

Win Snap 5.1

By Apuromafo



24 de abril de 2019

# INDICE

Contenido

[INDICE 1](#_Toc8240818)

[Introducción 2](#_Toc8240819)

[Frase. 2](#_Toc8240820)

[Herramientas usadas en el Escrito: 3](#_Toc8240821)

[Analizando al Setup: 4](#_Toc8240822)

[***Reparando UI Access+ permiso de admin*** 8](#_Toc8240823)

[***Eliminando Firma Digital*** 8](#_Toc8240824)

[***Reparando checksum*** 9](#_Toc8240825)

[DEPURANDO X86 11](#_Toc8240826)

[DEPURANDO X64 25](#_Toc8240827)

[Anexos: 33](#_Toc8240828)

[Palabras Finales 38](#_Toc8240829)

Introducción

|  |  |
| --- | --- |
| Programa | WinSnap v5.1.0 x86- x64 |
| Descarga | <https://www.ntwind.com/download/WinSnap_5.1.0-setup.exe> |
| Dificultad | Depende de quien lo mire |
| Objetivo | Hacerlo funcional |
| Información | Programa para realizar capturas de pantalla |
| Herramientas usadas | X64dbg , cff explorer |
| Fecha | 24-04-2019 |
| Fecha Liberación Tutorial | 08-05-2019 |
| Autor | Apuromafo |

Frase.

***"El sabio puede cambiar de opinión. El necio, nunca."***

***Immanuel Kant***



***<img1.Splash inicial, contiene timer, y un botón desactivado(que se habilita)>***

Herramientas usadas en el Escrito:

|  |  |  |
| --- | --- | --- |
| ***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* |
| ***X64dbg*** | ***http://x64dbg.com/*** | *Depurador* |
| ***7zip*** | ***http://www.7-zip.org/download.html*** | *Descomprimir archivos* |
| ***Notepad ++ \**** | https://notepad-plus-plus.org/ | *Editar archivos (no hexadecimal)* |
| ***ExeinfoPe 0.55*** | ***https://www71.zippyshare.com/v/Oa1dLDlM/file.html*** | *Analizador de ejecutables* |
| ***Cff explorer*** | <https://ntcore.com/?page_id=388> | Editor Pe header |
|  |  |  |

\* Herramienta opcional

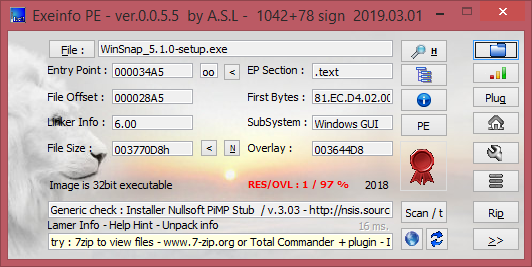
Historia :

Me invitaron a conocer el programa, veamos que tal lo encontramos.

|  |
| --- |
| **WinSnap - Create Superior Screenshots** WinSnap is a fast user-friendly utility for taking and editing screenshots. It easily captures windows of non-rectangular form with transparent backgrounds including Aero Glass and instantly enhances screenshots with professional-looking shadows, reflections, outlines, watermarks and color effects. |

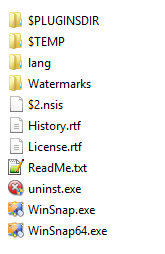
En resumen, es un programa con múltiples opciones.

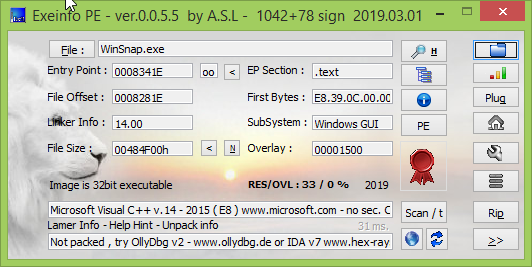
Analizando al Setup:



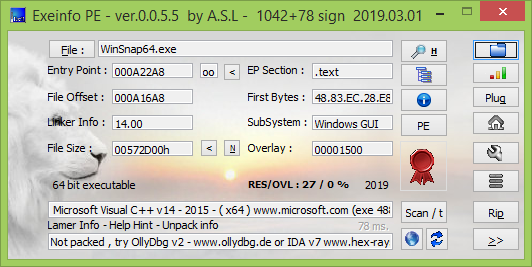
Nullsoft es un generador de setup gratuito disponible en sourceforge, además tiene capacidad de usar macros y detectar que tipo de arquitectura usamos (x86 y x64) <https://sourceforge.net/p/nsis/code/HEAD/tree/NSIS/trunk/Include/x64.nsh>

Casi siempre usan algoritmos aplib, el cual el 7zip descomprime de forma extraordinaria, para ver un poco mas los otros archivos, favor ver en \*Anexo\*, asi que probemos:



Usamos 7zip y tenemos ahora los 2 ejecutables (marcados con marca digital) tanto para x86 y x64  
  


<el x86 >



<el x64>

La firma digital contiene la información:

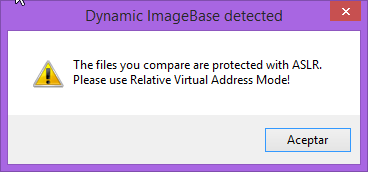
|  |  |
| --- | --- |
| Nombre | Avdonin Aleksandr Nikolaevich Ip |
| Correo | [alexander@ntwind.com](mailto:alexander@ntwind.com) |
| Fecha | ‎martes, ‎23‎ de ‎abril‎ de ‎2019 |

Comenzamos a explorar en el depurador:

Intentamos usarlo en x64dbg o x32dbg según corresponda y nos muestra: el mensaje de error

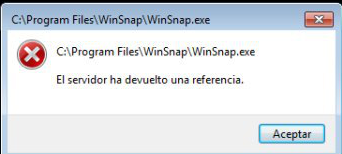


Abrimos cff explorer y modifico un byte cualquiera, guardo y comparo en DUP refiere:



Ósea además tiene aslr, pero es editable.

Si observo una ejecución parchada y ejecutarlo sin ser admin:



Strings internas:  
validación de versión (posiblemente para updates)

|  |
| --- |
| <http://www.ntwind.com/software/winsnap/version.txt>  (el día del tutorial muestra 5.1.0.0)  Tutorial de uso  <http://www.ntwind.com/tutorials.html?tags=winsnap>  como registrarse.  <http://www.ntwind.com/software/winsnap/how-to-register.html>  <http://www.ntwind.com/software/winsnap/register.html> |

Resumimos

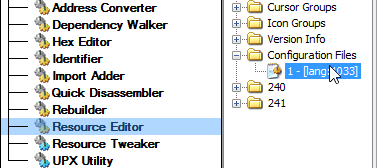
1. Hay un certificado digital (requerirá modificar o quitarlo)
2. Hay ASLR (requerirá parchar por offset o bien desactivarlo)
3. Servidor ha devuelto una referencia= Hay exigencia de permisos de administrador (verificar si se puede modificar esa petición)
4. Hay necesidad de reparar uiAccess (o quitar el manifiest y luego volver a colocarlo)
5. Validaciones adicionales como Usuario+clave , registrado/no registrado título, about watermark.

Manos a la Obra

***Reparando UI Access+ permiso de admin***

Lo más complicado es que no se pueda abrir en el depurador, esto ocurre porque no tiene permisos en manifiest, y exige admin luego.

