

C-Linx 7.5.0.0



Escrito por:

- Apuromafo y KarMa [954] ©Full

Fecha de Creación: 23-01-2020

- Creado con Fines Educativos

Herramientas

Herramienta	Enlace	Uso
Editor de texto	http://docs.google.com	Si cuentas con una cuenta en google drive puedes almacenar algunos documentos y compartirlos con los demás en un enlace, además de usar google docs.
ExeinfoPE	http://www.exeinfo.xn.pl/	Analizador de código PE
Dnspy	https://github.com/0xd4d/dnSpy	Depurador de .net y ensamblador
Everything	https://voidtools.com/	buscador
Inno Decompiler		*opcional
Cff explorer		*opcional

Frase

Todo fuego se consume a sí mismo. *"La decisión más difícil"* (2004), Jodi Picoult

Información

Programa:	C-Linx 7.5.0.0 Fike® is a registered trademark of Fike Corporation.
Herramientas a usar	exeinfope, cffexplorer, notepad Dnspy, visual studio
Dificultad	no está empacada.
Utilidad del programa	Programa para monitoreo de paneles de seguridad y alarmas de incendios, programa solo para sistemas de incendios, no apto para otras personas.
Objetivo	Registrar
Fecha	Enero 2019
Autor	Apuromafo & KarMa [954] ©Full

Introducción

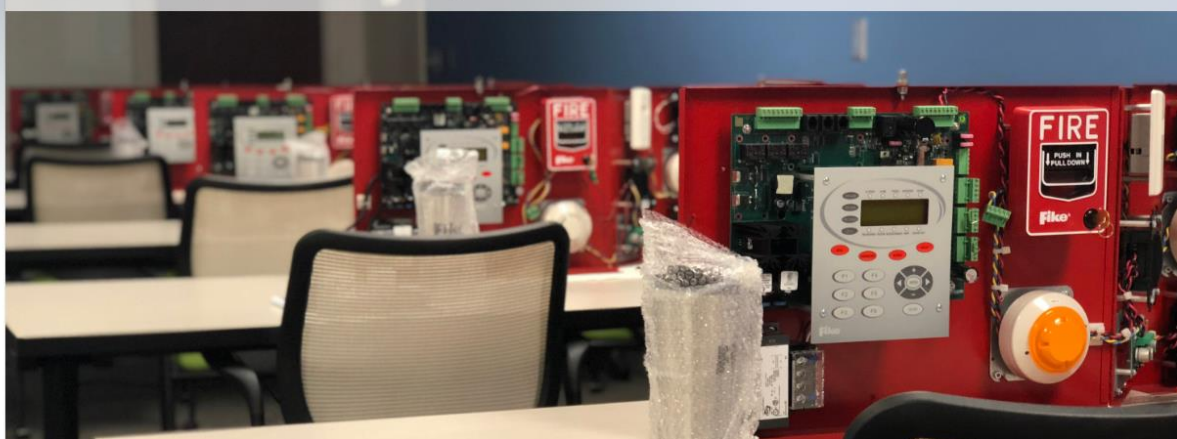
Hola, ha pasado tiempo desde que escribir tutoriales se refiere, me animé a revisar una aplicación, con fines educativos, espero les guste

Desarrollo

Comenzamos la historia diciendo que me compartieron el ejecutable, y al usar google veamos

tenemos una aplicación del 2018, según version 7.5, al buscar sobre fike, llegamos que quienes tienen acceso a esta información tienen cursos

NOTICE: As of November 1st, the myFike URL will be changing to <https://training.fike.com>. Also on November 1st, <https://my.fike.com> will be the new location of Fike's new customer experience platform- MyFike (formerly portal.fike.com).



<https://training.fike.com/courses/clinx> luego de terminar el curso se puede instalar este tipo de aplicación.

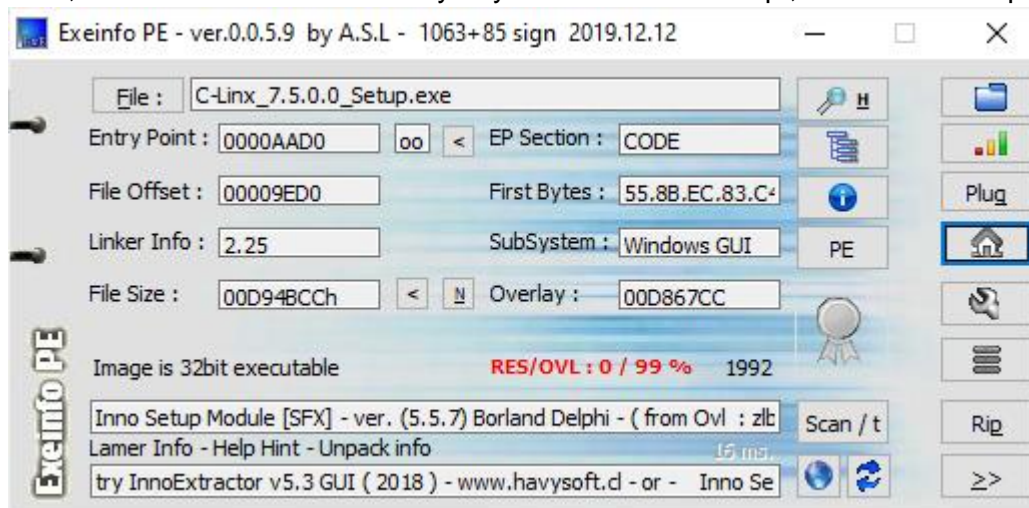
comenzamos la experiencia:

Comenzamos que dice un escáner de signatura, para esto usaré exeinfope



Item :	Download link	Date
Exeinfo Pe ver.0.0.5.9 1063 / 85 signatures (Ext_detect: v.0.5.4.0) userdbCAT : 442?	Download link many bugs fixed. Fixed EXEP scanning Engine Added / NOT WIN EXE -> ARM exe (32bit only, ARM file - CPU) NT500 CPU8 Added sign : NoobSub v.0.02 signature Installer update : InstallCompiler 3.0 - Install Creator - Windows sign power & (Setup / Uninstall) file ota - Install starts 2019 for new version Fixed - Windows - 0000 BOM Extension (X-00) fixed 1015 VMProtect v3.00 - 3.3.2.1 "read EP-1" fixed 048 EXE7 Uninstaller - fix update many errors ignored and fixed 64 bit on Windows (added 20 sign)	updated ! 03.07.2020 

debido al tiempo que llevo, he realizado una donación y hay acceso a la zona vip , veamos el Setup:



es un instalador de inno

al usar un decompiler

```
innounp, the Inno Setup Unpacker. Version 0.49
Usage: innounp [command] [options] <setup.exe or setup.0> [@filelist] [filemask ...]
Commands:
  (no)      display general installation info
  -v        verbosely list the files (with sizes and timestamps)
  -x        extract the files from the installation (to the current directory, also see -d)
  -e        extract files without paths
  -t        test files for integrity
Options:
  -b        batch (non-interactive) mode - will not prompt for password or disk changes
  -q        do not indicate progress while extracting
  -m        extract internal embedded files (such as license and uninstall.exe)
  -pPASS    decrypt the installation with a password
  -dDIR     extract the files into DIR (can be absolute or relative path)
  -cDIR     specifies that DIR is the current directory in the installation
  -n        don't attempt to unpack new versions
  -fFILE    same as -p but reads the password from FILE
  -a        extract all copies of duplicate files
  -y        assume Yes on all queries (e.g. overwrite files)

C:\Users\W10\Downloads\Nueva carpeta\Compressed\wetransfer-42c0fc\C-Linx_7.5.0.0_Setup>innounp -m C-Linx_7.5.0.0_Setup.exe
; Version detected: 5507
Compression used: lzma
Files: 27 ; Bytes: 36571970
Compiled Pascal script: 511 byte(s)

C:\Users\W10\Downloads\Nueva carpeta\Compressed\wetransfer-42c0fc\C-Linx_7.5.0.0_Setup>innounp -m -x C-Linx_7.5.0.0_Setup.exe
; Version detected: 5507
#1 {app}\C-Linx.exe
Reading slice C:\Users\W10\Downloads\Nueva carpeta\Compressed\wetransfer-42c0fc\C-Linx_7.5.0.0_Setup\C-Linx_7.5.0.0_Setup.exe
#2 {app}\06-448 C-Linx - Rev 0.pdf
#3 {app}\C-Linx.ico
```

tenemos el setup

```
;InnoSetupVersion=5.5.7
```

```
[Setup]
```

```
AppName=C-Linx 7.5.0.0
```

```
AppVerName=C-Linx 7.5.0.0
```

```
AppId={{AE05BD30-0B03-4D41-9489-528A435976FC}}
```

```
AppVersion=7.5.0.0
```

```
AppPublisher=Fike Corporation
```

```
AppPublisherURL=http://www.fike.com
```

```
AppSupportURL=http://www.fike.com
```

```
AppUpdatesURL=http://www.fike.com
```

```
DefaultDirName={pf}\Fike Corporation\C-Linx
```

```
UninstallDisplayIcon={app}\C-Linx.ico
```

```
OutputBaseFilename=C-Linx_7.5.0.0_Setup
```

```
Compression=lzma
```

```
DisableDirPage=auto
```

```
DisableProgramGroupPage=yes
```

```
ChangesAssociations=yes
```

```
WizardImageFile=embedded\WizardImage0.bmp
```

```
WizardSmallImageFile=embedded\WizardSmallImage0.bmp
```

```
[Files]
```

```
Source: "{app}\C-Linx.exe"; DestDir: "{app}"; MinVersion: 0.0,5.0; Flags: ignoreversion
```

```
Source: "{app}\06-448 C-Linx - Rev 0.pdf"; DestDir: "{app}"; MinVersion: 0.0,5.0; Flags: ignoreversion
```

```
Source: "{app}\C-Linx.ico"; DestDir: "{app}"; MinVersion: 0.0,5.0; Flags: ignoreversion
```

```
Source: "{app}\Cheetah Xi 1016 Battery Worksheet Rev 2.xls"; DestDir: "{app}";
```

```
MinVersion: 0.0,5.0; Flags: ignoreversion
```

Source: "{app}\Cheetah Xi 50 Battery Worksheet Rev 2.xls"; DestDir: "{app}"; MinVersion: 0.0,5.0; Flags: ignoreversion
Source: "{app}\CyberCat 1016 Battery Worksheet Rev 3.xls"; DestDir: "{app}"; MinVersion: 0.0,5.0; Flags: ignoreversion
Source: "{app}\CyberCat 254 Battery Worksheet Rev 2.xls"; DestDir: "{app}"; MinVersion: 0.0,5.0; Flags: ignoreversion
Source: "{app}\CyberCat 50 Battery Worksheet Rev 2.xls"; DestDir: "{app}"; MinVersion: 0.0,5.0; Flags: ignoreversion
Source: "{app}\C-Linx.exe.config"; DestDir: "{app}"; MinVersion: 0.0,5.0; Flags: ignoreversion
Source: "{app}\CLinxSecurity.dll"; DestDir: "{app}"; MinVersion: 0.0,5.0; Flags: ignoreversion
Source: "{app}\DevExpress.Data.v13.1.dll"; DestDir: "{app}"; MinVersion: 0.0,5.0; Flags: ignoreversion
Source: "{app}\DevExpress.Office.v13.1.Core.dll"; DestDir: "{app}"; MinVersion: 0.0,5.0; Flags: ignoreversion
Source: "{app}\DevExpress.Printing.v13.1.Core.dll"; DestDir: "{app}"; MinVersion: 0.0,5.0; Flags: ignoreversion
Source: "{app}\DevExpress.RichEdit.v13.1.Core.dll"; DestDir: "{app}"; MinVersion: 0.0,5.0; Flags: ignoreversion
Source: "{app}\DevExpress.Sparkline.v13.1.Core.dll"; DestDir: "{app}"; MinVersion: 0.0,5.0; Flags: ignoreversion
Source: "{app}\DevExpress.Utils.v13.1.dll"; DestDir: "{app}"; MinVersion: 0.0,5.0; Flags: ignoreversion
Source: "{app}\DevExpress.XtraBars.v13.1.dll"; DestDir: "{app}"; MinVersion: 0.0,5.0; Flags: ignoreversion
Source: "{app}\DevExpress.XtraEditors.v13.1.dll"; DestDir: "{app}"; MinVersion: 0.0,5.0; Flags: ignoreversion
Source: "{app}\DevExpress.XtraPrinting.v13.1.dll"; DestDir: "{app}"; MinVersion: 0.0,5.0; Flags: ignoreversion
Source: "{app}\DevExpress.XtraTreeList.v13.1.dll"; DestDir: "{app}"; MinVersion: 0.0,5.0; Flags: ignoreversion
Source: "{app}\pcDSPCfgCodec.exe"; DestDir: "{app}"; MinVersion: 0.0,5.0; Flags: ignoreversion
Source: "{app}\PreciseVisionDevices.txt"; DestDir: "{app}"; MinVersion: 0.0,5.0; Flags: ignoreversion

