



LOCK XLS 6.1

Proteccion sobre tus archivos de Excel by Apuromafo



21 DE MAYO DE 2018

CLS

Release:25/05/2018

INDICE

Contenido

INDICE	1
Introducción	2
Frase.	2
Herramientas usadas en el Escrito:	2
Analizando la víctima:.....	3
Comenzando a Modificar el exe	8
ISDebuggerPresent	8
X_days Version de prueba	9
Is_registred searching.....	10
Is_registred_found	11
Resultado_is_registred	13
Palabras Finales	25

Introducción

Programa	XML a CTPQi (marca registrada)
Descarga	(fue directo a mi privado en telegram desde "TITINO73")
Dificultad	Depende de quien lo mire.
Objetivo	Registrarnos
Información	Módulo protector para excel
Herramientas usadas	Excel, Notepad++, X64dbg , PID , 7zip
Fecha	21/05/2018
Cracker	Apuromafo

Frase.

El gran enemigo del conocimiento no es la ignorancia sino la ilusión de conocimiento.

S.Hawking

Herramientas usadas en el Escrito:

Herramienta	Descarga	Utilidad
Procesador de texto	<i>(está incluido con el suite de office)</i>	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.7zip.org/download.html	Descomprimir archivos
Uniextractor*	http://filehippo.com/es/download_universal_extractor/	Extractor de archivos
Notepad ++	https://notepad-plus-plus.org/	Editar archivos (no hexadecimal)
Cff Explorer	http://www.ntcore.com/files/ExplorerSuite.exe	Editor de recursos , explorador de pe header
Vba toolkit*.		
Reset VBA Password*		

* Herramienta opcional

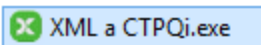
Historia :

Hola, luego de un tiempo me han compartido un archivo en Excel que está en formato .exe, al explorar encuentro novedad que tiene algunos errores al iniciar y al tener macros, también llama mi curiosidad, entonces iremos por él, coloquen atención que esto no está muy común de ver todos los días

Cuando queremos proteger un archivo Excel, que tiene macros buscamos por la red herramientas que nos permitan proteger la gran mayoría del código, en el caso puntual de este autor, ha colocado una segunda validación interna, el cual además permite asegurarse que si vencen el protector (compran) envíen un mail o mensaje para que deban activar el programa con un ID único y asegurar al 100% algo mas seguro bueno, con eso en mente, comencemos a ver este programa y Excel en general.

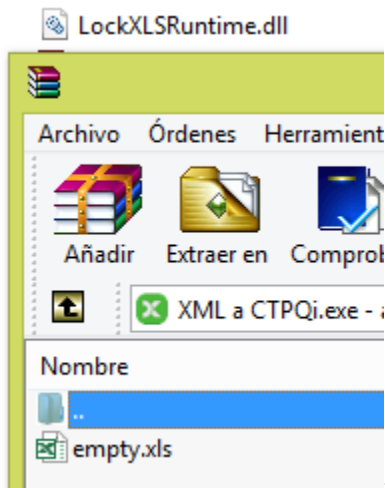
Analizando la víctima:

Al ver el archivo .exe no tiene ninguna protección visible, pero tiene varios módulos en su interior



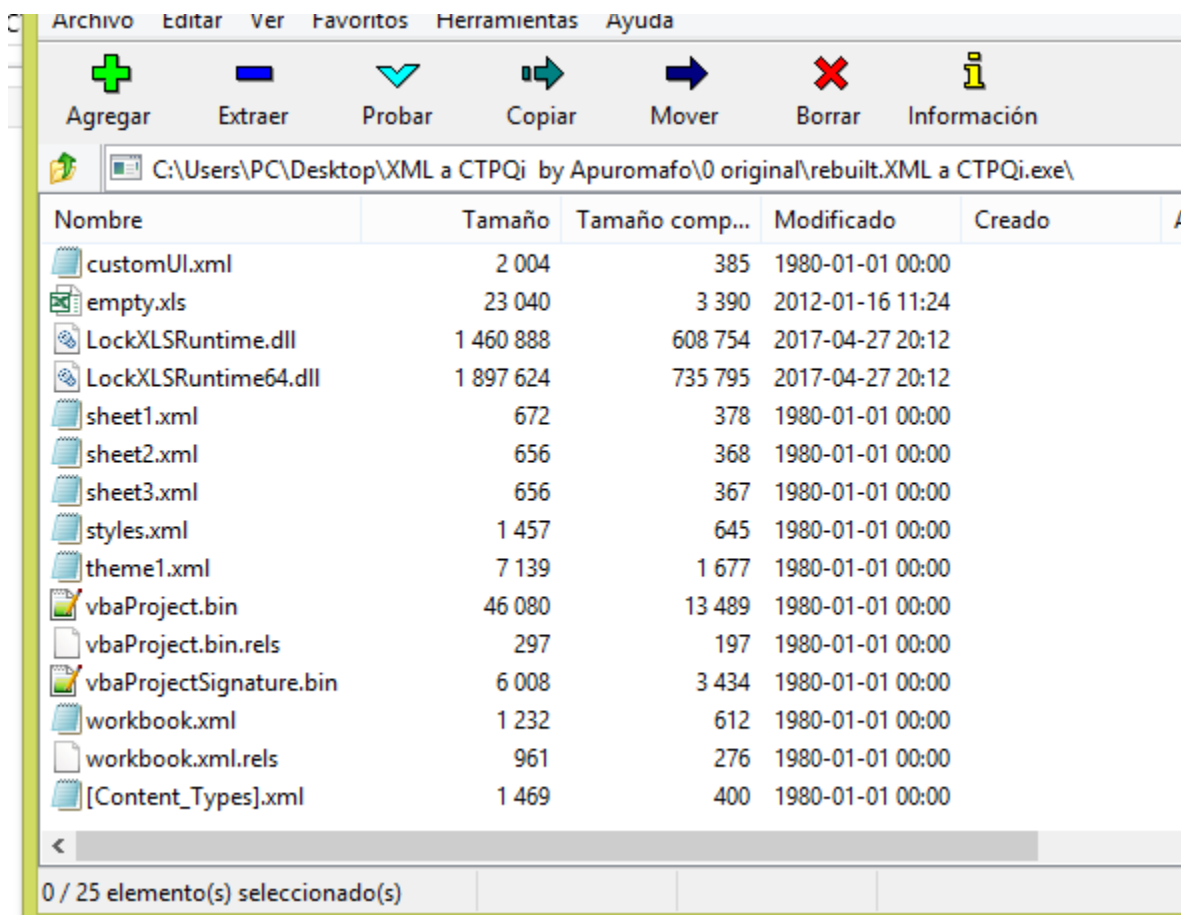
<original.exe>

Si uso 7zip o winrar obtengo en este exe como algún sfx 2 archivos a simple vista inocentes LockXLSRuntime.dll desde 7zip y empty.xls desde winrar



<vista en winrar>

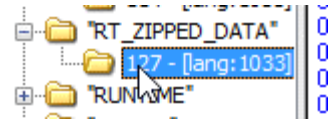
Si coloco en winrar reparar archivo como si fuere zip
Obtengo un archivo bastante llamativo para ser visto en 7zip:



<viendo el rebuilt en 7zip>

Hasta aquí no hemos explorado el programa , pero de seguro nos estamos acercando a darnos cuenta de lo que tenemos, tenemos un Excel que maneja 2 runtimes, hay proyectos de vba y además hay xml de por medio en el tema visual, osea de seguro no nos dejarán guardar nuestro archivo como corresponde.

