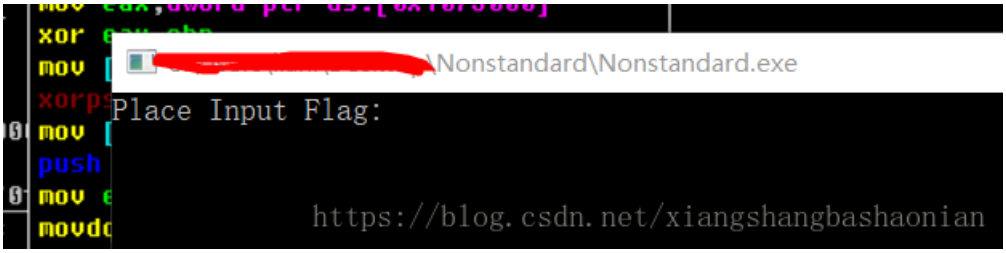
i春秋2017第二届广东省强网杯线上赛Nonstandard题目writeup

原创

iqiqiya 于 2018-08-20 15:49:25 发布 1161 收藏 1
分类专栏： 我的CTF之路 我的CTF进阶之路 文章标签： 2017第二届广东省强网杯线上赛Nonstandard writeup Nonstandard ichunqiu

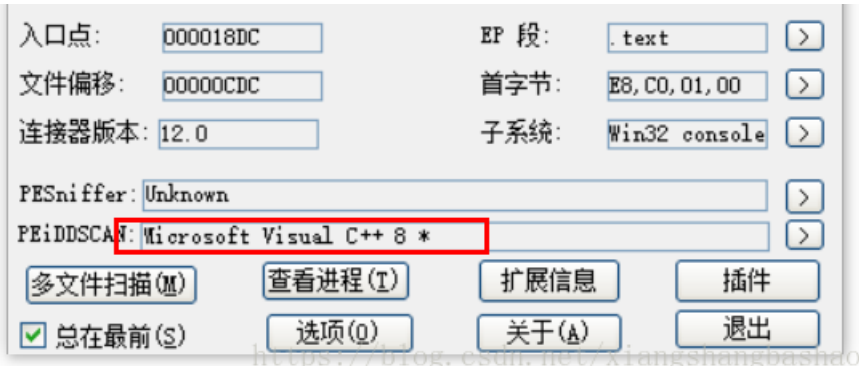

我的CTF之路 同时被 2 个专栏收录
92 篇文章 5 订阅
订阅专栏
我的CTF进阶之路
108 篇文章 18 订阅
订阅专栏

0x01：运行之后就提示让输入flag 如果输入不正确 就退出

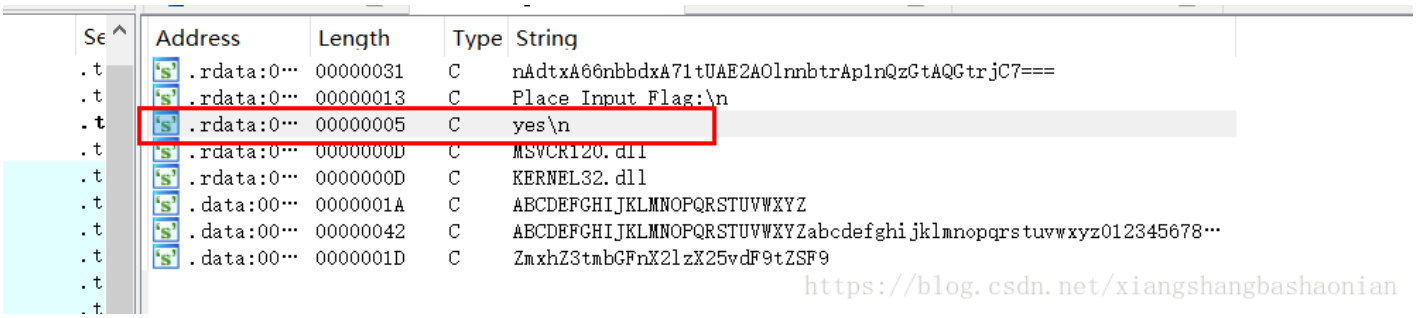


0x02:PEiD查壳发现没有加壳 VC的程序

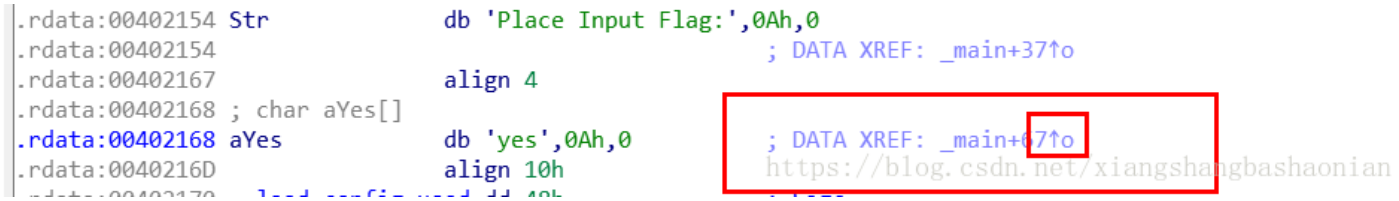
0x03:IDA载入分析 关键字符串



双击进入

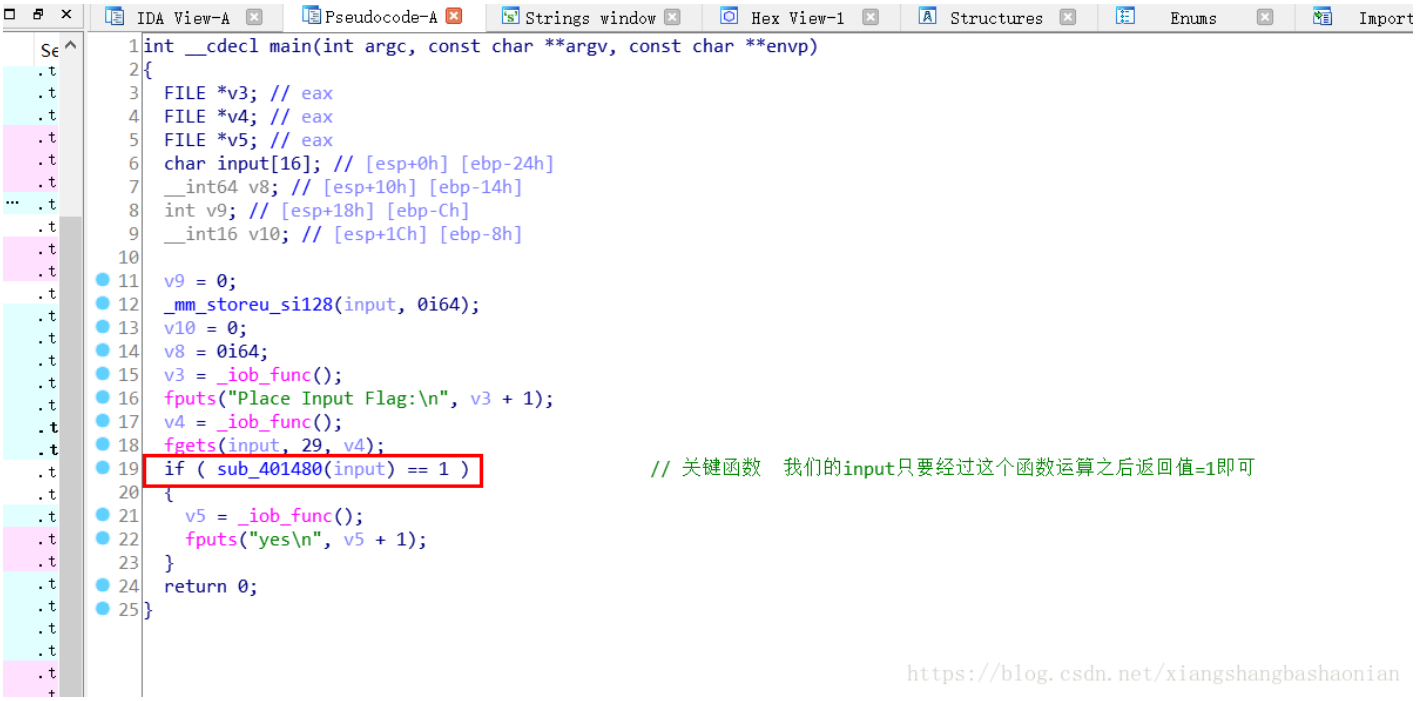


Address	Length	Type	String
.rdata:0…	00000031	C	nAdtxA66nbbdxA71tUAE2AOlnnbtrAp1nQzGtAQGtrjC7===
.rdata:0…	00000013	C	Place Input Flag:\n
.rdata:0…	00000005	C	yes\n
.rdata:0…	0000000D	C	MSVCR120.dll
.rdata:0…	0000000D	C	KERNEL32.dll
.data:00…	0000001A	C	ABCDEFGHIJKLMNOPQRSTUVWXYZ
.data:00…	00000042	C	ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz012345678…
.data:00…	0000001D	C	ZmxhZ3tmbGFnX2lzX25vdF9tZSF9



```
.rdata:00402154 Str             db 'Place Input Flag:',0Ah,0