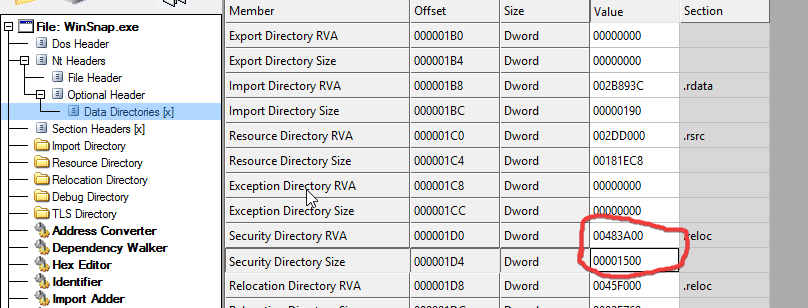
Comenzamos con cff explorer>Resource editor.



Quitamos en el editor de recurso el configuration files

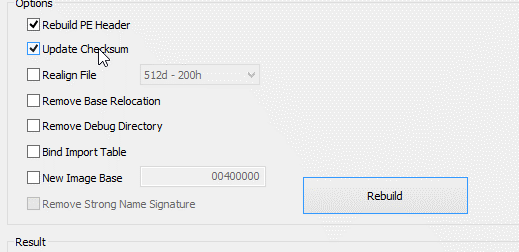
Luego Editamos certificado (a 0) (firma digital)

***Eliminando Firma Digital***



Y luego en rebuilder que reconstruya el header y checksum, así todo estará en orden.

***Reparando checksum***



Guardamos el cambio , para el caso de x86 bajo 3 kb  y lo mismo en x64 

Actualmente con estos cambios ya podremos probarlo en el depurador y comenzar a explorar.



Tenemos una aplicación que no podíamos depurar y ahora la podremos depurar, ahora ¿ que encontraré?

Clasificaré en 2 grandes bloques el de x86 y x64

Ahora abrimos en el depurador, quiero ver su ejecución y logro apreciar que no hay crash ni permisos de admin que solicite, así que solo debemos enfocarnos a las limitaciones y alguna zona de registro.



# DEPURANDO X86

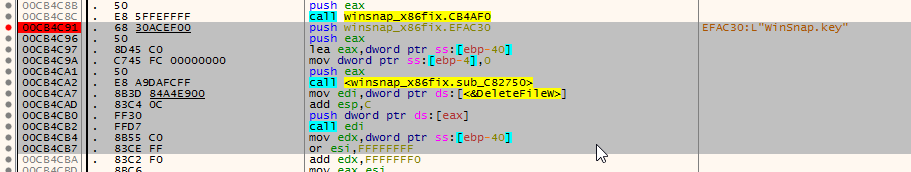
Dejo que se ejecute dice trial,Aparece un mensaje y coloco registrar(enter Licensekey>registrar).



*<nag>*

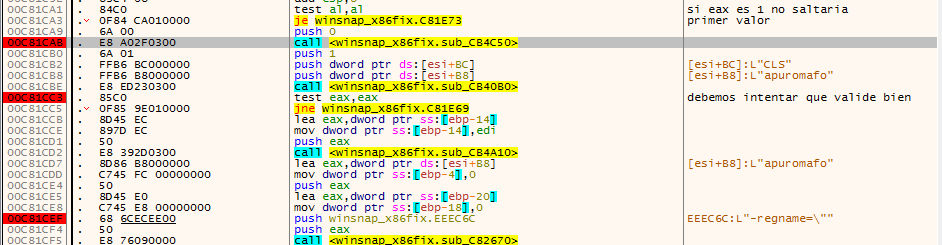
A pesar de encontrar el timer y ver como se va evaluando el continue, prefiero ahora encontrar que referencias llamativas encuentro.  
Busco las referencias, encuentro llamativo “WinSnap.key”

Encuentro una porción de código que refiere



*<elimina winsnap.key>*

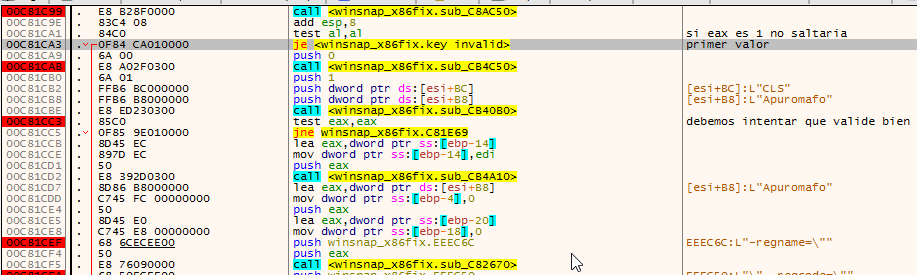
Eliminar el archivo de winsnap.key, esto hace alusión al lugar incorrecto, pero mas arriba debería haber una validación entonces coloco usuario:apuromafo clave:CLS y volveré a mirar.



*<validación usuario+serial>*

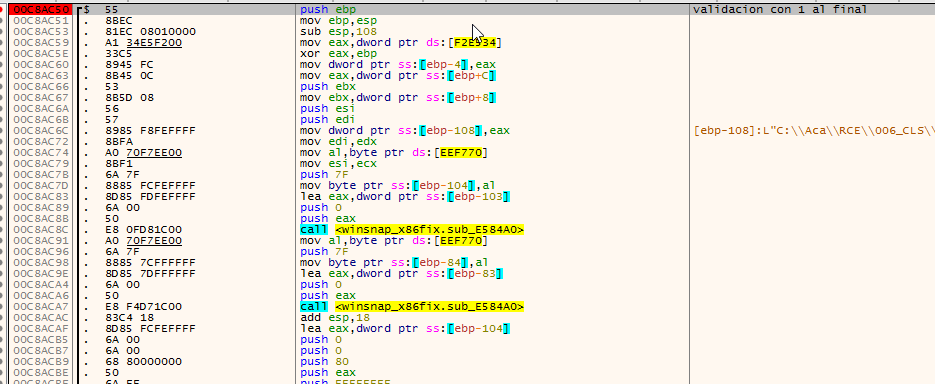
Vemos que la llamada de validación es esta call cb4c50 , entonces solo debo intentar que valide bien y manejará en eax

Entonces ese primer eax va a validar si es válida o no la llave



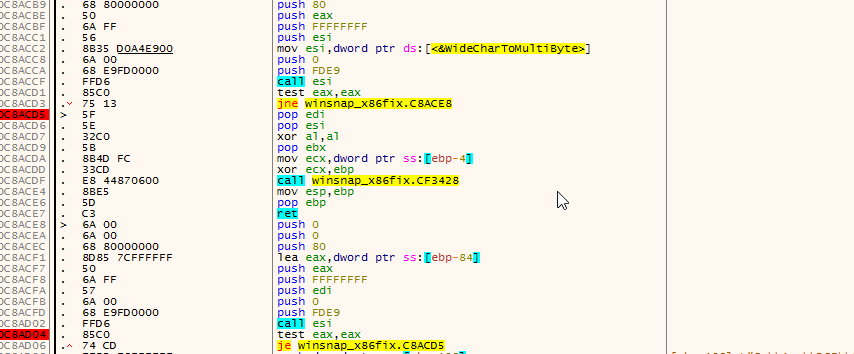
*<comparación antes de elegir si lo ingresa a regedit o no>*

