷键，Ctrl+F 打开搜索窗口，输入 wrong，点确定后，将定位出错信息的那一行代码：



双击这一行代码，就会定位反汇编中的相应代码处：



破解部分：

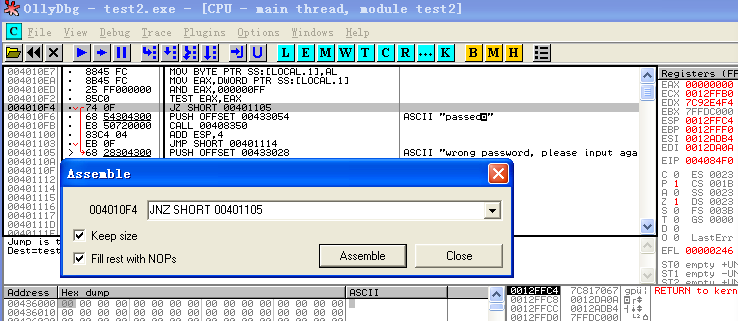
（1）破解方法一：通过判断逻辑来进行修改，观察反汇编语言（跳转语句与上文有连线），可知核心分支判断在于：

Test eax, eax

Jz short 00401105

如果 jz 条件成立，则跳转到 00401105 处，即显示错误密码分支语句中。如果将 jz 该指

令改为 jnz，则程序截然相反。输入了错误密码，将同样可以进入验证成功的分支中。满足破解的目的。双击 jz 密码一行，对其进行修改：（如下图）

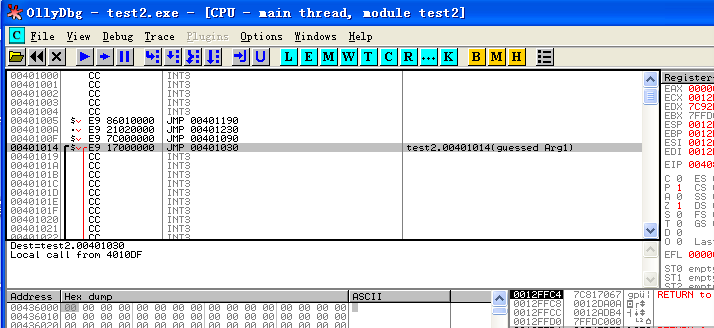


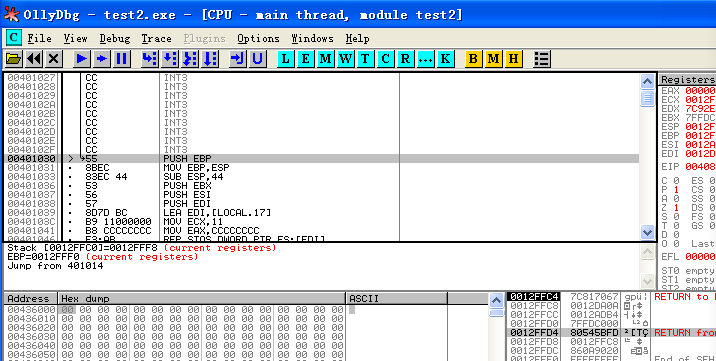
然后要在反汇编窗口，点右键，选择“编辑 复制当前修改到可执行文件”。保存后的可执

行文件，将是破解后的文件。经验证，可以成功修改程序。在我随便输入一个错误密码之后都可以通过进入到之后的核心程序部分。

补充说明：由于修改跳转逻辑时，正反同时修改，所以在输入正确的密码时反而会报错wrong password。

（2）破解方法二：通过更改函数来实现目的，通过分析汇编语句，可知，验证命令使用的是 verifyPwd 函数，点右键选择跟随，逐步进入该函数： 如下图：





函数的返回值通过 eax 寄存器来完成的，核心语句即 sete al。

对于函数中的代码：

flag=strcmp(password, pwd);

return flag==0;

被解释为汇编语言是：

Mov dword ptr [ebp-8], eax //将 strcmp 函数调用后的返回值（存在 eax 中）赋值给

函数的局部变量 flag

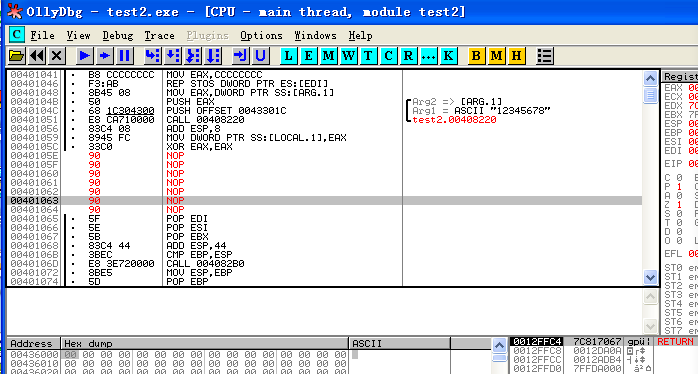
Xor eax, eax //将 eax 的值清零

Cmp dword ptr [ebp-8], 0 //将 flag 的值与 0 进行比较，即 flag==0;

Sete al //sete 是根据状态寄存器的值，如果相等，则设置，如果不等，则不设置

要想实现目的，需要在 cmp dword ptr [ebp-8], 0 处开始更改，将其更改为：mov al,01。强制al设置成01来实现目的，取消保持代码空间大小（如果新代码超长，将无法完成更改）

更改完发现剩余空间被空指令nop占用，同时sete al语句无存在意义，也将其改为空指令NOP。如下图：



然后同理上文要在反汇编窗口，点右键，选择“编辑 复制当前修改到可执行文件”。保存后的可执行文件，将是破解后的文件。经验证，可以成功修改程序。在我随便输入一个错误密码之后都可以通过进入到之后的核心程序部分。

**心得体会：**

1. 通过这次实验更加清晰和熟练的掌握了ollydbg进行单步调试的过程和一些方法，对于内在逻辑的理解更加透彻，
2. 对于实验一要求的函数栈帧的使用过程更加清晰明了。
3. 对于如何破解简单可执行程序有了自己的见解，掌握了几种简单的破解方法。有较大的收获。
4. 在半夜最后破解成功增强了软件安全方面的兴趣与成就感，更加喜欢学习软件安全的知识和方法。