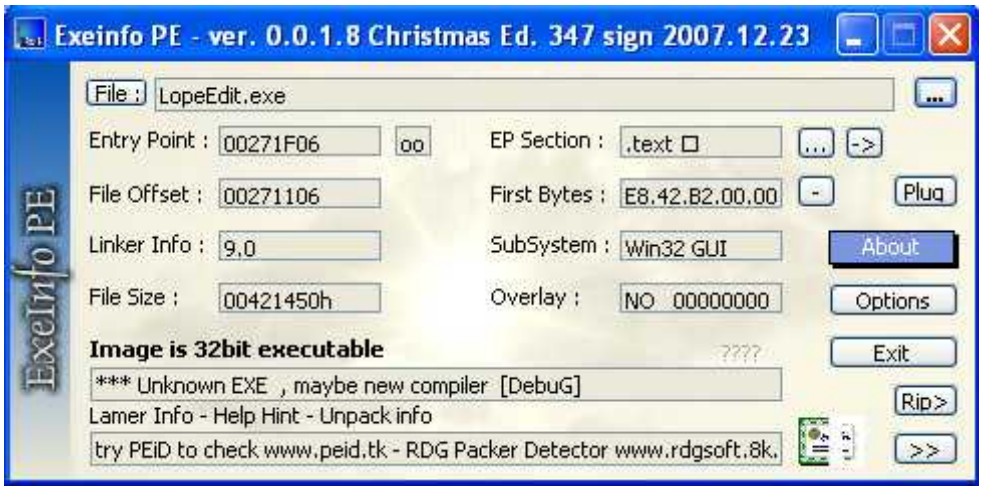
LopeEdit Pro 5.6.1 por Aguml

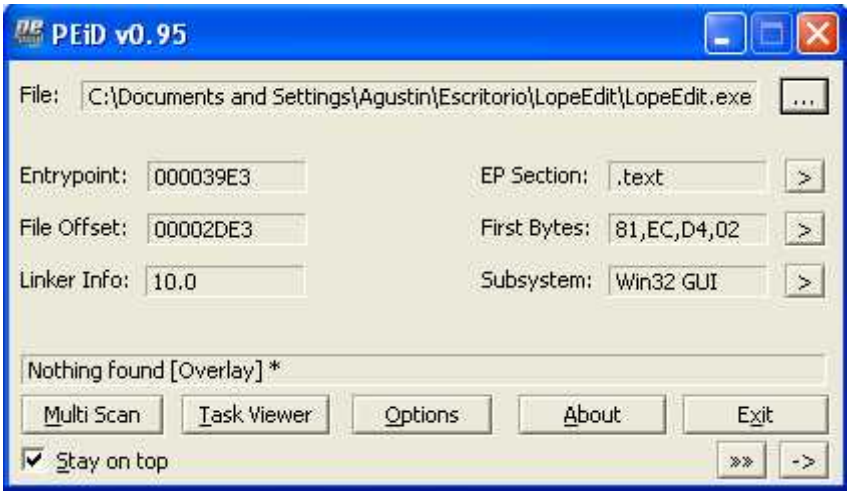
Fecha	**5 de junio de 2014**
Victima	**http://www.programacionfacil.com**
URL de descarga	**http://www.lopesoft.com/es/lopeedit/download?download=26:lopeedit**
Herramientas	**“OllyDbg 1.10”, “WinHex 12.25 SR-3”, “LordPE Deluxe”**
Objetivo	**Conseguir registrar su versión Pro.**

INDICE

1. Examinando a la víctima

Lo primero es ver con que está hecha:

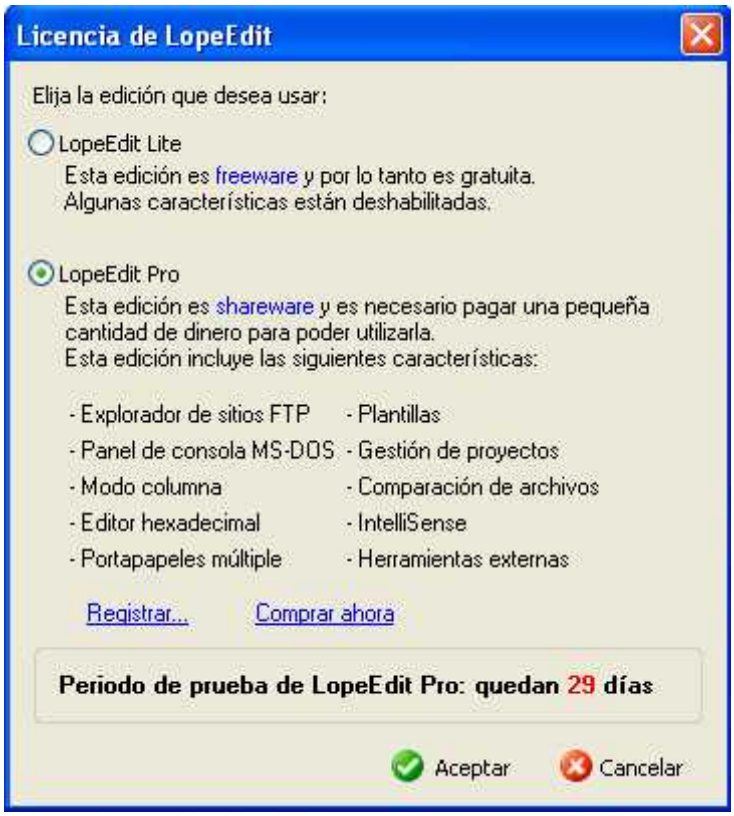




Ya se que no serán las versiones mas nuevas pero son las que vienen en el “Todo en uno” que tengo a mano.
Si lo vemos en Olly vemos esto:

Address	Hex dump	Disassembly		Comment
00671F06	E8 42B20000	call	0067D14D	LopeEdit.0067D14D
00671F0B	E9 79FEFFFF	jmp	00671D89	LopeEdit.00671D89
00671F10	8BFF	mov	edi, edi	
00671F12	55	push	ebp	
00671F13	8BEC	mov	ebp, esp	
00671F15	51	push	ecx	
00671F16	53	push	ebx	
00671F17	8B45 0C	mov	eax, ss:[ebp+C]	
00671F1A	83C0 0C	add	eax, 0C	
00671F1D	8945 FC	mov	ss:[ebp-4], eax	
00671F20	64:8B1D 000000(	mov	ebx, fs:[0]	

La verdad es que no estoy seguro si es un Visual C++ 6 ya que no conozco como es el OEP de este pero es lo que tengo.
Ejecuto el programa y veo que me da a elegir entre Lite y Pro y la Pro sólo me deja 30 días.



Si pasan los 30 días, aunque te deja elegirlo, arranca con la versión Lite.

Si doy a Registrar... me sale una ventana para que busque un archivo llamado LopeEditPro.lic.

2. Buscando un registro válido

Lo primero que hago es buscar esa cadena en las strings y pongo BPs en todas las que salen.
Arranco y para aquí:

```
004902BF    68 B0036F00     push    6F03B0                  ; UNICODE "LopeEditPro.lic"
```

Traceo un poco y al llegar aquí:

```
004902E3    50              push    eax
```

veo esto:

```
eax=01109460, (UNICODE "C:\Documents and Settings\USUARIO\Datos de
programa\LopeSoft\LopeEdit\LopeEditPro.lic")
```

Es donde busca la licencia así que voy a esa ruta y creo el archivo con algo dentro y reinicio a ver que pasa.
Si existe ese archivo en esa ruta no salta en el siguiente salto:

```
004902EE   /74 31           je      short 00490321              ; LopeEdit.00490321
```

Y si no existe lo buscará en el directorio del programa.
Sigo traceando y, una vez paso el retn, llego aquí:

```
0048E6CD    E8 BE1B0000     call    00490290                //Aquí busca el archivo de
licencia
0048E6D2    C745 FC 0200000>mov     dword ptr ss:[ebp-4], 2
0048E6D9    6A 00           push    0
0048E6DB    8D4D E4         lea     ecx, ss:[ebp-1C]
0048E6DE    E8 BDA8F8FF     call    00418FA0                        ; LopeEdit.00418FA0
0048E6E3    50              push    eax
0048E6E4    8B4D C4         mov     ecx, ss:[ebp-3C]
0048E6E7    83C1 7C         add     ecx, 7C
0048E6EA    E8 A1DE0700     call    0050C590                //Aquí lee los datos del mismo
```

He salido a la línea 0048E6D2 así que voy traceando a ver que va haciendo y al entrar en 0048E6EA veo esto:

```
0050C7D7    68 A8407000     push    7040A8                          ; UNICODE "License"
0050C7DC    8D8D E0EFFFFF   lea     ecx, ss:[ebp-1020]
0050C7E2    51              push    ecx
0050C7E3    E8 5854EFFF     call    00401C40                        ; LopeEdit.00401C40
```

