

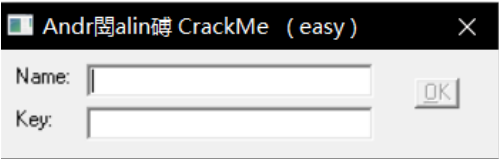
# 随风而逝的白色相簿

博客园    首页    新随笔    联系    订阅    管理    随笔 - 3   文章 - 0   评论 - 0

## Crackme009

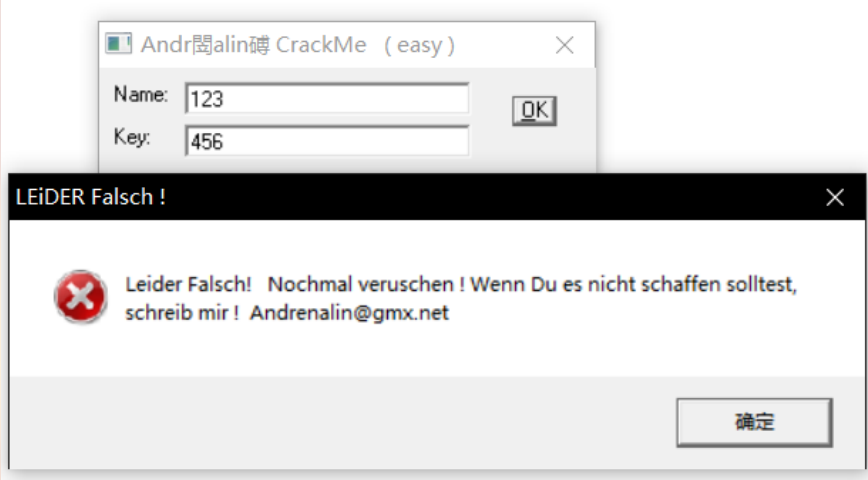
### Crackme009 的逆向分析

#### 1.程序观察



009 和 008 相比，多出来了一个输入 Name 的输入框。

分别输入 name 和 key，可以看到错误提示：



#### 2.简单查壳

#### 公告

昵称： 随风而逝的白色相簿  
园龄： 1年8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(3)

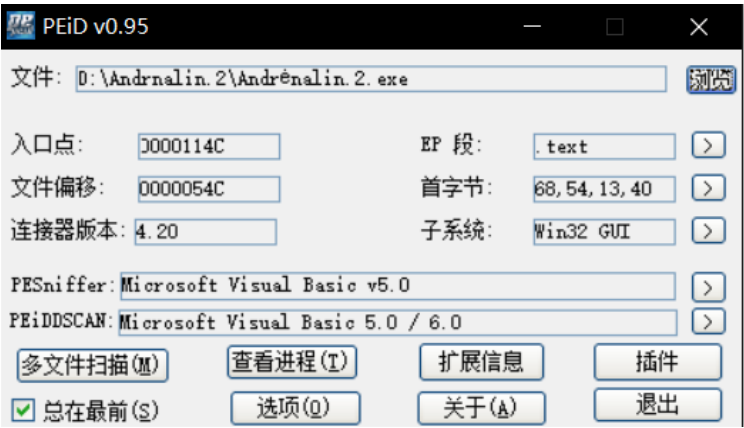
#### 随笔档案

2019年9月(3)  
2018年10月(1)

