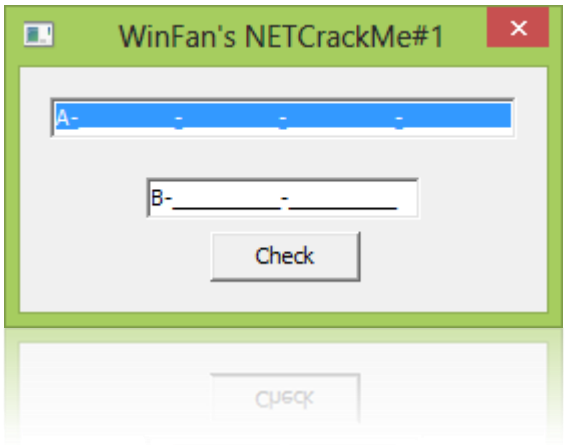
Keygen para el CrackmeMe#01 de WinFan

Sacándole jugo a un .Net



By deurus
27/08/2014

ÍNDICE

Equipo utilizado:

S.O: Windows 7 x32 /Windows 8 x64

Depurador: Ollydbg 1.10 (32bits) con plugins

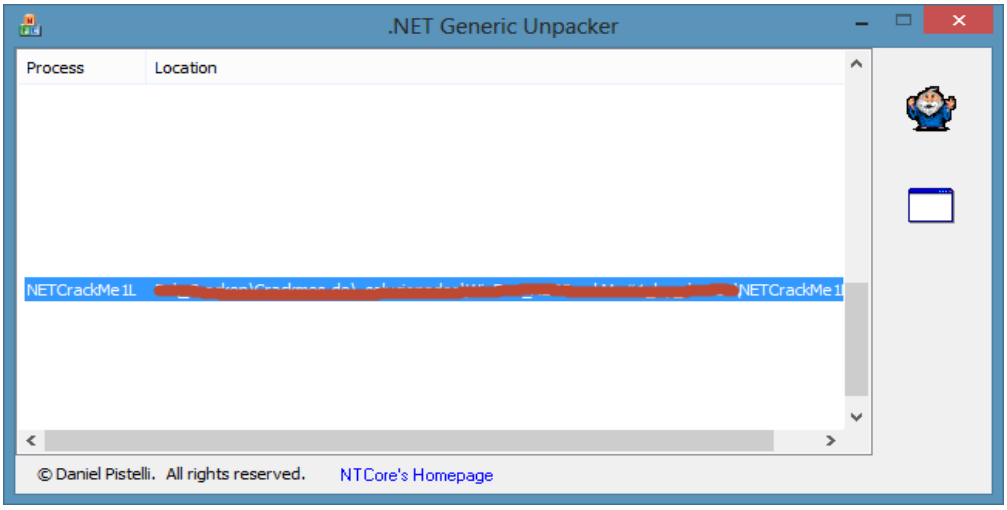
Analizador: PEiD 0.95

1. Introducción

Tal y como nos adelanta el creador está programado en .NET. Lo abrimos para ver su comportamiento y a simple vista ya vemos algo que no nos gusta, y es que se abre una ventana de DOS y posteriormente aparece el crackme. Esto indica que el ejecutable está escondido dentro de otro, empaquetado, encriptado o vete a saber.

2. Desempaquetado

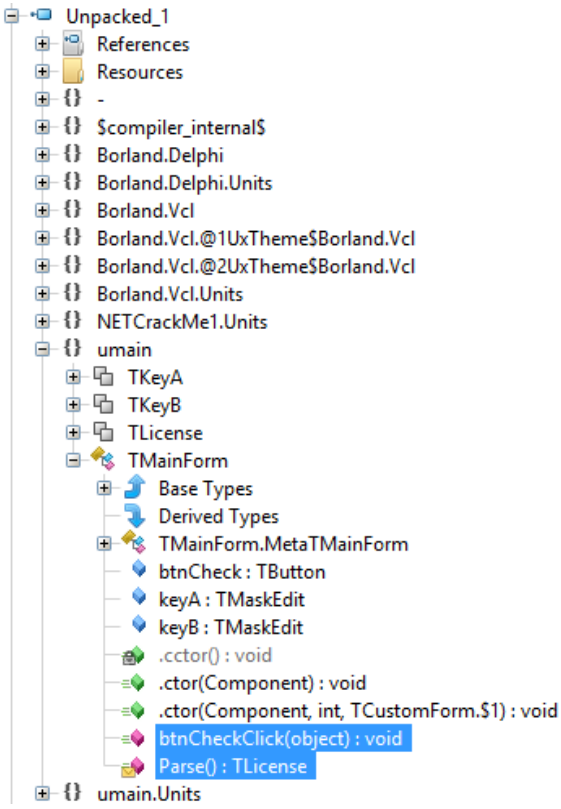
Nuestras sospechas eran ciertas, abrimos el executable con **ILSpy** y no encontramos lo que buscamos, pero si vemos que al assembly se le hace algo parecido a un XOR. Probemos con algo sencillo, abrimos el crackme y la herramienta **.Net Generic Unpacker** y probamos a desempaquetar.



Esto nos genera un par de "exes" que ahora si abre correctamente nuestro decompilador.

3. Decompilado

Vamos a fijarnos en la rutina de comprobación del serial. Lo interesante se encuentra en **btnCheckClick** y **TLicense**.