Busco Apis que puedan abrir el archivo y leerlo y pongo BPs en todas pero no obtengo resultados.
Durante unos minutos me quedé en blanco y me volví loco pensando como hincarle el diente y se me ocurrió buscar strings de tipo User, Pass, Key... y, después de un rato, encontré esto:

```
0050B306   push    703D38                          UNICODE "Microsoft Base Cryptographic
Provider v1.0"
0050B323   push    703D90                          UNICODE "Microsoft Base Cryptographic
Provider v1.0"
0050B5D7   push    703DE8                          UNICODE "%08X"
0050B692   push    703DF8                          UNICODE "Microsoft Base Cryptographic
Provider v1.0"
0050BAEF   mov     eax, 50BB04                     ASCII "ÆEü"
0050BB99   mov     eax, 50BBAE                     ASCII "ÆEü"
0050BC68   push    703E50                          UNICODE "Microsoft Base Cryptographic
Provider v1.0"
0050C197   mov     dword ptr ss:[ebp+1C], 703EAC   UNICODE "00000000"
0050C1BD   push    703EC0                          UNICODE "%Y/%m/%d"
0050C1F5   push    703ED4                          UNICODE "0000/00/00"
0050C22D   push    703EEC                          UNICODE "%s|%s|%s|%s|%s|%s|%d"
0050C36C   push    703F18                          UNICODE "%02X"
0050C3B5   push    703F24                          UNICODE "[License]",LF
0050C3DB   push    703F3C                          UNICODE "UserName=%s",LF
0050C401   push    703F58                          UNICODE "Company=%s",LF
0050C427   push    703F70                          UNICODE "E-Mail=%s",LF
0050C44D   push    703F88                          UNICODE "NoUsers=%d",LF
0050C46F   push    703FA0                          UNICODE "00000000"
0050C486   push    703FB4                          UNICODE "HardwareID=%s",LF
0050C4B2   push    703FD4                          UNICODE "ExpirationDate=%s",LF
0050C4D8   push    703FFC                          UNICODE "LicenseKey=%s",LF,LF
0050C602   push    704020                          UNICODE "00000000"
0050C65C   push    704034                          UNICODE "License"
0050C676   push    704044                          UNICODE "License%d"
0050C69F   push    70405C                          UNICODE "UserName"
0050C6CF   push    704074                          UNICODE "LicenseKey"
0050C6FF   push    704090                          UNICODE "HardwareID"
0050C7D7   push    7040A8                          UNICODE "License"
0050C7F4   push    7040B8                          UNICODE "License%d"
0050C81D   push    7040D0                          UNICODE "UserName"
0050C82E   call    ds:[6E0478]                     (Initial CPU selection)
0050C851   push    7040E8                          UNICODE "LicenseKey"
```

```
0050C881   push    704104                          UNICODE "HardwareID"
0050C8B2   push    704120                          UNICODE "00000000"
0050CA8E   push    704134                          UNICODE "0000/00/00"
0050CA9D   push    70414C                          UNICODE "%Y/%m/%d"
0050CB94   push    704160                          UNICODE "%s|%s|%s|%s|%s|%s|%d"
0050CBC0   push    70418C                          UNICODE "%u"
0050CFE1   push    70419C                          UNICODE "HardwareID"
0050D011   push    7041B8                          UNICODE "00000000"
0050D0FE   push    7041D4                          UNICODE "UserName"
0050D1DE   push    7041F0                          UNICODE "Company"
0050D2BE   push    704208                          UNICODE "E-Mail"
0050D377   push    70421C                          UNICODE "NoUsers"
0050D447   push    704230                          UNICODE "ExpirationDate"
0050D4AB   push    704250                          UNICODE "%d/%d/%d"
0050D58E   push    70426C                          UNICODE "LicenseKey"
```

Busco esas cadenas y a los pocos intentos me encuentro con esto:

```
0050C69F    68 5C407000     push    70405C              ; UNICODE "UserName"
0050C6A4    8D8D E8F7FFFF   lea     ecx, ss:[ebp-818]
0050C6AA    E8 F1C8F0FF     call    00418FA0            ; LopeEdit.00418FA0
0050C6AF    50              push    eax
0050C6B0    FF15 78046E00   call    ds:[6E0478]         ; kernel32.GetPrivateProfileStringW
```

Y busco la Api GetPrivateProfileString en la MSDN y dice esto:

> *Retrieves a string from the specified section in an initialization file.*
>
> *Note This function is provided only for compatibility with 16-bit Windows-based applications. Applications should store initialization information in the registry.*

Luego leo que busca algo asi en el archivo:

```
[section]
key=string
      .
      .
      .
```

La verdad es que no vi en ningún momento donde inicializa el archivo pero pondré un BP en todas las llamadas a esa Api a ver que pasa y la primera vez que para veo esto:

```
0012E684   0110CEE0  |Section = "License"
0012E688   007040D0  |Key = "UserName"
0012E68C   007040CC  |Default = ""
0012E690   0012E6A8  |ReturnBuffer = 0012E6A8
0012E694   00000800  |BufSize = 800 (2048.)
0012E698   0110D808  \IniFileName = "C:\Documents and Settings\USUARIO\Datos de
programa\LopeSoft\LopeEdit\LopeEditPro.lic"
```

Y si vamos al buffer y pasamos la Api vemos que no recupera nada. Leyendo la info anterior de la Api vemos que tengo que hacer algo así en el archivo de licencia:

```
[License]
```

```
UserName=Agustin
```

Así que reinicio Olly y modifico el archivo de licencia por lo de arriba y al llegar a la Api la pasamos y ahora en el buffer si obtiene mi nombre jejeje.
Pues ahora toca ir a ver que pone en cada llamada a esa Api e ir completando nuestro archivo de licencia y así quedó:

```
[License]
UserName=Agustin
LicenseKey=FIACA
HardwareID=98989898
```

Luego comprueba que el HardwareID no esté vacio:

```
0050C8A8    FF15 44056E00   call    ds:[6E0544]                     ; kernel32.lstrcmpW
0050C8AE    85C0            test    eax, eax
0050C8B0    74 16           je      short 0050C8C8                  ; LopeEdit.0050C8C8

0012EDD0   0012EDE4  |String1 = "98989898"
0012EDD4   0070411C  \String2 = ""
```

