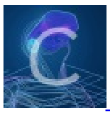
[看雪CTF]2019晋级赛Q1第一题流浪者

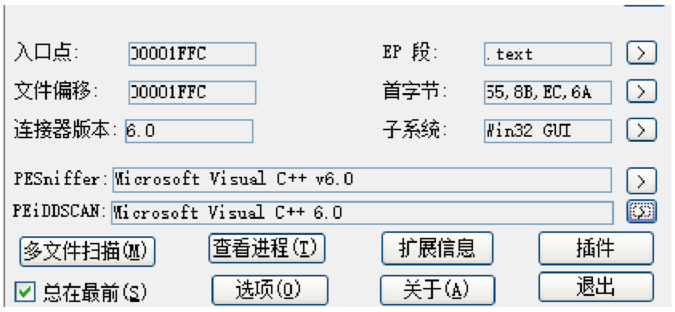
原创

xuenixiang 于 2019-03-27 15:41:46 发布 425 收藏
分类专栏： CTF 文章标签： CTF reverse

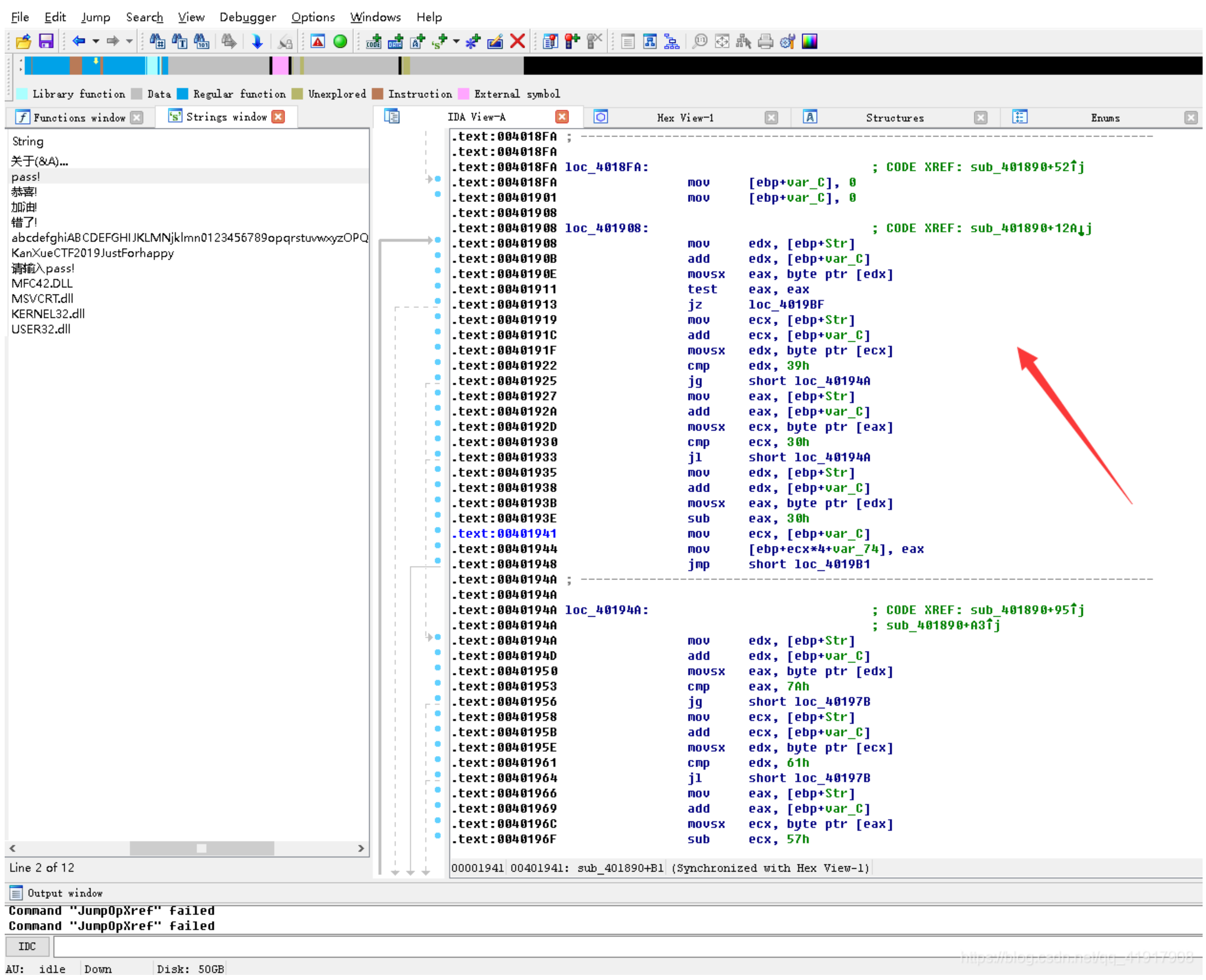

CTF 专栏收录该内容
2 篇文章 0 订阅
订阅专栏

一直在忙，晚上抽空打开题目，看到大佬们300多秒就解出来了，我只能写个详细点的wp来混存在感了。

首先查壳，VC程序，没壳


入口点: 00001FFC　EP 段: .text
文件偏移: 00001FFC　首字节: 55,8B,EC,6A
连接器版本: 6.0　子系统: Win32 GUI
PESniffer: Microsoft Visual C++ v6.0
PEiDDSCAN: Microsoft Visual C++ 6.0