Si queremos analizar esos .bin
Veremos que hay un modulo setup



Comenzando a ver los recursos podemos apreciar la extensión

El archivo rebuilded que hace poco tenemos ha sido parcialmente desde este recurso

Y además hay la existencia de un xla y xlam

Offset	U	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
00000000	50	4B	03	04	14	00	06	00	08	00	00	00	21	00	2F	78	PK /x
00000010	8E	41	90	01	00	00	BD	05	00	00	13	00	08	02	5B	43	!A [C
00000020	6F	6E	74	65	6E	74	5F	54	79	70	65	73	5D	2E	78	6D	ontent_Types].xm
00000030	6C	20	A2	04	02	28	A0	00	02	00	00	00	00	00	00	00	l.. (.. . . .
00000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

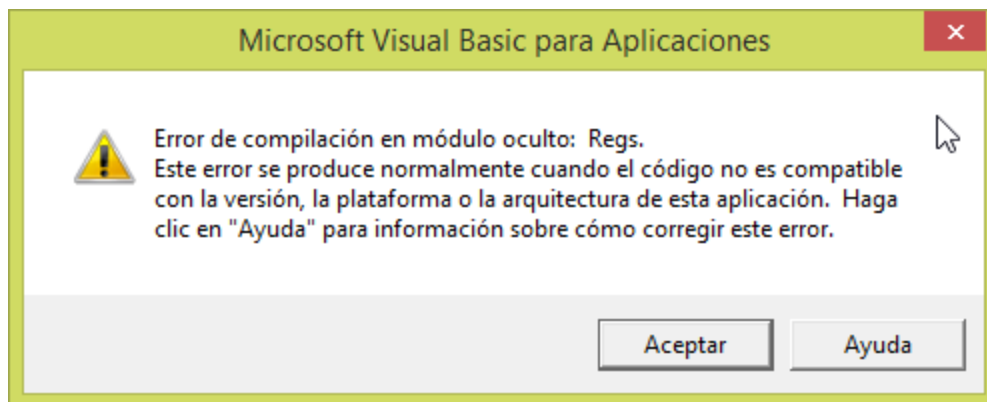
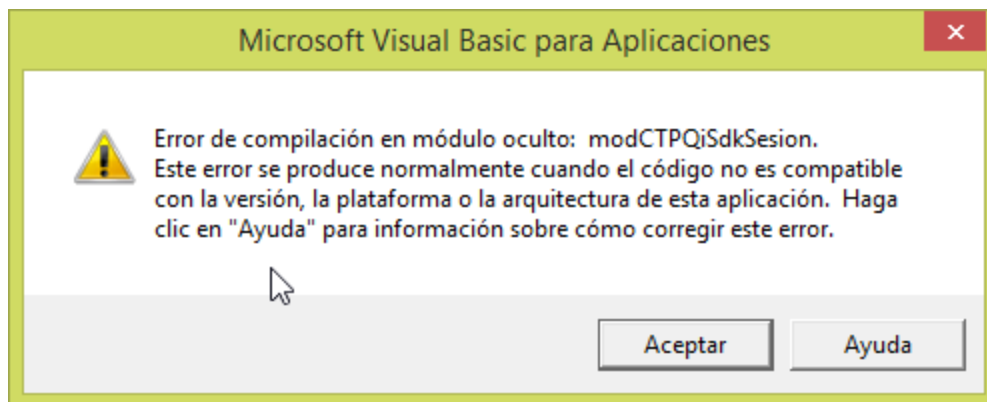
Del cual han salido los archivos xml que hemos visto hace poco

El archivo xla y xlam son algún pase directo a ejecutar el stub interior que posee el archivo, posiblemente cifrado o quien sabe que algoritmo, pero el tema es que tenemos esta realidad EXE (valida según lo configurado) ejecutará el xla/xlam para ejecutar las macros en nuestro archivo real.xlsm hasta ahí suena un poco mal porque cualquier cambio será manipulado desde dos entornos que desconocemos, entonces tendremos que ver como vencer aquello.

la gran mayoría de las herramientas automáticas ayudan a bypasear las claves de las macros (sobre el primer Excel), pero aquí habrán de seguro una ejecución adicional, entonces tendremos que retomar el escrito de bypass de macros (de Excel) y volver a tenerlo en consideración

Para no agotar la lectura basta el attach al Excel con permisos de admin (cuando estemos con el visor de proyectos de visual basic alt+f11) y buscar cuando la clave del proyecto es invalida y retroceder hasta el primer dialogo que permita cerrar luego del algoritmo de cifrado criptográfico (suele ser un salto) y esa dll no siempre está presente pero asocia al nombre de vb7.dll en mi pc , bueno para no cansar

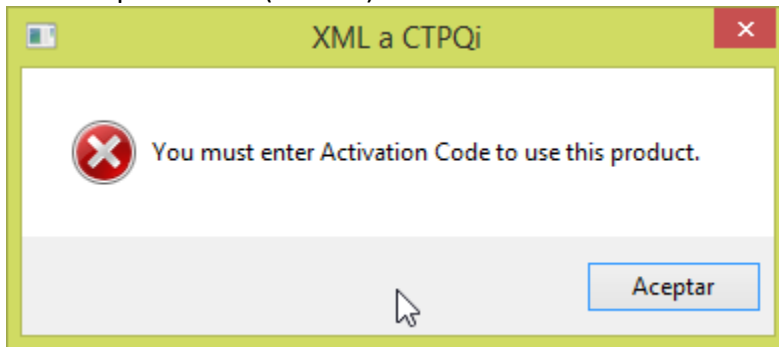
Comencemos viendo lo que tenemos



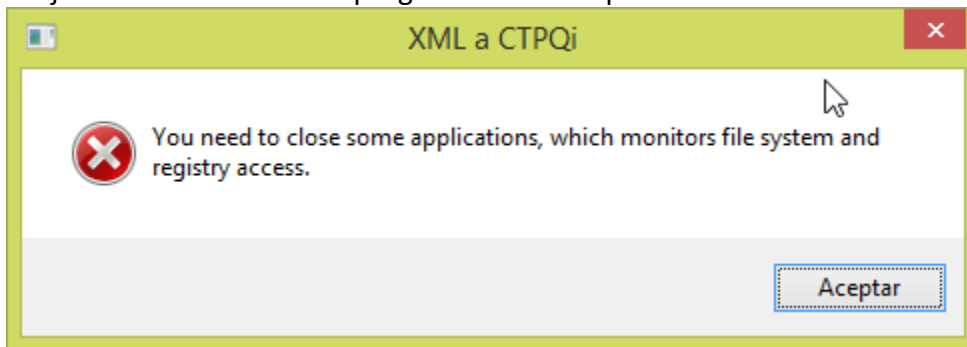
Luego de expirarlo, podemos comenzar a ver que tanto logramos habilitar

Comenzando a Modificar el exe

Primera protección (del exe)



Si ejecuto directamente el programa con el depurador me muestra

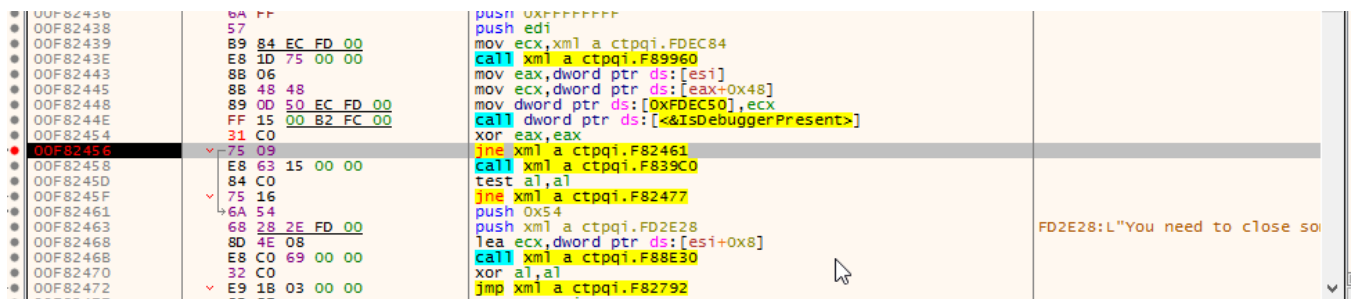


<depurador detectado>

Asi que comenzamos a parchar

ISDebuggerPresent

- 1) No sea alarmante si aparece is_debuggerpresent (comparación con eax a "xor eax,eax")