Y como ya existe pues no salta.
A continuacion vuelve a comparar pero ahora con "00000000":

```
0050C8B2    68 20417000     push    704120                          ; UNICODE "00000000"
0050C8B7    8D95 E4EFFFFF   lea     edx, ss:[ebp-101C]
0050C8BD    52              push    edx
0050C8BE    FF15 44056E00   call    ds:[6E0544]                     ; kernel32.lstrcmpW
0050C8C4    85C0            test    eax, eax
0050C8C6    75 15           jnz     short 0050C8DD                  ; LopeEdit.0050C8DD

0012EDD0   0012EDE4  |String1 = "98989898"
0012EDD4   00704120  \String2 = "00000000"
```

Y ahí si que salta así que sigo traceando y veo que vuelve a hacer uso de la Api pero ahora busca otra sección:

```
0050C817    52              push    edx
0050C818    68 CC407000     push    7040CC
0050C81D    68 D0407000     push    7040D0                          ; UNICODE "UserName"
0050C822    8D8D E0EFFFFF   lea     ecx, ss:[ebp-1020]
0050C828    E8 73C7F0FF     call    00418FA0                        ; LopeEdit.00418FA0
0050C82D    50              push    eax
0050C82E    FF15 78046E00   call    ds:[6E0478]                     ;
kernel32.GetPrivateProfileStringW

0012EDC0   0110C078  |Section = "License1"
0012EDC4   007040D0  |Key = "UserName"
0012EDC8   007040CC  |Default = ""
0012EDCC   0012EDE4  |ReturnBuffer = 0012EDE4
0012EDD0   00000800  |BufSize = 800 (2048.)
0012EDD4   01109460  \IniFileName = "C:\Documents and Settings\USUARIO\Datos de
programa\LopeSoft\LopeEdit\LopeEditPro.lic"
```

Vi que era un bucle que iba leyendo todas las licencias así que creé unas pocas a ver que pasaba y al leer la ultima veo que sale del call y, después de terminar de leer todas. La verdad es que no veo como lo hace para comprobar

esos datos pero del call tiene que salir valiendo 1 y sale valiendo 0 así que como no se como seguir para obtener los datos correctos pues opto por parchear.

3. Parcheando para quitar limitaciones

Una vez salimos del call llegamos a la comprobación de la línea 0048E6EF:

```
0048E6EA    E8 A1DE0700     call    0050C590                        ; LopeEdit.0050C590
0048E6EF    85C0            test    eax, eax
0048E6F1    74 2C           je      short 0048E71F                  ; LopeEdit.0048E71F
```

Y una vez hacemos que no salte, a continuación, nos encontramos con esto:

```
0048E6F3    8B4D C4         mov     ecx, ss:[ebp-3C]
0048E6F6    83C1 7C         add     ecx, 7C
0048E6F9    E8 62E20700     call    0050C960                        ; LopeEdit.0050C960
0048E6FE    85C0            test    eax, eax
0048E700    74 1D           je      short 0048E71F                  ; LopeEdit.0048E71F
```

Al salir de ese otro call tampoco tiene que saltar así que eax no puede valer 0.
Esos dos saltos no pueden tomarse así que si nopeamos esos dos saltos ya estaría a full el programa pero por si acaso hice una búsqueda a ver si se llamaba más veces a ambas calls y esto es lo que vi:

Para el primer Call:

```
References in LopeEdit:.text to 0050C590
Address    Disassembly                              Comment
0048E5A0   call    0050C590                         LopeEdit.0050C590
0048E6EA   call    0050C590                         LopeEdit.0050C590
0048E84B   call    0050C590                         LopeEdit.0050C590
004900F8   call    0050C590                         LopeEdit.0050C590
004901F9   call    0050C590                         LopeEdit.0050C590
```

Casi todos tienen la comprobación y el salto condicional justo debajo pero hay uno que no tiene ni la comprobación ni el salto condicional así que nopear los condicionales igual no sea la mejor solución así que busqué otra manera de hacer que, al retornar, eax valiese 1 y es cambiando dentro del call este salto condicional de 0050C91B por un jmp con lo que no pondría eax a 0 y pasaríamos la primera condición:

```
0050C91B  |.  85C9          test    ecx, ecx
0050C91D   /74 04           je      short 0050C923                  ; LopeEdit.0050C923
0050C91F   |33C0           xor     eax, eax
0050C921   |EB 17           jmp     short 0050C93A                  ; LopeEdit.0050C93A
0050C923   \8B55 08         mov     edx, ss:[ebp+8]
```

Y las referencias para la otra call son:

```
References in LopeEdit:.text to 0050C960
Address    Disassembly                              Comment
0048DEFE   call    0050C960                         LopeEdit.0050C960
0048E5AF   call    0050C960                         LopeEdit.0050C960
0048E6F9   call    0050C960                         (Initial CPU selection)
0048E85E   call    0050C960                         LopeEdit.0050C960
00490107   call    0050C960                         LopeEdit.0050C960
```

Para ésta si que se cumple que debajo de todas hay la comprobación y el condicional así que podríamos modificar todos esos saltos y listo aunque si entramos y vemos donde puede convertir a eax en 0 veremos que tenemos que parchear todos estos saltos para conseguir lo mismo:

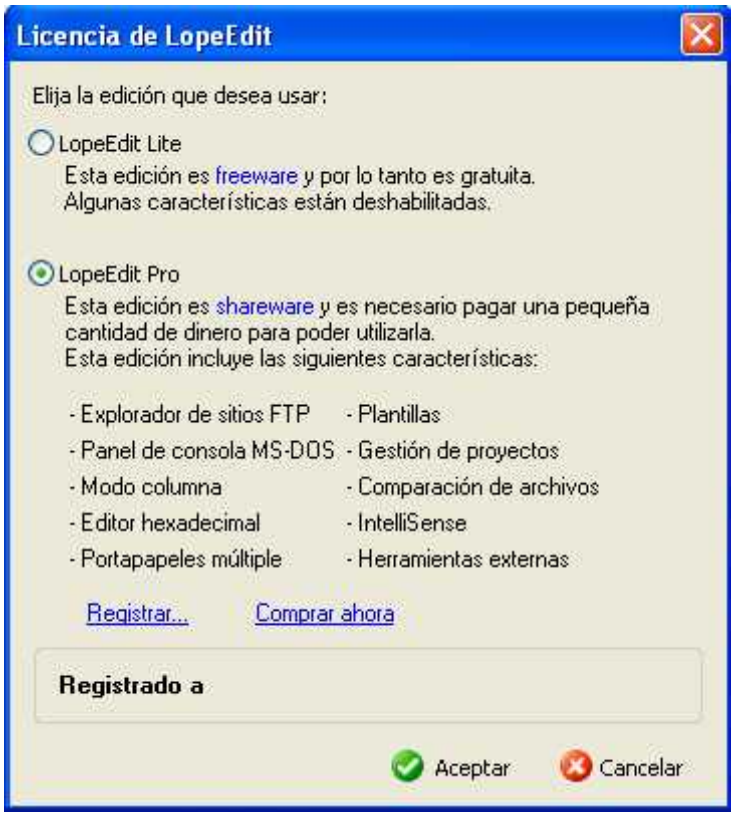
Address	Size	State	Old		New		Comment
0050C9D8	2.	Active	je	short 0050C9E1	jmp	short 0050C9E1	tiene que saltar
0050CAEC	2.	Active	jnz	short 0050CB0C	nop		no tiene que saltar
0050CAFB	2.	Active	jnz	short 0050CB0C	nop		no tiene que saltar
0050CB0A	2.	Active	je	short 0050CB66	jmp	short 0050CB66	tiene que saltar
0050CD60	2.	Active	je	short 0050CDE0	jmp	short 0050CDE0	tiene que saltar
0050CDFB	6.	Active	jl	0050CEAA	jmp	0050CEAA	tiene que saltar

¿Por qué tantos cambios? Porque hay que evitar que se ejecute cualquier salto incondicional que apunte a 0050CF29 y se ve que hay varias comprobaciones.

Por lo tanto podríamos elegir entre los cambios de la imagen de arriba en la que hacemos los cambios dentro de la call o los cambios de la siguiente imagen en la que parcheamos los saltos que hay después de cada call aunque yo opto más por la de arriba ya que al modificar la call evitamos que cualquier llamada oculta nos fastidie.

Address	Size	State	Old		New	
0048DF05	2.	Active	jnz	short 0048DF1D	jmp	short 0048DF1D
0048E5B6	2.	Active	je	short 0048E5E5	nop	
0048E700	2.	Active	je	short 0048E71F	nop	
0048E865	6.	Active	je	0048E97C	nop	
0049010E	2.	Active	je	short 0049011C	nop	
0050C91D	2.	Active	je	short 0050C923	jmp	short 0050C923

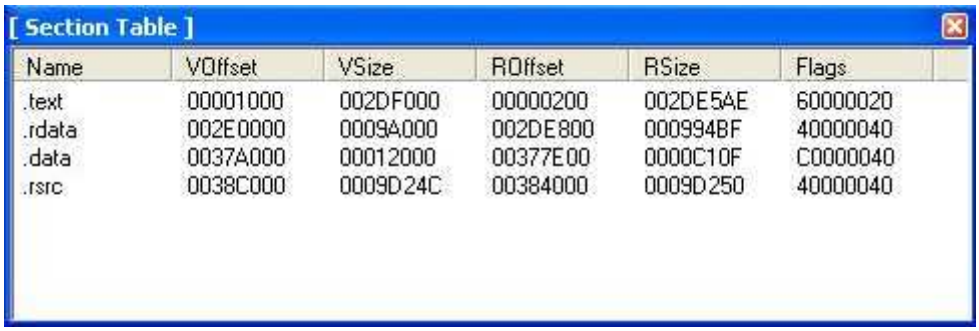
Con esas modificaciones ya nos libramos de todas las comprobaciones de esas dos calls.
Si ejecutamos la aplicación vemos que ya sale registrado pero no pone nombre alguno.



4. Maquillando el resultado

Después de darle muchas vueltas no encontré la manera de hacer que aparezca mi nombre y se me ocurrió entrar como elefante en cacharrería.
El binario no tiene espacio ni para una simple cadena así que al abrirlo en LordPE veo que no está alineado y que al darle al Rebuild PE del LordPE me deja un binario inservible así que toca arreglar eso a mano.

