

Indice

Introducción	3
Herramientas usadas:	
Comenzando a presentar el programa	
Limitaciones:	3
Comenzamos la depuración BUSCANDO LA LIMITACION	
Limitaciones reales	
Parchando WATERMARK COMO STRING	8
Parchando titulo demo	8
GUARDAR COMOsolo para la primera pagina	
Parchando el certificado	
Agregando la nueva seccion para la extensión de código	11
USANDO MULTILINE ULTIMATE ASSEMBLER	11
USANDO MULTILINE ULTIMATE ASSEMBLER con el depurador	14
Ultima limitación, ejecución de web sin nuestro consentimiento.	

INTRODUCCIÓN

Programa	Spdfedit 5.5
Descarga	http://www.cadkas.com/spdfedit!.exe
Dificultad	Depende de quien lo mire.
Información	http://www.cadkas.com/pdf-editor-spanish.php
Herramientas usadas	X64dbg ,PID
Fecha Creación Tutorial	22/07/2017
Fecha Compartir Tutorial	08/08/2017
Cracker	Apuromafo

"Deja que Hablen de ti. Total, tú sabes quién eres, qué haces, qué hiciste, qué dijiste y ellos no." Anónimo

HERRAMIENTAS USADAS:

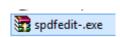
Herramienta	Descarga	Utilidad
Procesador de texto	(está incluido con el suite de office)	Para redactar el tutorial
Sharex	https://getsharex.com/	Para capturar las imágenes
Everything	http://www.voidtools.com/	Para buscar los archivos en el pc
Winrar	https://www.winrar.es/descargas	Compresor/Descompresor de archivos
7zip	http://www.7-zip.org/	Compresor/Descompresor de archivos
X64dbg	http://x64dbg.com/	Depurador
PID	https://web.archive.org/web/20170620171730/http://pid.gcwstorage.xyz/dl.php?f=ProtectionId.685.December.2016.rar	Analizador de Ejecutables
ТОРО	http://ricardonarvaja.info/WEB/OTROS/HERRAMI ENTAS2/Q-R-S-T-U/topo12corregido.rar	Permite Agregar una sección
Section Adder v0.1	https://reversingfiles.info/dls/tools/SecTion.Adde R.O.1-Tool_CiM.zip?Submit=Download	Permite Agregar una sección
CffExplorer	www.ntcore.com/exsuite.php	PE editor para editar a gusto. Permite explorar un ejecutable x86 /x64 sin forma de ver bien los form
Pexplorer	http://www.heaventools.com/download-pe- explorer.htm	Permite explorar un ejecutable x86 , viendo bien los form
Resource hacker	http://www.angusj.com/resourcehacker/	Permite modificar ejecutables sobre todo en la parte de recursos, incluyendo los form
IDR	https://web.archive.org/web/20170501145746/ht tp://kpnc.org/idr32/en/ https://github.com/crypto2011/IDR	Analizador de Delphi (interactive delphi)
Uniextractor*	http://filehippo.com/es/download_univers al_extractor/	Extractor de archivos

Comenzando a presentar el programa

Bienvenidos a esta pequeña lectura e historia explorando un programa demo o capado, pero no significa que no podamos aprender cosas nuevas o extender la funcionalidad, me he dedicado a depurar este programa en breves lapsos y hasta entonces me concentré en hacer parches funcionales intentando entender o comprender lo que estaba haciendo.

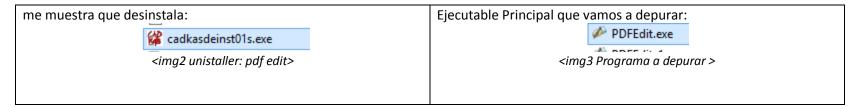
¿Setup.exe?

La primera impresión es ver el setup que proviene de la web (sin análisis, vemos que usa un stub de Winrar)



<img1 setup: Winrar sfx>

Luego es descomprimir con un descompresor, sea Winrar o 7Zip , con ello podemos ver los archivos más llamativos:



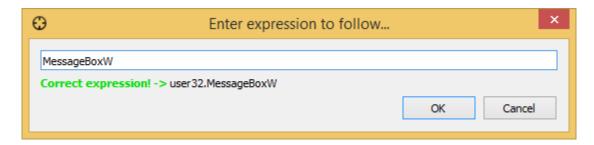
LIMITACIONES:

Si vamos al sitio web refiere de forma muy vaga, que la versión pro puede crear botón en el pdf, además otros tipos de checkbox y opciones (controles), pero al pulsar en el demo los botones si deja, aquí viene la pregunta, entonces ¿ cuales son las limitaciones? Solo puedo afirmar que el programador nos avisará de forma oportuna. Ahora bien, si yo creo un programa versión 5.1 y cambio algo, genero un programa 5.2, pero aquí modifican cada semana y siempre sigue siendo la versión 5.5 asumo que lo cambian por presencia de varios errores (asumo que debe ser por eso), en su web no es clara su limitación, así que solo espero encontrarlos mientras lo usemos, guardo el archivo pdf y aparece el primer mensaje.



COMENZAMOS LA DEPURACIÓN BUSCANDO LA LIMITACION

Así que para poder comenzar, necesitamos un pdf cualquiera con más de una página para poder probar hasta donde llega, hacemos la acción de "guardar archivo como", para luego leer un nag , este mensaje informativo usa MessageBoxA/W pero depurando he visto la letra W, así que comenzaré con este BP(BreakPoint). Comenzamos el programa en el depurador favorito x64dbg (que recordemos que tiene 2 opciones uno para x64 y x86, este programa está en x86 por lo cual depurará con x32dbg) , no está empacado, se logran apreciar algunas String o palabras, pero nada muy decisivo que me diga "DEMO VERISION", nos muestra un mensaje que han añadido una marca de agua, esto será nuestro comienzo y de ahí a retornar API para comenzar, pulsamos para seguir una expresión, en este caso la API (de mensaje)



<img6 : Api MessageBox en x32dbg >

Me importa el retorno (donde termina el mensaje) para saber quien lo ha llamado.

```
8B FF
                                              mov edi
                                                         edi
                                                                                                   MessageBoxW
                                              push ebp
 '4AE7892
                 55
74AE7893
                 8B EC
                                             mov ebp,
                                                         esp
                 6A 00
FF 75 14
FF 75 10
FF 75 0C
FF 75 08
E8 78 FD FF FF
74AE7895
                                                                      ebp + 0x14
ebp + 0x10
ebp + 0xC
74AE7897
                                              push dword ptr
                                              push dword ptr
74AE789A
74AE789D
                                             push dword ptr
74AE78A0
                                              push dword
74AE78A3
                                                    <user3
                                              pop ebp
                 5D
74AE78A9
                 C2
                     10 00
 4AE78AD
                 90
                                              nop
                 90
 4AE78AE
                                                                                  B
                 90
 4AE78AF
                                             nop
 4AE78B0
                 90
```

<img7: retorno Api MessageBox en x32dbg >

al colocar ok (en el mensaje) retorno aquí llegamos (donde fue llamado)

```
84 C0
74 06
81 CE 00 00 10 00
                                              test al, al
je pdfedit.656484
or esi, 0x100000
                                              or esi, 0x100000
xor ecx, ecx
 0065647E
 00656484
                  33 C9
 00656486
                                              push ebp
                  68 OB 65 65 OO
64 FF 31
                                              push pdfedit.65<u>65</u>0B
 00656487
                                              push dword ptr fs:[ecx]
mov dword ptr fs:[ecx],
 00656480
                  64 89 21
 0065648F
 00656492
                                              mov eax, dword ptr ss:[ebp - 0x8]
 00656493
                  8B 45 F8
                                              push eax
push edi
 00656496
                  50
 00656497
                  57
                  E8 8E 07 DC FF
                                              mov dword ptr ss:[ebp - 0xC], eax
0065649E
                  89
                     45 F4
                  33 CO
                                              xor eax, eax
 006564A1
 006564A3
                                              pop edx
 006564A4
                  59
                                              pop ecx
                                             pop ecx
mov dword ptr s:[eax], edx
 006564A5
                  59
                  64 89 10
 006564A6
```

<img8: llamada de MessageBox en x32dbg >

Primer ret va hacia un poco más abajo (ret)

```
0065650A
                                            pdfedit.40ABB4
               E9 A4 46 DB FF
0065650B
00656510
               EΒ
                  9C
                  45 F4
00656512
               88
                                       mov eax, dword ptr ss:[ebp - 0xC]
                                       pop edi
pop esi
00656515
               5 F
00656516
               5E
00656517
                                       pop ebx
00656518
               8B E5
                                       mov esp,
                                                  ebp
               5D
C2 04 00
                                       pop ebp
ret 0x4
0065651A
0065651B
0065651E
               8B C0
                                            eax,
00656520
                                       push ebp
00656521
               8B EC
                                       mov ebp,
```

<img9 : retorno de función llamada en x32dbg >

Y al llegar al ret decisivo salta a donde fue invocado (ret4)



Como se ve un call bastante estructurado (edx,eax ecx), esto es común cuando se usan translate (traducción o multi/idioma) asi que puede que el formato sea similar, hay que mirar mas arriba... para saber quien realmente ha llamado el titulo y mensaje

```
call pdfedit.40C63C
   00A24EC4
                 E8 73 77 9E FF
                                       mov edx, eax
   00A24EC9
                 8B D0
                 A1 FC 83 B7 00
                                       mov eax, dword ptr ds:[0xB783FC]
   00A24ED0
                 8B 00
                                       mov eax, dword ptr ds:[eax]
                                       pop ecx
                                       call pdfedit.
                                                                                 llamada 1
                 E8 C8 14 C3 FF
00A24ED8
                 OF 94 45 DF
                                       sete byte ptr ss:[ebp - 0x21]
                                       xor eax, eax
   00A24EDD
                 33 CO
                                       pop edx
```

<img11 : retorno de función llamada en x32dbg , se transforma en la primera llamada >

Esta Illamada 1, corresponde al mensaje de showmessagebox...pero todo se preparó antes...

