alading writeup —— 代码动态变化

原创

R00cky 于 2015-05-02 18:03:38 发布
分类专栏： writeup 文章标签： 逆向 ctf writeup

Hint：

暂无HINT

题目描述：

阿拉灯的神丁 (ノ ` □')ノ ⌒┻━┻

Writeup：

IDA分析流程得：

0x0041204C处的256字节数据与输入的8字节key进行循环异或，结果覆盖0x0041204C处原始数据。

把0x00401000作为基址，0x0041204c处的数据依次作为偏移地址，把0x0041214c处的数据依次作为原始数据写入前面构造的地址中。

```
mov     ecx, [ebp+var_10]
xor     edx, edx
mov     dl, byte_41204C[ecx]
mov     eax, offset loc_401000
mov     ecx, [ebp+var_10]
mov     cl, byte_41214C[ecx]
mov     [eax+edx], cl
jmp     short loc_401266</span>
```

写入完成后call loc_401000

最初感觉没有地方下手啊，把原始数据写入目标地址中，可是目标地址是与输入的key有关，这怎么破……

后来猜测0x00401000处开始的语句肯定是push ebp之类的，可是这也凑不够256字节啊，后来找大神指点。大神说观察函数的头和尾，整个程序中的函数头尾形式都是一样的，可以根据其他函数确定0x00401000处的数据。

这是0x00401101处的代码，即写入256字节数据后面的代码：

```
pop     edi
pop     esi
pop     ebx
add     esp, 58
cmp     ebp, esp
call    00401350
mov     esp, ebp
pop     ebp
ret</span>
```

再观察其他函数的开头格式，后猜测256字节数据中前9个字节代码为：

```
55  push ebp
8B EC           mov ebp, esp
83 EC 58 sub esp, 58
53  push ebx
56  push esi
57  push edi</span>
```

想要的东西都有了，可是编程功底太差，写不出解密代码，只好手工解密。

```
1D8/*55h 196 19E 1B0 1D8 1FA
22E/*8Bh 153 178 18C 192 1AA 1AB 1AF 1B3 1DE 1EB 20C 20D 211 22E 238
19F/*ECh 19F
171/*83h 164 169 16C 16F 171 17A 1EE 212 228 231 246
1EA/*ECh 1EA
1DB/*58  1A0 1DB 222
167/*53  15B 167
1DD/*56  1DD
1FD/*57  1FD  注：这些地址省略了前缀0x00412</span>
```

比如第一个字节55h，虽然在0x0041214c~0x0041224c中有多处数据为55h，可是只有1D8h即0D8h处的6Fh像是0~8中的某个数字与某个字母异或而得的数据。

可能表达的有些不清楚，假设1D8h处的55h就是最终写入0x00401000处的数据，那么0D8h处的6Fh与key中的第[（0D8h-04Ch+1）% 8 = 5]个字符异或后为0。

最终结果：

```
A B C D - E F G H
  e l l   o z m g
E XOR 6Fh = 0
C XOR 6Dh = 1
D XOR 6Eh = 2
F XOR 79h = 3
G XOR 69h = 4
H XOR 62h = 5
D XOR 6Ah = 6
B XOR 62h = 7
B XOR 6Dh = 8</span>
```

还少第一位，猜测是“H”，然后key就是“Hellozmg”，然后就猜对了……