[Registry]

Root: HKCR; Subkey: ".lnx"; ValueType: String; ValueData: "C-Linx 7.5.0.0"; MinVersion: 0.0,5.0; Flags: uninsdeletevalue
Root: HKCR; Subkey: "C-Linx 7.5.0.0"; ValueType: String; ValueData: "Program C-Linx 7.5.0.0"; MinVersion: 0.0,5.0; Flags: uninsdeletekey
Root: HKCR; Subkey: "C-Linx 7.5.0.0\DefaultIcon"; ValueType: String; ValueData: "{app}\C-Linx.exe,0"; MinVersion: 0.0,5.0;
Root: HKCR; Subkey: "C-Linx 7.5.0.0\shell\open\command"; ValueType: String; ValueData: "\"{app}\C-Linx.exe\" \"%1\""; MinVersion: 0.0,5.0;

[Run]

Filename: "{app}\C-Linx.exe"; Description: "{cm:LaunchProgram,C-Linx 7.5.0.0}"; MinVersion: 0.0,5.0; Flags: postinstall skipifsilent nowait

[Icons]

Name: "{commonprograms}\C-Linx 7.5.0.0"; Filename: "{app}\C-Linx.exe"; MinVersion: 0.0,5.0;

Name: "{commondesktop}\C-Linx 7.5.0.0"; Filename: "{app}\C-Linx.exe"; Tasks: desktopicon; MinVersion: 0.0,5.0;

[Tasks]

Name: "desktopicon"; Description: "{cm:CreateDesktopIcon}"; GroupDescription: "{cm:AdditionalIcons}"; MinVersion: 0.0,5.0;

[InstallDelete]

Type: files; Name: "{app}\C-Linx.exe";
 Type: files; Name: "{app}\06-448 C-Linx - Rev 0.pdf";
 Type: files; Name: "{app}\C-Linx.ico";
 Type: files; Name: "{app}\Cheetah Xi 1016 Battery Worksheet Rev 2.xls";
 Type: files; Name: "{app}\Cheetah Xi 50 Battery Worksheet Rev 2.xls";
 Type: files; Name: "{app}\CyberCat 1016 Battery Worksheet Rev 3.xls";
 Type: files; Name: "{app}\CyberCat 254 Battery Worksheet Rev 2.xls";
 Type: files; Name: "{app}\CyberCat 50 Battery Worksheet Rev 2.xls";
 Type: files; Name: "{app}\C-Linx.exe.config";
 Type: files; Name: "{app}\pcDSPCfgCodec.exe";
 Type: files; Name: "{app}\PreciseVisionDevices.txt";

[CustomMessages]

english.NameAndVersion=%1 version %2
 english.AdditionalIcons=Additional shortcuts:
 english.CreateDesktopIcon=Create a &desktop shortcut
 english.CreateQuickLaunchIcon=Create a &Quick Launch shortcut
 english.ProgramOnTheWeb=%1 on the Web
 english.UninstallProgram=Uninstall %1
 english.LaunchProgram=Launch %1
 english.AssocFileExtension=&Associate %1 with the %2 file extension
 english.AssocFileExtension=Associating %1 with the %2 file extension...
 english.AutoStartProgramGroupDescription=Startup:
 english.AutoStartProgram=Automatically start %1
 english.AddonHostProgramNotFound=%1 could not be located in the folder you selected.%n%nDo you want to continue anyway?

[Languages]

; These files are stubs
 ; To achieve better results after recompilation, use the real language files
 Name: "english"; MessagesFile: "embedded\english.isl";

todo indica que es una aplicación .net pues usa dependencias .core , tiene integración de archivos **lnx**

además hay un código compilado (.bin), por lo tanto hay una función, veamos que hace usando inno setup decompiler, sobre el código binario refiere:

```
{
  Created with Inno Setup Decompiler 1.5.0.0

  http://vdisasm.com/isd/
}
```



```

function PREPARETOINSTALL(var Arg0: BOOLEAN): String;
var
    v_1: BOOLEAN;
    v_2, v_3: String;
begin
    label_232:
    v_3 := '{app}\C-Linx.exe';
    v_2 := EXPANDCONSTANT(v_3);
    v_1 := FILEEXISTS(v_2);
    flag := not v_1;
    if flag then goto label_510;
    label_317:
    result := 'Installation of C-Linx cannot continue because another version is already
installed. To remove the existing version of this product, use Add/Remove Programs on
the Control Panel.';
    label_510:
    exit;
end;

```

File Language Help

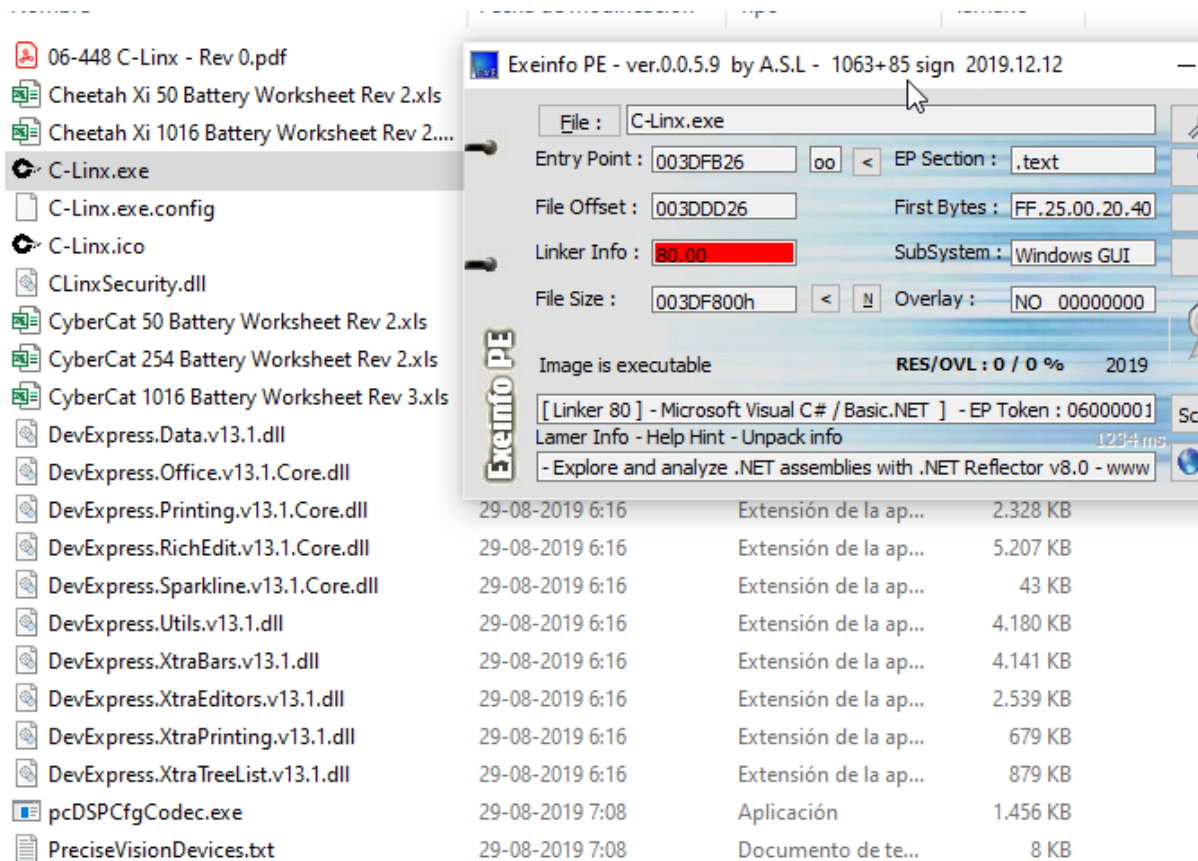
- ▲ Methods
 - !MAIN**
 - PREPARETOINSTALL
- ▲ Externals
 - FILEEXISTS
 - EXPANDCONSTANT
 - Externals (DLL)

```

1|{
2|    Created with Inno Setup Decompiler 1.5.0.0
3|
4|    http://vdisasm.com/isd/
5|}
6|
7|procedure !MAIN();
8|begin
9|    label_231:
10|    exit;
11|end;

```

por lo tanto, no hay nada adicional que ver, solo instala si no existe el ejecutable antes. Ahora vamos por la aplicación Principal: refiere estar en .net

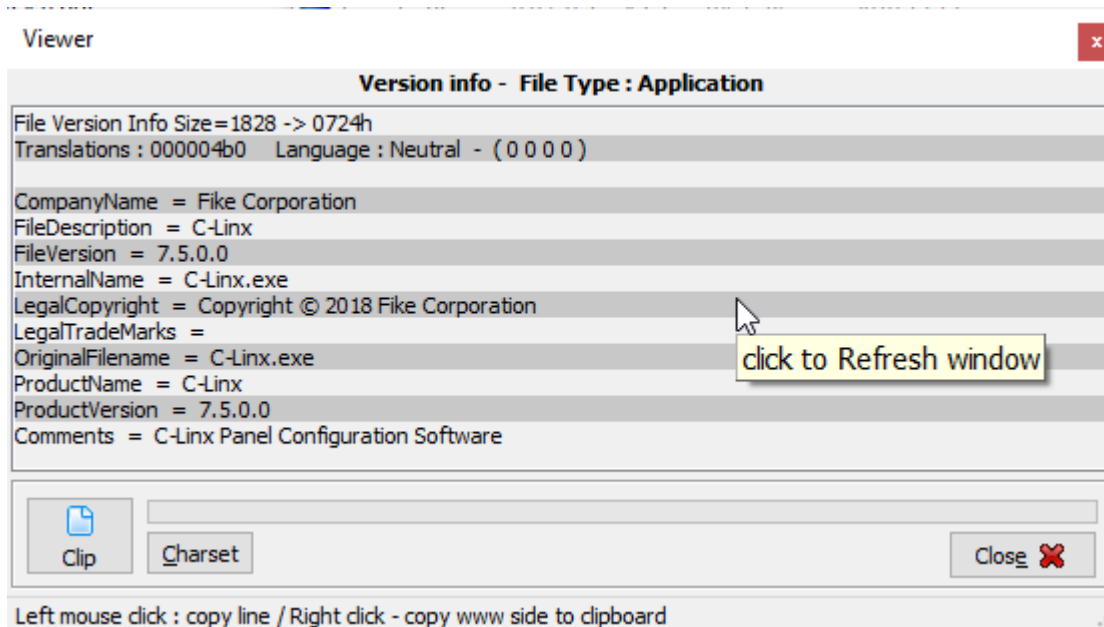


para herramientas .net me refiere:

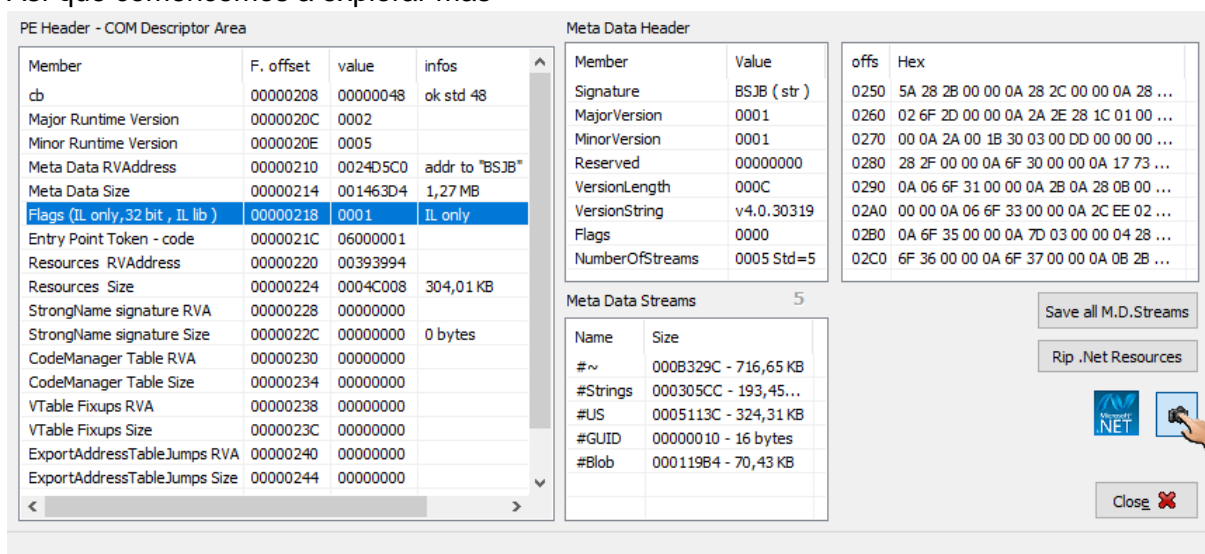
- Explore and analyze .NET assemblies with .NET Reflector v8.0 - www.red-gate.com/products/reflector
- IF file is packed try .NET Generic unpacker from : www.quequero.org/Pack_Unpack

en este caso, lo primero a tener en consideración , es que podría usar **jetbrains, dnspy, reflector , net resolver, ilspy,dis#** entre otras varias cmo **cfexplorer**

