

03 题- Afkayas.2.Exe

文件信息：32位、VB语言编写、无壳

法一：修改汇编指令绕过验证

法二：逆向出序列号生成算法

VBA (Visual Basic for Applications) 和 VB (Visual Basic) 虽然名字相似，但它们是不同的东西：

VBA：在 Office 应用内部开发，主要用于宏和自动化。

VB：在 Visual Studio 中开发，用于创建独立的 Windows 应用程序，VB 程序是独立的可执行文件 (.exe)。

在某些软件（比如试用版或分享版）中，开发者会加入一个“Nag Screen”（nag 是指烦扰的意思），这个窗口会提醒用户购买正式版，或者会在使用软件时不断弹出，给用户带来干扰。

“Kill the Nag”就是指破解这种机制，去掉这个烦人的提示窗口，让用户可以更顺畅地使用软件而不被打扰。简单来说，就是去掉那些让人觉得烦的购买提醒。

VB 是 Microsoft 发布的 GUI 程序编程语言。

1、MSVBVM60.dll 是编写 VB 程序的专用引擎（封装好的代码库，如 60.dll 文件中 rtcMsgBox () 的实现是基于 user32.dll 的 MessageBox()）。

2、VB 程序可由编译选项不同，而产生不同代码，分为本地代码（一般使用易于调试器解析的 IA-32 指令）和伪代码（是一种解释性语言，使用由 VB 引擎实现虚拟机并可予解析的指令（字节码））。因此，若想准确解析 VB 的伪代码，就需要分析 VB 引擎并实现模拟器。

3、VB 程序采用 windows 操作系统的事件驱动方式工作。因此要分析的用户代码不在 main 函数中，而在各事件处理程序之中。

4、间接调用：

jmp ThunRTMain()=IAT 导入地址表区域开头处-----运行顺序 (3)

push ThunRTMain 函数的参数 :EP 处压入 RT_MainStruct 结构体的地址-运行顺序 (1)

call jmp 地址---运行顺序 (2)

5、RT_MainStruct 结构体的成员是其他结构体的地址，也就是说，VB 引擎通过参数传递过来的 RT_MainStruct 结构体获取程序运行需要的所有信息。

1、检查 exe 文件信息：32 位 VB 语言编写 无壳

Detect It Easy v3.10 [Windows 10] (x86_64)

文件名
C:\Users\xx\Desktop\160题目\2Afkayas.2\AfKayAs.2.Exe

文件类型 PE32
文件大小 36.50 KiB

扫描
自动
字节序 LE
模式 32 位

PE32
Operation system: Windows(95)[I386, 32 位, GUI]
Compiler: Microsoft Visual Basic(5.0)
Language: VB

PEiD v0.95

File: C:\Users\xx\Desktop\160题目\2Afkayas.2\AfKayAs.2.Exe

Entrypoint: 00001170 EP Section: .text
File Offset: 00000570 First Bytes: 68,D4,67,40
Linker Info: 4.20 Subsystem: Win32 GUI

Microsoft Visual Basic 5.0 / 6.0

Multi Scan Task Viewer Options About Exit
 Stay on top ?? ->

2、添加 dl 到同目录后运行 exe 程序，弹出第一个窗，过一会儿才弹出第二个窗口。



法一：修改汇编指令绕过验证

3、直接修改指令

AfKayAs.2.Exe - PID: 1800 - 模块: afkayas.2.exe - 线程: 主线程 4492 - 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

CPU 日志 笔记 断点 内存布局 调用堆栈 SEH链 脚本

00408665	test si,si
00408668	mov dword ptr ss:[ebp-6C],eax
0040866B	mov dword ptr ss:[ebp-54],ecx
0040866E	mov dword ptr ss:[ebp-5C],eax
00408671	mov dword ptr ss:[ebp-44],ecx
00408674	mov dword ptr ss:[ebp-4C],eax
00408677	je afkayas.2.4086DB
00408679	mov esi,dword ptr ds:[<_vbaStrCat>]
0040867F	push afkayas.2.406FC0
00408684	push afkayas.2.406FDC
00408689	call esi
0040868B	mov edx,eax
0040868D	lea ecx,dword ptr ss:[ebp-18]
00408690	call dword ptr ds:[<_vbaStrMove>]
00408696	push eax
00408697	push afkayas.2.406FE8
00408699	call esi
0040869E	mov dword ptr ss:[ebp-34],eax
004086A1	lea eax,dword ptr ss:[ebp-6C]
004086A4	lea ecx,dword ptr ss:[ebp-5C]
004086A7	push eax
004086A8	lea edx,dword ptr ss:[ebp-4C]
004086AB	push ecx
004086AC	push edx
004086AD	lea eax,dword ptr ss:[ebp-3C]
004086B0	push 0
004086B2	push eax
004086B3	mov dword ptr ss:[ebp-3C],8
004086B4	call dword ptr ds:[406FC0+8] #50551

esi: "h舐@"
406FC0:L"You Get It"
406FDC:L"\r\n"

edx: "h舐@"

406FE8:L"KeyGen It Now"

ecx: "h舐@"
edx: "h舐@"

004086C0 L"You Get It"

Nop 替换 je 指令：

AfKayAs.2.Exe - PID: 1800 - 模块: afkayas.2.exe - 线程: 主线程 4492 - 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

00408665	test si,si
00408668	mov dword ptr ss:[ebp-6C],eax
0040866B	mov dword ptr ss:[ebp-54],ecx
0040866E	mov dword ptr ss:[ebp-5C],eax
00408671	mov dword ptr ss:[ebp-44],ecx
00408674	mov dword ptr ss:[ebp-4C],eax
00408677	nop
00408678	nop
00408679	mov esi,dword ptr ds:[<_vbaStrCat>]
0040867F	push afkayas.2.406FC0
00408684	push afkayas.2.406FDC
00408689	call esi
0040868B	mov edx,eax
0040868D	lea ecx,dword ptr ss:[ebp-18]
00408690	call dword ptr ds:[<_vbaStrMove>]
00408696	push eax
00408697	push afkayas.2.406FE8
00408699	call esi
0040869E	mov dword ptr ss:[ebp-34],eax
004086A1	lea eax,dword ptr ss:[ebp-6C]
004086A4	lea ecx,dword ptr ss:[ebp-5C]
004086A7	push eax
004086A8	lea edx,dword ptr ss:[ebp-4C]
004086AB	push ecx
004086AC	push edx
004086AD	lea eax,dword ptr ss:[ebp-3C]
004086B0	push 0
004086B2	push eax
004086B3	mov dword ptr ss:[ebp-3C],8

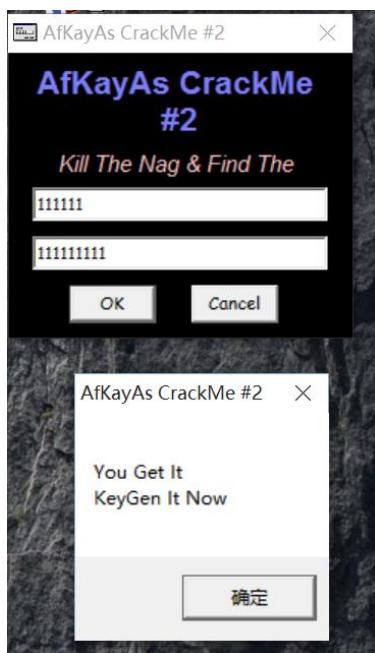
esi: "h舐@"
406FC0:L"You Get It"
406FDC:L"\r\n"

edx: "h舐@"

