

## 02 题- Afkayas.1.Exe

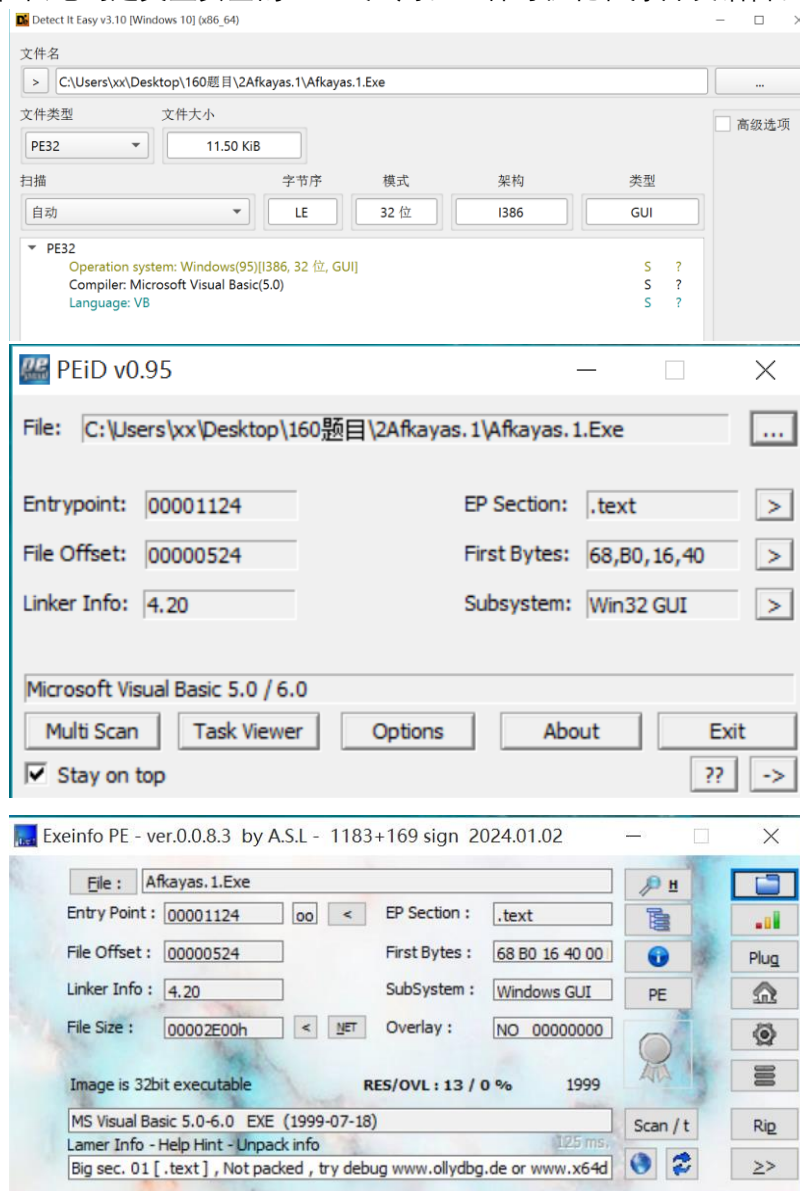
文件信息：32 位 、VB 语言编写、 无壳

法一：逆向出序列号生成算法

法二：修改汇编指令绕过验证

1、查看 exe 文件信息，结论：无壳，VB 语言编写。

Visual Basic 是 Microsoft 开发的一种面向对象的编程语言。使用 Visual Basic 即可快速、轻松地创建类型安全的 .NET 应用。一种可视化程序开发语言。



2、双击运行报错，查原因，下载安装。

msvbvm50.dll 文件的核心功能是提供 Visual Basic 5.0 编译的应用程序在运行时所需的基础代码支持。它包含了 Visual Basic 虚拟机的实现，使得 VB 5.0 编写的程序能够在没有 Visual Basic 运行时环境的情况下运行。



网上下载 dll 放到 exe 同目录：

扩展：

以下是正确的 DLL 查找路径顺序：

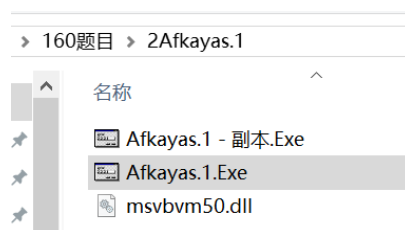
**正常模式下的查找顺序：**

1. 应用程序所在目录
2. 系统目录（GetSystemDirectory 返回的目录，通常是系统盘\Windows\System32）
3. 16 位系统目录（为了向前兼容，可以不考虑）
4. Windows 目录（GetWindowsDirectory 返回的目录，通常是系统盘\Windows）
5. 当前目录（GetCurrentDirectory 返回的目录）
6. 环境变量 PATH 中所有目录