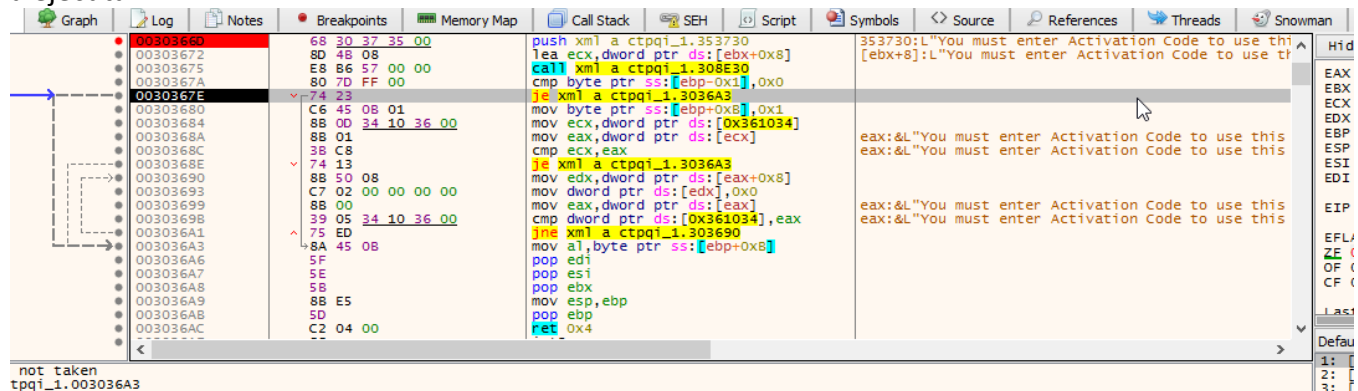
X_days Version de prueba

Expired v/s uso normal el programa tiene una cantidad de días y de uso

Observen como tiene 7 dias.



En el tiempo trial y normal el programa ejecuta, pero expirado no, entonces la idea es que siempre irá a ejecutar



Podemos apreciar como ese salto no es ejecutado en tiempo trial y en expired salta directo, asi que nopeamos y tenemos un programa que ejecutará aun en expired pero esta sección será accedida de otra forma si el programa es registrado asi que debemos tener cuidado en eax y ebx, asumo que será 1 pero primero veamos hasta donde llegamos...

El programa valida diferentes opciones desde los recursos siendo muchos validados como saltos
Call stub

Je chico/bueno-malo

Respecto a la cantidad de días es manejado en la variable //expires

Cmp edi,eax aquí apreciamos como muestra que expirará en x tiempo o 10 sesiones, osea tenemos 10 oportunidades , y ahora observen como se aprecian 386 días, ven, se puede modificar con los valores de ecx, eax, edi

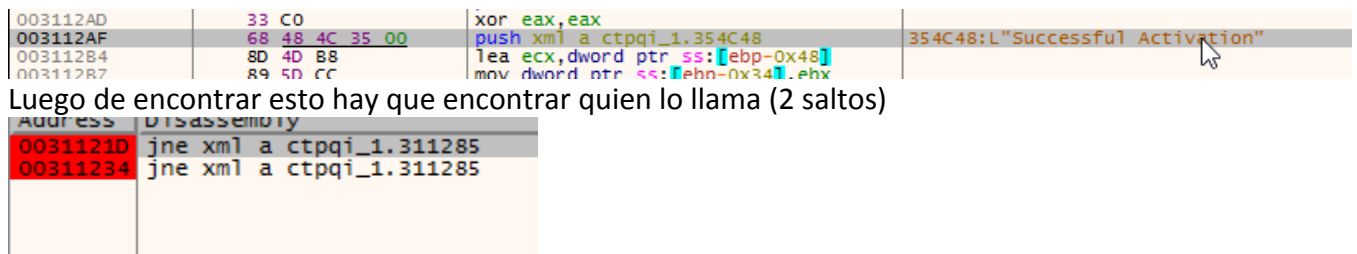


Ahora la idea es forzar el registrado asi que luego de entender como funciona el stub hay que llegar ahí de alguna forma

Seguimos revisando y ahora tenemos que lograr vencer ante el código, obtener un camino válido (entre los saltos)

Is_registed searching

Encuentro una frase de activation que me dice que esta en lo correcto:



2 saltos llevan ahí

Antes de ello hay un comienzo de rutina que a su vez es llamado

003111ED	C3	ret	
003111EE	BB 07 00 00 00	mov ebx,0x7	
003111F3	33 D2	xor edx,edx	
003111F5	89 5D E8	mov dword ptr ss:[ebp-0x18],ebx	
003111F8	66 89 55 D4	mov word ptr ss:[ebp-0x2C],dx	
003111FC	C6 45 FC 0F	mov byte ptr ss:[ebp-0x4],0xF	
00311200	38 96 BD 01 00 00	cmp byte ptr ds:[esi+0x18D],dl	
00311206	74 17	je xml a ctpqi_1.31121F	
00311208	6A FF	push 0xFFFFFFFF	
0031120A	57	push edi	
0031120B	8D 86 58 03 00 00	lea eax,dword ptr ds:[esi+0x358]	
00311211	50	push eax	
00311212	8D 4D D4	lea ecx,dword ptr ss:[ebp-0x2C]	
00311215	E8 D6 8A FF FF	call xml a ctpqi_1.309CF0	
0031121A	39 7D E4	cmp dword ptr ss:[ebp-0x1C],edi	
0031121D <xml a	75 66	jne xml a ctpqi_1.311285	sufec
0031121F	6A FF	push 0xFFFFFFFF	
00311221	57	push edi	
00311222	81 C6 3C 03 00 00	add esi,0x33C	
00311228	5E	push esi	

Para llegar ahí antes debe haber pasado validaciones

00310FCA	84 C0	test al,al	
00310FCC	74 13	je xml a ctpqi_1.310FE1	licencia reee
00310FCE	6A 2E	push 0x2E	
00310FDD	68 B0 4B 35 00	push xml a ctpqi_1.3548B0	3548B0:L"This Activation Code has been already entered.
00310FDE	8D 4D 9C	lea ecx,dword ptr ss:[ebp-0x64]	
00310FDF	E8 53 7E FF FF	call xml a ctpqi_1.308E30	
00310FDD	32 DB	xor bl,bl	
00310FDF	EB 0F	jmp xml a ctpqi_1.310FF0	

La licencia no se puede repetir

Y un poco mas arriba se observa un salto decisivo a lo incorrecto (esa call valida el tiempo, es posible que pueda durar la licencia 1 año solamente, para parcharlo y forzar basta colocar nop al salto.

Is_registered_found

00310F25	8D 8D 30 FF FF FF	lea ecx,dword ptr ss:[ebp-0xD0]	
00310F2B	E8 50 ED 00 00	call xml a ctpqi_1.31FC80	
00310F30	8A D8	mov bl,al	
00310F32	84 DB	test bl,bl	
00310F34	0F 84 A9 01 00 00	je xml a ctpqi_1.3110E3	incorrect
00310F3A	89 7D E0	mov dword ptr ss:[ebp-0x20],edi	
00310F3D	89 7D E4	mov dword ptr ss:[ebp-0x1C],edi	
00310F40	89 7D E8	mov dword ptr ss:[ebp-0x18],edi	
00310F43	89 7D 8C	mov dword ptr ss:[ebp-0x74],edi	
00310F46	89 7D 90	mov dword ptr ss:[ebp-0x70],edi	

Por lo cual dentro del call se validará mi licencia correcta (según el valor de bl se juega todo)

Dentro de ese call se aprecia

0031FC80	68 F0 00 00 00	push 0xF0	
0031FC85	B8 3C 77 34 00	mov eax,xml a ctpqi_1.34773C	
0031FC8A	E8 71 C1 00 00	call xml a ctpqi_1.328E00	
0031FC8F	88 75 08	mov esi,dword ptr ss:[ebp+0x8]	
0031FC92	88 45 0C	mov eax,dword ptr ss:[ebp+0xC]	
0031FC95	88 7D 10	mov edi,dword ptr ss:[ebp+0x10]	
0031FC98	89 8D 50 FF FF FF	mov dword ptr ss:[ebp-0x80],ecx	
0031FC9E	88 4D 14	mov ecx,dword ptr ss:[ebp+0x14]	
0031FCA1	89 8D 4C FF FF FF	mov dword ptr ss:[ebp-0x84],ecx	
0031FCA7	88 4D 18	mov ecx,dword ptr ss:[ebp+0x18]	
0031FCAA	89 8D 48 FF FF FF	mov dword ptr ss:[ebp-0x88],ecx	
0031FCB0	88 4D 1C	mov ecx,dword ptr ss:[ebp+0x1C]	
0031FCB3	89 8D 04 FF FF FF	mov dword ptr ss:[ebp-0xFC],ecx	
0031FCB9	88 4D 20	mov ecx,dword ptr ss:[ebp+0x20]	
0031FCBC	33 DB	xor ebx,ebx	
0031FCBE	89 85 54 FF FF FF	mov dword ptr ss:[ebp-0xAC],eax	
0031FCC4	89 8D 58 FF FF FF	mov dword ptr ss:[ebp-0xA8],ecx	
0031FCCA	3B F3	cmp esi,ebx	
0031FCCC	75 20	jne xml a ctpqi_1.31FCEE	
0031FCCE	BE B4 CF 34 00	mov esi,xml a ctpqi_1.34CFB4	34CFB4:L"Product Code is missed"
0031FCD3	56	push esi	

Vamos entonces desde 0 probando una y otra vez cual salto será decisivo

Ese jne no debe saltar , y ese je debe saltar.