406FE8:L"KeyGen It Now"

ecx: "h舐@"
edx: "h舐@"

输出结果成功：



法二：逆向出序列号生成算法：

浮点（寄存器）计算：

Fadd 指令两个操作数只能是浮点栈寄存器。

fld 指令:fld 相当于 push,作用是把数据压入到浮点栈寄存器 st0 中，因为 st0 是浮点栈的栈顶，所以每次 fld 压栈，都是压入到 st0 中，然后之前 st0 寄存器的内容会向下移动到 st1 中，st1 中的内容进入 st2 中…一直到 st6 的内容进入到 st7 中,st7 之前的内容消失，被 st6 覆盖。

浮点数计算完以后从寄存器中取出存入内存的出栈操作:

fst 指令和 fstp 指令

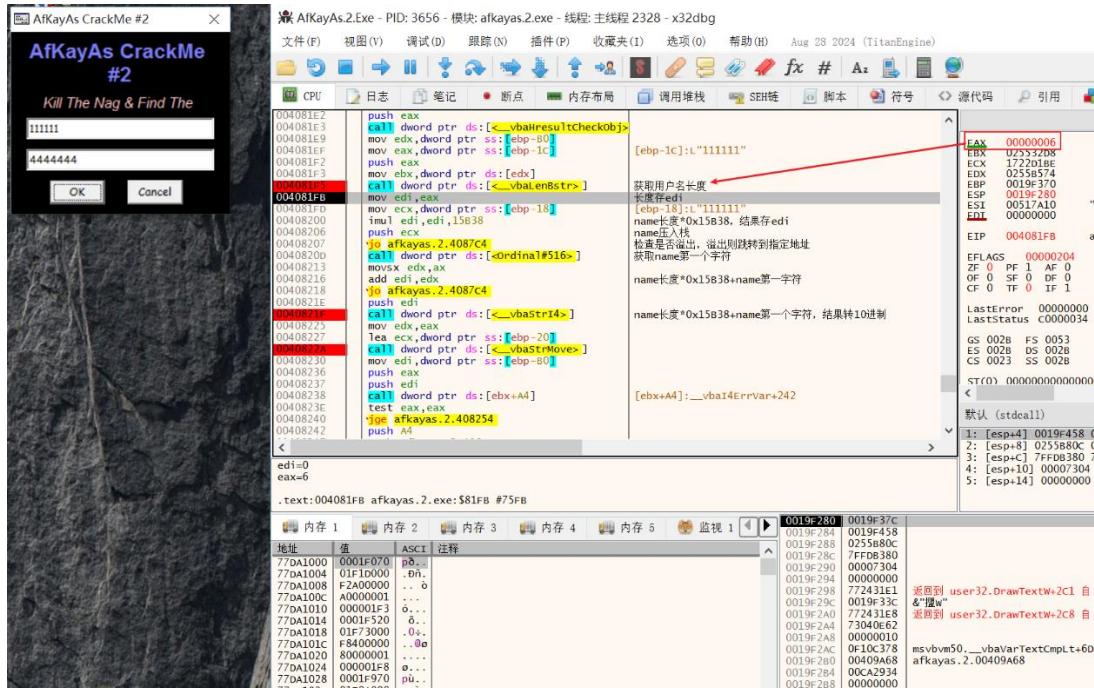
这 2 个指令的作用都是从浮点栈的栈顶 st0 取出数据存入到内存中。

区别

— fst 取出栈顶的数据后，栈顶不变

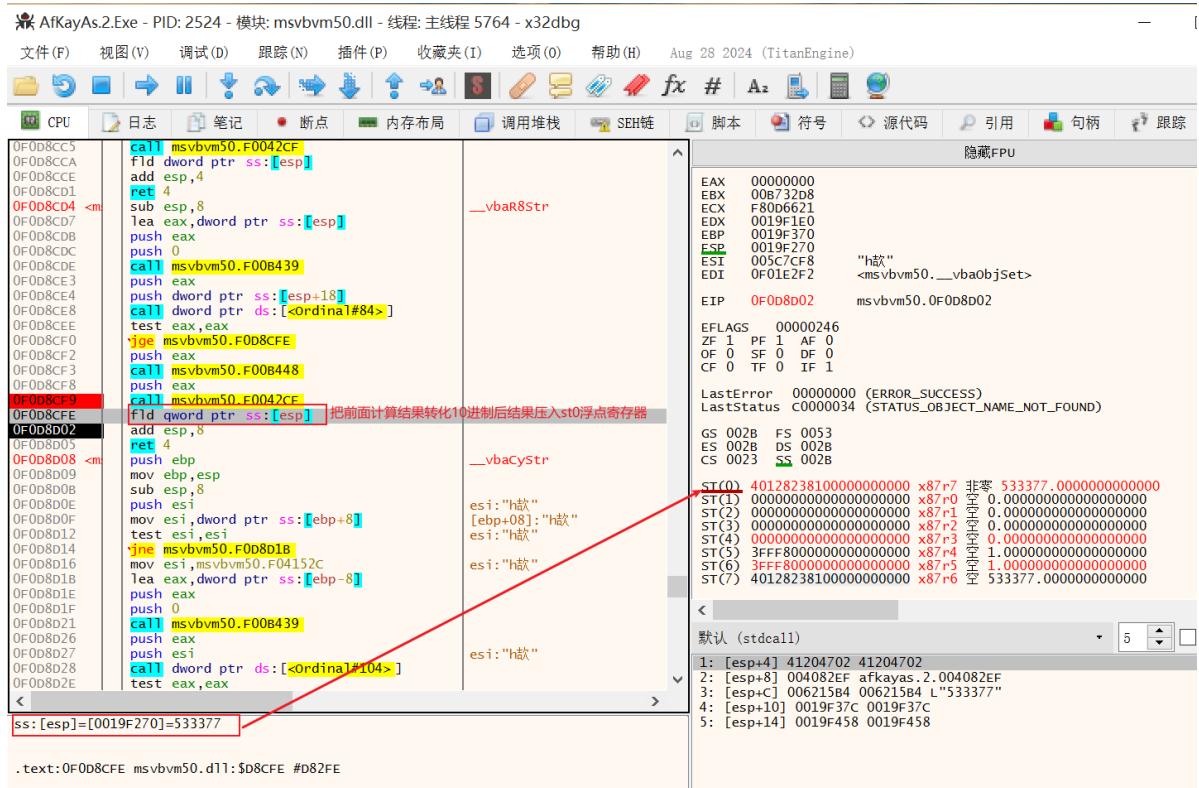
— fstp 取出栈顶的数据后，栈顶出栈，其他 st1~st7 寄存器向栈顶移动.相当于 pop 指令出栈

