

随风而逝的白色相簿

博客园

首页

新随笔

联系

订阅

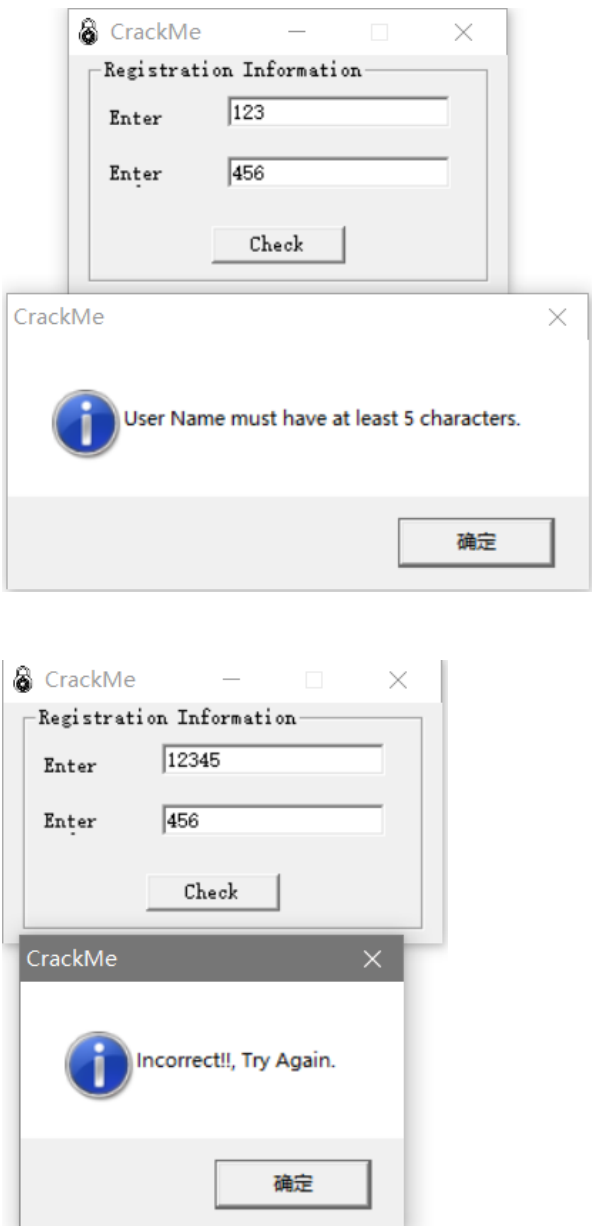
管理

随笔 - 10 文章 - 0 评论 - 1

Crackme019

Crackme019 的逆向分析

1.程序观察



可以看到，程序要求用户名至少要5位。

2.简单查壳

公告

昵称： 随风而逝的白色相簿
园龄： 1年9个月
粉丝： 1
关注： 4

2019年9月						
日	一	二	三	四	五	六
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5
6	7	8	9	10	11	12

搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

随笔分类

160 Crackme(10)

随笔档案

2019年9月(10)
2018年10月(1)

