



WINASO

Registry Optimizer 4.8.3

Programa para eliminar entradas de registro y optimizar el pc

Apuromafo y +Erisoft


16/08/2013

Introducción:

Hola, luego de un tiempo para poder aplicarme a la ingeniería inversa y de aplicarme en webs con un sistema de android, propuse a mi amigo apuromafo crear algún escrito con fines educativos ,si bien dice que no tiene mucho tiempo , nos las ingeniamos y salió el siguiente documento, aquí verán que se logró registrar con mover un byte y con algún nop, fueron en plataformas xp y Windows 7, Espero sea de su agrado Apuromafo y +Erisoft

Start: un programa creado hace algunos días:

Web: http://www.winaso.com/registry_optimizer/

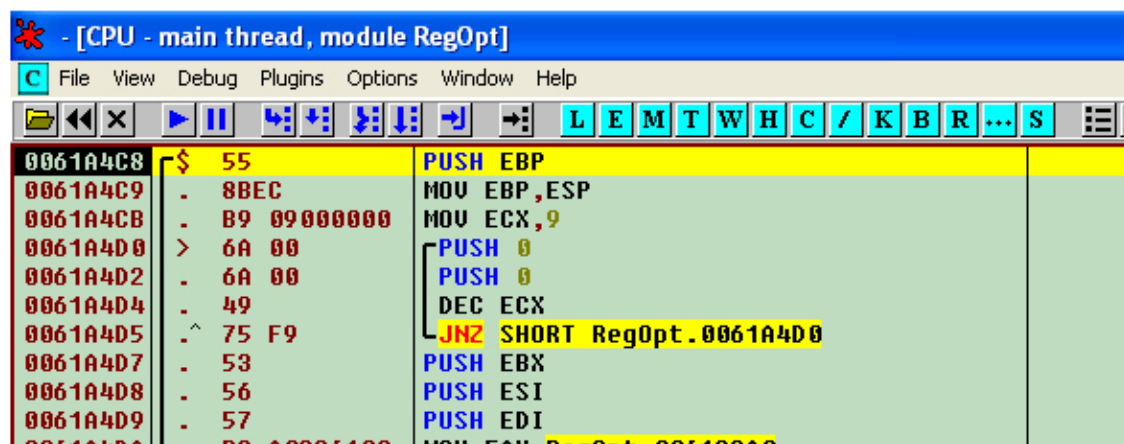
 **WinASO Registry Optimizer v4.8.3 - Aug 12, 2013**
New Feature - newly add free tools such as Auto Shutdown, Performance Monitor, Registry Cleaner and System Optimizer functions are improved as well.

WinASO Registry Optimizer
Version: 4.8.3
Update: Aug 12, 2013
Size: 7.62 MB

[Download](#) [Buy Now](#)

Es un programa para optimizar un poco el equipo.

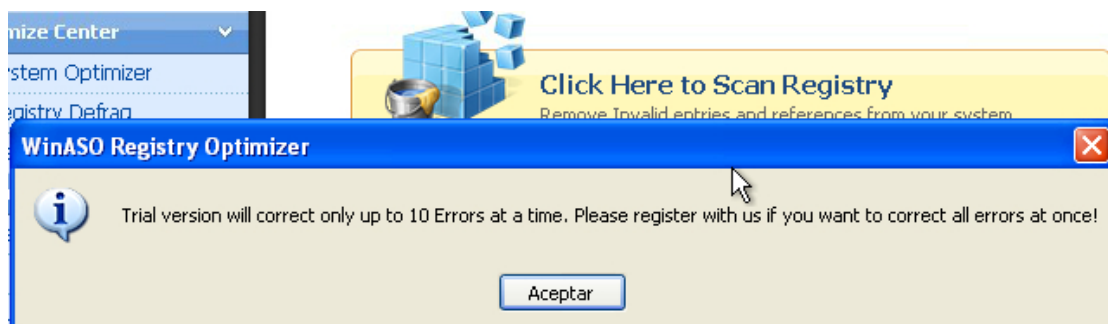
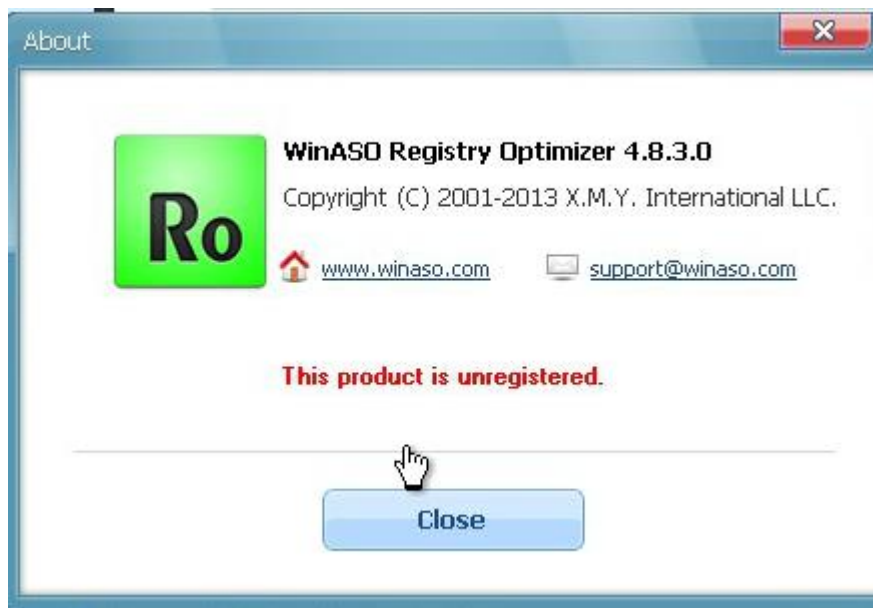
Instalo el programa, lo cargo en Ollydbg