#### 阅读排行榜

1. PHP一句话木马(6391)





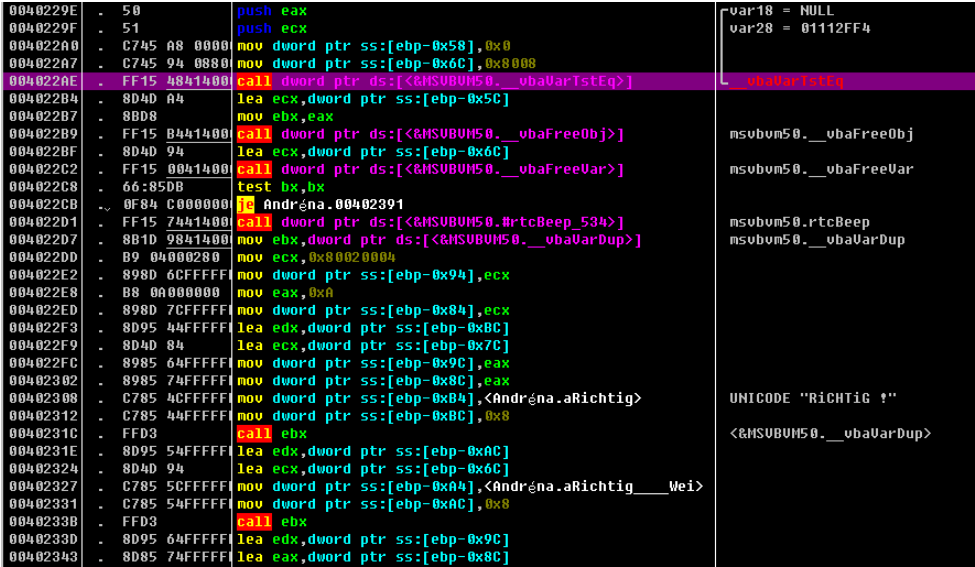
可以看到，程序没有壳，和 008 一样还是使用 VB 编写的。

3.程序分析

使用 OD 载入程序，搜索字符串

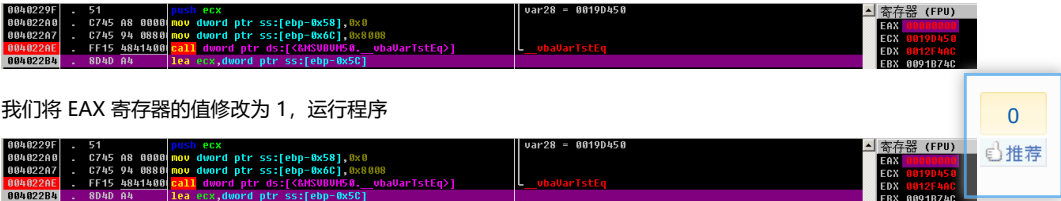


双击跟进程序，向上查找，和 008 一样

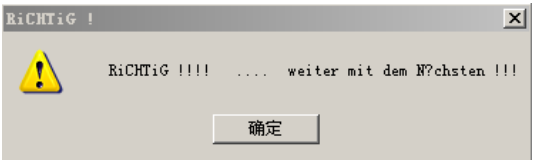


可以看到有一个比较的函数，一个 test 语句加一个跳转指令，还有正确提示的字符串。

我们在比较函数处下断点，运行程序，程序断在了断点处。  
继续运行程序，可以看到 比较函数运行完，EAX 寄存器的值为 0



我们将 EAX 寄存器的值修改为 1，运行程序



可以看到，程序出现正确提示，可以推测该比较函数就是关键点。

再次运行到该函数，查看参数

```
0012F3A0 0012F4AC | var28 = 0012F4AC
0012F3A4 0012F474 | var18 = 0012F474
```

在这里有一个关于 VB 的知识点，不知道的话这个程序很难分析出来。  
VB 的变量都类似于一个结构体。变量的前 8 个字节，存放的是该变量的数据类型信息。第 9 个字节开始存放的是该变量的真正地址。也就是说“首地址+8”才是该变量的真正地址。

分别查看两个参数：

第一个参数

```
0012F474 08 00 00 00 | E8 F4 12 00 | F4 2F 11 01 | F2 8E 45 42 | 桔.?.EB
```

```
01112FF4 34 00 35 00 | 36 00 00 00 | 00 00 00 00 | 00 06 03 00 | 4.5.6.....
```

第二个参数

```
0012F4AC 08 00 00 00 | D2 07 05 00 | D4 A7 0E 01 | 0C F3 12 00 | ...??.f?.
```

```
010EA7D4 31 00 38 00 | 35 00 2D 00 | 38 00 35 00 | 31 00 38 00 | 1.8.5.-.8.5.1.8.
010EA7E4 2D 00 35 00 | 30 00 30 00 | 00 00 00 00 | 04 00 05 00 | -.5.0.0.....
```

