



# 恶意文件分析

📅 2021-09-01 | 📁 Challenge , 2021 , 工业信息安全技能大赛 , 线上赛-第一场

Challenge | 2021 | 工业信息安全技能大赛 | 线上赛-第一场 | 恶意文件分析

[点击此处](#)获得更好的阅读体验

## WriteUp来源

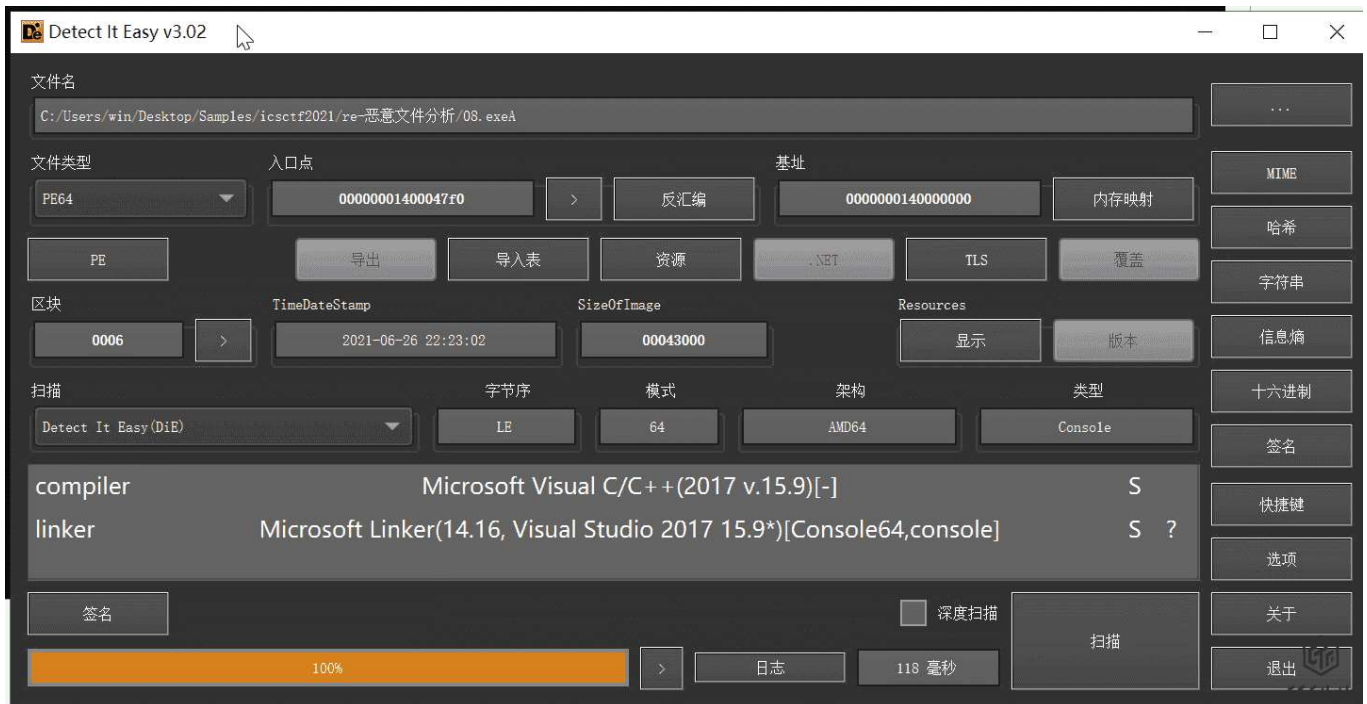
来自 **Venom** 战队

## 题目描述

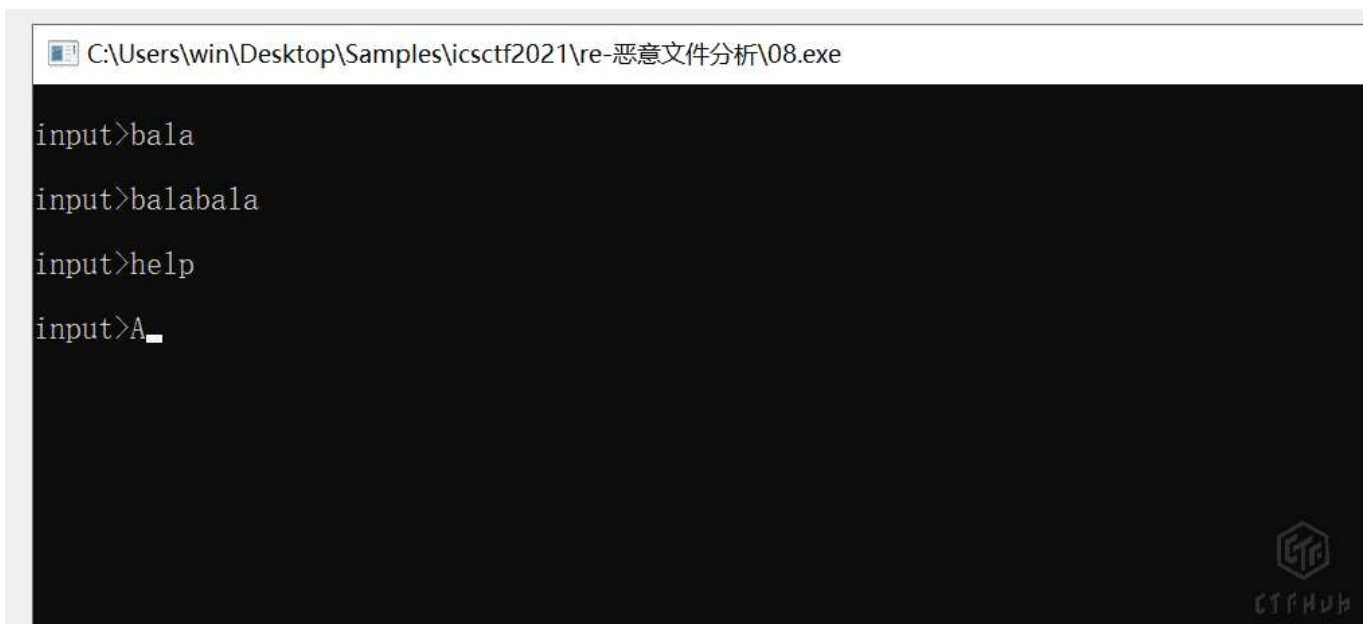
## 题目考点

## 解题思路

使用DIE查壳，没查到壳。



进去类似一个shell



程序流程，输入32长度的哈希字符串，bytes.fromhex然后运行关键 crackme 函数。

```

1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     char *ptr; // rcx
4     __int64 v4; // rcx
5     unsigned __int64 v5; // rax
6     __int64 bytes_fromhex; // rbx
7     __int64 v7; // rcx
8     char buf128[128]; // [rsp+20h] [rbp-98h] BYREF
9
10    GetCurrentThreadId();
11    memset(buf128, 0, sizeof(buf128));
12    while ( 1 )
13    {
14        ked_printf((__int64)ptr, (__int64)"\ninput>");
15        ked_gets(v4, buf128);
16        ptr = buf128;
17        v5 = -1i64;
18        do
19            ++v5;
20        while ( buf128[v5] );
21        if ( v5 >= 32 )
22        {
23            bytes_fromhex = ked_fromhex(buf128);
24            ked_printf(v7, (__int64)"Hello World!\n");
25            ked_crackme(bytes_fromhex);
26        }
27    }
28 }

```



findcrypt发现文件中有CRC32常量。