逐步调试分析，发现正确序列号生成与用户名有关：以下以用户名 **111111** 进行
调试分析。



结果变化: $533377 + (10/5) \Rightarrow 533379$

10 进制转换后的结果进行浮点计算：先将结果压入浮点寄存器 st0



继续压入一个内存地址中的 10 值：原 st0 的值向 st1 移动。

AfKayAs.2.Exe - PID: 2524 - 模块: afkayas.2.exe - 线程: 主线程 5764 - 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 引用 句柄 跟踪 线程

汇编视图:

```

004082D7 call dword ptr ds:[__vbaHRESULTCheckObj]
004082D0 mov ecx,dword ptr ss:[ebp-A8]
004082E3 mov edx,dword ptr ss:[ebp-18]
004082E6 push edx
004082E7 mov ebx,dword ptr ds:[ecx]
004082E9 call dword ptr ds:[401008]
004082F5 cmp dword ptr ds:[409000],0
004082F6 jne afkayas.2.408306
004082F7 fdiv dword ptr ds:[40100C]
00408304 jmp afkayas.2.408311
00408306 push dword ptr ds:[40100C]
0040830C call _JMP_&_adj_fdiv_m32>
00408311 sub esp,8
00408314 fnstsw ax
00408316 test al,D
00408318 jne afkayas.2.4087BF
0040831E faddp st(1),st(0)
00408320 fnstsw ax
00408322 test al,D
00408324 jne afkayas.2.4087BF
0040832A fstp qword ptr ss:[esp]
0040832D call dword ptr ds:[__vbaStrR8>]
00408333 mov edx, eax
00408335 lea ecx,dword ptr ss:[ebp-1C]
00408338 call dword ptr ds:[__vbaStrMove>]
0040833E mov dword ptr ss:[ebp-CC],ebx
00408344 mov ebx,dword ptr ss:[ebp-A8]
0040834A push eax
0040834B mov eax,dword ptr ss:[ebp-CC]
00408351 push ebx
00408352 call dword ptr ds:[eax+A4]
00408358 test eax, eax
0040835A jge afkayas.2.40830E
0040835C push A4
00408361 push afkayas.2.406FAC
00408366 push ebx
    
```

结果+2

dword_ptr ds:[afkayas.2.00401008]=10

.text:004082EF afkayas.2.exe:\$82EF #76EF

寄存器视图:

地址	值	ASCII	注释
00401008	41200000A	
0040100C	40A00000	...@	
00401010	40080000	...@	
00401014	40080000	...@	

0019F280 0019F37C
0019F284 0019F458
0019F288 00B7B80C 00B7B80C
0019F28C 7FFDB380
0019F290 00003704
0019F294 00000000
0019F298 74FC31E1
0019F29C 0019F38C

ST(0) 4002A0000000000000000000000000000 x87r6 非零 10.00000000000000000000000000000000
ST(1) 401282381000000000000000000000000 x87r7 非零 533377.00000000000000000000000000000000
ST(2) 00000000000000000000000000000000 x87r0 空 0.00000000000000000000000000000000
ST(3) 00000000000000000000000000000000 x87r1 空 0.00000000000000000000000000000000
ST(4) 00000000000000000000000000000000 x87r2 空 0.00000000000000000000000000000000
ST(5) 00000000000000000000000000000000 x87r3 空 0.00000000000000000000000000000000
ST(6) 3FFF80000000000000000000000000000 x87r4 空 1.00000000000000000000000000000000
ST(7) 3FFF80000000000000000000000000000 x87r5 空 1.00000000000000000000000000000000

默认 (stdcall)

1: [esp+4] 0019F458 0019F458
2: [esp+8] 00B7B80C 00B7B80C
3: [esp+C] 7FFDB380 7FFDB380
4: [esp+10] 00003704 00003704
5: [esp+14] 00000000 00000000

返回到 user32.DrawTextW+2C1 自 ???

浮点除法: 10/5=2

AfKayAs.2.Exe - PID: 2524 - 模块: afkayas.2.exe - 线程: 主线程 5764 - 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 引用 句柄 跟踪 线程

汇编视图:

```

004082D7 call dword ptr ds:[__vbaHRESULTCheckObj]
004082D0 mov ecx,dword ptr ss:[ebp-A8]
004082E3 mov edx,dword ptr ss:[ebp-18]
004082E6 push edx
004082E9 call dword ptr ds:[401008]
004082F5 fdiv dword ptr ds:[409000],0
004082F7 jmp afkayas.2.408306
004082F8 fdiv dword ptr ds:[40100C]
00408304 jmp afkayas.2.408311
00408306 push dword ptr ds:[40100C]
0040830C call _JMP_&_adj_fdiv_m32>
00408311 sub esp,8
00408314 fnstsw ax
00408316 test al,D
00408318 jne afkayas.2.4087BF
0040831E faddp st(1),st(0)
00408320 fnstsw ax
00408322 test al,D
00408324 jne afkayas.2.4087BF
0040832A fstp qword ptr ss:[esp]
0040832D call dword ptr ds:[__vbaStrR8>]
00408333 mov edx, eax
00408335 lea ecx,dword ptr ss:[ebp-1C]
00408338 call dword ptr ds:[__vbaStrMove>]
0040833E mov dword ptr ss:[ebp-CC],ebx
00408344 mov ebx,dword ptr ss:[ebp-A8]
0040834A push eax
0040834B mov eax,dword ptr ss:[ebp-CC]
00408351 push ebx
00408352 call dword ptr ds:[eax+A4]
00408358 test eax, eax
0040835A jge afkayas.2.40830E
0040835C push A4
00408361 push afkayas.2.406FAC
00408366 push ebx
    
```

结果+2

dword_ptr ds:[afkayas.2.0040100C]=5

.text:004082FE afkayas.2.exe:\$82FE #76FE

寄存器视图:

地址	值	ASCII	注释
00401008	41200000A	
0040100C	40A00000	...@	
00401010	40080000	...@	
00401014	40080000	...@	

0019F280 0019F37C
0019F284 0019F458
0019F288 00B7B80C 00B7B80C
0019F28C 7FFDB380
0019F290 00003704
0019F294 00000000
0019F298 74FC31E1
0019F29C 0019F38C

ST(0) 4002A0000000000000000000000000000 x87r6 非零 2.00000000000000000000000000000000
ST(1) 401282381000000000000000000000000 x87r7 非零 533377.00000000000000000000000000000000
ST(2) 00000000000000000000000000000000 x87r0 空 0.00000000000000000000000000000000
ST(3) 00000000000000000000000000000000 x87r1 空 0.00000000000000000000000000000000
ST(4) 00000000000000000000000000000000 x87r2 空 0.00000000000000000000000000000000
ST(5) 00000000000000000000000000000000 x87r3 空 0.00000000000000000000000000000000
ST(6) 3FFF80000000000000000000000000000 x87r4 空 1.00000000000000000000000000000000
ST(7) 3FFF80000000000000000000000000000 x87r5 空 1.00000000000000000000000000000000

默认 (stdcall)

1: [esp+4] 0019F458 0019F458
2: [esp+8] 00B7B80C 00B7B80C
3: [esp+C] 7FFDB380 7FFDB380
4: [esp+10] 00003704 00003704
5: [esp+14] 00000000 00000000