0031FFE4	E8 13 D7 00 00	call xml a ctpqi_1.3206FC	
0031FFE9	83 C4 0C	add esp,0xC	
0031FFEC	85 C0	test eax,eax	
0031FFEE	0F 85 38 01 00 00	jne xml a ctpqi_1.32012F	
0031FFF4	38 9D 5F FF FF FF	cmp byte ptr ss:[ebp-0xA1],b1	
0031FFFA	0F 84 83 00 00 00	je xml a ctpqi_1.320083	
00320000	BE 58 CE 34 00	mov esi,xml a ctpqi_1.34CE58	34CE58:L"Activation Code was generated by obsolete vers
00320005	56	push esi	
00320006	E8 06 C2 00 00	call xml a ctpqi_1.32C211	

En este lugar este salto es dado y lleva a chico malo, así que debo anularlo

En resumen el al no es 1 ni 2, pero me conviene que sea alguno de ellos para mover el registro A1 a 1, entonces basta que al sea 2 y el programa validará correctamente o sea en 3200cA me encargo con un nop

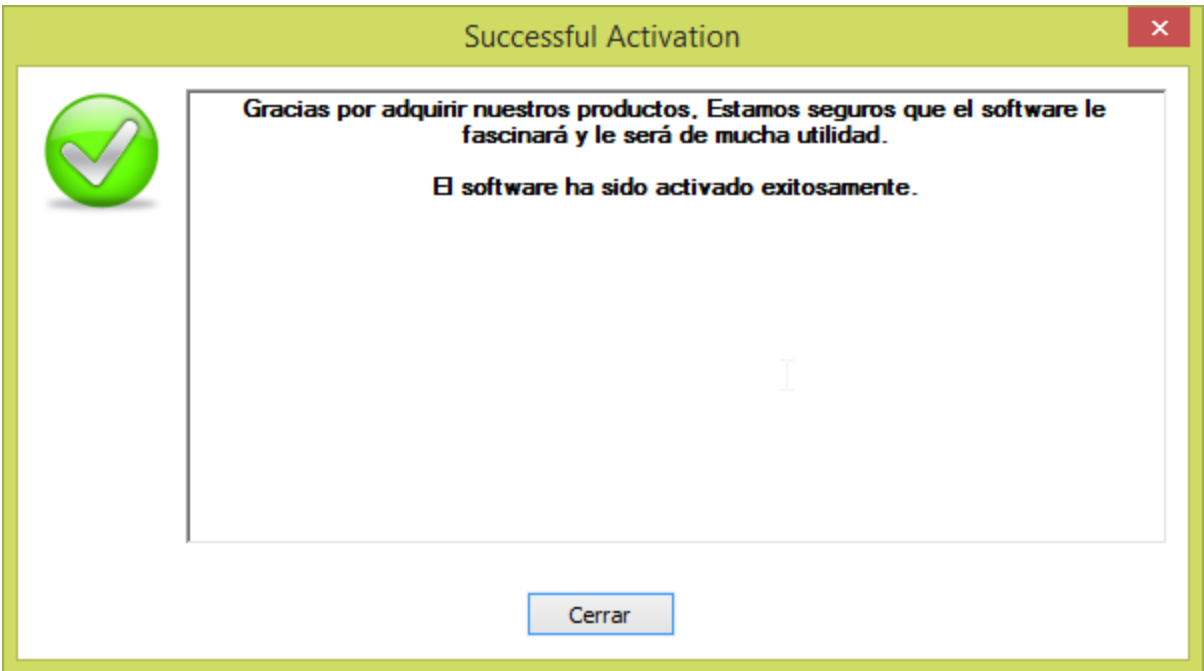
00320083	8B 85 38 FF FF FF	mov eax,dword ptr ss:[ebp-0xC8]	
00320089	8A 00	mov al,byte ptr ds:[eax]	
0032008B	3C 01	cmp al,0x1	
0032008D	75 39	jne xml a ctpqi_1.3200C8	
0032008F	8B 8D 50 FF FF FF	mov ecx,dword ptr ss:[ebp-0xB0]	
00320095	8D 85 28 FF FF FF	lea eax,dword ptr ss:[ebp-0xD8]	
00320098	50	push eax	
0032009C	83 C1 1C	add ecx,0x1C	
0032009F	E8 7C 3F FE FF	call xml a ctpqi_1.304020	
003200A4	8B 85 38 FF FF FF	mov eax,dword ptr ss:[ebp-0xC8]	
003200AA	6A 04	push 0x4	
003200AC	40	inc eax	
003200AD	50	push eax	
003200AE	FF B5 4C FF FF FF	push dword ptr ss:[ebp-0xB4]	
003200B4	E8 F7 B2 00 00	call xml a ctpqi_1.328380	
003200B9	83 C4 0C	add esp,0xC	
003200BC	C6 85 5F FF FF FF	mov byte ptr ss:[ebp-0xA1],0x1	
003200C3	E9 57 FF FF FF	jmp xml a ctpqi_1.32001F	
003200C8	3C 02	cmp al,0x2	
003200CA	75 5C	jne xml a ctpqi_1.320128	
003200CC	8B 8D 50 FF FF FF	mov ecx,dword ptr ss:[ebp-0xB0]	
003200D2	8D 85 28 FF FF FF	lea eax,dword ptr ss:[ebp-0xD8]	
003200D8	50	push eax	
003200D9	83 C1 1C	add ecx,0x1C	
003200DC	E8 3F 3F FE FF	call xml a ctpqi_1.304020	
003200E1	8B 85 38 FF FF FF	mov eax,dword ptr ss:[ebp-0xC8]	

De este trozo de memoria cuando es 2 el programa ingresa el valor correcto y dado que está forzado con nop, seguirá trabajando, en algunos casos si tardamos mucho dará excepción así que debe ser rápido el proceso. Forzando el valor de bl en 1 (para el que buscábamos en la primera validación) Fuerzo luego fuerza todos los valores que necesitábamos

Log	Notes	Breakpoints	Memory Map	Call Stack	SEH	Script	Symbols	Source	References	Threads	Snowman
00310FCC	EB 13						jmp xml a ctpqi_1.310FE1				
00310FCE	6A 2E						push 0x2E				
00310FDD	68 80 4B 35 00						push xml a ctpqi_1.3548B0				
00310FDE	8D 4D 9C						lea ecx,dword ptr ss:[ebp-0x64]				
00310FDF	E8 53 7E FF FF						call xml a ctpqi_1.308E30				
00310FDF	32 DB						xor bl,bl				
00310FDF	EB 0F						jmp xml a ctpqi_1.310FF0				
00310FE1	8D 45 E0						lea eax,dword ptr ss:[ebp-0x20]				
00310FE4	50						push eax				
00310FE5	8D 8E AC 01 00 00						lea ecx,dword ptr ds:[esi+0x1AC]				
00310FE8	E8 7E 17 01 00						call xml a ctpqi_1.32276E				
00310FF0	8D 8D 60 FF FF FF						lea ecx,dword ptr ss:[ebp-0xA0]				
00310FF6	C6 45 FC 0A						mov byte ptr ss:[ebp-0x4],0xA				