Originales:



Name	VOffset	VSize	ROffset	RSize	Flags
.text	00001000	002DF000	00000200	002DE5AE	60000020
.rdata	002E0000	0009A000	002DE800	000994BF	40000040
.data	0037A000	00012000	00377E00	0000C10F	C0000040
.rsrc	0038C000	0009D24C	00384000	0009D250	40000040

Alineadas:

[Section Table]

Name	VOffset	VSize	ROffset	RSize	Flags
.text	00001000	002DF000	00000200	002DE600	60000020
.rdata	002E0000	0009A000	002DE800	00099600	40000040
.data	0037A000	00012000	00377E00	0000C200	C0000040
.rsrc	0038C000	0009D24C	00384000	0009D250	40000040

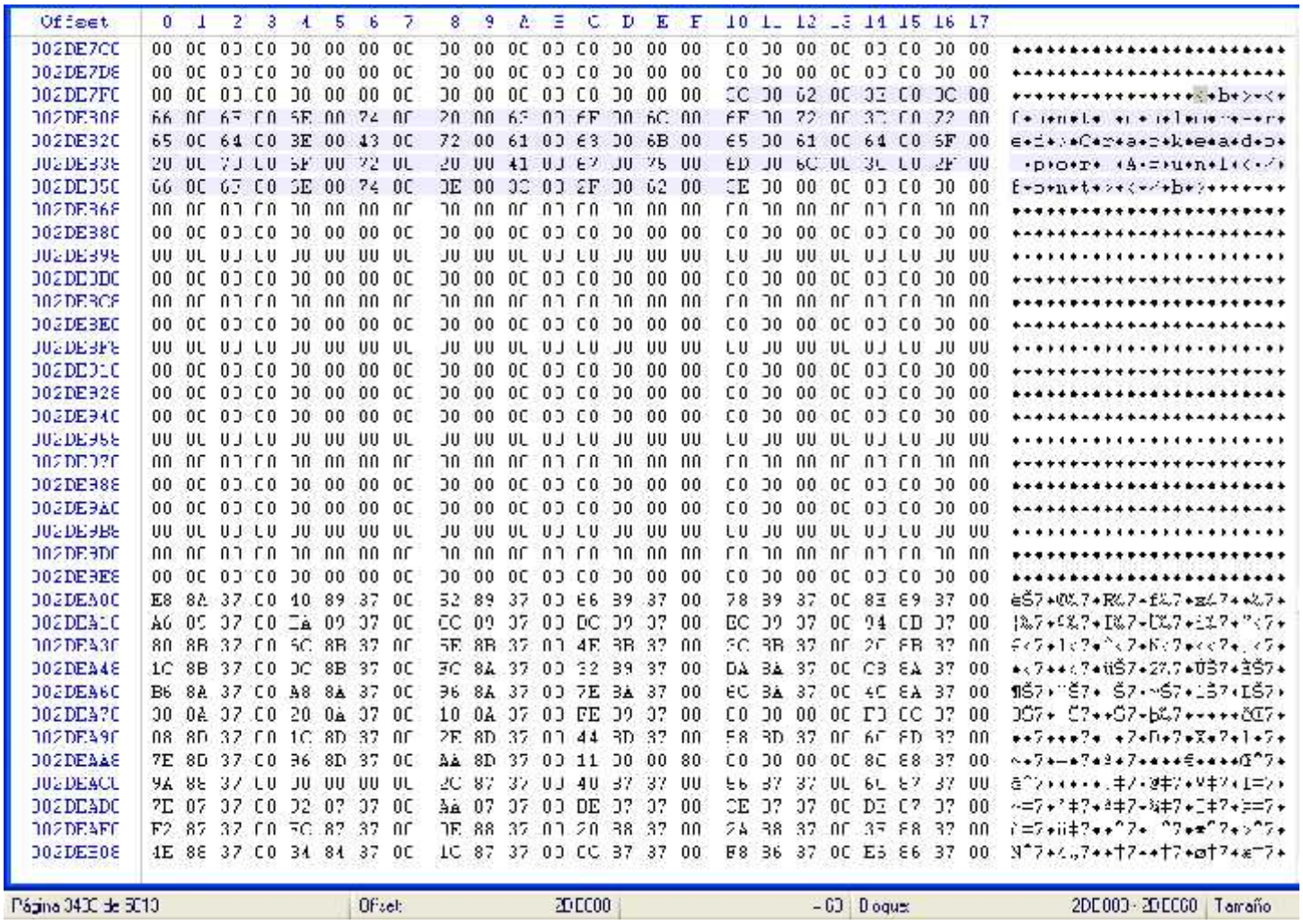
Después de esto miro al final de la sección .text a ver que espacio tengo abriéndolo con WinHex y no hay nada de espacio así que tocará añadir espacio al final de esta y realinear todas las secciones para así tener un poco de espacio al final de la sección .text para poder añadir mi cadena que usaré para arreglar lo del mensaje de “Registrado a”.
Solo necesité cambiar los valores de ROfset y RSize:

[Section Table]

Name	VOffset	VSize	ROffset	RSize	Flags
.text	00001000	002DF000	00000200	002DE800	60000020
.rdata	002E0000	0009A000	002DEA00	00099600	40000040
.data	0037A000	00012000	00378000	0000C200	C0000040
.rsrc	0038C000	0009D24C	00384200	0009D250	40000040

Se puede ver como al RSize de .text le he sumado 0x200, ya que es el valor de FileAlignment, y luego simplemente he sumado 0x200 al resto de ROffsets y con eso ya está solucionado el tema de la alineación pero nos queda un binario inservible ya que la alineación indica que la sección .text ocupa 0x200 bytes más de los que realmente ocupa así que habrá que arreglar eso y para ello lo abrí en WinHex y me fui al final de la sección .text y le añadí 0x200 bytes y listo, ya tengo una pequeña zona donde poder escribir mi cadena y el binario vuelve a estar operativo.

