《漏洞利用及渗透测试基础》实验报告

姓名：姜志凯 学号：2011937 班级：信安班

**实验名称：**

Shellcode编写及编码

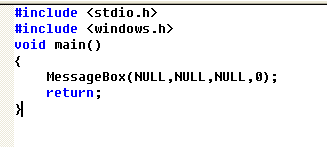
**实验要求：**

复现第五章实验三，并将产生的编码后的shellcode在示例5-4中进行验证，阐述shellcode编码的原理、shellcode提取的思想。

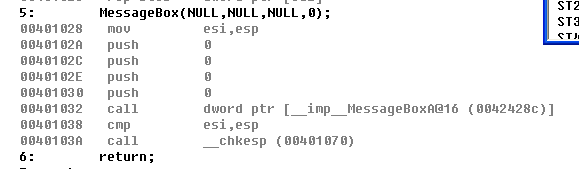
**实验过程：**

1. **Shellcode的编写与提取：**

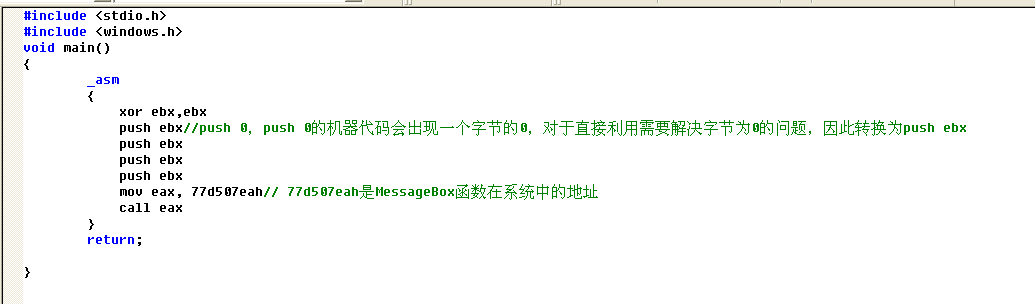
1、进入vc，用c语言书写要执行的Shellcode



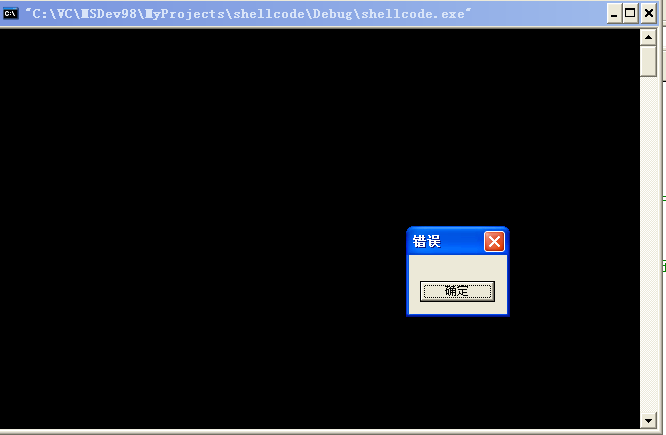
2、进入反汇编，换成对应的汇编代码



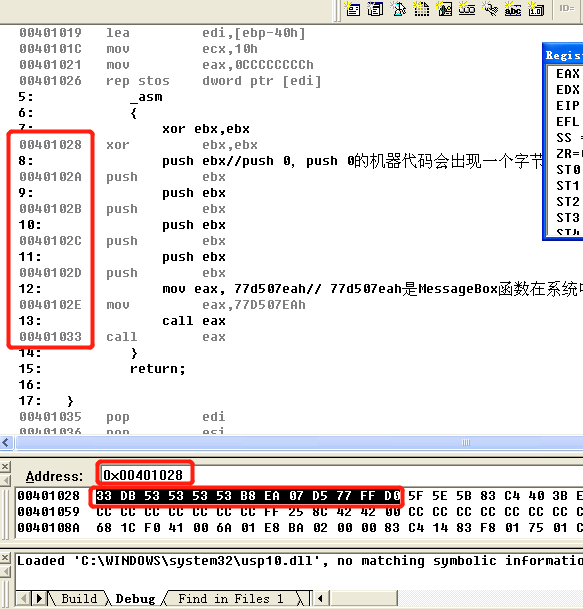
3、利用\_asm{}验证汇编代码的正确性



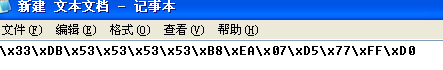