Veamos la rutina por encima



*<manejo de cadenas>*

Valida que sea 1, pero primero convierte los caracteres a algo manejable

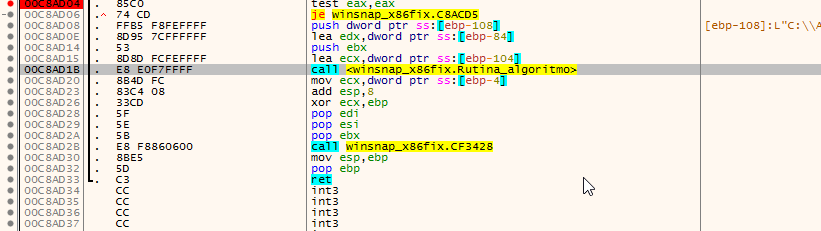


*<manejo de cadenas: son widechar ósea character>*

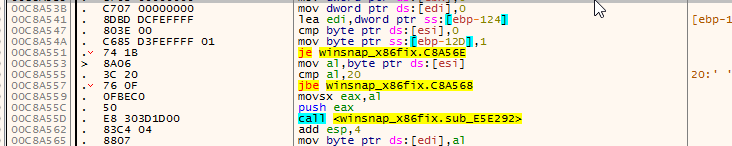
*Retomando la idea* Chr(66) = B

En una procesa el nombre y en otra el serial manipulando cada cadena como widechar, luego de ello comienza la verdadera rutina del serial

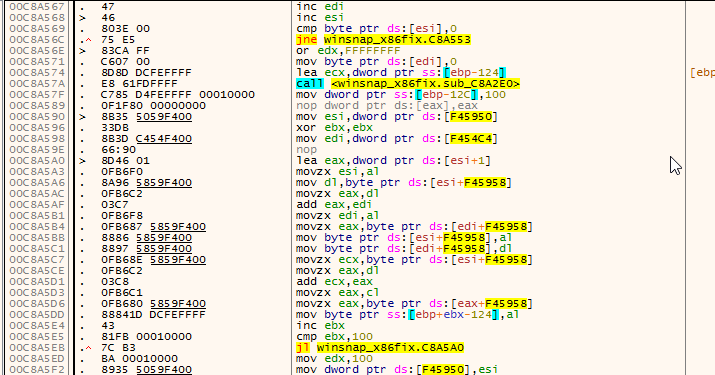
***Rutina\_Algoritmo:***

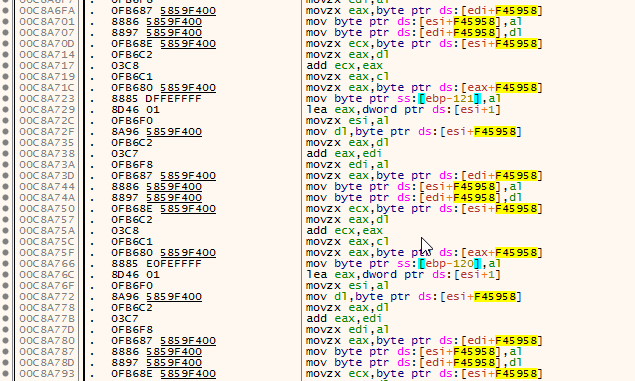


En esta rutina de algoritmo verifica los espacios y más. Si el camino es correcto va sumando o no el valor a eax/edi según el ciclo

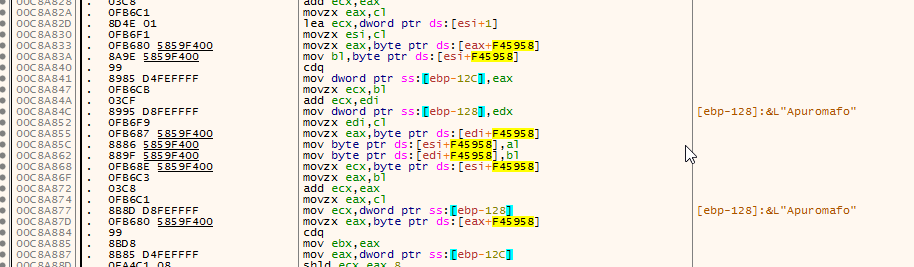


Verifica mediante OR+lea+add+xor los valores del serial según un alfabeto personalizado

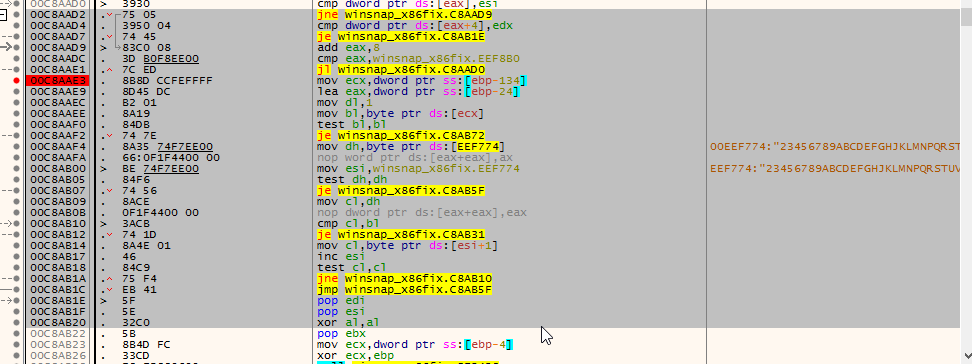




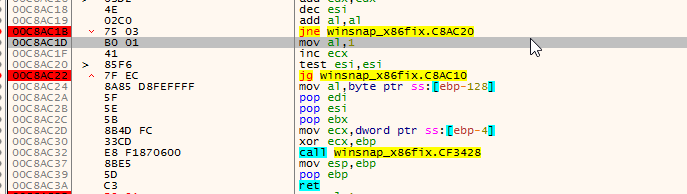
Conforme avanza aparecen divisiones (cdq) y



Va procesando el nombre en palabras simples si el serial es hola, ahora verifica si funciona bien Ahola y así otros procesos (para verificar que no valida cualquier serial)

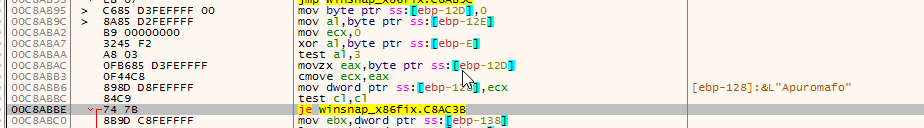


Y luego recién a procesar el serial, Un buen camino es que llegara a ser 1

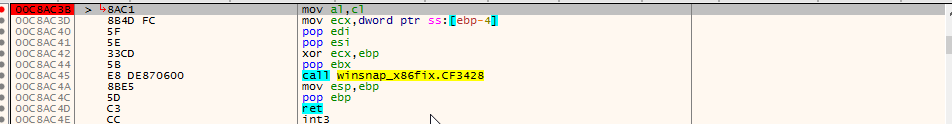


Pero como sabemos con un serial creado inválido imposible que ello pase

Asi que la realidad es que pasará por aquí donde refiere test cl,cl

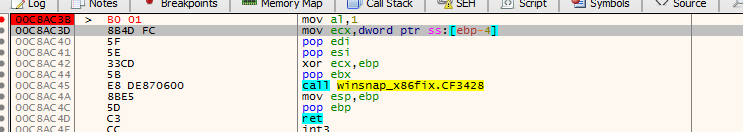


Y en ese salto, saltará al mov al,cl (ambos son 0)



En resumen el final del algoritmo se resumen en el manejo de eax sobre ecx, si ecx es 1, entonces moverá eax el valor de 1, y así vamos viendo hacia atrás.

