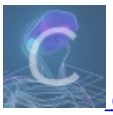
re学习笔记（19）i春秋-re-crackme

原创

Forgo7ten 于 2019-12-28 15:15:14 发布 389 收藏

分类专栏： ctf小白成长ing # reverse 文章标签： ctf reverse i春秋 crackme

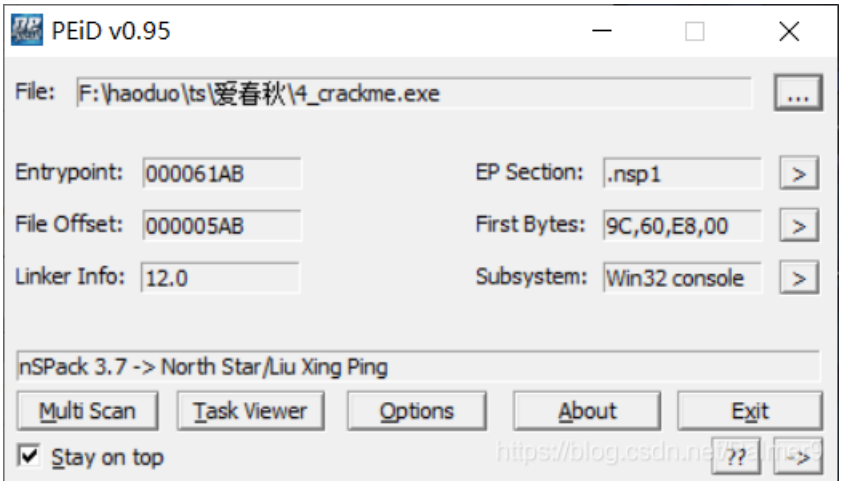
本文链接：https://blog.csdn.net/Palmer9/article/details/103587042



