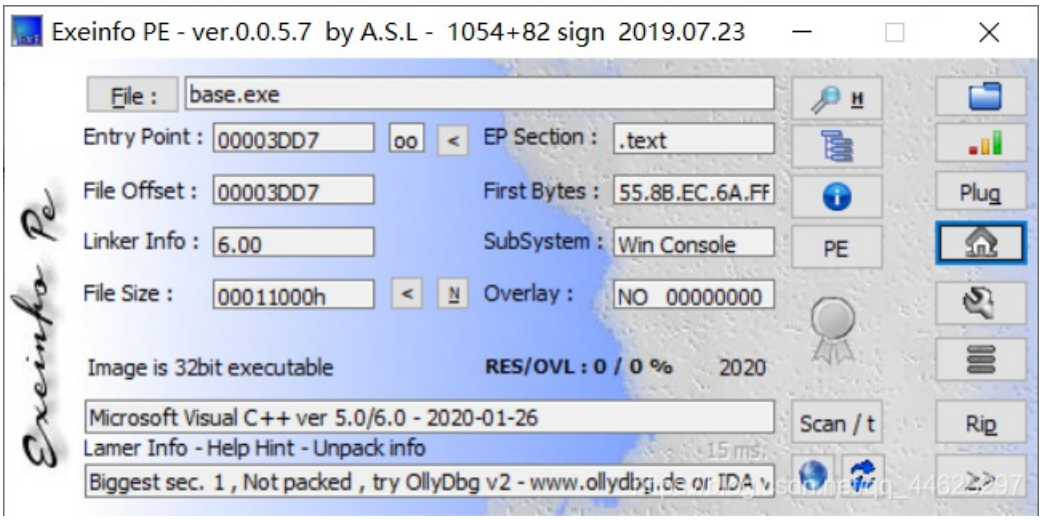
buuctf-re-[ACTF新生赛2020]usualCrypt

原创

不学无术@ 于 2020-07-20 15:51:49 发布 1006 收藏 2

拿到程序首先查壳。



发现没有加壳，用IDA32位打开。

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int v3; // esi
  int result; // eax
  int v5; // [esp+8h] [ebp-74h]
  int v6; // [esp+Ch] [ebp-70h]
  int v7; // [esp+10h] [ebp-6Ch]
  __int16 v8; // [esp+14h] [ebp-68h]
  char v9; // [esp+16h] [ebp-66h]
  char v10; // [esp+18h] [ebp-64h]

  sub_403CF8(&unk_40E140);                      // 输出
  scanf(aS, &v10);                              // 输入
  v5 = 0;
  v6 = 0;
  v7 = 0;
  v8 = 0;
  v9 = 0;
  sub_401080((int)&v10, strlen(&v10), (int)&v5);// 加密函数
  v3 = 0;
  while ( *((_BYTE *)&v5 + v3) == byte_40E0E4[v3] )// 比较函数
  {
    if ( ++v3 > strlen((const char *)&v5) )
      goto LABEL_6;
  }
  sub_403CF8(aError);
LABEL_6:
  if ( v3 - 1 == strlen(byte_40E0E4) )
    result = sub_403CF8(aAreYouHappyYes);
  else
    result = sub_403CF8(aAreYouHappyNo);
  return result;
}
```


找到主函数。分析一下，大概就是输入，加密，然后比较。进入加密函数看一下。

```
int __cdecl sub_401080(int a1, int a2, int a3)
{
  int v3; // edi
  int v4; // esi
  int v5; // edx
  int v6; // eax
  int v7; // ecx
  int v8; // esi
  int v9; // esi
  int v10; // esi
  int v11; // esi
  _BYTE *v12; // ecx
  int v13; // esi
  int v15; // [esp+18h] [ebp+8h]

  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)
                                            + (((signed int)*(unsigned __int8 *)(a1 + v3 - 2) >> 4) & 0xF)];
      *(_BYTE *)(v8 + a3 - 1) = byte_40E0A0[4 * (*(_BYTE *)(a1 + v3 - 2) & 0xF)
                                          + (((signed 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 );
    v5 = v15;
  }
  if ( v5 == 1 )
  {
    LOBYTE(v7) = *(_BYTE *)(v3 + a1);
    v9 = v4 + 1;
    *(_BYTE *)(v9 + a3 - 1) = byte_40E0A0[(v7 >> 2) & 0x3F];
    v10 = v9 + 1;
    *(_BYTE *)(v10 + a3 - 1) = byte_40E0A0[16 * (*(_BYTE *)(v3 + a1) & 3)];
    *(_BYTE *)(v10 + a3) = 61;
LABEL_8:
    v13 = v10 + 1;
    *(_BYTE *)(v13 + a3) = 61;
    v4 = v13 + 1;
    goto LABEL_9;
  }
  if ( v5 == 2 )
  {
    v11 = v4 + 1;
    *(_BYTE *)(v11 + a3 - 1) = byte_40E0A0[((signed int)*(unsigned __int8 *)(v3 + a1) >> 2) & 0x3F];
    v12 = (_BYTE *)(v3 + a1 + 1);
```

```
    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);
}
```

经过分析（猜），看到有个base64转换表，大概这是一个base64的加密。。。同时看一下sub_401000()函数。

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

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


这里是将base64转换表个别顺序进行交换。函数中两个数组分别是转换表的不同位置。再看一下加密函数最后的sub_401030()函数。

```
int __cdecl sub_401030(const char *a1)
{
  __int64 v1; // rax
  char v2; // al

  v1 = 0i64;
  if ( strlen(a1) != 0 )
  {
    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;
}
```


这个函数就是将加密后的结果大小写互换。加密后的结果就在byte_40E0E4数组中。
大概思路就是：–>结果大小写互换–>修改base64转换表–>加密结果通过转换表得到正常的加密后的结果–>base64解密。
写脚本。

```python
import base64
#secret = 'MXHz3TIgnxLxJhFAdtZn2fFk3lYCrtPC2l9'
#secret = secret.swapcase()
#不知道为啥，用上述代码会报错
secret = 'zMXHz3TIgnxLxJhFAdtZn2fFk3lYCrtPC2l9'.swapcase() #大小写转换
a = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
dict = {}
offset = 10
flag = ''
for i in range(len(a)):
    dict[a[i]] = a[i]
for i in range(6,15):  #构造修改后的表
    b = dict[a[i]]
    dict[a[i]] = dict[a[i+offset]]
    dict[a[i+offset]] = b
for i in range(len(secret)):
    flag += dict[secret[i]]
flag = base64.b64decode(flag)
print(flag)

flag{bAse64_h2s_a_Surprise}
```