浮点寄存器相加, 结果存于 st0=533379。

AfKayAs.2.Exe - PID: 2524 - 模块: afkayas.2.exe - 线程: 主线程 5764 - 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 引用 句柄 跟踪 隐含FPU

```

004082D7 mov ecx,dword ptr ds:[ebp-A8]
004082D9 mov edx,dword ptr ss:[ebp-18]
004082E1 push edx
004082E2 mov ebx,dword ptr ds:[ecx]
004082E3 call dword ptr ds:[<_vbaR8Str>]
004082E5 fld dword ptr ds:[401008]
004082F0 cmp dword ptr ds:[409000],0
004082F5 jne afkayas.2.408306
004082F8 fdiv dword ptr ds:[40100C]
004082FC jmp afkayas.2.408311
00408304 push dword ptr ds:[40100C]
0040830C call <JMP.&_adj_fdiv_m32>
00408311 sub esp,8
00408314 fnstsw ax
00408316 test al,0
00408318 jne afkayas.2.4087BF
0040831E faddp st(1),st(0) 浮点寄存器st1+st0, 结果存在st0
00408320 fnstsw ax
00408322 test al,0
00408324 jne afkayas.2.4087BF
0040832A fstp qword ptr ss:[esp]
0040832D call dword ptr ds:[<_vbaStrR8>]
00408333 mov edx, eax
00408338 lea ecx,dword ptr ss:[ebp-1C]
0040833E call dword ptr ds:[<_vbaStrMove>]
00408344 mov dword ptr ss:[ebp-C],ebx
0040834A push eax
0040834B mov eax,dword ptr ss:[ebp-CC]
00408352 push ebx
00408355 call dword ptr ds:[eax+A4]
00408358 test eax, eax
0040835A jge afkayas.2.40836E
0040835C push A4
00408361 push afkayas.2.406FAC
00408366 push ebx

```

stl=0.0000000000000000
st0=533379.000000000000
.text:0040831E afkayas.2.exe:\$831E #771E

AfKayAs.2.Exe - PID: 2524 - 模块: oleaut32.dll - 线程: 主线程 5764 - 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 引用 句柄 跟踪 线程

```

773A1CF9 int3
773A1CFA int3
773A1CFB int3
773A1CFC int3
773A1CFD mov edi,edi
773A1CFF push ebp
773A1D00 mov ebp,esp
773A1D01 push edi
773A1D02 push ecx
773A1D03 call <oleaut32.SysAllocStringLen>
773A1D09 mov ecx,dword ptr ss:[ebp+8]
773A1D0C mov dword ptr ds:[ecx],eax
773A1D0E neg eax
773A1D0F shr eax
773A1D10 and eax,FFF8FF2
773A1D12 add eax,8007000
773A1D17 pop ebp

```

EAX 00000000 EBX 00000001 ECX 00000003 EDX 00000000 EBP 0019E937 ESP 0019E937 ESI 0005C7CF8 EDI 001E2F2 EIP 773A1D09 oleaut32.773A1D09

EFLAGS 00000024 DF 1 PF 1 AF 0 OF 0 SF 0 DE 0 CF 0 TF 0 IF 1

AfKayAs.2.Exe - PID: 4032 - 模块: afkayas.2.exe - 线程: 主线程 3024 - 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 引用 句柄 跟踪

```

004082D7 push eax
004082D9 call dword ptr ds:[<_vbaHRESULTCheckObj>]
004082DD mov ecx,dword ptr ss:[ebp-A8]
004082E1 mov edx,dword ptr ss:[ebp-18]
004082E2 push edx
004082E3 mov ebx,dword ptr ds:[ecx]
004082E4 call dword ptr ds:[<_vbaR8Str>]
004082E5 fld dword ptr ds:[401008]
004082F0 cmp dword ptr ds:[409000],0
004082F5 jne afkayas.2.408306
004082F8 fdiv dword ptr ds:[40100C]
004082FC jmp afkayas.2.408311
00408304 push dword ptr ds:[40100C]
0040830C call <JMP.&_adj_fdiv_m32>
00408311 sub esp,8
00408314 fnstsw ax
00408316 test al,0
00408318 jne afkayas.2.4087BF
0040831E faddp st(1),st(0)
00408320 fnstsw ax
00408322 test al,0
00408324 jne afkayas.2.4087BF
0040832A fstp qword ptr ss:[esp]
0040832D call dword ptr ds:[<_vbaStrR8>]
00408333 mov edx, eax
00408338 lea ecx,dword ptr ss:[ebp-1C]
0040833E call dword ptr ds:[<_vbaStrMove>]
00408344 mov dword ptr ss:[ebp-CC],ebx

```

eax:L"533379" [ebp-18]:L"533377"

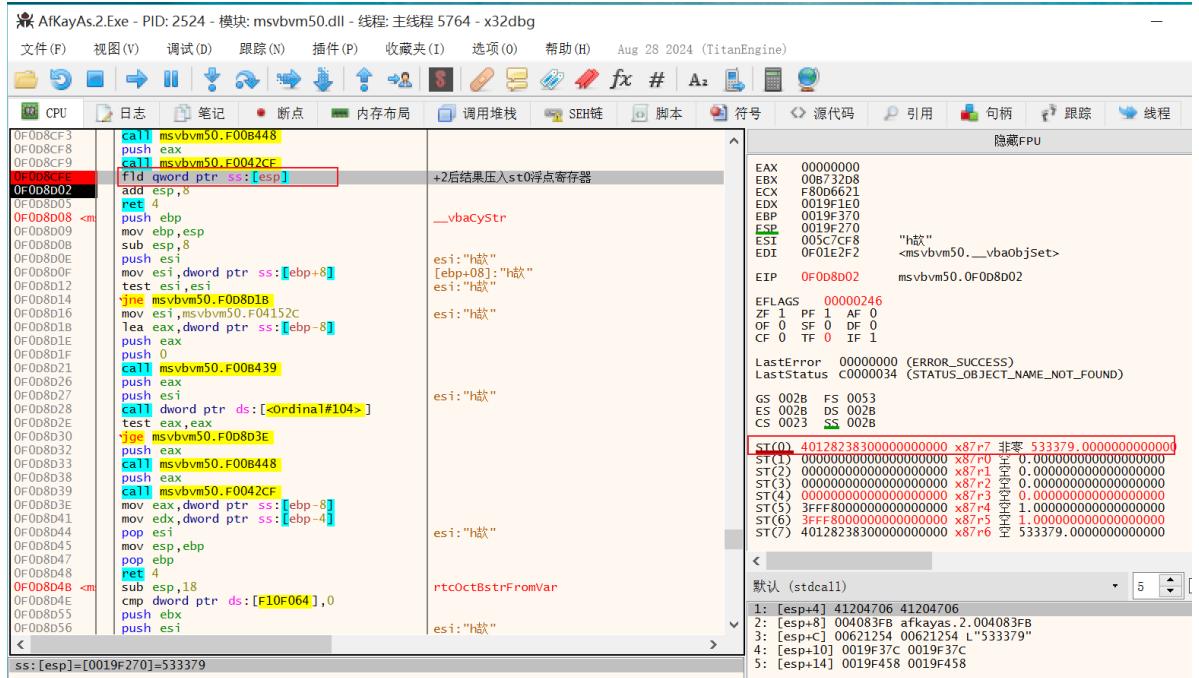
结果+2 eax:L"533379"