最新评论

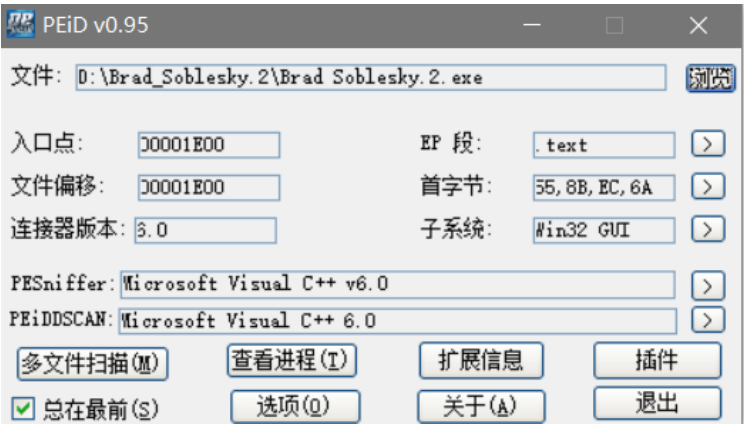
1. Re:Crackme014
写的真好

--随风而逝的白色相簿

阅读排行榜

1. PHP一句话木马(6803)
0 Crackme007(7)

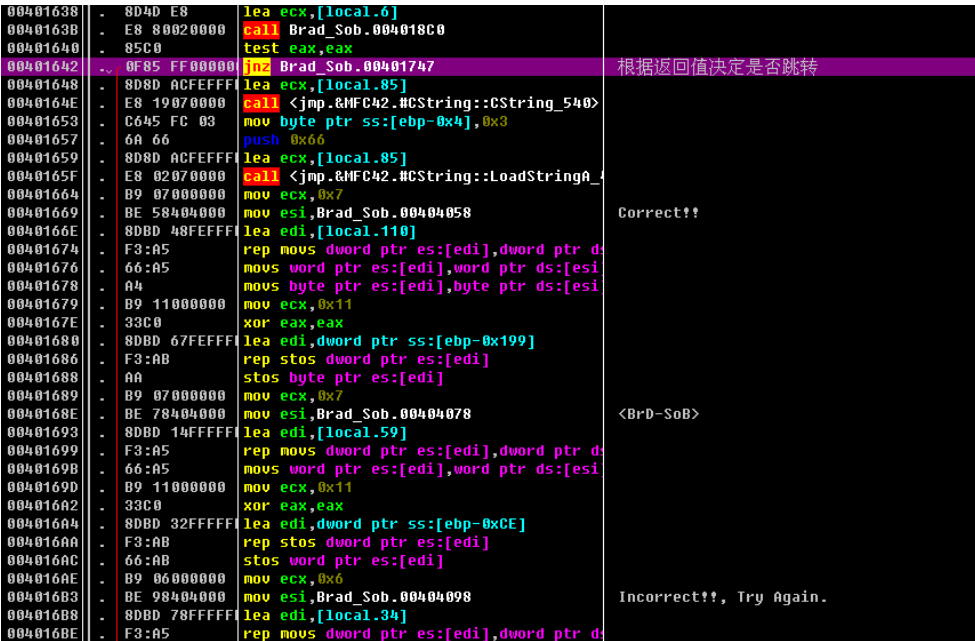
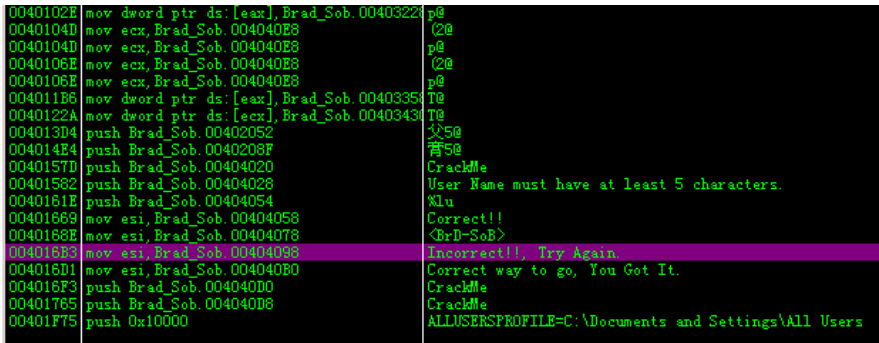
评论排行榜