可以看出来，第一个参数是我们输入的 key。经过测试，发现第二个参数就是正确的 key。

那么这个 key 是哪里来的呢？我们向上找。

发现了一个循环

0

推荐

004020EC	-	8D55 BC	lea edx,dword ptr ss:[ebp-0x44]	
004020EF	-	51	push ecx	
004020F0	-	8D45 94	lea eax,dword ptr ss:[ebp-0x6C]	Step8 = 0012F4AC
004020F3	-	BB 02000000	mov ebx,0x2	
004020F8	-	52	push edx	var18 = 00180000
004020F9	-	50	push eax	retBuffer8 = 0012F474
004020FA	-	899D 54FFFFFF	mov dword ptr ss:[ebp-0xAC],ebx	
00402100	-	899D 44FFFFFF	mov dword ptr ss:[ebp-0x8C],ebx	
00402106	-	FF15 18414000	call dword ptr ds:[<RMSUBUM50.__vbaLenVar>]	计算出用户名的长度
0040210C	-	8D8D 44FFFFFF	lea ecx,dword ptr ss:[ebp-0x8C]	
00402112	-	50	push eax	End8 = 0012F474
00402113	-	8D95 E8FFFFFF	lea edx,dword ptr ss:[ebp-0x118]	
00402119	-	51	push ecx	Start8 = 0012F4AC
0040211A	-	8D85 F8FFFFFF	lea eax,dword ptr ss:[ebp-0x108]	
00402120	-	52	push edx	THMPend8 = 00180000
00402121	-	8D4D DC	lea ecx,dword ptr ss:[ebp-0x24]	
00402124	-	50	push eax	THMPstep8 = 0012F474
00402125	-	51	push ecx	Counter8 = 0012F4AC
00402126	-	FF15 20414000	call dword ptr ds:[<RMSUBUM50.__vbaVarForInit>]	__vbaVarForInit
0040212C	-	8B3D 04414000	mov edi,dword ptr ds:[<RMSUBUM50.__vbaFreeVarList>]	msubum50.__vbaFreeVarList
00402132	>	85C0	test eax,ecx	根据字符串长度进行循环
00402134	-	0F84 9C000000	js Andrgna.004021D6	
0040213A	-	8D55 94	lea edx,dword ptr ss:[ebp-0x6C]	
0040213D	-	8D45 DC	lea eax,dword ptr ss:[ebp-0x24]	
00402140	-	52	push edx	
00402141	-	50	push eax	
00402142	-	C745 9C 0100	mov dword ptr ss:[ebp-0x64],0x1	
00402149	-	895D 04	mov dword ptr ss:[ebp-0x6C],ebx	
0040214C	-	FF15 90414000	call dword ptr ds:[<RMSUBUM50.__vbaI4Var>]	msubum50.__vbaI4Var
00402152	-	8D4D DC	lea ecx,dword ptr ss:[ebp-0x44]	
00402155	-	50	push eax	Start = 0x12F474
00402156	-	8D55 84	lea edx,dword ptr ss:[ebp-0x7C]	
00402159	-	51	push ecx	dString8 = 0012F4AC
0040215A	-	52	push edx	RetBUFFER = 00180000
0040215B	-	FF15 38414000	call dword ptr ds:[<RMSUBUM50.#rtcMidCharVar_632>]	依次取单个字符
00402161	-	8D45 84	lea ecx,dword ptr ss:[ebp-0x7C]	
00402164	-	8D4D A8	lea ecx,dword ptr ss:[ebp-0x58]	