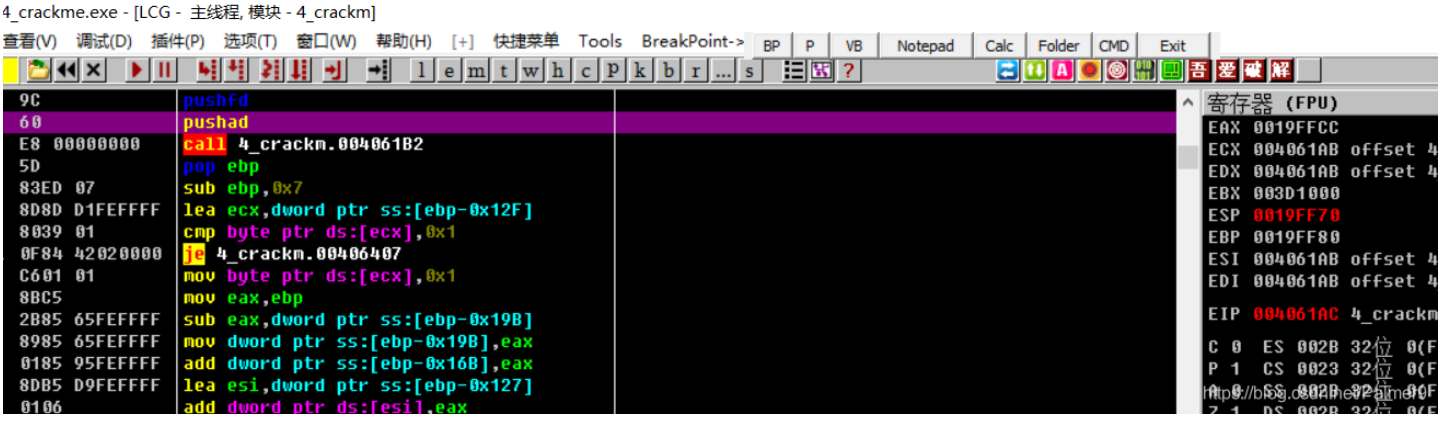
新手一枚，如有错误（不足）请指正，谢谢！！

题目下载：下载地址

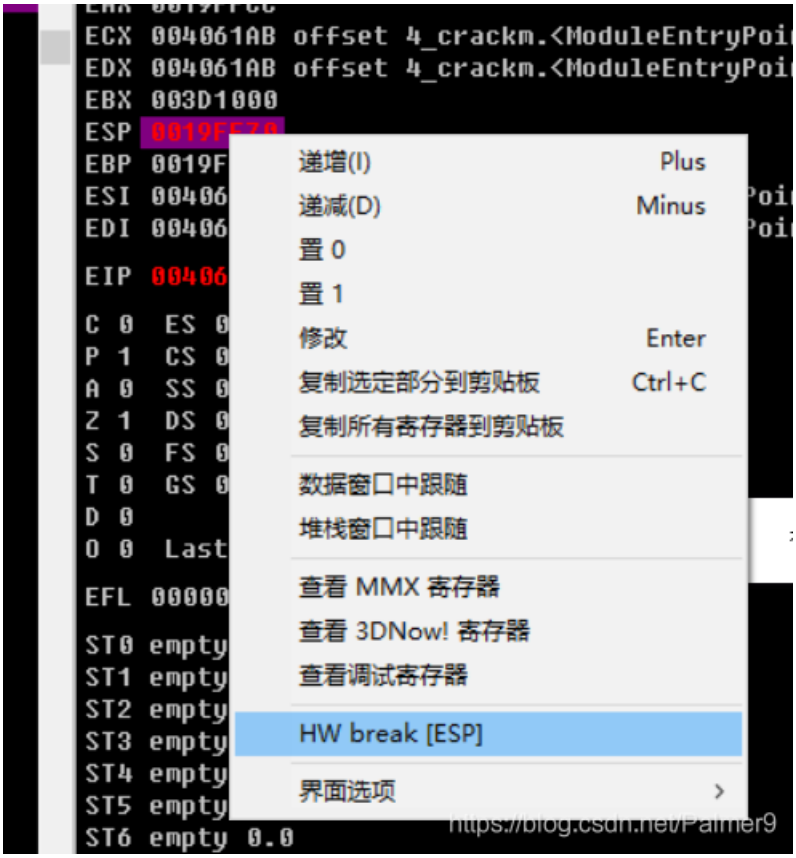
先用peid查壳



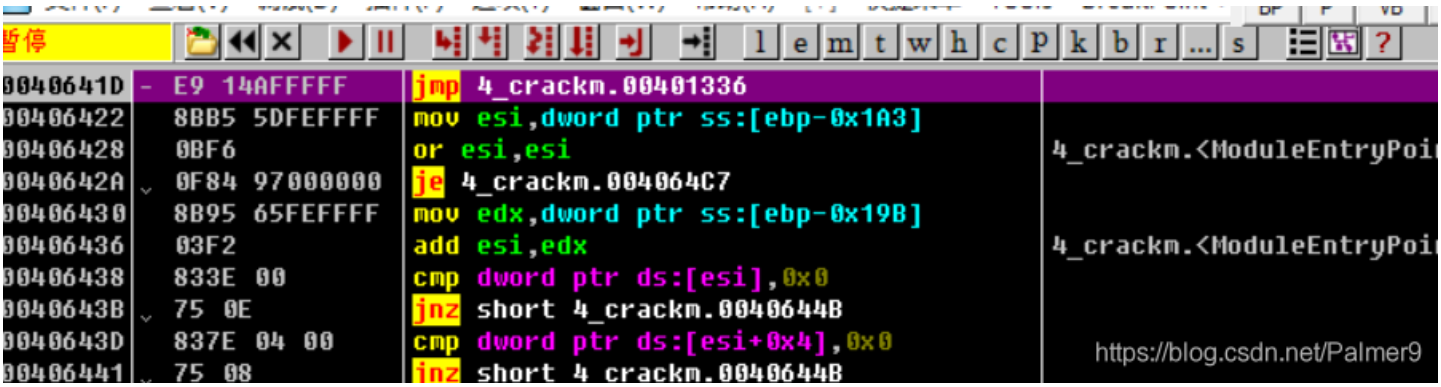
nsp1的壳，exe可执行文件，打开od，F8单步一下，发现可以使用esp定律脱壳



