

《漏洞利用及渗透测试基础》实验报告

姓名：齐明杰 学号：2113997 班级：信安2班

实验名称：

Ollydbg 软件破解

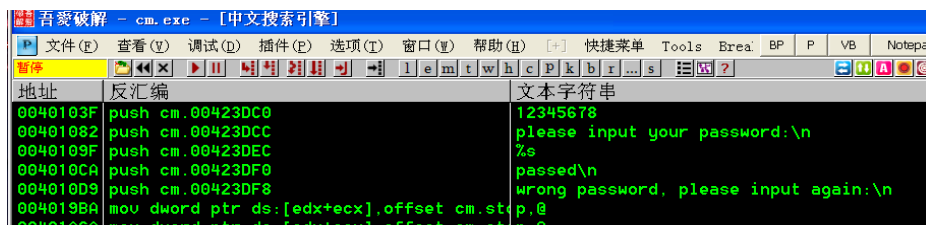
实验要求：

实验要求：

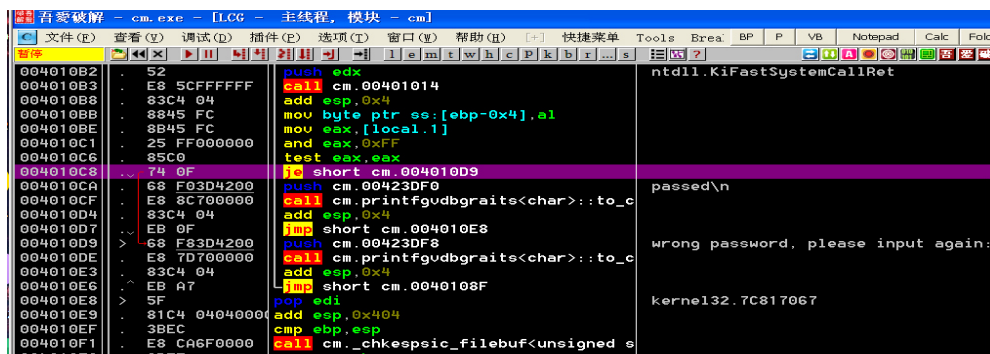
1. 请在 XPVC6 生成课本第三章软件破解的案例(DEBUG 模式，示例 3-1)。进而，使用 ollyDBG 进行单步调试，获取 verifyPWD 函数对应 flag==0 的汇编代码，并对这些汇编代码进行解释。
2. 对生成的 DEBUG 程序进行破解复现课本上提供的两种破解方法、