Para cualquier persona que mira un programa que ha invocado un messagebox, intenta saber de dónde ha salido el mensaje, pero cuando no hay string, esto solo significa algo, hay que pensar que el autor ha pensado alguna forma más general o recursiva, uso de índices o algún array para usar el mismo lugar para diferentes palabras, veamos si encuentro algo al respecto, Aquí se ven como hay 2 mov eax y algo mas curioso, creo que no hay muchos escritos que hablen de esto.

```
push 0x41
             FF 32 40 01 00 00
A24E90
             6A 41
                       FE F3 FF
                                      lea edx,
                                                dword ptr ss:[ebp - 0xC0184]
             8D 95
                    7C
                                     mov eax, 0x4E
call pdfedit.7A45D4
A24E98
             B8 4E 00 00 00
                                                                                     mov eax, value
            E8 32 F7 D7 FF
8B 85 7C FE F3 FF
E8 8F 77 9E FF
                                                                       0xE0184]
A24EA2
                                      mov eax, dword ptr
                                      call pdf
A24EA8
                                               edit.40C63C
A24EAD
                                      oush eax
A24EAE
             8D 95 78 FE F3 FF
                                      lea edx, dword ptr ss:[ebp - 0xC0188]
                                      mov eax, 0x4D
call pdfedit.7A45D4
                                                                                     4D:'M'
A24EB4
             B8 4D 00 00 00
A24EB9
             E8 16 F7 D7 FF
                                      mov eax, dword ptr
call pdfedit.40C63C
A24EBE
             8B 85 78 FE F3 FF
                                                            ss: [ebp - 0xC0188]
                    77
                      9E FF
                                      mov edx, eax
A24EC9
             8B D0
```

<img12 : índice con eax, para armar MessageBox >

Mov eax, value, esto es sospechoso porque son valores muy pequeños, y con llamadas muy similares, esto solo significa que tiene un índice para diversos mensajes, dependiendo de ellos va creando el mensaje y luego va juntando las opciones

Seamos prácticos, si hay valores dados asi "mov eax, constante", podemos encontrar todo lo que pasa en el programa y no tendremos que hacer gran exploración, pero bueno veamos que me enfrento, pero como tenemos tiempo, seguiremos depurando

Esta pequeña tabla posee valores entre 1 Hex y c4 (hex), osea hay mucho que está adentro..veamos en decimal es mas fácil de entender (entre 1 y 196 mensajes)

```
007A45D4
                   53
                                             push ebx
                   56
   007A45D5
                                             push esi
                   8B F2
                                             mov esi, edx
                                            mov ebx, eax
   007A45D8
                   8B D8
                                            cmp ebx,
                   83 FB 01
7C 16
   007A45DA
                                                       0x1
   007A45DD
                                                pdfedit.7A45F5
   007A45DF
                   81 FB C4 00 00 00
                                             cmp ebx, 0xC4
-0
   007A45E5
                   7F 0E
                                               pdfedit.7A45F5
                                            mov edx, esi
mov eax, dword ptry
call pdfedit.7A37F
                   8B D6
   007A45E7
                                                                      :[ebx * 4 + 0xB60F0C]
                   8B 04 9D 0C 0F B6 00
    007A45F0
                   E8 03 F2 FF FF
   007A45F5
                                             pop esi
   007A45F6
•
                   5 B
                                             pop ebx
    007A45F7
                   C3
                                             ret
    007A45F8
                                             push ebp
    007A45E9
                   SR FC
```

<img13 : Rutina Importante para mensajes >

Colocado el bp reinicio el programa, ahora vere sus mensajes, el primero refiere optimizing, (sigo ejecutando el programa con f9), no es relevante

```
lea eax, dword ptr ds:[eax]
push ebx
07A45D1
                    8D 40 00
   A45D4
                    53
56
  A45D5
                                                         bush esi
                   8B F2
8B D8
                                                        mov ebx, eax
cmp ebx, 0x1
jl pdfedit.7A45F5
cmp ebx, 0xC4
jg pdfedit.7A45F5
  A45D8
                   88 D8
83 FB O1
7C 16
81 FB C4 00 00 00
7F OE
8B D6
   A45D/
   A45DD
  A45DF
                                                        mov edx, esi
mov eax, dword ptr ds:[ebx * 4 + OxB60F0C]
call pdfedit.7A37F8
                   8B 04 9D 0C 0F B6 00
E8 03 F2 FF FF
                                                                                                                                                                          [ebx*4+B60F0C]:L"Optimize to:"
                    5E
                                                               ebx
   A45 F7
                    C3
                                                        push ebp
mov ebp, esp
                   55
8B EC
   445F9
                   6A 00
```

<img14 : Verificando mensajes en x32dbg >

Es posible apreciar cadenas (mientras tanto pulso f9) que indican palabras alusivas a acciones que realizará el programa, en el primer inicio aparece la de la nag, "Image Artist"

```
eax=55 'U'
dword ptr[ebx*4+B60F0C]=[00B61060]=007A0C0C L"PDF Editor: ImageArtist"
.text:007A45E9 pdfedit.exe:$3A45E9 #3A39E9
```

<img15: Nag: ImageArtist desde x32dbg>

Luego refiere un "of", luego proviene uno mas interesante "demo", para saber entonces donde estamos, debemos llegar al "ret", vemos lo siguiente

7A45f0 se puede apreciar la string "DEMO VERSION"

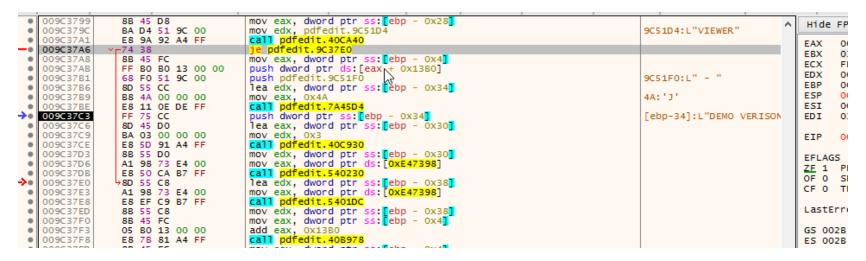
```
7A45f0 se puede apreciar la string "D

push esi
mov esi, edx
mov ebx, eax
cmp ebx, 0x1
jl pdfedit.7A45F5
cmp ebx, 0xC4
jd pdfedit.7A45F5
mov edx, esi
) mov eax, dword ptr ds: [ebx * 4 + 0xB60FoC]

call pdfedit.7A37F8
pop esi
pop ebx
ret
push ebp
mov ebp, esp
push 0x0
                                                    56
8B F2
8B D8
                               445D6
445D8
                               A45DA
                                                    83 FB 01
                                                                                                                                                                                                                                                                                                                                          EIP
                              A45DD
                                                    81 FB C4 00 00 00
7F 0E
8B D6
                                                                                                                                                                                                                                                                                                                                         EFLAG:
                                                                                                                                                                                                                                                                                                                                         OF 0
                                                    8B 04 9D 0C 0F B6 0
E8 03 F2 FF FF
                                                                                                                                                                                                                                                     [ebx*4+B60F0C]:L"DEMO VERSION"
                                                                                                                                                                                                                                                                                                                                        CF 1
                                                    5E
5B
                                                                                                                                                                                                                                                                                                                                         LastE
                                                    C3
55
8B EC
6A 00
6A 00
                                                                                                                                                                                                                                                                                                                                         GS 00:
                                                                                                                                                                                                                                                                                                                                        CS 003
                             7A45FD
                             7A45FF
                              A4601
                                                     6A 00
                                                                                                    push 0x0
                                                                                                                                                                                                                                                                                                                                         x87r1
                                                    6A 00
53
56
                                                                                                    push 0x0
push ebx
push esi
push edi
                              A460
                                                                                                                                                                                                                                                                                                                                         x87r2
x87r3
                                                                                                                                                                                                                                                                                                                                         x87r4
                                                    57
89 4D F4
89 55 F8
8B F8
                                                                                                                                                                                                                                                                                                                                         x87r5
x87r6
                                                                                                   mov dword ptr ss: [ebp - 0xC], ecx
mov dword ptr ss: [ebp - 0x8], edx
mov edi, eax
xor eax, eax
                         007A4608
                         007A460B
                             7A460E
                                                                                                                                                                                                                                                                                                                                          x87r7
                              A4610
                                                          CO
                                                                                                                                                                                                                                                                                                                                        Default (
                                                                                                                                                                                                                                                                                                                                        1: [es
2: [es
bx*4+B60F0C]=[00B61034]=007A05FC L"DEMO VERSION"
5E9 pdfedit.exe:$3A45E9 #3A39E9
```

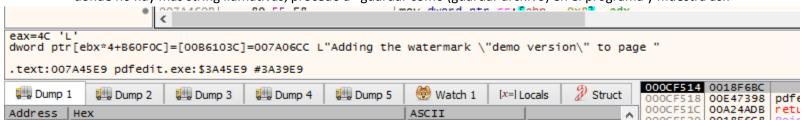
<img16 : Demo versión, posible limitación >

Sobre este demo, llego al ret, al volver vemos la siguiente rutina, la cual claramente tiene un condicional fácil de modificar, debemos cambiar ese salto de je a jmp para que no tengamos en el titulo del programa esta marca.



<imq17 : Demo versión, Título del programa >

cuando abrimos el programa agregamos el pdf, podemos ver como nos va diciendo cuantas paginas tiene, verifica paginas borradas y más, llega el minuto final donde no hay mas string llamativas, procedo a guardar como (guardar archivo) en el programa y muestra así:



<img18: Segunda Limitación, Watermark >

Si traduzco dice agregando marca de agua "demo version a la pagina", ósea nos dice claramente que nos está agregando el watermark, wow, que expresivo, retorno para ver de dónde viene:

```
0A24A87
0A24A8D
                                                         mov eax, dword ptr ds:[esi]
cmp byte ptr ds:[eax + 0x139D], 0x0
ine <pdfedit.mm2>
10A24A94
10A24A96
                     8B 06
                         B8 9D 13 00 00 00
                     80
                          85 3A 04 00 00
                                                         mov eax, dword ptr ds:[esi]
mov eax, dword ptr ds:[eax + 0x122C]
mov eax, dword ptr ds:[eax + 0x10]
                     8B 06
                     8B 80 2C 12 00 00
8B 40 10
0A24AA5
0A24AAB
0A24AAE
0A24AB0
                     33 D2
52
                                                                        edx
                                                         xor edx
                                                         push edx
10A24AB1
10A24AB2
                                                         push eax
imul eax, dword ptr ss:[ebp - 0x2C], 0x64
                     6B 45 D4 64
                                                        imul tax, chicked cdq cdq call pdfedit.40D68C mov edx, eax mov eax, dword ptr ds:[esi] mov eax, dword ptr ds:[eax + 0x95C] call pdfedit.59D160 lea edx, dword ptr ss:[ebp - 0xC0174] mov eax, 0x4C
10A24AB6
                     99
                     E8 D0 8B 9E FF
IOA24ABC
                     8B D0
0A24ABE
0A24AC0
                     8B 80 5C 09 00 00
10A24AC6
10A24ACB
                     E8 95 86 B7 FF
8D 95 8C FE F3 FF
104244D1
                     B8 4C 00 00 00
E8 F9 FA D7 FF
                                                         mov eax, 0x4C
call pdfedit.7
                                                         lea eax, dword ptr ss:[ebp - 0xC01,4]
OA24ADB
                     8D 85 8C FE F3 FF
                                                                                                                                                                          saliendo del adding watermark
0A24AE1
                                                          push eax
                                                         lea edx, dword ptr ss:[ebp - 0xC0178]
mov_eax, dword ptr ss:[ebp - 0x2C]
                     8D 95 88 FE F3 FF
10A24AE2
                     8B 45 D4
E8 F8 13 A0 FF
                                                         mov eax, dword ptr ss:[ebp - 0x2C]

call pdfedit.425EE8

mov edx, dword ptr ss:[ebp - 0xC0178]
0A24AE8
OA24AEB
                     8B 95 88 FE F3 FF
                                                         mov edx,
pop eax

call pdfedit.40C850

mov edx, dword ptr ss: [ebp - 0xC0174]

mov eax, dword ptr ds: [ebx]

mov eax, dword ptr ds: [eax + 0xC30]
0A24AF6
0A24AF7
                     E8 54 7D 9E FF
8B 95 8C FE F3 FF
IOA24AFC
0A24B02
                     8B 80 30 0C 00 00
                                                         mov eax, dword

call pdfedit.5

fild dword ptr
0A24B04
0A24B0A
0A24B0F
                     E8 21 B7 B1 FF
DB 45 D4
                                                                             t.540230
ptr ss:[ebp - 0x2C]
0A24B12
                     83 C4 F4
                                                         add esp, OxFFFFFFF4
```