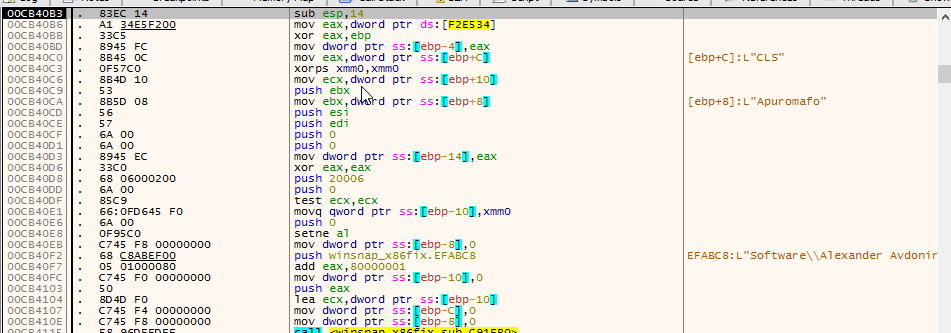
Solución para lograr algo registrado es que el valor de eax sea 1 y con ello cada variable al pasar en esta validación saldrá correcta.



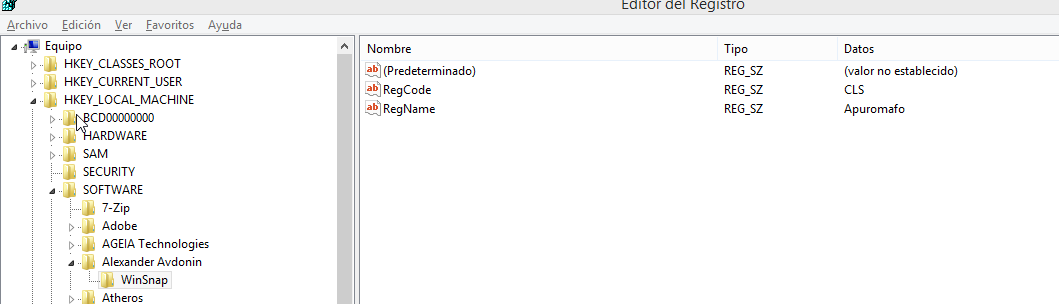
Listo, con eso ya será eax 1, y por lo tanto la validación del serial será el indicado.

Analisis de datos por Regedit.

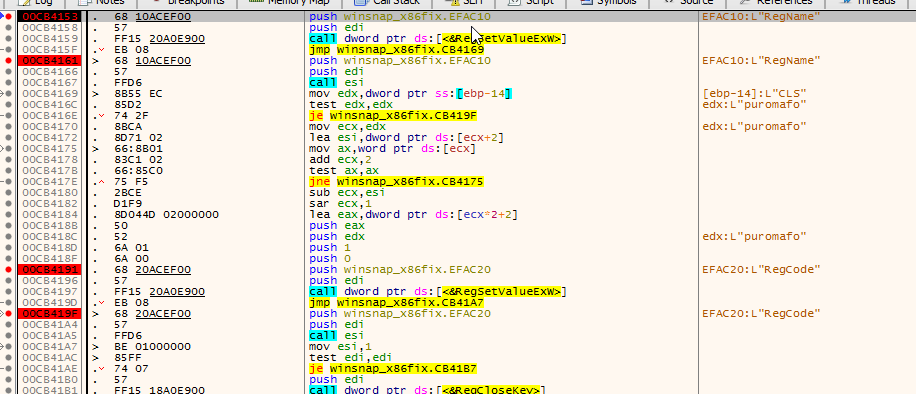
En la rutina de Regedit se aprecia que según el permiso escribirá el valor, recordemos que exigía admin, esto es para escribir en la rama de HKLM (h\_key\_local machine), solo los datos de registro, respecto a los otros, se suelen almacenar en hkey\_current\_user



El valor será escrito sobre el nombre del autor de la firma digital (alexander avdonin)



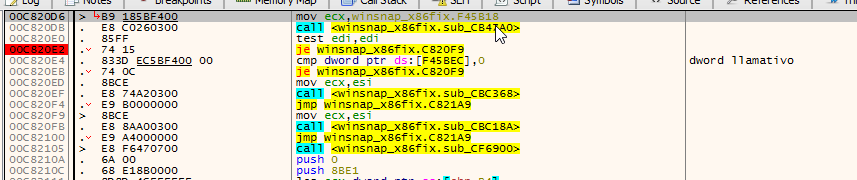
En HKEY\_LOCAL\_MACHINE como RegCode y Regname los cuales serán invocados y verificados cuando inicie el programa otra vez



Y establecido estos valores llegará a la mov edi,1 y a un buen camino

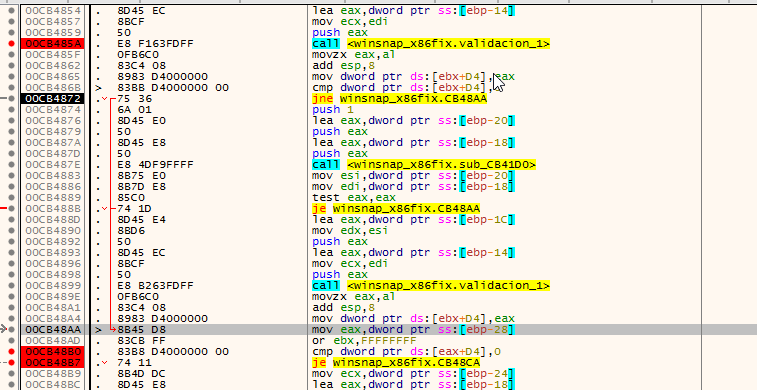


Superior al mensaje de key invalid que habíamos conocido hace poco y que hace luego una serie de validaciones adicionales que tiene unos saltos entre medio, si edi es 1 entonces el primer salto no irá, entonces comparará con un dword.



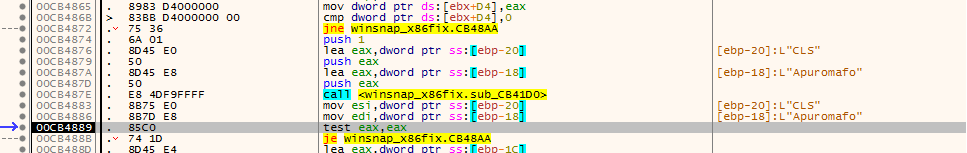
Y ahí viene el primer dword llamativo

Veamos el primer call

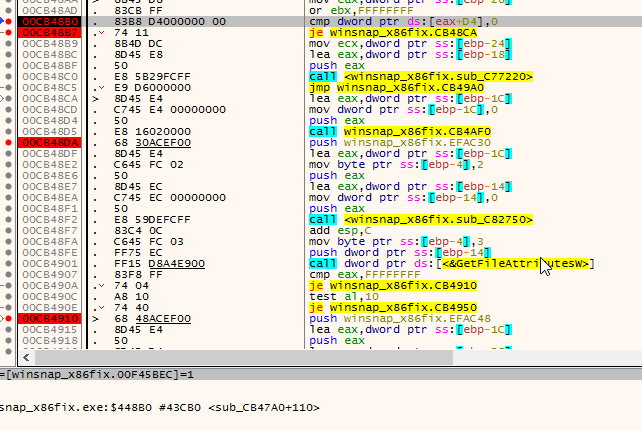


Refiere varias llamadas a los ebx+d4 que son cerca de las validaciones, entonces hay verificación del stack escrito con 1, y tendremos que tener cuidado que dword está apuntando.

