一、汇编（参考网络资料《汇编指令机器码对应表》）

普通指令

mov：传送字或字节

mov eax,1 -- 将1存入寄存器eax中

数据寄存器, eax 32bit, 其中 ax 为 eax 低十六位

var\_14是-14h EC+14=100

var\_8是-8h F8+8=100

8B 45 EC

MOV reg16, reg16/mem16

mov eax, [ebp+var\_14]

66 8B 40 04

MOV reg16, reg16/mem16

mov ax, [eax+4]

89 45 F8

MOV reg16/mem16, reg16

mov [ebp+var\_8], eax

66 89 45 E8 66为操作数长度覆盖，用在16/32位切换。比如mov ax，1234h 66 B8 34 12

mov [ebp+var\_18], ax

C7 45 F4（-0Ch）3B 00 00 00

MOV reg16/mem16, immed16

mov [ebp+var\_C], 3Bh

0F BF 45 E8

movsx eax, [ebp+var\_18]

C6

MOV reg8/mem8, immed8

C6 45 E8 00 00

Mov [ebp+var\_18] 00h

E9 5D 00 00 00

48C EF3 -> 48C F50 = 00 00 00 5D

4 8CEF3 -> 4 E9B40 = 00 05 CC 4D

8C2EE:

E9 4D CC 05 00

E8F40:

66 C6 45 E8 00 00

E9 05 34 FA FF

4 E9B4B -> 4 8CF50 = FF FA 34 05

E8F40-2:

mov eax, [ebp+var\_14]

mov [ebp+var\_1C], eax

**mov eax, 0**

**jmp loc\_48CF5A**

8B 45 EC

89 45 E4

**B8 00 00 00 00**(~~0F BF 45 E8)~~

~~(85 C0)~~

~~(0F 85 26 00 00 00)~~

**E9 0A 34 FA FF**

末尾E8F50 + 400C00=4E9B50

4 E9B50 ->4 8CF5A = FFFA340A

二、exe分析

**loc\_48CEE7（IDA）** 对应8C2E7（WinHex），差值为400C00。

使用stud\_PE查看exe的.text区段物理偏移（WinHex偏移）为400-E9000（E8F40-E8FFF为00空值）。

考虑差值，IDA空值区域为（4E9B 40- 4E9B FF）

三、反编译分析

招募逻辑位于sub\_48CE90中。

其中最关键的是loc\_48CEE7，大致逻辑可能是：通过判断是何种生物，将生物名字赋给相应的变量，其中case0就是ghost。

后面紧接的逻辑def\_48CF35进行了2个变量的处理，接着test判断可招募数量是否为0，如果不是0则jnz到招募程序，如果是0则跳到无法招募else的程序。

修改思路1：loc\_48CEE7末尾直接跳到else的程序，结果会死机，推测逻辑def\_48CF35进行的2个变量处理是必须的。

修改思路2：loc\_48CEE7末尾跳到空白位置（48CEF3 -> 4E9B40，exe对应E8F40），新位置复述2个变量的处理，将eax置0（让程序认为可招募数量为0）再跳回test判断处（4E9B50 ->48CF5A）。修改成功

48CEEE（8C2EE）：E9 5D 00 00 00 改为 E9 4D CC 05 00

4E9B40（E8F40）：

mov eax, [ebp+var\_14]

mov [ebp+var\_1C], eax

**mov** eax, 0

**jmp** loc\_48CF5A

8B 45 EC

89 45 E4

**B8** 00 00 00 00

**E9** 0A 34 FA FF

（4E9B50 ->48CF5A jmp往回跳的偏移算法）