二者效果相同，如下

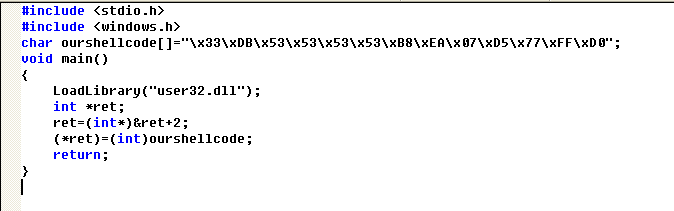


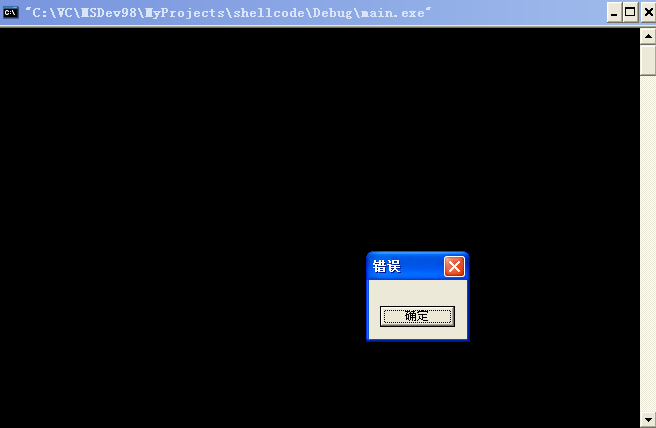
4、接下来将汇编代码转换为相应的机器码，进入反汇编，找到函数开始和结束的地址，中间的二进制代码，即为机器码



1. 将机器码保存下来，空格用\x替换，用另一个程序验证该机器码的正确性







至此，shellcode的编写与提取已经完成。

1. **Shellcode编码解码：**

由于字符集的差异、要绕过坏字符或绕过安全防护机制，常常要对shellcode进行编码，有编码就要有解码。这里介绍**异或编码解码**方法。

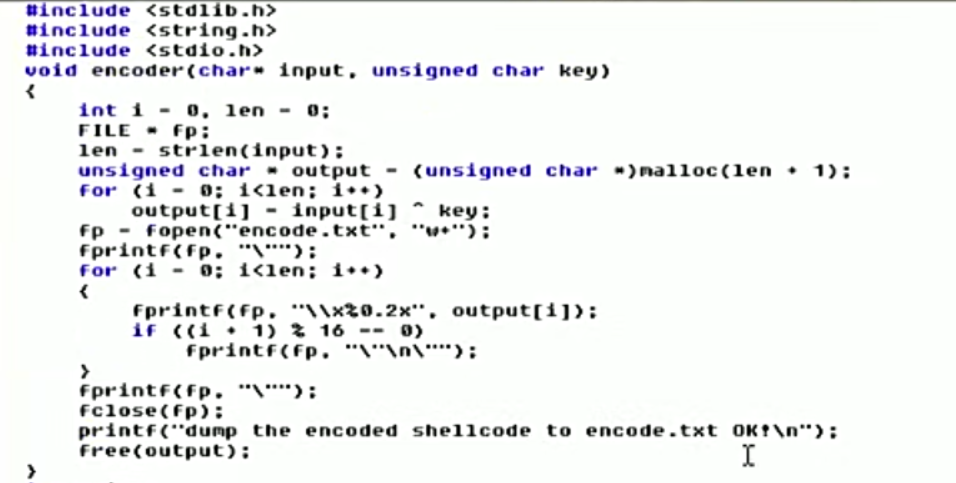
异或编码是一种简单易用的shellcode编码方法，它的编解码程序非常简单。

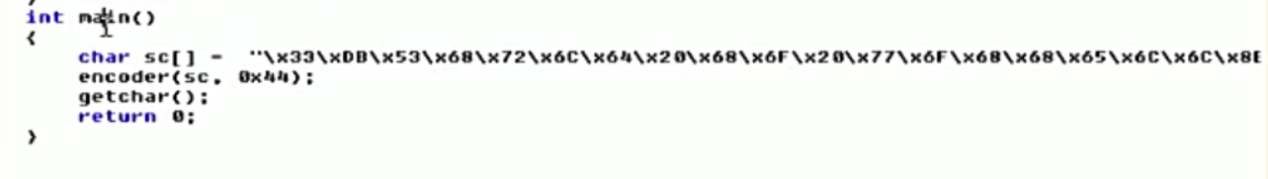
但是，它也存在很多限制，比如在选取编码字节时，不可与已有字节相同，否则会出现0。