跳转不会执行

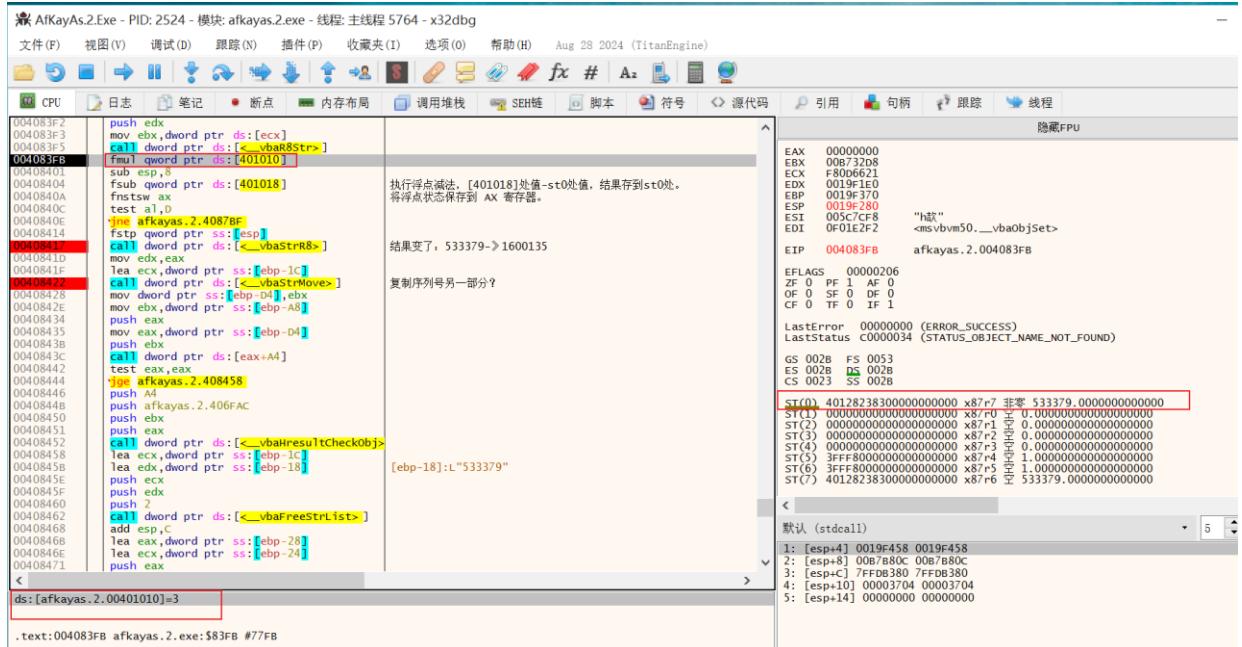
继续继续分析分析！！！！

结果变化： $533379 \times 3 = > 1600137$

+2 后结果压入 st0 浮点寄存器: 533379



533379*3:



Fmul 指令执行乘法 $533379.0 \times 3 = 1600137.0$ 结果存于 st0 浮点寄存器：

```

004083F2 push edx
004083F3 mov ebx,dword ptr ds:[ecx]
004083F5 call dword ptr ds:[__vba8Str>]
004083F9 sub esp,8
00408401 fsub qword ptr ds:[401018]      执行浮点减法。[401018]处值-st0处值，结果存到st0处。
00408404 fstsw ax
0040840A test al,0
0040840C jne afkayas.2.4087BF
00408414 fstp qword ptr ss:[esp]
00408417 call dword ptr ds:[__vbaStrR8>]
0040841D mov edx, eax
0040841E test ecx, dword ptr ss:[ebp-1C]
00408422 call dword ptr ds:[__vbastrMove>]
00408425 mov ebx,dword ptr ss:[ebp-4],ebx
00408434 push eax
00408435 mov eax,dword ptr ss:[ebp-D4]
0040843B push ebx
0040843C call dword ptr ds:[eax+A4]
00408442 test eax, eax
00408444 jne afkayas.2.408458
00408446 push A4
0040844B push afkayas.2.406FAC
00408450 push ebx
00408451 push eax
00408452 fsub dword ptr ds:[__vbaResultCheckObj>]
00408453 lea ecx, dword ptr ss:[ebp-1C]
00408458 lea edx, dword ptr ss:[ebp-18]      [ebp-18]:L"533379"
0040845B push ecx
0040845C push edx
00408460 push 2
00408462 call dword ptr ds:[__vbaFreeStrList>]
00408468 add esp,C
0040846B lea eax, dword ptr ss:[ebp-28]
0040846E lea ecx, dword ptr ss:[ebp-24]
00408471 push eax
    
```

默认 (stdcall)

- 1: [esp+4] 0019E458 0019E458
- 2: [esp-8] 00B7B80C 00B7B80C
- 3: [esp+C] 7FFDB380 7FFDB380
- 4: [esp+10] 00003704 00003704
- 5: [esp+14] 00000000 00000000

结果变化： $1600137 - 2 = 1600135$

```

004083F2 push edx
004083F3 mov ebx,dword ptr ds:[ecx]
004083F5 call dword ptr ds:[__vba8Str>]
004083F9 sub esp,8
00408401 fsub dword ptr ds:[401018]      执行浮点减法。[401018]处值-st0处值，结果存到st0处。
00408404 fstsw ax
0040840A test al,0
0040840C jne afkayas.2.4087BF
00408414 fstp qword ptr ss:[esp]
00408417 call dword ptr ds:[__vbaStrR8>]
0040841D mov edx, eax
0040841E test ecx, dword ptr ss:[ebp-1C]
00408422 call dword ptr ds:[__vbastrMove>]
00408425 mov ebx,dword ptr ss:[ebp-D4],ebx
00408434 push eax
00408435 mov eax,dword ptr ss:[ebp-D4]
0040843B push ebx
0040843C call dword ptr ds:[eax+A4]
00408442 test eax, eax
00408444 jne afkayas.2.408458
00408446 push A4
0040844B push afkayas.2.406FAC
00408450 push ebx
00408451 push eax
00408452 fsub dword ptr ds:[__vbaResultCheckObj>]
00408453 lea ecx, dword ptr ss:[ebp-1C]
00408458 lea edx, dword ptr ss:[ebp-18]      [ebp-18]:L"533379"
0040845B push ecx
0040845C push edx
00408460 push 2
00408462 call dword ptr ds:[__vbaFreeStrList>]
00408468 add esp,C
0040846B lea eax, dword ptr ss:[ebp-28]
0040846E lea ecx, dword ptr ss:[ebp-24]
00408471 push eax
    
```

默认 (stdcall)

- 1: [esp+4] 0019E458 0019E458
- 2: [esp-8] 00B7B80C 00B7B80C
- 3: [esp+C] 7FFDB380 7FFDB380
- 4: [esp+10] 00003704 00003704
- 5: [esp+14] 00000000 00000000

