HOW TO REGISTER *ANIMAGIC GIF 1.03*

Tutorial by UmE

Introduction: finally guys here is my first tutorial about the *Code Reverse Engeneering* in which I will try to explain you how to register Animagic Gif 1.03.

Necessary tools: W32Dasm version 8.9, an hex editor (I've used Winhex 8.0).

Program description: Animagic version 1.03, animagic32.exe, 536.576 bytes.

Let's start!!!

Step 1: as before thing after having installed the program we start the application from the start menu or clicking twice on the lines animagic32.exe. As you will observe an annoying nag screen will appear telling us that we have used the program for N days and that after 30 days some functions will be disabled. As continuation is possible to record ourselves on-line, to compile a form for the recording via e-mail or to continue shareware. Let's click on "**Continue**" and the program will shows us his main window. We now go on the "**Help**" menu and we select the voice "**Enter name and password**"; we write our name and a number of recording that naturally it will result wrong, and in fact the program will communicate that the inserted number is not valid ("**Registration failed: Ivalid password**").

Step 2: let's open W32Dasm and let's dissasemble the program that we want to unprotect (animagic32.exe). We go on the **Refs** menu, we select the voice **String Data Reference** and we search for the string that Animagic Gif has returned us when we have inserted the wrong code. At this point click 2 times on the found string and W32Dasm will brings us in the section of code to which it makes reference. We will find in fact:

: 0042045A 83C414 add esp, 0000014
: 0042045D EB3D jmp 0042049C

- **Referenced by a (U)nconditional or (C)onditional Jump at Address:**

| : 004203ED (C)
|

- **Possible StringData Ref from Data obj -> "Registration failed: Invalid Password"**

: 00420465 680B954600 push 0046950B

This means that the program jumps to the address 00420465 according to the result of the conditioned jump (C) that is located at the address 004203ED.
Always on W32Dasm we go on the menù "**Goto**", we select the voice "**Goto code location**" and we insert the address 004203ED to see so what determines the jump to the address 00420465.

W32Dasm bring us once more to the requested address and this time the visualized code will be the following:

```
: 004203E1   A2D1334700          mov byte ptr [004733D1], al
: 004203E6   803DD133470000      cmp byte ptr [004733D1], 00
: 004203ED   7470                je  0042045F
```

The first instruction moves the content of al in the cell of memory 004733D1, the second instruction compares this value with 0 and the third jumps to the instruction 0042045F (wrong registration number!!!) if the two values are equal. This lets think that the program set the value of al second that is registered or less: 0 = not registered, 1 = registered. Is necessary therefore to do in way that al should be always equal to 1, but where is the al value decided? If we scroll a little with the up arrow key on the code departing from the instruction at the address 004203ED we will notice that a little before there is a call to a function that is really what determines the value of al.

```
: 004203DB   E81DD3FFFF          call 0041D6FD
```

Let's enter in the function, and scrolling down we will see:

- **Referenced by a (U)nconditional or (C)onditional Jump at Address:**

```
|  : 0041D75E (C)
|
: 0041D767   B001                mov al, 01
: 0041D769   EB09                jmp 0041D774
```

The first instruction is the one that set the value of al = 1 (registered), while the second is jumping to the final part of the function that brings to the call. Also in this case the instruction at 0041D767 is referenced to a conditioned jump located at the address 0041D75E. We go to discover what there is at that address.

```
: 0041D757  3B45FC               cmp eax, dword ptr [ebp-04]
: 0041D75A  750F                 jne 0041D76B
: 0041D75C  85DB                 test ebx, ebx
: 0041D75E  7507                 jne 0041D767
```

In way thatt the program follows the correct flow and performs the instruction at the address 0041D767 is necessary to change the instruction jne 0041D76B in je 0041D76B (all it takes is changing with an hex editor the byte 75 in 74) and the jne 0041D767 in jmp 0041D767 (all it takes is changing the byte 75 in EB). Once effected the change let's restart Animagic Gif....et voilà!! The nag screen brings the writing "**Registered to**: ...". and it is possible to disable it for the next times when the program will be executed.

That's all for now, I hope this tutorial will be useful for someone!!!

For corrections and suggestions contact me at ume15@hotmail.com

Thanks to **Volatitlity** and all the **Immortal Descendants**.

UmE