Comenzamos a ver sus características en exeinfope:



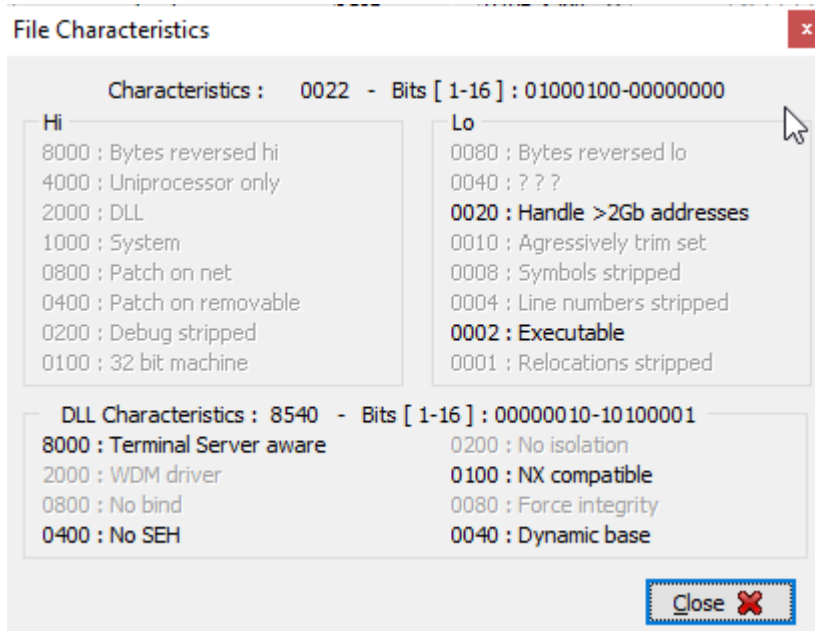
Así que comencemos a explorar mas



se aprecia que no tiene firmas strongname, si alguno nunca ha usado .net aqui pueden comenzar

<https://dotnet.microsoft.com/download>

por otro lado



si usamos cffexplorer, vemos cómo hay muchísima información

C-Linx.exe

Module (1)

- 1 - (C-Linx.exe)

TypeRef (399)

TypeDef (378)

Field (6522)

Method (13625)

Param (11990)

InterfaceImpl (24)

MemberRef (1187)

Constant (1227)

CustomAttribute (14594)

FieldMarshal (39)

ClassLayout (2)

StandAloneSig (948)

EventMap (10)

Event (63)

PropertyMap (169)

Property (5089)

MethodSemantics (9746)

MethodImpl (16)

ModuleRef (1)

TypeSpec (65)

ImplMap (6)

FieldRVA (2)

Assembly (1)

AssemblyRef (10)

- 1 - (mscorlib)
- 2 - (System)
- 3 - (Microsoft.VisualBasic)
- 4 - (System.Windows.Forms)
- 5 - (System.Drawing)
- 6 - (DevExpress.XtraPrinting.v13.1)
- 7 - (DevExpress.Printing.v13.1.Core)
- 8 - (DevExpress.Data.v13.1)
- 9 - (CLinxSecurity.v13.1)**
- 10 - (System.Core)

ManifestResource (80)

NestedClass (89)

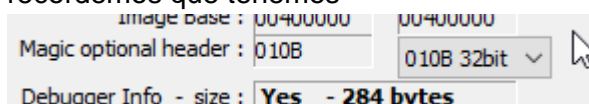
GenericParam (5)

MethodSpec (54)

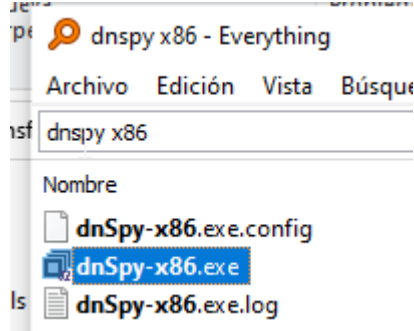
Member	Offset	Size	Value	Meaning
MajorVersion	002FE34C	Word	0001	
MinorVersion	002FE34E	Word	0000	
BuildNumber	002FE350	Word	0000	
RevisionNumber	002FE352	Word	0000	
Flags	002FE354	Dword	00000000	Click here
PublicKeyOrTok...	002FE358	Dword	00000000	Blob Index
Name	002FE35C	Dword	000303F1	CLinxSecurity
Culture	002FE360	Dword	00000000	
HashValue	002FE364	Dword	00000000	Blob Index

hay varias dll que usará (core), pero al parecer la de mayor interés será esta que refiere “Seguridad”, veamos cómo nos va.

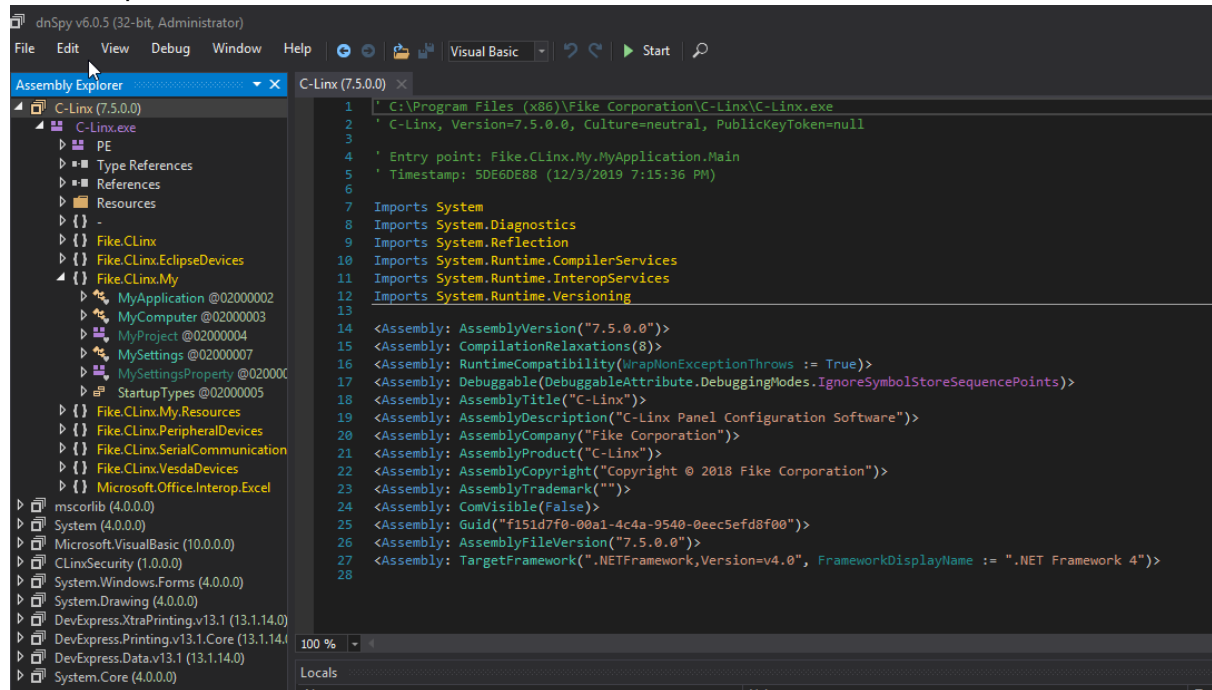
recordemos que tenemos



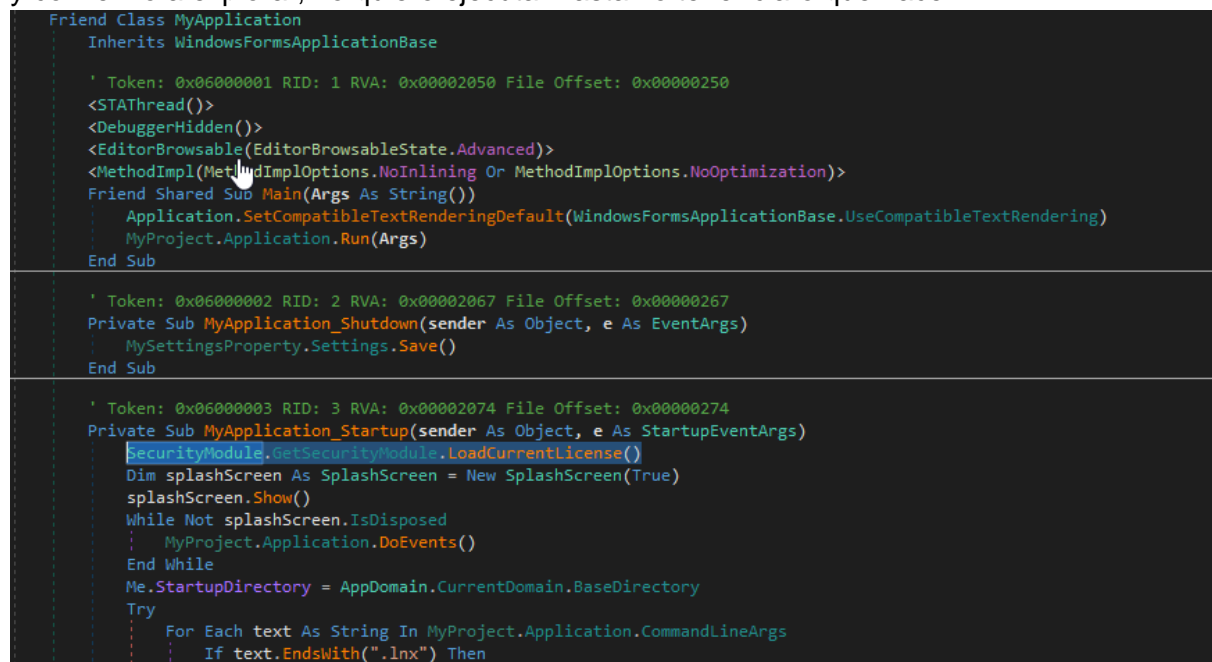
usando everything, procedo a buscar dnspy, y lo ejecuto



abro la aplicación dentro del mismo



y comienzo a explorar, no quiero ejecutar hasta no tener claro que hace



cuando esta el main (entrypoint) ejecutará un dialogo , tiene opciones de guardar, opciones de un form inicial (splash), y hará diversos eventos, según si fue invocado por abrir el programa o desde una extensión .LNX

respecto a la licencia, está en el modulo de seguridad

```
Public Sub LoadCurrentLicense()
    Dim registryKey As RegistryKey = MyProject.Computer.Registry.CurrentUser.OpenSubKey("SOFTWARE")
    If registryKey IsNot Nothing Then
        Dim registryKey2 As RegistryKey = registryKey.OpenSubKey("Fike Corporation")
        If registryKey2 IsNot Nothing Then
            Dim registryKey3 As RegistryKey = registryKey2.OpenSubKey("C-Linx")
            If registryKey3 IsNot Nothing Then
                Dim text As String = Conversions.ToString(registryKey3.GetValue("Item2"))
                Conversions.ToString(registryKey3.GetValue("Item3"))
                Conversions.ToString(registryKey3.GetValue("Item4"))
                Dim text2 As String = Conversions.ToString(registryKey3.GetValue("Item5"))
                If String.IsNullOrEmpty(text) OrElse text.Length <> 35 Then
                    Me.m_license = String.Empty
                Else
                    Me.m_license = text
                End If
                If String.IsNullOrEmpty(text2) OrElse text2.Length <> 35 Then
                    Me.m_tempLicense = String.Empty
                Else
                    Me.m_tempLicense = text2
                End If
                registryKey3.Close()
            End If
            registryKey2.Close()
        End If
        registryKey.Close()
    End If
    If Not Me.LoadLicenseInfo(Me.m_license) Then
```

refiere que la licencia verifica desde un valor de regedit, el cual tiene varios items, según lo que se logra apreciar