编码程序，是独立的。是在生成shellcode的编码阶段使用。将shellcode代码输入后，输出异或后的shellcode编码。

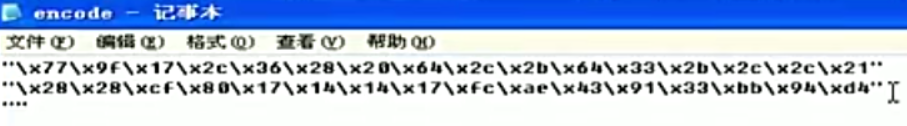
1. 编码：（编码程序与编入shellcode的代码是分离的，也就是对机器码单独编码）

对\x33\xDB\x53\x68\x72\x6C\x64\x20\x68\x6F\x20\x77\x6F\x68\x68\x65\x6C\x6C\x8B\xC4\x53\x50\x50\x53\xB8\xEA\x07\xD5\x77\xFF\xD0\x90（这里的\x90作为解码的结束符）进行编码，如下：





利用这个程序可以对机器码进行编码，密钥为0x44（与0x44异或），将结果存在encode.txt里，如下：

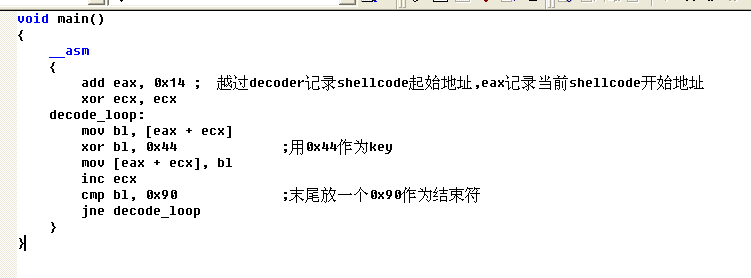


至此，完成编码。

1. 解码程序以及shellcode的编写：（将解码程序写进shellcode里）

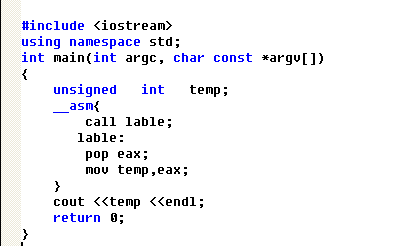
解码程序：

解码程序是shellcode的一部分。下面的解码程序中，默认EAX在shellcode开始时对准shellcode起始位置，程序将每次将shellcode的代码异或特定key（0x44）后重新覆盖原先shellcode的代码。末尾，放一个空指令0x90作为结束符。



将编码后的shellcode写到解码程序后边，**假设eax初始时指向解码程序开始的地方**，而解码程序的长度为0x14，所以eax加0x14，eax就指向了shellcode的起始位置，就可以运行解码程序对编码进行解码了，**循环对编码进行异或0x44进行解码**（碰到90，解码结束）。