直接导入IDA定位到关键算法部分

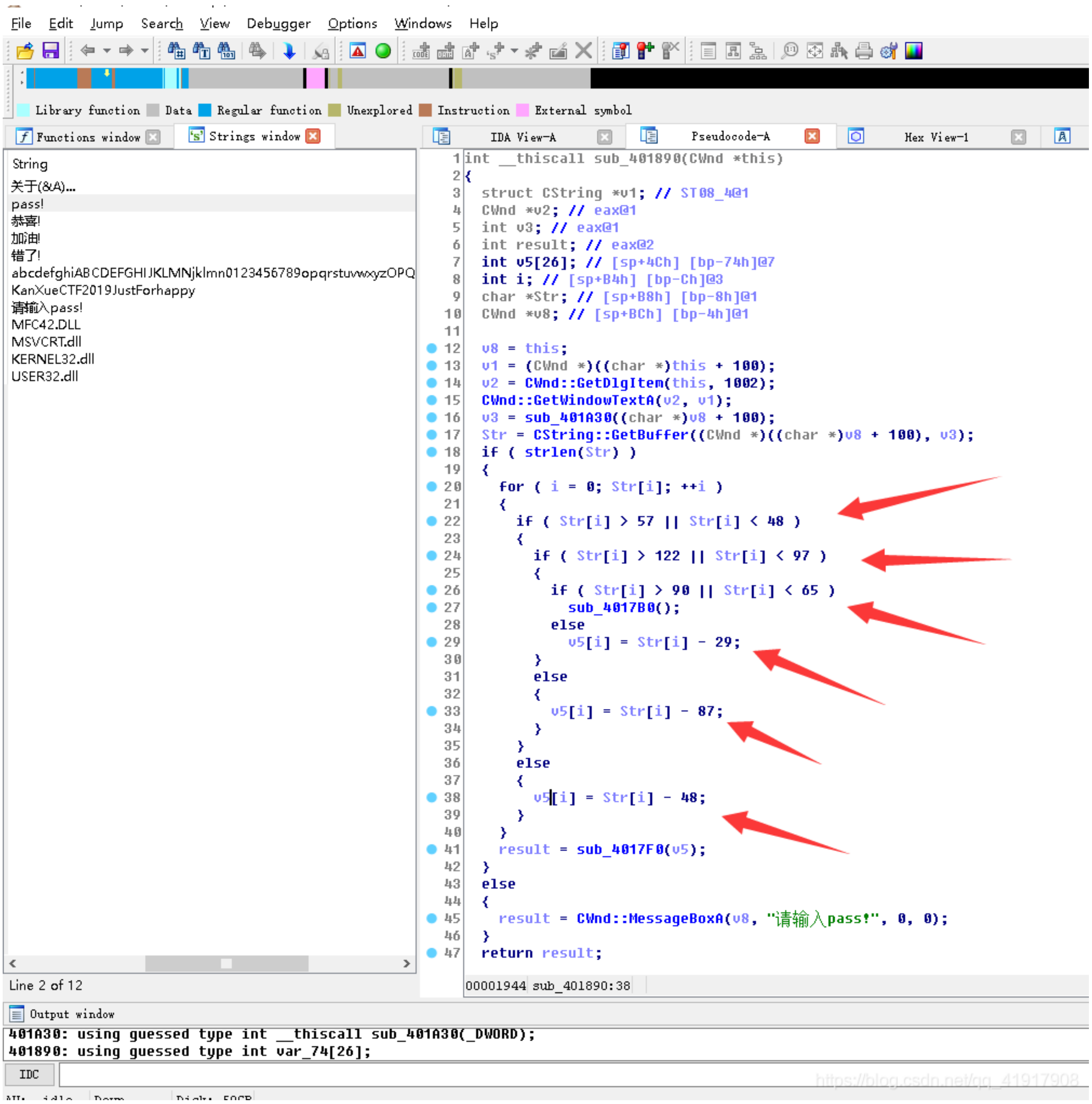

File Edit Jump Search View Debugger Options Windows Help
Library function Data Regular function Unexplored Instruction External symbol
Functions window
Strings window
IDA View-A
Hex View-1
Structures
Enums

```
String
关于(&A)...
pass!
恭喜!
加油!
错了!
abcdefghiABCDEFGHIJKLMNjklmn0123456789opqrstuvwxyzOPQ
KanXueCTF2019JustForhappy
请输入pass!
MFC42.DLL
MSVCRT.dll
KERNEL32.dll
USER32.dll
Line 2 of 12
```

```
.text:004018FA ; ---------------------------------------------------------------------------
.text:004018FA
.text:004018FA loc_4018FA:                             ; CODE XREF: sub_401890+52↑j
.text:004018FA                 mov     [ebp+var_C], 0
.text:00401901                 mov     [ebp+var_C], 0
.text:00401908
.text:00401908 loc_401908:                             ; CODE XREF: sub_401890+12A↓j
.text:00401908                 mov     edx, [ebp+Str]
.text:0040190B                 add     edx, [ebp+var_C]
.text:0040190E                 movsx   eax, byte ptr [edx]
.text:00401911                 test    eax, eax
.text:00401913                 jz      loc_4019BF
.text:00401919                 mov     ecx, [ebp+Str]
.text:0040191C                 add     ecx, [ebp+var_C]
.text:0040191F                 movsx   edx, byte ptr [ecx]
.text:00401922                 cmp     edx, 39h
.text:00401925                 jg      short loc_40194A
.text:00401927                 mov     eax, [ebp+Str]
.text:0040192A                 add     eax, [ebp+var_C]
.text:0040192D                 movsx   ecx, byte ptr [eax]
.text:00401930                 cmp     ecx, 30h
.text:00401933                 jl      short loc_40194A
.text:00401935                 mov     edx, [ebp+Str]
.text:00401938                 add     edx, [ebp+var_C]
.text:0040193B                 movsx   eax, byte ptr [edx]
.text:0040193E                 sub     eax, 30h
.text:00401941                 mov     ecx, [ebp+var_C]
.text:00401944                 mov     [ebp+ecx*4+var_74], eax
.text:00401948                 jmp     short loc_4019B1
.text:0040194A ; ---------------------------------------------------------------------------
.text:0040194A
.text:0040194A loc_40194A:                             ; CODE XREF: sub_401890+95↑j
.text:0040194A                                         ; sub_401890+A3↑j
.text:0040194A                 mov     edx, [ebp+Str]
.text:0040194D                 add     edx, [ebp+var_C]
.text:00401950                 movsx   eax, byte ptr [edx]
.text:00401953                 cmp     eax, 7Ah
.text:00401956                 jg      short loc_40197B
.text:00401958                 mov     ecx, [ebp+Str]
.text:0040195B                 add     ecx, [ebp+var_C]
.text:0040195E                 movsx   edx, byte ptr [ecx]
.text:00401961                 cmp     edx, 61h
.text:00401964                 jl      short loc_40197B
.text:00401966                 mov     eax, [ebp+Str]
.text:00401969                 add     eax, [ebp+var_C]
.text:0040196C                 movsx   ecx, byte ptr [eax]
.text:0040196F                 sub     ecx, 57h
00001941 00401941: sub_401890+B1 (Synchronized with Hex View-1)
```

```
Output window
Command "JumpOpXref" failed
Command "JumpOpXref" failed
IDC
AU:  idle   Down    Disk: 50GB
```

通过F5可以看到伪C代码，主要讲的是我们输入的假码如果在0-9之间，就把对应的ASCII减0x30，假码如果在a-z之间，就把对应的ASCII减0x57，假码如果在A-Z之间，就把对应的ASCII减0x1D



```
String
关于(&A)...
pass!
恭喜!
加油!
错了!
abcdefghiABCDEFGHIJKLMNjklmn0123456789opqrstuvwxyzOPQ
KanXueCTF2019JustForhappy
请输入pass!
MFC42.DLL
MSVCRT.dll
KERNEL32.dll
USER32.dll
```

Line 2 of 12

```
int __thiscall sub_401890(CWnd *this)
{
  struct CString *v1; // ST08_4@1
  CWnd *v2; // eax@1
  int v3; // eax@1
  int result; // eax@2
  int v5[26]; // [sp+4Ch] [bp-74h]@7
  int i; // [sp+B4h] [bp-Ch]@3
  char *Str; // [sp+B8h] [bp-8h]@1
  CWnd *v8; // [sp+BCh] [bp-4h]@1

  v8 = this;
  v1 = (CWnd *)((char *)this + 100);
  v2 = CWnd::GetDlgItem(this, 1002);
  CWnd::GetWindowTextA(v2, v1);
  v3 = sub_401A30((char *)v8 + 100);
  Str = CString::GetBuffer((CWnd *)((char *)v8 + 100), v3);
  if ( strlen(Str) )
  {
    for ( i = 0; Str[i]; ++i )
    {
      if ( Str[i] > 57 || Str[i] < 48 )
      {
        if ( Str[i] > 122 || Str[i] < 97 )
        {
          if ( Str[i] > 90 || Str[i] < 65 )
            sub_4017B0();
          else
            v5[i] = Str[i] - 29;
        }
        else
        {
          v5[i] = Str[i] - 87;
        }
      }
      else
      {
        v5[i] = Str[i] - 48;
      }
    }
    result = sub_4017F0(v5);
  }
  else
  {
    result = CWnd::MessageBoxA(v8, "请输入pass!", 0, 0);
  }
  return result;
}
```

00001944 sub_401890:38

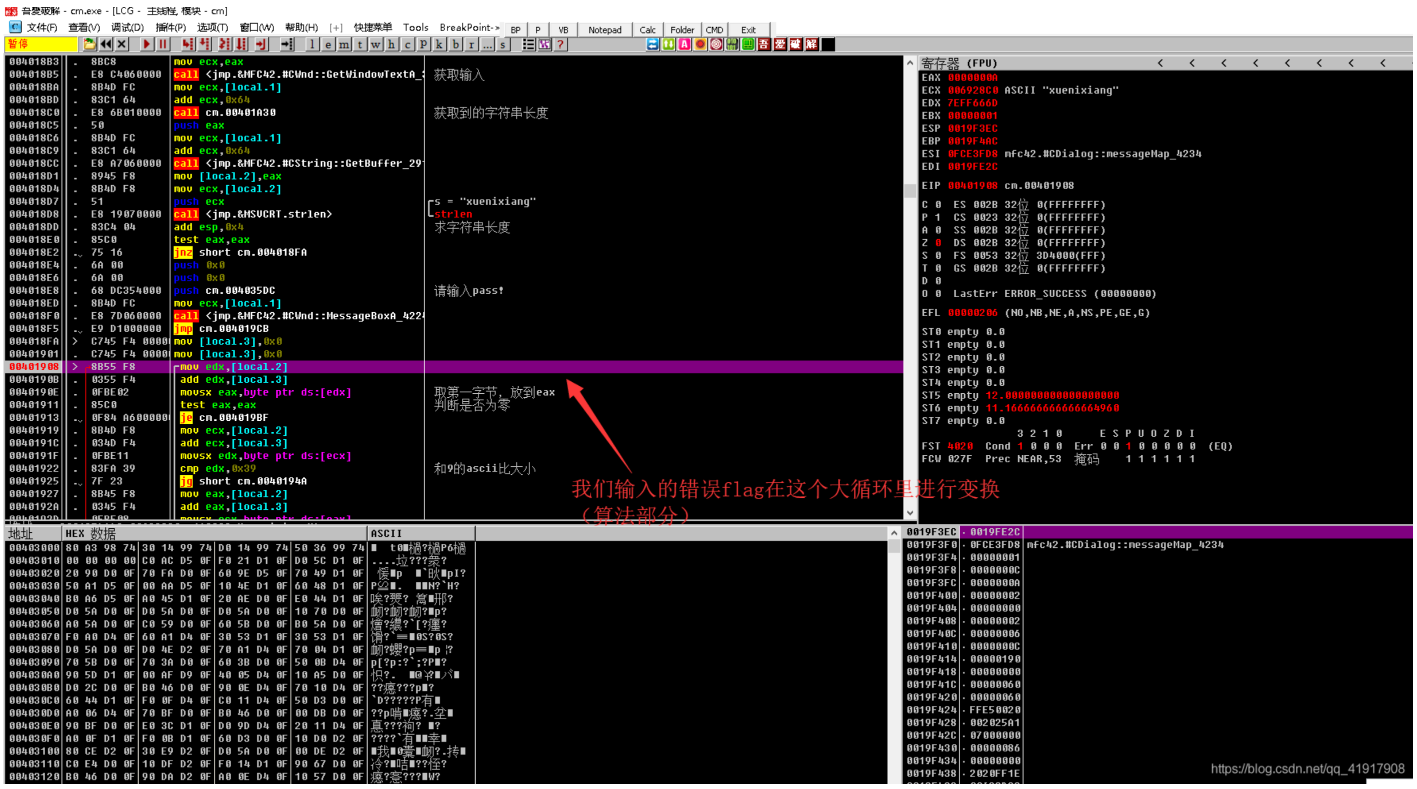
```
401A30: using guessed type int __thiscall sub_401A30(_DWORD);
401890: using guessed type int var_74[26];
```