Resultando nuestro lugar de destino:

Resultado_is_registered



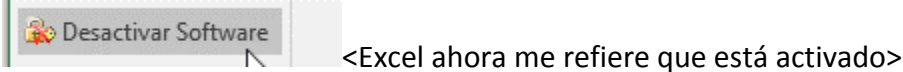
Y para finalizar pasa por el lugar correcto en caso x que la licencia no se duplique y que no esté expirada:

003036CD	89 5D F8	mov dword ptr ss:[ebp-0x8],ebx	
003036D0	E8 CB D3 00 00	call xml a ctpqi_1.310AA0	
003036D5	83 C4 10	add esp,0x10	
003036D8	3A C3	cmp al,b1	
003036DA	0F 84 8A 00 00 00	je xml a ctpqi_1.30376A	not
003036E0	8B 0E	mov ecx,dword ptr ds:[esi]	
003036E2	C6 81 BD 01 00 00 0	mov byte ptr ds:[ecx+0x18D],0x1	
003036E9	8B 16	mov edx,dword ptr ds:[esi]	
003036EB	88 9A BC 01 00 00	mov byte ptr ds:[edx+0x18C],b1	
003036F1	8B 0E	mov ecx,dword ptr ds:[esi]	
003036F3	8B 55 08	mov edx,dword ptr ss:[ebp+0x8]	
003036F6	89 91 B0 01 00 00	mov dword ptr ds:[ecx+0x180],edx	
003036FC	8B 0E	mov ecx,dword ptr ds:[esi]	

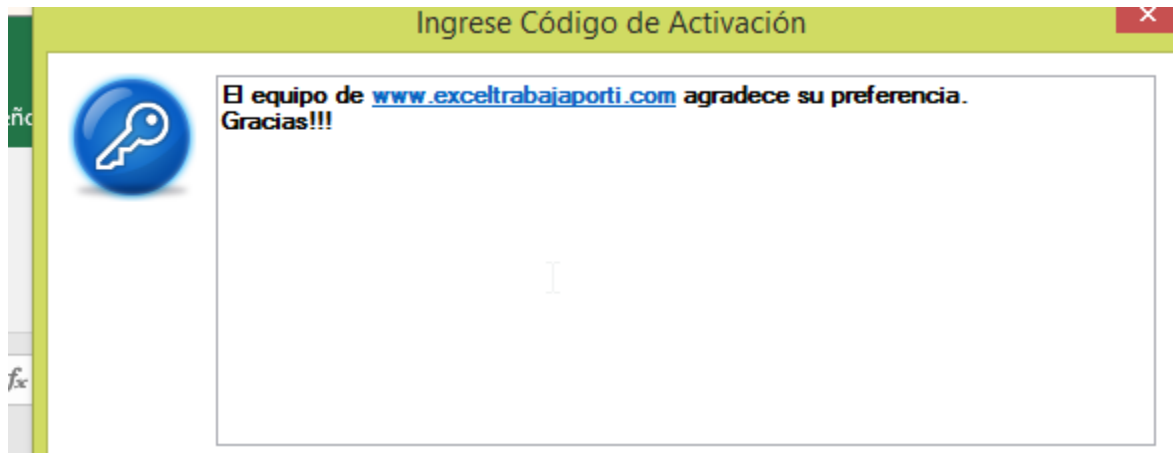
Sobre el otro lugar que validaba la activación salta con eax 1 y edx 1

00303669	75 19	jne xml a ctpqi_1.303684	salta a , eax es 1 y edx es 1
0030366B	6A 33	push 0x33	
0030366D	68 30 37 35 00	push xml a ctpqi_1.353730	353730:L"You must enter Activation Code to use this pro
00303672	8D 48 08	lea ecx,dword ptr ds:[ebx+0x8]	
00303675	E8 B6 57 00 00	call xml a ctpqi_1.308E30	
0030367A	90 7D FF 00	cmp byte ptr ss:[ebp-0x1],0x0	
0030367E	74 23	je xml a ctpqi_1.3036A3	
00303680	C6 45 0B 01	mov byte ptr ss:[ebp+0xB],0x1	
00303684	8B 0D 34 10 36 00	mov ecx,dword ptr ds:[0x361034]	
0030368A	8B 01	mov eax,dword ptr ds:[ecx]	
0030368C	3B C8	cmp ecx,eax	
0030368E	74 13	je xml a ctpqi_1.3036A3	
00303690	8B 50 08	mov edx,dword ptr ds:[eax+0x8]	
00303693	C7 02 00 00 00 00	mov dword ptr ds:[edx],0x0	
00303699	8B 00	mov eax,dword ptr ds:[eax]	

Y logramos apreciar que es ahora en el menú de Excel el menú de desactivar software, asi que correctamente hemos validado el archivo



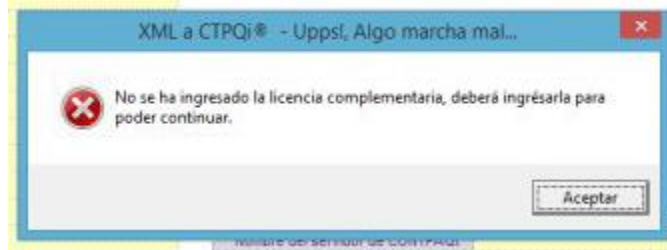
Si forzamos que nos muestre el mensaje actual de introducir un código de activación refiere que nos agradecen la preferencia.



Con esto tenemos la Primera parte del programa listo, la cantidad de ejecuciones ilimitadas
Con el programa activado tenemos ahora un exe que corre directamente sin mostrar favor registrar osea lo tenemos funcionando sin problema.

Pero hay un tema, Ahora a la protección de las macros:

Ahora en plantilla control hay un registro desde las macros que exige registrar



Entonces viene el tema de fondo, ¿como accedo a las macros y al código si no me permite?

Entonces dado que el programa ya se ejecuta y tiene en memoria el Excel, deberé atacharlo
Pienso en una maravillosa idea, que tal si pudiera mover la hoja desde Excel ¿?

Y el código no permite mostrar esa ventana, asi que es manipulado desde lockxls, luego de vencer ya tengo el código visible , asi que comparto el código del setup una vez logrado todo el código:

```
Attribute VB_Name = "Setup"
```

```
#If VBA7 Then
```

```
Private Declare PtrSafe Function LoadLibrary Lib "kernel32" Alias "LoadLibraryA" (ByVal  
lpLibFileName As String) As LongPtr
```

```

Private Declare PtrSafe Function GetProcAddress Lib "kernel32" (ByVal hModule As LongPtr, ByVal lpProcName As String) As LongPtr

Private Declare PtrSafe Function CallWindowProc Lib "user32" Alias "CallWindowProcA" (ByVal lpPrevWndFunc As LongPtr, ByVal hwnd As Object, ByVal Msg As Long, ByVal wParam As Long, ByVal lParam As Long) As Object

#Else

Private Declare Function LoadLibrary Lib "kernel32" Alias "LoadLibraryA" (ByVal lpLibFileName As String) As Long
Private Declare Function GetProcAddress Lib "kernel32" (ByVal hModule As Long, ByVal lpProcName As String) As Long

Private Declare Function CallWindowProc Lib "user32" Alias "CallWindowProcA" (ByVal lpPrevWndFunc As Long, ByVal hwnd As Object, ByVal Msg As Long, ByVal wParam As Long, ByVal lParam As Long) As Object

#End If

Dim g_oRuntime As Object

Dim g_oRibbonCustomUI As Object

Public Sub SetupLockXLSRtmModule(sPath As String, sId As String)
    On Error GoTo err_handler

#If VBA7 Then
    Dim hLib As LongPtr
#Else
    Dim hLib As Long
#End If

    hLib = LoadLibrary(sPath)

    If 0 = hLib Then
        MsgBox "ERROR: Library '" & sPath & "' was not loaded."
    Else

#If VBA7 Then
        Dim pStartupFunc As LongPtr
#Else
        Dim pStartupFunc As Long
#End If
    #End If