.rdata:00402154                                         ; DATA XREF: _main+37↑o
.rdata:00402167                 align 4
.rdata:00402168 ; char aYes[]
.rdata:00402168 aYes            db 'yes',0Ah,0          ; DATA XREF: _main+67↑o
.rdata:0040216D                 align 10h
```


双击进入 F5大法 找到关键代码



IDA View-A | Pseudocode-A | Strings window | Hex View-1 | Structures | Enums | Import

```
1  int __cdecl main(int argc, const char **argv, const char **envp)
2  {
3    FILE *v3; // eax
4    FILE *v4; // eax
5    FILE *v5; // eax
6    char input[16]; // [esp+0h] [ebp-24h]
7    __int64 v8; // [esp+10h] [ebp-14h]
8    int v9; // [esp+18h] [ebp-Ch]
9    __int16 v10; // [esp+1Ch] [ebp-8h]
10
11   v9 = 0;
12   _mm_storeu_si128(input, 0i64);
13   v10 = 0;
14   v8 = 0i64;
15   v3 = _iob_func();
16   fputs("Place Input Flag:\n", v3 + 1);
17   v4 = _iob_func();
18   fgets(input, 29, v4);
19   if ( sub_401480(input) == 1 )              // 关键函数  我们的input只要经过这个函数运算之后返回值=1即可
20   {
21     v5 = _iob_func();
22     fputs("yes\n", v5 + 1);
23   }
24   return 0;