大佬看到这里就可以关掉IDA开始写脚本了，但是我还是决定用OD来找找捷径

首先载入OD定位到关键点

004018B5处的GetWindowsTextA函数获取输入

00401908这里就开始进行算法变换


吾爱破解 - cm.exe - [LCG - 主线程, 模块 - cm]
文件(F) 查看(V) 调试(D) 插件(P) 选项(T) 窗口(W) 帮助(H) [+] 快捷菜单 Tools BreakPoint-> BP P VB Notepad Calc Folder CMD Exit
暂停
l e m t w h c P k b r ... s

```
004018B3  .  8BC8          mov ecx,eax
004018B5  .  E8 C4060000   call <jmp.&MFC42.#CWnd::GetWindowTextA_   获取输入
004018BA  .  8B4D FC       mov ecx,[local.1]
004018BD  .  83C1 64       add ecx,0x64
004018C0  .  E8 6B010000   call cm.00401A30                          获取到的字符串长度
004018C5  .  50            push eax
004018C6  .  8B4D FC       mov ecx,[local.1]
004018C9  .  83C1 64       add ecx,0x64
004018CC  .  E8 A7060000   call <jmp.&MFC42.#CString::GetBuffer_29
004018D1  .  8945 F8       mov [local.2],eax
004018D4  .  8B4D F8       mov ecx,[local.2]
004018D7  .  51            push ecx                                  ┌s = "xuenixiang"
004018D8  .  E8 19070000   call <jmp.&MSVCRT.strlen>                 └strlen
004018DD  .  83C4 04       add esp,0x4                               求字符串长度
004018E0  .  85C0          test eax,eax
004018E2  .v 75 16         jnz short cm.004018FA
004018E4  .  6A 00         push 0x0
004018E6  .  6A 00         push 0x0
004018E8  .  68 DC354000   push cm.004035DC                          请输入pass!
004018ED  .  8B4D FC       mov ecx,[local.1]
004018F0  .  E8 7D060000   call <jmp.&MFC42.#CWnd::MessageBoxA_422
004018F5  .v E9 D1000000   jmp cm.004019CB
004018FA  >  C745 F4 0000  mov [local.3],0x0
00401901  .  C745 F4 0000  mov [local.3],0x0
00401908  >  8B55 F8       mov edx,[local.2]
0040190B  .  0355 F4       add edx,[local.3]
0040190E  .  0FBE02        movsx eax,byte ptr ds:[edx]               取第一字节，放到eax
00401911  .  85C0          test eax,eax                              判断是否为零
00401913  .v 0F84 A6000000 je cm.004019BF
00401919  .  8B4D F8       mov ecx,[local.2]
0040191C  .  034D F4       add ecx,[local.3]
0040191F  .  0FBE11        movsx edx,byte ptr ds:[ecx]
00401922  .  83FA 39       cmp edx,0x39                              和9的ascii比大小
00401925  .v 7F 23         jg short cm.0040194A
00401927  .  8B45 F8       mov eax,[local.2]
0040192A  .  0345 F4       add eax,[local.3]
```

我们输入的错误flag在这个大循环里进行变换
（算法部分）

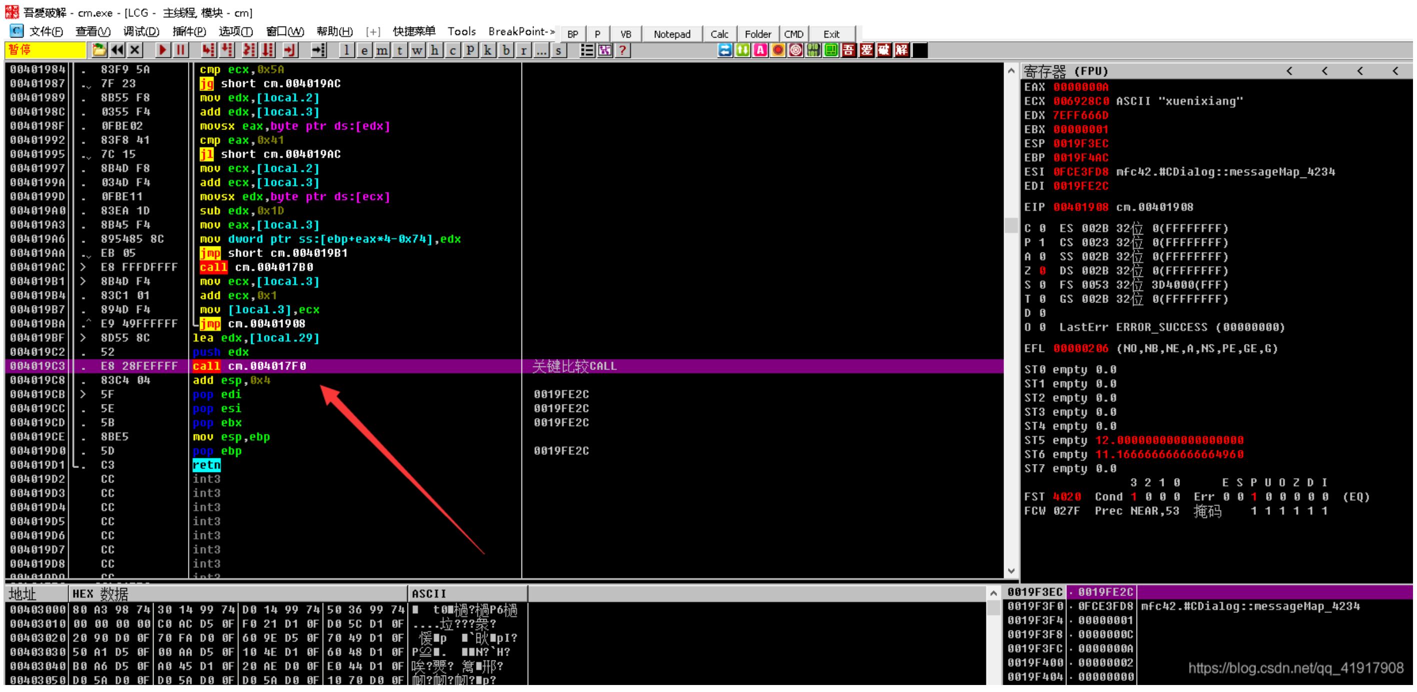
```
寄存器 (FPU)
EAX 0000000A
ECX 006928C0 ASCII "xuenixiang"
EDX 7EFF666D
EBX 00000001
ESP 0019F3EC
EBP 0019F4AC
ESI 0FCE3FD8 mfc42.#CDialog::messageMap_4234
EDI 0019FE2C

EIP 00401908 cm.00401908

C 0  ES 002B 32位 0(FFFFFFFF)
P 1  CS 0023 32位 0(FFFFFFFF)
A 0  SS 002B 32位 0(FFFFFFFF)
Z 0  DS 002B 32位 0(FFFFFFFF)
S 0  FS 0053 32位 3D4000(FFF)
T 0  GS 002B 32位 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000206 (NO,NB,NE,A,NS,PE,GE,G)

ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 12.000000000000000000
ST6 empty 11.166666666666664960
ST7 empty 0.0
               3 2 1 0      E S P U O Z D I
FST 4020  Cond 1 0 0 0  Err 0 0 1 0 0 0 0 0  (EQ)
FCW 027F  Prec NEAR,53  掩码    1 1 1 1 1 1
```

```
地址      HEX 数据                                            ASCII
00403000 80 A3 98 74 30 14 99 74 D0 14 99 74 50 36 99 74
00403010 00 00 00 00 C0 AC D5 0F F0 21 D1 0F D0 5C D1 0F
00403020 20 90 D0 0F 70 FA D0 0F 60 9E D5 0F 70 49 D1 0F
00403030 50 A1 D5 0F 00 AA D5 0F 10 4E D1 0F 60 48 D1 0F
00403040 B0 A6 D5 0F A0 45 D1 0F 20 AE D0 0F E0 44 D1 0F
00403050 D0 5A D0 0F D0 5A D0 0F D0 5A D0 0F 10 70 D0 0F
00403060 A0 5A D0 0F C0 59 D0 0F 60 5B D0 0F B0 5A D0 0F
00403070 F0 A0 D4 0F 60 A1 D4 0F 30 53 D1 0F 30 53 D1 0F
00403080 D0 5A D0 0F D0 4E D2 0F 70 A1 D4 0F 70 04 D1 0F
00403090 70 5B D0 0F 70 3A D0 0F 60 3B D0 0F 50 0B D4 0F
004030A0 90 5D D1 0F 00 AF D9 0F 40 05 D4 0F 10 A5 D0 0F
004030B0 D0 2C D0 0F B0 46 D0 0F 90 0E D4 0F 70 10 D4 0F
004030C0 60 44 D1 0F F0 0F D4 0F C0 11 D4 0F 50 D3 D0 0F
004030D0 A0 06 D4 0F 70 BF D0 0F B0 46 D0 0F 00 DB D0 0F
004030E0 90 BF D0 0F E0 3C D1 0F D0 9D D4 0F 20 11 D4 0F
004030F0 A0 0F D1 0F F0 0B D1 0F 60 D3 D0 0F 10 D0 D2 0F
00403100 80 CE D2 0F 30 E9 D2 0F D0 5A D0 0F 00 DE D2 0F
00403110 C0 E4 D0 0F 10 DF D2 0F F0 14 D1 0F 90 67 D0 0F
00403120 B0 46 D0 0F 90 DA D2 0F A0 0E D4 0F 10 57 D0 0F
```

```
0019F3EC  0019FE2C
0019F3F0  0FCE3FD8  mfc42.#CDialog::messageMap_4234
0019F3F4  00000001
0019F3F8  0000000C
0019F3FC  0000000A
0019F400  00000002
0019F404  00000000
0019F408  00000002
0019F40C  00000006
0019F410  0000000C
0019F414  00000190
0019F418  00000000
0019F41C  00000060
0019F420  00000060
0019F424  FFE50020
0019F428  002025A1
0019F42C  07000000
0019F430  00000086
0019F434  00000000
0019F438  2020FF1E
```