Rama de regedit: Equipo\HKEY_CURRENT_USER\Software\Fike Corporation\C-Linx
cuando vemos las instrucciones, son cómo para no entender, pero cuando termina, hay algunas interesantes, load licence y save licence

```
Me.m_lastRunDate = DateTime.Now.[Date]
Me.SaveToRegistry()
End Sub

' Token: 0x06000015 RID: 21 RVA: 0x0002F14 File Offset: 0x0001114
Private Sub SaveToRegistry()
    Dim registryKey As RegistryKey = MyProject.Computer.Registry.CurrentUser.OpenSubKey("SOFTWARE", True)
    If registryKey Is Nothing Then
        registryKey = MyProject.Computer.Registry.CurrentUser.CreateSubKey("SOFTWARE")
    End If
    Dim registryKey2 As RegistryKey = registryKey.OpenSubKey("Fike Corporation", True)
    If registryKey2 Is Nothing Then
        registryKey2 = registryKey.CreateSubKey("Fike Corporation", RegistryKeyPermissionCheck.ReadWriteSubTree)
    End If
    Dim registryKey3 As RegistryKey = registryKey2.OpenSubKey("C-Linx", True)
    If registryKey3 Is Nothing Then
        registryKey3 = registryKey2.CreateSubKey("C-Linx")
    End If
    registryKey3.SetValue("Item2", Me.m_license)
    registryKey3.SetValue("Item3", Me.m_lastRunDate.ToString("MMddyyyy"))
    registryKey3.SetValue("Item4", Me.DaysRemaining)
    registryKey3.SetValue("Item5", Me.m_tempLicense)
    registryKey3.SetValue("Item6", Me.DaysRemaining)
    registryKey3.Close()
    registryKey2.Close()
    registryKey.Close()
End Sub
```

Ahi podemos apreciar todo, el item 2, es la variable m_license , item 3 maneja el dia del ultimo uso ejecutado, item4 y 6 manejan dias restantes (exceso de información?), y el item 5 solamente en caso de una licencia temporal

tenemos varios tipos de licencia

```

ityModule SecurityLicenseTypes X
1 Imports System
2
3 Namespace CLinxSecurity
4     ' Token: 0x02000009 RID: 9
5     Public Enum SecurityLicenseTypes
6         ' Token: 0x04000013 RID: 19
7         Permanent
8         ' Token: 0x04000014 RID: 20
9         Temporary
10        ' Token: 0x04000015 RID: 21
11        Training
12        ' Token: 0x04000016 RID: 22
13        Invalid
14    End Enum
15 End Namespace
16

```

Activada de forma permanente, temporal , entrenamiento e invalida

respecto a las licencias, se aprecian

```

SecurityProductAvailabilityGroups X
1 Imports System
2
3 Namespace CLinxSecurity
4     ' Token: 0x02000008 RID: 8
5     Public Enum SecurityProductAvailabilityGroups
6         ' Token: 0x04000009 RID: 9
7         Evaluation
8         ' Token: 0x0400000A RID: 10
9         FireAlarmLevel1 = 10
10        ' Token: 0x0400000B RID: 11
11        FireAlarmLevel2
12        ' Token: 0x0400000C RID: 12
13        FireAlarmLevel3
14        ' Token: 0x0400000D RID: 13
15        SuppressionLevel1 = 50
16        ' Token: 0x0400000E RID: 14
17        HybridLevel1 = 100
18        ' Token: 0x0400000F RID: 15
19        HybridLevel2
20        ' Token: 0x04000010 RID: 16
21        HybridLevel3
22        ' Token: 0x04000011 RID: 17
23        Fake = 900
24    End Enum
25 End Namespace
26

```

se logra apreciar un = 10, 50, 100, pero en los otros no se aprecia, si cambiamos la vista a il


```
// Token: 0x02000008 RID: 8
.class public auto ansi sealed CLinxSecurity.SecurityProductAvailabilityGroups
    extends [mscorlib]System.Enum
{
    // Fields
    // Token: 0x04000008 RID: 8
    .field public specialname rtspecialname int32 value__
    // Token: 0x04000009 RID: 9
    .field public static literal valuetype CLinxSecurity.SecurityProductAvailabilityGroups Evaluation = int32(0)
    // Token: 0x0400000A RID: 10
    .field public static literal valuetype CLinxSecurity.SecurityProductAvailabilityGroups FireAlarmLevel1 = int32(10)
    // Token: 0x0400000B RID: 11
    .field public static literal valuetype CLinxSecurity.SecurityProductAvailabilityGroups FireAlarmLevel2 = int32(11)
    // Token: 0x0400000C RID: 12
    .field public static literal valuetype CLinxSecurity.SecurityProductAvailabilityGroups FireAlarmLevel3 = int32(12)
    // Token: 0x0400000D RID: 13
    .field public static literal valuetype CLinxSecurity.SecurityProductAvailabilityGroups SuppressionLevel1 = int32(50)
    // Token: 0x0400000E RID: 14
    .field public static literal valuetype CLinxSecurity.SecurityProductAvailabilityGroups HybridLevel1 = int32(100)
    // Token: 0x0400000F RID: 15
    .field public static literal valuetype CLinxSecurity.SecurityProductAvailabilityGroups HybridLevel2 = int32(101)
    // Token: 0x04000010 RID: 16
    .field public static literal valuetype CLinxSecurity.SecurityProductAvailabilityGroups HybridLevel3 = int32(102)
    // Token: 0x04000011 RID: 17
    .field public static literal valuetype CLinxSecurity.SecurityProductAvailabilityGroups Fike = int32(900)
}
```

se pueden apreciar los numeros de int, asociados a las licencias, asumiendo que las de menor acceso serían las de evaluación y la mejor la de fike

si pulsas analizar donde es usado la licencia, aquí tenemos el programa en si, para ser modificado o bien registrado.

```
SecurityProductAvailabilityGroups @02000008
└─ Exposed By
└─ Extension Methods
└─ Used By
    CLinxSecurity.SecurityModule.ActivateLicense(String) : SecurityActivationResults @06000017
    CLinxSecurity.SecurityModule.CheckPAGCompatibility(SecurityProductAvailabilityGroups) : Boolean @0600001B
    CLinxSecurity.SecurityModule.ClearLicenseInformation() : Void @06000019
    CLinxSecurity.SecurityModule.ExtractLicenseInformation(String) : Void @06000018
    CLinxSecurity.SecurityModule.GenerateLicense(String, SecurityLicenseTypes, DateTime, DateTime, DateTime, SecurityProductAvailabilityGroups) : String @0600001F
    CLinxSecurity.SecurityModule.IsEvalLicense : Boolean @17000013
    CLinxSecurity.SecurityModule.IsFikeEmployee : Boolean @17000010
    CLinxSecurity.SecurityModule.IsFireLicense : Boolean @17000014
    CLinxSecurity.SecurityModule.IsLicensedForSmokeControl : Boolean @17000011
    CLinxSecurity.SecurityModule.IsLicensedForVoice : Boolean @17000012
    CLinxSecurity.SecurityModule.IsSuppressionLicense : Boolean @17000015
    CLinxSecurity.SecurityModule.m_productAvailabilityGroup : SecurityProductAvailabilityGroups @04000029
    CLinxSecurity.SecurityModule.PAGDescription(SecurityProductAvailabilityGroups) : String @0600001A
    CLinxSecurity.SecurityModule.ProductAvailabilityGroup : SecurityProductAvailabilityGroups @1700000B
    Fike.CLinX.My.MyApplication.MyApplication_Startup(Object, StartupEventArgs) : Void @06000003
    Fike.CLinX.PanelExplorerForm.m_SerialProtocol_PanelsFound(Object, EventArgs) : Void @06001F6A
    Fike.CLinX.ProjectExplorerForm.ProjectExplorerForm(SingleBoardPanelCollection, Form) : Void @06002169
    Fike.CLinX.SplashScreen.SplashScreen_Load(Object, EventArgs) : Void @06002346
    Fike.CLinX.StartupForm.StartupForm(Boolean) : Void @0600236A
    Fike.CLinX.UserMainForm.FileSave(String, Boolean) : Void @060023EC
    Fike.CLinX.UserMainForm.ProcessFile(String) : Void @060023E9
```

Aquí me detengo un poco, hay que revisar todas esas funciones entiendo que para la vista se verá un poco largo, pero dejemos algún comentario de mas o menos que se ve, al menos para poder conocer cómo funcionan las validaciones, esto significa si voy por el lado de parchar, tendría un día completo o más, y si vamos por el lado de registrar, deberemos comprender cómo valida los valores de regedit y que condiciones necesita. veamos de a poco estos lugares y cómo puede afectar:


```

Public Function ActivateLicense(licenseData As String) As SecurityActivationResults
    Dim result As SecurityActivationResults
    If String.IsNullOrEmpty(licenseData) Then
        result = SecurityActivationResults.CodeInvalid
    ElseIf Operators.CompareString(licenseData, Me.m_license, False) = 0 OrElse Operators.CompareString(licenseData,
        Me.m_templicense, False) = 0 Then
        result = SecurityActivationResults.CodeUsed
    ElseIf Me.LoadLicenseInfo(licenseData) Then
        Dim num As Long = Math.Abs(DateAndTime.DateDiff(DateInterval.Day, Me.m_licenseIssueDate, DateAndTime.Now.[Date],
            FirstDayOfWeek.Sunday, FirstWeekOfYear.Jan1))
        If num > 14L OrElse (num > 7L And Me.IsLicenseTemporary) Then
            result = SecurityActivationResults.CodeExpired
            Me.LoadCurrentLicense()
        ElseIf Me.m_productAvailabilityGroup = SecurityProductAvailabilityGroups.Evaluation Then
            result = SecurityActivationResults.ActivationFailed
            Me.LoadCurrentLicense()
        Else
            If Me.IsLicenseTemporary Then
                result = SecurityActivationResults.TempActivated
                Me.m_templicense = licenseData
            Else
                result = SecurityActivationResults.Activated
                Me.m_license = licenseData
                Me.m_templicense = String.Empty
            End If
            Me.m_lastRunDate = DateAndTime.Now.[Date]
            Me.SaveToRegistry()
        End If
    Else

```

el primero sobre activarte, se pueden apreciar los diversos niveles de licencia y uso de savetoregistry que ya vimos, hace cálculos de días, con un límite de días.

```

Public Function CheckPAGCompatibility(pag As SecurityProductAvailabilityGroups) As Boolean
    Dim result As Boolean = False
    Dim productAvailabilityGroup As SecurityProductAvailabilityGroups = Me.m_productAvailabilityGroup
    If productAvailabilityGroup <= SecurityProductAvailabilityGroups.FireAlarmLevel3 Then
        If productAvailabilityGroup <> SecurityProductAvailabilityGroups.Evaluation Then
            Select Case productAvailabilityGroup
                Case SecurityProductAvailabilityGroups.FireAlarmLevel1
                    If pag <> SecurityProductAvailabilityGroups.Evaluation Then
                        If pag <> SecurityProductAvailabilityGroups.FireAlarmLevel1 Then
                            If pag = SecurityProductAvailabilityGroups.HybridLevel1 Then
                                result = True
                            End If
                        Else
                            result = True
                        End If
                    Else
                        result = True
                    End If
                End If
            End If

```

respecto a PAG Compatibility, es claramente según el tipo de licencia, le permitiría acceder a leer los archivos, manejando cómo un boolean (true o false) hay opciones de eliminar licencia

```

Private Sub ClearLicenseInformation()
    Me.m_licenseVersion = String.Empty
    Me.m_licenseType = SecurityLicenseTypes.Invalid
    Me.m_licenseStartDate = New DateTime(632716704000000000L)
    Me.m_licenseEndDate = New DateTime(632716704000000000L)
    Me.m_licenseIssueDate = New DateTime(632716704000000000L)
    Me.m_productAvailabilityGroup = SecurityProductAvailabilityGroups.Evaluation
End Sub

```

esto significa que además de validar que sean válidos algunos datos, también puede dejar sin licencia al usuario.