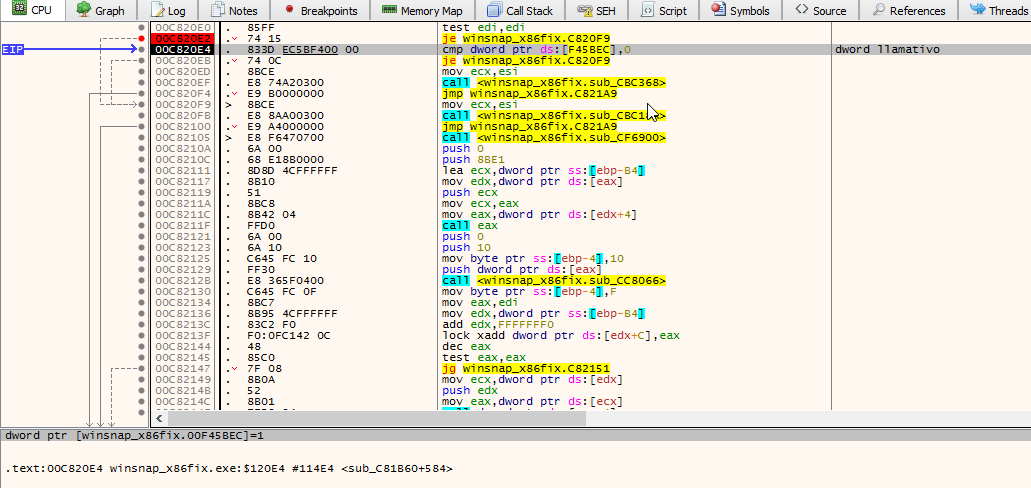
A primera vista rescata usuario+serial

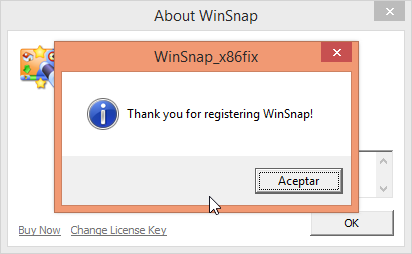


Como podemos apreciar como hemos parchado el valor a 1 en eax



Automáticamente se ha seteado en 1 ese valor, este valor al parecer es una constante conocida en otros escritos como is\_registred, es una variable 0/1 que determina si está registrado o no.

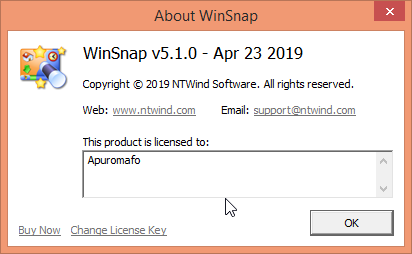




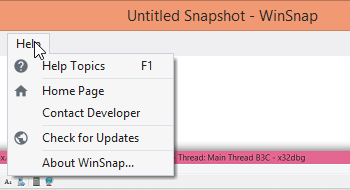
y nos refiere gracias por registrar winSnap

Ahora bien, el botón de enter license , se cambia a change license (una validación del dword)

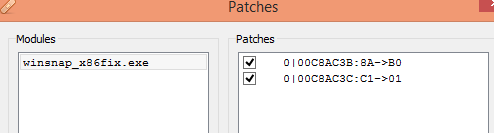
Y ahora cambio de no registrado a Licenciado a:



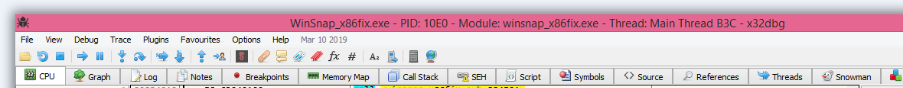
El menú se transforma (también se aprecia el dword que maneja tanto título y largo del menú)



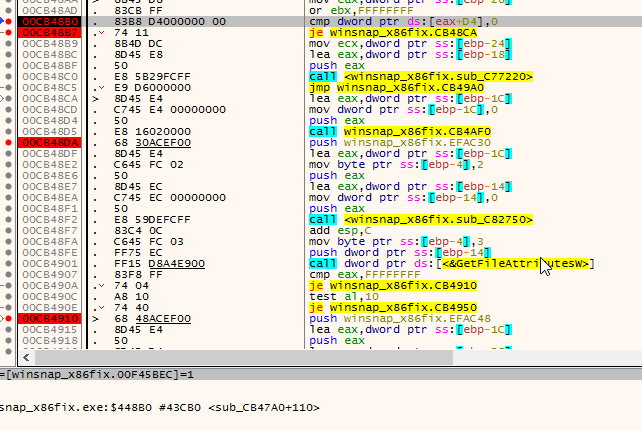
No hay unregistred



Guardo la imagen en escritorio para probar si hay watermark

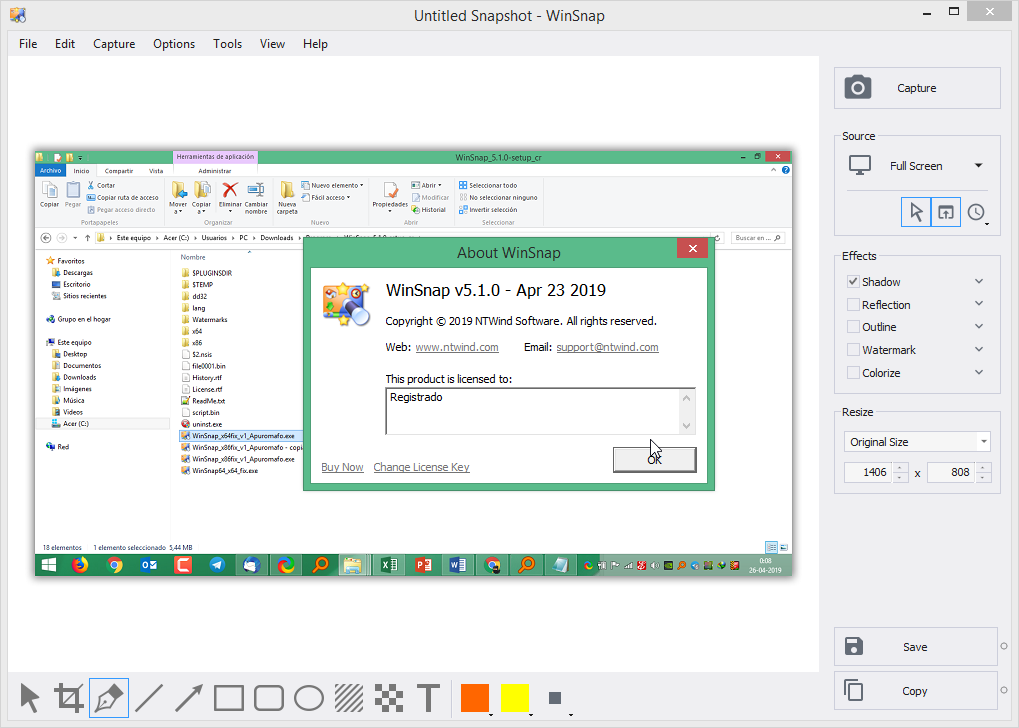


Sorpresa, no hay watermark en la imagen capturado sin grid y tampoco pasa por los timers que se apreciaban, para verificar el recorrido he colocado bp en el dword antes descrito y en la constante



En esa comparación con 1, debe ser siempre 1 para ser full, siendo 0 es demo, ya verificaré en el x64 si es similar.

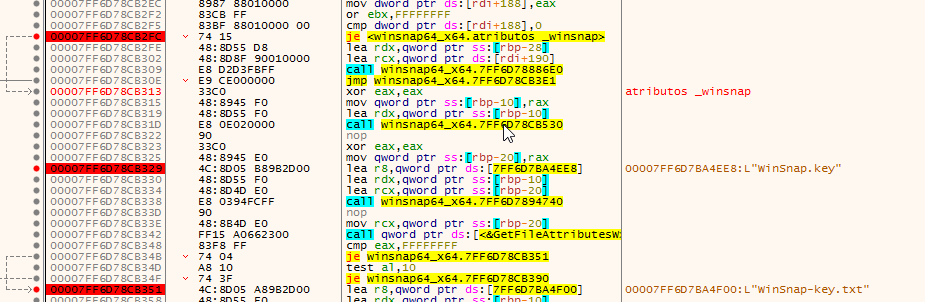
Resultado al estar en 1 el is\_registred (dword):



Queda mas que claro que el programa es funcional y está registrado y tenemos solucionado la primera depuración en x86.