https://blog.csdn.net/qq_41917908

出了大循环之后，004019C3这里是关键比较CALL


吾爱破解 - cm.exe - [LCG - 主线程, 模块 - cm]
文件(F) 查看(V) 调试(D) 插件(P) 选项(T) 窗口(W) 帮助(H) [+] 快捷菜单 Tools BreakPoint-> BP P VB Notepad Calc Folder CMD Exit
暂停
l e m t w h c P k b r ... s

```
00401984  .  83F9 5A       cmp ecx,0x5A
00401987  .v 7F 23         jg short cm.004019AC
00401989  .  8B55 F8       mov edx,[local.2]
0040198C  .  0355 F4       add edx,[local.3]
0040198F  .  0FBE02        movsx eax,byte ptr ds:[edx]
00401992  .  83F8 41       cmp eax,0x41
00401995  .v 7C 15         jl short cm.004019AC
00401997  .  8B4D F8       mov ecx,[local.2]
0040199A  .  034D F4       add ecx,[local.3]
0040199D  .  0FBE11        movsx edx,byte ptr ds:[ecx]
004019A0  .  83EA 1D       sub edx,0x1D
004019A3  .  8B45 F4       mov eax,[local.3]
004019A6  .  895485 8C     mov dword ptr ss:[ebp+eax*4-0x74],edx
004019AA  .v EB 05         jmp short cm.004019B1
004019AC  >  E8 FFFDFFFF   call cm.004017B0
004019B1  >  8B4D F4       mov ecx,[local.3]
004019B4  .  83C1 01       add ecx,0x1
004019B7  .  894D F4       mov [local.3],ecx
004019BA  .^ E9 49FFFFFF   jmp cm.00401908
004019BF  >  8D55 8C       lea edx,[local.29]
004019C2  .  52            push edx
004019C3  .  E8 28FEFFFF   call cm.004017F0                          关键比较CALL
004019C8  .  83C4 04       add esp,0x4
004019CB  >  5F            pop edi                                   0019FE2C
004019CC  .  5E            pop esi                                   0019FE2C
004019CD  .  5B            pop ebx                                   0019FE2C
004019CE  .  8BE5          mov esp,ebp
004019D0  .  5D            pop ebp                                   0019FE2C
004019D1  .  C3            retn
004019D2     CC            int3
004019D3     CC            int3
004019D4     CC            int3
004019D5     CC            int3
004019D6     CC            int3
004019D7     CC            int3
004019D8     CC            int3
```

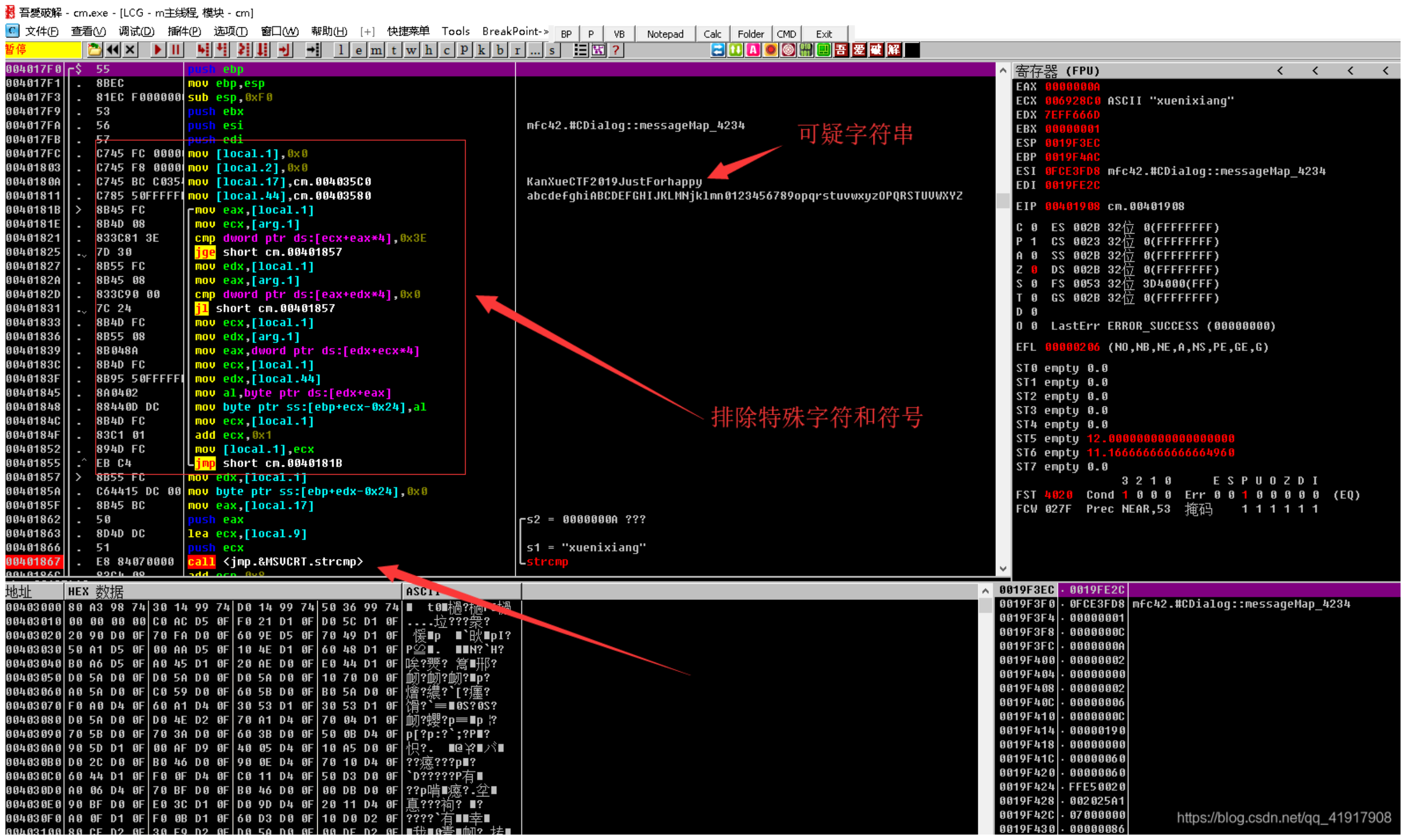
```
寄存器 (FPU)
EAX 0000000A
ECX 006928C0 ASCII "xuenixiang"
EDX 7EFF666D
EBX 00000001
ESP 0019F3EC
EBP 0019F4AC
ESI 0FCE3FD8 mfc42.#CDialog::messageMap_4234
EDI 0019FE2C

EIP 00401908 cm.00401908

C 0  ES 002B 32位 0(FFFFFFFF)
P 1  CS 0023 32位 0(FFFFFFFF)
A 0  SS 002B 32位 0(FFFFFFFF)
Z 0  DS 002B 32位 0(FFFFFFFF)
S 0  FS 0053 32位 3D4000(FFF)
T 0  GS 002B 32位 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000206 (NO,NB,NE,A,NS,PE,GE,G)

ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 12.000000000000000000
ST6 empty 11.166666666666664960
ST7 empty 0.0
               3 2 1 0      E S P U O Z D I
FST 4020  Cond 1 0 0 0  Err 0 0 1 0 0 0 0 0  (EQ)
FCW 027F  Prec NEAR,53  掩码    1 1 1 1 1 1
```

```
地址      HEX 数据                                            ASCII
00403000 80 A3 98 74 30 14 99 74 D0 14 99 74 50 36 99 74
00403010 00 00 00 00 C0 AC D5 0F F0 21 D1 0F D0 5C D1 0F
00403020 20 90 D0 0F 70 FA D0 0F 60 9E D5 0F 70 49 D1 0F
00403030 50 A1 D5 0F 00 AA D5 0F 10 4E D1 0F 60 48 D1 0F
00403040 B0 A6 D5 0F A0 45 D1 0F 20 AE D0 0F E0 44 D1 0F
00403050 D0 5A D0 0F D0 5A D0 0F D0 5A D0 0F 10 70 D0 0F
```

```
0019F3EC  0019FE2C
0019F3F0  0FCE3FD8  mfc42.#CDialog::messageMap_4234
0019F3F4  00000001
0019F3F8  0000000C
0019F3FC  0000000A
0019F400  00000002
0019F404  00000000
```

https://blog.csdn.net/qq_41917908

进CALL看看


吾爱破解 - cm.exe - [LCG - m主线程, 模块 - cm]
文件(F) 查看(V) 调试(D) 插件(P) 选项(T) 窗口(W) 帮助(H) [+] 快捷菜单 Tools BreakPoint-> BP P VB Notepad Calc Folder CMD Exit
暂停
可疑字符串
排除特殊字符和符号

```
004017F0  ┌$  55             push ebp
004017F1  .   8BEC           mov ebp,esp
004017F3  .   81EC F0000000  sub esp,0xF0
004017F9  .   53             push ebx
004017FA  .   56             push esi                                   mfc42.#CDialog::messageMap_4234
004017FB  .   57             push edi
004017FC  .   C745 FC 00000  mov [local.1],0x0
00401803  .   C745 F8 00000  mov [local.2],0x0
0040180A  .   C745 BC C035   mov [local.17],cm.004035C0                 KanXueCTF2019JustForhappy
00401811  .   C785 50FFFFF   mov [local.44],cm.00403580                 abcdefghiABCDEFGHIJKLMNjklmn0123456789opqrstuvwxyzOPQRSTUVWXYZ
0040181B  >   8B45 FC        ┌mov eax,[local.1]
0040181E  .   8B4D 08        │mov ecx,[arg.1]
00401821  .   833C81 3E      │cmp dword ptr ds:[ecx+eax*4],0x3E
00401825  .v  7D 30          │jge short cm.00401857
00401827  .   8B55 FC        │mov edx,[local.1]
0040182A  .   8B45 08        │mov eax,[arg.1]
0040182D  .   833C90 00      │cmp dword ptr ds:[eax+edx*4],0x0
00401831  .v  7C 24          │jl short cm.00401857
00401833  .   8B4D FC        │mov ecx,[local.1]
00401836  .   8B55 08        │mov edx,[arg.1]
00401839  .   8B048A         │mov eax,dword ptr ds:[edx+ecx*4]
0040183C  .   8B4D FC        │mov ecx,[local.1]
0040183F  .   8B95 50FFFFF   │mov edx,[local.44]
00401845  .   8A0402         │mov al,byte ptr ds:[edx+eax]
00401848  .   88440D DC      │mov byte ptr ss:[ebp+ecx-0x24],al
0040184C  .   8B4D FC        │mov ecx,[local.1]
0040184F  .   83C1 01        │add ecx,0x1
00401852  .   894D FC        │mov [local.1],ecx
00401855  .^  EB C4          └jmp short cm.0040181B
00401857  >   8B55 FC        mov edx,[local.1]
0040185A  .   C64415 DC 00   mov byte ptr ss:[ebp+edx-0x24],0x0
0040185F  .   8B45 BC        mov eax,[local.17]
00401862  .   50             push eax                                   ┌s2 = 0000000A ???
00401863  .   8D4D DC        lea ecx,[local.9]                          │
00401866  .   51             push ecx                                   │s1 = "xuenixiang"
00401867  .   E8 84070000    call <jmp.&MSVCRT.strcmp>                  └strcmp
```