<imq19: saliendo de la generación de palabra add Watermark >

Por lo que al mirar mas arriba vemos entre ensayo error, para que sirve, si asumo que son saltos chico bueno y malo como en el anterior, el programa no procesa, así que realmente estos saltos son para saber si procesa o no la página en esa porción de código, por lo cual no hay nada que me permita pensar que realmente no me creara el watermark con la presencia de algún archivo o licencia pues realmente lo hará si o si, asi que comenzamos a parchar este watermark, vemos hacia un pequeño scroll (bajando):

```
E8 46 70 00 00
8B 06
66 80 9E 13 00 00 01
FF 45 D4
4F

call pdfedit.A2BC7C
mov eax, dword ptr ds:[esi]
mov byte ptr ds:[eax + 0x139E], 0x1
inc dword ptr ss:[ebp - 0x2C]
dec edi
 00A24C36
00A24C38
00A24C3F
                                                                  inc dword pt.

dec edi
jne pdfedit.A24A94
mov eax, dword ptr ds:[0x878734]
mov eax, dword ptr ds:[eax]
mov eax, dword ptr ds:[eax + 0x3E8]
mov edx, pdfedit.A252C0

Afedit.540230
 00A24C42
00A24C43
00A24C49
                          OF 85 4B FE FF FF
A1 34 87 B7 00
 00A24C4E
                           8B 00
                         80 E8 03 00 00
                                                                                                                                                                         A252CO:L"This text only appears in the demo vers
00A24C56
 00A24C62
00A24C69
00A24C6B
00A24C72
 00A24C74
00A24C7B
00A24C7D
00A24C83
00A24C86
00A24C88
 00A24C8E
```

<img20: Watermark este texto...>

En traducción dice este texto solo aparece en la versión demo del programa.., pero no veo donde lo ha escrito, esto significa otra cosa, hay además algún encode/decode si subo mucho mas arriba encuentro al responsable tiene el mismo largo ©



<img21: Watermark cambiado con la versión..>

Ahí está todo cifrado y más encima el descifrado arriba, no hay misterio, ahora También se aprecia un www

```
00A24A0F
00A24A13
00A24A14
                                                                                                                                                                                                                                                              lea eax, dword ptr ds:[eax + edx * 2 - 0x2]
push eax
                                                                                                                                                   8D 44 50 FE
                                                                                                                                                8B 45 F4
8B 55 D8
0F B7 44 50 FE
                                                                                                                                                                                                                                                              mov eax, dword ptr ss: [ebp - 0xC]
mov edx, dword ptr ss: [ebp - 0x28]
movzx eax, word ptr ds: [eax + edx * 2 - 0x2]
                                                                                00A24A17
00A24A1A
                                                                                                                                                                                                                                                             pop edx
mov word ptr ds:[edx +
sub eax, 0xA
pop edx
mov word ptr ds:[edx], ax
inc dword ptr ss:[ebp - 0x28]
dec edi
jne pdfedit.A24A04
push dword ptr ss:[ebp - 0x10]
                                                                                00A24A1F
00A24A22
00A24A23
                                                                                                                                                  83 E8 OA
                                                                                                                                                 66 89 02
FF 45 D8
                                                                                00A24A26
00A24A29
00A24A2A
                                                                                                                                                   4F
75 D8
                                                                                                                                             dec edi
jne pdfedit.A24A04
FF 75 F0

68 A8 52 A2 00
FF 75 F4
BA 03 00 00 00
E8 E9 7E 9E FF
A1 34 87 B7 00
B8 80 E8 03 00 00
E8 D1 B7 B1 FF
BA 06
C6 80 F4 14 00 00 01
B8 06
C6 80 F4 14 00 00 01
B8 06
C6 80 F4 14 00 00 01
B8 06
C6 80 F4 14 00 00 01
B8 06
C6 80 F4 14 00 00 01
B8 06
C6 80 F4 14 00 00 01
B8 06
C6 80 F4 14 00 00 01
B8 06
C6 80 F4 14 00 00 01
BB 06
C6 80 F4 14 00 00 01
BB 06
C6 80 F4 14 00 00 01
BB 06
C6 80 F4 14 00 00 01
BB 06
C6 80 F4 14 00 00 01
BB 06
C6 80 F4 14 00 00 01
BB 06
C6 80 F4 14 00 00 01
BB 06
C6 80 F4 14 00 00 01
BB 06
C6 80 F4 14 00 00 01
BB 06
C6 80 F4 14 00 00 01
BB 06
C7 F5 F7
C8 BB 01 00 00
C8 B01 C8 D1 00 00
C9 C8 C8 C8 C8 C8 C8 C8
C8 C8 C8 C8 C8 C8
C8 C8 C8 C8 C8
C8 C8 C8 C8 C8
C8 C8 C8 C8 C8
C8 C8 C8 C8
C8 C8 C8 C8
C8 C8 C8 C8
C8 C8 C8 C8
C8 C8 C8 C8
C8 C8 C8 C8
C8 C8 C8 C8
C8 C8 C8 C8
C8 C8 C8 C8
C8 C8 C8
C8 C8 C8
C8 C8 C8
C8 C8 C8
C8 C8 C8
C8 C8 C8
C8 C8 C8
C
                                                                              00A24A2C
00A24A2F
00A24A34
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       A252A8: L"www
                                                                                00A24A37
00A24A3D
00A24A42
                                                                                00A24A47
00A24A4D
00A24A52
00A24A54
00A24A5A
00A24A5F
                                                                                00A24A61
00A24A68
00A24A6A
00A24A71
00A24A73
00A24A7A
                                                                                00A24A7C
00A24A82
00A24A85
                                                                                00A24A87
00A24A8D
                                                                                00A24A94
ptr[ebp-10]=[0018F680]=0684F064 L"Cambiado con la DEMO VERSION de CAD-KAS PDF-Editor (http://"
OOA24A2C pdfedit.exe:$624A2C #623E2C <demo?>
mp 1 Dump 2 Dump 3 Dump 4 Dump 5 Match 1 Ixell ceals 9 Struct 000CF520 0018F6C8 Pointer to SEH.R
```

<img22: Watermark www...>

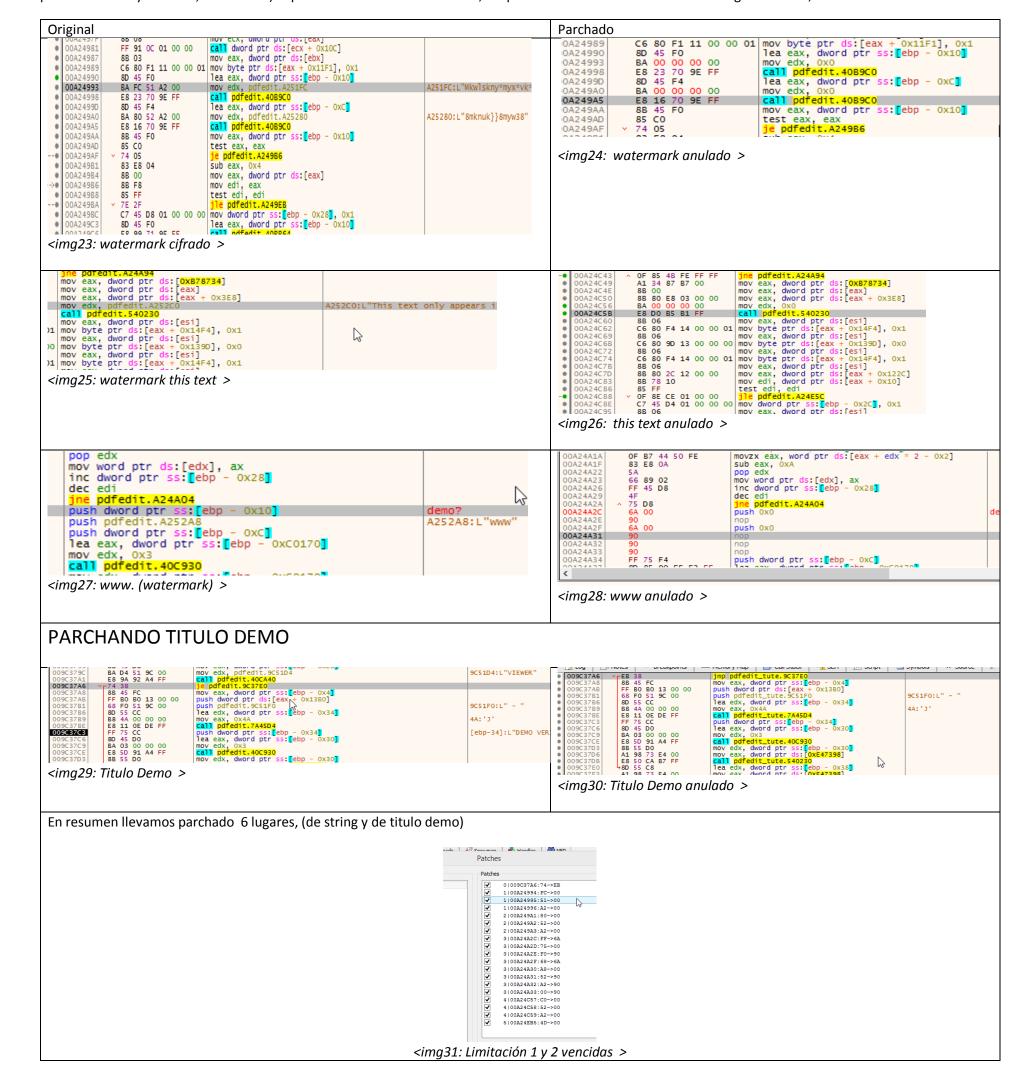
LIMITACIONES REALES

Es bueno hacer una lista y que haremos:

- 1) Titulo con mensaje demo (se cambia de je a jmp)
- 2) Watermark(se cambia anulando la string ,como lo haré ahí veremos)
- 3) Limitación de páginas (se extiende mediante Multiline Ultimate Assembler en x32dbg) y topo o alguna tool similar.
- 4) Certificado Digital (se anula con cffexplorer o pexplorer)
- 5) Menús refiriendo solo pro(se edita con resource hacker)
- 6) Funciones capadas(se edita con resource hacker para anularlas)
- 7) Abre la pagina web del autor al cerrar el programa