luego tenemos

```

Private Sub ExtractLicenseInformation(decodeLicense As String)
    Me.m_licenseVersion = String.Concat(New String() { decodeLicense.Substring(0, 1), ".", decodeLicense.Substring(1, 1),
    ".", decodeLicense.Substring(2, 1), ".", decodeLicense.Substring(3, 1) })
    Dim left As String = decodeLicense.Substring(4, 1)
    If Operators.CompareString(left, "0", False) <> 0 Then
        If Operators.CompareString(left, "1", False) <> 0 Then
            If Operators.CompareString(left, "2", False) <> 0 Then
                Me.m_licenseType = SecurityLicenseTypes.Invalid
            Else
                Me.m_licenseType = SecurityLicenseTypes.Training
            End If
        Else
            Me.m_licenseType = SecurityLicenseTypes.Temporary
        End If
    Else
        Me.m_licenseType = SecurityLicenseTypes.Permanent
    End If
    Me.m_licenseStartDate = New DateTime(Conversions.ToInteger("20" + decodeLicense.Substring(10, 2)), Conversions.ToInteger(
    decodeLicense.Substring(6, 2)), Conversions.ToInteger(decodeLicense.Substring(8, 2)))
    Me.m_licenseEndDate = New DateTime(Conversions.ToInteger("20" + decodeLicense.Substring(16, 2)), Conversions.ToInteger(
    decodeLicense.Substring(12, 2)), Conversions.ToInteger(decodeLicense.Substring(14, 2)))
    Me.m_licenseIssueDate = New DateTime(Conversions.ToInteger("20" + decodeLicense.Substring(22, 2)), Conversions.ToInteger(
    decodeLicense.Substring(18, 2)), Conversions.ToInteger(decodeLicense.Substring(20, 2)))
    Me.m_productAvailabilityGroup = CType(Conversions.ToInteger(decodeLicense.Substring(24, 3)),
    SecurityProductAvailabilityGroups)
End Sub

```

la extracción de licencia, esto significa que lee porciones de la licencia en cierto formato, a primera impresión esto verifica si es invalida, entrenamiento , temporal o permanente. hay una función llamada decodeLicense...la siguiente es para no creerlo

```

' Token: 0x0600001F RID: 31 RVA: 0x00003720 File Offset: 0x00001920
Public Function GenerateLicense(licVer As String, licType As SecurityLicenseTypes, startDate As DateTime, endDate As
    DateTime, issueDate As DateTime, pag As SecurityProductAvailabilityGroups) As String
    Dim text As String = licVer
    Select Case licType
        Case SecurityLicenseTypes.Permanent
            text += "0"
        Case SecurityLicenseTypes.Temporary
            text += "1"
        Case SecurityLicenseTypes.Training
            text += "2"
    End Select
    text = String.Concat(New String() { text, "0", startDate.ToString("MMddyy"), endDate.ToString("MMddyy"),
    issueDate.ToString("MMddyy") })
    If pag <= SecurityProductAvailabilityGroups.FireAlarmLevel3 Then
        If pag <> SecurityProductAvailabilityGroups.Evaluation Then
            Select Case pag
                Case SecurityProductAvailabilityGroups.FireAlarmLevel1
                    text += "010"
                Case SecurityProductAvailabilityGroups.FireAlarmLevel2
                    text += "011"
                Case SecurityProductAvailabilityGroups.FireAlarmLevel3
                    text += "012"
            End Select
        Else
            text += "000"
        End If
    ElseIf pag <> SecurityProductAvailabilityGroups.SuppressionLevel1 Then
        Select Case pag
            Case SecurityProductAvailabilityGroups.HybridLevel1
                text += "100"
            Case SecurityProductAvailabilityGroups.HybridLevel2

```

se aprecia una función generate license, literalmente permitiría generar una licencia válida con los parámetros, aquí es donde nos detendremos para resolver el programa, es increíble que un programa tenga además un keygen incorporado...el autor nos ha ganado con la idea de generar el keygen

veamos Lic ver es el tipo de licencia ,lictype, es la fecha , con parámetros start y end, luego issuedate además, y la última la capacidad de la licencia, ahora bien esto tiene además de cases hay algunas funciones .

```

        text += "101"
    Case SecurityProductAvailabilityGroups.HybridLevel3
        text += "102"
    Case Else
        If pag = SecurityProductAvailabilityGroups.Fike Then
            text += "900"
        End If
    End Select
Else
    text += "050"
End If
text += "002"
Dim text2 As String = Me.Encode(text)
If text2 Is Nothing Then
    text2 = "Unable to Generate License"
End If
Return text2
End Function

```

ese encode no me suena cómo algo genérico.

```

Private Function Encode(rawData As String) As String
    Dim result As String = Nothing
    If rawData.Length = 30 Then
        Dim flag As Boolean = True
        Dim num As Integer = 0
        While Operators.CompareString(rawData.Substring(num, 1), "0", False) >= 0 And Operators.CompareString(
            (rawData.Substring(num, 1), "9", False) <= 0
            num += 1
            If num > 29 Then
                IL_55:
                If flag Then
                    Dim text As String = String.Empty
                    text += rawData.Substring(5, 5)
                    text += rawData.Substring(15, 5)
                    text += rawData.Substring(0, 5)
                    text += rawData.Substring(25, 5)
                    text += rawData.Substring(10, 5)
                    text += rawData.Substring(20, 5)
                    Dim str As String = Me.CalculateLicenseCRC(text)
                    text += str
                    Dim num2 As Integer = 0
                    Dim text2 As String = String.Empty
                    Dim num3 As Integer = 0
                    Do
                        Dim text3 As String = text.Substring(num3, 1)
                        Dim num4 As UInteger = <PrivateImplementationDetails>.ComputeStringHash(text3)
                        If num4 <= 873244444UI Then
                            If num4 <= 822911587UI Then
                                If num4 <> 806133968UI Then

```

no me equivoqué, hay bastante que explorar, esto de encode genera un valor de 30 de largo, además refiere calcular crc

```

' Token: 0x0600001C RID: 28 RVA: 0x0000365A File Offset: 0x0000185A
Private Function CalculateLicenseCRC(reorderData As String) As String
    Return Strings.Format(Me.CalculateCRC(reorderData), "00000")
End Function

```

```

' Token: 0x0600001E RID: 30 RVA: 0x000036BC File Offset: 0x000018BC
Private Function CalculateCRC(msg As String) As Integer
    Dim num As Integer = 0
    Dim num2 As Integer = msg.Length - 1
    For i As Integer = 0 To num2
        Dim b As Byte = CByte(Strings.Asc(msg.Substring(i, 1)))
        Dim num3 As Integer = (num Xor CInt(b)) And 15
        num = (num / 16 Xor num3 * 4225)
        num3 = ((num Xor CInt((b / 16))) And 15)
        num = (num / 16 Xor num3 * 4225)
    Next
    Return num
End Function

```

la cual tiene un largo de 5, esto significa que el serial es de $30+5=$ largo de 35

si lográramos codear todo esto, tendríamos un keygen...pero aun así falta ver el computer hash

```

' Token: 0x0600002C RID: 44 RVA: 0x00003A20 File Offset: 0x00001C20
Friend Shared Function ComputeStringHash(s As String) As UInteger
    Dim num As UInteger = 2166136261UI
    If s IsNot Nothing Then
        For i As Integer = 0 To s.Length - 1
            num = (CUInt(s(i)) Xor num) * 16777619UI
        Next
    End If
    Return num
End Function
End Class

```

a primera impresión intentaré hacer algo en visual studio, pero sin embargo tengo algun error, para evitar perder más tiempo **he pedido ayuda a mi buen amigo Karma, el cual tiene mucha experiencia en .net (codeando), así que podrá arreglar los errores, que uno no logra ver a simple vista.**

siguiendo, con las funciones,

```

' Token: 0x0600001A RID: 26 RVA: 0x0000333C File Offset: 0x0000153C
Public Shared Function PAGDescription(availabilityGroup As SecurityProductAvailabilityGroups) As String
    Dim result As String = ""
    If availabilityGroup <= SecurityProductAvailabilityGroups.FireAlarmLevel3 Then
        If availabilityGroup <> SecurityProductAvailabilityGroups.Evaluation Then
            Select Case availabilityGroup
                Case SecurityProductAvailabilityGroups.FireAlarmLevel1
                    result = "Fire Alarm Level 1"
                Case SecurityProductAvailabilityGroups.FireAlarmLevel2
                    result = "Fire Alarm Level 2"
                Case SecurityProductAvailabilityGroups.FireAlarmLevel3
                    result = "Fire Alarm Level 3"
            End Select
        Else
            result = "Evaluation Software"
        End If
    ElseIf availabilityGroup <> SecurityProductAvailabilityGroups.SuppressionLevel1 Then
        Select Case availabilityGroup
            Case SecurityProductAvailabilityGroups.HybridLevel1
                result = "Fire Alarm Level 1 and Suppression"
            Case SecurityProductAvailabilityGroups.HybridLevel2
                result = "Fire Alarm Level 2 and Suppression"
            Case SecurityProductAvailabilityGroups.HybridLevel3
                result = "Fire Alarm Level 3 and Suppression"
            Case Else
                If availabilityGroup = SecurityProductAvailabilityGroups.Fike Then
                    result = "Fike Employee"
                End If
            End Select
        Else
            result = "Commercial Suppression"
        End If
    End If
    Return result

```

aquí podemos apreciar el nombre de cada licencia

```

' Token: 0x06000003 RID: 3 RVA: 0x00002074 File Offset: 0x00000274
Private Sub MyApplication_Startup(sender As Object, e As StartupEventArgs)
    SecurityModule.GetSecurityModule.LoadCurrentLicense()
    Dim splashScreen As SplashScreen = New SplashScreen(True)
    splashScreen.Show()
    While Not splashScreen.IsDisposed
        MyProject.Application.DoEvents()
    End While
    Me.StartupDirectory = AppDomain.CurrentDomain.BaseDirectory
    Try
        For Each text As String In MyProject.Application.CommandLineArgs
            If text.EndsWith(".lnx") Then
                Me.StartupFilename = text
                Me.StartupType = StartupTypes.StartupNone
            End If
        Next
    Finally
        Dim enumerator As IEnumerator(Of String)
        If enumerator IsNot Nothing Then
            enumerator.Dispose()
        End If
    End Try
    If Operators.CompareString(Me.StartupFilename, "", False) = 0 Then
        If MySettingsProperty.Settings.ShowStartupScreen Then
            Dim showOpen As Boolean = SecurityModule.GetSecurityModule.ProductAvailabilityGroup >
                SecurityProductAvailabilityGroups.Evaluation
            Dim startupForm As StartupForm = New StartupForm(showOpen)
            startupForm.ShowDialog()
            Me.StartupType = startupForm.StartupType
            startupForm.Dispose()
            Return
        End If
        Me.StartupType = StartupTypes.StartupNone
    End If

```

en el inicio de la aplicación está el splash, el cual según la licencia será si hay que hacer click o no, y hay forms que son activados según el tipo de licencia

```

Token: 0x0600236A RID: 9066 RVA: 0x0019C220 File Offset: 0x0019A420
Public Sub New(showOpen As Boolean)
    Me.m_startupType = StartupTypes.StartupNone
    Me.InitializeComponent()
    Dim productAvailabilityGroup As SecurityProductAvailabilityGroups =
        SecurityModule.GetSecurityModule.ProductAvailabilityGroup
    If productAvailabilityGroup <= SecurityProductAvailabilityGroups.FireAlarmLevel3 Then
        If productAvailabilityGroup = SecurityProductAvailabilityGroups.Evaluation Then
            Me.cmdNewCheetahXi.Visible = False
            Me.lblNewCheetahXi.Visible = False
            Me.cmdNewCheetahXi50.Visible = False
            Me.lblNewCheetahXi50.Visible = False
            Me.cmdNewCC1016.Top = Me.cmdNewCC1016.Top - 70
            Me.lblNewCC1016.Top = Me.lblNewCC1016.Top - 70
            Me.cmdNewCC254.Top = Me.cmdNewCC254.Top - 70
            Me.lblNewCC254.Top = Me.lblNewCC254.Top - 70
            Me.cmdNewCC50.Top = Me.cmdNewCC50.Top - 70
            Me.lblNewCC50.Top = Me.lblNewCC50.Top - 70
            Me.cmdOpenProject.Top = Me.cmdOpenProject.Top - 70
            Me.lblOpenProject.Top = Me.lblOpenProject.Top - 70
            Me.cmdMainMenu.Top = Me.cmdMainMenu.Top - 70
            Me.lblMainMenu.Top = Me.lblMainMenu.Top - 70
            Me.grpGetStarted.Height = Me.grpGetStarted.Height - 70
            MyBase.Height -= 45
            GoTo IL_53B
        End If
        If productAvailabilityGroup = SecurityProductAvailabilityGroups.FireAlarmLevel1 <= 2 Then
            Me.cmdNewCheetahXi.Visible = False
            Me.lblNewCheetahXi.Visible = False
            Me.cmdNewCheetahXi50.Visible = False
            Me.lblNewCheetahXi50.Visible = False
            Me.cmdNewCC1016.Top = Me.cmdNewCC1016.Top - 70
            Me.lblNewCC1016.Top = Me.lblNewCC1016.Top - 70

