### 软件破解示例

本节将对一个简单的密码验证程序，演示如何使用OllyDBG进行破解。

**实验二：对示例3-1源代码生成的Debug模式的可执行文件，使用OllyDBG进行破解。**

**示例3-1：**

|  |
| --- |
| #include <iostream>  using namespace std;  #define password "12345678"  bool verifyPwd(char \* pwd)  {  int flag;  flag=strcmp(password, pwd);  return flag==0;  }  void main()  {  bool bFlag;  char pwd[1024];  printf("please input your password:\n");  while (1)  {  scanf("%s",pwd);  bFlag=verifyPwd(pwd);  if (bFlag)  {  printf("passed\n");  break;  }else{  printf("wrong password, please input again:\n");  }  }  } |

破解对象是该程序生成的Debug模式的exe程序。注意：Debug模式和Release模式生成的可执行文件是不同的，采用了不同的编译和连接过程。Release模式生成的可执行文件不包含调试信息，代码更加精简、干练。

对得到的exe程序（假定不知道上面的源代码），有多种方式实现破解。比如，一种方式是使用OllyDBG，通过运行程序，观察关键信息，通过对关键信息定位，来得到关键分支语句，通过对该分支语句进行修改，达到破解的目的。另外一种方式，可以通过IDA Pro来观察代码结构，确定函数入口地址，对函数体返回值进行更改。