.text:00408404 afkayas.2.exe:\$8404 #804

Fsub 指令执行浮点减法 $1600137.0 - 2 = 1600135$ 结果存在 st0 浮点寄存器：

AfKayaS.2.Exe - PID: 2524 - 模块: afkayas.2.exe - 线程: 主线程 5764 - 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 引用 句柄 跟踪 线程 隐藏FPU

```

004083F2 push edx
004083F3 mov ebx,dword ptr ds:[ecx]
004083F5 call dword ptr ds:[__vbaR8Str>]
004083FB fmul qword ptr ds:[401010]
00408401 sub esp,8
00408403 sub qword ptr ds:[401018]
00408404 fntsc al,0
00408405 test al,0
00408406 jne afkayas.2.4087BF
0040840E fstp qword ptr ss:[esp]
00408414 mov edx,ecx
00408416 mov eax,dword ptr ss:[__vbaStrR8>]
0040841D lea ecx,dword ptr ss:[ebp-1C]
00408428 mov eax,dword ptr ss:[ebp-D4]
00408431 push eax
00408434 mov eax,dword ptr ss:[ebp-D4]
0040843B push ebx
0040843C lea eax,dword ptr ss:[eax+A4]
00408442 test eax,ecx
00408444 jne afkayas.2.408458
00408446 push A4
0040844B push afkayas.2.406FAC
00408450 push ebx
00408451 push esp
00408452 fmul qword ptr ds:[__vbaResultCheckObj>]
00408458 lea ecx,dword ptr ss:[ebp-1C]
0040845B lea edx,dword ptr ss:[ebp-18]
[ebp-18]:L"533379"
00408460 push edx
00408462 push 2
00408463 call dword ptr ds:[__vbaFreeStrList>]
00408466 add esp,C
00408468 lea eax,dword ptr ss:[ebp-28]
0040846E push eax
00408471

```

533379.0x3=1600137.0
执行浮点减法。[401018]处值-st0处值，结果存到st0处。
将浮点状态保存到 AX 寄存器。

结果变了：533379->1600135
复制序列号另一部分？

EAX 00000000 EBX 006732D8 ECX F8000621 EDX 00000000 EBP 0019E370 ESP 0019E278 "h款" ESI 005C7CF8 <msvbvm50.__vbaobjSet> EDI 0F01E2F2 EIP 0040840A afkayas.2.0040840A

EFLAGS 000000216 ZF 0 PF 1 AF 1 OF 0 SF 0 DF 0 CF 0 TF 0 IF 1

LastError 00000000 (ERROR_SUCCESS)
LastStatus C0000034 (STATUS_OBJECT_NAME_NOT_FOUND)

GS 002B FS 0053 ES 002B DS 002B CS 0023 SS 002B

ST(0) 0000000000000000 x87r7 空 0.0000000000000000
ST(1) 0000000000000000 x87r1 空 0.0000000000000000
ST(2) 0000000000000000 x87r3 空 0.0000000000000000
ST(3) 0000000000000000 x87r2 空 0.0000000000000000
ST(4) 3FFF8000000000000 x87r4 空 1.0000000000000000
ST(5) 3FFF8000000000000 x87r5 空 1.0000000000000000
ST(6) 4023BA43B400000000 x87r6 空 10000000000000000
ST(7) 4038E1ED8C61160000 x87r7 空 16001350000000000

默认 (stdcall)

1: [esp+4] 00621254 00621254 L"533379"
2: [esp+8] 0019E37C 0019E37C
3: [esp+12] 0019E458 0019E458
4: [esp+16] 00003704 00003704

复制结果到另一内存地址：

AfKayaS.2.Exe - PID: 2524 - 模块: afkayas.2.exe - 线程: 主线程 5764 - 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 引用 句柄 跟踪 线程 隐藏FPU

```

004083EF mov edx,dword ptr ss:[ebp-18]
004083F2 push edx
004083F3 mov ebx,dword ptr ds:[ecx]
004083F5 call dword ptr ds:[__vbaR8Str>]
004083FB fmul qword ptr ds:[401010]
00408401 sub esp,8
00408403 sub qword ptr ds:[401018]
00408404 fntsc al,0
00408405 test al,0
00408406 jne afkayas.2.4087BF
0040840E fstp qword ptr ss:[esp]
00408414 mov edx,ecx
00408416 mov eax,dword ptr ss:[__vbaStrR8>]
0040841D lea ecx,dword ptr ss:[ebp-1C]
00408428 mov eax,dword ptr ss:[ebp-D4]
00408431 push eax
00408434 mov eax,dword ptr ss:[ebp-D4]
0040843B push ebx
0040843C lea eax,dword ptr ss:[eax+A4]
00408442 test eax,ecx
00408444 jne afkayas.2.408458
00408446 push A4
0040844B push afkayas.2.406FAC
00408450 push ebx
00408451 push esp
00408452 fmul qword ptr ds:[__vbaResultCheckObj>]
00408458 lea ecx,dword ptr ss:[ebp-1C]
0040845B lea edx,dword ptr ss:[ebp-18]
[ebp-18]:L"533379"
00408460 push edx
00408462 call dword ptr ds:[__vbaFreeStrList>]
00408466 add esp,C
00408468 lea eax,dword ptr ss:[ebp-28]
0040846E push eax
00408471

```

533379.0x3=1600137.0
执行浮点减法。[401018]处值-st0处值，结果存到st0处。
将浮点状态保存到 AX 寄存器。

结果变了：533379->1600135
复制序列号另一内存地址

EAX 006215B4 L"1600135" EBX 006732D8 ECX 0019F354 EDX 006215B4 L"1600135" EBP 0019E370 ESP 0019E280 ESI 005C7CF8 "h款" EDI 0F01E2F2 <msvbvm50.__vbaobjSet> EIP 00408422 afkayas.2.00408422

EFLAGS 000000246 ZF 1 PF 1 AF 0 OF 0 SF 0 DF 0 CF 0 TF 0 IF 1

LastError 00000000 (ERROR_SUCCESS)
LastStatus C0000034 (STATUS_OBJECT_NAME_NOT_FOUND)

GS 002B FS 0053 ES 002B DS 002B CS 0023 SS 002B

ST(0) 0000000000000000 x87r0 空 0.0000000000000000
ST(1) 0000000000000000 x87r1 空 0.0000000000000000
ST(2) 0000000000000000 x87r3 空 0.0000000000000000
ST(3) 0000000000000000 x87r2 空 0.0000000000000000
ST(4) 3FFF8000000000000 x87r4 空 1.0000000000000000
ST(5) 3FFF8000000000000 x87r5 空 1.0000000000000000
ST(6) 4023BA43B400000000 x87r6 空 10000000000000000
ST(7) 4038E1ED8C61160000 x87r7 空 16001350000000000

默认 (stdcall)

1: [esp+4] 0019F37C 0019F37C
2: [esp+8] 0019F458 0019F458
3: [esp+12] 00B7B80C 00B7B80C
4: [esp+16] 7FFDB380 7FFDB380
5: [esp+20] 00003704 00003704

内存 1 内存 2 内存 3 内存 4 内存 5 监视 1 局部变量

地址	值	ASCII	注释
0019F354	00000000	...	
0019F358	00621254	T.B.	L"533379"
0019F35C	0019F8EC	10..	指向SEH_Record[1]的指针
0019F360	00401030	V.B.	afkayas.2.JMP.&__vbaExceptionHandler
0019F364	0039C30	...	
0019F368	00401030	0.B.	afkayas.2.00401030

0019F354 00000000 L"533379"
指向SEH_Record[1]的指针
afkayas.2.JMP.&__vbaExceptionHandler
afkayas.2.00401030

0019F360 00401036 0019F8EC
0019F364 00401030 0019F280
0019F368 00401030 0019F36C
0019F370 0019F37C 00000001
0019F374 0F01E5A9 0019F374
0019F378 005C7CF8 返回到 msvbvm50.__vbaErase+2A0 自 ???