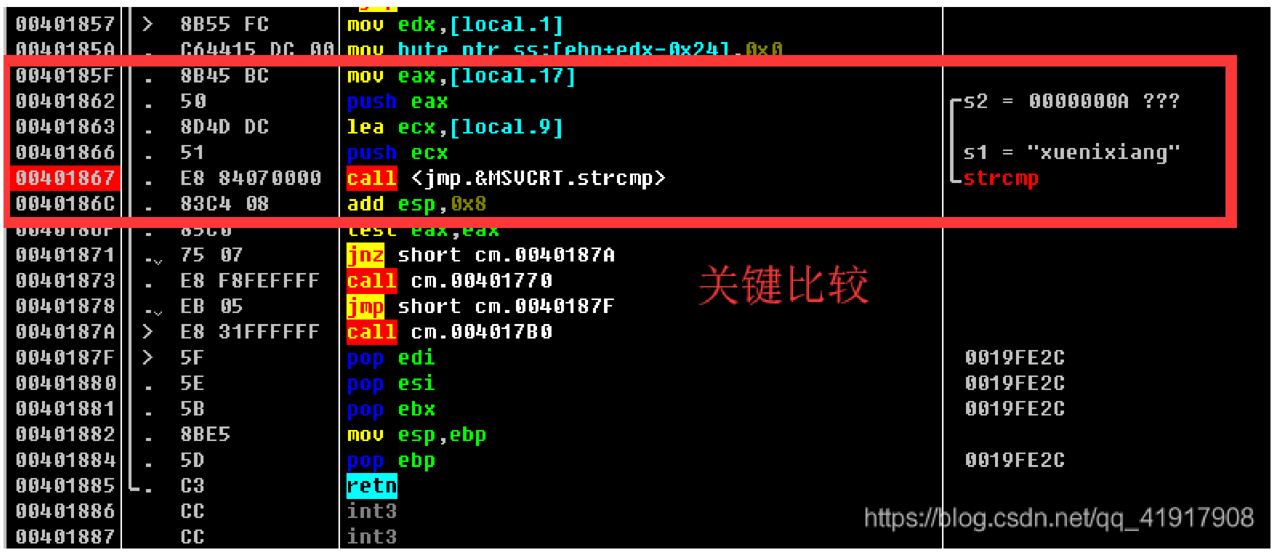
```
寄存器 (FPU)
EAX 0000000A
ECX 006928C0 ASCII "xuenixiang"
EDX 7EFF666D
EBX 00000001
ESP 0019F3EC
EBP 0019F4AC
ESI 0FCE3FD8 mfc42.#CDialog::messageMap_4234
EDI 0019FE2C

EIP 00401908 cm.00401908

C 0  ES 002B 32位 0(FFFFFFFF)
P 1  CS 0023 32位 0(FFFFFFFF)
A 0  SS 002B 32位 0(FFFFFFFF)
Z 0  DS 002B 32位 0(FFFFFFFF)
S 0  FS 0053 32位 3D4000(FFF)
T 0  GS 002B 32位 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000206 (NO,NB,NE,A,NS,PE,GE,G)

ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 12.000000000000000000
ST6 empty 11.166666666666664960
ST7 empty 0.0
               3 2 1 0      E S P U O Z D I
FST 4020  Cond 1 0 0 0  Err 0 0 1 0 0 0 0 0  (EQ)
FCW 027F  Prec NEAR,53  掩码    1 1 1 1 1 1
```

```
地址      HEX 数据                                             ASCII
00403000  80 A3 98 74  30 14 99 74  D0 14 99 74  50 36 99 74
00403010  00 00 00 00  C0 AC D5 0F  F0 21 D1 0F  D0 5C D1 0F
00403020  20 90 D0 0F  70 FA D0 0F  60 9E D5 0F  70 49 D1 0F
00403030  50 A1 D5 0F  00 AA D5 0F  10 4E D1 0F  60 48 D1 0F
00403040  B0 A6 D5 0F  A0 45 D1 0F  20 AE D0 0F  E0 44 D1 0F
00403050  D0 5A D0 0F  D0 5A D0 0F  D0 5A D0 0F  10 70 D0 0F
00403060  A0 5A D0 0F  C0 59 D0 0F  60 5B D0 0F  B0 5A D0 0F
00403070  F0 A0 D4 0F  60 A1 D4 0F  30 53 D1 0F  30 53 D1 0F
00403080  D0 5A D0 0F  D0 4E D2 0F  70 A1 D4 0F  70 04 D1 0F
00403090  70 5B D0 0F  70 3A D0 0F  60 3B D0 0F  50 0B D4 0F
004030A0  90 5D D1 0F  00 AF D9 0F  40 05 D4 0F  10 A5 D0 0F
004030B0  D0 2C D0 0F  B0 46 D0 0F  90 0E D4 0F  70 10 D4 0F
004030C0  60 44 D1 0F  F0 0F D4 0F  C0 11 D4 0F  50 D3 D0 0F
004030D0  A0 06 D4 0F  70 BF D0 0F  B0 46 D0 0F  00 DB D0 0F
004030E0  90 BF D0 0F  E0 3C D1 0F  D0 9D D4 0F  20 11 D4 0F
004030F0  A0 0F D1 0F  F0 0B D1 0F  60 D3 D0 0F  10 D0 D2 0F
```

```
0019F3EC  0019FE2C
0019F3F0  0FCE3FD8  mfc42.#CDialog::messageMap_4234
0019F3F4  00000001
0019F3F8  0000000C
0019F3FC  0000000A
0019F400  00000002
0019F404  00000000
0019F408  00000002
0019F40C  00000006
0019F410  0000000C
0019F414  00000190
0019F418  00000000
0019F41C  00000060
0019F420  00000060
0019F424  FFE50020
0019F428  002025A1
0019F42C  07000000
0019F430  00000086
```

https://blog.csdn.net/qq_41917908

我们看到了字符串比较函数，我们在字符串比较函数下个CC断点，然后我们就可以通过穷举字母和数字来手动建立密码的对应关系



```
00401857  >  8B55 FC        mov edx,[local.1]
0040185A  .  C64415 DC 00   mov byte ptr ss:[ebp+edx-0x24],0x0
0040185F  .  8B45 BC        mov eax,[local.17]
00401862  .  50             push eax                        ┌s2 = 0000000A ???
00401863  .  8D4D DC        lea ecx,[local.9]               │
00401866  .  51             push ecx                        │s1 = "xuenixiang"
00401867  .  E8 84070000    call <jmp.&MSVCRT.strcmp>       └strcmp
0040186C  .  83C4 08        add esp,0x8
0040186F  .  85C0           test eax,eax
00401871  .v 75 07          jnz short cm.0040187A
00401873  .  E8 F8FEFFFF    call cm.00401770
00401878  .v EB 05          jmp short cm.0040187F
0040187A  >  E8 31FFFFFF    call cm.004017B0
0040187F  >  5F             pop edi                         0019FE2C
00401880  .  5E             pop esi                         0019FE2C
00401881  .  5B             pop ebx                         0019FE2C
00401882  .  8BE5           mov esp,ebp
00401884  .  5D             pop ebp                         0019FE2C
00401885  └. C3             retn
00401886     CC             int3
00401887     CC             int3
```

