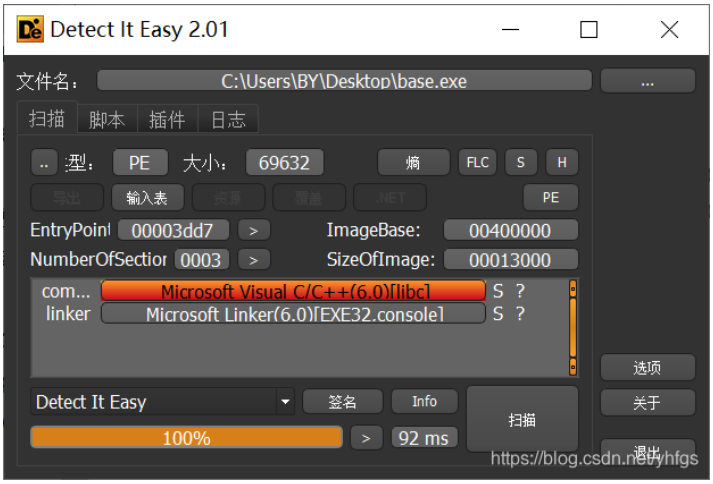
buuctf——（ACTF新生赛2020）usualCrypt

原创

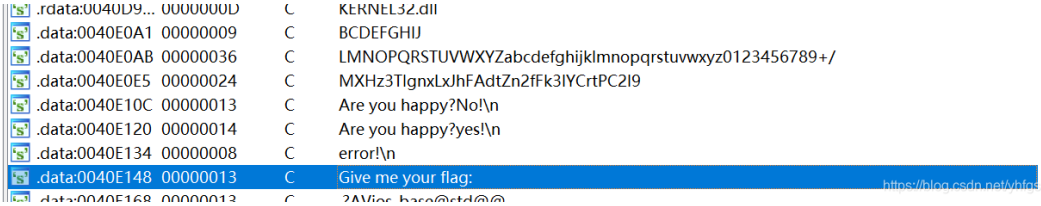
re3sry 于 2021-06-01 19:28:21 发布 115 收藏

1。查壳。

无壳，32位。



2.丢到IDA中，查看字符串。



找到falg字样，（在Are you happy?No!上面还看到ABCD。。。789+/猜测可能有base64加密。）

找到主函数。反编译。

```
  printf((int)&unk_40E140);
  scanf("%s", v8);
  v5[0] = 0;
  v5[1] = 0;
  v5[2] = 0;
  v6 = 0;
  v7 = 0;
  sub_401080(v8, strlen(v8), v5);               // base64（换表）
  v3 = 0;
  while ( *((_BYTE *)v5 + v3) == byte_40E0E4[v3] )
  {
    if ( ++v3 > strlen((const char *)v5) )
      goto LABEL_6;
  }
  printf((int)aError);
LABEL_6:
  if ( v3 - 1 == strlen(byte_40E0E4) )
    result = printf((int)aAreYouHappyYes);
  else
    result = printf((int)aAreYouHappyNo);
  return result;
}
```


3。分析代码。发现关键在sub_401080（加密过程）,最后比较（给出了加密后的字符串：zMXHz3TlgnxLxJhFAdtZn2fFk3lYCrtPC2l9）

进入sub_401080分析。

```
  v3 = 0;
  v4 = 0;
  sub_401000();
  v5 = a2 % 3;
  v6 = a1;
  v7 = a2 - a2 % 3;
  v15 = a2 % 3;
  if ( v7 > 0 )
  {
    do
    {
      LOBYTE(v5) = *(_BYTE *)(a1 + v3);
      v3 += 3;
      v8 = v4 + 1;
      *(_BYTE *)(v8 + a3 - 1) = byte_40E0A0[(v5 >> 2) & 0x3F];
      *(_BYTE *)(++v8 + a3 - 1) = byte_40E0A0[16 * (*(_BYTE *)(a1 + v3 - 3) & 3)
                                           + (((int)*(unsigned __int8 *)(a1 + v3 - 2) >> 4) & 0xF)];
      *(_BYTE *)(++v8 + a3 - 1) = byte_40E0A0[4 * (*(_BYTE *)(a1 + v3 - 2) & 0xF)
                                           + (((int)*(unsigned __int8 *)(a1 + v3 - 1) >> 6) & 3)];
      v5 = *(_BYTE *)(a1 + v3 - 1) & 0x3F;
      v4 = v8 + 1;
      *(_BYTE *)(v4 + a3 - 1) = byte_40E0A0[v5];
    }
    while ( v3 < v7 );
```



```
  {
    v11 = v4 + 1;
    *(_BYTE *)(v11 + a3 - 1) = byte_40E0A0[((int)*(unsigned __int8 *)(v3 + a1) >> 2) & 0x3F];
    v12 = (_BYTE *)(v3 + a1 + 1);
    LOBYTE(v6) = *v12;
    v10 = v11 + 1;
    *(_BYTE *)(v10 + a3 - 1) = byte_40E0A0[16 * (*(_BYTE *)(v3 + a1) & 3) + ((v6 >> 4) & 0xF)];
    *(_BYTE *)(v10 + a3) = byte_40E0A0[4 * (*v12 & 0xF)];
    goto LABEL_8;
  }
LABEL_9:
  *(_BYTE *)(v4 + a3) = 0;
  return sub_401030(a3);
}
```


发现大体是base64加密。but，在前有个sub_401000函数，在最后还有sub_401030函数。

分别进入。

sub_401000:发现是对base64的表进行的变换（即换表，很简单）。

```
int sub_401000()
{
  int result; // eax
  char v1; // cl

  for ( result = 6; result < 15; ++result )
  {
    v1 = byte_40E0AA[result];
    byte_40E0AA[result] = byte_40E0A0[result];
    byte_40E0A0[result] = v1;
  }
  return result;
}
```


sub_401030:是对bsae64换表加密后字符串的又一层简单算法

（大写字母转小写，小写转大写，数字不变）。

```
  v1 = 0i64;
  if ( strlen(a1) )
  {
    do
    {
      v2 = a1[HIDWORD(v1)];
      if ( v2 < 97 || v2 > 122 )
      {
        if ( v2 < 65 || v2 > 90 )
          goto LABEL_9;
        LOBYTE(v1) = v2 + 32;
      }
      else
      {
        LOBYTE(v1) = v2 - 32;
      }
      a1[HIDWORD(v1)] = v1;
LABEL_9:
      LODWORD(v1) = 0;
      ++HIDWORD(v1);
    }
    while ( HIDWORD(v1) < strlen(a1) );
  }
  return v1;
}
```


4.写脚本。

File Edit Format Run Options Window Help

```
import base64
import string

#换表
a='ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
a1=list(a)
for i in range(6,15):
    x=a1[10+i]
    a1[10+i]=a1[i]
    a1[i]=x
string1="".join(a1) #新表
print(string1)

#字符串转化
s="zMXHz3TIgnxLxJhFAdtZn2fFk31YCrtPC219"
str1=''
for i in s:
    if 'a'<=i<='z':
        str1+=chr(ord(i)-32)
    elif 'A'<=i<='Z':
        str1+=chr(ord(i)+32)
    else:
        str1+=i
print(str1)

#base64换表解密

string2="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"    #旧

flag=""
flag += str(base64.b64decode(str1.translate(str.maketrans(string1,string2))))
print(flag)
```


5.get flag

flag{bAse64_h2s_a_Surprise}