```



```
Dim sFuncName As String
sFuncName = "StartupFunc"
```

```
pStartupFunc = GetProcAddress(hLib, sFuncName)
```

```
If 0 = pStartupFunc Then
```

```
    MsgBox "ERROR: startup function not found"
```

```
Else
```

```
    Set g_oRuntime = CallWindowProc(pStartupFunc, Application, 0, 0, 0)
```

```
    Call g_oRuntime.PrepareWorkbook(sId)
```

```
    ' initialize ribbon controls
```

```
    If Not (g_oRibbonCustomUI Is Nothing) Then
```

```
        Call g_oRuntime.OnLoadCustomUI(g_oRibbonCustomUI)
```

```
    End If
```

```
End If
```

```
End If
```

```
Exit Sub
```

```
err_handler:
```

```
    MsgBox "ERROR. Description: " & Err.Description & " Source: " & Err.Source
```

```
End Sub
```

```
Public Function IsHDataPresent() As Boolean
```

```
    IsHDataPresent = False
```

```
    If Not (g_oRuntime Is Nothing) Then
```

```
        IsHDataPresent = g_oRuntime.IsHDataPresent
```

```
    End If
```

```
End Function
```

```
Public Function GetRtmModule() As Variant
```

```
    Set GetRtmModule = g_oRuntime
```

```
End Function
```

```
Public Function LoadHData() As Boolean
```

```
    LoadHData = False
```

```
    If Not (g_oRuntime Is Nothing) Then
```

```
        LoadHData = g_oRuntime.LoadHData
```

```
    End If
```

```
End Function
```

```

'Callback for customUI.onLoad
Sub lxOnLoadCustomUI(ribbon As IRibbonUI)
    Set g_oRibbonCustomUI = ribbon
    If Not (g_oRuntime Is Nothing) Then
        Call g_oRuntime.OnLoadCustomUI(g_oRibbonCustomUI)
    End If
End Sub

'Callback for customUI.loadImage
Sub LoadImage(imageID As String, ByRef returnedVal)
    If Not (g_oRuntime Is Nothing) Then
        Set returnedVal = g_oRuntime.LoadImage(imageID)
    End If
End Sub

'Callback for tabLockXLS getLabel
Sub lxGetTabLabel(control As IRibbonControl, ByRef returnedVal)
    If Not (g_oRuntime Is Nothing) Then
        returnedVal = g_oRuntime.GetTabLabel(control)
    End If
End Sub

'Callback for tabLockXLS getVisible
Sub lxGetTabVisible(control As IRibbonControl, ByRef returnedVal)
    If Not (g_oRuntime Is Nothing) Then
        returnedVal = g_oRuntime.GetTabVisible(control)
    Else
        returnedVal = False
    End If
End Sub

'Callback for LockXLS_EnterAC onAction
Sub lxOnButtonClicked(control As IRibbonControl)
    If Not (g_oRuntime Is Nothing) Then
        Call g_oRuntime.ButtonClicked(control)
    End If
End Sub

'Callback for LockXLS_EnterAC getLabel
Sub lxGetButtonLabel(control As IRibbonControl, ByRef returnedVal)
    If Not (g_oRuntime Is Nothing) Then
        returnedVal = g_oRuntime.GetLabel(control)
    End If

```

```

End Sub

'Callback for LockXLS_EnterAC getImage
Sub lxOnGetImage(control As IRibbonControl, ByRef returnedVal)
    If Not (g_oRuntime Is Nothing) Then
        Set returnedVal = g_oRuntime.GetImage(control)
    End If
End Sub

'Callback for LockXLS_EnterAC getScreentip
Sub lxGetButtonScreentip(control As IRibbonControl, ByRef returnedVal)
    If Not (g_oRuntime Is Nothing) Then
        returnedVal = g_oRuntime.GetScreentip(control)
    End If
End Sub

'Callback for LockXLS_EnterAC getSupertip
Sub lxGetButtonSupertip(control As IRibbonControl, ByRef returnedVal)
    If Not (g_oRuntime Is Nothing) Then
        returnedVal = g_oRuntime.GetSupertip(control)
    End If
End Sub

'Callback for LockXLS_EnterAC getEnabled
Sub lxGetButtonEnabled(control As IRibbonControl, ByRef returnedVal)
    If Not (g_oRuntime Is Nothing) Then
        returnedVal = g_oRuntime.GetEnabled(control)
    End If
End Sub

'Callback for LockXLS_EnterAC getVisible
Sub lxGetButtonVisible(control As IRibbonControl, ByRef returnedVal)
    If Not (g_oRuntime Is Nothing) Then
        returnedVal = g_oRuntime.GetVisible(control)
    End If
End Sub

Public Function IsLockXLSRuntimePresent() As Boolean
    IsLockXLSRuntimePresent = True
End Function

```

En resumen refiere IsLockXLSRuntimePresent = True y que está instalando en una ruta en específico que trabajará como embebido.

Desde la gui del excel , a partir de archivo reparado (exe que extrajo el zip) tenemos