25 }
```


既然关键在于sub_401480这个函数 那就双击进入一探究竟

```
signed int __thiscall sub_401480(const char *input)
{
  const char *v1; // esi
  const char *v2; // eax
  unsigned int v3; // eax
  unsigned int v4; // kr04_4
  signed int result; // eax
  char v6; // [esp+4h] [ebp-38h]
  char Dst; // [esp+5h] [ebp-37h]

  v6 = 0;
  v1 = input;                                   // 将我们的输入赋值给v1  那么v1就是input
  memset(&Dst, 0, 0x31u);
  if ( strlen(v1) != 28 )                       // input长度需要等于28
    goto LABEL_10;
  v2 = sub_401070(v1, 0x1Cu);                   // 对input做一系列操作(重点)
  strncpy_s(&v6, 0x32u, v2, 0x30u);
  v3 = 0;
  v4 = strlen(&v6);
  if ( !v4 )
    goto LABEL_10;
  do                                            // 正确的走向
  {
    if ( aNadtxa66nbbdxa[v3] != *(&v6 + v3) )
      break;                                    // 验证经过sub_401070这个函数运算之后是否等于v3  也就是nAdtxA66nbbdxA71tUAE2AOlnnbtrAp1nQzGtAQGtrjC7===
    ++v3;
  }
  while ( v3 < v4 );
  if ( v3 == '0' )
    result = 1;
  else
LABEL_10:
    result = -1;
  return result;
}
```