Al no estar comprimido, lo ejecuto y reviso un poco el entorno



Limitaciones:



Ahora comenzaremos la acción, tenemos este mensaje (pauso)

0061258C	. 8085 2CFFFFFF	MOV EAX,[LOCAL:53]	
00612582	. BA 542C6100	MOV EDX,RegOpt.00612C54	
00612587	. E8 1047DFFF	CALL RegOpt.00406CCC	
0061258C	. 75 04	JNZ SHORT RegOpt.006125C2	
0061258E	. C645 A2 00	MOV BYTE PTR SS:[EBP-5E],0	
006125C2	> A1 843C6200	MOV EAX,DWORD PTR DS:[623C84]	Pondremos un BreakPoint para ver que pasa con los salt
006125C7	. 8038 00	CMP BYTE PTR DS:[EAX],0	
006125CA	. 75 38	JNZ SHORT RegOpt.00612604	
006125CC	. A1 8C406200	MOV EAX,DWORD PTR DS:[62408C]	
006125D1	. 8038 00	CMP BYTE PTR DS:[EAX],0	
006125D4	. 75 2E	JNZ SHORT RegOpt.00612604	
006125D6	. 6A 40	PUSH 40	ACA INICIA NUESTRO BAD BOY
006125D8	. A1 40386200	MOV EAX,DWORD PTR DS:[623840]	
006125DD	. 8B40 44	MOV EAX,DWORD PTR DS:[EAX+44]	
006125E0	. E8 5740DFFF	CALL RegOpt.0040663C	
006125E5	. 50	PUSH EAX	
006125E6	. A1 40386200	MOV EAX,DWORD PTR DS:[623840]	
006125EB	. 8B80 58010000	MOV EAX,DWORD PTR DS:[EAX+158]	
006125F1	. E8 4640DFFF	CALL RegOpt.0040663C	
006125F6	. 50	PUSH EAX	
006125F7	. 8BC3	MOV EAX,EBX	
006125F9	. E8 4A78E7FF	CALL RegOpt.00409E48	
006125FE	. 50	PUSH EAX	h0wner
006125FF	. E8 F07ADFFF	CALL <JMP.&user32.MessageBoxW>	MessageBoxW

Colocamos el BP en 6125C2 y reiniciamos el pulsar el botón (acepto el mensaje de la nag y vuelvo a pulsar el botón de scan)

006125B7	. E8 1047DFFF	CALL RegOpt.00406CCC	
006125BC	. 75 04	JNZ SHORT RegOpt.006125C2	
006125BE	. C645 A2 00	MOV BYTE PTR SS:[EBP-5E],0	
006125C2	> A1 843C6200	MOV EAX,DWORD PTR DS:[623C84]	Pondremos un BreakPoint
006125C7	. 8038 00	CMP BYTE PTR DS:[EAX],0	