# DEPURANDO X64

Usando la misma estrategia, vamos en búsqueda de la .key, solo debemos catalogar lo que encontremos (atributos winsnap, borra licencia, etc)

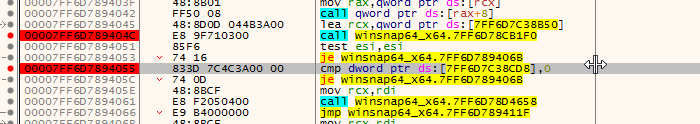


Luego buscamos la llamada



En este caso son 2

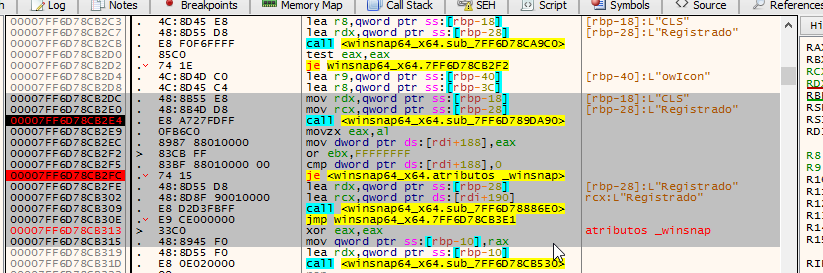
Veamos una comparación con esi y luego un dword, posiblemente cambia el registro pero la idea no.



El Dword que hablé en el pasado, también existe en este?

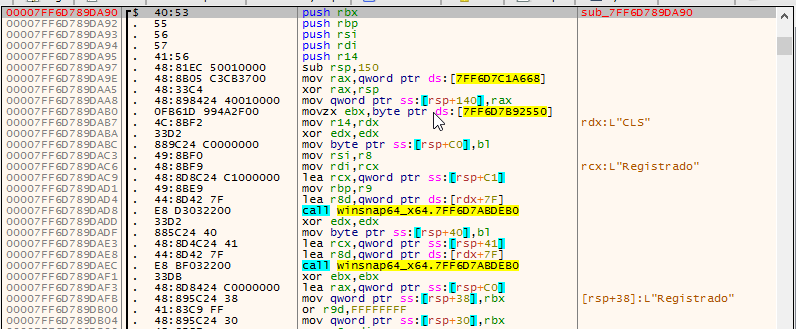
Veamos, la validación anterior era un valor que comparaba eax con otro valor y de ahí validaba en x64 es un poco mas grande, pero debería ser el mismo principio.

Veamos estábamos en atributos

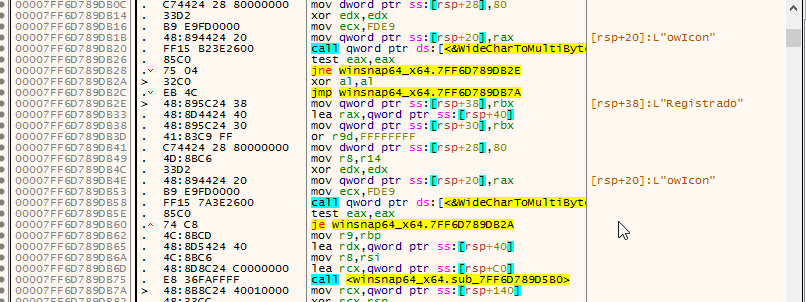


De los atributos era llamado 0 y 1 en el valor de eax y había un dword, ese call debería validarse antes..

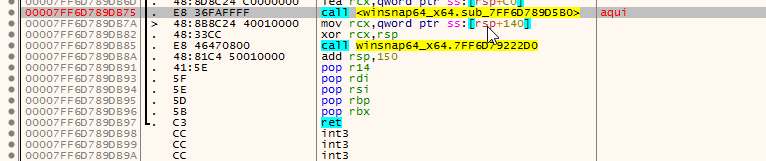
Verifico



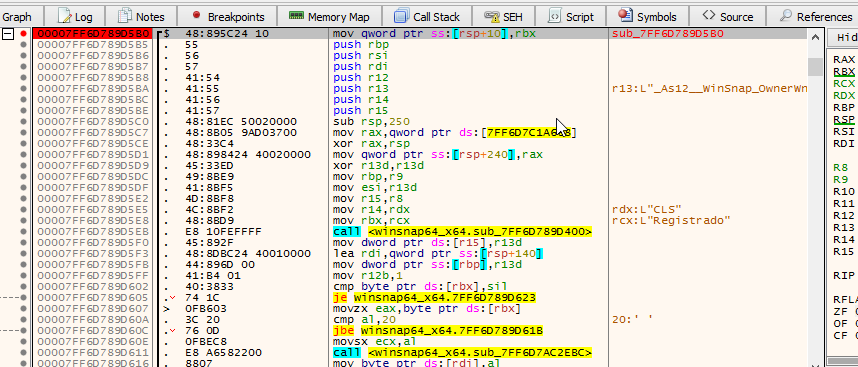
En la rutina debería verse un wide



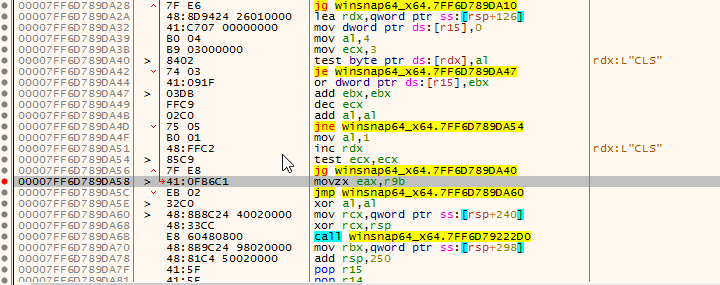
Si, es como antes, el penultimo call validaba el serial



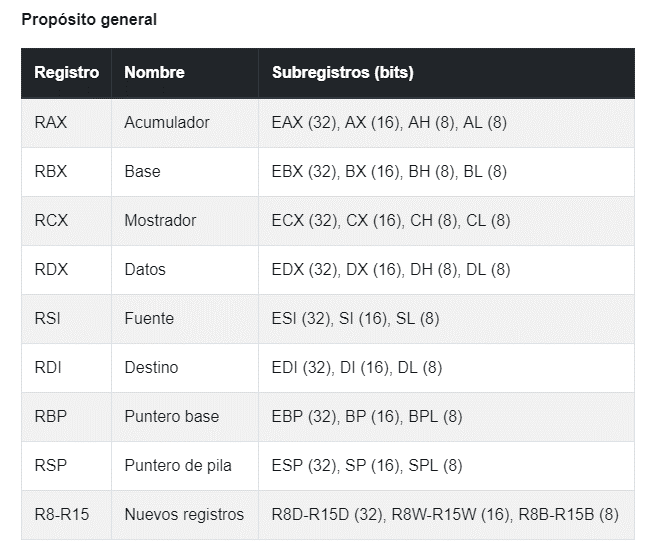
De esta rutina vienen las validaciones de espacio y más



Debido a la poca validación de CLS (espacio, guion etc)



Aquí como siempre es necesario aprender algo nuevo se lee r9b , que significa eso?

