HOW TO CRACK *STARDUST SCREEN SAVER TOOLKIT 1.0*

Tutorial by UmE

Introduction: this time I'll show you how to crack a time trial program. In fact you can use this application for 7 days, after that you must purchase the full version.

Necessary tools: SoftIce 3.24 or better, W32Dasm version 8.9 and an hex editor (I've used Winhex 8.0).

Program description: Stardust screen saver toolkit, SSWizard.exe, 905.216 bytes.

Good work guys!!

Step 1: when you run the program for the first time, you can notice that a nag screen appear, telling us that we have others 7 days to evaluate the application (with the Trial Usage Meter). Pressing "OK" will appear the **Screen Saver Toolkit Wizard** that guide you through the construction of your screensaver. If we change the date of our system increasing of 1 week, the nag screen will tell us that the time trial is ended and that you have to buy the full version.

Step 2: let's enter in SoftIce pressing Ctrl+D and set a breakpoint in the **GetSystemTime** function (type **bpx getsystemtime**). By this way we can break the program when it'll go to check the current date and make the comparison with the 7 days of the time trial. Press Ctrl+D another time to return to the operating system, and run the program…..BOOM! You're in SoftIce again! Press F11 to return to the piece of code that call the function Get SystemTime. You should be here:

```
014F:0040F1E7  CALL      [KERNEL32!GetSystemTime]
014F:0040F1F2  CMP       [00440032],CX
014F:0040F1F9  JNZ       0040F23B
014F:0040F23B  LEA       EAX,[ESP+24]
014F:0040F23F  PUSH      EAX
014F:0040F240  CALL      [KERNEL32!GetTimeZoneInformation]
014F:0040F246  MOV       EDX,00000001
……
……
```

Start to trace the code pressing F10 until another call to the GetSystemTime will appear. Press F11 to return to the caller and start traceing another time the code until you see the following lines:

```
014F:004079ED  CALL      [KERNEL32!CreateFileA]
014F:004079F3  CMP       EAX,-01
014F:004079F6  MOV       EBX,EAX
014F:004079F8  JZ        00407A27
014F:004079FA  LEA       EAX,[ESP+18]
014F:004079FE  PUSH      EDI
014F:004079FF  LEA       ECX,[ESP+18]
014F:00407A03  PUSH      EAX
014F:00407A04  PUSH      04
014F:00407A06  MOV       EBP,[KERNEL32!ReadFile]
014F:00407A0C  PUSH      ECX
```

```
014F:00407A0D  PUSH      EBX
014F:00407A0E  CALL      EBP
014F:00407A10  LEA       ECX,[ESP+18]
014F:00407A14  PUSH      EDI
014F:00407A15  LEA       EAX,[ESP+14]
014F:00407A19  PUSH      ECX
014F:00407A1A  PUSH      04
014F:00407A1C  PUSH      EAX
014F:00407A1D  PUSH      EBX
014F:00407A1E  CALL      EBP
014F:00407A20  PUSH      EBX
014F:00407A21  CALL      [KERNEL32!CloseHandle]
014F:00407A27  CMP       ESI,[ESP+14]
014F:00407A2B  JBE       00407A4F                  NO JUMP
014F:00407A2D  CMP       ESI,[ESP+10]
014F:00407A31  JAE       00407A4F                  JUMP
014F:00407A4F  MOV       EAX,EDI
014F:00407A51  POP       EBP
```

With the functions **CreateFile**, **ReadFile** and **CloseHandle** the program reads from a specified file (SSWizard.spd, you can find it typeing in SoftIce “d ESP+18”) in which are encripted some date informations of the program (probably the installation date and the expiration date). The **JBE 00407A4F** instruction jump if the current date is before the installation date (encrypted in ESP+14), and the **JAE 00407A4F** instruction jump if it’s above the expiration date (encrypted in ESP+10). If we NOP the two conditional jumps the program will follow the normal flow that brings to compare the number of days followed the installation with the 7 days of the trial period: in base at this comparison the program tells you how many days remains until the end of the trial period. If you NOP the two conditional jumps you have:

```
014F:00407A27  CMP       ESI,[ESP+14]
014F:00407A2B  NOP                         ----→ We have nopped the 2 bytes
014F:00407A2C  NOP                         -----> of the JBE
014F:00407A2D  CMP       ESI,[ESP+10]
014F:00407A31  NOP                         ----→ We have nopped the 2 bytes
014F:00407A32  NOP                         ----→ of the JAE
014F:00407A33  MOV       EAX,[ESP+10]
014F:00407A37  MOV       ECX,00015180
014F:00407A3C  SUB       EAX,ESI
014F:00407A3E  SUB       EDX,EDX
014F:00407A40  DIV       ECX
014F:00407A42  LEA       EDI,[EAX+01]
014F:00407A45  CMP       EDI,07
014F:00407A48  JBE       00407A4F                                     JUMP
```

