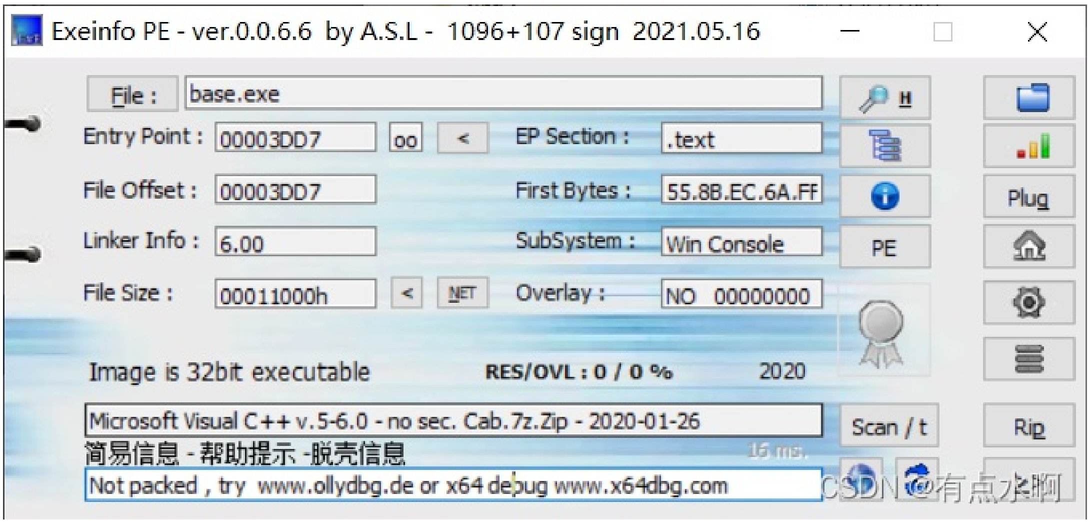
buu-[ACTF新生赛2020]usualCrypt

原创

有点水啊 于 2022-03-02 20:41:14 发布 135 收藏
分类专栏： buuctf-reserve 文章标签： 安全

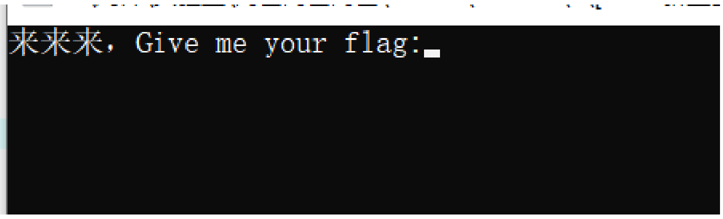
本文链接：https://blog.csdn.net/qaq517384/article/details/123216150
版权
buuctf-reserve 专栏收录该内容
59 篇文章 0 订阅
订阅专栏
32位无壳



Exeinfo PE - ver.0.0.6.6 by A.S.L - 1096+107 sign 2021.05.16

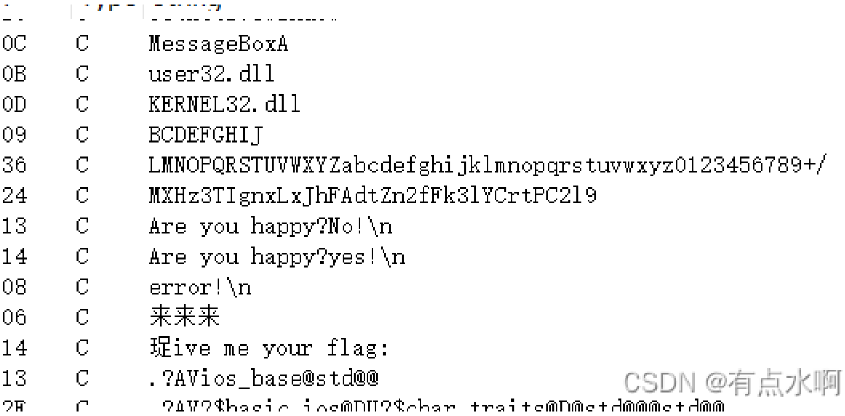
File : base.exe
Entry Point : 00003DD7
EP Section : .text
File Offset : 00003DD7
First Bytes : 55.8B.EC.6A.FF
Linker Info : 6.00
SubSystem : Win Console
File Size : 00011000h
Overlay : NO 00000000
Image is 32bit executable
RES/OVL : 0 / 0 % 2020
Microsoft Visual C++ v.5-6.0 - no sec. Cab.7z.Zip - 2020-01-26
简易信息 - 帮助提示 -脱壳信息
Not packed , try www.ollydbg.de or x64 debug www.x64dbg.com