Address	Rules file	Name	String	Value
.text:0000000140003FE5	global	CRC32c_poly_Constant_140003FE5	\$c0	b'\xf6\x82'
.text:0000000140003FF4	global	CRC32c_poly_Constant_140003FF4	\$c0	b'\xf6\x82'
.text:0000000140004003	global	CRC32c_poly_Constant_140004003	\$c0	b'\xf6\x82'
.text:0000000140004012	global	CRC32c_poly_Constant_140004012	\$c0	b'\xf6\x82'
.text:0000000140004021	global	CRC32c_poly_Constant_140004021	\$c0	b'\xf6\x82'
.text:0000000140004030	global	CRC32c_poly_Constant_140004030	\$c0	b'\xf6\x82'
.text:000000014000403F	global	CRC32c_poly_Constant_14000403F	\$c0	b'\xf6\x82'
.text:0000000140004052	global	CRC32c_poly_Constant_140004052	\$c0	b'\xf6\x82'
.rdata:0000000140036470	global	CRC32_poly_Constant_140036470	\$c0	b'\x83\xb8\xed'
.rdata:0000000140036270	global	CRC32_table_140036270	\$c0	b'\x00\x00\x00\x00\x960\x07w,a\x0e\xee\xbaQ\t\x99
.rdata:0000000140036060	global	Rijndael_AES_CHAR_140036060	\$c0	b'c]w[\xf2ko\x50\x01g+\xfel\x07\xabv\xca\x82\x9j\
.rdata:0000000140036060	global	Rijndael_AES_LONG_140036060	\$c0	b'c]w[\xf2ko\x50\x01g+\xfel\x07\xabv\xca\x82\x9j\
.rdata:0000000140036160	global	Rijndael_AES_LONG_inv_140036160	\$c0	b'R\tj\x0506\xa58\xbf@\xa3\x9e\x81\x07\x07\x0e