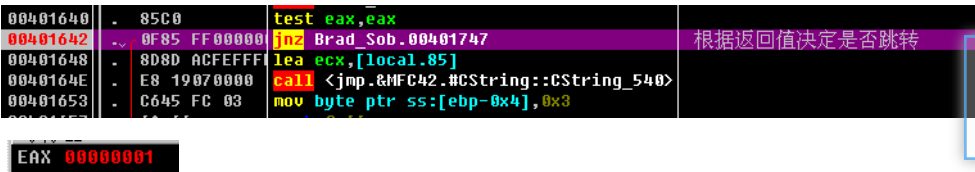
无壳。

3.程序分析

OD 载入程序，搜索字符串。

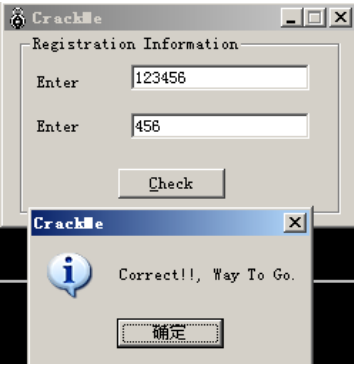


可以看到，字符串上方不远处有一个跳转语句。
在 JNZ 语句处下断点，运行程序，中断在了断点处



修改 ZF 标志位

```
C 0 ES 0023 32位 0(FFFFFFFF)
P 0 CS 001B 32位 0(FFFFFFFF)
A 0 SS 0023 32位 0(FFFFFFFF)
Z 1 DS 0023 32位 0(FFFFFFFF)
S 0 FS 003B 32位 7FFDE000(FFF)
T 0 GS 0000 NULL
D 0
```



因为 eax 的值是 JNZ 语句上面的函数返回的

0040163B	. E8 80020000	call Brad_Sob.004018C0	
00401640	. 85C0	test eax, eax	
00401642	. 0F85 FF000000	jnz Brad_Sob.00401747	根据返回值决定是否跳转

所以我们进入这个函数里面看一看

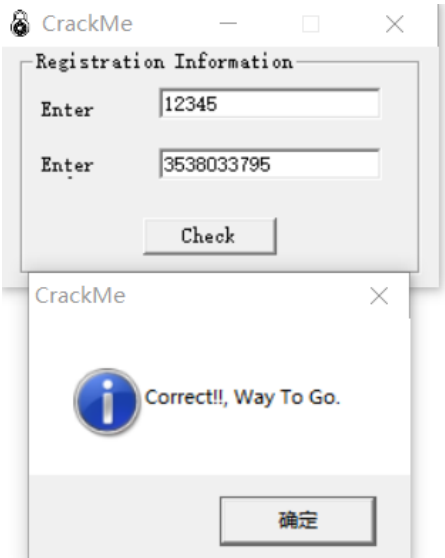
004018C0	. \$ 55	push ebp	
004018C1	. 8BEC	mov ebp, esp	
004018C3	. 51	push ecx	
004018C4	. 894D FC	mov [local.1], ecx	
004018C7	. 8B45 08	mov eax, [arg.1]	
004018CA	. 50	push eax	
004018CB	. 8B4D FC	mov ecx, [local.1]	
004018CE	. 8B11	mov edx, dword ptr ds:[ecx]	edx = 注册码
004018D0	. 52	push edx	
004018D1	. E8 0A000000	call Brad_Sob.004018E0	进行比较
004018D6	. 83C4 08	add esp, 0x8	
004018D9	. 8BE5	mov esp, ebp	
004018DB	. 5D	pop ebp	00393800
004018DC	. C2 0400	ret 0x4	

```
0012F6D0 . 00393800 ASCII "456"
0012F6D4 . 00393850 ASCII "3538033795"
```

可以看到，004018C0 这个函数在 004018D1 处调用了函数，参数有两个，其中一个是我们输入的假码，另一个可能是真码，我们试一下

0

推荐



那 004018D1 处的这个函数有可能就是比较函数了，我们进入这个函数内部看一下

004018E0	55	push ebp	
004018E1	8BEC	mov ebp,esp	
004018E3	8B45 0C	mov eax,[arg.2]	
004018E6	50	push eax	s2 = "3538033795"
004018E7	8B4D 08	mov ecx,[arg.1]	
004018EA	51	push ecx	s1 = "456"
004018EB	FF15 B4314000	call dword ptr ds:[<MSVCRT.7B5C8A7E>]	mbcmp
004018F1	83C4 08	add esp,0x8	
004018F4	5D	pop ebp	00393B00
004018F5	C3	ret	