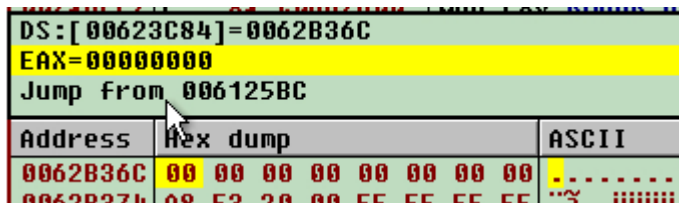
Cae en el bp

006125B7	. E8 1047DFFF	CALL RegOpt.00406CCC	
006125BC	. 75 04	JNZ SHORT RegOpt.006125C2	
006125BE	. C645 A2 00	MOV BYTE PTR SS:[EBP-5E],0	
006125C2	> A1 843C6200	MOV EAX,DWORD PTR DS:[623C84]	
006125C7	. 8038 00	CMP BYTE PTR DS:[EAX],0	
006125CA	. 75 38	JNZ SHORT RegOpt.00612604	
006125CC	. A1 8C406200	MOV EAX,DWORD PTR DS:[62408C]	
006125D1	. 8038 00	CMP BYTE PTR DS:[EAX],0	
006125D4	. 75 2E	JNZ SHORT RegOpt.00612604	
006125D6	. 6A 40	PUSH 40	
006125D8	. A1 40386200	MOV EAX,DWORD PTR DS:[623840]	
006125DD	. 8B40 44	MOV EAX,DWORD PTR DS:[EAX+44]	
006125E0	. E8 5740DFFF	CALL RegOpt.0040663C	
006125E5	. 50	PUSH EAX	
006125F1	. E8 4640DFFF	CALL RegOpt.0040663C	
006125F6	. 50	PUSH EAX	
006125F7	. 8BC3	MOV EAX,EBX	
006125F9	. E8 4A78E7FF	CALL RegOpt.00409E48	
006125FE	. 50	PUSH EAX	
006125FF	. E8 F07ADFFF	CALL <JMP.&user32.MessageBoxW>	
DS:[00623C84]=0062B36C			
EAX=00000000			
Jump from 006125BC			

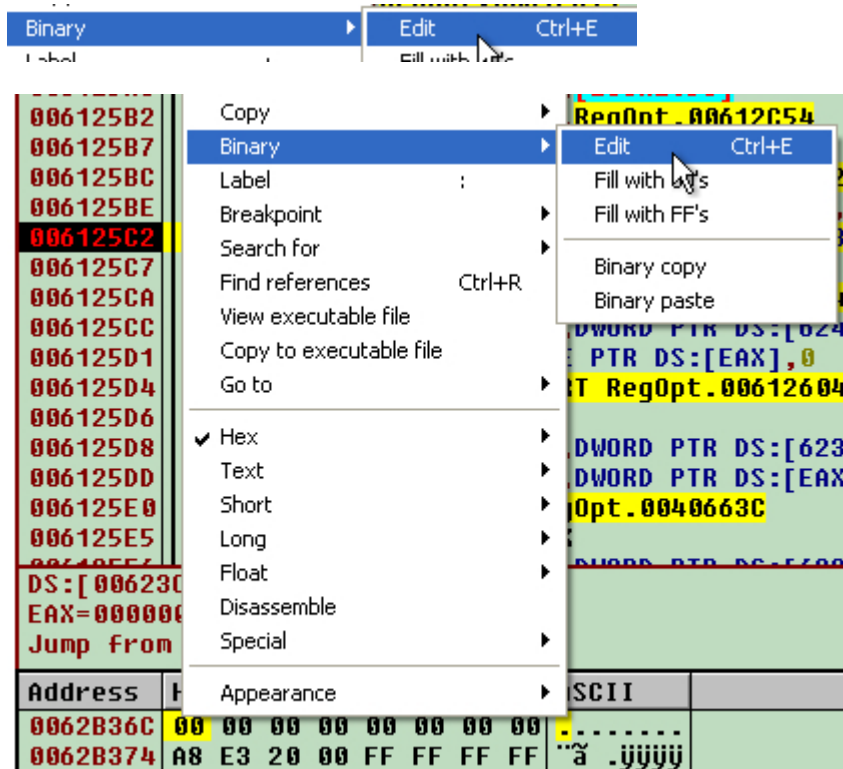
Luego

006125F1	. E8 4640DFFF	CALL RegOpt.0040663C	
006125F6	. 50	PUSH EAX	
006125F7	. 8BC3	MOV EAX,EBX	
006125F9	. E8 4A78E7FF	CALL RegOpt.00409E48	
006125FE	. 50	PUSH EAX	
006125FF	. E8 F07ADFFF	CALL <JMP.&user32.MessageBoxW>	
DS:[00623C84]=0062B36C			
EAX=0000			
Jump fro			
Address			
0061B000			
0061B000			