又是一个直接退出的



```
来来来，Give me your flag:_
```

留意一下特殊字符 ~~（先猜一个base扔在这里~~



```
0C    C    MessageBoxA
0B    C    user32.dll
0D    C    KERNEL32.dll
09    C    BCDEFGHIJ
36    C    LMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
24    C    MXHz3TIgnxLxJhFAdtZn2fFk3lYCrtPC2l9
13    C    Are you happy?No!\n
14    C    Are you happy?yes!\n
08    C    error!\n
06    C    来来来
14    C    琨ive me your flag:
13    C    .?AVios_base@std@@
2E    C     ?AV?$basic_ios@DU?$char_traits@D@std@@@std@@
```

跟进main函数

```
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((int)&byte_40E140);
  scanf(aS, &v10);
  v5 = 0;
  v6 = 0;
  v7 = 0;
  v8 = 0;
  v9 = 0;
  sub_401080((int)&v10, strlen(&v10), (int)&v5);
  v3 = 0;
  while ( *((_BYTE *)&v5 + v3) == byte_40E0E4[v3] )
  {
    if ( ++v3 > strlen((const char *)&v5) )
      goto LABEL_6;
  }
  sub_403CF8((int)aError);
LABEL_6:
  if ( v3 - 1 == strlen(byte_40E0E4) )
    result = sub_403CF8((int)aAreYouHappyYes);
  else
    result = sub_403CF8((int)aAreYouHappyNo);
  return result;
}
```

输出输入，通过sub_401080()函数加密后与byte_40E0E4比较
byte_40E0E4=`zMXHz3TIgnxLxJhFAdtZn2fFk3lYCrtPC2l9`

```
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) = dword_40E0A0[(v5 >> 2) & 0x3F];
      *(_BYTE *)(v8++ + a3 - 1) = dword_40E0A0[16 * (*(_BYTE *)(a1 + v3 - 3) & 3)
                                             + (((signed int)*(unsigned __int8 *)(a1 + v3 - 2) >> 4) & 0xF)];
      *(_BYTE *)(v8 + a3 - 1) = dword_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) = dword_40E0A0[v5];
    }
    while ( v3 < v7 );
    v5 = v15;
  }
  if ( v5 == 1 )
  {
    LOBYTE(v7) = *(_BYTE *)(v3 + a1);
    v9 = v4 + 1;
    *(_BYTE *)(v9 + a3 - 1) = dword_40E0A0[(v7 >> 2) & 0x3F];
    v10 = v9 + 1;
    *(_BYTE *)(v10 + a3 - 1) = dword_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) = dword_40E0A0[((signed int)*(unsigned __int8 *)(v3 + a1) >> 2) & 0x3F];
    v12 = (_BYTE *)(v3 + a1 + 1);
    LOBYTE(v6) = *v12;
    v10 = v11 + 1;
    *(_BYTE *)(v10 + a3 - 1) = dword_40E0A0[16 * (*(_BYTE *)(v3 + a1) & 3) + ((v6 >> 4) & 0xF)];
    *(_BYTE *)(v10 + a3) = dword_40E0A0[4 * (*v12 & 0xF)];
    goto LABEL_8;
    goto LABEL_8;
  }
LABEL_9:
  *(_BYTE *)(v4 + a3) = 0;
  return sub_401030(a3);
}
```