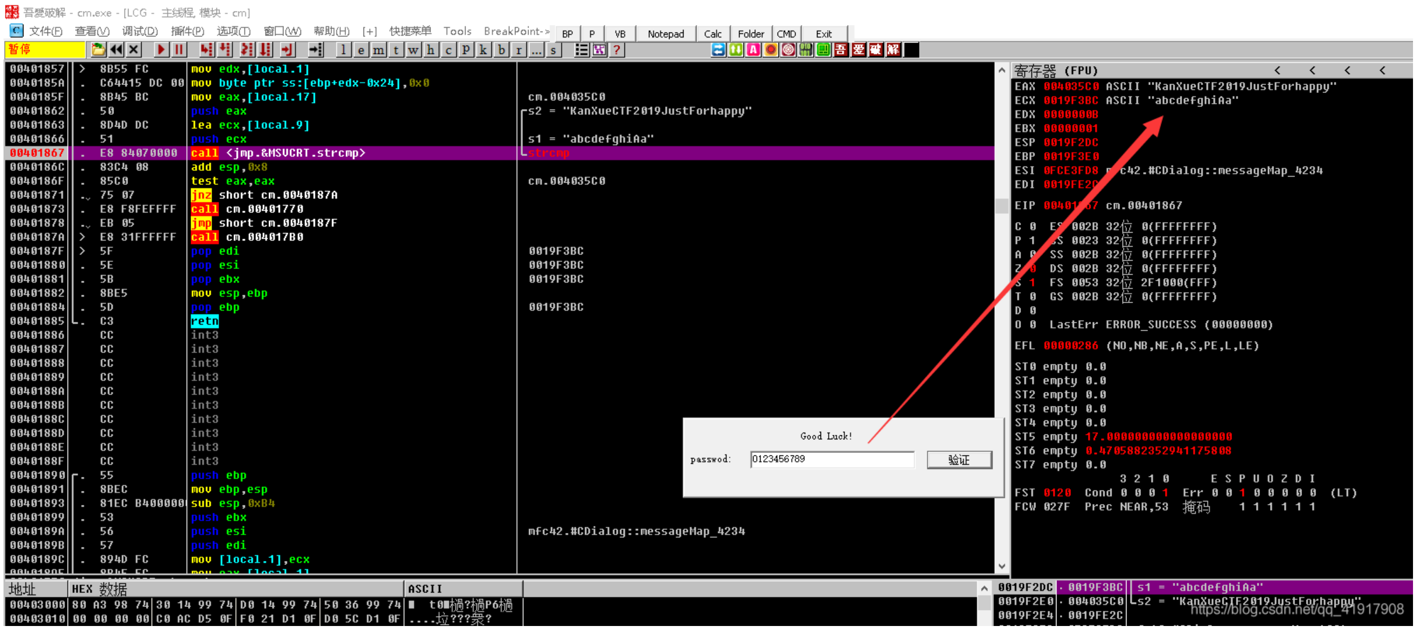
https://blog.csdn.net/qq_41917908

我们先观察和假码比较的那个字符串KanXueCTF2019JustForhappy

这个字符串包含了字母大写小写和数字

接下来就可以缩小举例的范围（排除某些符号）

0123456789对应abcdefghiAa

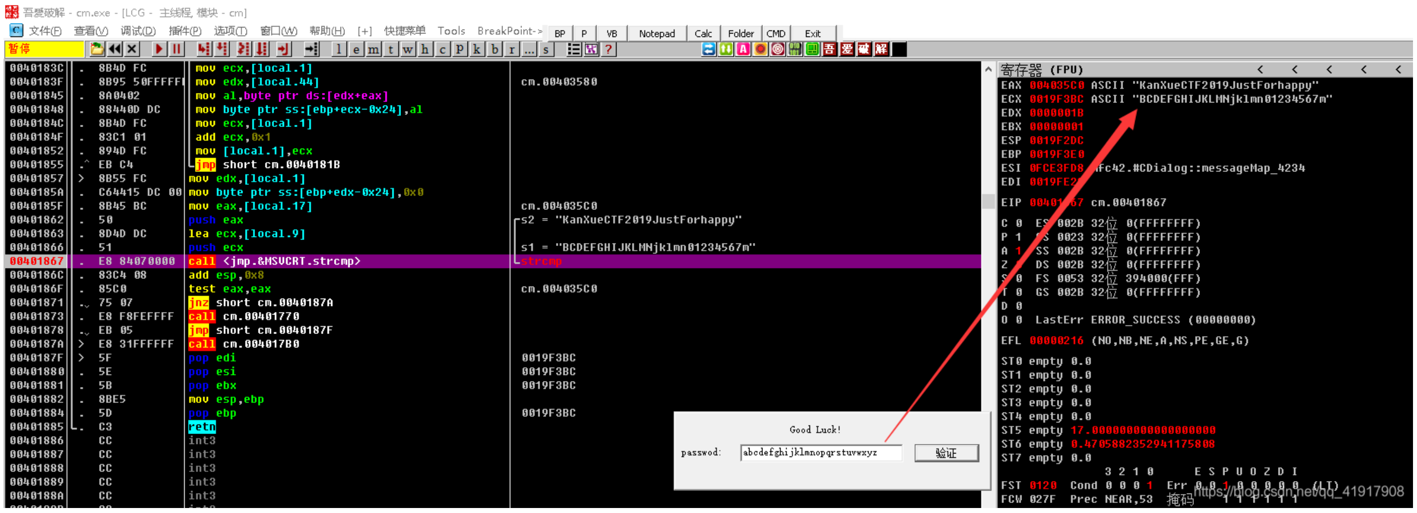

吾爱破解 - cm.exe - [LCG - 主线程, 模块 - cm]
文件(F) 查看(V) 调试(D) 插件(P) 选项(T) 窗口(W) 帮助(H) [+] 快捷菜单 Tools BreakPoint-> BP P VB Notepad Calc Folder CMD Exit
暂停
Good Luck!
验证

```
00401857  > 8B55 FC         mov edx,[local.1]
0040185A  . C64415 DC 00    mov byte ptr ss:[ebp+edx-0x24],0x0
0040185F  . 8B45 BC         mov eax,[local.17]                     cm.004035C0
00401862  . 50              push eax                               ┌s2 = "KanXueCTF2019JustForhappy"
00401863  . 8D4D DC         lea ecx,[local.9]                      │
00401866  . 51              push ecx                               │s1 = "abcdefghiAa"
00401867  . E8 84070000     call <jmp.&MSVCRT.strcmp>              └strcmp
0040186C  . 83C4 08         add esp,0x8
0040186F  . 85C0            test eax,eax                           cm.004035C0
00401871  .v 75 07          jnz short cm.0040187A
00401873  . E8 F8FEFFFF     call cm.00401770
00401878  .v EB 05          jmp short cm.0040187F
0040187A  > E8 31FFFFFF     call cm.004017B0
0040187F  > 5F              pop edi                                0019F3BC
00401880  . 5E              pop esi                                0019F3BC
00401881  . 5B              pop ebx                                0019F3BC
00401882  . 8BE5            mov esp,ebp
00401884  . 5D              pop ebp                                0019F3BC
00401885  . C3              retn
00401886    CC              int3
00401887    CC              int3
00401888    CC              int3
00401889    CC              int3
0040188A    CC              int3
0040188B    CC              int3
0040188C    CC              int3
0040188D    CC              int3
0040188E    CC              int3
0040188F    CC              int3
00401890  ┌. 55              push ebp
00401891  . 8BEC            mov ebp,esp
00401893  . 81EC B4000000   sub esp,0xB4
00401899  . 53              push ebx
0040189A  . 56              push esi                               mfc42.#CDialog::messageMap_4234
0040189B  . 57              push edi
0040189C  . 894D FC         mov [local.1],ecx

寄存器 (FPU)
EAX 004035C0 ASCII "KanXueCTF2019JustForhappy"
ECX 0019F3BC ASCII "abcdefghiAa"
EDX 0000000B
EBX 00000001
ESP 0019F2DC
EBP 0019F3E0
ESI 0FCE3FD8 mfc42.#CDialog::messageMap_4234
EDI 0019FE2C

EIP 00401867 cm.00401867

C 0  ES 002B 32位 0(FFFFFFFF)
P 1  CS 0023 32位 0(FFFFFFFF)
A 0  SS 002B 32位 0(FFFFFFFF)
Z 0  DS 002B 32位 0(FFFFFFFF)
S 1  FS 0053 32位 2F1000(FFF)
T 0  GS 002B 32位 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000286 (NO,NB,NE,A,S,PE,L,LE)

ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 17.000000000000000000
ST6 empty 0.4705882352941175808
ST7 empty 0.0
               3 2 1 0      E S P U O Z D I
FST 0120  Cond 0 0 0 1  Err 0 0 1 0 0 0 0 0  (LT)
FCW 027F  Prec NEAR,53  掩码    1 1 1 1 1 1

地址      HEX 数据                                          ASCII
00403000  80 A3 98 74|30 14 99 74|D0 14 99 74|50 36 99 74  t0 檛?檛P6檛
00403010  00 00 00 00|C0 AC D5 0F|F0 21 D1 0F|D0 5C D1 0F  ....垃???椝?

0019F2DC  . 0019F3BC  ┌s1 = "abcdefghiAa"
0019F2E0  . 004035C0  └s2 = "KanXueCTF2019JustForhappy"
0019F2E4  . 0019FE2C
```

