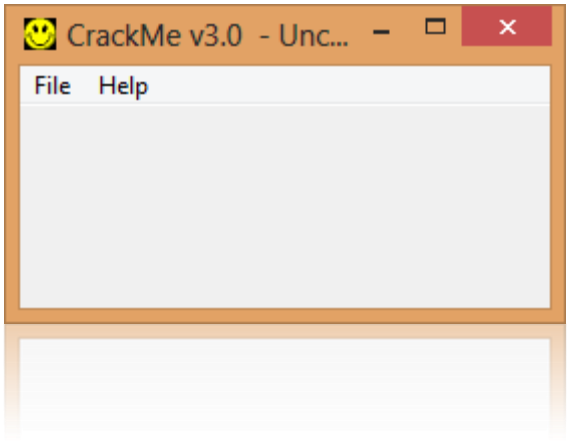
Keygen para el CrackMe#03 de Cruehead

Archivo llave



By deurus
08/09/2014

ÍNDICE

1. Introducción 2
2. El algoritmo 2
3. Recapitulando 4
4. Enlaces 6
5. Crackeando Crackmes by deurus 6

Equipo utilizado:

S.O: Windows 7 x32 /Windows 8 x64

Depurador: Ollydbg 1.10 (32bits) con plugins

Analizador: PEiD 0.95

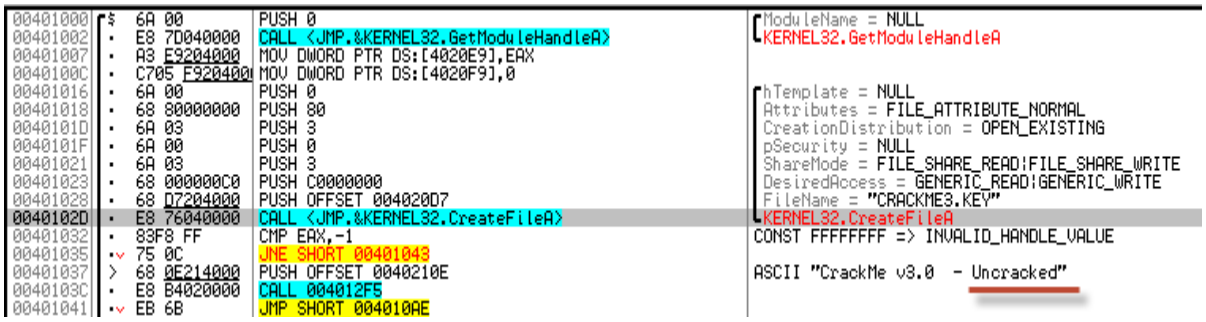
1. Introducción

Esta es la tercera y última entrega de los crackmes de **Cruehead**. En esta ocasión nos enfrentamos a un "**keyfile**", un **archivo llave** para que nos entendamos. Tiene un poco más de dificultad que los anteriores pero es ideal para los que empiezan.

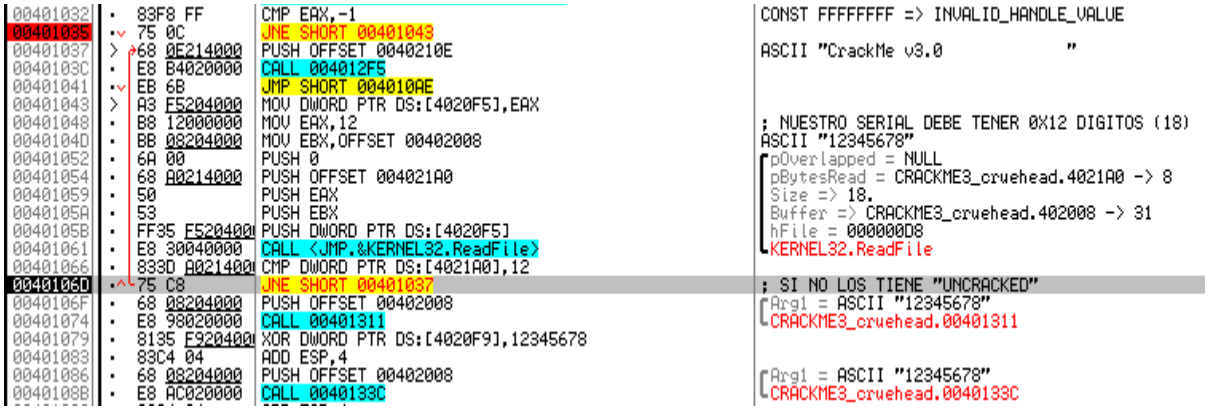
2. El algoritmo

Si iniciamos el crackme no pasa nada, lo único que vemos es la palabra "**UNCRACKED**" en el título. Abrimos el crackme con Olly y empezamos. En las "**string references**" vemos el nombre del archivo llave "**crackme3.key**". Lo creamos y escribimos el serial 12345678 y empezamos a tracear.