PARCHANDO WATERMARK COMO STRING

Creo que en resumen eso tenemos, asi que debemos comenzar a anular de la mejor forma Si anulo el salto, no habrá donde guardar(pero si anulo el string, no puede escribir bytes nulos, buena idea) el parche será colocar el valor a 0, Respecto mas arriba el cifrado anulamos igual con edx, 0



GUARDAR COMO...SOLO PARA LA PRIMERA PAGINA

La estrategia para encontrar de donde venia el mensaje me permite esta vez saber donde ha sido llamado, encontramos la llamada aquí:

```
call pdfedit.9C2660
                                     E8 9D D8 F4 FF
     00A74DCA
     OOA74DCE
.
                                     6A 40
                                                                                       lea edx, dword ptr ss:[ebp - 0x11C]
mov eax, 0x4A
call pdfedit.7A45D4
mov eax, dword ptr ss:[ebp - 0x11C]
call pdfedit.40C63C
push eax
                                                                                          oush 0x40
                                     8D 95 E4 FE FF FF
     00A74DD1
                                    B8 4A 00 00 00
E8 F3 F7 D2 FF
     00A74DD7
                                                                                                                                                                                                                   4A: 'J'
     00A74DDC
                                    8B 85 E4 FE FF FF
E8 50 78 99 FF
     00A74DE1
00A74DE7
     00A74DEC
00A74DED
                                                                                       push eax
lea edx, dword ptr ss:[ebp - 0x120]
mov eax, 0x71
call pdfedit.7A45D4
mov eax, dword ptr ss:[ebp - 0x120]
call pdfedit.40C63C
mov edx, eax
mov eax, dword ptr ds:[0x8783FC]
mov eax, dword ptr ds:[eax]
                                    8D 95 E0 FE FF FF

88 71 00 00 00

E8 D7 F7 D2 FF

8B 85 E0 FE FF FF

E8 34 78 99 FF

8B D0
     00A74DF3
                                                                                                                                                                                                                    71: 'q'
     00A74DF8
00A74DFD
     00A74E03
00A74E08
     00A74E0A
00A74E0F
00A74E11
                                     A1 FC 83 B7 00
                                    8B 00
59
                                                                                        mov eax, dword ptr ds:[eax]
                                                                                       mov eax, dword ptr ds:[eax]
pop ecx

call pdfedit.6563Adv

mov eax, dword ptr ss:[ebp - 0x1c]
mov eax, dword ptr ds:[eax + 0x94c]
mov dl, 0x1
mov ecx, dword ptr ds:[eax]
call dword ptr ds:[ecx + 0x88]
xor eax, eax
pop edx
                                     E8 89 15 BE FF
     00A74E12
                                                                                                                                                                                                                   llamada de txt
     00A74E17
00A74E1A
00A74E20
                                     8B 45 E4
                                     8B 80 4C 09 00 00
                                     B2 01
     00A74E22
00A74E24
                                    8B 08
FF 91 88 00 00 00
33 C0
     00A74E2A
00A74E2C
                                                                                        pop edx
pop ecx
                                    5 A
5 9
      00A74E2D
                                                                                       pop ecx
pop ecx
mov dword ptr :[eax], edx
push pdfedit.A74E7F
lea eax, dword ptr ss:[ebp - 0x120]
     00A74E2E
00A74E2F
                                    64 89 10
68 7F 4E A7 00
8D 85 E0 FE FF FF
     00A74E32
00A74E37
```

<img32: Buscando la limitación 3 >

Si vemos la función que nos da la limitación también usa indice y la otra que tenemos pendiente de su llamada a messagebox

```
00A74DB3
00A74DB6
00A74DBC
00A74DC1
                                                                                     mov eax, dword ptr ss:[ebp - 0x1C]
mov eax, dword ptr ds:[eax + 0x3D0]
call <pdfedit.sub_594658>
dec eax
                               8B 45 E4
8B 80 D0 03 00 00
                                E8 97 F8 B1 FF
                                48
                                                                                             pdfedit.A74DCF
00A74DC2
                                                                                    mov edx, dword ptr ss: ebp - 0x1C mov eax, dword ptr ss: ebp - 0x1C call spdfedit.sub_9C266C>
                                8B 55 E4
00A74DC
00A74DC
                               E8 9D D8 F4 FF
00A74DC
                                                                                    push 0x40
lea edx, dword ptr ss:[ebp - 0x11C]
mov eax, 0x4A
call <pdfedit.sub_7A45D4>
mov eax, dword ptr ss:[ebp - 0x11C]
call <pdfedit.sub_40C63C>
push eax
lea edx, dword ptr ss:[ebp - 0x120]
mov eax, 0x71
call <pdfedit.sub_7A45D4>
mov eax, dword ptr ss:[ebp - 0x120]
mov eax, dword ptr ss:[ebp - 0x120]
mov eax, dword ptr ss:[ebp - 0x120]
                                      95 E4 FE FF FF
00A74DD7
                               B8 4A 00 00 00
                               E8 F3 F7 D2 FF
8B 85 E4 FE FF FF
00A74DE1
                                                                                                                                                                                                                     [ebp-11C]:L"DEMO VERISON"
00A74DE7
                               E8 50 78 99 FF
                               8D 95 E0 FE FF FF
B8 71 00 00 00
E8 D7 F7 D2 FF
8B 85 E0 FE FF FF
E8 34 78 99 FF
00A74DED
00A74DF3
                                                                                                                                                                                                                     71: 'q'
00A74DF8
                                                                                     mov eax, dword ptr ss:[ebp
call <pdfedit.sub_40C63C>
DOA74E03
```

<img33: Buscando la limitación 3 >

Pillamos denuevo el demo versión

```
48
75 OB
                                                                                                                                                                               dec eax
                  00A74DC2
                                                                                                                                                                                 jne pdfedit.A74DCF
                                                                                                                                                                             mov edx, dword ptr ss:[ebp - 0x1C]
mov eax, dword ptr ss:[ebp - 0x1C]
call 
call 
cpdfedit.sub_9C266C>
push 0x40
                  00A74DC
                                                                              8B 55 E4
                 00A74DC7
                                                                             8B 45 E4
                  00A74DCA
                                                                             E8 9D D8 F4 FF
                                                                             8D 95 E4 FE FF FF
B8 4A 00 00 00
                 00A74DD1
00A74DD7
                                                                                                                                                                               lea edx, dword ptr ss:[ebp - 0x11C]
                                                                                                                                                                              mov eax, 0x4A

call <pd style="text-align: center;">call call call <pd style="text-align: center;">call call ca
                                                                             E8 F3 F7 D2 FF
                                                                                                                                                                                                                                                                                                                                                                                                                        demo
                 00A74DDC
                                                                             8B 85 E4 FE FF FF
E8 50 78 99 FF
                 00A74DE1
00A74DE7
                 00A74DED
00A74DF3
                                                                             8D 95 E0 FE FF FF
B8 71 00 00 00
                                                                                                                                                                               lea edx, dword ptr ss:[ebp - 0x120]
                                                                                                                                                                             mov eax, 0x71

call 
call <p
                                                                                                                                                                                                                                                                                                                                                                                                                         71: 'q'
                                                                             E8 D7 F7 D2 FF
                                                                             8B 85 E0 FE FF FF
E8 34 78 99 FF
→ 00A74DFD
                                                                                                                                                                                                                                                                                                                                                                                                                        you can only
                                                                             A1 FC 83 B7 00
8B 00
                                                                                                                                                                              mov eax, dword ptr ds:[OxB783FC]
mov eax, dword ptr ds:[eax]
                  00A74E0A
                  00A74E0F
                                                                                                                                                                               pop ecx
call pdfedit.6563A0
                   00A74E11
                 00A74E12
                                                                             E8 89 15 BE FF
                                                                                                                                                                             mov eax, dword ptr ss:[ebp - 0x1C]
mov eax, dword ptr ds:[eax + 0x94C]
mov dl, 0x1
                                                                                                                                                                                                                                                                                                                                                                                                                         llamada de txt
                  00A74E17
                                                                             8B 45 E4
                   00A74E1A
                                                                             8B 80 4C 09 00 00
                  00A74E20
                                                                             B2 01
                  00A74E22
                                                                                                                                                                              mov_ecx, dword ptr ds:[eax]
                                                                             8B 08
```

<img34: Buscando la limitación 3 >

El tema es aprovechar bien el espacio que tenemos, porque si alcanza no necesitamos una nueva sección, pero si crearemos una nueva sección tendrá un espacio suficiente para crear lo que podamos imaginar, solo haremos jmp (codecave o nuevo espacio) o push label +ret y ahí ejecutaremos las instrucciones faltantes... demo versión hace alusión que hay un mensajito o messagebox que me dice que es demo y solo procesará la primera página. ahora vamos a lo importante luego se dedica a procesar el archivo abierto y mueve valores a través de ebp, eax, hasta aquí cualquiera que hubiera depurado archivos de abrir, guardar, verá que es algo similar siempre.

```
00A74CD4
00A74CD5
                                                                                                             83 FB FF
                                                                                                                                                                                                                                                                                              cmp ebx, Oxffffffff
jne pdfedit.A74C72
                                                                                                               75
                                                                                                                                      98
                                                                                                        75 98
88 45 E4
88 80 30 12 00 00
88 45 E4
88 80 30 12 00 00
88 45 E4
88 80 30 12 00 00
89 45 E4
89 80 30 12 00 00
89 80 00
89 80 00
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
89 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90
80 90

                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            importante
    00A74CDD
    OOA74CEA
   OOA74CED
00A74CF7
   00A74D05
                                                                                                            89 50 48 mov dword ptr ds:[eax + 0x48], edx

FF 85 5C FF FF FF inc dword ptr ss:[ebp - 0xA4]

83 BD 5C FF FF FF 02 cmp dword ptr ss:[ebp - 0xA4], 0x2
  00A74D12
                                                                                                           OF 85 E3 EC FF FF
8B 45 E4
    00A74D19
00A74D1F
                                                                                                                                                                                                                                                                                              mov eax, dword ptr ss:[ebp - 0x1C]
mov eax, dword ptr ds:[eax + 0x95C]
                                                                                                            8B 80 5C 09 00 00
```

La porción de código dice que solo permite la primera página, el programa si maneja cuanto es el total, pero compara con una constante, los números mas comunes a buscar (si es solo la primera página...), o es el 1 o es el 2 o al revez 1,0 y menos 1,

en este caso busca en la porción (A74D09 a 00A74D12) el 2 (dado que mueve el dword a edx y luego lo incrementa 1 vez) ,dado que la pagina 0 no es ninguna al estar lista la primera, incrementa con 1 y tiene en hex el primer valor, si el valor es igual (jne) entonces no salta...y termina la nag..