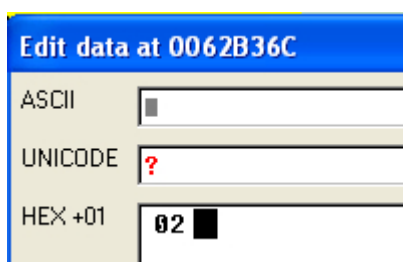
Al hacer el Follow Value in Dump tenemos el valor 0 (un poco mas arriba ví una comparación con 2, intentaré hacer este cambio)



Procedo a hacer el binary Edit



Y lo cambio al 2

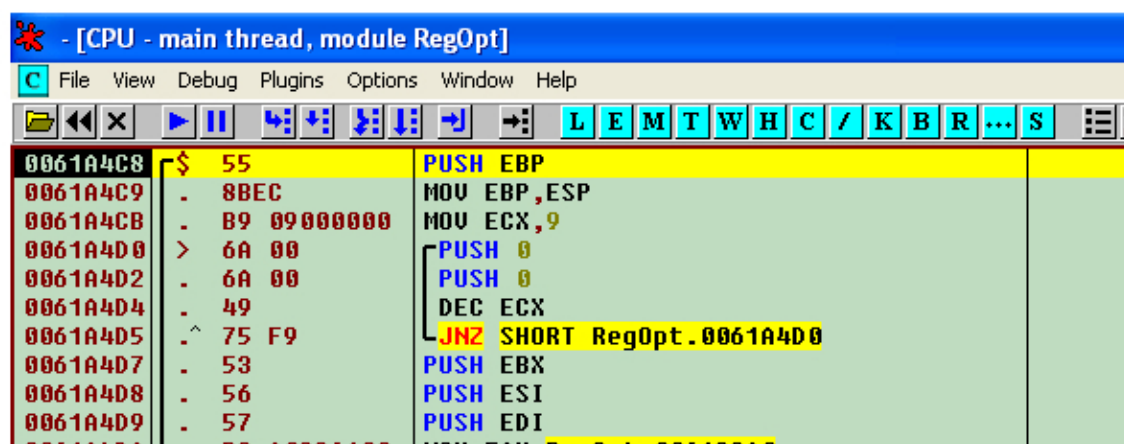


Luego miro el About

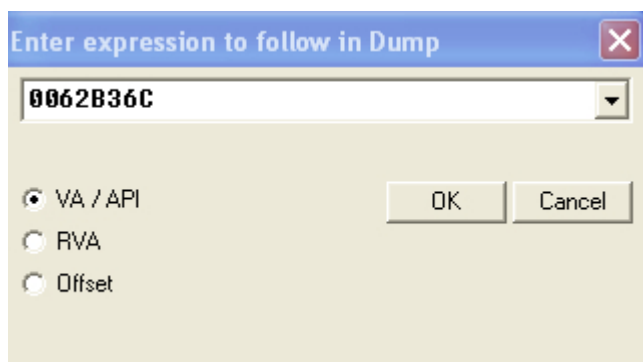


Así que lo primero deberá ser investigar cómo acceder a aquel byte que me permitió registrar a primera vista.

