# buuctf刷题记录19 buuctf FindKey

原创

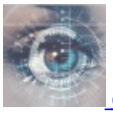
ytj00 于 2020-08-01 18:30:20 发布 407 收藏

分类专栏： ctf 逆向



本文链接：https://blog.csdn.net/ytj00/article/details/107734901





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

个人感觉这个题出的不好，md5根本找不到（是我太菜）

经典无壳，拖进ida，f12找到关键字符串，发现左边地址是红色的，应该是用了花指令，果然

```
.text:00401910                 push    eax
.text:00401910                 call    _memcpy
.text:00401915                 add     esp, 0Ch
.text:00401918                 push    offset byte_428C54
.text:0040191D
.text:0040191D loc_40191D:                             ; CODE XREF: .text:0040193D↓j
.text:0040191D                 push    offset byte_428C54
.text:00401922                 call    _strlen
.text:00401927                 add     esp, 4
.text:0040192A                 push    eax
.text:0040192B                 push    offset byte_428C54
```

两个一样的指令连在一起，把下面的去掉，按p声明函数，再f5看反编译

```
    if ( strlen((const char *)pbData) > 6 )
      ExitProcess(0);
    if ( strlen((const char *)pbData) )
    {
      memset(&v24, 0, 0x100u);
      v6 = strlen((const char *)pbData);
      memcpy(&v24, pbData, v6);
      v10 = (const char *)pbData;
      do
      {
        v7 = strlen(v10);
        sub_40101E(pbData, v7, v11);
      }
      while ( &v12 && !&v12 );
      strcpy(&v20, "0kk`d1a`55k222k2a776jbfgd`06cjjb");
      memset(&v21, 0, 220u);
      v22 = 0;
      v23 = 0;
      strcpy(v16, "SS");
      v17 = 0;
      v18 = 0;
      v19 = 0;
      v8 = strlen(&v20);
      sub_401005(v16, (int)&v20, v8);          // 关键运算
      if ( _strcmpi((const char *)pbData, &v20) )// 判断条件
      {
        SetWindowTextA(hWndParent, "flag{}");
        MessageBoxA(hWndParent, "Are you kidding me?", "^_^", 0);
        ExitProcess(0);
      }
      memcpy(&v15, &unk_423030, 50u);
      v9 = strlen(&v15);
      sub_401005(&v24, (int)&v15, v9);         // 关键运算
      MessageBoxA(hWndParent, &v15, 0, 0x32u);
    }
    ++dword_428D54;
000019BB sub_401640:72 (4019BB)
```



程序逻辑并不是太难，有两处关键运算和一个关键判断

第一个关键运算是将输入的pbdata与v16进行异或，最后与v20进行比较

```
unsigned int __cdecl sub_401590(LPCSTR lpString, int a2, int a3)
{
  unsigned int result; // eax
  unsigned int i; // [esp+4Ch] [ebp-Ch]
  unsigned int v5; // [esp+54h] [ebp-4h]

  v5 = lstrlenA(lpString);
  for ( i = 0; ; ++i )
  {
    result = i;
    if ( i >= a3 )
      break;
    *(_BYTE *)(i + a2) ^= lpString[i % v5];
  }
  return result;
}
```



逻辑很简单，直接写脚本，得到第一个字符串：c8837b23ff8aaa8a2dde915473ce0991

然后如果直接按照第二个关键运算算的话，算不出来

看别人的wp说，根据库函数可以看出这里经过md5加密（就这步咋也找不到，真不知道这个人是咋找的）

链接

[]:
https://www.dongzt.cn/archives/2019%E5%B9%B43%E6%9C%88%E5%AE%89%E6%81%92%E5%B9%B3%E5%8F%B0%E5%91%A8%E5%91%A8%E7%BB%83%E7%9A%84%E5%81%9A%E9%A2%98%E6%80%9D%E8%B7%AF%E5%88%86%E4%BA%AB.html#0x02findkey

然后把这个字符串去md5解密网站上找一哈，得到：123321

然后再进行第二个关键运算，跟第一个基本一样，只不过是换了换参数，

脚本

```
#include <stdio.h>
#include<string.h>
int main()
{
int a[] =
{
   48, 107, 107,  96, 100,  49,  97,  96,  53,  53,
  107,  50,  50,  50, 107,  50,  97,  55,  55,  54,
  106,  98, 102, 103, 100,  96,  48,  54,  99, 106,
  106,  98
};
int b[]={83,83};
int c[]=
{
   87,  94,  82,  84,  73,  95,   1, 109, 105,  70,
    2, 110,  95,   2, 108,  87,  91,  84,  76
};
int d[]={49,50,51,51,50,49};
int i;
for (i=0;i<32;i++)
{
 a[i]^=b[i%2];
}
for(i=0;i<32;i++)
{
 printf("%c",a[i]);
}
printf("\n");
for (i=0;i<19;i++)
{
 c[i]^=d[i%6];
}
for(i=0;i<19;i++)
{
 printf("%c",c[i]);
}
}
```

得到flag： flag{n0_Zu0_n0_die}