```

habiendo varios paneles a considerar

```
' Token: 0x0600236B RID: 9067 RVA: 0x0019C7D7 File Offset: 0x0019A9D7
Private Sub cmdNewCheetahXi_Click(sender As Object, e As EventArgs)
    Me.m_startupType = StartupTypes.StartupCreateXi1016
    MyBase.Close()
End Sub
```

```
' Token: 0x0600236C RID: 9068 RVA: 0x0019C7E6 File Offset: 0x0019A9E6
Private Sub cmdNewCC1016_Click(sender As Object, e As EventArgs)
    Me.m_startupType = StartupTypes.StartupCreateCc1016
    MyBase.Close()
End Sub
```

```
' Token: 0x0600236D RID: 9069 RVA: 0x0019C7F5 File Offset: 0x0019A9F5
Private Sub cmdNewCC254_Click(sender As Object, e As EventArgs)
    Me.m_startupType = StartupTypes.StartupCreateCc254
    MyBase.Close()
End Sub
```

```
' Token: 0x0600236E RID: 9070 RVA: 0x0019C804 File Offset: 0x0019AA04
Private Sub cmdNewCheetahXi50_Click(sender As Object, e As EventArgs)
    Me.m_startupType = StartupTypes.StartupCreateXi50
    MyBase.Close()
End Sub
```

```
' Token: 0x0600236F RID: 9071 RVA: 0x0019C813 File Offset: 0x0019AA13
Private Sub cmdNewCC50_Click(sender As Object, e As EventArgs)
    Me.m_startupType = StartupTypes.StartupCreateCc50
    MyBase.Close()
End Sub
```

```
' Token: 0x06002370 RID: 9072 RVA: 0x0019C822 File Offset: 0x0019AA22
Private Sub cmdOpenProject_Click(sender As Object, e As EventArgs)
    Me.m_startupType = StartupTypes.StartupOpenProject
    MyBase.Close()
```

en resumen

```

1 Imports System
2
3 Namespace Fike.CLinx.My
4     ' Token: 0x02000005 RID: 5
5     Public Enum StartupTypes
6         ' Token: 0x0400000C RID: 12
7         StartupNone
8         ' Token: 0x0400000D RID: 13
9         StartupCreateCc1016
0         ' Token: 0x0400000E RID: 14
1         StartupCreateCc254
2         ' Token: 0x0400000F RID: 15
3         StartupCreateCc50
4         ' Token: 0x04000010 RID: 16
5         StartupCreateXi1016
6         ' Token: 0x04000011 RID: 17
7         StartupCreateXi254
8         ' Token: 0x04000012 RID: 18
9         StartupCreateXi50
0         ' Token: 0x04000013 RID: 19
1         StartupOpenProject
2     End Enum
3 End Namespace
4

```

tenemos diferentes paneles y opciones pero además estas opciones

```

' Token: 0x06001F6A RID: 8042 RVA: 0x00171778 File Offset: 0x0016F978
Private Sub m_SerialProtocol_PanelIsFound(sender As Object, e As EventArgs)
    Dim flag As Boolean = False
    If Me.SerialProtocolExists Then
        Dim productAvailabilityGroup As SecurityProductAvailabilityGroups =
            SecurityModule.GetSecurityModule.ProductAvailabilityGroup
        If productAvailabilityGroup <= SecurityProductAvailabilityGroups.FireAlarmLevel3 Then
            If productAvailabilityGroup = SecurityProductAvailabilityGroups.Evaluation Then
                flag = True
                GoTo IL_279
            End If
            If productAvailabilityGroup - SecurityProductAvailabilityGroups.FireAlarmLevel1 <= 2 Then
                If Me.m_serialProtocol.PanelType = SingleBoardPanelTypes.CyberCat1016 Or Me.m_serialProtocol.PanelType =
                    SingleBoardPanelTypes.CyberCat254 Or Me.m_serialProtocol.PanelType = SingleBoardPanelTypes.CyberCat50 Or
                    Me.m_serialProtocol.PanelType = SingleBoardPanelTypes.HandHeldPanel Or Me.m_serialProtocol.PanelType =
                    SingleBoardPanelTypes.DemoPanel Then
                    Me.m_serialProtocol.Panel = New SingleBoardPanel(Me.m_serialProtocol.PanelType, 1)
                    Me.m_serialProtocol.Panel.FirmwareVersion = Me.m_serialProtocol.PanelFirmwareVersion
                    Me.m_serialProtocol.Panel.CommStatus = DeviceCommunicationStatus.StatusCommDeviceNotUploaded
                    Me.m_serialProtocol.Panel.PanelIsOnline = True
                    GoTo IL_279
                End If
                flag = True
                GoTo IL_279
            End If
            ElseIf productAvailabilityGroup <> SecurityProductAvailabilityGroups.SuppressionLevel1 Then
                If productAvailabilityGroup - SecurityProductAvailabilityGroups.HybridLevel1 <= 2 Then
                    Me.m_serialProtocol.Panel = New SingleBoardPanel(Me.m_serialProtocol.PanelType, 1)
                    Me.m_serialProtocol.Panel.FirmwareVersion = Me.m_serialProtocol.PanelFirmwareVersion
                    Me.m_serialProtocol.Panel.CommStatus = DeviceCommunicationStatus.StatusCommDeviceNotUploaded
                    Me.m_serialProtocol.Panel.PanelIsOnline = True
                End If
            End If
        End If
    End Sub

```

tienen relación al tipo de licencia, por lo tanto hay paneles donde se comunica siendo los paneles


```

Namespace Fike.Clinx
    ' Token: 0x020000A3 RID: 163
    Public Enum SingleBoardPanelTypes As Byte
        ' Token: 0x040012C7 RID: 4807
        None
        ' Token: 0x040012C8 RID: 4808
        CyberCat1016 = 10
        ' Token: 0x040012C9 RID: 4809
        CyberCat254
        ' Token: 0x040012CA RID: 4810
        CheetahXi254
        ' Token: 0x040012CB RID: 4811
        DemoPanel = 35
        ' Token: 0x040012CC RID: 4812
        HandHeldPanel
        ' Token: 0x040012CD RID: 4813
        CyberCat50 = 50
        ' Token: 0x040012CE RID: 4814
        CheetahXi50
        ' Token: 0x040012CF RID: 4815
        CheetahXi1016 = 78
    End Enum
End Namespace

```

un mensaje si estuviera activado, pero con una licencia inválida:

```

Dim method As PanelExplorerForm.SerialProtocolPanelFoundHandler = AddressOf Me.PanelIsFound
MyBase.Invoke(method, New Object() { e })
Else
    Interaction.MsgBox("Connection was made to a Panel that is incompatible with the License Product Group.
    Panel Explorer will close.", MsgBoxStyle.OkOnly, "Invalid Panel Type")
    Dim method2 As PanelExplorerForm.PanelExplorerCloseHandler = AddressOf Me.PanelExplorerClose
    MyBase.Invoke(method2, New Object() { e })
End If
Return
Catch ex As Exception
Return

```

respecto a los proyectos internos, también verifica cada grupo de licencia y esto permite ver si permite más opciones, por ejemplo el de guardar o manipular los grid

```

' (remove) Token: 0x06002168 RID: 8552 RVA: 0x001844E8 File Offset: 0x001826E8
Public Event ProjectExplorerSelectedItemChanged As ProjectExplorerForm.ProjectExplorerSelectedItemChangedEventHandler

' Token: 0x06002169 RID: 8553 RVA: 0x00184520 File Offset: 0x00182720
Public Sub New(netDevs As SingleBoardPanelCollection, parentForm As Form)
    AddHandler MyBase.FormClosing, AddressOf Me.ProjectExplorerForm_FormClosing
    AddHandler MyBase.Load, AddressOf Me.ProjectExplorerForm_Load
    Me.m_RefreshGrid = False
    Me.m_RightMouse = False
    Me.m_GridCellInvalid = False
    Me.m_DragOverCounter = 0
    Me.m_DragOverDirection = 0
    Me.m_DragOverPreviousNode = Nothing
    Me.m_DragOverPreviousRow = -1
    Me.m_DragOverCtrlKey = False
    Me.m_ClipBoard = Nothing
    Me.m_ClipBoardCut = False
    Me.m_Communicating = False
    Me.InitializeComponent()
    Me.AddImagesFromResource()
    Me.NetDevices = netDevs
    MyBase.MdiParent = parentForm
    CType(parentForm, UserMainForm).mnuViewProjectExplorer.Checked = True
    Dim productAvailabilityGroup As SecurityProductAvailabilityGroups =
        SecurityModule.GetSecurityModule.ProductAvailabilityGroup
    If productAvailabilityGroup <= SecurityProductAvailabilityGroups.SuppressionLevel1 Then
        If productAvailabilityGroup <> SecurityProductAvailabilityGroups.Evaluation Then

```

respecto a otro más:

```
Token: 0x06002346 RID: 9030 RVA: 0x001982B4 File Offset: 0x001994B4
Private Sub SplashScreen_Load(sender As Object, e As EventArgs)
    Me.Timer.Enabled = False
    If String.IsNullOrEmpty(MyProject.Application.Info.Title) Then
        Me.ApplicationTitle.Text = Path.GetFileNameWithoutExtension(MyProject.Application.Info.AssemblyName)
    Else
        Me.ApplicationTitle.Text = MyProject.Application.Info.Title
    End If
    Me.ApplicationTitle1.Text = "Panel Configuration Software"
    Me.ProductAvailabilityGroup.Text = SecurityModule.PAGDescription
    (SecurityModule.GetSecurityModule.ProductAvailabilityGroup)
    If SecurityModule.GetSecurityModule.IsEvalLicense Then
        Me.ProductAvailabilityGroup.BackColor = Color.Red
        Me.ProductAvailabilityGroup.ForeColor = Color.White
    End If
    Me.LicenseDaysLeft.Text = SecurityModule.GetSecurityModule.DaysRemaining.ToString() + " Days Remain"
    If SecurityModule.GetSecurityModule.DaysRemaining <= 30L Then
        Me.LicenseDaysLeft.BackColor = Color.Red
        Me.LicenseDaysLeft.ForeColor = Color.White
        Me.LicenseDaysLeft.Text = Me.LicenseDaysLeft.Text + " - Click to continue"
    End If
    Me.Version.Text = String.Format(Me.Version.Text, New Object() { MyProject.Application.Info.Version.Major,
        MyProject.Application.Info.Version.Minor, MyProject.Application.Info.Version.Build,
        MyProject.Application.Info.Version.Revision })
    Me.Copyright.Text = MyProject.Application.Info.Copyright
    If SecurityModule.GetSecurityModule.IsLicenseTemporary Then
        Me.Misc.Text = "Temporary License in Effect"
    Else
        Me.Misc.Text = MyProject.Application.Info.Trademark
    End If
End Sub
```

el splash según los días que quedan , exige o no click para continuar

```
End If
If SecurityModule.GetSecurityModule.IsEvalLicense Then
    Me.m_AutoClose = False
End If
If SecurityModule.GetSecurityModule.DaysRemaining <= 30L Then
    Me.m_AutoClose = False
End If
If Me.m_AutoClose Then
    Me.Timer.Interval = 3000
    Me.Timer.Enabled = True
End If
End Sub
```

en el caso de querer cerrar automáticamente, vemos que hay un timer que maneja un tiempo de 3000 milisegundos.

por otro lado

```

StartupForm X
386 <MethodImpl(MethodImplOptions.Synchronized)>
387 Set(value As Button)
388     Dim value2 As EventHandler = AddressOf Me.cmdNewCheetahXi50_Click
389     Dim cmdNewCheetahXi As Button = Me._cmdNewCheetahXi50
390     If cmdNewCheetahXi IsNot Nothing Then
391         RemoveHandler cmdNewCheetahXi.Click, value2
392     End If
393     Me._cmdNewCheetahXi50 = value
394     cmdNewCheetahXi = Me._cmdNewCheetahXi50
395     If cmdNewCheetahXi IsNot Nothing Then
396         AddHandler cmdNewCheetahXi.Click, value2
397     End If
398 End Set
399 End Property
400
401 ' Token: 0x0600236A RID: 9066 RVA: 0x0019C220 File Offset: 0x0019A420
402 Public Sub New(showOpen As Boolean)
403     Me.m_startupType = StartupTypes.StartupNone
404     Me.InitializeComponent()
405     Dim productAvailabilityGroup As SecurityProductAvailabilityGroups =
406         SecurityModule.GetSecurityModule.ProductAvailabilityGroup
407     If productAvailabilityGroup <= SecurityProductAvailabilityGroups.FireAlarmLevel3 Then
408         If productAvailabilityGroup = SecurityProductAvailabilityGroups.Evaluation Then
409             Me.cmdNewCheetahXi.Visible = False
410             Me.lblNewCheetahXi.Visible = False
411             Me.cmdNewCheetahXi50.Visible = False
412             Me.lblNewCheetahXi50.Visible = False