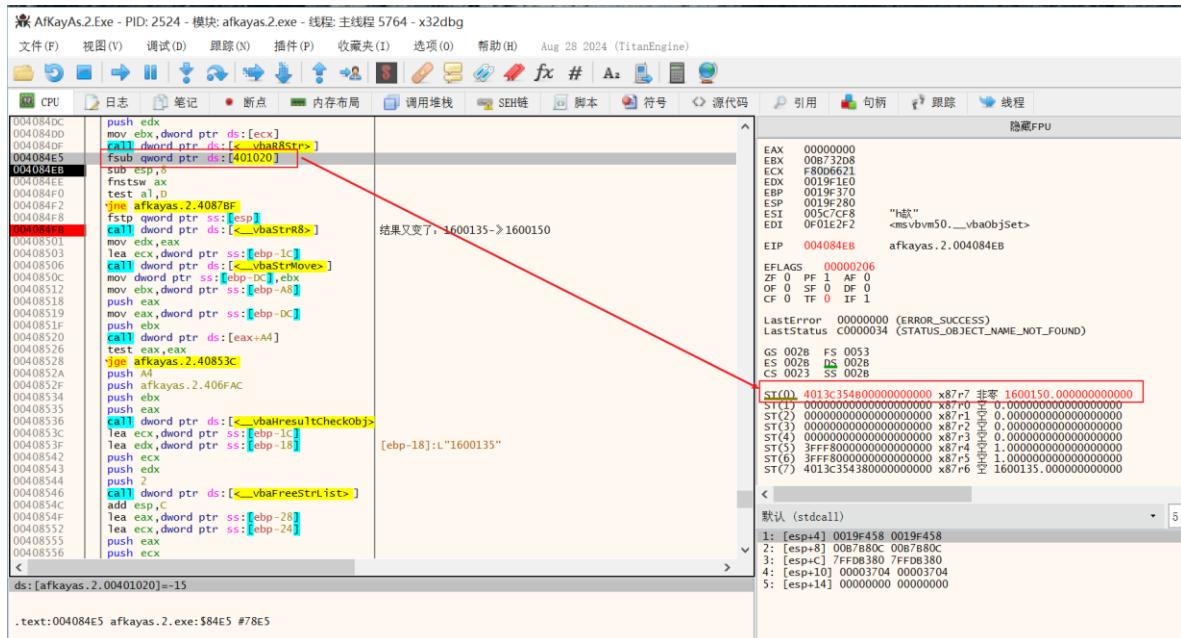
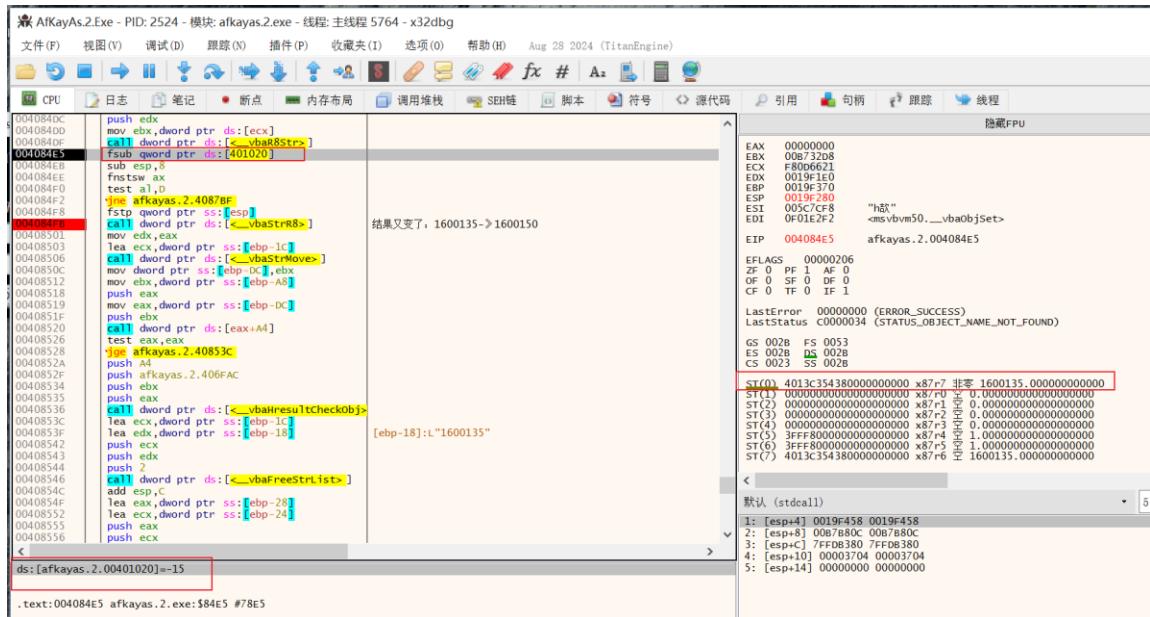
结果变化: 1600135- (-15) =>1600150

```
004084CD call dword ptr ds:[_vbaResultCheckObj]
004084D3 mov ecx,dword ptr ss:[ebp-A8]
004084D9 mov edx,dword ptr ss:[ebp-18] [ebp-18]:L"1600135"
004084DC push edx
004084DD mov ebx,dword ptr ds:[ecx]
004084DF call dword ptr ds:[_vba8Str]
004084E5 fsub qword ptr ds:[401020]
004084EB sub esp,8
004084EE fnstsw ax
004084F0 test al,D
004084F2 jne afkayas.2.4087BF
004084F8 fstp qword ptr ss:[esp]
004084FB call dword ptr ds:[_vbaStr8>] 结果又变了：1600135->1600150
00408501 mov edx,eax
00408503 lea ecx,dword ptr ss:[ebp-1C]
00408506 call dword ptr ds:[_vbAstrMove>]
```