实验过程：

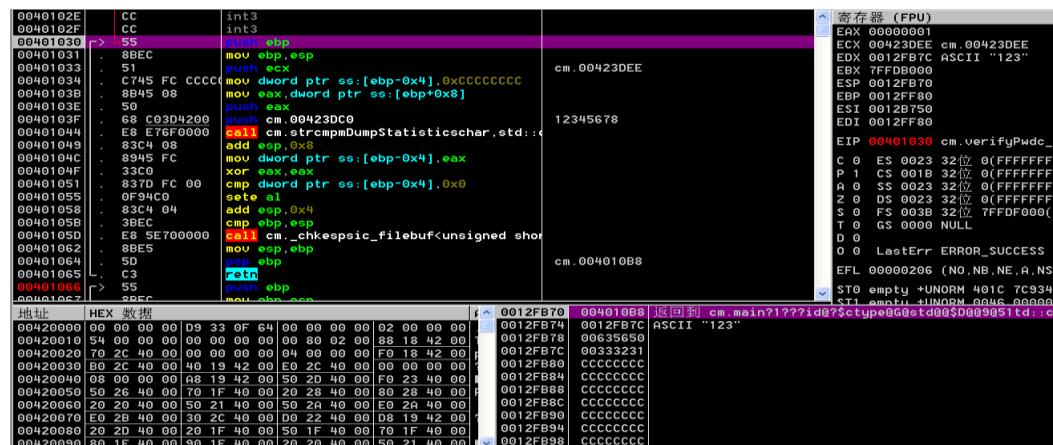
用 ollydbg 运行编译出的 exe 文件，搜索文本字符串：



双击进入到引用 'wrong password' 的代码处，如下图：

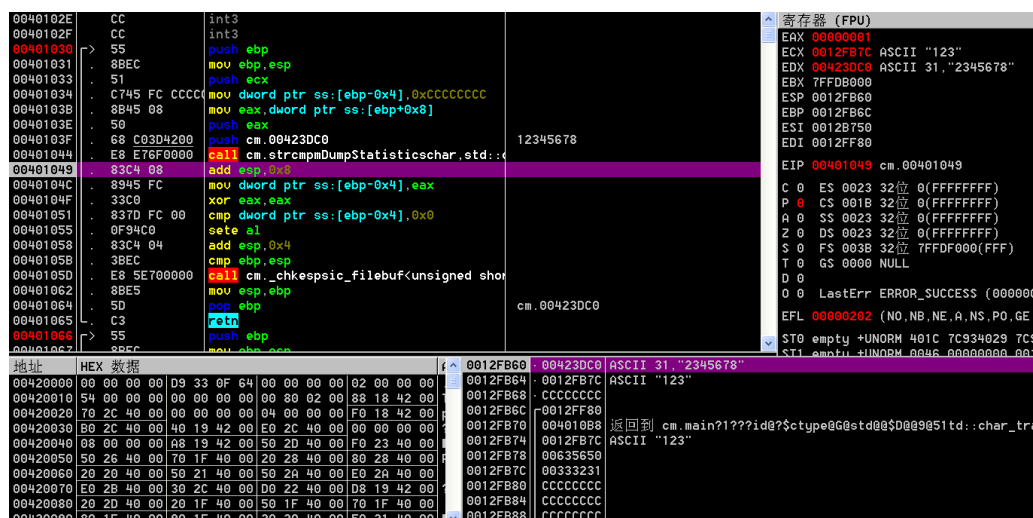


向上发现 004010B3 call cm.00401014，即调用 verifyPwd 函数，随便输入一个口令 123，然后 F7 步入，栈和寄存器等状态如下图：



在栈中我们可以看到输入的口令'123'，以及返回地址 004010B8。

之后调用了 strcmp 函数来比较输入的口令和正确口令，比较后如下图所示：

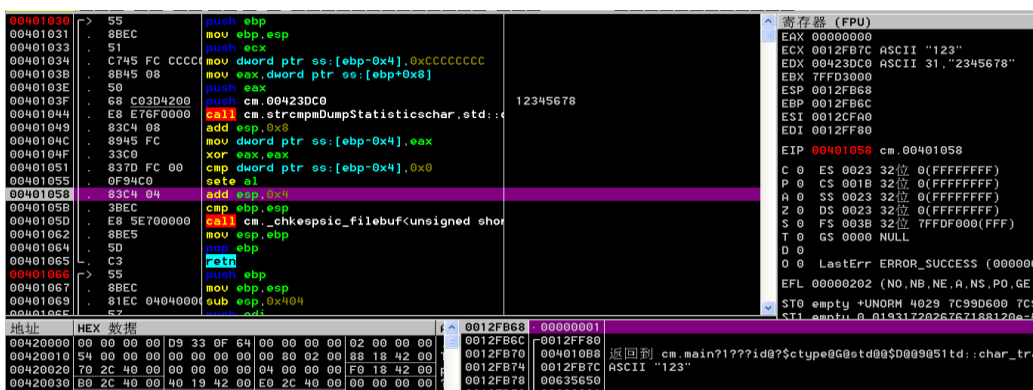


EAX 被赋值为 1，说明 strcmp 返回不相等（即输入错误口令）。

之后把 eax 存入局部变量（即 flag）后清零 eax，然后通过判断 flag 值是否为 0 来决定是否设置 al 为 1，如果值为 0 则设置 al 为 1。这便是 flag == 0 语句的汇编实现，如下图：

00401051	83D FC 00	cmp dword ptr ss:[ebp-0x4],0x0
00401055	0F94C0	sete al

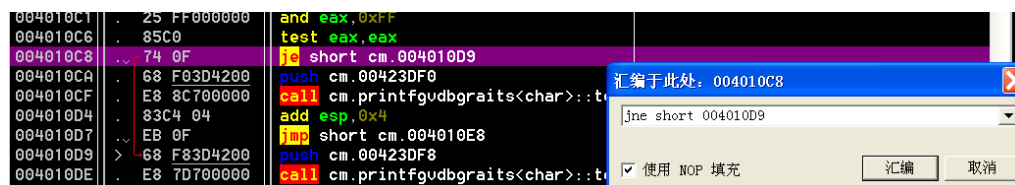
运行 sete al 后，最终 eax 结果为 0，如下图：



第一种破解方式：修改关键跳转

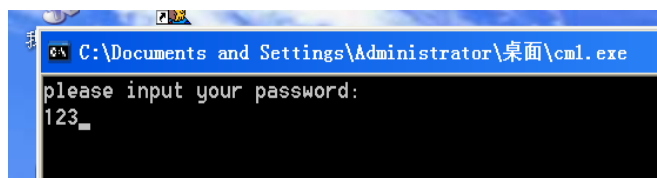
可以看出，关键跳是 004010C8 je short cm.004010C9，若跳转则提示错误信息。