password: 0123456789

https://blog.csdn.net/qq_41917908

abcdefghijklmnopqrstuvwxyz对应BCDEFGHIJKLMNjklmn01234567

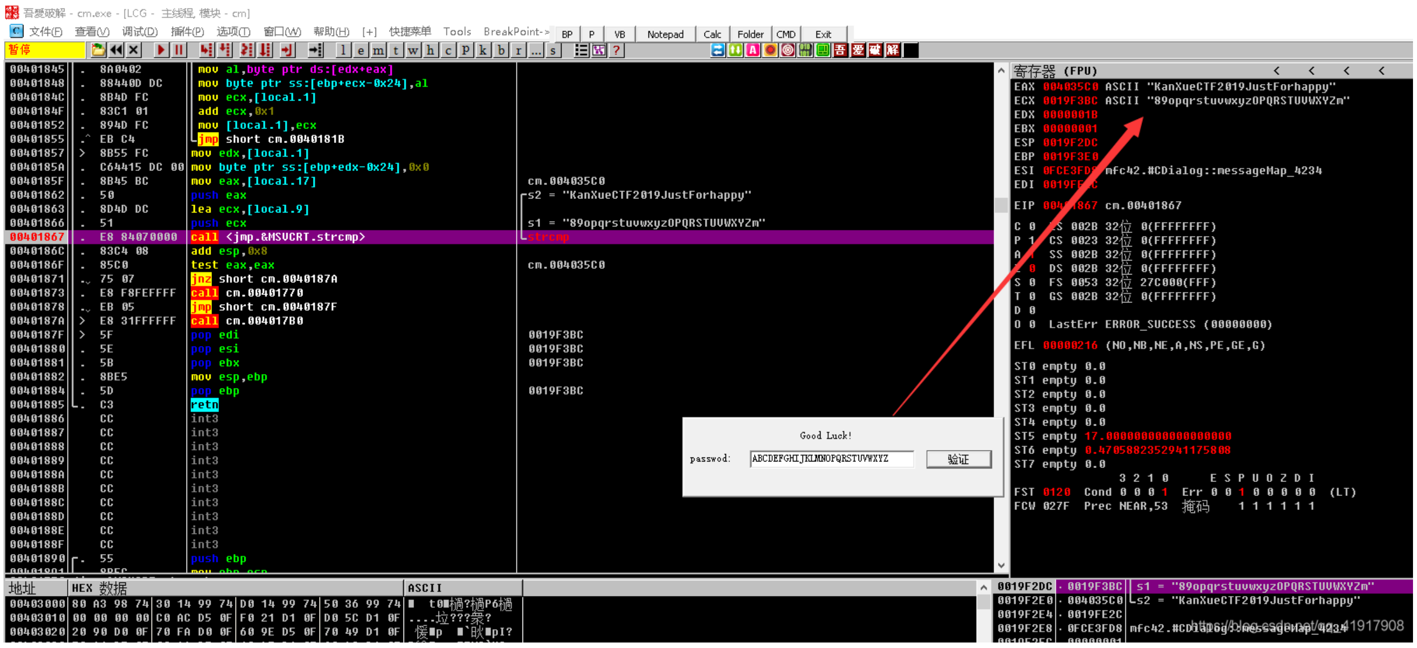

吾爱破解 - cm.exe - [LCG - 主线程, 模块 - cm]
文件(F) 查看(V) 调试(D) 插件(P) 选项(T) 窗口(W) 帮助(H) [+] 快捷菜单 Tools BreakPoint-> BP P VB Notepad Calc Folder CMD Exit
暂停
Good Luck!
验证

```
0040183C  . 8B4D FC         mov ecx,[local.1]
0040183F  . 8B95 50FFFFFF   mov edx,[local.44]                     cm.00403580
00401845  . 8A0402          mov al,byte ptr ds:[edx+eax]
00401848  . 88440D DC       mov byte ptr ss:[ebp+ecx-0x24],al
0040184C  . 8B4D FC         mov ecx,[local.1]
0040184F  . 83C1 01         add ecx,0x1
00401852  . 894D FC         mov [local.1],ecx
00401855  .^ EB C4          jmp short cm.0040181B
00401857  > 8B55 FC         mov edx,[local.1]
0040185A  . C64415 DC 00    mov byte ptr ss:[ebp+edx-0x24],0x0
0040185F  . 8B45 BC         mov eax,[local.17]                     cm.004035C0
00401862  . 50              push eax                               ┌s2 = "KanXueCTF2019JustForhappy"
00401863  . 8D4D DC         lea ecx,[local.9]                      │
00401866  . 51              push ecx                               │s1 = "BCDEFGHIJKLMNjklmn01234567m"
00401867  . E8 84070000     call <jmp.&MSVCRT.strcmp>              └strcmp
0040186C  . 83C4 08         add esp,0x8
0040186F  . 85C0            test eax,eax                           cm.004035C0
00401871  .v 75 07          jnz short cm.0040187A
00401873  . E8 F8FEFFFF     call cm.00401770
00401878  .v EB 05          jmp short cm.0040187F
0040187A  > E8 31FFFFFF     call cm.004017B0
0040187F  > 5F              pop edi                                0019F3BC
00401880  . 5E              pop esi                                0019F3BC
00401881  . 5B              pop ebx                                0019F3BC
00401882  . 8BE5            mov esp,ebp
00401884  . 5D              pop ebp                                0019F3BC
00401885  . C3              retn
00401886    CC              int3
00401887    CC              int3
00401888    CC              int3
00401889    CC              int3
0040188A    CC              int3

寄存器 (FPU)
EAX 004035C0 ASCII "KanXueCTF2019JustForhappy"
ECX 0019F3BC ASCII "BCDEFGHIJKLMNjklmn01234567m"
EDX 0000001B
EBX 00000001
ESP 0019F2DC
EBP 0019F3E0
ESI 0FCE3FD8 mfc42.#CDialog::messageMap_4234
EDI 0019FE2C

EIP 00401867 cm.00401867

C 0  ES 002B 32位 0(FFFFFFFF)
P 1  CS 0023 32位 0(FFFFFFFF)
A 1  SS 002B 32位 0(FFFFFFFF)
Z 0  DS 002B 32位 0(FFFFFFFF)
S 0  FS 0053 32位 394000(FFF)
T 0  GS 002B 32位 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000216 (NO,NB,NE,A,NS,PE,GE,G)

ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 17.000000000000000000
ST6 empty 0.4705882352941175808
ST7 empty 0.0
               3 2 1 0      E S P U O Z D I
FST 0120  Cond 0 0 0 1  Err 0 0 1 0 0 0 0 0  (LT)
FCW 027F  Prec NEAR,53  掩码    1 1 1 1 1 1
```

password: abcdefghijklmnopqrstuvwxyz

https://blog.csdn.net/qq_41917908

ABCDEFGHIJKLMNOPQRSTUVWXYZ对应89opqrstuvwxyzOPQRSTUVWXYZ


吾爱破解 - cm.exe - [LCG - 主线程, 模块 - cm]
文件(F) 查看(V) 调试(D) 插件(P) 选项(T) 窗口(W) 帮助(H) [+] 快捷菜单 Tools BreakPoint-> BP P VB Notepad Calc Folder CMD Exit
暂停
Good Luck!
验证

```
00401845  . 8A0402          mov al,byte ptr ds:[edx+eax]
00401848  . 88440D DC       mov byte ptr ss:[ebp+ecx-0x24],al
0040184C  . 8B4D FC         mov ecx,[local.1]
0040184F  . 83C1 01         add ecx,0x1
00401852  . 894D FC         mov [local.1],ecx
00401855  .^ EB C4          jmp short cm.0040181B
00401857  > 8B55 FC         mov edx,[local.1]
0040185A  . C64415 DC 00    mov byte ptr ss:[ebp+edx-0x24],0x0
0040185F  . 8B45 BC         mov eax,[local.17]                     cm.004035C0
00401862  . 50              push eax                               ┌s2 = "KanXueCTF2019JustForhappy"
00401863  . 8D4D DC         lea ecx,[local.9]                      │
00401866  . 51              push ecx                               │s1 = "89opqrstuvwxyzOPQRSTUVWXYZm"
00401867  . E8 84070000     call <jmp.&MSVCRT.strcmp>              └strcmp
0040186C  . 83C4 08         add esp,0x8
0040186F  . 85C0            test eax,eax                           cm.004035C0
00401871  .v 75 07          jnz short cm.0040187A
00401873  . E8 F8FEFFFF     call cm.00401770
00401878  .v EB 05          jmp short cm.0040187F
0040187A  > E8 31FFFFFF     call cm.004017B0
0040187F  > 5F              pop edi                                0019F3BC
00401880  . 5E              pop esi                                0019F3BC
00401881  . 5B              pop ebx                                0019F3BC
00401882  . 8BE5            mov esp,ebp
00401884  . 5D              pop ebp                                0019F3BC
00401885  . C3              retn
00401886    CC              int3
00401887    CC              int3
00401888    CC              int3
00401889    CC              int3
0040188A    CC              int3
0040188B    CC              int3
0040188C    CC              int3
0040188D    CC              int3
0040188E    CC              int3
0040188F    CC              int3
00401890  ┌. 55              push ebp

寄存器 (FPU)
EAX 004035C0 ASCII "KanXueCTF2019JustForhappy"
ECX 0019F3BC ASCII "89opqrstuvwxyzOPQRSTUVWXYZm"
EDX 0000001B
EBX 00000001
ESP 0019F2DC
EBP 0019F3E0
ESI 0FCE3FD8 mfc42.#CDialog::messageMap_4234
EDI 0019FE2C

EIP 00401867 cm.00401867

C 0  ES 002B 32位 0(FFFFFFFF)
P 1  CS 0023 32位 0(FFFFFFFF)
A 1  SS 002B 32位 0(FFFFFFFF)
Z 0  DS 002B 32位 0(FFFFFFFF)
S 0  FS 0053 32位 27C000(FFF)
T 0  GS 002B 32位 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000216 (NO,NB,NE,A,NS,PE,GE,G)

ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 17.000000000000000000
ST6 empty 0.4705882352941175808
ST7 empty 0.0
               3 2 1 0      E S P U O Z D I
FST 0120  Cond 0 0 0 1  Err 0 0 1 0 0 0 0 0  (LT)
FCW 027F  Prec NEAR,53  掩码    1 1 1 1 1 1

地址      HEX 数据                                          ASCII
00403000  80 A3 98 74|30 14 99 74|D0 14 99 74|50 36 99 74  t0 檛?檛P6檛
00403010  00 00 00 00|C0 AC D5 0F|F0 21 D1 0F|D0 5C D1 0F  ....垃???椝?
00403020  20 90 D0 0F|70 FA D0 0F|60 9E D5 0F|70 49 D1 0F   惾 p `瞅 pI?

0019F2DC  . 0019F3BC  ┌s1 = "89opqrstuvwxyzOPQRSTUVWXYZm"
0019F2E0  . 004035C0  └s2 = "KanXueCTF2019JustForhappy"
0019F2E4  . 0019FE2C
0019F2E8  . 0FCE3FD8  mfc42.#CDialog::messageMap_4234
```