**安全 DLL 查找模式下的查找顺序：**

1. 应用程序所在目录
2. 系统目录（GetSystemDirectory 返回的目录，通常是系统盘\Windows\System32）
3. Windows 目录（GetWindowsDirectory 返回的目录，通常是系统盘\Windows）
4. 16 位系统目录（为了向前兼容，可以不考虑）
5. 环境变量 PATH 中所有目录

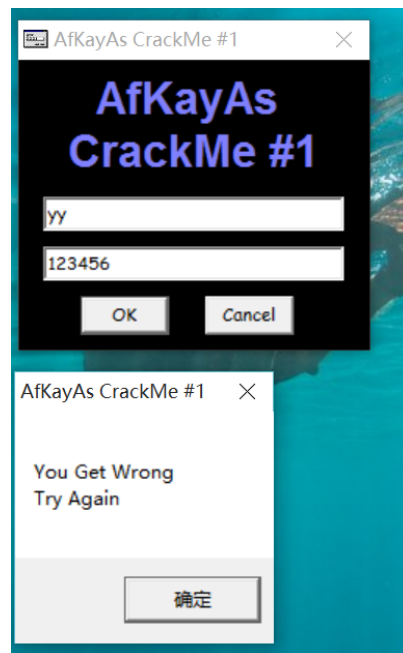
在安全模式下，当前目录是被排除的。



2、解决 dll 缺少问题后，双击 exe 运行起来。界面需要输入用户名+序列号。



4、随便输入一组数据，结果报错：



5、X64dbg 调试程序，发现很多 VB 函数：找关键的打断点，缩小分析范围

- `__vbaStrCat`：用于连接字符串。它通常用于处理字符串连接的细节，确保在合并字符串时正确管理内存和数据类型。
- `__vbaStrMove`：用于在内存中移动字符串的内容。
- `__vbaLenBstr`：获取字符串长度
- `__vbaStrCmp`：比较字符串
- `StrComp`：比较两个字符串，并返回整数值指示它们的相对顺序。可以指定比较的方式（区分大小写或不区分）。

VB (Visual Basic)：

是一种独立的编程语言和开发环境，用于创建 Windows 应用程序。VB 程序是独立的可执行文件 (.exe)，可以在 Windows 系统上运行。

简单来说，VBA 用于 Office 应用程序的宏和自动化，而 VB 用于开发独立的 Windows 应用程序。尽管它们的语法有相似之处，但它们的用途和环境是不同的。

