enc 虎符CTF

原创

北风~ 于 2020-05-04 17:22:23 发布 2812 收藏 2

分类专栏： CTF

本文链接：https://blog.csdn.net/weixin_45055269/article/details/105919217

工具

IDA+IDA动调+Angr

思路展开

此题给了两个文件，一个可执行文件、一个enc为后缀的文件。根据题目描述可知exe相当于一个encoder（人话：加密的机器），而enc文件就是flag经过exe加密后的文件，里面保存着密文。分析exe，逆向算法即可。

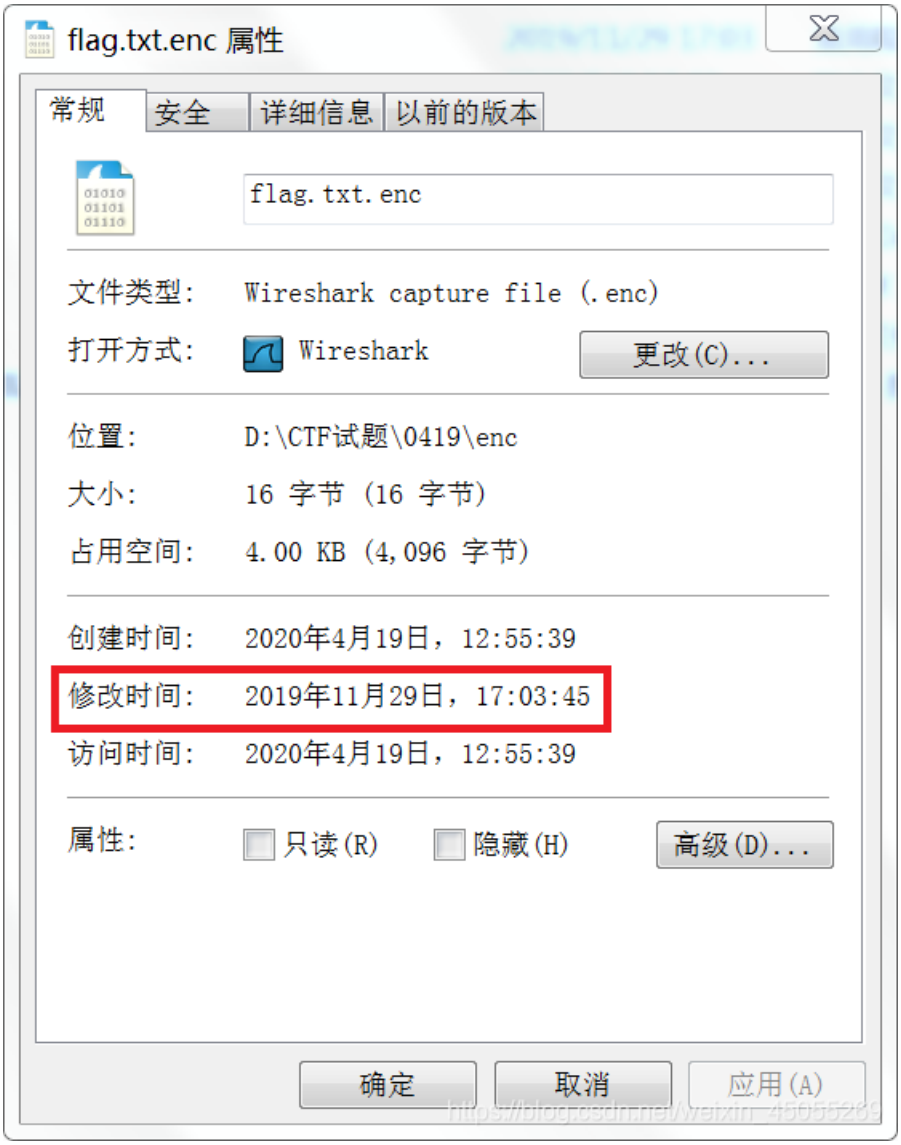
```
int __cdecl sub_401490(signed int a1, int a2)
{
  int v2; // esi
  int v3; // eax
  __int128 *v4; // edi
  signed int i; // esi
  unsigned int v6; // kr00_4
  char *v7; // edi
  unsigned int v8; // ecx
  int v9; // esi
  __int128 v11; // [esp+4h] [ebp-54h]
  __int128 v12; // [esp+14h] [ebp-44h]
  char v13[16]; // [esp+24h] [ebp-34h]
  __int128 v14; // [esp+34h] [ebp-24h]
  __int128 v15; // [esp+44h] [ebp-14h]

  v14 = 0i64;
  v15 = 0i64;
  if ( a1 > 1 )
  {
    v2 = fopen(*(_DWORD *)(a2 + 4), (int)"rb");
    fread(&v14, 1, 16, v2);
    fclose(v2);
    v11 = 0i64;
    v12 = 0i64;
    v3 = time64(0);
    srand(v3 % 177);
    rand();
    sub_401050((int)v13);
    v4 = &v11;
    i = 0;
    do
    {
      sub_401010((int)v4, (int)"%02x", (unsigned __int8)v13[i++]);
      v4 = (__int128 *)((char *)v4 + 2);
    }
    while ( i < 16 );
    sub_4012A0((int)&v11, (int *)&v14);
    v6 = strlen(*(const char **)(a2 + 4));
    v7 = (char *)malloc((v6 + 5) | -__CFADD__(v6, 5));
    strcpy(v7, *(const char **)(a2 + 4));
    v8 = (unsigned int)&v7[strlen(v7)];
    *(_DWORD *)v8 = 1668179246;
    *(_BYTE *)(v8 + 4) = 0;
    v9 = fopen((int)v7, (int)&unk_41A5CC);
    fwrite((unsigned int)&v14, 1, 16, v9);
    fclose(v9);
  }
  return 0;
}
```