运行程序，输入一个密码，发现运行结果如下图3-32：

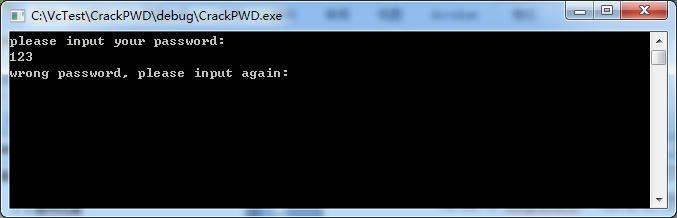


图3-32 程序运行结果

在OllyDBG中，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，如图3-33：

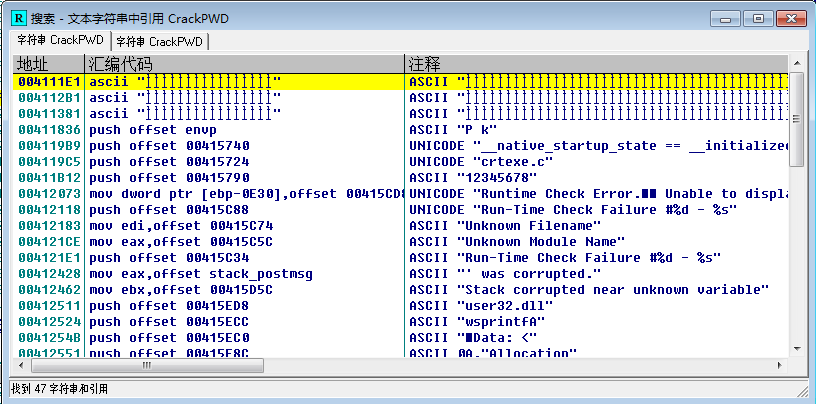


图3-33 查找所有引用的字符串

然后，使用快捷键，Ctrl+F打开搜索窗口，输入wrong，点确定后，将定位出错信息的哪一行代码，图3-34标黄处：

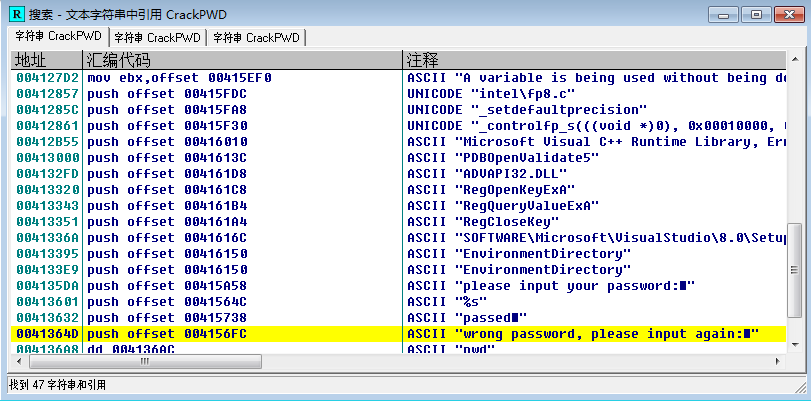


图3-34 找到出错信息的位置

双击这一行代码，就会定位反汇编中的相应代码处，如图3-35：

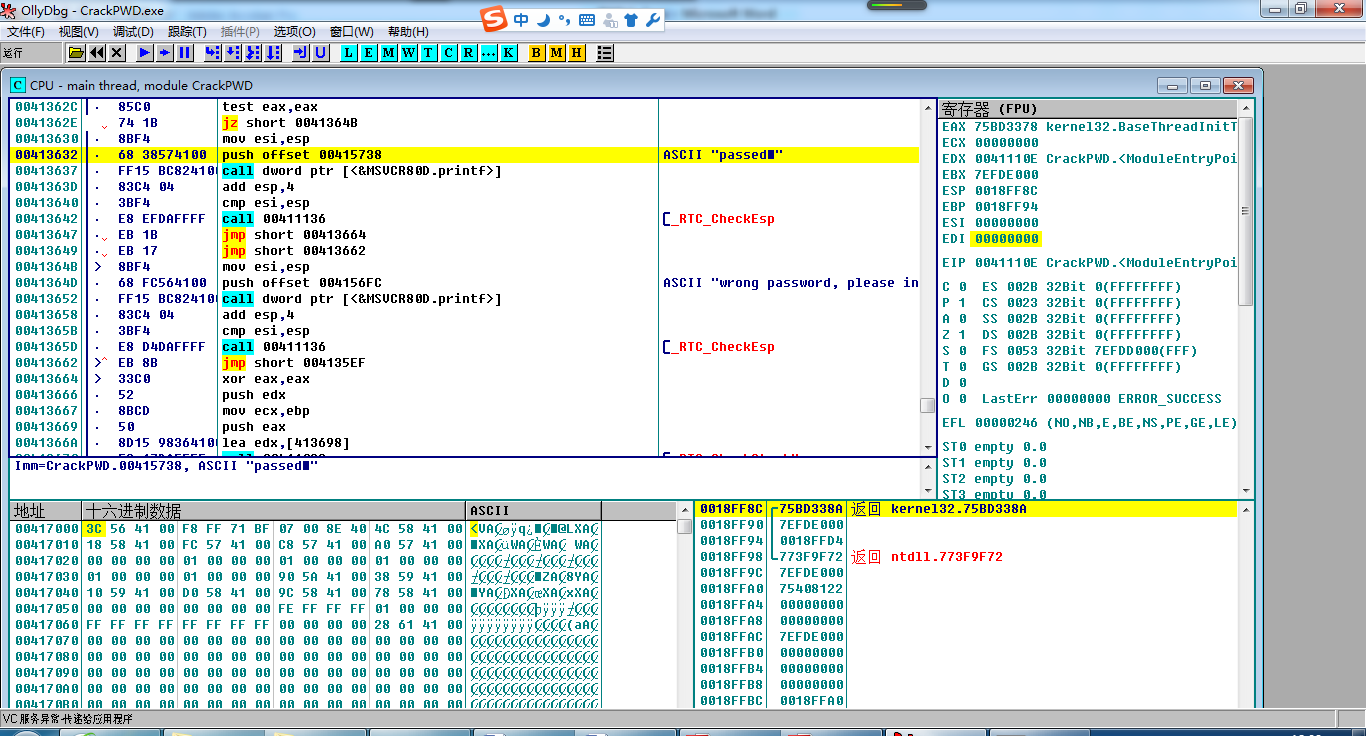


图3-35 定位到反汇编中的相应代码处

**1. 破解方式一**

观察反汇编语言，可知核心分支判断在于：

Test eax, eax

Jz short 0041364b

如果jz条件成立，则跳转到0041364b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，将进入验证成功的分支中。