```

startup, verifica los niveles de licencia para ver los menús visibles, tal cual cómo lo hace el panel , por lo tanto hay más de una verificación cómo se puede apreciar. luego quedarían los 2 menús el que abre el archivo y el que guarda el archivo

```

' Token: 0x060023EC RID: 9196 RVA: 0x0019F728 File Offset: 0x0019D928
Private Sub FileSave(fileName As String, Optional showConfirmation As Boolean = True)
    Dim fileStream As FileStream
    If File.Exists(fileName) Then
        fileStream = New FileStream(fileName, FileMode.Truncate, FileAccess.ReadWrite)
    Else
        fileStream = New FileStream(fileName, FileMode.OpenOrCreate, FileAccess.ReadWrite)
    End If
    Dim streamWriter As StreamWriter = New StreamWriter(fileStream)
    Me.NetDevices.ProjectFileName = Me.dlgSaveFile.FileName
    streamWriter.WriteLine("*****")
    streamWriter.WriteLine(" C-LINX PROJECT FILE")
    streamWriter.WriteLine("-----")
    streamWriter.WriteLine(" PROJECT SAVED ON " + DateAndTime.Now.ToLongDateString() + " AT " +
        DateAndTime.Now.ToLongTimeString())
    streamWriter.WriteLine(" PROJECT FILENAME: " + Me.dlgSaveFile.FileName)
    streamWriter.WriteLine("*****")
    streamWriter.WriteLine("")
    streamWriter.WriteLine("PRODUCTAVAILABILITYGROUP=" + Strings.Format(CInt
        (SecurityModule.GetSecurityModule.ProductAvailabilityGroup), "00000"))
    streamWriter.WriteLine("PRODUCTVERSION=" + Application.ProductVersion + "")
    If Me.NetDevices.Save(streamWriter) Then
        streamWriter.Close()
        fileStream.Close()
        fileStream = New FileStream(fileName, FileMode.Open)
        Dim streamReader As StreamReader = New StreamReader(fileStream)
        Dim num As Integer = CRCCalculator.CalculateCRC(streamReader)
        streamReader.Close()
        fileStream.Close()

```

el que guarda, deja claro el tipo de licencia que usamos, por lo tanto, un archivo lnx será leído según la licencia, además, usará un cálculo de crc, el cual ya aprendimos que son de 5 valores.

respecto a la version del programa lo extrae del mismo ejecutable

```
' (get) Token: 0x060005BB RID: 1467 RVA: 0x000109E4 File Offset: 0x0000EBE4
Public Shared ReadOnly Property ProductVersion As String
    Get
        Dim obj As Object = Application.internalSyncObject
        SyncLock obj
            If Application.productVersion Is Nothing Then
                Dim entryAssembly As Assembly = Assembly.GetEntryAssembly()
                If entryAssembly IsNot Nothing Then
                    Dim customAttributes As Object() = entryAssembly.GetCustomAttributes(GetType(
                        AssemblyInformationalVersionAttribute), False)
                    If customAttributes IsNot Nothing AndAlso customAttributes.Length > 0 Then
                        Application.productVersion = CType(customAttributes(0),
                            AssemblyInformationalVersionAttribute).InformationalVersion
                    End If
                End If
                If Application.productVersion Is Nothing OrElse Application.productVersion.Length = 0 Then
                    Application.productVersion = Application.GetAppFileVersionInfo().ProductVersion
                    If Application.productVersion IsNot Nothing Then
                        Application.productVersion = Application.productVersion.Trim()
                    End If
                End If
                If Application.productVersion Is Nothing OrElse Application.productVersion.Length = 0 Then
                    Application.productVersion = "1.0.0.0"
                End If
            End If
        End SyncLock
        Return Application.productVersion
    End Get
End Property
```

para finalizar la exploración:

```

└─ SecurityProductAvailabilityGroups @02000008
  └─ Exposed By
  └─ Extension Methods
  └─ Used By
    └─ CLinuxSecurity.SecurityModule.ActivateLicense(String) : SecurityActivationResults @06000017
    └─ CLinuxSecurity.SecurityModule.CheckPAGCompatibility(SecurityProductAvailabilityGroups) : Boolean @0600001B
    └─ CLinuxSecurity.SecurityModule.ClearLicenseInformation() : Void @06000019
    └─ CLinuxSecurity.SecurityModule.ExtractLicenseInformation(String) : Void @06000018
    └─ CLinuxSecurity.SecurityModule.GenerateLicense(String, SecurityLicenseTypes, DateTime, DateTime, SecurityProductAvailabilityGroups) : String @0600001F
    └─ CLinuxSecurity.SecurityModule.IsEvalLicense : Boolean @17000013
    └─ CLinuxSecurity.SecurityModule.IsFakeEmployee : Boolean @17000010
    └─ CLinuxSecurity.SecurityModule.IsFireLicense : Boolean @17000014
    └─ CLinuxSecurity.SecurityModule.IsLicensedForSmokeControl : Boolean @17000011
    └─ CLinuxSecurity.SecurityModule.IsLicensedForVoice : Boolean @17000012
    └─ CLinuxSecurity.SecurityModule.IsSuppressionLicense : Boolean @17000015
    └─ CLinuxSecurity.SecurityModule.m_productAvailabilityGroup : SecurityProductAvailabilityGroups @04000029
    └─ CLinuxSecurity.SecurityModule.PAGDescription(SecurityProductAvailabilityGroups) : String @0600001A
    └─ CLinuxSecurity.SecurityModule.ProductAvailabilityGroup : SecurityProductAvailabilityGroups @1700000B
    └─ Fake.CLinux.My.MyApplication.MyApplication_Startup(Object, StartupEventArgs) : Void @06000003
    └─ Fake.CLinux.PanelExplorerForm.m_SerialProtocol_PanelsFound(Object, EventArgs) : Void @06001F6A
    └─ Fake.CLinux.ProjectExplorerForm.ProjectExplorerForm(SingleBoardPanelCollection, Form) : Void @06002169
    └─ Fake.CLinux.SplashScreen.SplashScreen_Load(Object, EventArgs) : Void @06002346
    └─ Fake.CLinux.StartupForm.StartupForm(Boolean) : Void @0600236A
    └─ Fake.CLinux.UserMainForm.FileSave(String, Boolean) : Void @060023EC
    └─ Fake.CLinux.UserMainForm.ProcessFile(String) : Void @060023E9
```

la lectura del Inx: es en process File

```

        GoTo IL_856
    End If
    GoTo IL_813
ElseIf Operators.CompareString(text2, "7.4.0.0", False) <> 0 Then
    GoTo IL_856
End If
Else
    If Operators.CompareString(text2, "6.4.4.0", False) <> 0 Then
        GoTo IL_856
    End If
    GoTo IL_813
End If
ElseIf num2 <> 1518720536UI Then
    If num2 <> 1555045527UI Then
        If num2 <> 1644765708UI Then
            GoTo IL_856
        End If
        If Operators.CompareString(text2, "6.1.0.0", False) <> 0 Then
            GoTo IL_856
        End If
        GoTo IL_813
    ElseIf Operators.CompareString(text2, "7.5.0.0", False) <> 0 Then
        GoTo IL_856
    End If
Else
    If Operators.CompareString(text2, "3.1.1.0", False) <> 0 Then
        GoTo IL_856
    End If
    GoTo IL_813
End If

```

es capaz de identificar las versiones, y además de hacer una compatibilidad mediante "converted"

```

End If
IL_813:
Interaction.MsgBox(String.Concat(New String() { "The Project File '", filename, "' was saved with C-Linx
Version ", text2, ". In order to maintain future compatability, please save the project with this
version of C-Linx." })), MsgBoxStyle.Information, "File Save Needed")
flag4 = True
GoTo IL_88F
IL_856:
Interaction.MsgBox(String.Concat(New String() { "The Project File '", filename, "' was saved with C-Linx
Version ", text2, ". This project cannot be converted and will not be loaded. Save the file using C-
Linx Version 3.0.0.2 first, then load with this version." })), MsgBoxStyle.Information, "Cannot Load
Project File")
IL_88F:
If Not flag4 Then
    Exit While
End If
If Me.NetDevices.Load(streamReader, text2) Then
    Interaction.MsgBox("The Project File '" + filename + "' has been loaded.", MsgBoxStyle.Information,
"Open File")
    Me.pnlMain.Text = filename
    Me.ProjectExplorer = New ProjectExplorerForm(Me.NetDevices, Me)
    Me.ProjectExplorer.Show()
    Exit While
End If
Interaction.MsgBox("Unable to load the Project File '" + filename + "'.", MsgBoxStyle.Information, "Open
File")
Me.pnlMain.Text = filename
Exit While
ElseIf text.StartsWith("CONVERTED") Then
    flag2 = True
ElseIf text.StartsWith("PRODUCTAVAILABILITYGROUP") Then
    Dim num As Integer = 0

```

cuando el proyecto es correcto

```

If Not flag4 Then
    Exit While
End If
If Me.NetDevices.Load(streamReader, text2) Then
    Interaction.MsgBox("The Project File '" + filename + "' has been loaded.", MsgBoxStyle.Information,
        "Open File")
    Me.pnlMain.Text = filename
    Me.ProjectExplorer = New ProjectExplorerForm(Me.NetDevices, Me)
    Me.ProjectExplorer.Show()
Exit While
End If

```

este bloque hace su función,

```

' Token: 0x060023EA RID: 9194 RVA: 0x0019F701 File Offset: 0x0019D901
Private Sub FileMerge(fileName As String)
    Me.ProcessFile(fileName)
End Sub

' Token: 0x060023EB RID: 9195 RVA: 0x0019F70A File Offset: 0x0019D90A
Private Sub FileLoad(fileName As String)
    Me.NetDevices = Nothing
    Me.NetDevices = New SingleBoardPanelCollection()
    Me.ProcessFile(fileName)
End Sub

```

en el cual abrirá al panel indicado y guardado

respecto a los paneles guardados, o al ser cargado hay un nombre resumido

```

If Operators.CompareString(Strings.Trim(text), "", False) <> 0 And Not text.StartsWith("") Then
    If text.StartsWith("NEW") Then
        Dim num As Integer = 0
        num = text.IndexOf(":", num, StringComparison.Ordinal)
        Dim num2 As Integer = text.IndexOf(":", num + 1, StringComparison.Ordinal)
        Dim netDeviceAddress As Byte = Conversions.ToByte(text.Substring(num2 + 2, 3))
        Dim singleBoardPanel As SingleBoardPanel = Nothing
        Dim left As String = text.Substring(num + 1, num2 - num - 1)
        If Operators.CompareString(left, "NETCC1016", False) <> 0 Then
            If Operators.CompareString(left, "NETCC254", False) <> 0 Then
                If Operators.CompareString(left, "NETCC50", False) <> 0 Then
                    If Operators.CompareString(left, "NETCHEETAHXI", False) <> 0 Then
                        If Operators.CompareString(left, "NETXI254", False) <> 0 Then
                            If Operators.CompareString(left, "NETXI50", False) = 0 Then
                                If SecurityModule.GetSecurityModule.IsSuppressionLicense Then
                                    singleBoardPanel = New SingleBoardPanel(SingleBoardPanelTypes.CheetahXi
                                        netDeviceAddress)
                                End If
                            End If
                        End If
                    End If
                End If
            End If
        End If
    End If
End If

```

y cada uno de ellos, será abierto solo según en nivel de licencia, además de verificar en el puerto .