然后下硬件断点



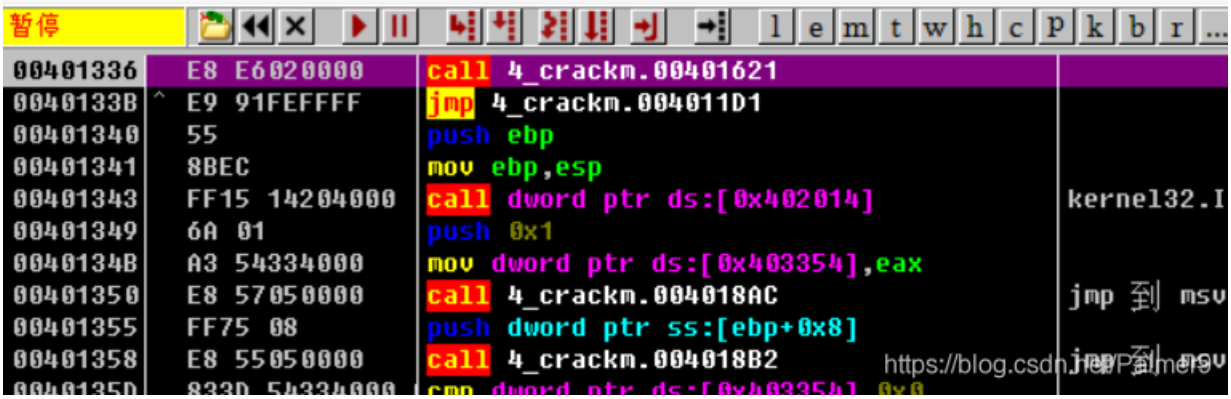
F9运行程序



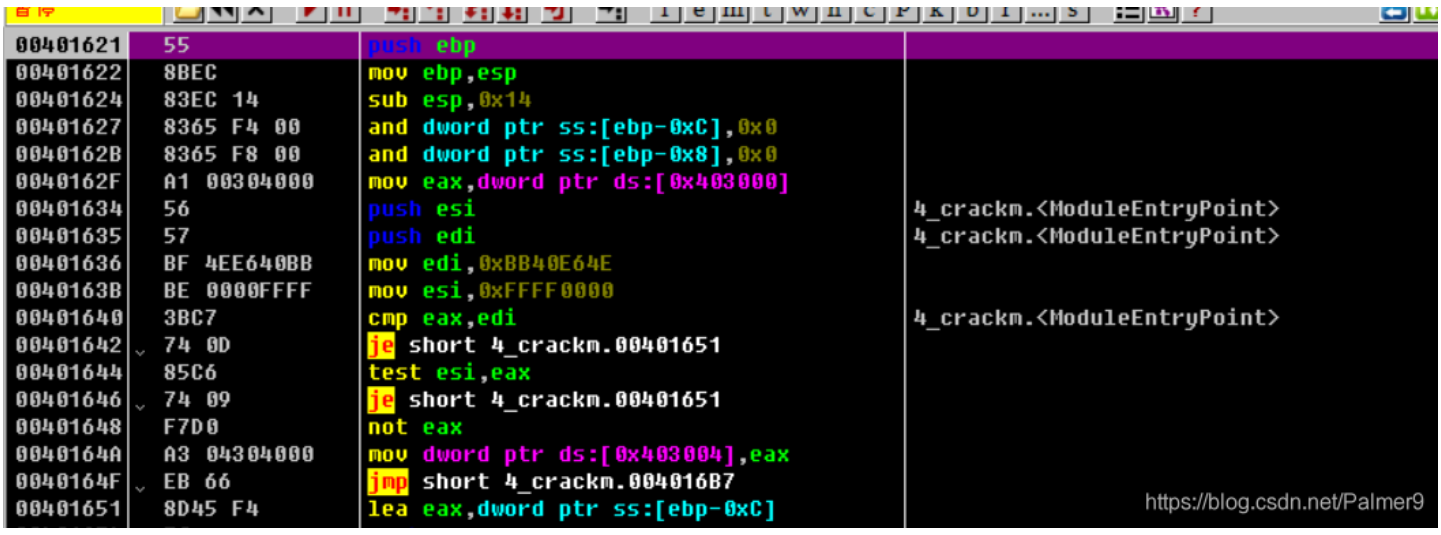
F8单步

在这里对代码进行分析，删除分析

来到这里，继续单步



来到OEP



右键用od自带的脱壳工具脱壳

IDA载入脱壳后的程序

IDA View-A | Strings window | Hex View-1 | Structures

dress	Length	Type	String
.nsp0:00…	00000013	C	Please Input Flag:
.nsp0:00…	00000008	C	error!\n
.nsp0:00…	00000008	C	right!\n
.nsp0:00…	00000010	C	his_is_not_flag
.nsp0:00…	00000030	C	D:\\projects\\crackme\\crackme\\Release\\crackme.pdb
.nsp0:00…	0000017E	C	<?xml version='1.0' encoding='UTF-8' standalone='yes'?>\r\n<a…
.nsp0:00…	00000200	C	GetSystemTimeAsFileTimeGetCurrentThreadIdGetCurrentProcessIdQ…
.nsp1:00…	00000018	C	)Ov _/vp|/v岬/v館/v館Ov
.nsp1:00…	0000000D	C	KERNEL32.DLL
.nsp1:00…	0000000D	C	MSVCR120.DLL
.nsp1:00…	0000000D	C	LoadLibraryA
.nsp1:00…	0000000E	C	etProcAddress
.nsp1:00…	0000000F	C	VirtualProtect
.nsp1:00…	00000014	C	_configthreadlocale
.idata2:…	0000000D	C	kernel32.dll
.idata2:…	0000000D	C	msvcr120.dll