Reiniciando Ollydbg con el WINASO

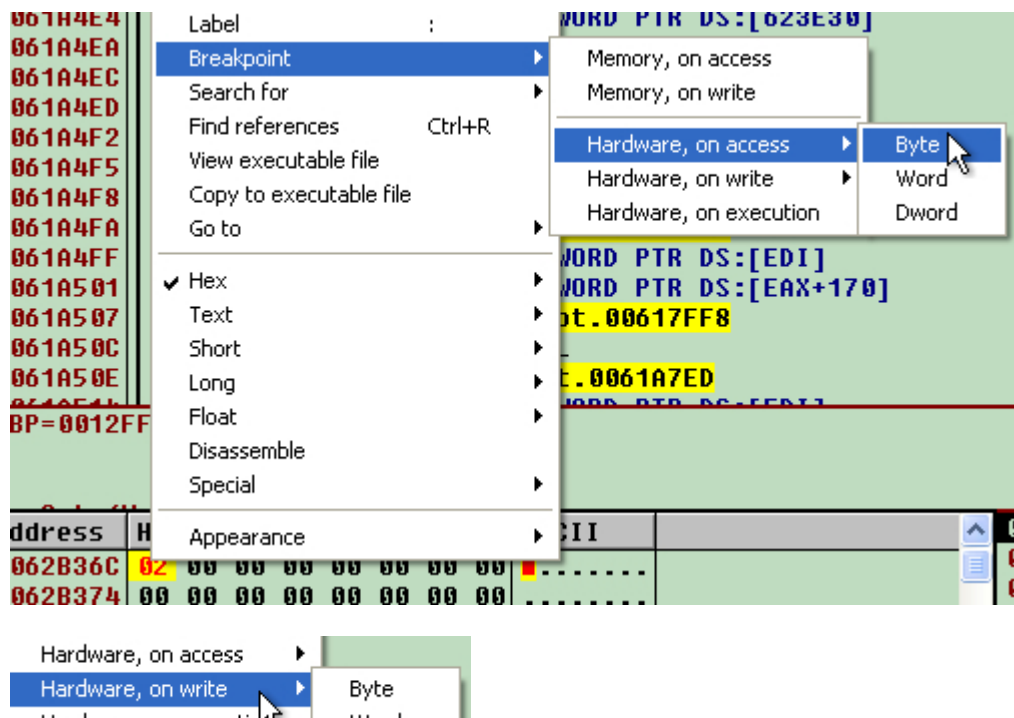


Así que en el dump vamos a ese byte



Address	Hex dump
0062B36C	00 00 00 00
0062B374	00 00 00 00

Cambio denuevo el valor a 2 (binary Edit) y luego un HW bp en el byte tanto de acceso como de escritura



Luego de ejecutar



Cae nuestro bp

00605242	- 8B83 88040000	MOV EAX,DWORD PTR DS:[EBX+408]	
00605248	- 8B80 14030000	MOV EAX,DWORD PTR DS:[EAX+314]	
0060524E	- 33D2	XOR EDX,EDX	
00605250	- E8 AFC2E7FF	CALL RegOpt.00481504	
00605255	> A1 1C396200	MOV EAX,DWORD PTR DS:[62391C]	
0060525A	- 8338 00	CMP DWORD PTR DS:[EAX],0	
0060525D	- 74 0A	JE SHORT RegOpt.00605269	
0060525F	- A1 1C396200	MOV EAX,DWORD PTR DS:[62391C]	Culpable de Hacernos Trial :=(
00605264	- 8338 01	CMP DWORD PTR DS:[EAX],1	
00605267	- 75 79	JNZ SHORT RegOpt.006052E2	
00605269	> A1 843C6200	MOV EAX,DWORD PTR DS:[623C84]	
0060526E	- C600 00	MOV BYTE PTR DS:[EAX],0	
00605271	- A1 40386200	MOV EAX,DWORD PTR DS:[623840]	
00605276	- FF70 44	PUSH DWORD PTR DS:[EAX+44]	
00605279	- 68 34536000	PUSH RegOpt.00605334	
0060527E	- 68 44536000	PUSH RegOpt.00605344	UNICODE "4.8.3"
00605283	- 68 5C536000	PUSH RegOpt.0060535C	UNICODE " ("
00605288	- A1 40386200	MOV EAX,DWORD PTR DS:[623840]	
0060528D	- FF70 48	PUSH DWORD PTR DS:[EAX+48]	