si el valor salta continua, pero donde salta, nada más ni nada menos hay una llamada a punteros del total de páginas (en esta parte donde salta, es tomado el final de las paginas.(segunda parte importante para ver)

```
00A73A02 >
                       12 00 00
                                    mov eax, dword ptr
              8B 80
                                                            eax
00A73A0B
             8B 40 10
                                    mov eax, dword ptr ds:[eax
00A73A0E
             33 D2
                                    xor edx,
00A73A10
             52
                                    push edx
00A73A11
                                    push eax
```

<img36: Manejo del total de paginas >

En resumen estas 3 instrucciones me dan información adecuada

```
el total de paginas del archivo

00A73A02 mov eax, dword ptr ss: ebp - 0x1C 00A73A05 mov eax, dword ptr ds:[eax + 0x122C] 00A73A0B mov eax, dword ptr ds:[eax + 0x10] 00A74D05 mov eax, dword ptr ds:[eax + 0x1230] 00A74D05 mov eax, dword ptr ds:[eax + 0x10]
```

como comparo las 2, si no hay más espacio...pues generando un espacio nuevo, aquí debemos entrar en algo no tan usado, llamado inline, debemos luego que llame a la página actual, que compara con las 3 instrucciones (que llama al total de páginas) y luego continuar el flujo a debajo de las 3 instrucciones, me es difícil de explicar pero lo intentaré hay 2 tipos de secuencias que llamare instrucción a y b , sea el a el que procesa la página actual y b el que me dice el total de páginas, debo comparar a con b para que sea un valor real y no sea el 2 que me ofrece entonces debo ordenar esta información, si llego y modifico el flujo sin mantenerlo crasheará, debo hacer que la instrucción a, siga tal cual, luego agregar una comparación con una instrucción b que guardaré en otro lugar, luego que compare con un número relativamente grande para que pueda seguir procesando, el nuevo código en cuestión será:

```
//inline patch que indica que tenemos el flujo número total
mov eax, dword ptr [ebp - 0x1c]
mov eax, dword ptr [eax + 0x122c]
mov eax, dword ptr [eax + 0x10]

//y usamos la misma constante de la 1 para comparar con el valor total
cmp dword ptr [ebp - 0xa4], eax

//si son distintos que siga procesando (pasara al loop normal), pero si son igual al final del documento entonces saltara al a73a02 donde
es que termina
jne 0xa73a02
```

Hasta aquí es entendible la idea, pero pensemos si salta se desajustará stack, entonces debemos ordenar en pushad/popad las nuevas instrucciones

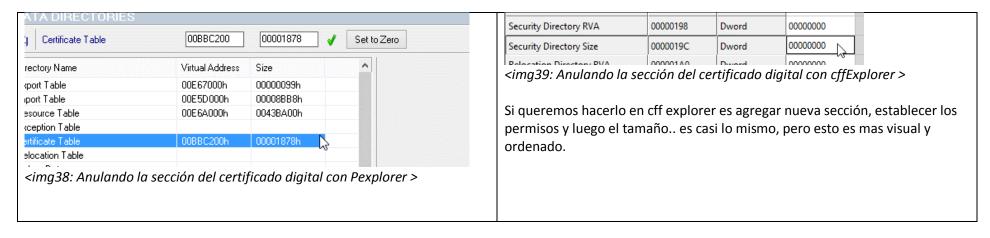
La comparación con el 2, ahora será modificada a un valor mucho mayor..., y la comparación dentro del pushad/popad servirá para salir del bucle (del número mayor) Para este inline haremos uso del MultimateAssembler en x64dbg debemos definir algunas cosas eso si, dirección y las instrucciones, tenemos esto:

```
<00A74D0C> //dirección donde esta la comparación del texto
inc dword ptr [ebp - 0xa4] //aquí le dice que termino la pagina 1 pushad //pushad guarda registros
mov eax, dword ptr [ebp - 0x1c] // instrucción 1 mov eax, dword ptr [eax + 0x122c] // instrucción 2 mov eax, dword ptr [eax + 0x10] // instrucción 3 cmp dword ptr [ebp - 0xa4], eax //comparación que realmente necesito je @L00000001 //si son iguales devuelve el flujo de lo que continua popad //devuelve el valor antes del injerto
@L00000008:
cmp dword ptr [ebp - 0xa4],99999 //comparación con 99999 hex paginas jne 0xa73a02 //instrucción original que comparaba pagina actual versus numero gigante (antes decía 2)
@L000000001:
mov eax, dword ptr [ebp - 0x1c] //instrucción que continua el flujo
```

Comenzando a parchar dado que ya tenemos las nociones básicas del programa y con esto podemos comenzar bien

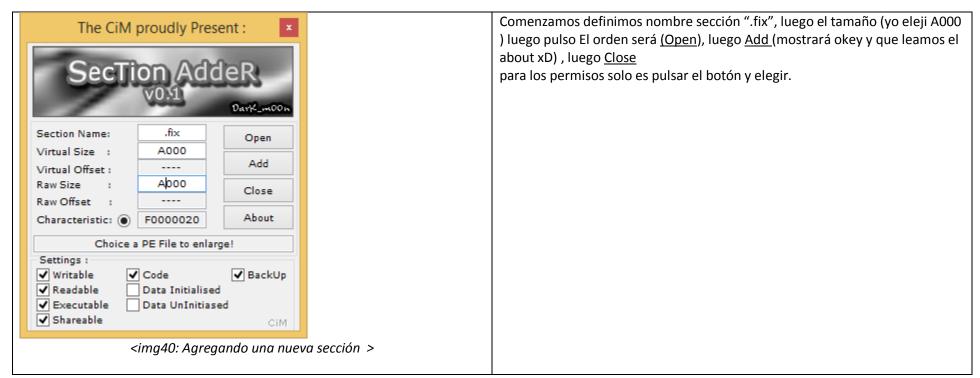
PARCHANDO EL CERTIFICADO.

Primer paso, debemos usar cff explorer para anular la sección de firma digital, recordemos que el programa posee una firma digital. Si usamos pexplorer basta colocar set to zero,



AGREGANDO LA NUEVA SECCION PARA LA EXTENSIÓN DE CÓDIGO

Agregamos la sección con Section Adder,



El único principio que tenemos base es que podemos definir label, comentarios sin mayor problema (leer el faq del Multiline Ultimate Assembler)

Ahora al ir a la sección nueva, nos llevamos esta sorpresa que está escrito



<img41: Viendo la nueva sección >

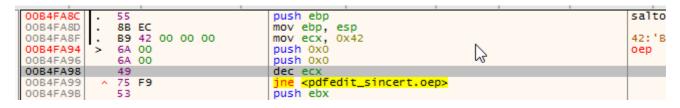
Está escrito, jaja, bueno no es mala noticia, esto es el certificado digital (recuerden que luego del ejecutable, es ingresado el certificado digital sin asignar ese espacio, una vez que creamos una nueva sección con un espacio este está incluido ahí, lo hacemos fill con 00 (hex 00), solo eliminamos que dijiera que la sección no está, pero su contenido aun está ahí.



USANDO MULTILINE ULTIMATE ASSEMBLER PARA EJERCITAR

Luego comenzamos a crear el código de inline, esta vez desde el ejecutable original solo que hemos quitado el certificado.

Antes asi es el entrypoint, debemos ensayar antes de cambiar algo



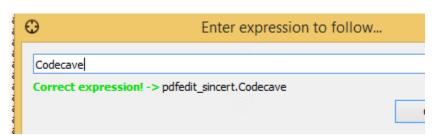
<img44:Entrypoint antes del cambio en x32dbg >

Después de una prueba de inline.

```
push <pdfedit_sincert.Codecave>
ret
                                                                                                                                                                                               salto al codecave
                                                                                                                                                                                                                                                                                                            Hide FPU
00B4FA91 -.
                                                                              push 0x0
push 0x0
dec ecx
jne kpdfedit_sincert.oep>
push ebx
                                                                                                                                                                                                                                                                                                          EBX
                                                                                                                                                                                               oep
                                                                                                                                                                                                                                                                                                          ECX
EDX
                                                                                                                                                                                                                                                                                                          EBP
                                                                                                                                                                                                                                                                                                                        0018F9
                                   F9
  0B4FA99
                                                                                                                                                                                                                                                                                                          ESP
ESI
  OB4FA9E
                                                                              push esi
push edi
   0B4FA90
                                                                                                                                                                                                                                                                                                                        000007
                                                                              push edi
mov eax, pdfedit_sincert.B3E988
call pdfedit_sincert.41111C
mov edi, dword ptr ds:[0xB783FC]
   DB4FA9D
                           57
B8 88 E9 B3 00
E8 74 16 8C FF
8B 3D FC 83 B7 00
33 C0
55
68 36 11 B5 00
64 FF 30
64 89 20
8B 07
E8 2D 65 B0 FF
8B 07
  OB4FA9E
OB4FAA3
OB4FAA8
                                                                                                                                                                                                                                                                                                          Multiline Ultimate Assembler
                                                                                                                                                                                                              4
                                                                                                                                                                                                                           5
                                                                                                                                                                                                                                         6
                                                                                                                                                                                                                                                     7
   OB4FAAB
                                                                              xor eax, eax
push ebp
push <pdfedit_sincert.sub_B51136>
push dword ptr ff:[eax]
mov dword ptr ff:[eax], esp
mov eax, dword ptr ds:[edi]
call pdfedit_sincert.65FF0
mov eax, dword ptr ds:[edi]
mov edx, pdfedit_sincert.B51154
call pdfedit_sincert.65FA20
mov di, 0x1
mov eax, dword ptr ds:[0x492994]
                                                                              xor eax. eax
                                                                                                                                                                                    2
                                                                                                                                                                                                  3
   OB4FAB0
                                                                                                                                                                                <012A6000>
    B4FAB:
                                                                                                                                                                               @Codecave:
                                                                                                                                                                                                                                                                                                                          ٨
                                                                                                                                                                                push ebp ;instruccion original del entrypoint
mov ebp, esp;instruccion original del entrypoint
mov ecx, 0x42;instruccion original del entrypoint
                                                                                                                                                                               push ebp
                                  07
2D 65 B0 FF
   OB4FAB0
                                                                                                                                                                                                                                                                                                                               01 01 7!
  OB4FABE
 0084FAC3
0084FAC3
0084FACA
0084FACA
0084FACA
                            8B 07
BA 54
E8 51
B2 01
A1 94
E8 9D
                                  07
54 11 B5 00
51 5F B0 FF
01
                                                                                                                                                                               push @oep :retorno al oep
                                                                                                                                                                                 et ;retorno al oep
                                                                                                                                                                                <00B4FA8C>
                                                                              mov eax, dword ptr ds:[0x492994]
call pdfedit_sincert.487178
                                         29 49 00
76 96 FF
 OOB4FAD6
                                                                                                                                                                               push @Codecave ;salto al codecave
                             8B F0
E8 5E
8B D8
                                                                              mov esi, eax
call <pdfedit
mov ebx, eax
   OB4FADE
                                                                                                                                                                               ret
nop
                                          64 8C FF
                                                                                                             sincert.GetTickCount:
```

<img45:Entrypoint despues del cambio en x32dbg >

Lo especial y lo que me gusta cuando usas label, es que puedes ir fácilmente con x32dbg al nuevo label creado



<img46:comandos en x32dbg permiten ir a la dirección del label>

Hemos hecho una pausa para practicar como podemos robar unas pequeñas instrucciones, hasta aquí lo defino como Codecave, Esto es para probar solamente que podemos raptarnos código de donde queramos, el principio será el mismo en todos los lados que robemos instrucción original y me permitirá extender la funcionalidad, ósea si robamos 10 instrucciones, esas instrucciones, deberán estar en el lugar del inline, pero además debemos mantener balanceado nuestros valores para esp por lo que podemos hacer uso de pushad o bien push (registro), aquí definimos labels para hacer los llamados al igual como hacíamos con el mensajebox, aquí usamos una etiqueta creada por nosotros, un label es una etiqueta que se realiza en la dirección definido en MLA. (multiline ultimate assembler.), es similar al principio de comentario, pero el label es definido automáticamente o por el usuario como un arroba @, así como una api es llamada por su nombre, este label es llamado solo si lo necesitamos usar (en el depurador es llamado sin el arroba).