CTFHub

crackme函数中包含了一个硬编码的key

```

21  int aes_key[4]; // [rsp+128h] [rbp+28h] BYREF
22
23  aes_key[0] = 0x16157E2B;
24  v17 = 0i64;
25  v1 = 0i64;
26  v18 = 0i64;
27  v14 = 0i64;
28  v3 = 1;
29  v15 = 0i64;
30  aes_key[1] = 0xA6D2AE28;
31  aes_key[2] = 0x8815F7AB;
32  aes_key[3] = 0x3C4FCF09;
33  v16[0] = 0xB47BD73A;
34  v16[1] = 0x60367A0D;
35  v16[2] = 0xF3CA9EA8;
36  v16[3] = 0x97EF6624;
37  v13[0] = 0xF3EBF07D;
38  v13[1] = 0x49833EAA;
39  v13[2] = 0xD6DB0614;
40  v13[3] = 0xE346C757;
41  do
42  {
43      sub_1400011E0(v12, aes_key);
44      ked_aes_wtf((unsigned __int8 *)v16, (__int64)v12);
45      v4 = *((_BYTE *)v16 + v1);

```



从开源项目中可以检索到。

[https://github.com/TurboPack/LockBox3/blob/master/run/ciphers/uTPLb\\_AES.pas](https://github.com/TurboPack/LockBox3/blob/master/run/ciphers/uTPLb_AES.pas)

[https://chromium.googlesource.com/chromiumos/platform/ec/+refs/heads/stabilize-7797.B/test/tpm\\_test/crypto\\_test.xml](https://chromium.googlesource.com/chromiumos/platform/ec/+refs/heads/stabilize-7797.B/test/tpm_test/crypto_test.xml)

输入与硬编码的Key、加密结果作比较。16轮。

```

41  do
42  {
43      ked_aes_keygen(buf256, aes_key);
44      ked_aes_wtf((unsigned __int8 *)plaintext, (__int64)buf256);
45      encrypted_i = *((_BYTE *)plaintext + val_rbx);
46      v5 = 0;
47      if ( *((_BYTE *) (val_rbx + input)) == encrypted_i )// 输入与AES输出第i位比较
48          v5 = v3;
49      *((_BYTE *)&v17 + val_rbx++) = encrypted_i;
50      v3 = v5;
51  }
52  while ( val_rbx < 16 );
53  _LOBYTE(v14) = v17;

```





循环 16 次看 ECX 值即可 dump 出预期输入。拼起来可以得到：

22d72a581f3a61e61e5b127e47ad8c0c



将这个值输入程序，得到flag

```
input>01000100010001000100010001000100
Hello World!

input>22d72a581f3a61e61e5b127e47ad8c0c
Hello World!

Offset 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 0123456789ABCDEF
-----
000000 66 6C 61 67 7B 72 6F 62 6F 74 53 61 79 48 69 7D flag{robotSayHi}
```


## Flag

1 flag{robotSayHi}

**本文作者：** CTFHub

**本文链接：** <https://writeup.ctfhub.com/Challenge/2021/工业信息安全技能大赛/线上赛-第一场/qKFsrSPACt5ULrvZZf2DtG.html>

**版权声明：** 本博客所有文章除特别声明外，均采用 [CC BY-NC-SA](#) 许可协议。转载请注明出处！

© 2019 – 2022  CTFHub  
由 [Hexo](#) & [NexT.Gemini](#) 强力驱动