Hardware breakpoint 1 at RegOpt.00605271 - EIP points to next instruction

Al subir encontramos un salto que nos hace llegar a donde estamos

006051C0	- 8B00	MOV EAX,EAX	
006051C2	- 33C0	XOR EAX,EAX	
006051C4	- 55	PUSH EBP	
006051C5	- 68 1A536000	PUSH RegOpt.0060531A	
006051CA	- 64:FF30	PUSH DWORD PTR FS:[EAX]	
006051CD	- 64:8920	MOV DWORD PTR FS:[EAX],ESP	
006051D0	- A1 1C396200	MOV EAX,DWORD PTR DS:[62391C]	
006051D5	- 8338 02	CMP DWORD PTR DS:[EAX],2	
006051D8	- 75 7B	JNZ SHORT RegOpt.00605255	
006051DA	- A1 843C6200	MOV EAX,DWORD PTR DS:[623C84]	
006051DF	- C600 01	MOV BYTE PTR DS:[EAX],1	
006051E2	- A1 40386200	MOV EAX,DWORD PTR DS:[623840]	
006051E7	- FF70 44	PUSH DWORD PTR DS:[EAX+44]	
006051EA	- 68 34536000	PUSH RegOpt.00605334	
006051EF	- 68 44536000	PUSH RegOpt.00605344	UNICODE "4.8.3"
006051F4	- 8D45 FC	LEA EAX,[LOCAL.1]	
006051F7	- BA 03000000	MOV EDI,3	
006051FC	- E8 2319E0FF	CALL RegOpt.00406B24	
00605201	- 8B55 FC	MOV EDI,[LOCAL.1]	
00605204	- A1 8880F000	MOV EAX,DWORD PTR DS:[0F80881]	

Lo comentamos

006051C4	- 55	PUSH EBP	
006051C5	- 68 1A536000	PUSH RegOpt.0060531A	
006051CA	- 64:FF30	PUSH DWORD PTR FS:[EAX]	
006051CD	- 64:8920	MOV DWORD PTR FS:[EAX],ESP	
006051D0	- A1 1C396200	MOV EAX,DWORD PTR DS:[62391C]	
006051D5	- 8338 02	CMP DWORD PTR DS:[EAX],2	
006051D8	- 75 7B	JNZ SHORT RegOpt.00605255	
006051DA	- A1 843C6200	MOV EAX,DWORD PTR DS:[623C84]	
006051DF	- C600 01	MOV BYTE PTR DS:[EAX],1	

Add comment at 006051D8

Si brinca somos trial :=(

OK

006051C2	- 33C0	XOR EAX,EAX	
006051C4	- 55	PUSH EBP	
006051C5	- 68 1A536000	PUSH RegOpt.0060531A	
006051CA	- 64:FF30	PUSH DWORD PTR FS:[EAX]	
006051CD	- 64:8920	MOV DWORD PTR FS:[EAX],ESP	
006051D0	- A1 1C396200	MOV EAX,DWORD PTR DS:[62391C]	
006051D5	- 8338 02	CMP DWORD PTR DS:[EAX],2	
006051D8	- 90	NOP	
006051D9	- 90	NOP	