## 法一：逆向出序列号生成算法

6、利用调试器的字符串搜索定位上面关键函数，并 F2 打上断点，调试分析，注意看输入数据传递过程。

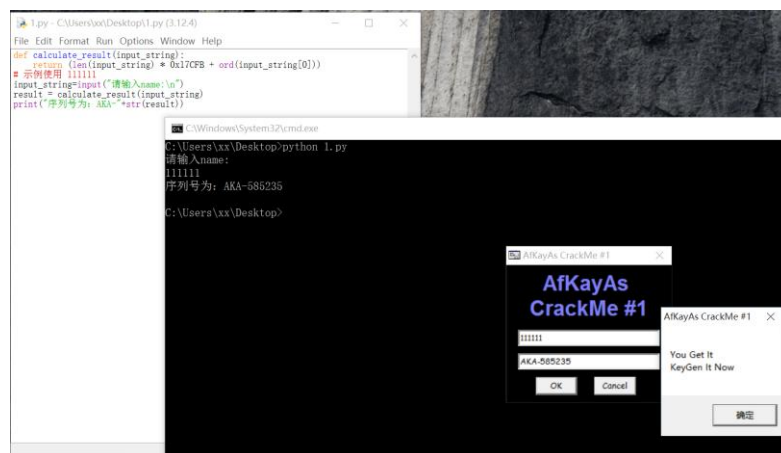


逆向出序列号生成算法：

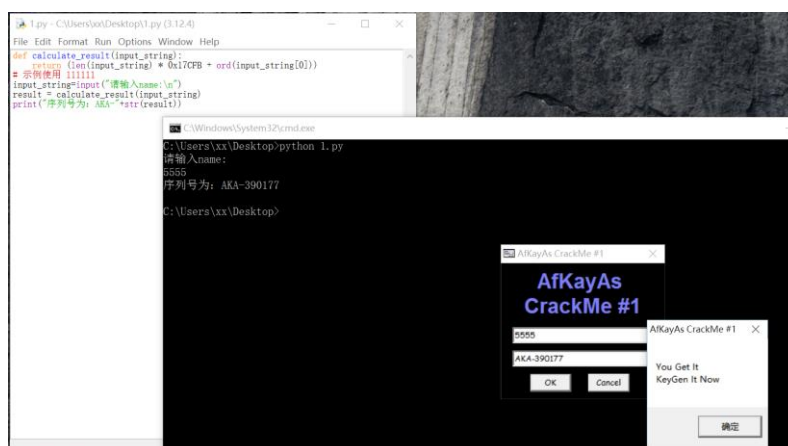
取 name 长度 \* 常量 + name 第一字符，结果转 10 进制。

```
python 脚本：
def calculate_result(input_string):
    return (len(input_string) * 0x17CFB + ord(input_string[0]))
# 示例使用 111111
input_string=input("请输入 name:\n")
result = calculate_result(input_string)
print("序列号为：AKA-"+str(result))
```

**踩坑：** print("序列号为：AKA-",result)输出结果：字符串和可变量之间有个空格，原因是此句中是打印两个参数（序列号的字符串和 int 型变量）中间用空格分隔，并不是输出一个完整的序列号。



多试一下：逆向出的序列号生成算法成功。



## 法二：修改汇编指令绕过验证

1、通过字符串搜索定位"You Get It"，发现有个 test 判断+je 跳转，在验证界面输入的不正确的用户名和序列号都是会之间跳转到 4025E5 地址去，为了使此处不跳转，采取直接使用 nop 将 je 指令替换，不执行该指令。

在汇编语言中，nop (No Operation) 是一条空操作指令，它的作用是不做任何操作，仅仅是占用一个时钟周期。

Afkayas.1.Exe - PID: 4524 - 模块: afkayas.1.exe - 线程: 主线程 3328 - x32dbg

文件(F) 视图(V) 调试(D) 跟踪(N) 插件(P) 收藏夹(I) 选项(O) 帮助(H) Aug 28 2024 (TitanEngine)

CPU 日志 笔记 断点 内存布局 调用堆栈 SEH链 脚本 符号 源代码

地址	汇编指令	注释
00402576	mov dword ptr ss:[ebp-64],ecx	
00402579	test si,si	检测si中是否=0，若si=0则代码执行跳到4025E5处
0040257C	mov dword ptr ss:[ebp-6C],eax	
0040257F	mov dword ptr ss:[ebp-54],ecx	
00402582	mov dword ptr ss:[ebp-5C],eax	
00402585	mov dword ptr ss:[ebp-44],ecx	
00402588	mov dword ptr ss:[ebp-4C],eax	
0040258B	je afkayas.1.4025E5	
0040258D	push afkayas.1.401880	401880:L"You Get It"
00402592	push afkayas.1.40189C	40189C:L"\r\n"
00402597	call edi	edi:__vbaStrCat
00402599	mov edx,eax	
0040259B	lea ecx,dword ptr ss:[ebp-18]	
0040259E	call ebx	ebx:__vbaStrMove
004025A0	push eax	
004025A1	push afkayas.1.4018A8	4018A8:L"KeyGen It Now"
004025A6	call edi	edi:__vbaStrCat
004025A8	lea ecx,dword ptr ss:[ebp-6C]	
004025AB	mov dword ptr ss:[ebp-34],eax	
004025AE	lea edx,dword ptr ss:[ebp-5C]	
004025B1	push ecx	
004025B2	lea eax,dword ptr ss:[ebp-4C]	
004025B5	push edx	
004025B6	push eax	
004025B7	lea ecx,dword ptr ss:[ebp-3C]	
004025BA	push 0	
004025BC	push ecx	
004025BD	mov dword ptr ss:[ebp-3C],8	
004025C4	call dword ptr ds:[<Ordinal#595>]	
004025CA	lea ecx,dword ptr ss:[ebp-18]	
004025CD	call dword ptr ds:[<_vbaFreeStr>]	
004025D3	lea edx,dword ptr ss:[ebp-6C]	
004025D6	lea eax,dword ptr ss:[ebp-5C]	
004025D9	push edx	
004025DA	lea ecx,dword ptr ss:[ebp-4C]	
004025DD	push eax	
004025DE	lea edx,dword ptr ss:[ebp-3C]	
004025E1	push ecx	
004025E2	push edx	
004025E3	jmp afkayas.1.40263B	
004025E5	push afkayas.1.4018C8	4018C8:L"You Get Wrong"
004025EA	push afkayas.1.40189C	40189C:L"\r\n"
004025EF	call edi	edi:__vbaStrCat
004025F1	mov edx,eax	
004025F3	lea ecx,dword ptr ss:[ebp-18]	
004025F6	call ebx	ebx:__vbaStrMove
004025F8	push eax	
004025F9	push afkayas.1.4018E8	4018E8:L"Try Again"

Nop 指令替换 je 指令：