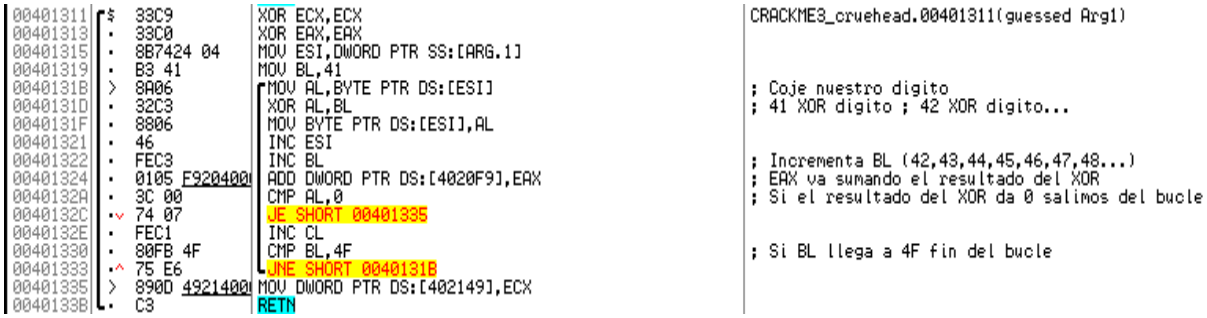
El **CMP EAX,-1** significa que está comprobando que el archivo no esté vacío, como no es nuestro caso continuamos.



A continuación vemos que compara nuestra longitud de serial con **0x12** (**18** en decimal). Nuestro serial tiene 8 dígitos así que nos tira fuera.



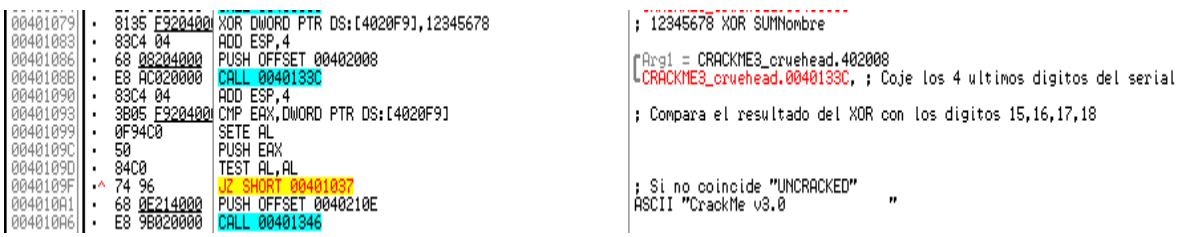
Escribimos en el archivo llave el serial "**deurus123456789012**" y volvemos a tracear. Vemos que ahora si pasa los filtros iniciales y llegamos a la primera zona interesante. En la imágen está explicado pero os hago un resumen. En el bucle lo que hace es un XOR a los primeros 14 dígitos de nuestro serial con los valores del 41 al 4E (4F finaliza). **El bucle solo se rompe si llegamos a 4F o si el resultado del XOR da 0**. Además en **EAX acumula la suma** del resultado **del XOR**.



Ejemplo:

1	d	e	u	r	u	s	1	2	3	4	5	6	7	8	9	0	1	2
2	64	65	75	72	75	73	31	32	33	34	35	36	37	38				
3																XOR		
4	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E				
5	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
6	25	27	36	36	30	35	76	7A	7A	7E	7E	7A	7A	76	=	4ED	(Suma)	

A continuación hace **XOR** entre **12345678** y **4ED**, coje los 4 últimos dígitos de nuestro serial y los compara.



Ejemplo:

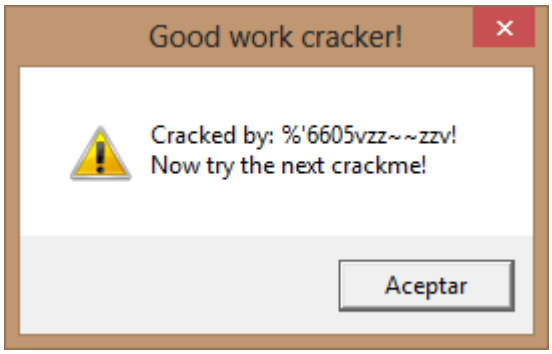
```
1 12345678 XOR 4ED = 12345295
2 Compara 12345295 con 32313039
3 32313039 = 2109, nuestros 4 últimos dígitos al revés. Recordemos que nuestro serial era "deurus123456789012"
```

El serial bueno para el nombre **deurus12345678** serían los bytes correspondientes de "12345295", es decir, nuestro serial bueno sería:

Ejemplo:

```
1 Necesitamos 12345295 para la comparación.
2 12 34 52 95 hexadecimal
3 18 52 82 149 decimal
4 Tenemos que escribirlo al revés. Con Alt-Izq + 149 escribimos el primer caracter, el resto igual.
5 Nuestro serial quedaría: deurus12345678òR4↕
```

Metemos el serial y vemos que lo acepta pero que nos muestra un nombre extraño. Esto es porque nos está mostrando **los bytes del nombre xoreados**, tendremos que hacer un XOR antes al nombre que queramos para que lo muestre correctamente.