La acción de parcharlo de forma ordenada desde el oep, requiere tiempo, comenzamos con la primera idea (no es la que usaremos al final, pero es para demostrar cómo se hace. Veamos aquí necesitamos hacer fill de mov edx, valor a mov edx,0)

<img47 watermark encriptado>

El primer byte en el caso de mov (es el BA) (mov edx), los siguientes bytes son de la string, así que las direcciones reales a parchar serian A24994 y 0A249a1 con 0, Aquí comenzaremos directamente con multi line Multiline Ultimate Assembler en x32dbg, Comenzamos a definirlas como string1, string2

```
Definimos label llamadas
                               Y ahora el parche será
                                                                            Veamos como estamos al minuto : no puede ensamblar porque no hay
String1 y String2
                                 mov dword ptr
                                                 [@string1], 0
                                                                            permisos:
                                                [@string2],
                                 mov dword ptr
                                                                              First chance exception on 012A601B (C0000005, EXCEPTION_ACCESS_VIOLATION)!
                                push @oep ; retorno al oep
  <00A24994>
                               <imq49 usando Multiline Ultimate
 @string1:
                                                                            <img50 usando Multiline Ultimate Assembler>
                               Assembler>
 @string2:
<img48 usando Multiline
Ultimate Assembler>
```

Entonces invocaré a virtualprotect, con el orden determinado, dirección, tamaño, protección, se ensamblan en inverso, para tener permisos en la sección

```
push 4 ; PAGE_READWRITE

push 4 ; tamaño

push @string1

call 0x416180 ; virtualprotect

; 0018FF4C 009DCB00 /CALL to VirtualProtect from soundman.00!

; 0018FF50 00400000 |Address = soundman.00400000

; 0018FF54 00001000 |Size = 1000 (4096.)

; 0018FF58 00000004 |NewProtect = PAGE_READWRITE
```

<img51 invocando a Virtualprotect en Multiline Ultimate Assembler>

En la aplicación disponemos de virtualprotect

```
| 00416180 | V FF 25 E4 ED E5 00 | jmp dword ptr ds:[<&VirtualProtect>] | VirtualProtect | 00416186 | 88 C0 | mov eax, eax
```

Luego identifico el objetivo a cambiar

<img53 watermark>

Para que quede así:

```
00A24A29 4F dec edi
00A24A2A 75 D8 jne pdfedit.A24A04

00A24A2C 6A 00 push 0x0
00A24A2F 6A 00 push 0x0
00A24A2F 6A 00 push 0x0
00A24A31 90 nop
00A24A32 90 nop
00A24A33 90 nop
00A24A34 FF 75 F4 push dword ptryss:[ebp - 0xc]
```

<img54 modificado (sin watermark)>

Vamos a colocar desde A24A2c hasta A24A33 NOP (la zona de finalizar el nop seria A24A34 y el el dword esperado 6A00 en las 2 posiciones que quiero

Para el bucle del nop coloco

```
pushad
xor eax,eax
mov eax,@InicioNop1
@bucle1:
mov byte ptr [eax], 0x90
inc eax
cmp eax,@FinNop1
jne @bucle1
popad
```

Luego para parchar los bytes 6A00 debo moverlos en sentido inverso al push 0, osea 0x006A

```
012A604F
012A6050
012A6059
                                                                                                                                               push @oep ;r
                   66 C7 05 2C 4A A2 00
66 C7 05 2F 4A A2 00
                                                  mov word ptr ds:[<InicioNop1>], 0x6A
mov word ptr ds:[<string3], 0x6A
push <pdfedit_sincert.oep>
                                                                                                                                                ret ; retorno
                   68 94 FA B4 00
012A6062
                                                                                                             <pdd><pdfedit_sincert.InicioNop1>
                                                   ret
012A6067
                                                                                                push 0x0
012A6068
<
                                                                                                push 0x0
                                                                                                 nop
                                                                                                 nop
                                                                                                 gon
                                                                                                push dword ptr ss:[ebp_- 0xC]
_sincert.exe:$EA6062 #BBC262
                                                                                                lea eax, dword ptr ss:[ebp -
                                                                                                mov edx,
```

<img55 probando capacidad de Multiline Ultimate Assembler>

Seguimos

```
00A24C43
                 OF 85 4B FE FF FF
                 A1 34 87 B7 00
8B 00
00A24C49
                                                   eax, dword ptr
                                                                       ds:[eax]
ds:[eax + 0x3E8]
00A24C4E
                                             mov eax,
                                                         dword ptr
                 8B 80 E8 03 00 00
                                                         dword ptr
                                             mov eax.
00A24C56
                 BA CO
                                                                                                                   A252CO:L"This text only appears i
                                             mov edx,
                 E8 D0 B5 B1 FF
8B 06
00A24C5B
00A24C60
                                             mov eax, dword ptr ds:[esi]
mov byte ptr ds:[eax + 0x14F4], 0x1
                 C6 80 F4 14 00 00 01
00A24C62
                                                                                                                          B
00A24C69
00A24C6B
                 8B 06 mov eax, dword ptr ds:[esi]
C6 80 9D 13 00 00 mov byte ptr ds:[eax + 0x139D], 0x0
8B 06 mov eax, dword ptr ds:[esi]
00A24C72
00A24C74
                         F4 14 00 00 01 mov byte ptr ds:[eax + 0x14F4], 0x1
```

<img56 watermark>

Esta zona requiere también que sea mov edx,0

Repetimos la forma anterior (darle permisos, luego mover a 0)

Asi que al minuto vamos así

```
<00A24994>
@string1:
<00A249A1>
@string2:
<00A24a2F>
@string3:
<00A24c57>
@string4:
<009C37A6> ;je a jmp
jmp 0x9c37e0
<00A24a2C>
@InicioNop1:;@string3
<00A24a34>
@FinNop1:
<012A6000> ;codecave
@Codecave:
```

```
push ebp ;instruccion original del entrypoint
mov ebp, esp;instruccion original del entrypoint
mov ecx, 0x42;instruccion original del entrypoint
push 4; PAGE READWRITE
push 4 ;tamaño
push @string1
call 0x416180 ;virtualprotect
push 4; PAGE_READWRITE
push 4 ;tamaño
push @string2
call 0x416180 ;virtualprotect
push 4; PAGE_READWRITE
push 4 ;tamaño
push @string4
call 0x416180 ;virtualprotect
;0018FF4C 009DCB00 /CALL to VirtualProtect from soundman.009DCAFE
;0018FF50 00400000 | Address = soundman.00400000
;0018FF54 00001000 | Size = 1000 (4096.)
;0018FF58 00000004 | NewProtect = PAGE_READWRITE
mov dword ptr [@string1], 0
mov dword ptr [@string2], 0
mov dword ptr [@string4], 0
pushad
xor eax, eax
mov eax,@InicioNop1
@bucle1:
mov byte ptr [eax], 0x90
inc eax
cmp eax,@FinNop1
ine @bucle1
popad
mov word ptr [@InicioNop1], 0x006A
mov word ptr [@string3], 0x006A
push @oep ;retorno al oep
ret; retorno al oep
nop
<00B4FA8C>
push @Codecave ;salto al codecave
ret
nop
nop
@oep:
```

Dado que llevamos muchos parches, es ideal también saber que no existe solo la forma de usar ensamblador como si inlinearamos, sino además podemos ensamblar directamente la instrucción (este es la forma que usaremos esta vez) y es la más efectiva, dado que no necesitamos nuevamente pedir permisos a las secciones

USANDO MULTILINE ULTIMATE ASSEMBLER CON EL DEPURADOR

Veamos el salto je a jmp 9C37A6 puedo decir que lo ensamblaré directamente.

```
mov eax, dword ptr ss:[ebp - 0x28]
mov edx, pdfedit.9C51D4
call pdfedit.4OCA40
je pdfedit.9C37E0
                                         8B 45 D8
                                                                                                                                                                                                                                                                                                                                                                                         Hide FP
                                                                                                                                                                                                                                                                                                          9C51D4:L"VIEWER"
 009C37A6
                                                                                                       mov eax, dword ptr ss:[ebp - 0x4]
push dword ptr ds:[eax + 0x1380]
push pdfedit.9C51F0
lea edx, dword ptr ss:[ebp - 0x34]
                                        -74 38
8B 45 FC
FF B0 B0 13 00 00
68 F0 51 9C 00
8D 55 CC
B8 4A 00 00 00
                                                                                                                                                                                                                                                                                                                                                                                        ECX
                                                                                                                                                                                                                                                                                                                                                                                       EDX
EBP
                                                                                                                                                                                                                                                                                                          9C51F0:L" - '
                                                                                                     lea eux, 0x4A
mov eax, 0x4A
call pdfedit.7A45D4
push dword ptr ss:[ebp - 0x34]
lea eax, dword ptr ss:[ebp - 0x30]
mov edx, 0x3
mov edx, 0x3
                                                                                                                                                                                                                                                                                                          4A:'J'
                                        E8 11 0E DE FF
FF 75 CC
8D 45 D0
009C37C3
                                                                                                                                                                                                                                                                                                           [ebp-34]:L"DEMO VERISON
                                                                                                                                                                                                                                                                                                                                                                                        EDI
                                        BA 03 00 00 00
E8 5D 91 A4 FF
                                                                                                                                                                                                                                                                                                                                                                                       EIP
                                                                                                      call pdfedit.40C930
mov edx, dword ptr ss:[ebp - 0x30]
mov eax, dword ptr ds:[0xE47398]
call pdfedit.540230
lea edx, dword ptr ss:[ebp - 0x38]
mov eax, dword ptr ds:[0xE47398]
call pdfedit.5401DC
mov edx, dword ptr ss:[ebp - 0x38]
mov eax, dword ptr ss:[ebp - 0x4]
add eax, 0x1380
call pdfedit.40B978
                                        88 55 D0
A1 98 73 E4 00
E8 50 CA B7 FF
SD 55 C8
A1 98 73 E4 00
   009C37D3
009C37D6
                                                                                                                                                                                                                                                                                                                                                                                        EFLAGS
                                                                                                                                                                                                                                                                                                                                                                                       ZE 1 PI
OF 0 SI
CF 0 TI
   009C37DB
                                       88 50 CA B/ FF

8D 55 C8

A1 98 73 E4 00

E8 EF C9 B7 FF

8B 55 C8

8B 45 FC
   009C37E3
  009C37E8
009C37ED
009C37F0
                                                                                                                                                                                                                                                                                                                                                                                        LastErre
                                        05 B0 13 00 00
E8 7B 81 A4 FF
                                                                                                                                                                                                                                                                                                                                                                                         GS 002B
                                                                                                                                                                                                                                                                                                                                                                                        ES 002B
```

