

查壳  
分析程序  
校验结果  
写出注册机

## 查壳



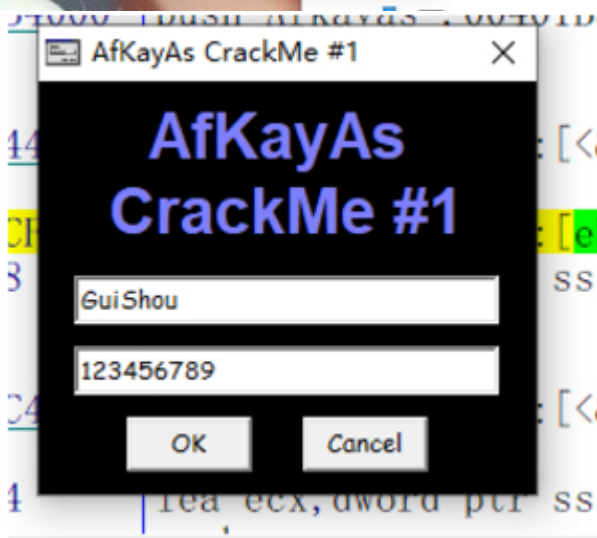
程序使用Virtual Basic写的 无壳

## 分析程序



程序是使用用户名和序列号的方式加密 载入OD 随便输入一个

用户名和密码



来到第一处关键校验处 校验过程如下

00402402	. 50	push eax	
00402403	. FF15 04414000	call dword ptr ds:[<&MSVBVM50. __vbaHresu	msvbvm50. __vbaHresultCh
00402409	> 8B95 50FFFFFF	mov edx,dword ptr ss:[ebp-0xB0]	
0040240F	. 8B45 E4	mov eax,dword ptr ss:[ebp-0x1C]	
00402412	. 50	push eax	String = " ! □ 蛭啖?脩 "
00402413	. 8B1A	mov ebx,dword ptr ds:[edx]	
00402415	. FF15 E4404000	call dword ptr ds:[<&MSVBVM50. __vbaLenB	求用户名的长度
0040241B	. 8BF8	mov edi,eax	edi=用户名长度
0040241D	. 8B4D E8	mov ecx,dword ptr ss:[ebp-0x18]	ecx=用户名
00402420	. 69FF FB7C0100	imul edi,edi,0x17CFB	edi=用户名长度*0x17CFB
00402426	. 51	push ecx	String = "h?@"
00402427	. 0F80 91020000	jo Afkayas_.004026BE	溢出跳转 长度不能太长
0040242D	. FF15 F8404000	call dword ptr ds:[<&MSVBVM50. #rtcAnsiVa	返回第一个字符的ASCII
00402433	. 0FBFD0	movsx edx,ax	
00402436	. 03FA	add edi,edx	edi+第一个字符的ASCII
00402438	. 0F80 80020000	jo Afkayas_.004026BE	
0040243E	. 57	push edi	Afkayas_.<ModuleEntryPc
0040243F	. FF15 E0404000	call dword ptr ds:[<&MSVBVM50. __vbaStrI4	十六进制转十进制
00402445	. 8BD0	mov edx,eax	
00402447	. 8D4D E0	lea ecx,dword ptr ss:[ebp-0x20]	
0040244A	. FF15 70414000	call dword ptr ds:[<&MSVBVM50. __vbaStrM	字符串拷贝
00402450	. 8BBD 50FFFFFF	mov edi,dword ptr ss:[ebp-0xB0]	
00402456	. 50	push eax	
00402457	. 57	push edi	Afkayas_.<ModuleEntryPc
00402458	. FF93 A4000000	call dword ptr ds:[ebx+0xA4]	
0040245E	. 85C0	test eax,eax	
00402460	. 7D 12	jge short Afkayas_.00402474	
00402462	. 68 A4000000	nush 0xA4	

ds:[004040E4]=0F00BD80 (msubvm50. \_\_vbaLenBstr)

吾爱破解 - Afkayas.1.Exe - [LCG - 主线程 模块 - Afkayas]