但怎么**让eax指向当前指令**的地址呢？如下：

（C与汇编混合）

Call会执行push EIP；

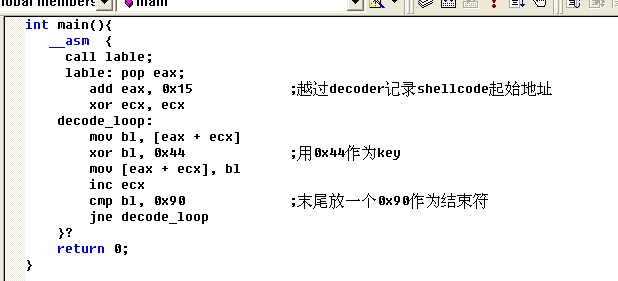
EIP的值又是下一条指令pop EAX的地址；

Pop Eax会将栈顶EIP（自身指令地址）出栈，保存到EAX中

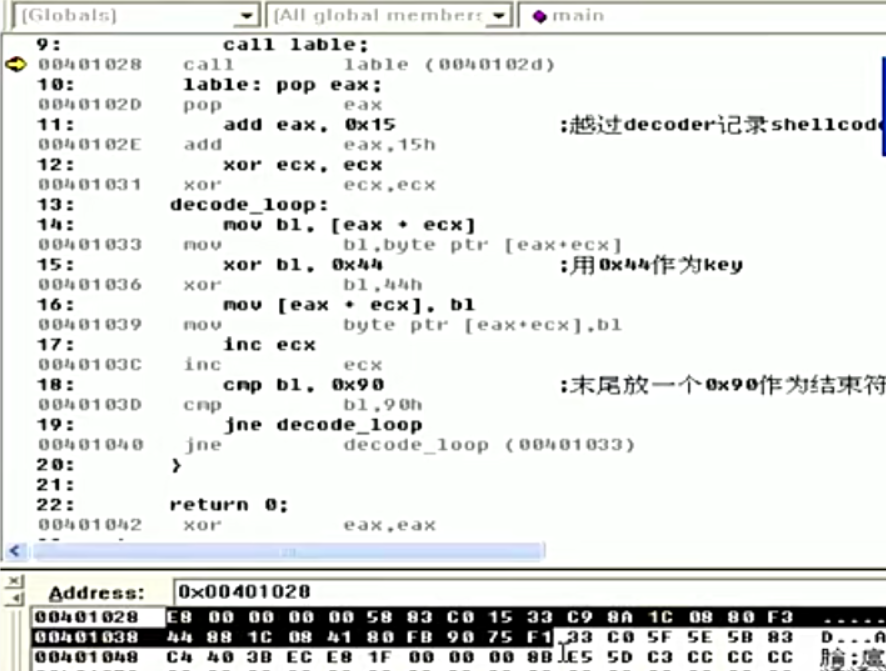


这样就实现了**eax指向当前位置**。

**将二者合并**，得到如下代码：



然后用之前实验用的shellcode编写提取的方法得到解码程序的**机器码**。

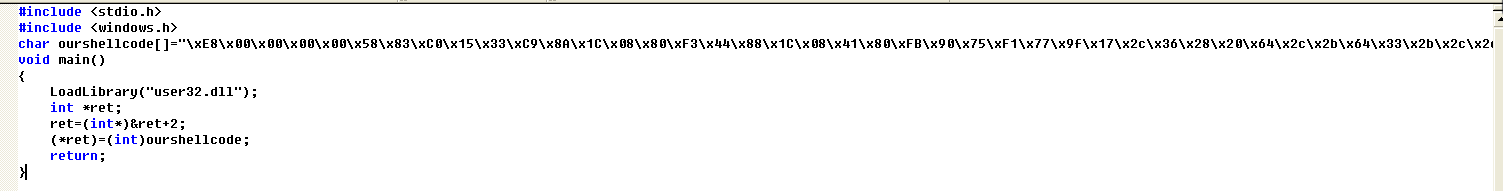


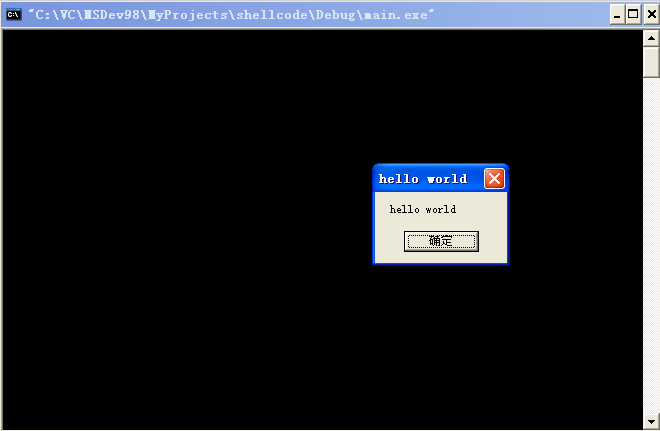
解码程序的机器码：\xE8\x00\x00\x00\x00\x58\x83\xC0\x15\x33\xC9\x8A\x1C\x08\x80\xF3\x44\x88\x1C\x08\x41\x80\xFB\x90\x75\xF1

后边连上编码shellcode即可组成最终的shellcode：

\xE8\x00\x00\x00\x00\x58\x83\xC0\x15\x33\xC9\x8A\x1C\x08\x80\xF3\x44\x88\x1C\x08\x41\x80\xFB\x90\x75\xF1\x77\x9f\x17\x2c\x36\x28\x20\x64\x2c\x2b\x64\x33\x2b\x2c\x2c\x21\x28\x28\xcf\x80\x17\x14\x14\x17\xfc\xae\x43\x91\x33\xbb\x94\xd4

用程序框架验证shellcode的正确性：





**心得体会：**

**Shellcode编码原理：**

首先要对shellcode原码进行编码，这里用异或编码，及循环对原码的每个字节和设定的密钥做异或操作，得到新的序列，为编码后的shellcode；

解码：

默认EAX在shellcode开始时对准shellcode起始位置，程序将每次将shellcode的代码异或特定key（0x44）后重新覆盖原先shellcode的代码。

总结：

在解码程序的机器码后边，加上通过异或编码后的shellcode代码，组成新的shellcode，可以进行漏洞利用。（注意，这里的shellcode必须以0x90结尾且中间不出现0x90，判断解码是否结束）

**Shellcode提取思想：**

先用**高级语言**（如C++）进行shellcode代码的编辑，调试后进入**反汇编模式**，找到实现相应功能的代码区域，看它的**首末地址**，在地址框内**搜索**，找到完整的**机器码**。

此外，通过本实验，掌握了多个汇编语言的用法。