<img57 limitación de titulo >

Para ello usaremos el jmp directo, los comentarios en MLA. Se hacen con punto y coma ";"

```
<009C37A6> ;je a jmp
jmp 0x9c37e0
```

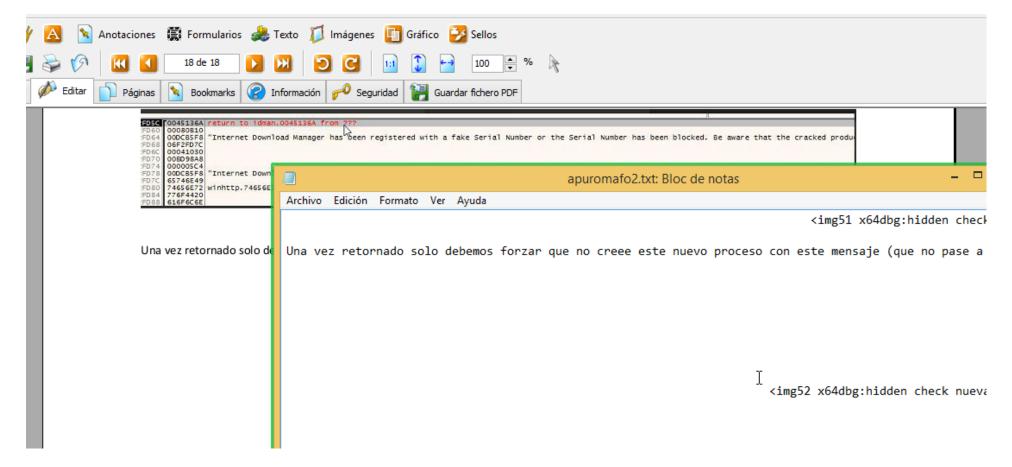
Ósea en palabras simples, si queremos parchar uno a uno las instrucciones, haciendo uso de esta herramienta es posible hacerlo colocando que instrucciones necesitamos colocar una a una, obviamente nosotros sabremos decirle al programa que instrucciones necesitamos colocar, asi podemos ir parchando uno a uno, y aunque reiniciemos 100 veces podemos ir parchando las direcciones de forma precisa, sin tener que hacer cambio_1.exe cambio_2.exe y más, sin necesidad de parchar nada mediante nuevos permisos, podemos ensamblar directamente los bytes a los bytes necesarios dado que usa el depurador en cuestión y nos damos cuenta que usa muy poco espacio:

```
<00A24993>
mov edx,0; string1
<00A249A0>
mov edx,0 ;string2
<00A24a2F>
push 0; ;string3
push 0; ;string4
nop
nop
nop
nop
<00A24c56>
mov edx,0; ;string5
<009C37A6> ;je a jmp
jmp 0x9c37e0 ;demo1
;ahora el lugar para definir el espacio de los nuevos parches
<012A6000> ;codecave
@Codecave:
```

Hasta aquí diríamos que estamos relativamente listo para la parte final y comienzo a ver si realmente logro extender la funcionalidad

```
<00A74D0C> //dirección donde esta la comparación del texto
inc dword ptr [ebp - 0xa4] //aquí le dice que termino la pagina 1 pushad //pushad guarda registros
mov eax, dword ptr [ebp - 0x1c] // instrucción 1 mov eax, dword ptr [eax + 0x122c] // instrucción 2 mov eax, dword ptr [eax + 0x12c] // instrucción 3 cmp dword ptr [ebp - 0xa4], eax //comparación que realmente necesito je @L00000001 //si son iguales devuelve el flujo de lo que continua popad //devuelve el valor antes del injerto
@L00000008:
cmp dword ptr [ebp - 0xa4],99999 //comparación con 99999 hex paginas jne 0xa73a02 //instrucción original que comparaba pagina actual versus numero gigante (antes decía 2)
@L00000001:
mov eax, dword ptr [ebp - 0x1c] //instrucción que continua el flujo
```

Hasta aquí estaría feliz Pero aun hay un error, si el valor a comparar con 99999 hay un pushad, cuando salte no hara el popad, por lo que debemos forzar devolver el flujo luego del popad y luego recién saltar al flujo normal Con eso reparado nuestro injerto funciona, pero pasa algo, aun asi no tenemos todas las paginas(porque si tenemos que comparar debe ser con 1 mas xD)



<img58 verificando cuantas paginas llevaba convertida >

(estoy probando sobre el pdf que escribi sobre idm tiene 18 páginas) Le falta solo 1 pagina, probemos 2 casos (antes de la comparación, el valor de eax, tiene el total de páginas)

Agregando dec eax, antes de la comparación:	Agregando inc eax,antes de la comparación
Me convierte 2 paginas menos	Me convierte con éxito las paginas antes buscado

```
<00A24993>
mov edx,0
<00A249A0>
mov edx,0
<00A24a2F>
push 0
push 0
nop
nop
nop
nop
<008B5A44>
@sigue2:
<00A24c56>
mov edx,0
<009C37A6> ;je a jmp
jmp 0x9c37e0
<012A6000> ;codecave
@limitacion_txt:
mov dword ptr [eax + 0x48], edx
inc dword ptr [ebp - 0xa4] ;aquí le dice que termino la pagina 1
       pushad ;pushad guarda registros
mov eax, dword ptr [ebp - 0x1c] ;instrucción 1
       mov eax, dword ptr [eax + 0x122c]; instrucción 2
       mov eax, dword ptr [eax + 0x10]
                                             ; instrucción 3
       inc eax
       cmp dword ptr [ebp - 0xa4], eax; comparación que realmente necesito
       je @continua txt ;si son iguales devuelve el flujo de lo que continua
       popad ;devuelve el valor antes del injerto
       cmp dword ptr [ebp - 0xa4],0x9999999 ;comparación con 99999 hex paginas
       jne 0xa73a02 ;instrucción original que comparaba pagina actual versus numero gigante (antes decía 2)
@continua_txt:
popad
jmp @sigue_txt
@limitacion_word:
inc dword ptr [ebp - 0x174]
pushad ;pushad guarda registros
       mov eax, dword ptr [0xb77d28]
       mov eax, dword ptr [eax]
       mov eax, dword ptr [eax + 0x122c]
       mov eax, dword ptr [eax + 0x10]
       inc eax
cmp dword ptr [ebp - 0x174],eax
je @continua_word
popad
dec dword ptr [ebp - 0x1f0]
jmp 0x8b2b53
@continua_word:
popad
jmp @sigue2
<008B5A32>
Push @limitacion_word
ret
nop
nop
nop
nop
nop
nop
<00A74D09>
push @limitacion_txt
ret
       nop
       nop
```

```
nop
nop
nop
@sigue_txt:
```

Luego me propongo ir a la versión de exportar en excel (es muy similar a la de txt), ahora bien luego me propongo ir a la versión de exportar imágenes y aquí si lo pensamos de forma crítica, no nos advierte ningún mensaje, por lo que deberemos buscar como eliminamos el watermark en el pasado

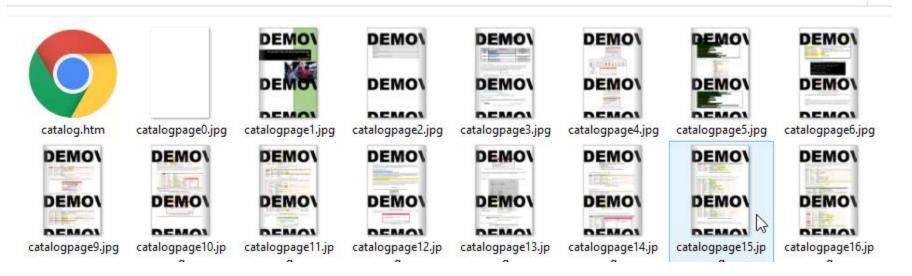
aquí por ejemplo llamaba a 40b9c0

```
Original
                                                              call dword ptr ds:[ecx + 0x10C]
mov eax, dword ptr ds:[ebx]
mov byte ptr ds:[eax + 0x11F1], 0x1
lea eax, dword ptr ss:[ebp - 0x10]
                            FF 91 0C 01 00 00
         0A24981
         DA24987
                            88 03
                            C6 80 F1 11 00 00 01
                            80 45 F0
BA FC 51 A2 00
   00A24993
                           E8 23 70 9E FF
8D 45 F4
BA 80 52 A2 00
         0A2499D
                                                                lea eax, dword ptr ss:[ebp - 0xC]
                                                               mov edx, pdfedit.A2528
call pdfedit.4089C0
                                                                                                                                                    A25280:L"8mknuk}}8myw38"
         DA249A5
                            E8 16 70 9E FF
         DA249AA
                                                                mov eax, dword ptr ss:[ebp - 0x10]
                                                               test eax, eax
je pdfedit.A24986
sub eax, 0x4
         0A249AD
                            85 CO
         DA249AF
                            74 05
         DA24981
                                                              mov eax, dword ptr ds:[eax]
mov edi, eax
test edi, edi
jle pdfedit.A249E8
mov dword otr ss:[ebn = 0v26] fivi
        00A249B4
00A249B6
                            88 00
                            8B F8
                            85 FF
        0A249BA
```


Probamos exportar como imagen y llegamos al lugar en cuestión 9BD3F4 aquí tenemos la palabra en otro orden