Assemble at 006051D9

nop

☒ Fill with NOP's

Assemble

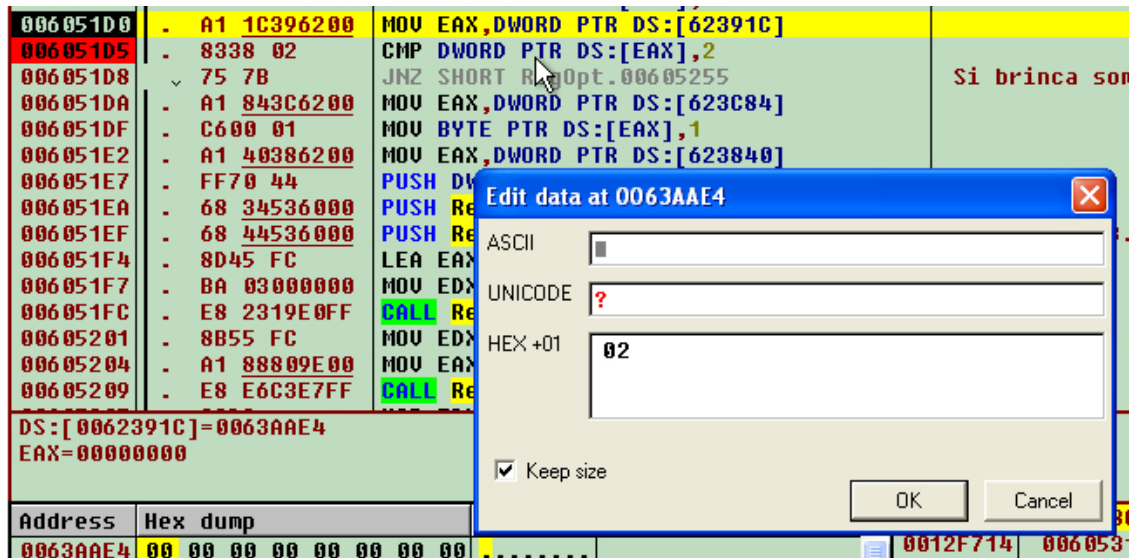
Cancel

Si brinca somos trial :=(Asi que NOP

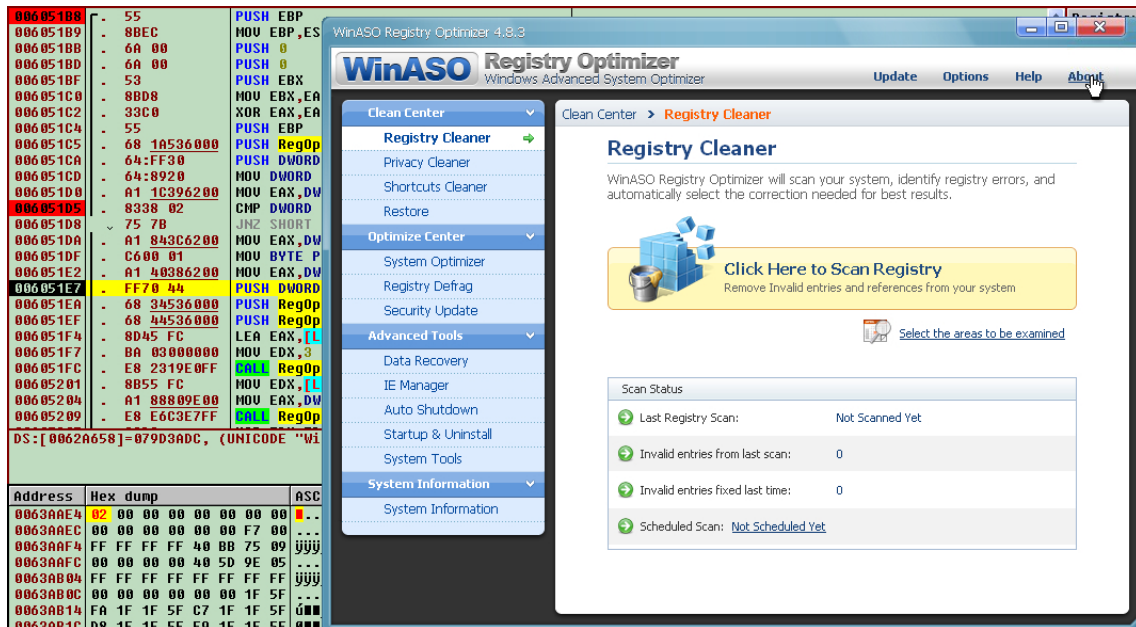
Pero con esto no es suficiente, pues debemos Tener la seguridad que ese byte sea 2

6051D5 compara que sea 2, por ende ahora lo fuerzo a que sea 2

MOV BYTE PTR DS:[EAX],2



Luego de pasar el salto lo vemos así el programa (full)



Como ya testee lo más básico Podemos continuar en otro s.o y confirmar si la dirección es la misma:

Apuromafo: Ahora confirmo en otro equipo y la dirección no es la misma, así que buscaré un pattern de la comparación con el 2 y con el salto (833802757b)

83 38 02 75 7B

Al encontrarlo:

```

00613064 . 33      PUSH EBP
00613065 . 8BEC    MOV EBP,ESP
00613067 . 6A 00   PUSH 0
00613069 . 6A 00   PUSH 0
0061306B . 53      PUSH EBX
0061306C . 8BD8    MOV EBX,EAX
0061306E . 33C0    XOR EAX,EAX
00613070 . 55      PUSH EBP
00613071 . 68 C6316100 PUSH 00613106
00613076 . 64:FF30 PUSH DWORD PTR FS:[EAX]
00613079 . 64:8920 MOV DWORD PTR FS:[EAX],ESP
0061307C . A1 1C396200 MOV EAX,DWORD PTR DS:[62391C]
00613081 . 8338 02 CMP DWORD PTR DS:[EAX],2
00613084 . 75 7B   JNE SHORT 00613101
00613086 . A1 843C6200 MOV EAX,DWORD PTR DS:[623C84]
0061308B . C600 01 MOV BYTE PTR DS:[EAX],1
0061308E . A1 40386200 MOV EAX,DWORD PTR DS:[623840]
00613093 . FF70 44 PUSH DWORD PTR DS:[EAX+44]
00613096 . 68 F0316100 PUSH 006131E0