Ya con esto revisamos el funcionamiento de donde actúa la licencia y cómo es por dentro las validaciones, así que ahora manos a la obra, mostremos la experiencia de Karma, y con esto ya caerá

Dia 1:

luego de lo explorado y viendo el código en dnspy, modificando parámetros, se puede obtener una licencia



Dia2: Comienzo a codear las funciones en visual Studio

```

Friend Function Encode(rowData As String) As String
    Dim result As String = Nothing
    If rowData.Length = 30 Then
        Dim text As String = String.Empty
        text += rowData.Substring(5, 5)
        text += rowData.Substring(15, 5)
        text += rowData.Substring(0, 5)
        text += rowData.Substring(25, 5)
        text += rowData.Substring(10, 5)
        text += rowData.Substring(20, 5)
        Dim str As String = iUtils.CalculateLicenseCRC(text) '<<<<<<<---- Mal Calculo -(
        text += str
        Dim num2 As Integer = 0
        Dim text2 As String = String.Empty
        Dim num3 As Integer = 0
        Do
            Dim text3 As String = text.Substring(num3, 1)
            Dim num4 As UInteger = ComputeStringHash(text3)
            If num4 <= 873244444UI Then
                If num4 <= 822911587UI Then
                    If num4 <> 806133968UI Then
                        If num4 = 822911587UI Then
                            If Operators.CompareString(text3, "4", False) = 0 Then
                                Select Case num2
                                    Case 0
                                        text2 += "I"
                                    Case 1
                                        text2 += "J"
                                    Case 2
                                        text2 += "5"
                                End Select
                            End If
                        End If
                    End If
                End If
            End If
        Loop
    End If
End Function

```

ya está la función operativa, pero da errores de cálculo

```

Dim text3 As String = text.Substring(num3, 1)
Dim num4 As UInteger = ComputeStringHash(text3) '<<<<<<<<<---- Tambien daba Error
If num4 <= 873244444UI Then
    If num4 <= 822911587UIT Then

```

entonces, luego viene el tema operativo, el manejo de strings...

```

Friend Shared Function ComputeStringHash(s As String) As UInteger
    Dim num As UInteger = 2166136261UI
    If s IsNot Nothing Then
        For i As Integer = 0 To s.Length - 1
            num = (CUInt(s(i)) Xor num) * 16777619UI
        Next
    End If
End Function

```

al revisar el código estos resultados se desbordan, por lo tanto da otros valores, pero en el manejo real de la función de la string, trabaja 1 a 1, por lo cual karma hace la siguiente modificación:

```

Friend Function ComputeStringHash(ByVal s As Integer) As UInteger
    Select Case s
        Case 1 : Return 873244444
        Case 2 : Return 923577301
        Case 3 : Return 906799682
        Case 4 : Return 822911587
        Case 5 : Return 806133968
        Case 6 : Return 856466825
        Case 7 : Return 839689206
        Case 8 : Return 1024243015
        Case 9 : Return 1007465396
        Case 0 : Return 890022063
    End Select
End Function

```

luego queda corregir la segunda función:

```

' Token: 0x06000027 RID: 39 RVA: 0x00003844 File Offset: 0x00001A44
Private Function CalculateCRC(msg As String) As Integer
    Dim num As Integer = 0
    Dim num2 As Integer = msg.Length - 1
    For i As Integer = 0 To num2
        Dim b As Byte = CByte(Strings.Asc(msg.Substring(i, 1)))
        Dim num3 As Integer = (num Xor CInt(b)) And 15
        num = (num / 16 Xor num3 * 4225)
        num3 = ((num Xor CInt((b / 16))) And 15)
        num = (num / 16 Xor num3 * 4225)
    Next
    Return num
End Function

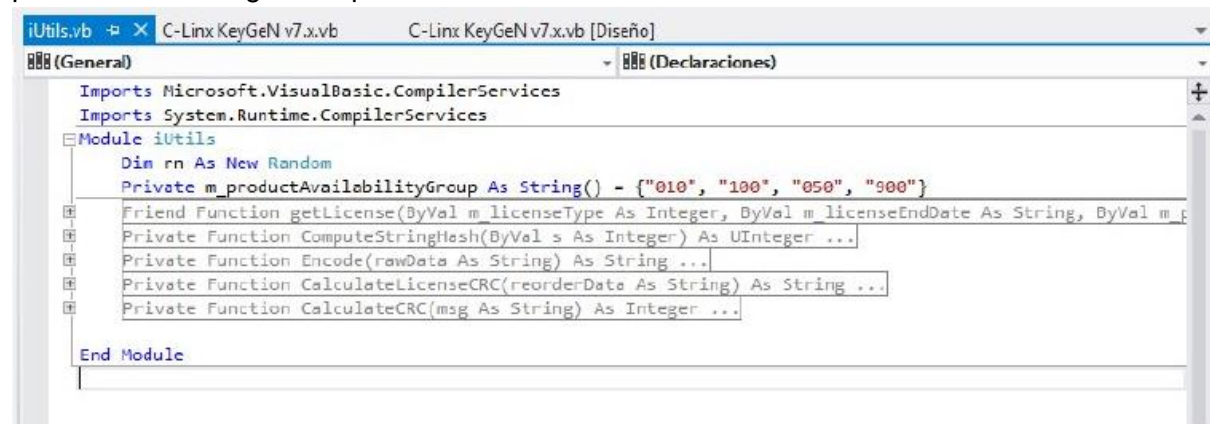
```

los valores en codificación quedan aproximados con valores .5 al valor siguiente se cambia al siguiente, siendo que el programa trabajaba con el valor truncado/redondeado, había que verificarlo

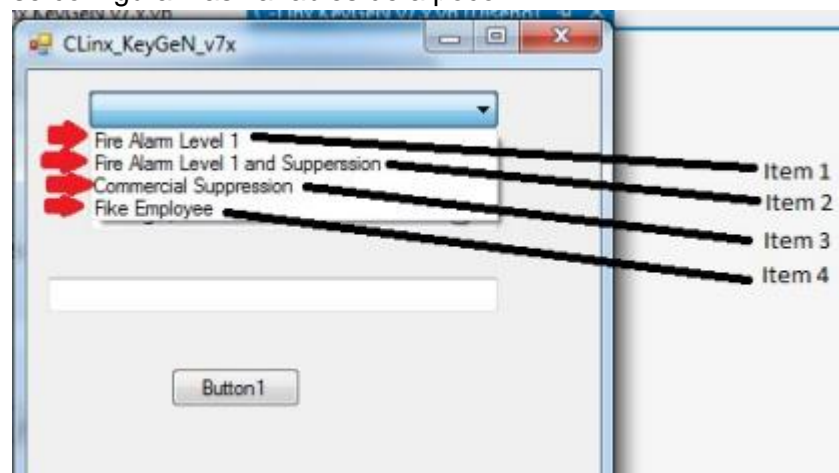
idea: 972.5 el programa redondeada o más bien operaba solo con el entero sin el punto decimal "972" así que busque forma de redondear el "972.5" pero al hacer eso pasaba a ser "973" prácticamente la misma solo que separa las operaciones y use la instrucción INT para tomar solo el valor antes del punto decimal y este hace el cálculo correcto

```
Private Function CalculateCRC(msg As String) As Integer
    Dim num As Integer = 0
    Dim num2 As Integer = msg.Length - 1
    For i As Integer = 0 To num2
        Dim b As Byte = CByte(Strings.Asc(msg.Substring(i, i + 1)))
        Dim num3 As Integer = (num Xor b) And 15
        num = Int(num / 16) Xor (num3 * 4225)
        num3 = (num Xor Int(b / 16)) And 15
        num = Int(num / 16) Xor Int(num3 * 4225)
    Next
    Return num
End Function
```

para lo demás luego será probar cómo van funcionando los valores

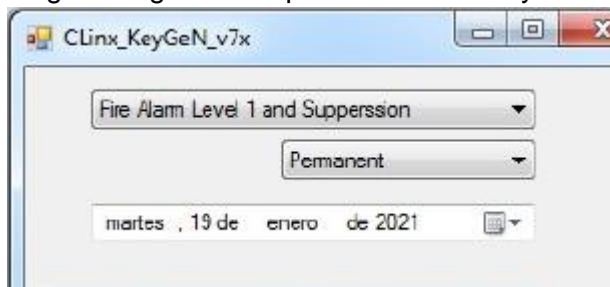


se configuran las variables de a poco





luego configurar los tipos de activación y la fecha final



para terminar la experiencia queda un keygen operativo:



se terminó en un keygen

probamos:



C-Linx

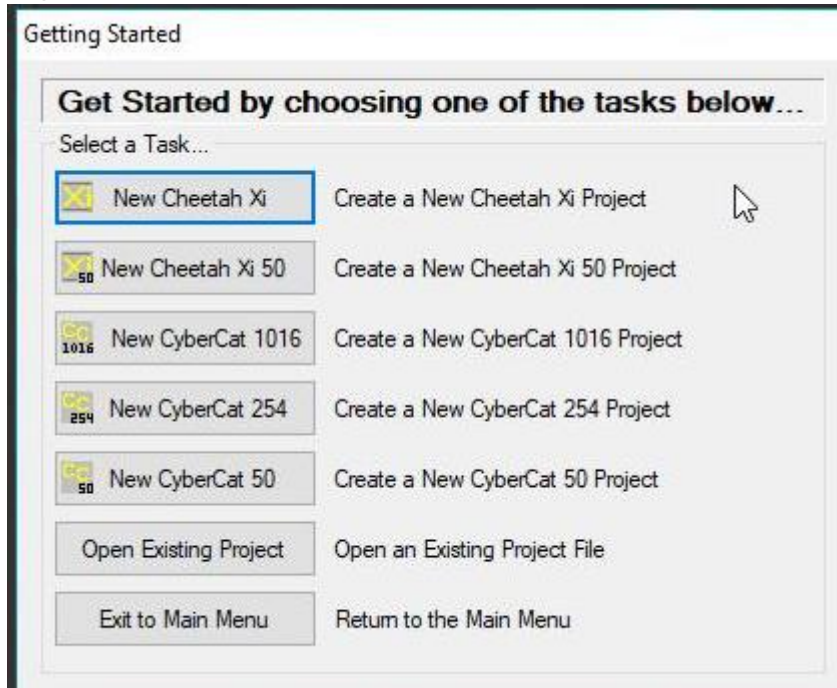
Panel Configuration Software

Fire Alarm Level 3 and Suppression

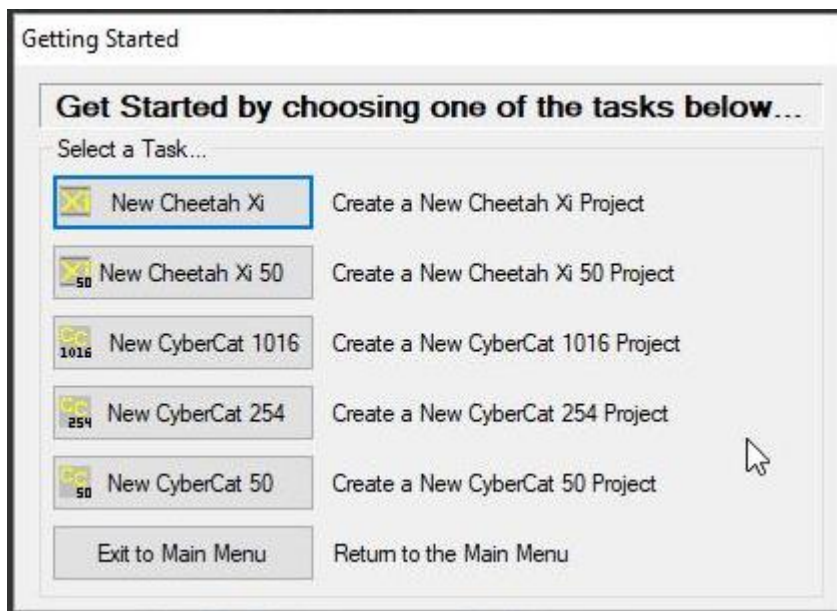
para las otras licencia solo cambia el nombre : 366 Days Remain

en temas generales ahi se aprecia el tipo de licencia y dias restantes, y se logran todos los niveles de licencia.

El panel ahora se ve así al iniciar con licencias de alto número:



si abrimos el proyecto existente y cerramos luego vemos así:



por lo tanto aplicación, Registrada y funcional.

Fike® is a registered trademark of Fike Corporation.

Resumen de la experiencia

Se logró el objetivo, y no se comparten licencias, debido a que no es de distribución pública el ejecutable ,solo se compartió al interesado :) que nos facilitó el ejecutable, espero le guste el tutorial.

Palabras finales

Pues la mejor experiencia de aprender es intencionado, no hacer por hacer, el éxito irá de la mano del compromiso que te propongas.

Mis Agradecimientos para quienes han logrado leer hasta aquí,les dejo cordialmente invitados a conocer el proceso de Ingeniería inversa a través de los manuales, maestros y conocimiento que existe hoy en día.

saludos cordiales a la lista de PeruCrackers, Crackslatinos, Apuromafo.net

Apuromafo CLS & KarMa [954] ©Full