

## 汇编语言与逆向工程第一次作业：对 Cpp1.exe 逆向分析

```
rep stosd
push offset aPlaseGiveMeYou ; "Plase give me your answer:\n"
call _printf
add esp, 4
lea eax, [ebp+Str]
push eax
push offset Format ; "%s"
call _scanf
add esp, 8
lea ecx, [ebp+Str]
push ecx ; Str
call _strlen
add esp, 4
mov [ebp+var_18], eax
cmp [ebp+var_18], 8
jz short loc_40107C
```

在找到“Plase give me your answer:”后定位到函数，

函数 scanf 有两个参数，将输入的字符串存入 Str；

然后取 Str 的地址到 ecx

函数 strlen 得到 Str 的长度在 eax

比较，若长度是 8 位，跳绿色的线

然后看跳到右面后：

```
loc_40107C:
mov [ebp+var_4], 0
jmp short loc_40108E

loc_40108E:
mov eax, [ebp+var_4]
cmp eax, [ebp+var_18]
jge short loc_401088
```

var\_4 是初始化的 0；下面用这个 var\_4 和 8 比较，大于等于 8 跳到右面，所以要先向左跳

```
mov ecx, [ebp+var_4]
movsx edx, [ebp+ecx+Str]
and edx, 7Eh
mov eax, [ebp+var_4]
movsx ecx, [ebp+eax+Str]
and ecx, 80h
sar ecx, 7
or edx, ecx
mov eax, [ebp+var_4]
movsx ecx, [ebp+eax+Str]
and ecx, 1
shl ecx, 7
or edx, ecx
mov eax, [ebp+var_4]
mov [ebp+eax+Str], dl
mov ecx, [ebp+var_4]
movsx edx, [ebp+ecx+Str]
mov eax, [ebp+var_4]
movsx ecx, byte_429A38[eax]
xor edx, ecx
mov eax, [ebp+var_4]
mov [ebp+eax+Str], dl
jmp short loc_401085

loc_401085:
mov edx, [ebp+var_4]
add edx, 1
mov [ebp+var_4], edx
```

左边: ecx 存那个 var\_4

[ebp+ecx+Str]所以就是 Str[var\_4]的值

and, 和 01111110 与运算, 取八位二进制数的中间 6 位到 edx

and, 和 10000000 与运算, 取八位二进制数的最高位到 ecx

然后 ecx 右移 7 位

再把 ecx 和 edx 组合到一块到 edx

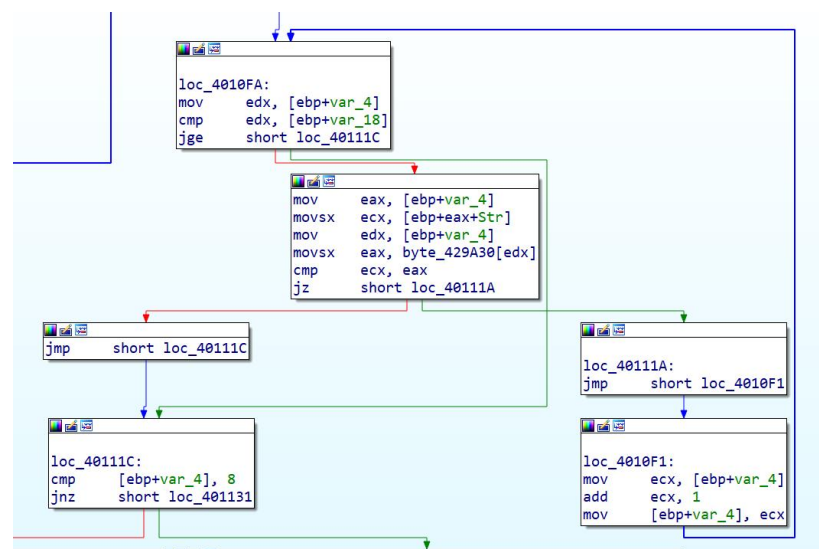
再把最低位移到最高位

然后取 byte\_429A38[eax]到 ecx

用 ecx 对 edx 异或加密

再循环

总的来说, 就是每一位密码要把 8 位 2 进制的最高位移到最低位, 最低位移到最高位; 然后和 byte\_429A38[eax]异或加密, byte\_429A38[eax]中为{6, 7, 8, 9, 10, 11, 12, 13}



这里又是一个 for 循环, 循环数是 var\_4, 从 0 到 8, 和 byte\_429A30[var\_4]里面的数比较而 byte\_429A30[var\_4]分别是{52h,C7h,C2h,CDh,EEh,EBh,FEh,F5h}

知道了 52h,C7h,C2h,CDh,EEh,EBh,FEh,F5h 密文再反推明文。

把这 8 个数分别和 6, 7, 8, 9, 10, 11, 12, 13 异或, 再把最低位和最高位调换就能得到明文

**C++代码:**

```
Main.cpp
任写 (全局范围)
1  #include<iostream>
2  using namespace std;
3
4  int main() {
5
6      int h = 0;
7      int b = 0;
8      int l = 0;
9
10     int array[8] = {0X52,0X0C7,0X0C2,0X0CD,0X0EE,0X0EB,0X0FE,0X0F5};
11     for (int i = 0; i < 8; i++)
12     {
13         array[i] = array[i] ^ (i + 6);
14         h = array[i] & 0X80;
15         b = array[i] & 0X7E;
16         l = array[i] & 0X01;
17         l=l << 7;
18         h = h >> 7;
19         array[i] = l | b | h;
20     }
21
22     for (int i = 0; i < 8; i++)
23     {
24         cout << char(array[i]);
25     }
26
27     return 0;
28 }
29
```

运行结果:

```
Microsoft Visual Studio 调试 任写
TAKEeasy
C:\repos\任写\x64\Debug\任写.exe (进程 39940)已退出, 代码为 0。
按任意键关闭此窗口. . . |
```

测试:

```
C:\Users\wangm\WPSDrive\15
Plase give me your answer:
TAKEeasy
Congratulations! You are right!
请按任意键继续. . . |
```

结果正确!