双击jz密码一行，对其进行修改，如下图3-36：

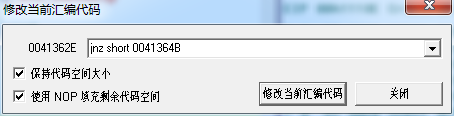


图3-36 修改当前汇编代码

点修改当前汇编代码即可。

注意：此时并没有真正修改二进制文件中的有关代码，如果想要修改二进制文件中的代码，需要在反汇编窗口，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。

**2. 破解方式二**

更改函数。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3-37：

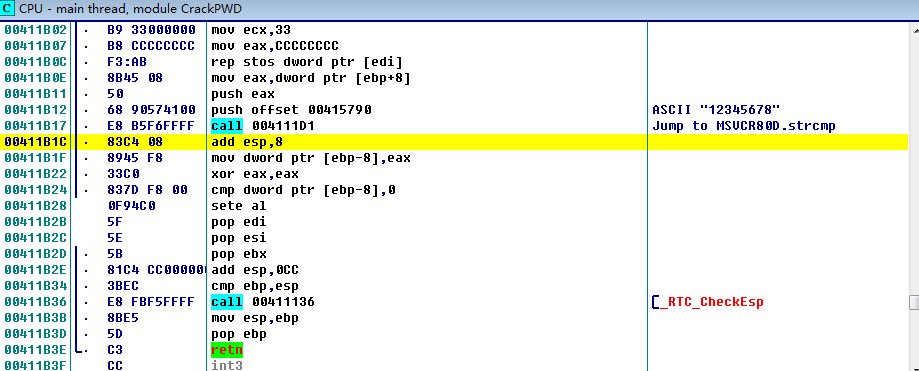


图3-37 逐步进入函数

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：  Mov dword ptr [ebp-8], eax //将strcmp函数调用后的返回值（存在eax中）赋值给变量flag  Xor eax, eax //将eax的值清空  Cmp dword ptr [ebp-8], 0 //将flag的值与0进行比较，即flag==0;  //注意cmp运算的结果只会影响一些状态寄存器的值  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 |

要想更改该语句，在cmp dword ptr [ebp-8], 0处开始更改，将其更改为：mov al,01。如图3-38所示。取消保持代码空间大小，如果新代码超长，将无法完成更改。

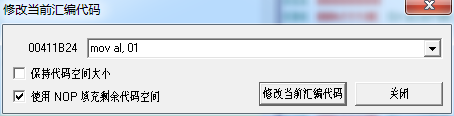


图3-38 修改当前汇编代码

并将sete al改为NOP。

得到结果如下图3-39：

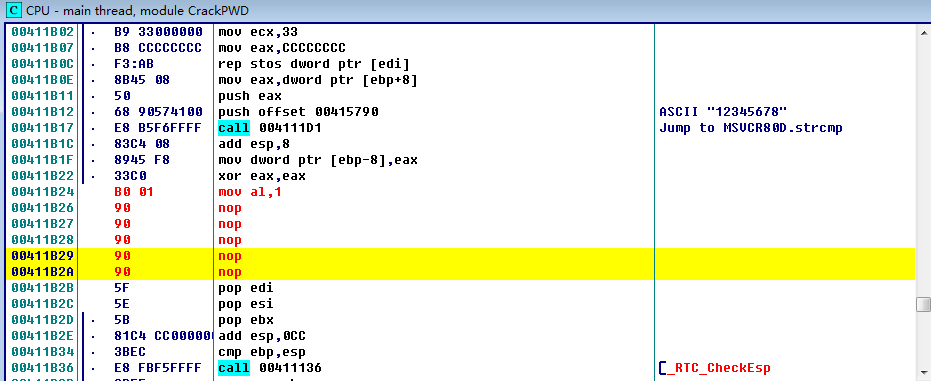


图3-39 修改后的汇编代码

此时，无论输入正确与否的密码，均将通过测试！