Pongo 512 porque el FileAlignment es 0x200 con lo que ese es el valor mínimo y 512 es igual a 0x200.



Si observan empiezo en 0x2DE800 ya que si sumamos el ROffset y el RSize de la sección .text tenemos esto:

```
0x00000200 + 0x002DE600 = 0x002DE800
```

Y a partir de ahí fue donde añadí los 0x200 bytes así que a partir de ahí puedo escribir hasta 0x200 bytes y, como ven, he añadido una cadena que usaré para mi cometido.

Si lo abrimos en Olly y buscamos la cadena que he añadido vemos esto:

```
006DF600  3C 00 62 00 3E 00 3C 00 66 00 6F 00 6E 00 74 00  <.b.>.<.f.o.n.t.
006DF610  20 00 63 00 6F 00 6C 00 6F 00 72 00 3D 00 72 00   .c.o.l.o.r.=.r.
006DF620  65 00 64 00 3E 00 43 00 72 00 61 00 63 00 6B 00  e.d.>.C.r.a.c.k.
006DF630  65 00 61 00 64 00 6F 00 20 00 70 00 6F 00 72 00  e.a.d.o. .p.o.r.
006DF640  20 00 41 00 67 00 75 00 6D 00 6C 00 3C 00 2F 00   .A.g.u.m.l.<./.
006DF650  66 00 6F 00 6E 00 74 00 3E 00 3C 00 2F 00 62 00  f.o.n.t.>.<./.b.
006DF660  3E 00                                            >.
```

A ver si alguien puede explicarme como hacer el cálculo para saber en que dirección virtual empezaría sin tener que hacer una búsqueda ya que es útil saber como hacerlo y no se.

He comprobado que si desinstalo y vuelvo a instalar la aplicación los valores originales pueden variar (me di cuenta porque en este caso el ROffset de .text es 0x200 y otra vez era 0x400) así que toca hacer el cálculo.

Busqué la cadena “Registrado a” en el binario y no aparecía por ningún sitio así que mirando en los directorios vi que había archivos de idiomas y en el ingles encontré esto:

ID5300=Registrado a %s

Si busco la constante 5300 (0x14B4 en hexa) llego aquí:

```
0048E94A  |> \8B45 E4       mov     eax, [local.7]
0048E94D  |.  50            push    eax
0048E94E  |.  68 B4140000   push    14B4
0048E953  |.  8D4D F0       lea     ecx, [local.4]
0048E956  |.  51            push    ecx
0048E957  |.  E8 94110800   call    0050FAF0                        ;  LopeEdit.0050FAF0
```

Y después de tracear un poco mas llegamos aquí:

```
0048EAA8  |> \8D4D F0       lea     ecx, [local.4]
0048EAAB  |.  E8 F0A4F8FF   call    00418FA0                        ;  LopeEdit.00418FA0
0048EAB0  |.  50            push    eax                             ; /Arg2 = 01109DF0
0048EAB1  |.  68 A8050000   push    5A8                             ; |Arg1 = 000005A8
0048EAB6  |.  8B4D C0       mov     ecx, [local.16]                 ; |
0048EAB9  |.  E8 EE0C1B00   call    0063F7AC                        ; \LopeEdit.0063F7AC
```

El push eax lo que hace es meter la cadena que se mostrará que en este caso es “Registrado a” pero si no estuviese registrado sería “Periodo de prueba de LopeEdit Pro: quedan %d días” donde el %d es un valor entero.
Cambio para mostrar Crackeado por Aguml:

```
0048EAA8      B8 00F66D00   mov     eax, 6DF600                     ;  UNICODE "<b><font
color=red>Crackeado por Aguml</font></b>"
0048EAAD      90            nop
0048EAAE      90            nop
0048EAAF      90            nop
0048EAB0  |.  50            push    eax                             ; /Arg2
0048EAB1  |.  68 A8050000   push    5A8                             ; |Arg1 = 000005A8
0048EAB6  |.  8B4D C0       mov     ecx, [local.16]                 ; |
0048EAB9  |.  E8 EE0C1B00   call    0063F7AC                        ; \LopeEdit.0063F7AC
```

Con esto ya no aparece el cartelito de que es de prueba ni el “Registrado a” y nos aparece el mensaje “Crackeado por Aguml” en negrita y en rojo. Cambiando la cadena podemos poner lo que queramos jejeje.