这个函数又调用了 cmp 函数进行比较

看来，004018C0 函数就是用来比较注册码是否正确的，正确 eax 返回0，不正确返回非0。

下面分析程序的算法

00401542	8D45 EC	lea eax,[local.5]	
00401545	50	push eax	
00401546	68 E8030000	push 0x3E8	
00401548	8B8D 40FEFF	mov ecx,[local.112]	
00401551	E8 34080000	call <jmp.&NFC42.8CWnd::GetDlgItemTextA>	取得输入用户名
00401556	8D4D E8	lea ecx,[local.6]	
00401559	51	push ecx	
0040155A	68 E9030000	push 0x3E9	
0040155F	8B8D 40FEFF	mov ecx,[local.112]	
00401565	E8 20080000	call <jmp.&NFC42.8CWnd::GetDlgItemTextA>	取得输入注册码
0040156A	8D4D EC	lea ecx,[local.5]	
0040156D	E8 DE020000	call Brad_Sob.00401850	计算用户名长度
00401572	8945 E4	mov [local.7],eax	
00401575	837D E4 05	cmp [local.7],0x5	比较长度。小于5不跳转
00401579	7D 43	jge short Brad_Sob.0040158E	
0040157B	6A 40	push 0x40	

程序首先求得用户名和注册码的长度，如果用户名长度小于5就会报错。

然后程序建立循环，循环次数为用户名的长度

- 取用户名一个字符 name[n]，n 为循环次数
- 让一个十六进制的默认值 0x81276345 加上 name[n]
- 取循环次数 n，将 n 左移 8 位
- 步骤2的结果与 步骤3的结果进行异或运算
- 取循环次数加一
- 让用户名长度乘以循环次数，然后按位取反
- 5和6的结果相乘
- 4的结果再和7的结果相乘

以上就是循环的内容，如下图




004015C5	> EB 09	jmp short Brad_Sob.004015D0	
004015C7	> 8B55 E0	mov edx,[local.8]	
004015CA	> 83C2 01	add edx,0x1	循环次数 n 加1
004015CD	> 8955 E0	mov [local.8],edx	
004015D0	> 8B45 E0	mov eax,[local.8]	
004015D3	> 3B45 E4	cmp eax,[local.7]	比较循环是否达到次数
004015D6	> 7D 42	jge short Brad_Sob.0040161A	达到次数就跳出循环
004015D8	> 8B4D E0	mov ecx,[local.8]	
004015DB	> 51	push ecx	
004015DC	> 8D4D EC	lea ecx,[local.5]	
004015DF	> E8 1C030000	call Brad_Sob.00401900	a1 = key[n]
004015E4	> 0FBED0	movsx edx,a1	edx = a1
004015E7	> 8B45 F0	mov eax,[local.4]	eax = 81276345
004015EA	> 03C2	add eax,edx	eax = eax + edx
004015EC	> 8945 F0	mov [local.4],eax	code = 81276345 + key[n]
004015EF	> 8B4D E0	mov ecx,[local.8]	
004015F2	> C1E1 08	shl ecx,0x8	
004015F5	> 8B55 F0	mov edx,[local.4]	
004015F8	> 3D01	xor edx,ecx	code = code xor(n<<8)
004015FA	> 8955 F0	mov [local.4],edx	
004015FD	> 8B45 E0	mov eax,[local.8]	eax = n
00401600	> 83C0 01	add eax,0x1	eax = n + 1
00401603	> 8B4D E4	mov ecx,[local.7]	ecx = 5
00401606	> 0FAF4D E0	imul ecx,[local.8]	ecx = 5 * n
00401609	> F7D1	not ecx	~ecx
0040160C	> 0FAFC1	imul eax,ecx	eax = eax*ecx
0040160F	> 8B55 F0	mov edx,[local.4]	edx = code
00401612	> 0FAFD0	imul edx,eax	code = code * eax
00401615	> 8955 F0	mov [local.4],edx	
00401618	> EB AD	jmp short Brad_Sob.004015C7	