00402167	-	50	push eax	String8 = 0012F474
00402168	-	51	push ecx	ARG2 = 0012F4AC
00402169	-	FF15 70414000	call dword ptr ds:[<RMSUBUM50.__vbaStrVarVal>]	__vbaStrVarVal
0040216F	-	50	push eax	String = ""
00402170	-	FF15 0C414000	call dword ptr ds:[<RMSUBUM50.#rtcAnsiValueBstr_516>]	rtcAnsiValueBstr
00402176	-	66:8985 4CFF	mov word ptr ss:[ebp-0x84],ax	ax = ascii()
0040217D	-	8D55 CC	lea edx,dword ptr ss:[ebp-0x34]	
00402180	-	8D85 44FFFFFF	lea eax,dword ptr ss:[ebp-0x8C]	
00402186	-	52	push edx	var18 = 00180000
00402187	-	8D8D 74FFFFFF	lea ecx,dword ptr ss:[ebp-0x8C]	
0040218D	-	50	push eax	var28 = 0012F474
0040218E	-	51	push ecx	saveto8 = 0012F4AC
0040218F	-	899D 44FFFFFF	mov dword ptr ss:[ebp-0x8C],ebx	
00402195	-	FF15 94414000	call dword ptr ds:[<RMSUBUM50.__vbaVarAdd>]	__vbaVarAdd
00402198	-	8B00	mov edx,ecx	
0040219D	-	8D4D CC	lea ecx,dword ptr ss:[ebp-0x34]	
004021A0	-	FFD6	call esi	msubum50.__vbaVarMove
004021A2	-	8D4D A8	lea ecx,dword ptr ss:[ebp-0x58]	
004021A5	-	FF15 B8414000	call dword ptr ds:[<RMSUBUM50.__vbaFreeStr>]	msubum50.__vbaFreeStr
004021AB	-	8D55 84	lea edx,dword ptr ss:[ebp-0x7C]	
004021AE	-	8D45 94	lea eax,dword ptr ss:[ebp-0x6C]	
004021B1	-	52	push edx	
004021B2	-	50	push eax	
004021B3	-	53	push ebx	
004021B4	-	FFD7	call edi	msubum50.__vbaFreeVarList
004021B6	-	83C4 0C	add esp,0xC	
004021B9	-	8D8D E8FFFFFF	lea ecx,dword ptr ss:[ebp-0x118]	
004021BF	-	8D95 F8FFFFFF	lea edx,dword ptr ss:[ebp-0x108]	
004021C5	-	8D45 DC	lea eax,dword ptr ss:[ebp-0x24]	
004021C8	-	51	push ecx	THMPend8 = 0012F4AC
004021C9	-	52	push edx	THMPstep8 = 00180000
004021CA	-	50	push eax	Counter8 = 0012F474
004021CB	-	FF15 AC414000	call dword ptr ds:[<RMSUBUM50.__vbaVarForNext>]	__vbaVarForNext
004021D1	-	E9 5CFFFFFF	jmp Andrgna.00402132	
004021D6	>	8D4D CC	lea ecx,dword ptr ss:[ebp-0x34]	
004021D9	-	8D95 54FFFFFF	lea edx,dword ptr ss:[ebp-0xAC]	

这个循环其实很简单：

1. 首先计算出我们输入的 name 的长度
2. name 的长度就是循环的次数
3. 依次取 name 的字符：第一次取第一个字符，第二次取第二个字符
4. 将字符转化为 ascii 码
5. 然后相加

循环结束，程序又对循环得到的值进行了加工



0040210F	. 51	push ecx	var18 = 0012F4AC
004021E0	. 8D45 94	lea eax, dword ptr ss:[ebp-0x6C]	
004021E3	. 52	push edx	var28 = 0012F434
004021E4	. 50	push eax	SaveTo8 = 0012F4AC
004021E5	. C785 5CFFFFFF	mov dword ptr ss:[ebp-0xA4], 0x499602D2	
004021EF	. C785 54FFFFFF	mov dword ptr ss:[ebp-0xA0], 0x3	
004021F9	. FF15 5C414000	call dword ptr ds:[<&MSUBUH50.__vbaUVarMul>]	两数相乘
004021FF	. 8B00	mov edx, eax	
00402201	. 8D4D CC	lea ecx, dword ptr ss:[ebp-0x34]	
00402204	. FF06	call esi	msubum50.__vbaUVarMove
00402206	. 8B1D A0414000	mov ebx, dword ptr ds:[<&MSUBUH50.__vbaMidStntVar>]	msubum50.__vbaMidStntVar
0040220C	. 8D4D CC	lea ecx, dword ptr ss:[ebp-0x34]	
0040220F	. 51	push ecx	
00402210	. 6A 04	push 0x4	
00402212	. 8D95 54FFFFFF	lea edx, dword ptr ss:[ebp-0xA0]	
00402218	. 6A 01	push 0x1	
0040221A	. 52	push edx	
0040221B	. C785 5CFFFFFF	mov dword ptr ss:[ebp-0xA4], Andrena.00401C34	UNICODE "--"
00402225	. C785 54FFFFFF	mov dword ptr ss:[ebp-0xA0], 0x8	
0040222F	. FF03	call ebx	替换第4位; <&MSUBUH50.__vbaMidStntVar>
00402231	. 8D45 CC	lea eax, dword ptr ss:[ebp-0x34]	
00402234	. 8D8D 54FFFFFF	lea ecx, dword ptr ss:[ebp-0xA0]	
0040223A	. 50	push eax	
0040223B	. 6A 09	push 0x9	
0040223D	. 6A 01	push 0x1	
0040223F	. 51	push ecx	
00402240	. C785 5CFFFFFF	mov dword ptr ss:[ebp-0xA4], Andrena.00401C34	UNICODE "--"
0040224A	. C785 54FFFFFF	mov dword ptr ss:[ebp-0xA0], 0x8	
00402254	. FF03	call ebx	替换第9位