现在就明白了 我们的input经过sub_401070这个函数运算之后是否等于v3 也就是nAdtxA66nbbdxA71tUAE2AOlnnbtrAp1nQzGtAQGtrjC7===

如果相等 sub_40148函数返回值=1 就输出yes

则input就是我们要得到的flag

一定要注意这个字母n IDA分析后默认是没有和其他在一块的（这是一个大坑 还好我提前OD动态调试了一波）

```
.rdata:00402120 byte_402120     db 6Eh                  ; DATA XREF: sub_401480:loc_4014F0↑r
.rdata:00402121 aAdtxa66nbbdxa7 db 'AdtxA66nbbdxA71tUAE2AOlnnbtrAp1nQzGtAQGtrjC7===',0
```

转换之后

```
.rdata:00402116                 align 10h
.rdata:00402120 ; char aNadtxa66nbbdxa[]
.rdata:00402120 aNadtxa66nbbdxa db 'nAdtxA66nbbdxA71tUAE2AOlnnbtrAp1nQzGtAQGtrjC7===',0
.rdata:00402120                                         ; DATA XREF: sub_401480:loc_4014F0↑r
.rdata:00402151                 align 4
.rdata:00402154 ; char Str[]
.rdata:00402154 Str             db 'Place Input Flag:',0Ah,0
.rdata:00402154                                         ; DATA XREF: _main+37↑o
.rdata:00402167                 align 4
.rdata:00402168 ; char aYes[]
```

```
        + (v23 & 0x7C)) << 8);
  *v27 = ABCDEFGHIJKLMNOPQRSTUVWXYZ[(BYTE4(v20) >> 3)];
  v27[1] = ABCDEFGHIJKLMNOPQRSTUVWXYZ[(v20 >> 30) & 0x1F];
  v27[2] = ABCDEFGHIJKLMNOPQRSTUVWXYZ[(v20 >> 25) & 0x1F];
  v27[3] = ABCDEFGHIJKLMNOPQRSTUVWXYZ[(v20 >> 20) & 0x1F];
  v27[4] = ABCDEFGHIJKLMNOPQRSTUVWXYZ[(v20 >> 15) & 0x1F];
  v27[5] = ABCDEFGHIJKLMNOPQRSTUVWXYZ[(v20 >> 10) & 0x1F];
  v3 = v28;
  v27[6] = ABCDEFGHIJKLMNOPQRSTUVWXYZ[(v20 >> 5) & 0x1F];
  v2 = v35;
  v27[7] = ABCDEFGHIJKLMNOPQRSTUVWXYZ[v20 & 0x1F];
  v27 += 8;
}
while ( v8 < v28 );
```


发现5个bit为一组，分别赋给8个值，每个值5位

就是base32呀

```
39  v30 = a2;
40  v35 = a1;
41  sub_401000();                                   // 下边是base32
42  v3 = 0;
```


接着分析sub_401000()