Tambien estuve mirando un poco mas arriba y vi las siguientes cosas:

1. Aquí es donde realmente comprueba si estamos registrados o no para mostrar un mensaje u otro en la ventana Registro (si saltase nos mostraría el mensaje de “Periodo de prueba de LopeEdit Pro” y si no salta vamos a la zona donde muestra el mensaje “Registrado a”). Con las modificaciones que ya hicimos pasamos esta comprobación sin problemas:

```
0048E84B  |.  E8 40DD0700   call    0050C590                        ;  LopeEdit.0050C590
0048E850  |.  85C0          test    eax, eax
0048E852  |.  0F84 24010000 je      0048E97C                        ;  LopeEdit.0048E97C
```

2. Aquí tendremos el valor en hexa de los segundos a esperar para que se active el boton OK cuando es trial (prueben a cambiar el valor del dword adonde apunta la comparación y verán):

```
0048E97F  |.  83BA 98000000>cmp     dword ptr ds:[edx+98], 0         ;
0048E986  |. /0F8E F6000000 jle     0048EA82                         ;  LopeEdit.0048EA82
```

3. Estas son las direcciones para el chico bueno y el chico malo:

ID5249=LopeEdit Pro ha sido registrado satisfactoriamente.

0049020D push 1481 chico bueno

ID5304=La licencia no es válida.

00490165 push 14B8 chico malo

Con las modificaciones que tenemos hechas hasta ahora no solo llega a ejecutarse el chico bueno si no que ni siquiera necesitamos el archivo de licencia para nada.

Para eliminar el “Register...” y el “Buy Now!” usé ResHacker y borré ambos pero luego al mostrar la ventana me daba error y era porque ambas cadenas las escribe en ejecución y al eliminar las etiquetas pues no podía escribir así que busqué las constantes e hice que no escriba nada para evitar el error.

Para Register...:

ID1453=Register...

```
0048DE8C    52              push    edx
0048DE8D    68 AD050000     push    5AD
0048DE92    8B45 08         mov     eax, ss:[ebp+8]
0048DE95    50              push    eax
0048DE96    E8 EE931C00     call    00657289                         ; LopeEdit.00657289
```

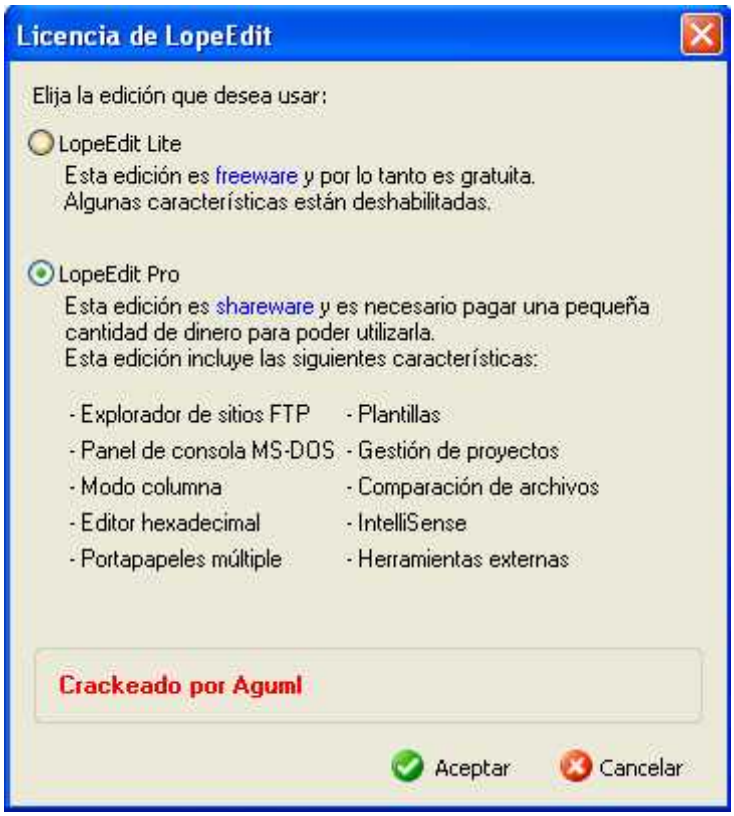
Para Buy Now!:

ID1454=Buy Now!

```
0048DEA4    51              push    ecx
0048DEA5    68 AE050000     push    5AE
0048DEAA    8B55 08         mov     edx, ss:[ebp+8]
0048DEAD    52              push    edx
0048DEAE    E8 D6931C00     call    00657289                         ; LopeEdit.00657289
```

Nopeé ambas zonas y con eso ya sale la ventana Registrar como yo quiero. :-)
También podría haber cambiado las cadenas y haber puesto lo que quisiera al igual que hice con el mensaje de “Registrado a”.

Y este es el resultado:



El programa ya no caduca y es full jejeje.

La verdad es que me dio pena no poder sacar una licencia válida así que si alguien se anima a ver como conseguir un .lic válido…

NOTA: Si quereis hacer pruebas a ver si sacais una licencia válida no se os ocurra crear el archivo .lic con el nombre bueno en el mismo directorio del ejecutable ya que el programa lo primero que hace es borrar el archivo de registro LopeEditPro.lic del directorio donde está el ejecutable y luego intenta hacer una copia del nuestro con ese nombre en el mismo directorio y si ya lo borró antes pues no hará la copia con lo que no podrá luego colocarlo en el directorio `"C:\Documents and Settings\USUARIO\Datos de programa\LopeSoft\LopeEdit\LopeEditPro.lic"`.