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui" xmlns:x="lxNS"
onLoad="lxOnLoadCustomUI" loadImage="LoadImage">
  <ribbon startFromScratch="false">
    <tabs>
      <tab id="tabLockXLS" getLabel="lxGetTabLabel" getVisible="lxGetTabVisible">
        <group idQ="x:stLXRGroup" label="LockXLS">
          <button id="LockXLS_EnterAC" tag="LockXLS_EnterAC" getLabel="lxGetButtonLabel"
onAction="lxOnButtonClicked" getImage="lxOnGetImage" getScreenTip="lxGetButtonScreenTip"
getSupertip="lxGetButtonSupertip" getEnabled="lxGetButtonEnabled"
getVisible="lxGetButtonVisible" />
          <button id="LockXLS_RemoveAC" tag="LockXLS_RemoveAC" getLabel="lxGetButtonLabel"
onAction="lxOnButtonClicked" getImage="lxOnGetImage" getScreenTip="lxGetButtonScreenTip"
getSupertip="lxGetButtonSupertip" getEnabled="lxGetButtonEnabled"
getVisible="lxGetButtonVisible" />
          <button id="LockXLS_CustHelp" tag="LockXLS_CustHelp" getLabel="lxGetButtonLabel"
onAction="lxOnButtonClicked" getImage="lxOnGetImage" getScreenTip="lxGetButtonScreenTip"
getSupertip="lxGetButtonSupertip" getEnabled="lxGetButtonEnabled"
getVisible="lxGetButtonVisible" />
          <button id="LockXLS_Export" tag="LockXLS_Export" getLabel="lxGetButtonLabel"
onAction="lxOnButtonClicked" getImage="lxOnGetImage" getScreenTip="lxGetButtonScreenTip"
getSupertip="lxGetButtonSupertip" getEnabled="lxGetButtonEnabled"
getVisible="lxGetButtonVisible" />
          <button id="LockXLS_SaveAll" tag="LockXLS_SaveAll" getLabel="lxGetButtonLabel"
onAction="lxOnButtonClicked" getImage="lxOnGetImage" getScreenTip="lxGetButtonScreenTip"
getSupertip="lxGetButtonSupertip" getEnabled="lxGetButtonEnabled"
getVisible="lxGetButtonVisible" />
          <button id="LockXLS_Import" tag="LockXLS_Import" getLabel="lxGetButtonLabel"
onAction="lxOnButtonClicked" getImage="lxOnGetImage" getScreenTip="lxGetButtonScreenTip"
getSupertip="lxGetButtonSupertip" getEnabled="lxGetButtonEnabled"
getVisible="lxGetButtonVisible" />
        </group>
      </tab>
    </tabs>
  </ribbon>
</customUI>
```

En resumen Que todas las opciones de guardar y otras están siendo manejadas desde el lockxls con un custom gui.

El exe original ejecuta todo el modulo, Con el tiempo hasta podía ver cuando llegaba al setup de la macro

00D1B003 | 68 68 C3 D4 00 | push xml a ctpqi_expiredrun.D4C368 |
D4C368:L"SetupLockXLSRtmModule"

Pero aun asi solo lanza el Excel, por lo cual no es viable hacer un debug children..

Ahora comenzamos a depurar con permisos de admin haciendo attach a Excel y comenzando a dejar el punto de vista hacia las macros (recuerden que ya se desbloquear las contraseñas) y por otro lado ahora hay que analizar la dll del tutorial osea hay que conocer mas:

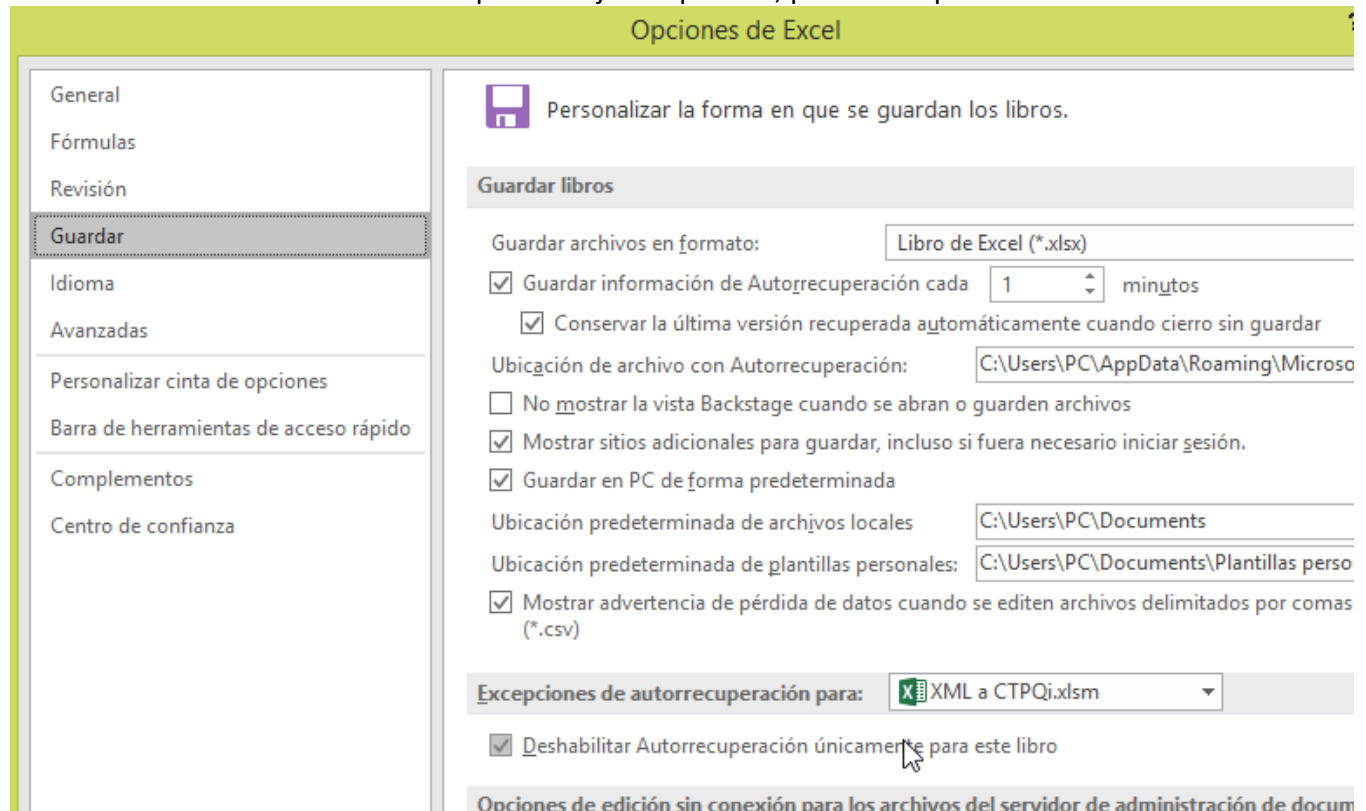
Si desactivo de casualidad la dll (matándola) desde el Excel (thread)

Tenemos en Excel un error grave:

```
-----  
Microsoft Excel  
-----  
ERROR: Library 'C:\Users\PC\AppData\Local\Temp\SpreadsheetTools\32\LockXLSRuntime.dll' was  
not loaded.  
-----  
Aceptar  
-----
```

Maneja Temporales, maneja las excepciones de autorecuperacion

Bueno al menos con esto sabemos que maneja temporales, podemos apreciar como además

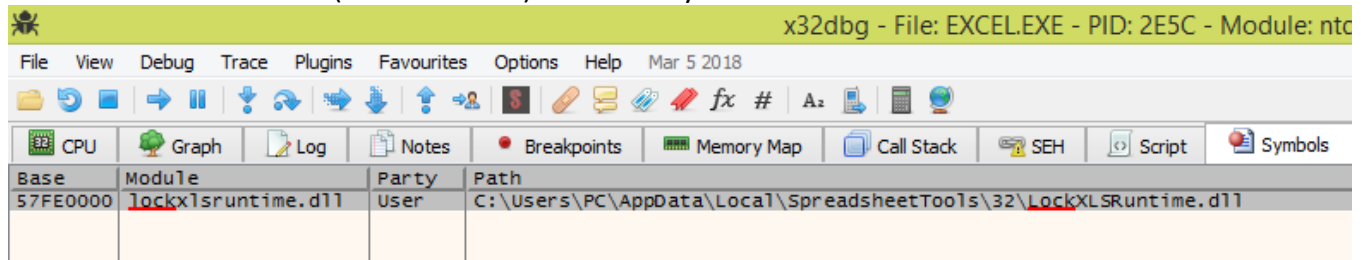


Maneja que desactive la autorecuperacion del libro y si detecta un archivo xlsb para recuperación lo daña, entonces tarde o temprano ir por la via de crashear el Excel y luego abrirlo, parece poco sensato que protege de todas formas el programa, puedo crashearlo al minuto, tener suspendida la dll y con ello tengo el primer xlsx con código en parte legible, pero además hay que seguir teniendo en cuenta algo, que necesito sacar las macros.

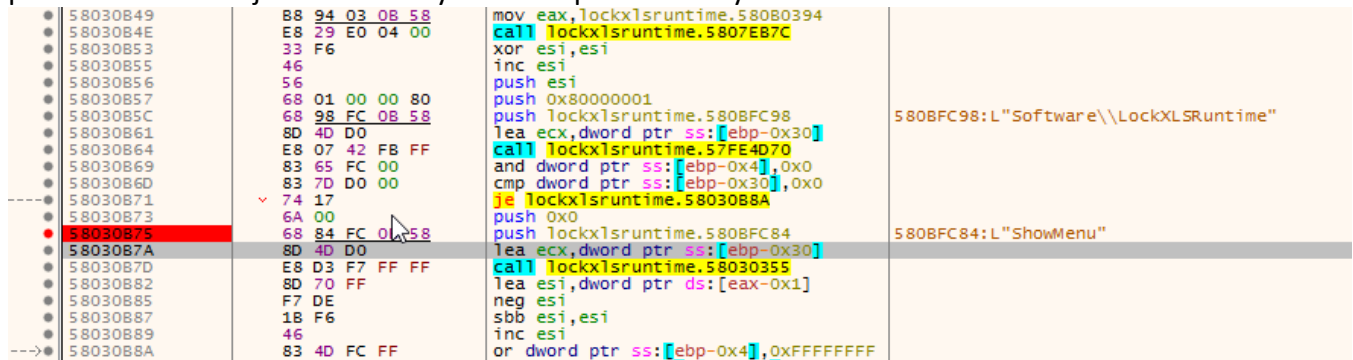
Lo intento de otra forma sin tener que usar el recuperado(ahora tengo acceso a mirar por encima las macros pero desde un simple modo Notepad)

Atacheo Excel como admin, luego en symbol voy a lockxlsruntime

Y con el tema de macros (alt+f11 visible) necesito ir y volver

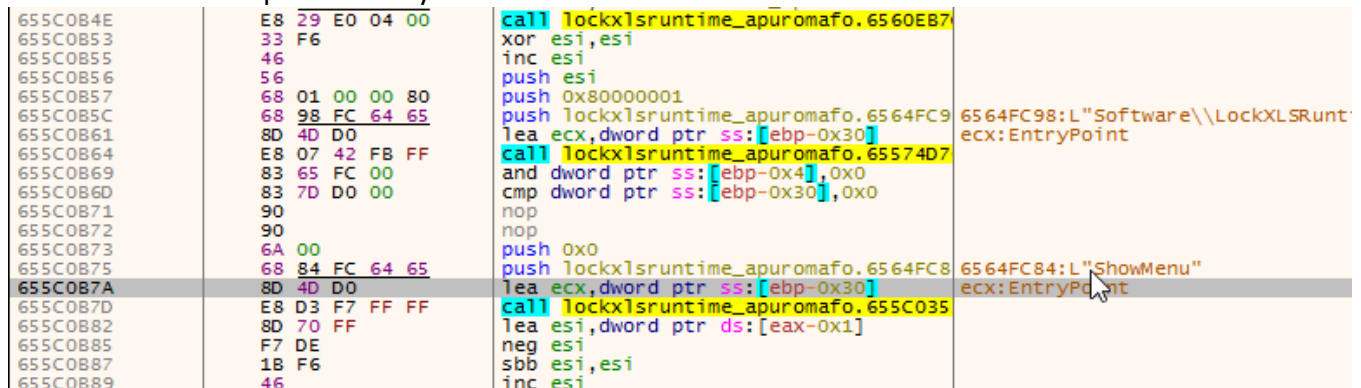


busco forzar saltos, que salten que den valores, que lo único necesario es Si logro habilitar showmenu puedo mover la hoja a otro libro y de ahí exportar todo ya con menos detalle.



Y los show Windows... si logro acceder a esos modulos puedo manipularlos y ver el menú

Al reiniciar coloco bp en esa dll y cambiaré a un show menú desactivado



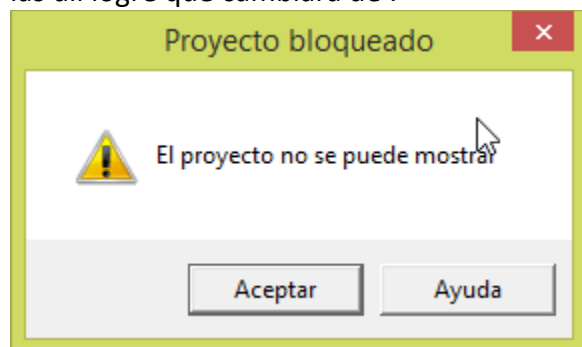
Y así suma y sigue, luego de encontrar propiedades veryhidden y otras, fui intentando desligar el programa versus la dll (que no reconociera todas las acciones, extensiones y otras)

En la zona de exportar un archivo siempre lo guarda como .exe, cuando logré vencer aquello entonces voy denuevo, Conforme pasó el tiempo me propuse ir venciendo este estilo de sandbox, necesitaba únicamente acceder a las macros y dado el permiso de todo ser manejado en el dll, no sería fácil Hice una y mil piruetas y en un momento dado de una semana logré algo inesperado:

se me habilitó el menú de mover 😊

¿que hice cuando se activó ese menú ? movi los libros a otro libro
Con alt+f11 comencé a mirar las macros de Excel

luego pude ver las macros de Excel 😊 pero el proyecto no es visible /veryhidden y otros, editando en las dll logré que cambiara de :

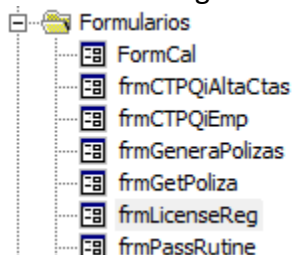


a algo visible

osea el proyecto ahora era visible y con esto miraba las macros, pero faltaba el ultimo detalle guardar todo, uno a uno, luego se podían importar y terminar el trabajo
en resumen

- 1) Guardado el libro en un 2do libro
- 2) Guardado las macros y codigos en una carpeta que luego debo importar
- 3) Debo intentar hacer funcionar el código de registro y con ello nunca mas usar el xlsx y puedo seguir usando el exe
- 4) Revisar el código para registrarnos

La macro del registro



En el botón ok

Antes:

```
Private Sub cmdOK_Click()

    aplicacion_name = VBA.Replace(VBA.Replace(Names("aplication.name"), "=", "", ""), "====", "")
    strCodProduct = VBA.Replace(Me.txbxGuidEq, " ", "")
    strLicense = VBA.Replace(Me.txbxSerialEq.Text, " ", "")

    If strLicense = "" Then
        MsgBoxEx "No haz ingresado la licencia complementaria, debes ingresarla para continuar.",
        vbCritical, AppName, , , strPathIComod
        Me.txbxSerialEq.SetFocus
        Exit Sub
    End If

    If Hash.DigestStrToHexStr(aplicacion_name & strCodProduct) <> strLicense Then
        BaseReg.RegVal "SOFTWARE\Classes\adox.x\" & aplicacion_name,
        Base64EncodeString(strCodProduct), ""
        MsgBoxEx "La licencia no es válida, no corresponde al equipo.", vbCritical, AppName, , ,
        strPathIComod
        Me.txbxSerialEq.SetFocus
        Exit Sub
    Else
        BaseReg.RegVal "SOFTWARE\Classes\adox.x\" & aplicacion_name,
        Base64EncodeString(strCodProduct), strLicense
        MsgBoxEx "La licencia es válida, se ha ingresado exitosamente.", vbInformation, AppName, , ,
        strPathIComod
    End If

    Unload Me

End Sub
```

Después:

```
Private Sub cmdOK_Click()

    aplicacion_name = VBA.Replace(VBA.Replace(Names("aplication.name"), "=", "", ""), "====", "")
    strCodProduct = VBA.Replace(Me.txbxGuidEq, " ", "")
    strLicense = VBA.Replace(Me.txbxSerialEq.Text, " ", "")

    If strLicense = "" Then
        MsgBoxEx "No haz ingresado la licencia complementaria, debes ingresarla para continuar.",
        vbCritical, AppName, , , strPathIComod
```



```

Me.txbxSerialEq.SetFocus
Exit Sub
End If

If Hash.DigestStrToHexStr(application_name & strCodProduct) <> strLicense Then
    BaseReg.RegVal "SOFTWARE\Classes\adox.x\" & application_name,
    Base64EncodeString(strCodProduct), ""
    strLicense = Hash.DigestStrToHexStr(application_name & strCodProduct)
    MsgBoxEx strLicense, vbCritical, AppName, , , strPathICOMod
    Me.txbxSerialEq.SetFocus
    Exit Sub
Else
    BaseReg.RegVal "SOFTWARE\Classes\adox.x\" & application_name,
    Base64EncodeString(strCodProduct), strLicense
    strLicense = Hash.DigestStrToHexStr(application_name & strCodProduct)
    MsgBoxEx strLicense, vbCritical, AppName, , , strPathICOMod

End If

Unload Me

End Sub


```


Para quitar la licencia se puede hacer en SOFTWARE\Classes\adox.x\

Hay archivos stub en %temp% o %appdata% ejemplo

C:\Users\PC\AppData\Local\SpreadsheetTools

Con la dll runtime analizada es tema de parchar y ahora que trabaje con una versión modificada

 LockXLSRuntime.dll

 LockXLSRuntime_apuromafo.dll

Con ello esta todo listo a ser explorado 😊

Precios de licencia:

Lock xls

You are licensed to use this software for evaluation purposes without charge for a period of 30 days.

If you wish to use this software after the 30-day evaluation period, you must buy a 2-month subscription (USD 50.95), or a full license (USD 249.99).

Programa no refiere que tipo de dólar, pero debido a que es de México asumimos

\$3,299.00 = Peso **Mexicano** 3299 MXN = 168.48 USD **Dólar** estadounidense

Con 2 licencias pagan el uso del lock xls.

Palabras Finales

No he expuesto los lugares de todas las zonas analizadas de lock xls solo por tiempo, pero de seguro es bastante variables que tiene en consideración para proteger a un simple xlsx, tiene runtime suficiente para ejecutar muchas opciones, espero no encontrar troyanos o virus con este tipo de protección, pero es solo cambiar saltos, alguien con conocimiento básico lo puede hacer sin problema si logra documentar todas las zonas, se deja como una experiencia vivida solamente este tutorial.

Saludos Cordiales Apuromafo CLS