字符串中找到right，双击
交叉引用

```
.nsp0:00402114 ; char aError[]
.nsp0:00402114 aError          db 'error!',0Ah,0       ; DATA XREF: sub_401000+52↑o
.nsp0:00402114                                         ; sub_401000:loc_4010A3↑o
.nsp0:0040211C aRight          db 'right!',0Ah,0       ; DATA XREF: sub_401000+9C↑o
.nsp0:00402124                 align 10h
.nsp0:00402130 byte_402130     db 74h                  ; DATA XREF: sub_401000:loc_40107F↑r
.nsp0:00402131 aHisIsNotFlag   db 'his_is_not_flag',0
.nsp0:00402141                 align 10h
.nsp0:00402150 dword_402150    dd 12h                  ; DATA XREF: sub_401000+8D↑r
.nsp0:00402154                 dd 4, 8, 14h, 24h, 5Ch, 4Ah, 3Dh, 56h, 0Ah, 10h, 67h, 0
.nsp0:00402184                 dd 41h, 0
```

xrefs to aRight

Direct	Ty	Address	Text
Up	o	sub_401000+9C	push offset aRight; "right!\n"

OK | Cancel | Search | Help

Line 1 of 1


```
1  signed int sub_401000()
2  {
3    signed int result; // eax
4    int v1; // eax
5    char Buf; // [esp+4h] [ebp-38h]
6    char Dst; // [esp+5h] [ebp-37h]
7
8    Buf = 0;
9    memset(&Dst, 0, 0x31u);
10   printf("Please Input Flag:");
11   gets_s(&Buf, 0x2Cu);
12   if ( strlen(&Buf) == 42 )
```

```
13  {
14    v1 = 0;
15    while ( (*(&Buf + v1) ^ byte_402130[v1 % 16]) == dword_402150[v1] )
16    {
17      if ( ++v1 >= 42 )
18      {
19        printf("right!\n");
20        goto LABEL_8;
21      }
22    }
23    printf("error!\n");
24LABEL_8:
25    result = 0;
26  }
27  else
28  {
29    printf("error!\n");
30    result = -1;
31  }
32  return result;
33}
```


F5伪代码之后

```
6
7   user_input[0] = 0;
8   memset(&user_input[1], 0, 0x31u);
9   printf("Please Input Flag:");
10  gets_s(user_input, 0x2Cu);
11  if ( strlen(user_input) == 42 )               // 用户的输入长度要等于42
12  {
13    v1 = 0;
14    while ( (user_input[v1] ^ str1[v1 % 16]) == str2[v1] )// 将用户输入与str1异或后与str2进行比较，要相等
15    {
16      if ( ++v1 >= 42 )
17      {
18        printf("right!\n");
19        goto LABEL_8;
```


写脚本

```
#include <stdio.h>
int main(void)
{
    int i;
    char str1[] = "this_is_not_flag";
    char str2[] = {0x12,0x4,0x8,0x14,0x24,0x5C,0x4A,0x3D,0x56,0x0A,0x10,0x67,0x0,0x41,0x0,0x1,0x46,0x5A,0x44,0x4
2,0x6E,0x0C,0x44,0x72,0x0C,0x0D,0x40,0x3E,0x4B,0x5F,0x2,0x1,0x4C,0x5E,0x5B,0x17,0x6E,0x0C,0x16,0x68,0x5B,0x12};
    char flag[43] = {0};
    for(i=0;i<42;i++)
        flag[i] = str2[i] ^ str1[i%16];
    puts(flag);
    return 0;
}
```

main.c

```c
#include <stdio.h>
int main(void)
{
    int i;
    char str1[] = "this_is_not_flag";
    char str2[] = {0x12,0x4,0x8,0x14,0x24,0x5C,0x4A,0x3D,0x56,0x0A
    char flag[43] = {0};
    for(i=0;i<42;i++)
        flag[i] = str2[i] ^ str1[i%16];
    puts(flag);
    return 0;
}
```

```
F:\haoduo\daima\C\ts1\bin\Debug\ts1.exe
flag{59b8ed8f-af22-11e7-bb4a-3cf862d1ee75}

Process returned 0 (0x0)   execution time : 0.218 s
Press any key to continue.
```


得到flag

flag{59b8ed8f-af22-11e7-bb4a-3cf862d1ee75}