BUUCTF--[ACTF新生赛2020]usualCrypt

原创

Hk_Mayfly 于 2020-04-15 00:33:00 发布 424 收藏

测试文件：https://lanzous.com/ibeasxg

代码分析

```
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]

  puts((int)&unk_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;
  }
  puts((int)aError);
LABEL_6:
  if ( v3 - 1 == strlen(byte_40E0E4) )
    result = puts((int)aAreYouHappyYes);
  else
    result = puts((int)aAreYouHappyNo);
  return result;
}
```

代码过程很明显，就是输入flag经过sub_401080函数处理后，与byte_40E0E4比较，相同就success

byte_40E0E4

```
zMXHz3TlgnxLxJhFAdtZn2fFk3lYCrtPC2l9
```

sub_401080函数

```
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) = BASE64_table_40E0A0[(v5 >> 2) & 0x3F];
      *(_BYTE *)(v8++ + a3 - 1) = BASE64_table_40E0A0[16 * (*(_BYTE *)(a1 + v3 - 3) & 3)
                                                    + (((signed int)*(unsigned __int8 *)(a1 + v3 - 2) >> 4)
& 0xF)];
      *(_BYTE *)(v8 + a3 - 1) = BASE64_table_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) = BASE64_table_40E0A0[v5];
    }
    while ( v3 < v7 );
    v5 = v15;
  }
  if ( v5 == 1 )
  {
    LOBYTE(v7) = *(_BYTE *)(v3 + a1);
    v9 = v4 + 1;
    *(_BYTE *)(v9 + a3 - 1) = BASE64_table_40E0A0[(v7 >> 2) & 0x3F];
    v10 = v9 + 1;
    *(_BYTE *)(v10 + a3 - 1) = BASE64_table_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) = BASE64_table_40E0A0[((signed int)*(unsigned __int8 *)(v3 + a1) >> 2) & 0x3F];
    v12 = (_BYTE *)(v3 + a1 + 1);
    LOBYTE(v6) = *v12;
    v10 = v11 + 1;
    *(_BYTE *)(v10 + a3 - 1) = BASE64_table_40E0A0[16 * (*(_BYTE *)(v3 + a1) & 3) + ((v6 >> 4) & 0xF)];
    *(_BYTE *)(v10 + a3) = BASE64_table_40E0A0[4 * (*v12 & 0xF)];
    goto LABEL_8;
  }
LABEL_9:
  *(_BYTE *)(v4 + a3) = 0;
  return sub_401030((const char *)a3);
}
```

这个函数就可以分为三个部分

- 开头sub_401000函数
- 中间base64加密
- 结尾sub_401030函数

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] = BASE64_table_40E0A0[result];
    BASE64_table_40E0A0[result++] = v1;
  }
  while ( result < 15 );
  return result;
}
```

这里就是一个对base64的变表处理

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;
}
```

将经过base64变表加密的结果，大小写互换。

因此，我们只需要反过来，先将byte_40E0E4大小写互换，构造base64变表，再利用变表将byte_40E0E4转换为正常的base64解密就行。

脚本

```
# -*- coding:utf-8 -*-
import base64

Str = list("ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/")
model = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
for i in range(6,15):
    Str[i],Str[i+10] = Str[i+10],Str[i]
Str = ''.join(Str)
enc = "zMXHz3TIgnxLxJhFAdtZn2fFk3lYCrtPC2l9".swapcase()
dec = ""
for i in range(len(enc)):
    dec += model[Str.find(enc[i])]
print (dec)
print (Str)
print (base64.b64decode(dec))
```

```
D:\Anaconda\python.exe C:/Users/10245/PycharmProjects/untitled1/pro109.py
ZmxhZ3tiQXNlNjRfaDJzX2FfU3VycHJpc2V9
ABCDEFQRSTUVWXYPGHIJKLMNOZabcdefghijklmnopqrstuvwxyz0123456789+/
b'flag{bAse64_h2s_a_Surprise}'
```

get flag！

> flag{bAse64_h2s_a_Surprise}