

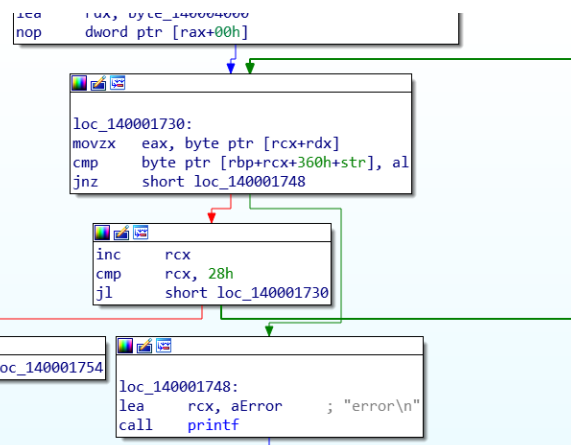
HGAME Week4 WriteUp

RE

easyVM

1. 通过字符串的查找 通过 error 找到对输入字符的判定 且可以判定字符串长度应为 40

```
assume cs:_data
;org 140004000h
00040000[48]
db 3Ah, 54h, 2Fh, 2Ah, 2Fh, 36h, 13h, 1, 2Eh, 3, 35h, 40h
; DATA XREF: sub_140001180+5A5↑to
db 47h, 0Eh, 5Fh, 59h, 1, 69h, 27h, 8, 3Dh, 4Ch, 33h, 1Ah
db 2Dh, 0Bh, 40h, 0Eh, 4Bh, 24h, 41h, 27h, 25h, 28h, 29h
db 2Ah, 2 dup(2), 5Dh, 24h, 8 dup(0)
dq 0FFFFD466D2205DCdh ; DATA XREF: __report_gsfailure+B4↑to
```



2. 往上追溯发现有对字符串的加密，是通过对内存里的指令进行的，再与内存中的数值进行比较

3. 带入到 x64dbg 中进行动态调式发现当指令执行到 0B 时回进行

B	0000000000000000C	
D	0501080501020500	
B	060303040C050E02	
D	0B05080103070A05	
B	0000000000000000D	
D	6F6F6F6F6F6F6F6F	
B	000000000000006F6F	
D	0000000000000000	
B	0000000000000000	

循环返回至 0E，循环里的操作总结起来就是对字符串逐位进行 xor 运算，在内存中可以找到运算的对象，太长就不截出来了。

4. 把数据写进脚本就完事了

```
#include <stdio.h>
int main(void)
{
    int kk[40] = {0x3A,0x54,0x2F,0x2A,0x2F,0x36,0x13,1,0x2E,3,0x35,0x40,
0x47,0x0E,0x5F,0x59,1,0x69,0x27,8,0x3D,0x4C,0x33,0x1A,0x2D,0x0B,0x40,0x0E,0x4B,
0x24,0x41,0x27,0x25,0x28,0x29,0x2A,2,2,0x5D,0x24};
    int mm[40] = {0x52,0x33,0x4E,0x47,0x4A,0x4D,0x67,0x69,0x47,0x70,0x6A,0x36,0x2A,
0x51,0x36,0x2A,0x5E,0x36,0x54,0x67,0x4E,0x23,0x40,0x75,0x5E,0x64,0x33,0x61,0x38,
0x4B,0x32,0x48,0x56,0x47,0x76,0x4F,0x63,0x71,0x24,0x59};
    for(int i=0;i<40;i++)
    {
        printf("%c", (mm[i] ^ kk[i]) % 256);
    }
}
```

```
选择C:\Windows\system32\cmd.exe
hgame{this_vm_is__sososososososo_easy}
请按任意键继续. . .
```