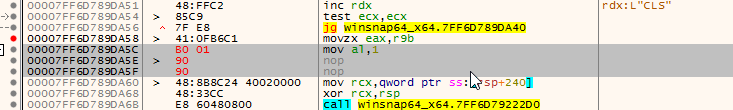


Fuente: <https://riptutorial.com/es/assembly/example/22706/x64-registros>

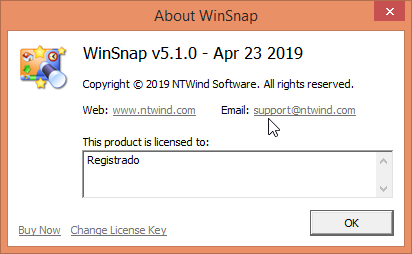
Es un registro para x64 que indica que es de 8 bytes, es como el equivalente cuando usábamos en eax el valor de al (8 bytes)

Llegaré al mov que se aprecia, este mov eax, debe estar en 1, por el tamaño usaré mov al,1 (recuerden que el registro es de 8 bytes, entonces el espacio es mínimo)

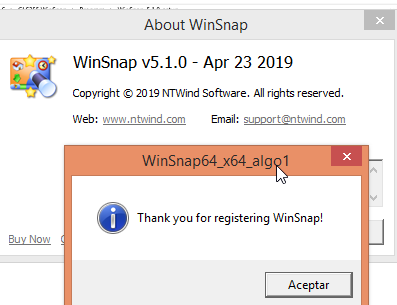
Parchamos



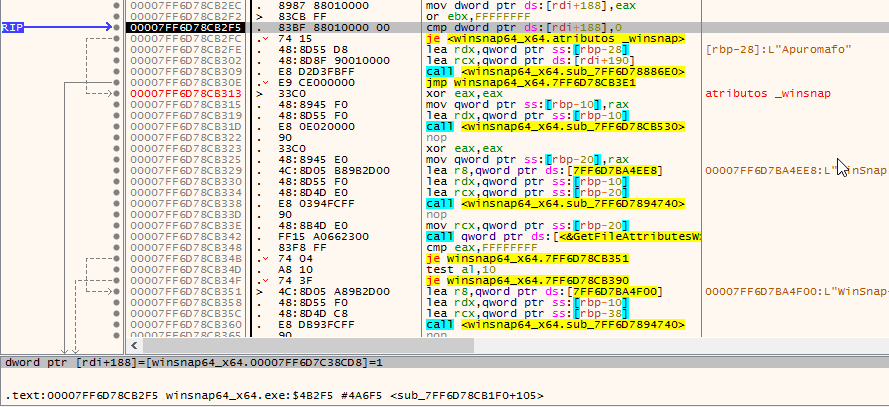
Y como se aprecia con 4 bytes guardo los cambios



Ya está registrado y acepta el serial como tal.

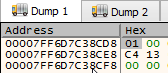


Ahora bien El dword que maneja en este x64 tambien en el lugar del atributo del .key

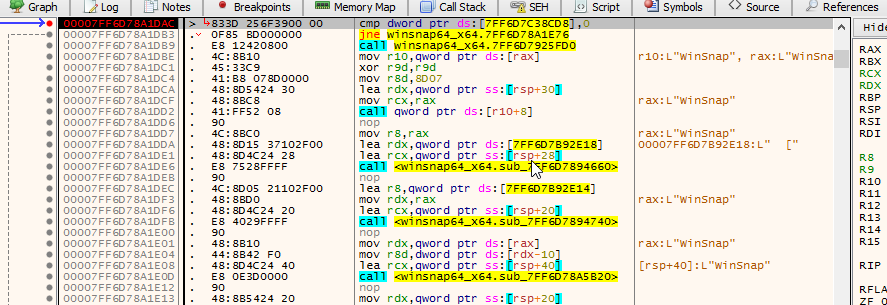


Compara el valor con 0/1 ahora hablemos de l lugar demo y como se puede rastrear, lo primero es buscar en dump el valor del rdi+188 que vemos con 1

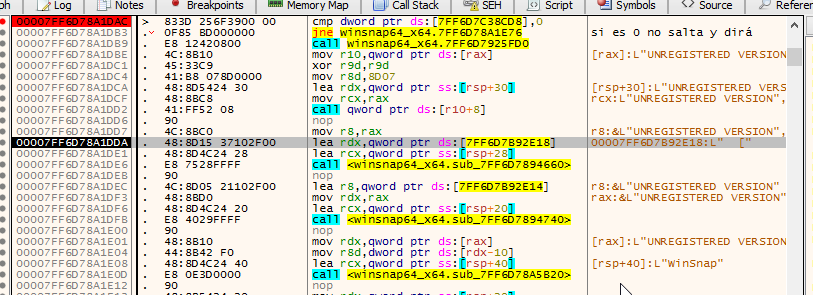
Esa dirección al buscarla

 coloco  y bp en todas, reinicio y vemos:

Este dword está asociado al título

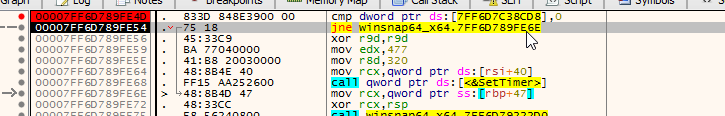


Que cuando se tracea es

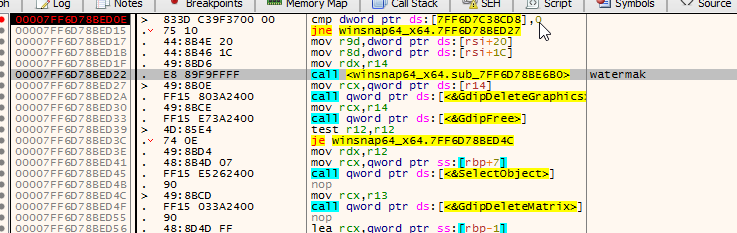


UNREGISTERED

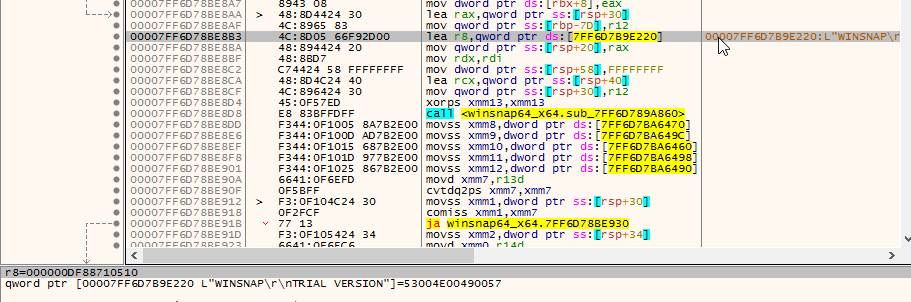
Está además asociado a un timer



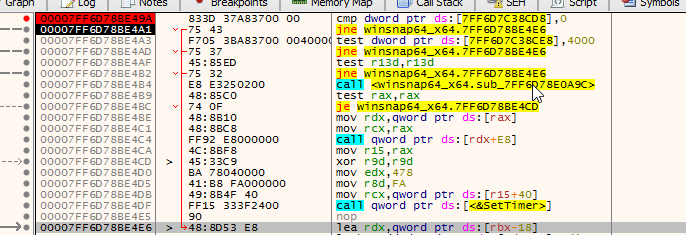
Al capturar el watermark



Que su string dice



Trial versión y otro timer



Así el demo es marcado y marcado con grid

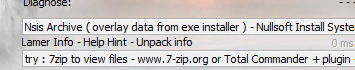


Sea como sea, al estar el dword como corresponde en 1 , todo lo anterior como demo no existe.

Espero les hubiera gustado el escrito.