In the instruction located at 00407A45 the code compares the number of days you’ve used the program with the 7 days of the trial period and jumps if you’ve used it less. If you change the JBE instruction in JMP instruction (change the first byte in EB) the program will work forever!!!

Let’s see now how to take away the initial nag screen.

We know that after the initial nag screen, the program show us a “**Screen saver toolkit wizard**” so in the code there will be a place where the program pushes this string as caption. Well, open W32Dasm, dissasemble the file SSWizard.exe and search for the text “**Screen saver toolkit wizard**” (you can do this going on the Search menu and selecting

Find Text…). You will find many of this string in the Dialog Information part of the code, but you need to find it in the ASM code. So continue to press “Next” until you find this:

```
7     :00407C22 E885A40100                    call 004220AC
6     :00407C27 83F801                        cmp eax, 00000001
5     :00407C2A 750A                          jne 00407C36
4     :00407C2C C78594FEFFFF01000000          mov dword ptr [ebp+FFFFFE94], 00000001

      * Referenced by a (U)nconditional or (C)onditional Jump at Address:
      |:00407C2A(C)
      |
      :00407C36 C745FCFFFFFFFF                mov [ebp-04], FFFFFFFF
      :00407C3D E8DE030000                    call 00408020

      * Referenced by a (U)nconditional or (C)onditional Jump at Addresses:
      |:00407BB2(U), :00407BF5(U)
      |
3     :00407C42 83BD94FEFFFF00                cmp dword ptr [ebp+FFFFFE94], 00000000
2     :00407C49 7512                          jne 00407C5D

      * Referenced by a (U)nconditional or (C)onditional Jump at Address:
      |:00407FFB(U)
      |
      :00407C4B 33C0                          xor eax, eax
      :00407C4D 8B4DF4                        mov ecx, dword ptr [ebp-0C]
      :00407C50 64890D00000000                mov dword ptr fs:[00000000], ecx
      :00407C57 5F                            pop edi
      :00407C58 5E                            pop esi
      :00407C59 8BE5                          mov esp, ebp
      :00407C5B 5D                            pop ebp
      :00407C5C C3                            ret

      * Referenced by a (U)nconditional or (C)onditional Jump at Address:
      |:00407C49(C)
      |
1     :00407C5D 6A00                          push 00000000
      :00407C5F 8D8D50EFFFFF                  lea ecx, dword ptr [ebp+FFFFEF50]
      :00407C65 6A00                          push 00000000

      * Possible StringData Ref from Data Obj ->"Screen Saver Toolkit Wizard"
                                  |
      :00407C67 6858B54300                    push 0043B558
      :00407C6C E86FACFFFF                    call 004028E0
```

I’ve numbered the interesting instruction in the order that you have to consider them.

1- The program jumps here from a conditional jump located at 00407C49
2- The instruction jumps if [ebp+FFFFFE94] is not equal to 0 (see number 3).
3- This instruction compare [ebp+FFFFFE94] with 0.
4- Looking above in the code we can see this instruction that moves 1 in [ebp+FFFFFE94].
5- The value of [ebp+FFFFFE94] is decided from this jump: is eax is not equal to 1 (see number 6) then [ebp+FFFFFE94]=0 (it skips the number 4) else [ebp+FFFFFE94]=1 (the instruction 4 is executed).
6- Compares the value of eax with 1…what is eax?

Eax is the return value of the call at the instruction number 7 that load the nag screen!!! It returns 1 if the nag loads succesfully else it returns 0 (the program will flow to the ExitProcess function). So dear guys to avoid the nag screen NOP the call at the address 00407C22 (remember that the NOP uses 1 byte and this call uses 5 byte so you got to change the first 5 byte in 90) and NOP the jne at 00407C2A. By this way you will never see any nag screen again!!!!

Ok guys that's all! I hope that this tutorial should be useful for someone!!

See you the next time!

UmE (ume15@hotmail.com)

Greetings to Volatility and all the Immortal Descendants (http://www.ImmortalDescendants.com)