Código fuente.

```
// umain.TMainForm
public void btnCheckClick(object Sender)
{
    TLicense tLicense = this.Parse();
    if ((tLicense.a.a ^ tLicense.a.b) == tLicense.b.a && (tLicense.a.c ^ tLicense.a.d) == tLicense.b.b)
    {
        Windows.MessageBox(base.Handle, "Serial correct!", "", 64u);
    }
    else
    {
        Windows.MessageBox(base.Handle, "Wrong serial!", "", 16u);
    }
}

// umain.TMainForm
internal TLicense Parse()
{
    string s = this.keyA.Text;
    System.WStrDelete(ref s, 1, 2);
    s = SysUtils.StringReplace(s, " ", "", TReplaceFlags.rfReplaceAll);
    TLicense tLicense;
    tLicense.a.a = SysUtils.StrToInt(System.WStrCopy(s, 1, 8));
    System.WStrDelete(ref s, 1, 9);
    tLicense.a.b = SysUtils.StrToInt(System.WStrCopy(s, 1, 8));
    System.WStrDelete(ref s, 1, 9);
    tLicense.a.c = SysUtils.StrToInt(System.WStrCopy(s, 1, 8));
    System.WStrDelete(ref s, 1, 10);
    tLicense.a.d = SysUtils.StrToInt(s);
    string s2 = this.keyB.Text;
    System.WStrDelete(ref s2, 1, 2);
    s2 = SysUtils.StringReplace(s2, " ", "", TReplaceFlags.rfReplaceAll);
    tLicense.b.a = SysUtils.StrToInt(System.WStrCopy(s2, 1, 9));
    System.WStrDelete(ref s2, 1, 10);
    tLicense.b.b = SysUtils.StrToInt(s2);
    tLicense = tLicense;
    return tLicense;
}
```

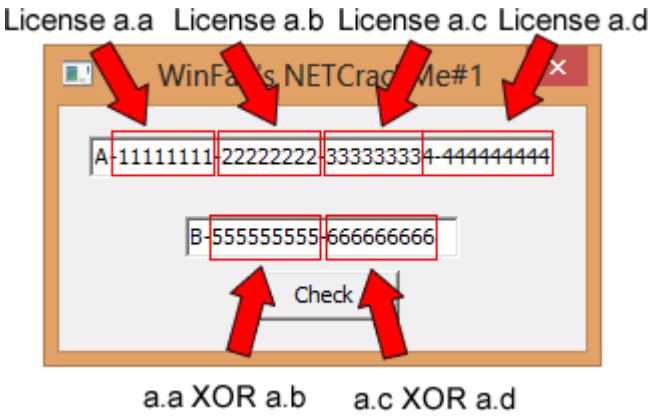
Como vemos en el código, License.a.a, License.a.b y License.a.c cogen 8 dígitos y License.a.d coge 10. A continuación comprueba que:

Licenseb.a = License.a.a XOR License.a.b

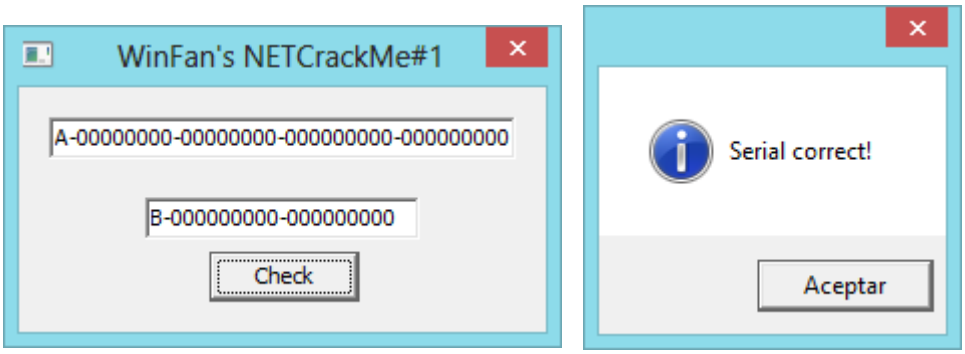
y que

Licenseb.b = License.a.c XOR License.a.d

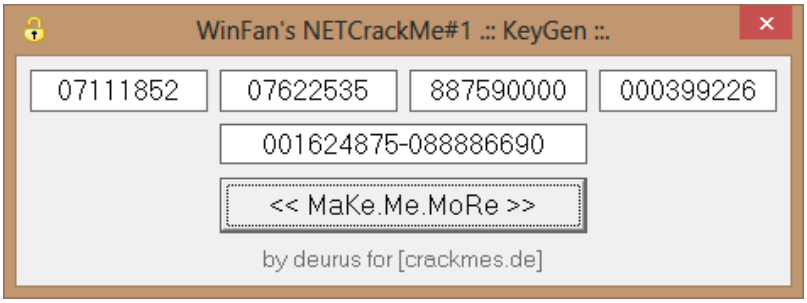
Una imagen vale más que mil palabras.



Como seguro que se os ha ocurrido, el xor de 0 es 0, lo que hace que un serial válido sea rellenar todo con ceros.



En su día hice un keygen, aquí tenéis una captura.



Podéis encontrar el crackme, mi solución y otras soluciones en crackmes.de.

4. Enlaces

- .Net Generic Unpacker
- ILSpy
- Crackme
- Entrada en el Blog

5. Crackeando Crackmes by deurus

- https://mega.co.nz/#F!88BRwYoT!O0TzTSZYCdczKLOrfrOyGw
- Lolabits.es/blogcracking (Clave: **blogcrackhack**)