一共有三步操作：

1. 将ABCDEFGHIJKLMNOPQRSTUVWXYZ偶数位变为小写字母　也就是 AbCdEfGhIjKlMnOpQrStUvWxYz

2. 再将字符串进行逆序

3. 后面加上765321

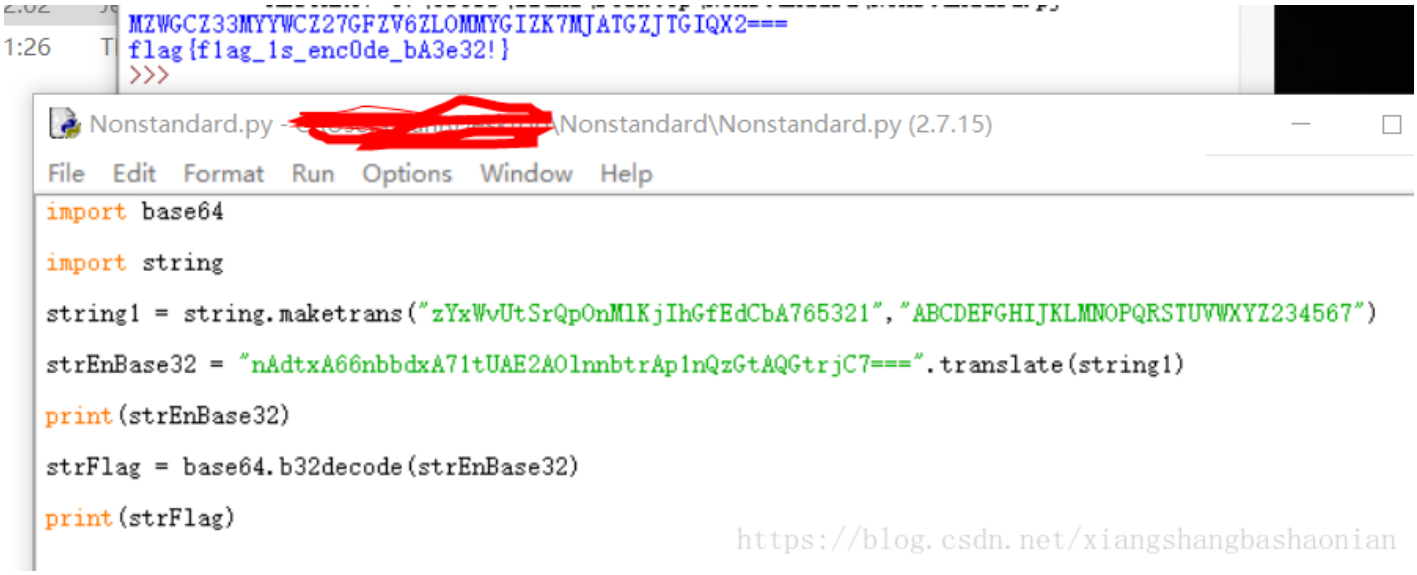
那么base32编码表从默认的ABCDEFGHIJKLMNOPQRSTUVWXYZ234567就变为了 zYxWvUtSrQpOnMlKjIhGfEdCbA765321

所以我们需要用Python translate() 方法与maketrans()方法来将 nAdtxA66nbbdxA71tUAE2AOlnnbtrAp1nQzGtAQGtrjC7===还原

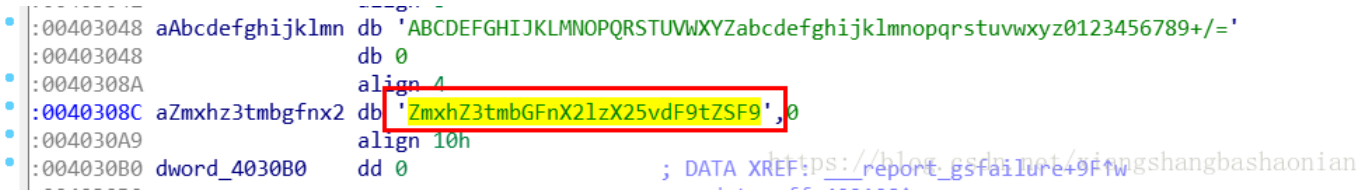
在解一次base322即可

至此，关键代码分析完毕。

Py大法实现：



得到flag{f1ag_1s_enc0de_bA3e32!}



对了，也有一个假flag

ZmxhZ3tmbGFnX2lzX25vdF9tZSF9

用base64解密一下（算是个彩蛋吧）

题目+idb分析文件+py脚本已全部打包

百度网盘下载链接: https://pan.baidu.com/s/1V-inJEXtPwopyKHtTqP02A 密码: qiu3