3. Recapitulando

Con lo que sabemos ahora hay que empezar a coger el toro por los cuernos. Lo primero que queremos que muestre el nombre deurus y no deurus12345678. Para ello **debemos cortar el bucle** y eso solo lo podemos hacer forzando que el resultado del XOR sea 0. Ok pues para **deurus** el siguiente valor de BL, es decir el séptimo, en el bucle sería 47 lo que corresponde a la letra G. Pues si ponemos de serial **deurusGxxxxxxxxxx** ya tendríamos la primera parte solucionada.

Pero recordemos que necesitamos XORear el nombre inicialmente, luego debemos escribir el resultado del XOR.

Ejemplo:

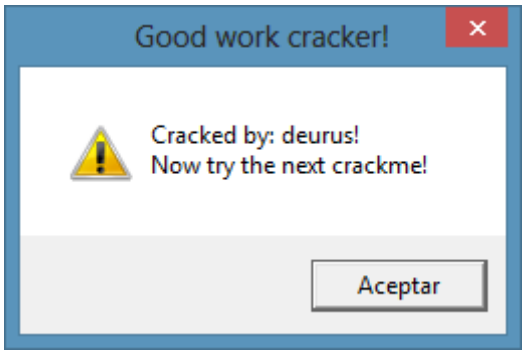
```
d  e  u  r  u  s
64 65 75 72 75 73
                  XOR
41 42 43 44 45 46
-----------------
25 27 36 36 30 35

25 27 36 36 30 35 ----- debemos meter esto en el archivo llave.
                  XOR
41 42 43 44 45 46
-----------------
64 65 75 72 75 73 ----- Al desencriptarlo el bucle se verá nuestro nombre correctamente.

En el archivo llave escribiriamos: %'6605
```

Ahora nos faltaría **calcular el nuevo SUM**. Como el resultado del XOR ahora es nuestro nombre, basta con sumar sus valores ascii (64+65+75+72+75+73 == 0x298)

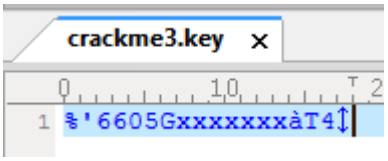
0x12345678 XOR 0x298 == 0x123454E0

Luego nuestros 4 últimos dígitos deben ser lo correspondiente a los bytes E0, 54, 34, 12. Los pasamos a decimal y los escribimos en el archivo llave con el truco del ALT-Izq que hemos comentado antes.

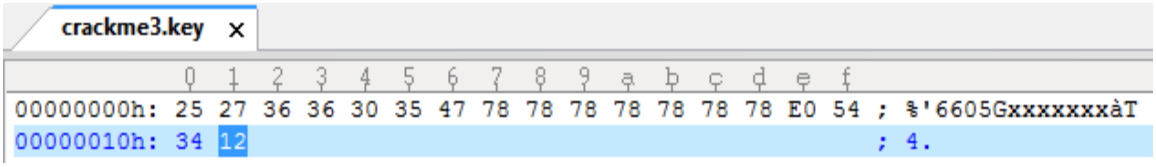
El contenido final del archivo llave para el nombre **deurus** sería: **%'6605GxxxxxxxÓT4↕**



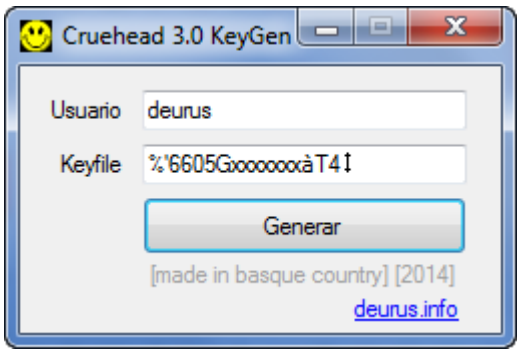
Aquí vemos el contenido del archivo llave normal.



Y aquí lo vemos con un editor hexadecimal. Como veis se ven claramente los bytes E0, 54, 34, 12.



Os dejo un keygen hecho en .Net para que probéis. Os genera el contenido del archivo y el archivo "**crackme3.key**".



4. Enlaces

- Crackme
- Keygen
- Cruehead's Crackme 1.0 Keygen [1/3]
- Cruehead's Crackme 2.0 Serial [2/3]

5. Crackeando Crackmes by deurus

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