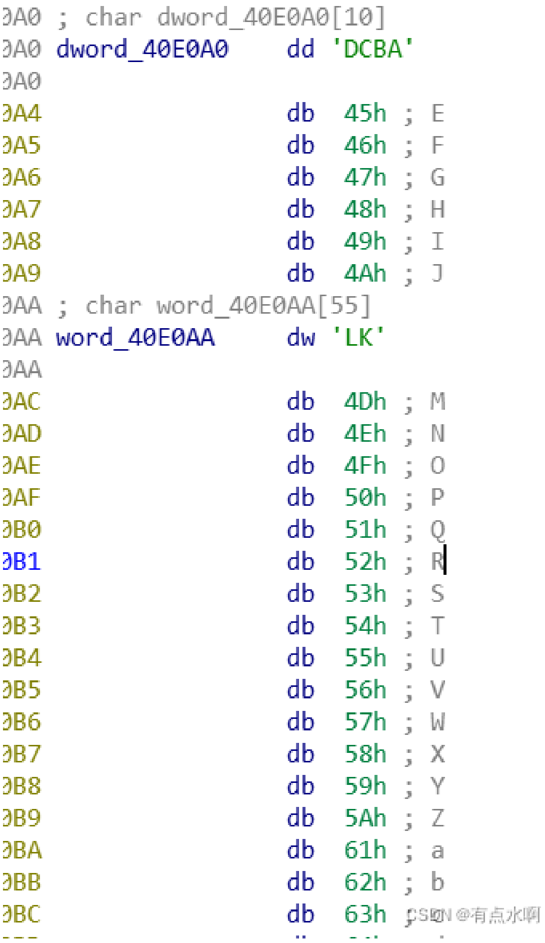
开头结尾一个 sub_401000()函数，一个sub_401030()函数,中间是一个base64加密

查看sub_401000()

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

  result = 6;
  do
  {
    v1 = word_40E0AA[result];
    word_40E0AA[result] = dword_40E0A0[result];
    dword_40E0A0[result++] = v1;
  }
  while ( result < 15 );
  return result;
}
```

把word_40E0AA和dword_40E0A0的下标6到14替换
查看地址发现是同一串字符串的顺序更改一下



```
0A0 ; char dword_40E0A0[10]
0A0 dword_40E0A0    dd 'DCBA'
0A0
0A4                 db  45h ; E
0A5                 db  46h ; F
0A6                 db  47h ; G
0A7                 db  48h ; H
0A8                 db  49h ; I
0A9                 db  4Ah ; J
0AA ; char word_40E0AA[55]
0AA word_40E0AA     dw 'LK'
0AA
0AC                 db  4Dh ; M
0AD                 db  4Eh ; N
0AE                 db  4Fh ; O
0AF                 db  50h ; P
0B0                 db  51h ; Q
0B1                 db  52h ; R
0B2                 db  53h ; S
0B3                 db  54h ; T
0B4                 db  55h ; U
0B5                 db  56h ; V
0B6                 db  57h ; W
0B7                 db  58h ; X
0B8                 db  59h ; Y
0B9                 db  5Ah ; Z
0BA                 db  61h ; a
0BB                 db  62h ; b
0BC                 db  63h ; c
```

也就是`QRSTUVWXY`和`GHIJKLMNOP`相互交换了一下，就是更改base64的密码表

查看另一个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;
}
```

一个大小写转换，过了

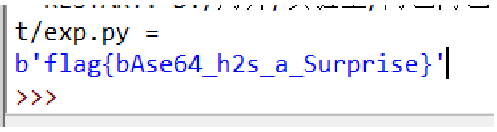
解题思路就是
转换大小写
更改base64密码表解密（参考）

```
import base64
import string

str1 = 'zMXHz3TIgnxLxJhFAdtZn2fFk3lYCrtPC2l9'.swapcase()
string1 = "ABCDEFQRSTUVWXYPGHIJKLMNOZabcdefghijklmnopqrstuvwxyz0123456789+/"
#更改后的密码表
string2 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"

print (base64.b64decode(str1.translate(str.maketrans(string1,string2))))
```

一跑就出了



```
t/exp.py =
b'flag{bAse64_h2s_a_Surprise}'
>>>
```

`flag{bAse64_h2s_a_Surprise}`