```

Procedo a hacer el cambio con mov byte y el valor 2 "MOV BYTE PTR DS:[EAX],2"

00613064	. 00	DB 00	ASCII "Main"	Registers (FPU)
00613065	. 00	DB 04		
0061306D	. 4D 61 69 6E	ASCII "Main"		
00613061	. 00	DB 00		
00613062	. 00	NOP		
00613063	. 55	PUSH EBP		
00613064	. 8BEC	MOV EBP,ESP		
00613067	. 6A 00	PUSH 0		
00613069	. 6A 00	PUSH 0		
0061306A	. 53	PUSH EBX		
0061306C	. 8BD8	MOV EBX,EAX	Installs SE handler 613106	EAX 0062A1B0 RegOpt.0062A1B0 ECX 00613064 Entry point EDX 0012F948 EBX 02064370 ESP 0012F764 EBP 0012F77C ESI 02064370 EDI 0012F3D4 EIP 00613081 RegOpt.00613081 C 0 ES 0023 32bit 0 (FFFFFFFF) P 1 CS 001B 32bit 0 (FFFFFFFF) D 0 SS 0023 32bit 0 (FFFFFFFF) I 1 DS 0023 32bit 0 (FFFFFFFF) F 0 FS 0038 32bit 7 (7FDF00014000) T 0 GS 0000 NULL D 0 0 LastErr 00000002 ERROR_FILE_NOT_FOUND EFL 00000246 (NO,NB,E,BS,PE,GE,LE) ST0 empty 0.0 ST1 empty 0.0 ST2 empty 0.0 ST3 empty 1.379246713444726000e+16 ST4 empty 2.9555870130585701000e+16 ST5 empty 2.955587013044453000e+16 ST6 empty 3.01182649629073000e+16 ST7 empty 2.955345807902765000e+16 3 2 1 0 ESP U 0 Z 0 I FST aux0 Conid 1 0 0 0 Err 0 0 0 0 0 0 0 0 (E0) FCW 1372 Proc NEAR,64 Mask 1 1 0 0 1 0 Last cnnd 0001B:00E73222 Register.00E73222
0061306E	. 33C0	XOR EAX,EAX		
00613070	. 55	PUSH EBP		
00613071	. 68 C6316100	PUSH 00613106		
00613076	. 64:FF30	PUSH DWORD PTR FS:[EAX]		
00613079	. 64:8920	MOV DWORD PTR FS:[EAX],ESP		
0061307C	. 61 1C396200	MOV EAX,DWORD PTR DS:[62391C]		
00613081	. C600 02	MOV BYTE PTR DS:[EAX],2		
00613084	. 90	NOP		
00613085	. 90	NOP		
00613086	. A1 843C6200	MOV EAX,DWORD PTR DS:[623C84]		
0061308B	. C600 01	MOV BYTE PTR DS:[EAX],1		
0061308E	. A1 40386200	MOV EAX,DWORD PTR DS:[623840]		
00613093	. FF70 44	PUSH DWORD PTR DS:[EAX+44]		
00613096	. 68 F0316100	PUSH 00613100		
00613099	. 68 F0316100	PUSH 00613100		
0061309B	. 8045 FC	LEA EAX,[LOCAL.1]		
0061309D	. B8 03000000	MOV EDI,3		
0061309E	. E8 773ADFFF	CALL 00406B24		
0061309F	. 8B55 FC	MOV EDI,DWORD PTR SS:[LOCAL.1]		
006130A0	. A1 8C309E00	MOV EAX,DWORD PTR DS:[9E980C]		
006130A5	. E8 3AE5E6FF	CALL 004815F4		
006130A7	. 33D2	XOR EDI,EDI		
006130A8	. 8B53 8C040000	MOV EAX,DWORD PTR DS:[EBX+48C]		
006130AC	. E8 3DE4E6FF	CALL 00481504		
006130AD	. 8B53 B8040000	MOV EAX,DWORD PTR DS:[EBX+4B8]		
[FPU=82 [0062A1B0]=000			CRegOpt.00481504	XTM0 00000000 00000000 00000000 00000000 XTM1 00000000 00000000 00000000 00000000 XTM2 00000000 00000000 00000000 00000000 XTM3 00000000 00000000 00000000 00000000 XTM4 00000000 00000000 00000000 00000000 XTM5 00000000 00000000 00000000 00000000 XTM6 00000000 00000000 00000000 00000000 XTM7 00000000 00000000 00000000 00000000