fld 指令将 ss:[ESP]c 处地址的值 1600135 压入 st0 浮点寄存器：

Fsub 指令用 st0 浮点寄存器的值 1600135 减去 (-15) 等于 1600150，结果存

于 st0 寄存器处。



AIKeyAs.2.Exe - PID: 2524 - 模块: afkayas.2.exe - 线程: 主线程 5764 - 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 引用 句柄 跟踪 线程 隐藏FPU

```

004084EB sub esp,8
004084F0    rts
004084F1    pop al,[bp]
004084F2    jne afkayas.2.4087BF
004084F8    fstp qword ptr ss:[esp]
004084F9    call dword ptr ds:[__vbaStrR8s]
00408501    mov edx, eax
00408503    lea ecx, dword ptr ss:[ebp+1C]
00408506    call dword ptr ds:[__vbaStrMove]
结果又变了: 1600135->1600150
edx:L"1600150", eax:L"1600150"
[ebp+1C]:L"1600150" 结果复制到另一内存地址
0040850C    mov dword ptr ss:[ebp+10], edx
00408510    mov ebx, edx
00408511    push eax
00408512    mov exx, dword ptr ss:[ebp+DC]
00408513    push ebx
00408514    call dword ptr ds:[eax+A]
00408516    test eax, eax
00408517    afkayas.2.40853C
00408518    push eax
00408519    push ebx
00408520    call dword ptr ds:[eax+A]
00408521    test eax, eax
00408522    afkayas.2.40853C
00408523    push eax
00408524    push ebx
00408525    push eax
00408526    call dword ptr ds:[__vbaResultCheckObj]
00408527    lea ecx, dword ptr ss:[ebp+C]
00408528    lea edx, dword ptr ss:[ebp+18]
00408529    push ebx
00408530    push edx
00408531    push 2
00408532    call dword ptr ds:[__vbaFreeStrList]
00408533    add esp,C
00408534    lea eax, dword ptr ss:[ebp-28]
00408535    lea edx, dword ptr ss:[ebp-24]
00408536    push eax
00408537    push ebx
00408538    push 2
00408539    call dword ptr ds:[__vbaFreeObjList]
00408540    add esp,C
00408541    push esi
00408542    push ebx
00408543    lea eax, dword ptr ss:[ebp-18]
00408544    lea edx, dword ptr ss:[ebp-24]
00408545    push eax
00408546    push ebx
00408547    push 2
00408548    call dword ptr ds:[__vbaFreeStrList]
00408549    add esp,C
00408550    lea eax, dword ptr ss:[ebp-28]
00408551    lea edx, dword ptr ss:[ebp-24]
00408552    push eax
00408553    push ebx
00408554    push 2
00408555    call dword ptr ds:[__vbaFreeObjList]
00408556    add esp,C
00408557    push 2
00408558    call dword ptr ds:[__vbaFreeObjList]
00408559    add esp,C
00408560    push esi
00408562

dword ptr ds:[0040B194] <afkayas.2.__vbaStrMove>=0msvbvm50.__vbaStrMove>

.text:00408506 afkayas.2.exe:$8506 #7906

```

内存 1 内存 2 内存 3 内存 4 内存 5 监视 1 局部变量

地址	值	ASCII	注释
0019E354 00621254	0xb..	L"1600150"	
0019E358 0062618C	.ab..	L"1600135"	
0019E8EC 0019E8C	io..	指向SEH_Record[1]的指针	afkayas.2.Record[1].&__vbaExceptionHandler
0019E360 00401056	v@..	afkayas.2.JMP.&__vbaExceptionHandler	afkayas.2.00401030
0019E364 0019F280	o..		0019E370 0019E37C

用户输入序列号和计算的序列号：尝试构造用户名和序列号输入界面。

AIKeyAs.2.Exe - PID: 2956 - 模块: afkayas.2.exe - 线程: 主线程 2552 - 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 引用 句柄 跟踪 线程 隐藏FPU

```

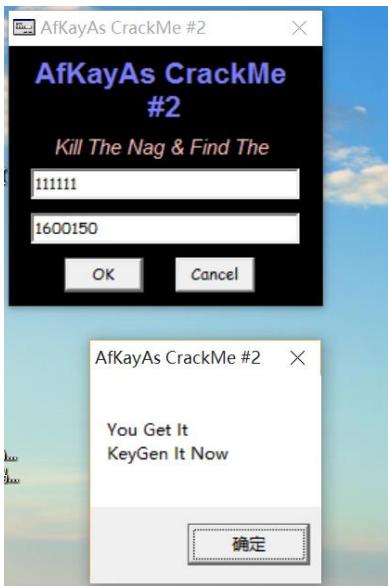
0040858A    jne afkayas.2.4085CE
0040858C    push afkayas.2.4086FA
0040858D    push esi
0040858E    push eax
0040858F    call dword ptr ds:[__vbaResultCheckObj]
00408590    mov eax, dword ptr ss:[ebp-18]
00408591    leh-18:L"444444"
00408592    call dword ptr ds:[__vbaR8Strs]
00408593    mov ecx, dword ptr ss:[ebp-1C]
00408594    fstp qword ptr ss:[ebp-E4]
004085E1    push ecx
004085E2    call dword ptr ds:[__vbaR8Strs]
004085E3    cmp byte ptr ds:[1090000], 0
004085E4    jne afkayas.2.408630
004085E5    fidvr qword ptr ss:[ebp-E4]
004085E6    jmp afkayas.2.40860A
004085F9    push dword ptr ss:[ebp-E0]
004085F0    push dword ptr ss:[ebp-E4]
004085F1    cmp [0x401056], _adjs_fidvr_m64>
0040860A    test al, 0
0040860C    test al, 0
00408614    lea eax, afkayas.2.4087BF
0040861A    fcomp qword ptr ds:[401028]
00408620    fstsw ah,40
00408622    test ah,40
00408625    jne afkayas.2.408626
00408627    mov esi, 1
00408628    jmp afkayas.2.408630
0040862E    xor esi, esi
00408630    lea edx, dword ptr ss:[ebp-1C]
00408631    lea eax, dword ptr ss:[ebp-18]
00408636    push edx
00408637    push eax
00408638    push 2
0040863A    call dword ptr ds:[__vbaFreeStrList]
正确序列号
edx=0
dword ptr ss:[ebp-1C]=[0019E354 &L"1600150"]=005C2794 L"1600150"
.text:00408630 afkayas.2.exe:$8630 #7A30

```

内存 1 内存 2 内存 3 内存 4 内存 5 监视 1 局部变量

地址	值	ASCII	注释
774C1000 0001E070	pd..		0019E354 005C2794 L"1600150"
0019E358 005C2114	005C2114	L"444444"	
0019E35C 0019F8EC	0019F8EC	指向SEH_Record[1]的指针	afkayas.2.JMP.&__vbaExceptionHandler
0019E360 00401056	00401056	00401056	0019E370 0019E37C

之间在界面手输入：111111/1600150--成功。



逆向出算法：以下是 python 脚本。

```
def calculate_result(input_string):
    return (len(input_string) * 0x15B38 + ord(input_string[0]))

# 示例使用
input_string = input("请输入 name:\n")
result = calculate_result(input_string)

# 定义浮点数 A
A = 10.0

# 将结果与 A 除以 5 的结果相加
final_result = result + (A / 5.0)

# 将 final_result 乘以 3.0
final_result *= 3.0

final_result -= 2.0

# 从 final_result 中减去 -15 (相当于加 15)
final_result -= -15.0 # 这行代码可以写为 final_result += 15.0

# 使用字符串格式化来输出序列号
#print("序列号为：" + str(final_result)) 输出结果有一位小数
print("序列号为：" + str(int(final_result)))
```

运行验证：算法逆出成功