程序先把得到的值，和 0x499602D2 进行相乘

然后依次使用 "-" 替换掉了计算结果的第4位和第9位，最后得到的值就是真正的 Key。

#### 4. 写出注册机

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

int Key()
{
    char szName[20] = { 0 };
    char szKey[30] = { 0 };
    int NameLen = 0;
    __int64 Result = 0;

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

    NameLen = strlen(szName);
    for (int i = 0; i < NameLen; i++)
    {
        Result += szName[i];
    }
    Result *= 1234567890;

    sprintf(szKey, "%I64d", Result);
    szKey[3] = '-';
    szKey[8] = '-';

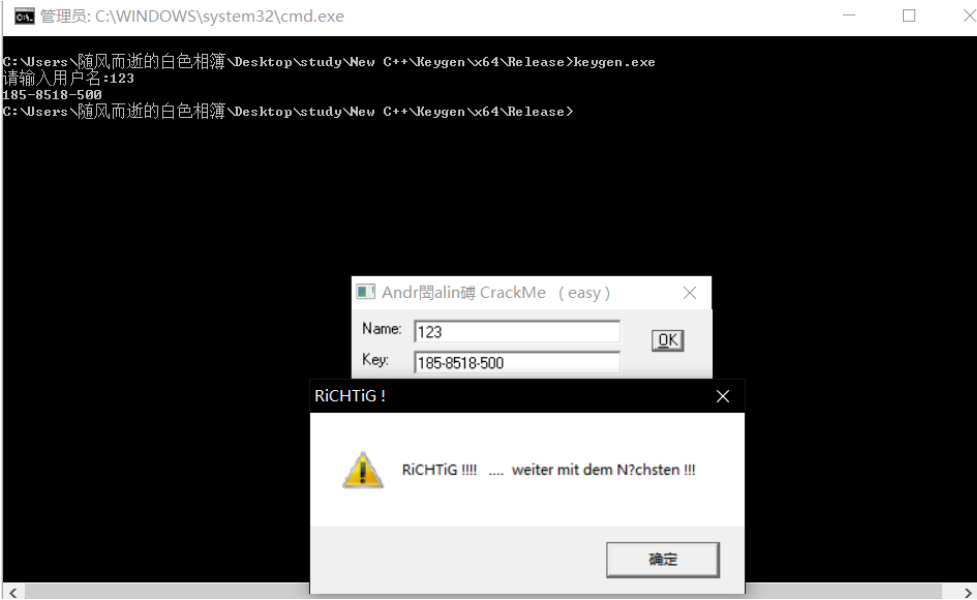
    printf("%s", szKey);

    return 0;
}

int main(int argc, char* argv[])
{
    Key();
    return 0;
}
```

0

推荐



相关文件在我的 Github: <https://github.com/UnreachableLove/160-Crackme/tree/master/Crackme009>

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: [Crackme008](#)

posted @ 2019-09-09 20:06 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

