SOLUCIÓN AL CRIPTO-RETO 4.
BUSCANDO UNA AGUJA EN UN PAJAR

31 de Diciembre de 2015 – Dr. Alfonso Muñoz @mindcrypt
(Co) Editor @criptored

- *¿Cómo han sabido que veníamos?*
- *Saben hasta lo que desayunamos.*
- *Entonces sólo hay una forma de vencerlos.*
- *¿Cómo?*
- *Averiguando lo que desayunan ellos.*

Michael Collins

Enunciado del Criptoreto4:
http://www.criptored.upm.es/paginas/criptoretos_info/criptoreto4.pdf
Texto incautado:
http://www.criptored.upm.es/paginas/criptoretos_info/TextoIncautado.txt
Imagen incautada:
http://www.criptored.upm.es/paginas/criptoretos_info/d1ec4c633664e2a02ab8fcec4eef1187.bmp
Crucigrama incautado:
http://www.criptored.upm.es/paginas/criptoretos_info/crucigrama.png

En las siguientes líneas se muestra una posible solución al criptoreto que diseñé y publiqué en Criptored. Este reto mezcla diversas técnicas criptográficas y esteganográficas. Su complejidad era media/alta por petición de anteriores participantes en los criptoretos publicados. Durante todo el proceso se fue proporcionando pistas para avanzar en el proceso cuando fue necesario.

Agradezco a todas las personas que han dedicado su tiempo, inteligencia y conocimiento a romper este "reto". Nos veremos en un futuro…

SOLUCIÓN

PASO 1. Para resolver el reto es importante observar la diferente información que se nos proporciona a modo de imágenes y textos interceptados.

PASO 2. El crucigrama incautado es en realidad un sudoku. Para ello es suficiente con realizar la siguiente sustitución: A=1, B=2, C=3, D=4, E=5, F=6, G=7, H=8, I=9. Una vez observado esto, el siguiente paso es resolver el sudoku.

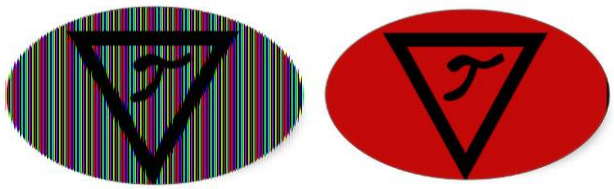
D								
A			E					B
					H		I	
			D					
	H						F	
		C	A					
		B			F			
						E		A
							G	D

4	9	5	3	2	7	8	1	6
1	6	8	**5**	9	4	7	3	**2**
3	2	7	6	1	**8**	4	**9**	5
9	7	1	**4**	6	3	2	5	8
2	**8**	4	9	7	5	1	**6**	3
6	5	**3**	**1**	8	2	9	4	7
5	1	**2**	7	4	**6**	3	8	9
7	4	6	8	3	9	**5**	2	**1**
8	3	9	2	5	1	6	**7**	**4**

PASO 3. Utilizamos el sudoku como una matriz que nos permitirá reordenar los píxeles de la imagen incautada. Para ello, se cogerán bloques de 9 bytes de manera consecutiva de la zona de pixeles y se reordenarán según indica el sudoku. El sudoku se recorre de izquierda a derecha y de arriba a abajo. Las matrices resultantes para reordenar son las siguientes:

```
grid1[]={4,9,5,1,6,8,3,2,7};
grid2[]={3,2,7,5,9,4,6,1,8};
grid3[]={8,1,6,7,3,2,4,9,5};
grid4[]={9,7,1,2,8,4,6,5,3};
grid5[]={4,6,3,9,7,5,1,8,2};
grid6[]={2,5,8,1,6,3,9,4,7};
grid7[]={5,1,2,7,4,6,8,3,9};
grid8[]={7,4,6,8,3,9,2,5,1};
grid9[]={3,8,9,5,2,1,6,7,4};
```

Es decir, abrimos el BMP, nos posicionamos en la zona de los píxeles, seleccionamos bloques de 9 en 9 bytes (pixel) y reordenamos. La imagen resultante sería:



PASO 4. Extraemos la información del bit menos significativo (técnica LSB – esteganografía) de cada byte (pixel)

```
01001110011010100101000101111001010011100101010001011001011
11100001001110001100100100111010011000001001110010101000100
00010011001101001110011010100100001001101100010011010111101
10010100010011000101001110011010100101000100110100010011101
01010100010110010011000101001110011010100101100100110100011
00111001010100010001010011000101001110011010010001010011001
10100111010111101001100011001100000100111001101010010101010
```

```
01100111001100100100111001010100010110010011010001001110010
10101001000001001100010100110101111010010110010111101001000
11100110101001001101001101000100111001010100011000110011000
10010011100110101001000101001100010100111001010100011001000
01101100010011100110101001001110011010110100110101111010010
00010101111010010011100110101001100011001101010100111001010
01000101000101111001010011100110101001000101011101110100110
10010101110101010111101001001110011010100100010101110111010
01001101001100100101001001101001010011100110101001001011010
10110101001101010101110101000100110010010011100110101001000
10010011000101001110010101000101100101111010010011100110100
10010011100110100101001101011110100101000100110000010011100
01101010011000110011000101001110010101110101011001101000010
00111001101101010001100110101001001110010101110101010100110
00010100111001101010010001010011000101001101011110100101000
01011110010100111001101010011010000110100001001110010101000
01010001011110000100111001101101010110010111100001001110010
01010001000001001101000100111001101010010010010011000001000
11010011001001000101011110100100111001101010011001000110100
01010011100101010001011001001101000100111001101101010010100
01101010010011100101011101010001001101000100111001101010010
00011001101001010011010111101001100011001100010100111000110
00100100100100110000010011100101010001100011001100110100110
10011010100100100100111010001001110010101000100110100110001
01001110011110100100100101110111010011010111101001000001010
11101001001110011010100100010100110001010011100101010001100
00110011010101001110011010100100100100110101010011100101010
00011010000110100101001110011011010101011001101000010011010
01111010010100100110100001001110011111010010000100110110100
```

PASO 5. Representamos la información en ASCII. Se observa un formato base64

```
NjQyNTYxN2M0NTA3NjBlMzQ1NjQ4NTY1NjY4NTE1NjE3Mzc0NjUzNTYyNj
g2NTY4NjA1MzYzNjM4NTc2NjE1NTdlNjNkMzEzNjc5NTQyNjEwNWUzNjEw
M2RiNjZmMWQ2NjI1NTYzNjNiMzQ0Njc1NWVhNmFjNWU1NjE1MzQyNjhhNT
QxNmYxNTA4NjI0M2EzNjdiNTY4NmJjNWQ4NjFiMzc1N2I0NTc3NjI4NTM1
NzIwMzAzNjE1NTc5NjI5NThiNmVhMzRhNzBm
```

PASO 6. Decodificamos la información obtenida (quitamos Base64) → 201 caracteres (3*67)

```
6425617c450760e345648565668515617374653562686568605363638 5
7661557e63d3136795426105e36103db66f1d662556363b3446755ea6a
c5e561534268a5416f15086243a367b5686bc5d861b3757b4577628535
72030361557962958b6ea34a70f
```

PASO 7. En este punto tenemos una información aparentemente cifrada o codificada que debemos invertir. Inicialmente se debe probar algoritmos criptográficos sencillos (criptografía clásica). De alguna manera tenemos que conseguir una clave utilizada en el proceso de cifrado.

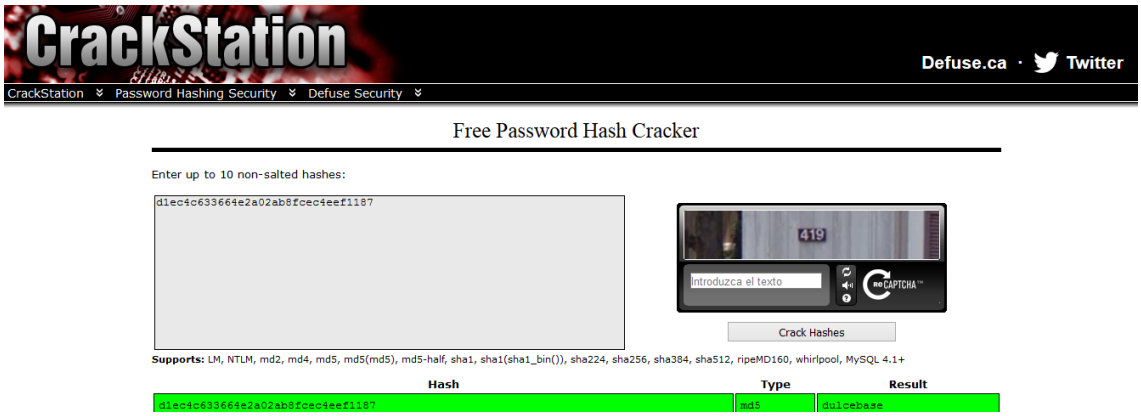
Entre la información incautada hay un fichero con un nombre un tanto particular:

d1ec4c633664e2a02ab8fcec4eef1187.bmp

d1ec4c633664e2a02ab8fcec4eef1187 = 32 bytes = 128 bits = es un código MD5

Intentamos invertir este código para ver qué información en claro protege:

https://crackstation.net/



PASO 8. Desciframos el mensaje utilizando la operación XOR y como clave la representación en hexadecimal de *dulcebase.* Esta clave podría haberse obtenido de otra manera observando que la imagen incautada es una imagen famosa de la base dulce (https://en.wikipedia.org/wiki/Dulce_Base).

```
String text=new
String("6425617c450760e345648565668515617374653562686568605363638
57661557e63d3136795426105e36103db66f1d662556363b3446755ea6ac5e5
61534268a5416f15086243a367b5686bc5d861b3757b45776285357203036155
7962958b6ea34a70f");

String clave=new
String("64756c63656261736564756c63656261736564756c63656261736564
756c63656261736564756c63656261736564756c63656261736564756c636562
61736564756c63656261736564756c63656261736564756c6365626173656475
6c636562617365647");

//d   u    l    c    e    b    a    s    e

//64 75 6c 63 65 62 61 73 65
```

Texto descifrado:

```
00500d1f206501902000f00905e0770001101400e0b000a01200607f01a0230
1c02a07603e02e0260810070be0284ba01300102c02103208609a0870020270c
d02d0c706a0330c603c00408a0ba00c0101f301b01e05713706605201501f0e9
0fd02f148
```

PASO 9. En este punto no hemos sido capaces de obtener un mensaje en claro. Todavía nos queda el texto incautado que podría servirnos como ayuda. Lo razonable sería probar algoritmos clásicos de criptografía para intentar obtener el mensaje en claro. Un posible camino para obtener el mensaje en claro, dado que no tenemos ninguna otra clave criptográfica, es suponer que el texto incautado “enmascara” de algún modo la clave necesaria para recuperar el mensaje en claro.

En concreto, el algoritmo utilizado para proteger el mensaje fue un cifrador por homófonos. El texto incautado se utiliza a modo de diccionario para recuperar el mensaje en claro.

PASO 10. El *Textoincautado.txt* tiene 16 palabras por frase (256 frases), lo que hace un total de 4096 palabras (que podríamos numerar de 000 a fff en hexadecimal). Con este diccionario obtendríamos que palabra corresponde a cada conjunto de 3 caracteres en el texto descifrado. Por ejemplo, 005 = have.

```
Primera frase: Life (000) on (001) other (002) Planets (003)
Meteorites (004) have (005)…
```

Haciendo pruebas se puede observar cómo seleccionando la primera letra de cada palabra se puede obtener una lista de nombres. Un ejemplo del diccionario de descodificación quedaría de la siguiente manera:

> **L**ife **o**n **o**ther **P**lanets **M**eteorites **h**ave **b**een **c**ollected **f**rom **t**he **i**ce **f**ields **o**f **A**ntarctica **a**nd **s**everal
>
> l,0,0,000
> o,0,1,001
> o,0,2,002
> p,0,3,003
> m,0,4,004
> h,0,5,005
> b,0,6,006
> c,0,7,007
> f,0,8,008
> t,0,9,009
> i,0,10,00a
> f,0,11,00b
> o,0,12,00c
> a,0,13,00d
> a,0,14,00e
> s,0,15,00f
>
> **o**f **t**hem **a**ppear **t**o **h**ave **c**ome **f**rom **M**ars. **T**race **e**lement **r**atios **s**uch **a**s **t**he **s**equence **o**f
>
> o,1,0,010
> t,1,1,011
> a,1,2,012
> t,1,3,013
> h,1,4,014
> c,1,5,015
> f,1,6,016
> m,1,7,017
> t,1,8,018
> e,1,9,019
> r,1,10,01a

s,1,11,01b
a,1,12,01c
t,1,13,01d
s,1,14,01e
o,1,15,01f

PASO 11. Recuperamos el texto en claro utilizando el diccionario anterior.

```
haydensteelthaliabernardbradclaytongaberowlingadambrookssarahcon
nor
```

PASO 12. Luego la lista de agentes es:

```
Hayden Steel
Thalia Bernard
Brad Clayton
Gabe Rowling
Adam Brooks
Sarah Connor
```