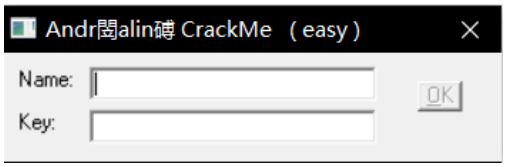
随风而逝的白色相簿

Crackme009

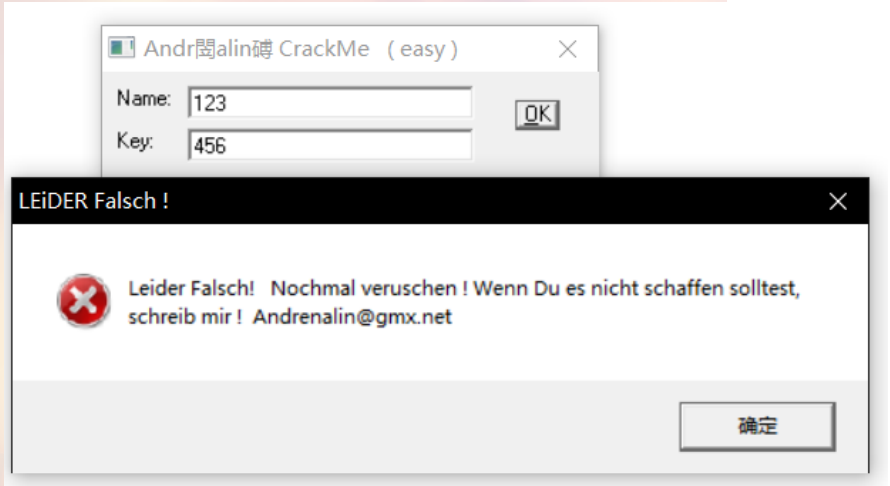
Crackme009 的逆向分析

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：

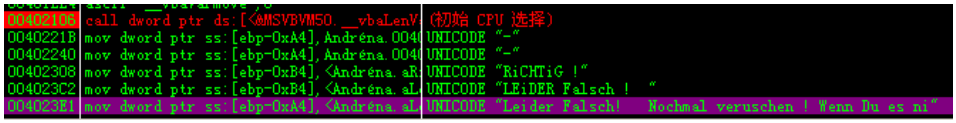


2.简单查壳

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

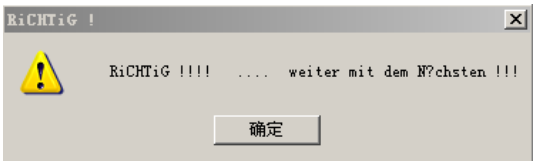


双击跟进程序，向上查找，和 008 一样

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

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：

