buuctf刷题记录10 [ACTF新生赛2020]usualCrypt

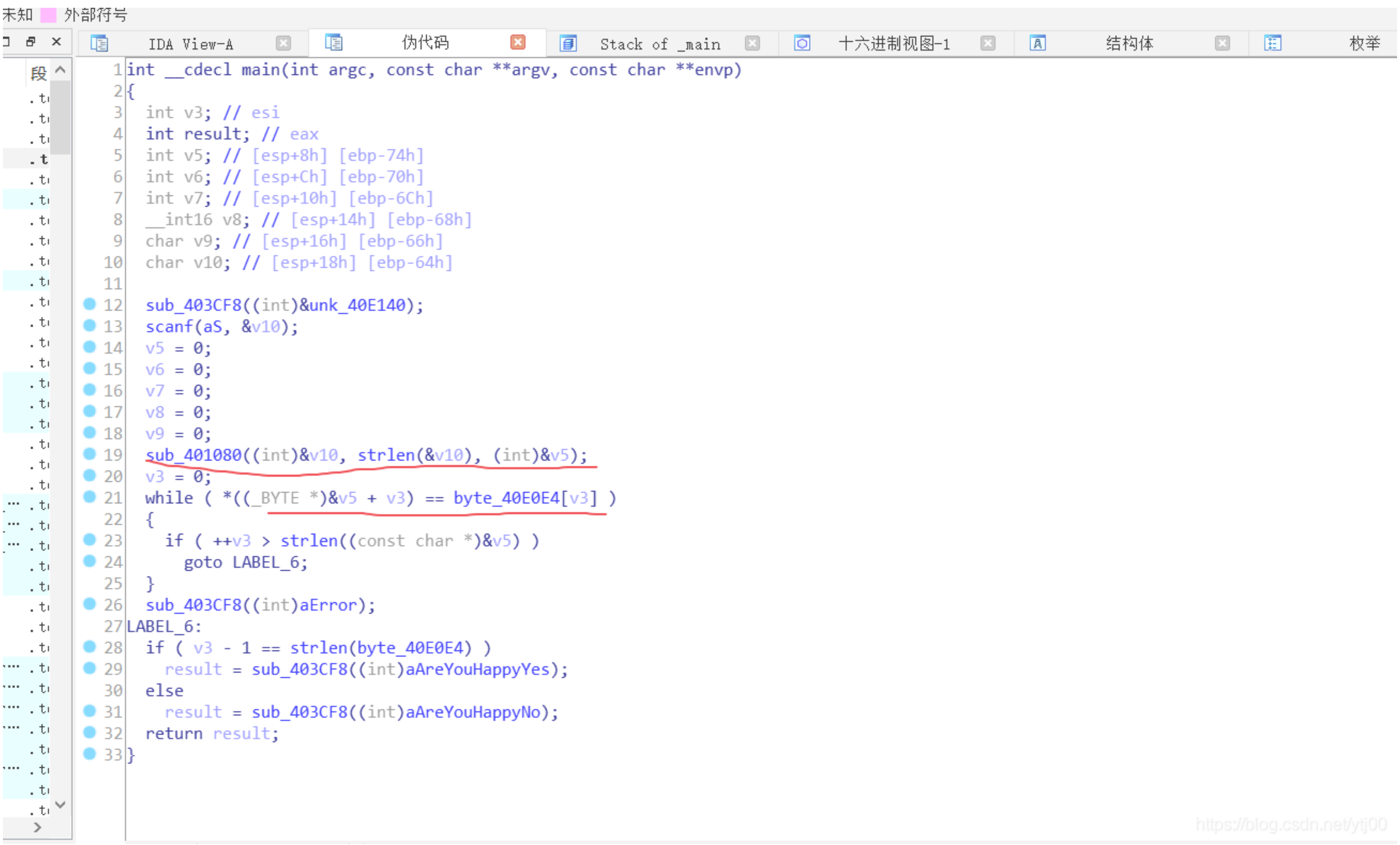
原创

ytj00 于 2020-07-30 17:19:01 发布 284 收藏
分类专栏： ctf 逆向


ctf 同时被 2 个专栏收录
28 篇文章 0 订阅
订阅专栏


逆向
27 篇文章 0 订阅
订阅专栏

ida打开，进入main函数



有一个关键函数，和一个关键比较

先进入关键函数里面

```
  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)];
```



```
    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);
    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((const char *)a3);
}
00001105 sub_401080:38 (401105)
```


最开始有一个sub_401000()函数，进入后

```
igned int sub_401000()

 signed int result; // eax
 char v1; // cl

 result = 6;
 do
 {
   v1 = unk_40E0AA[result];
   unk_40E0AA[result] = byte_40E0A0[result];
   byte_40E0A0[result++] = v1;
 }
 while ( result < 15 );
 return result;
```



```
.data:0040E0A0 aAbcdefghij     db 'ABCDEFGHIJ'         ; DATA XREF: sub_401000:loc_401005↑r
.data:0040E0A0                                         ; sub_401000+17↑w ...
.data:0040E0AA ; char aKlmnopqrstuvwx[]
.data:0040E0AA aKlmnopqrstuvwx db 'KLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
.data:0040E0AA                                         ; DATA XREF: sub_401000+B↑r
.data:0040E0AA                                         ; sub_401000+11↑w
```

感觉有点像base64加密，然后这个函数是变换base64的码表，写脚本将变换后的码表求出来

```c
#include <stdio.h>
int main()
{

char b[]="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/";
char c;
int i=6;
printf("sss\n");
do
{
 c=b[i+10];
 b[i+10]=b[i];
 b[i++]=c;
}
while(i<15);

printf("%s",b);

}
```

得到码表：ABCDEFQRSTUVWXYPGHIJKLMNOZabcdefghijklmnopqrstuvwxyz0123456789+/

然后注意到返回时的函数sub_401030()

```
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 < 'a' || v2 > 'z' )
      {
        if ( v2 < 'A' || v2 > 'Z' )
          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;
}
```


很明显是一个大小写互换的函数

所以我们可以通过main函数里的判断条件来逆推flag

```
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);
ABEL_6:
if ( v3 - 1 == strlen(byte_40E0E4) )
```



```
.data:0040E0E4 byte_40E0E4     db 'z'                  ; DATA XREF: _main+5C↑r
.data:0040E0E4                                         ; _main:loc_401238↑o
.data:0040E0E5 aMxhz3tignxlxjh db 'MXHz3TIgnxLxJhFAdtZn2fFk3lYCrtPC2l9',0
.data:0040E109                 align 4
```

比较数据为：zMXHz3TIgnxLxJhFAdtZn2fFk3lYCrtPC2l9

先转换为大写：ZmxhZ3tiGNXlXjHfaDTzN2FfK3LycRTpc2L9

然后根据之前的码表解密

```
================== RESTART: D:\常用脚本py\密码学+进制\base64解密.py ===========
========

     ************************************
     *    (1)encode          (2)decode    *
     ************************************

Please select the operation you want to perform:
2
Please enter a string that needs to be decrypted:
ZmxhZ3tiGNXlXjHfaDTzN2FfK3LycRTpc2L9
Decrypted String:
flag{bAse64_h2s_a_Surprise}
>>>
```


flag为：flag{bAse64_h2s_a_Surprise}