因此，我们修改指令为 jne short cm.004010C9，即逻辑相反，输入错误则提示正确，但输入正确会提示错误。



004010C1	25 FF000000	and eax,0xFF	
004010C6	85C0	test eax,eax	
004010C8	75 0F	jmp short cm.004010D9	
004010CA	68 F03D4200	push cm.00423DF0	passed\n
004010CF	E8 8C700000	call cm.printfugdbgraits<char>::to_c	
004010D4	83C4 04	add esp,0x4	
004010D7	EB 0F	jmp short cm.004010E8	
004010D9	68 F83D4200	push cm.00423DF8	wrong password, please input again
004010DE	E8 7D700000	call cm.printfugdbgraits<char>::to_c	
004010E3	83C4 04	add esp,0x4	

然后右键->复制到可执行文件->所有修改，将修改保存到文件中，命名为 cm1.exe
打开 cm1.exe 运行，发现输入错误口令能通过：



第二种破解方式：修改函数内的返回值。

单步跟入函数内，运行至 stete al 语句：

00401030	> 55	push ebp	
00401031	8BEC	mov ebp,esp	
00401033	51	push ecx	
00401034	C745 FC CCCC	mov dword ptr ss:[ebp-0x4],0xFFFFFFFF	
0040103B	8B45 08	mov eax,dword ptr ss:[ebp+0x8]	
0040103E	50	push eax	
0040103F	68 C03D4200	push cm.00423DC0	12345678
00401044	E8 E76F0000	call cm.strempmDumpStatisticschar,std::c	
00401049	83C4 08	add esp,0x8	
0040104C	8945 FC	mov dword ptr ss:[ebp-0x4],eax	
0040104F	33C0	xor eax,eax	
00401051	837D FC 00	cmp dword ptr ss:[ebp-0x4],0x0	
00401055	0F94C0	sete al	
00401058	83C4 04	add esp,0x4	
0040105B	3BEC	cmp ebp,esp	
0040105D	E8 5E700000	call cm._chkesplic_filebuf<unsigned shor	
00401062	8BE5	mov esp,ebp	
00401064	5D	pop ebp	
00401065	C3	ret	
00401066	> 55	push ebp	
00401067	8BEC	mov ebp,esp	

此处根据标志位来设置 al 的值，即返回值，因此我们可以直接修改这部分的语句，达到修改返回值的目的：

00401030	> 55	push ebp	
00401031	8BEC	mov ebp,esp	
00401033	51	push ecx	
00401034	C745 FC CCCC	mov dword ptr ss:[ebp-0x4],0xFFFFFFFF	
0040103B	8B45 08	mov eax,dword ptr ss:[ebp+0x8]	
0040103E	50	push eax	
0040103F	68 C03D4200	push cm.00423DC0	12345678
00401044	E8 E76F0000	call cm.strempmDumpStatisticschar,std::c	
00401049	83C4 08	add esp,0x8	
0040104C	8945 FC	mov dword ptr ss:[ebp-0x4],eax	
0040104F	33C0	xor eax,eax	
00401051	80 01	mov al,0x1	
00401053	90	nop	
00401054	90	nop	
00401055	90	nop	
00401056	90	nop	
00401057	90	nop	
00401058	83C4 04	add esp,0x4	
0040105B	3BEC	cmp ebp,esp	
0040105D	E8 5E700000	call cm._chkesplic_filebuf<unsigned shor	
00401062	8BE5	mov esp,ebp	

同样复制到可执行文件中，运行后发现无论输入错误还是正确的口令，均能通过。至此完成修改。

心得体会：

通过实验，掌握了 ollydbg 的基本用法
同时学会了通过修改指令来实现破解
此外，通过本次实验巩固了汇编代码的知识。