循环完成之后，程序将结果转化为 lu 类型的，也就是无符号长整形整数

0040161A	> 8B45 F0	mov eax,[local.4]	
0040161D	> 50	push eax	
0040161E	> 68 54404000	push Brad_Sob.00404054	%lu
00401623	> 8D4D DC	lea ecx,[local.9]	
00401626	> 51	push ecx	
00401627	> E8 52070000	call <jmp.&MFC42.#CString::Format_2818>	格式化计算出来的值
0040162C	> 83C4 0C	add esp,0xC	
0040162F	> 8D4D DC	lea ecx,[local.9]	
00401632	> E8 79020000	call Brad_Sob.004018B0	返回值的地址
00401637	> 50	push eax	
00401638	> 8D4D E8	lea ecx,[local.6]	
0040163B	> E8 80020000	call Brad_Sob.004018C0	比较函数
00401640	> 85C0	test eax,eax	
00401642	> 0F85 FF000000	jnz Brad_Sob.00401747	根据返回值决定是否跳转

这就是最终的注册码啦！

4.久违的注册机环节



```
#include <stdio.h>
#include <string.h>
#include <Windows.h>

int Key()
{
    char szName[20] = { 0 };
    int NameLen = 0;
    int code = 0x81276345;

    printf("请输入用户名:");
    scanf_s("%s", szName, 20);

    NameLen = strlen(szName);
    for (int i = 0; i < NameLen; i++)
    {
        code += szName[i];
        code = code ^ (i << 8);
        code = code * ((~(NameLen * i)) * (i + 1));
    }

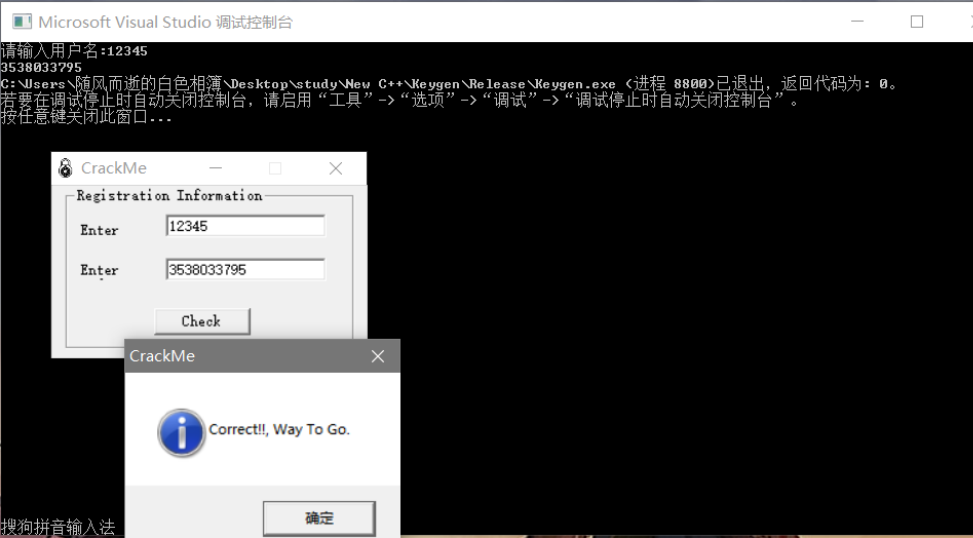
    printf("%lu", code);
    return 0;
}

int main(int argc, char* argv[])
{
    Key();
}
```

0

推荐

```
Key();
return 0;
}
```



相关文件在我的 Github: <https://github.com/UnreachableLove/160-Crackme/tree/master/Crackme019>

2019-09-20 19:27:13

分类: 160 Crackme

好文要顶

关注我

收藏该文

[随风而逝的白色相簿](#)
[关注 - 4](#)
[粉丝 - 1](#)

« 上一篇: [Crackme018](#)

posted @ 2019-09-20 19:27 随风而逝的白色相簿 阅读(1) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

发表评论

昵称: 随风而逝的白色相簿

评论内容:

提交评论

退出

[Ctrl+Enter快捷键提交]

0

推荐