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->

00402510	> 8B45 E8	mov eax,dword ptr ss:[ebp-0x18]	123456789
00402513	. 8B4D E4	mov ecx,dword ptr ss:[ebp-0x1C]	682788
00402516	. 8B3D 00414000	mov edi,dword ptr ds:[<&MSVBVM50. __vbaSt	msvbvm50. __vbaStrCat
0040251C	. 50	push eax	
0040251D	. 68 701B4000	push Afkayas_.00401B70	AKA-
00402522	. 51	push ecx	String = "h?@"
00402523	. FFD7	call edi	__vbaStrCat
00402525	. 8B1D 70414000	mov ebx,dword ptr ds:[<&MSVBVM50. __vbaSt	msvbvm50. __vbaStrMove
0040252B	. 8BD0	mov edx,eax	
0040252D	. 8D4D E0	lea ecx,dword ptr ss:[ebp-0x20]	
00402530	. FFD3	call ebx	<&MSVBVM50. __vbaStrMove
00402532	. 50	push eax	AKA-682788
00402533	. FF15 28414000	call dword ptr ds:[<&MSVBVM50. __vbaStrC	msvbvm50. __vbaStrCmp
00402539	. 8BF0	mov esi,eax	
0040253B	. 8D55 E0	lea edx,dword ptr ss:[ebp-0x20]	
0040253E	. F7DE	neg esi	Afkavas_.<ModuleEntryPc

堆栈 ss:[0019FF68]=00000000  
eax=0019FFCC

1. 求出了用户名的长度 2. 将用户名长度乘以0x17CFB 得到结果 如果溢出则跳转 3. 将结果再加上用户名的第一个字符的ASCII 4. 将结果转为十进制 5. 将结果和AKA进行拼接 得到最后的序列号

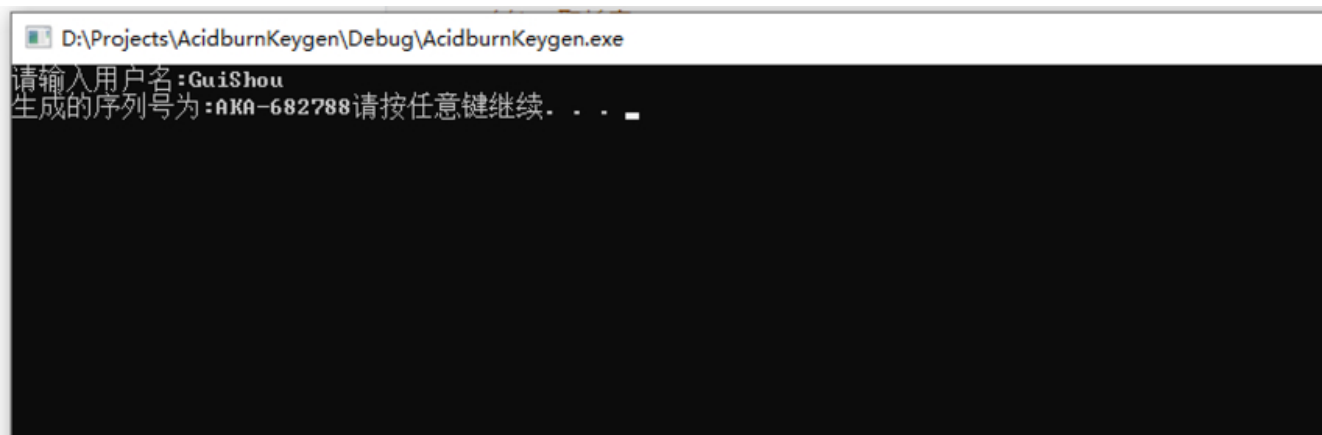
## 校验结果



输入我们计算的一组序列号 注册成功

## 写出注册机

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include <stdlib.h>
#include <windows.h>
int main()
{
    //密码
    char key1[4] = "AKA";
    //输入用户名
    char username[10] = { 0 };
    printf("请输入用户名:");
    scanf_s("%s", username, 10);
    //1\ 取长度
    int iUsernameLen = strlen(username);
    //2\ 将用户名长度乘以0x17CFB 得到结果
    int result = iUsernameLen * 0x17CFB;
    //3\ 将结果再加上用户名的第一个字符的ASCII
    result = result + username[0];
    //4\ 将结果转为十进制 此步骤省略
    //5\ 拼接序列号
    char key[MAX_PATH] = { 0 };
    sprintf(key, "%s-%d", key1, result);
    //打印序列号
    printf("生成的序列号为:%s\n", key);
    system("pause");
    return 0;
}
```



至此 002分析完成 需要源码的可以到我的Github下载: <https://github.com/TonyChen56/160-Crackme>