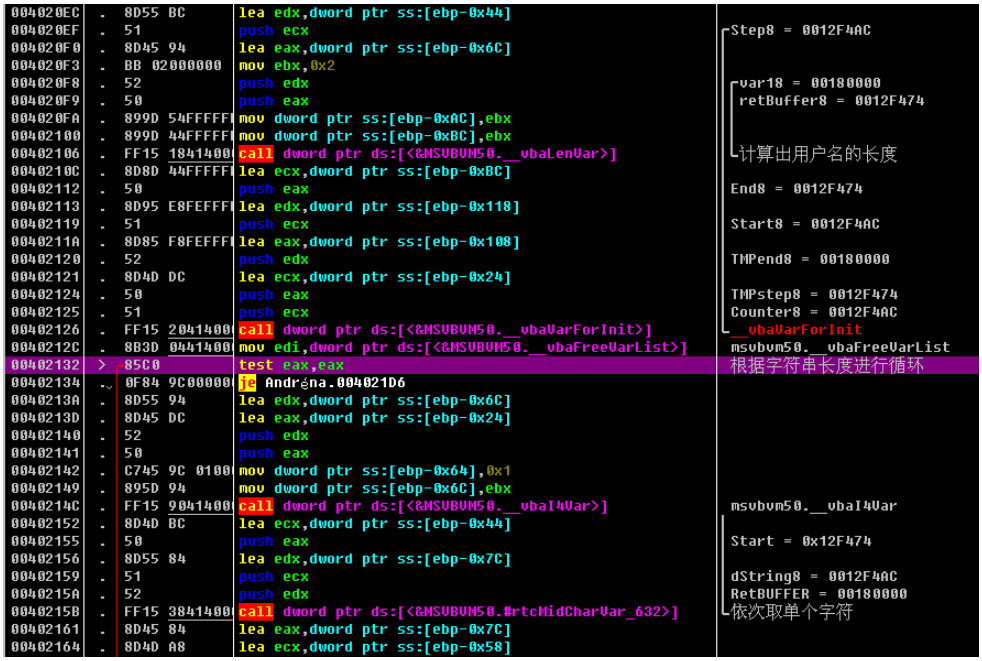
第一个参数

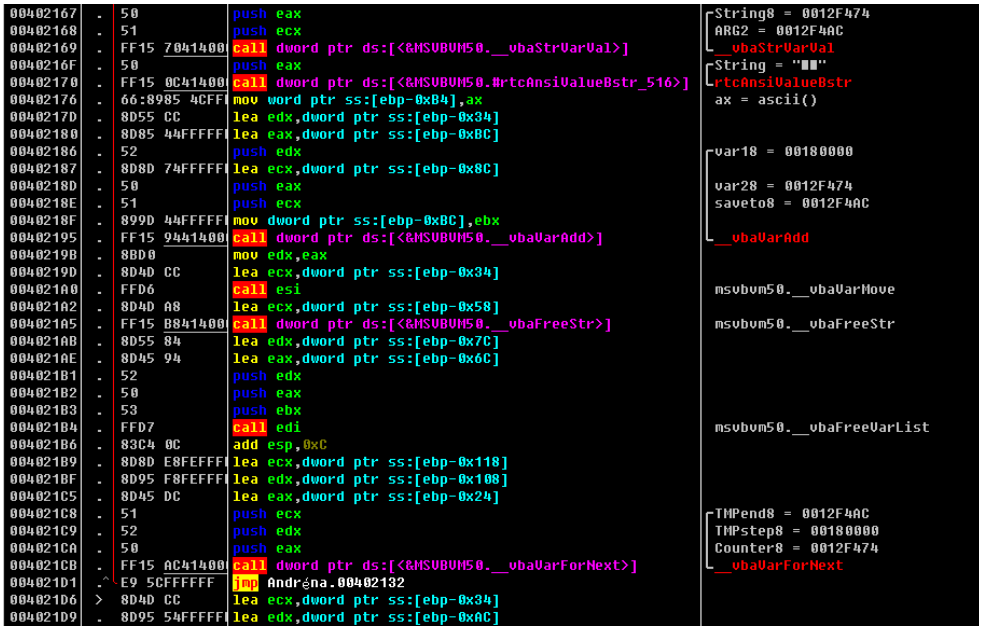
第二个参数

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

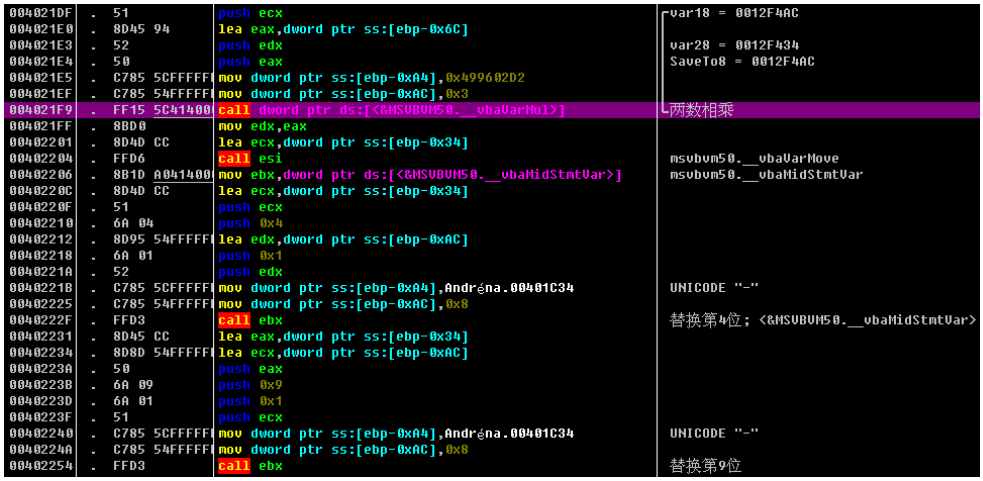




这个循环其实很简单：

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

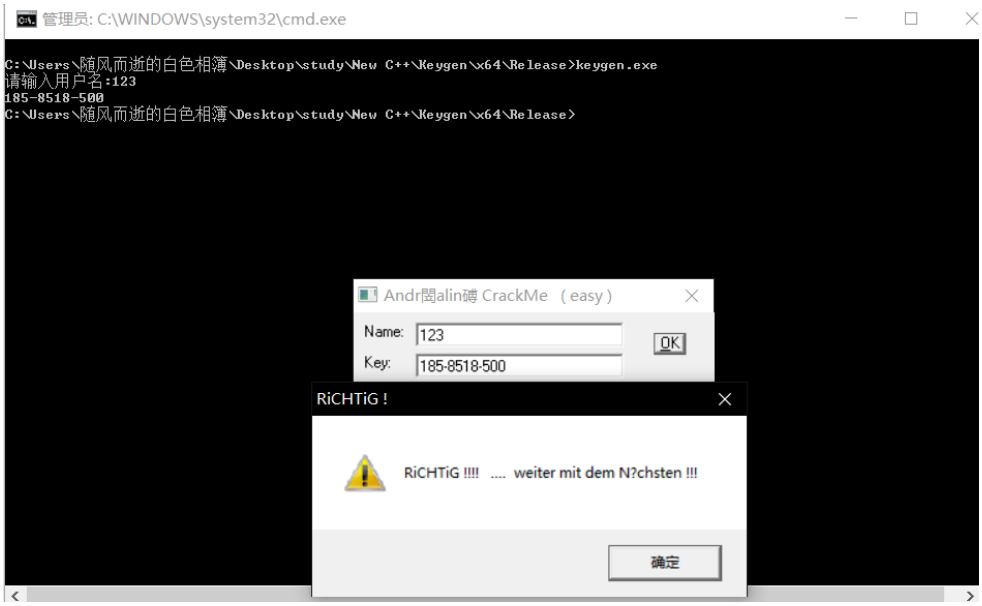
循环结束，程序又对循环得到的值进行了加工



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;
}
```



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