Anexos:

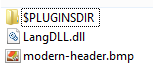
Cuando veníamos a depurar, exploré la carpeta con 7zip y había un archivo NSIS

 el exeinfope refiere usar 7zip

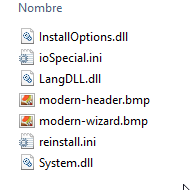
Si alguno es fanático de Python pueden usar de plantilla este link

<https://github.com/deepinstinct/NSISExtractor/blob/master/extract_ransom_from_nsis.py>

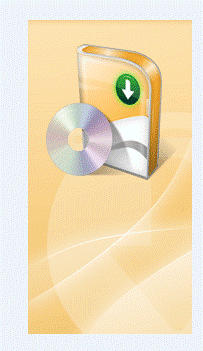
de lo contrario solo usar directamente el 7zip

y dentro del plugin dir  tiene las imágenes del header del setup

Sin embargo tienen además otros en el plugindir principal:



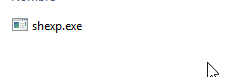
 modern header

modern wizzard

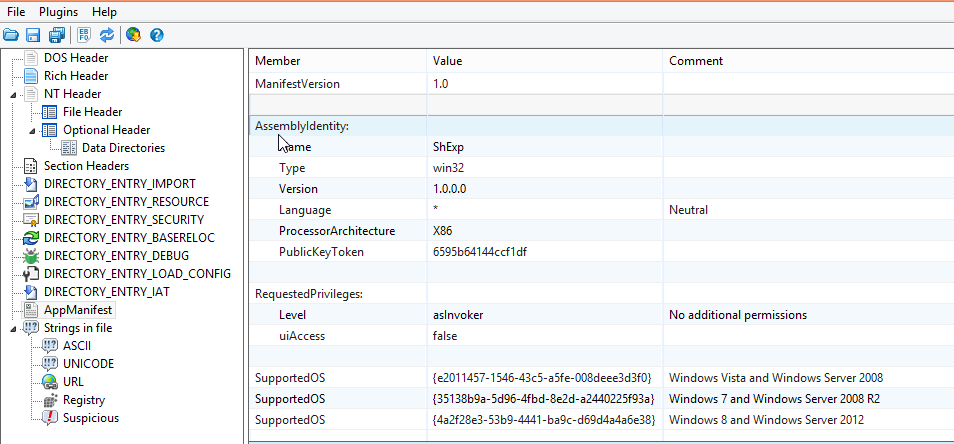
Las .ini son características de tamaño, y el system refiere algo de wsprint, asi que posiblemente maneje cadenas.

que referia en temporales tiene firma digital:





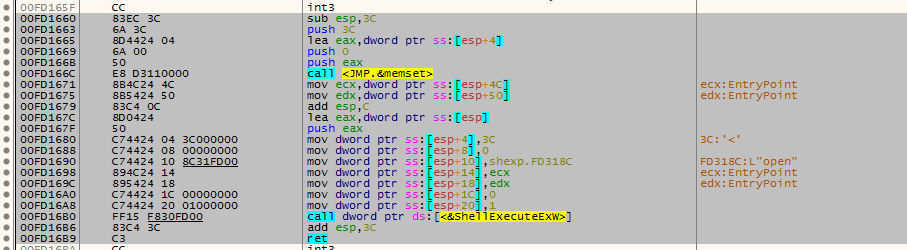
Verifico en PPEE.exe (puppy)



encontré un archivo que potencialmente es peligroso en el caso de cualquier setup descarga en temporal el archivo shexp.exe, el cual permite argumentos es más se puede ejecutar sin drama la calculadora (calc) y además cmd entre otros,al nivel de quien ejecuta el cual puede ser mal utilizado aquí un uso cualquiera (calc , cmd etc) obviamente esto no está si reinician el equipo (solo estará en temporal)



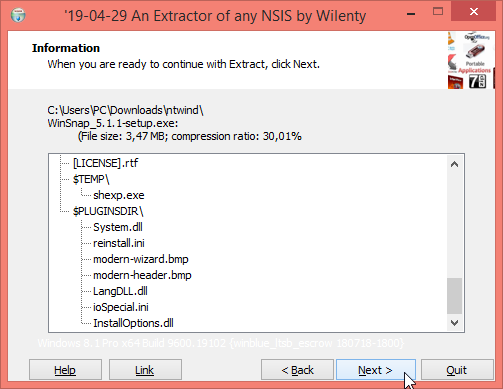
La porción de código reserva la memoria y luego ejecuta el argumento en open.



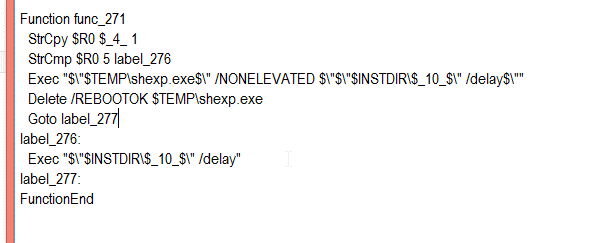
Si pueden ejecutar como administrador el setup, esta aplicación también tendrá que ser ejecutada como administrador , por lo tanto tocará investigar un poco mas al nsis

Hoy descargo y estamos ya con el winSnap 5.1.1 ha actualizado.

Luego de encontrar al indicado encuentro:



Al usar el extractor de wilenty, encontramos un archivo nsis completamente legible



El cual refiere que la ejecución de este es para el instalador, y luego lo borra al reiniciar..

En el script además podemos encontrar

|  |
| --- |
| ExecShell open http://www.ntwind.com/tutorials.html?tags=winsnap ; "open <http://www.ntwind.com/tutorials.html?tags=winsnap>" |

Y si existe el archivo .key que lo copie

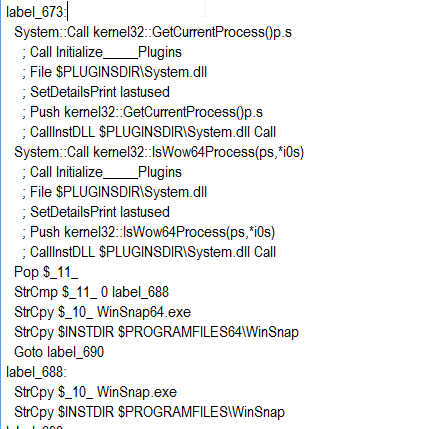
|  |
| --- |
| IfFileExists $EXEDIR\WinSnap.key 0 label\_320  CopyFiles /SILENT /FILESONLY $EXEDIR\WinSnap.key $INSTDIR ; $(LSTR\_7)$INSTDIR ; "Copy to "  label\_320:  IfFileExists $EXEDIR\WinSnap-key.txt 0 label\_322  CopyFiles /SILENT /FILESONLY $EXEDIR\WinSnap-key.txt $INSTDIR ; $(LSTR\_7)$INSTDIR ; "Copy to " |

La rama principal de regedit refiere

|  |
| --- |
| WriteRegStr HKLM "Software\Alexander Avdonin\WinSnap" "" $INSTDIR  WriteRegStr HKLM Software\Microsoft\Windows\CurrentVersion\Uninstall\WinSnap URLInfoAbout mailto:alexander@ntwind.com?subject=ASUNTO&body=MENSAJE |

¡Donde también es posible encontrar otra rama con una opción de enviarle un correo con asunto de la versión de winsnap y un mensaje de hi! (hola)

Y respecto a la detección del sistema operativo lo realiza mediante un label el cual usará iswow64 entre otros., usando el archivo system.dll



Palabras Finales

Me llevó mas tiempo del que pensaba, pero resultó revisarlo, dejo la instancia de revisar la nueva versión ,espero les hubiera gustado.