password: ABCDEFGHIJKLMNOPQRSTUVWXYZ

https://blog.csdn.net/qq_41917908

最后得出关系表

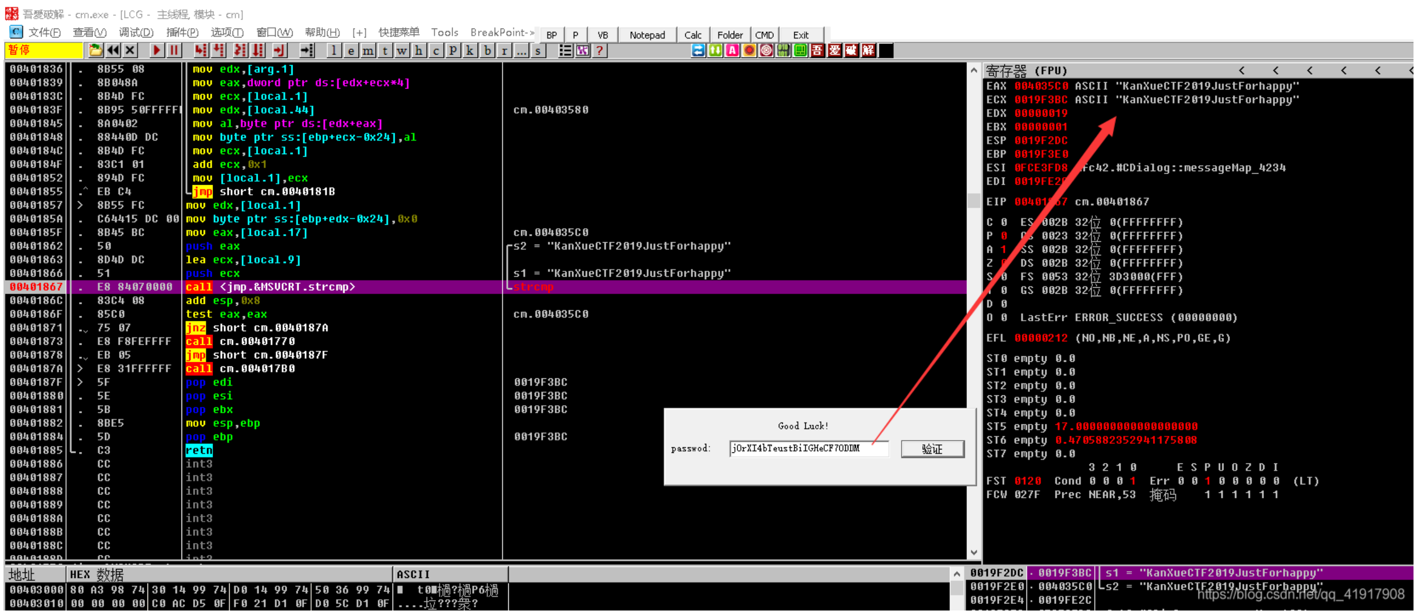
abcdefghiAa对应0123456789

BCDEFGHIJKLMNjklmn01234567对应abcdefghijklmnopqrstuvwxyz

89opqrstuvwxyzOPQRSTUVWXYZ对应ABCDEFGHIJKLMNOPQRSTUVWXYZ

通过手动计算可以得出：

KanXueCTF2019JustForhappy对应j0rXI4bTeustBiIGHeCF70DDM


吾爱破解 - cm.exe - [LCG - 主线程, 模块 - cm]
文件(F) 查看(V) 调试(D) 插件(P) 选项(T) 窗口(W) 帮助(H) [+] 快捷菜单 Tools BreakPoint->
BP P VB Notepad Calc Folder CMD Exit
暂停

```
00401836  .  8B55 08        mov edx,[arg.1]
00401839  .  8B048A         mov eax,dword ptr ds:[edx+ecx*4]
0040183C  .  8B4D FC        mov ecx,[local.1]
0040183F  .  8B95 50FFFFFF  mov edx,[local.44]                    cm.00403580
00401845  .  8A0402         mov al,byte ptr ds:[edx+eax]
00401848  .  88440D DC      mov byte ptr ss:[ebp+ecx-0x24],al
0040184C  .  8B4D FC        mov ecx,[local.1]
0040184F  .  83C1 01        add ecx,0x1
00401852  .  894D FC        mov [local.1],ecx
00401855  .^ EB C4          jmp short cm.0040181B
00401857  >  8B55 FC        mov edx,[local.1]
0040185A  .  C64415 DC 00   mov byte ptr ss:[ebp+edx-0x24],0x0
0040185F  .  8B45 BC        mov eax,[local.17]                    cm.004035C0
00401862  .  50             push eax                             ┌s2 = "KanXueCTF2019JustForhappy"
00401863  .  8D4D DC        lea ecx,[local.9]                    │
00401866  .  51             push ecx                             │s1 = "KanXueCTF2019JustForhappy"
00401867  .  E8 84070000    call <jmp.&MSVCRT.strcmp>            └strcmp
0040186C  .  83C4 08        add esp,0x8
0040186F  .  85C0           test eax,eax                          cm.004035C0
00401871  .v 75 07          jnz short cm.0040187A
00401873  .  E8 F8FEFFFF    call cm.00401770
00401878  .v EB 05          jmp short cm.0040187F
0040187A  >  E8 31FFFFFF    call cm.004017B0
0040187F  >  5F             pop edi                               0019F3BC
00401880  .  5E             pop esi                               0019F3BC
00401881  .  5B             pop ebx                               0019F3BC
00401882  .  8BE5           mov esp,ebp
00401884  .  5D             pop ebp                               0019F3BC
00401885  .  C3             retn
00401886     CC             int3
00401887     CC             int3
00401888     CC             int3
00401889     CC             int3
0040188A     CC             int3
0040188B     CC             int3
0040188C     CC             int3
```

```
寄存器 (FPU)
EAX 004035C0 ASCII "KanXueCTF2019JustForhappy"
ECX 0019F3BC ASCII "KanXueCTF2019JustForhappy"
EDX 00000019
EBX 00000001
ESP 0019F2DC
EBP 0019F3E0
ESI 0FCE3FD8 mfc42.#CDialog::messageMap_4234
EDI 0019FE2C

EIP 00401867 cm.00401867

C 0  ES 002B 32位 0(FFFFFFFF)
P 0  CS 0023 32位 0(FFFFFFFF)
A 1  SS 002B 32位 0(FFFFFFFF)
Z 0  DS 002B 32位 0(FFFFFFFF)
S 0  FS 0053 32位 3D3000(FFF)
T 0  GS 002B 32位 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000212 (NO,NB,NE,A,NS,PO,GE,G)

ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 17.000000000000000000
ST6 empty 0.4705882352941175808
ST7 empty 0.0
               3 2 1 0      E S P U O Z D I
FST 0120  Cond 0 0 0 1  Err 0 0 1 0 0 0 0 0  (LT)
FCW 027F  Prec NEAR,53  掩码    1 1 1 1 1 1
```

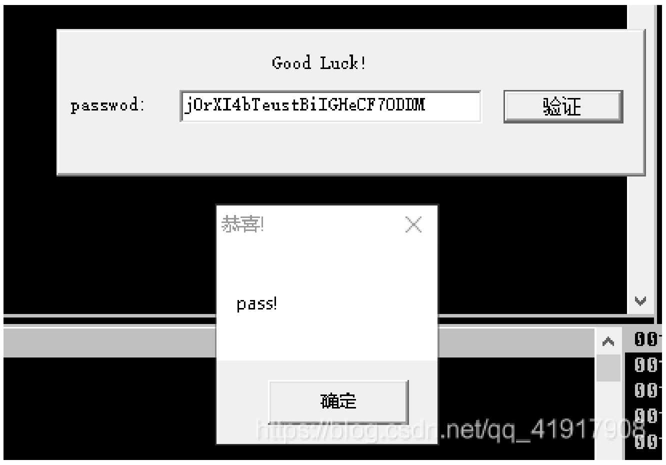
```
地址      HEX 数据                                        ASCII
00403000 80 A3 98 74|30 14 99 74|D0 14 99 74|50 36 99 74  ■  t0■樋?樋P6樋
00403010 00 00 00 00|C0 AC D5 0F|F0 21 D1 0F|D0 5C D1 0F  ....垃???棸?
```

```
0019F2DC  0019F3BC  ┌s1 = "KanXueCTF2019JustForhappy"
0019F2E0  004035C0  └s2 = "KanXueCTF2019JustForhappy"
0019F2E4  0019FE2C
```

Good Luck!

passwod: j0rXI4bTeustBiIGHeCF70DDM 验证

通过验证，flag正确



Good Luck!

passwod: j0rXI4bTeustBiIGHeCF70DDM 验证

恭喜!

pass!

确定

```
table = "abcdefghiABCDEFGHIJKLMNjklmn0123456789opqrstuvwxyzOPQRSTUVWXYZ"
s = "KanXueCTF2019JustForhappy"
ff = []
for i in s:
    ff.append(table.index(i))

flag = ""
for i in ff:
    if 0 <= i <= 9:
        flag += chr(i + 48)
    elif 9 < i <= 35:
        flag += chr(i + 87)
    elif i > 36:
        flag += chr(i + 29)

print flag
```