main函数核心在sub_401050和sub_4012A0，以这两个函数上下延伸。

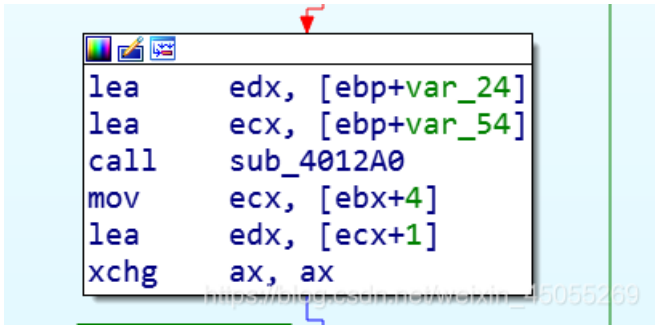
从上往下看，先是从某文件中取出flag放到v14备用。
经过time函数（下图），以文件产生的时间戳作为随机数种子，再经过sub401050函数加密（MD5算法），产生密钥放到v13，文件产生的时间固定，所以由时间产生的密钥固定（这点存疑_(:з」∠)_，应该是有177种情况），动调拿到密钥
794c87696d24d16e7b9e3dddad778c93

```
23    fclose(v2);
24    v11 = 0i64;
25    v12 = 0i64;
26    v3 = time64(0);
27    srand(v3 % 177);
28    rand();
29    sub_401050((int)v13);
30    v4 = &v11;
```



接下来的do while循环，将v13以文本形式赋值给v11。现在v11是密钥，v14是flag，两个参数传入sub_4012A0加密后的结果就是一个enc为后缀的文件里的内容（hxd打开的十六进制数据）。

```
34      sub_401010((int)v4, (int)"%02x", (unsigned __int8)v1
35      v4 = (__int128 *)((char *)v4 + 2);
36    }
37    while ( i < 16 );
38    sub_4012A0((int)&v11, (int *)&v14);
39    v6 = strlen(*(const char **)(a2 + 4));
40    v7 = (char *)malloc((v6 + 5) | -__CFADD__(v6, 5));
41    strcpy(v7, *(const char **)(a2 + 4));
42    v8 = (unsigned int)&v7[strlen(v7)];
43    *(_DWORD *)v8 = 1668179246;
44    *(_BYTE *)(v8 + 4) = 0;
```

sub_4012A0算法好复杂（orz），所以选择angr模拟执行。

```
from angr import *
from claripy import *

flag=BVS('flag',8*16) #flag的长度16字节 BVS创造输入
result=BVV(b'\xae\xed\x13\x5c\xbd\xd2\xa1\x74\x9c\x4c\x5e\x02\xd3\x28\x9b\x60',8*16) #创建变量
disasm=BVV(b'794c87696d24d16e7b9e3dddad778c93',8*32) #创建变量
p=Project('task.exe',auto_load_libs=False) #是否自动载入依赖的库
p.hook(0x405128,SIM_PROCEDURES['libc']['malloc']()) #hook SIM_PROCEDURES['模块名']['库函数名']
p.hook(0x4035e4,SIM_PROCEDURES['libc']['calloc']())
st=p.factory.full_init_state(addr=0x40154d,add_options={options.SYMBOLTC_WRITE_ADDRESSES,
options.REGION_MAPPING,options.SYMBOL_FILL_UNCONSTRAINED_REGISTERS}) #设置引擎
st.memory.store(st.regs.ebp, BVV(0, 32))  # angr bug
st.memory.store(st.regs.ebp - 0x24, flag) #传入参数
st.memory.store(st.regs.ebp - 0x54, disasm) #传入参数
sim = p.factory.simgr(st)  #模拟执行
sim.explore(find=0x401558) #正确的位置
f = sim.one_found
f.solver.add(f.memory.load(f.regs.ebp - 0x24, 16) == result) #添加约束条件
print(f.solver.eval(flag, cast_to=bytes)) #打印输入

#只模拟执行4012a0加密函数
```

flag{s3cReTH3rE}

------分割线------

ISCC 2020擂台赛Rimao-1和本题类似，密钥也是根据时间产生，可直接算出来，密钥传入TEA算法，TEA算法也可以逆，encode2上面的sub0函数是对输入的flag的操作，迭代了很多轮，肯定是我不知道的算法（自己是太菜）
key生成脚本

```
v0=0x5EACE9FB
v2=v0^0xDEADBEEF
v3=0
k=[0 for i in range(4)]
for i in range(4):
 for j in range(32):
  v3^=(v2>>j)&1
 v2=v3|2*v2
 k[i]=v2
print(k)
```

TEA解密

```
sum=0
shu=[0xBA7F,0x1CEA,0xC01A,0x4BDF,0x0DB0,0x57B6,0x0527,0x2C80,0x206A,0x2172,0x5428,0x4668,0x07FE,0x36A3,0xFFDA,0x
6075,0]
i=0
delta=0x21524111
key=[4295142953, 8590285906, 17180571813, 34361143627]
sum=delta<<4

for i in range(16):
    y=shu[i]
    z=shu[i+1]
    z+= (y + sum) ^ (8 * y + key[0]) ^ ((y >> 6) + key[1])
    y+= (z + sum) ^ (8 * z + key[2]) ^ ((z >> 6) + key[3])
    sum-= delta
    shu[i]=y
    shu[i+1]=z

print(shu)
```

存在的问题：
1.生成的密钥不是128位（TEA算法要求密钥为128位），2.密文要如何分组，
3.sub0在干啥

小白学习ing，大佬师傅们轻喷orz