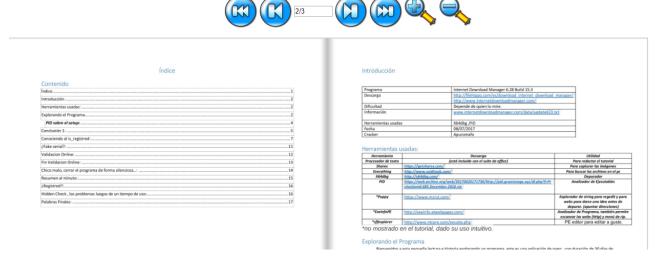
DE		MO	VE	RS	IO	N
9DB5AC		9DB5D4	9DB5FC	9DB5C0	9DB5AC	9DB610
	009BD3F	4 BA AC D5 9B 00	mov edx, apuroma	fo_tute.9BD5AC	9BD5AC:L"DE	"
	009BD3F	9 E8 C2 E5 A4 FF	<pre>call apuromafo_t</pre>	ute.40B9C0		
	009BD3F		mov eax, edi			
	009BD40	0 BA CO D5 9B 00	mov edx, apuroma		9BD5C0:L"RS	"
	009BD40	5 E8 B6 E5 A4 FF	<pre>call apuromafo_t</pre>	ute.40B9C0		
	009BD40		mov eax, esi			&
	009BD40	C BA D4 D5 9B 00	mov edx, apuroma		9BD5D4:L"MO	00
	009BD41		<pre>call apuromafo_t</pre>	ute.40C850		
	009BD41		mov eax, edi			
	009BD41		mov edx, apuroma		9BD5E8:L"IO	**
	009BD41	D E8 2E F4 A4 FF	<pre>call apuromafo_t</pre>	ute.40C850		
	009BD42		mov eax, esi			
	009BD42		mov edx, apuroma		9BD5FC:L"VE	
	009BD42		<pre>call apuromafo_t</pre>	ute.40C850		
	009BD42		mov eax, edi			
	009BD43		mov edx, apuroma			
	009BD43	5 E8 16 F4 A4 FF	call apuromafo t	ute.40C850		

Antes de modificar



Después de la modificación no hay rastro de demo version

Vemos que al anular esta vez edx con 0 mata 2 limitaciones (exportar imágenes) y el que crea los flip pages (usa el mismo lugar para marcar el watermark)



Probamos el que refiere imprimir, refiere que creara un watermark (pero al parecer también fue vencido con el primera), no hay watermark



Sigo probando el menú (imprimir pdf)

Mandamos a imprimir y existe el mismo watermark de imágenes si pasa a ser llamado por el call que teníamos alerta, asi que colocamos la misma estrategia y pillamos de nuevo nuestras frases demo versión en orden desordenado

```
mov eux, UXIE

call apuromafo_tute2.511634

mov eax, dword ptr ss:[ebp + OX:
add eax, OXFFFFFB44

mov_edx, apuromafo_tute2.9BD2F0
                                          BA 1E 00 00 00
E8 6F 45 B5 FF
8B 45 08
05 44 FB FF FF
BA FO D2 9B 00
009BD0C0
009BD0C5
                                                                                                                                                                                   ss:[ebp + 0x8]
                                                                                                                                                                                                                                                                                 9BD2F0:L"DE
                                                                                                                   mov eax, apuromaTo_tute2.9BD2F0

call apuromafo_tute2.40B9C0

mov eax, dword ptr ss:[ebp + 0x8]

add eax, 0xFFFFF9E8

mov edx, apuromafo_tute2.9BD304

call apuromafo_tute2.40B9C0

mov eax, dword ptr ss:[ebp + 0x8]

add eax, 0xFFFFFB44

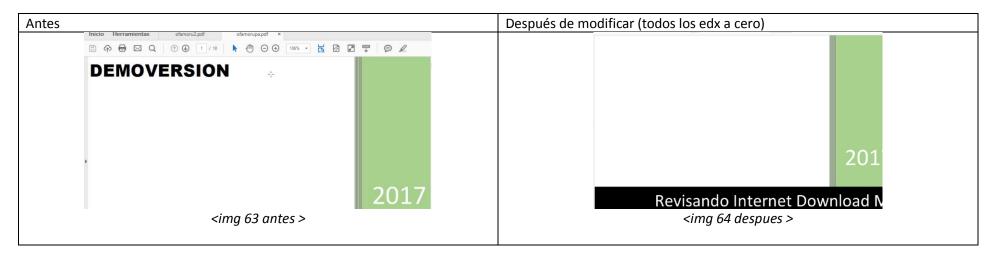
mov edx, apuromafo_tute2.9BD318
                                           E8 E9 E8 A4 FF
8B 45 08
05 E8 F9 FF FF
                                          05 E8 F9 FF FF
BA 04 D3 9B 00
E8 D7 E8 A4 FF
OO9BDODA
                                                                                                                                                                                                                                                                                  9BD304:L"RS"
                                          E8 D7 E8 A4 FF
8B 45 08
05 44 FB FF FF
BA 18 D3 9B 00
E8 55 F7 A4 FF
8B 45 08
05 E8 F9 FF FF
BA 2C D3 9B 00
E8 40 F7 A4 FF
8B 45 08
 009BD0E0
                                                                                                                    mov eax, dword ptr ss: [ebp + 0x8 mov eax]
 009BD0E1
                                                                                                                                                                                                                                                                                 9BD318:L"MO'
 009BD0FB
  009BD0FE
 009BD101
                                                                                                                     add eax, 0xFFFFF9E8
                                                                                                                    mov edx, apuromafo_tute2.9BD32C

call apuromafo_tute2.40C850

mov eax, dword ptr ss: [ebp + 0x8

mov eax, dword ptr ss: [ebp + 0x8
 009BD106
009BD10B
                                                                                                                                                                                                                                                                                 9BD32C:L"IO"
                                                    45 08
45 08
44 FB FF FF
40 D3 9B 00
2B F7 A4 FF
                                           8B
8B
05
BA
E8
                                                                                                                    mov eax, dword ptr s
mov eax, dword ptr s
add eax, OxFFFFFB44
 009BD110
 009BD113
 009BD116
                                                                                                                     mov edx,
call apur
  009BD11B
                                                                                                                                                                                                                                                                                 9BD340:L"VE"
                                                                                                                                                  comafo_tute2.40C850
dword ptr ss:[ebp + 0x8
dword ptr ss:[ebp + 0x8
 009BD120
                                          88 45 08
88 45 08
05 E8 F9 FF FF
BA 54 D3 98 00
E8 16 F7 A4 FF
 009BD125
009BD128
                                                                                                                                 eax, dword ptr
eax, dword ptr
                                                                                                                    add eax,
mov edx,
call apur
 009BD12B
                                                                                                                                                  0xFFFFF9E8
                                                                                                                     call apuromafo_tute2.40C850
mov eax. dword ptr ss:[ebp + 0x8]
 009BD135
 009BD13A
```


Antes de modificar en los pdf se aprecia la diferencia denuevo

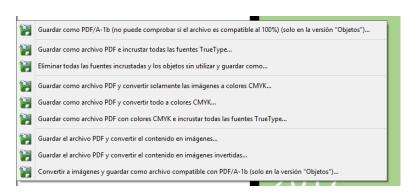


Medio cansado de explorar el programa, ya es funcional en parte, hacemos resumen

Desde el original que hemos quitado el certificado, se ha modificado el titulo, se ha quitado el watermark para pdf en varias opciones, se ha extendido la función para txt, para Excel, para Word, Ahora falta la ultima parte estética que es seguir probando que sirve y que no, quedan toda la parte estética en manos de resource hacker.

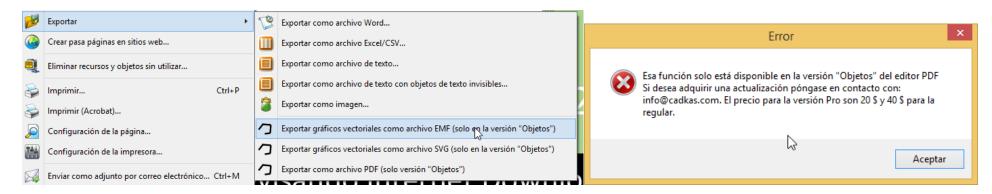
Adicionalmente quedan algunas opciones disponibles solo en objetos (otra versión he de asumir)

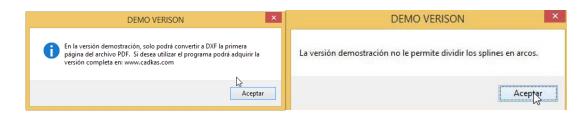
Por el lado estético guardar como (la primera y ultima funcionan) no es solo versión objetos..



LADO DEMO

Por el lado de solo objetos esta capado, aunque por dentro si vemos que se puede ver que si existen las extensión svg y emf, pero se deben eliminar del menú como tal porque los deja muy mal hechos.

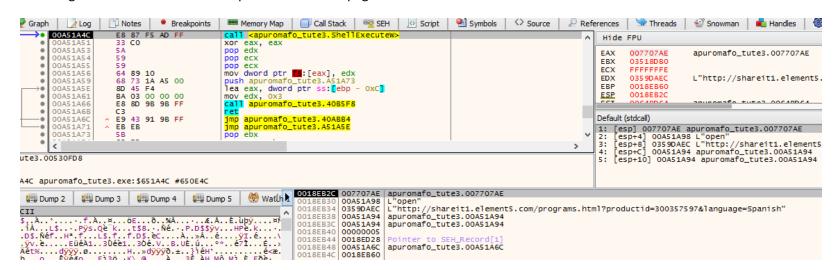




por otro lado en su web existen conversores triales para DXF(modo gráficos para software tipo autocad) http://www.cadkas.com/ehome.php

ULTIMA LIMITACIÓN, EJECUCIÓN DE WEB SIN NUESTRO CONSENTIMIENTO.

Ahora sigamos cuando cerramos la aplicación va a una pagina con BP en el call a ShellExecuteW



Proviene de aquí:



Para anular es fácil, o se fuerza el salto de je a jmp o se anula el call de A51A4c a otra instrucción ejemplo NOP/jmp A51A51 esto se parece mucho al comienzo cuando colocaba en el titulo demo.

El remate final para los valientes. **Aún con todo parchado en sus limitaciones aún quedan todos los nags** una idea dada por Jhon (que me ayudó a revisar la ortografía), seria que cada vez que llame a demo, se anulara el string, pero bueno, son temas de cada quien pueda y para rematar el descubrimiento si bajamos la versión actual al dia de hoy 8/8/17 vemos que ha incrementado un poco el ejecutable(setup), asi que ha cambiado el programa, por lo que mis direcciones no deberían ser iguales a las suyas, pero los principios seguirían siendo el mismo. Y además existe otra versión pdfedit!.exe osea quien sabe cuantos programas demo crean estos autores.



Palabras Finales:

Tengo un programa demo funcional sin watermark, para quien obviamente lo necesite, ahora bien si alguien puede dar soporte es ideal que compren los que pueden realmente, aquí queda plasmada una experiencia en un programa demo (capado) que si es posible quitarle el watermark y extender muy poquito su funcionalidad, agradecer a AbelJM por sugerir el programa, Jhon por revisar conmigo la redacción, y a ti por leer una porción de mis recuerdos.

Si te complica la ortografía, corrígela con este mismo software, que permite editar pdf 😅

Dedicado a los lectores que suelen practicar y/o aprender reversing o simplemente una lectura amena, está más que decir que si te ha gustado el software y si tienes la posibilidad de comprarlo no dejes de apoyar al soporte del programa.

saludos Apuromafo



Apuromato TSRh