

Reto: La caja fuerte

18 de noviembre

2010

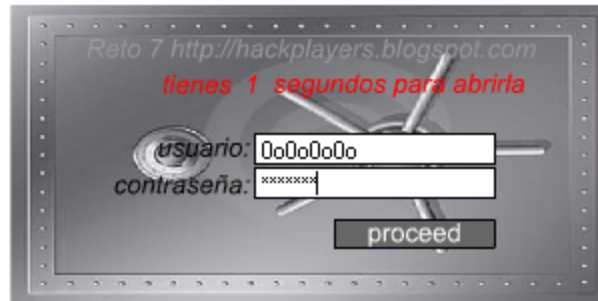
Por Daniel Correa

www.sinfocol.org

Descripción

Si habéis visto [Operación Swordfish \(2001\)](#), recordaréis la escena en que Stanley Jobson fue obligado a *hackear* una computadora del NSA en menos de un minuto y con una atractiva señorita "*desviando su atención*". En esta ocasión nuestro séptimo reto es algo parecido, aunque eso sí, sin distracciones añadidas ;-):

Simplemente, **tenéis 10 segundos para abrir la caja fuerte electrónica y obtener lo que se encuentra oculto en su interior.**

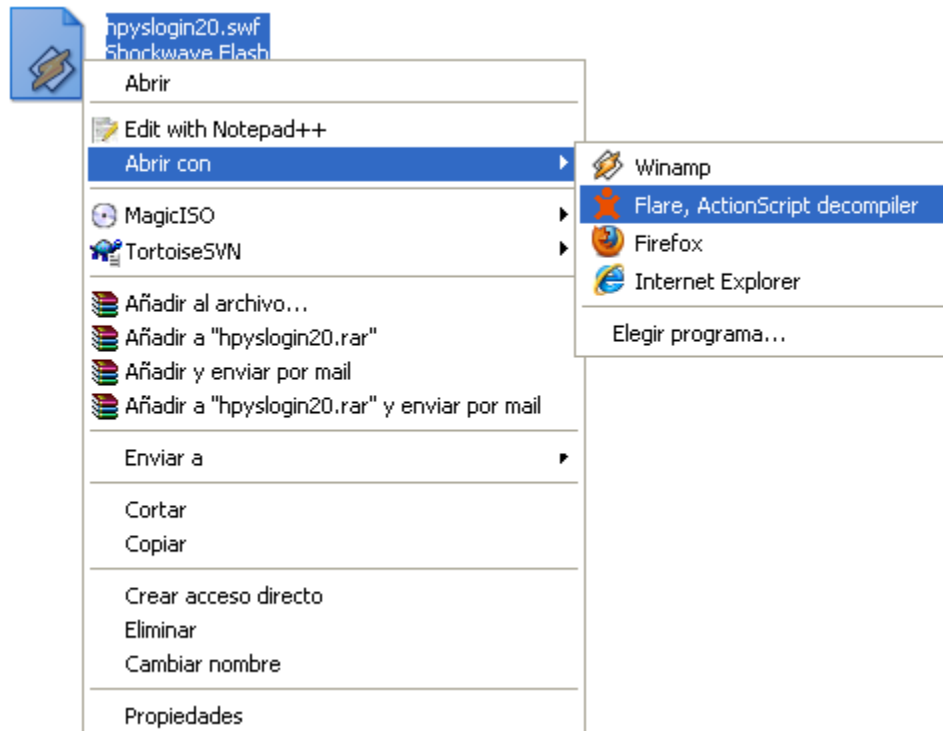


Resolución

Ingresamos al código fuente de la página para localizar la animación flash

`src="http://sites.google.com/site/h4ckpl4y3s/hpyslogin20.swf"`

Luego de descargarla, obtenemos su código fuente con [Flare](#) (Un decompilador gratuito para ActionScript)



Al generarse el archivo .flr podemos encontrar el código para el botón 13 también llamado “Proceed” y nos centramos en la operación de comparación

```
if (username == _0xb404[0] && password == _0xb404[1]) {  
    gotoAndPlay(26);  
    getURL(_0xb204, _0xb404[4]);  
} else {  
    gotoAndStop(25);  
}
```

Compara username con el valor del array _0xb404 en la posición 0 que pertenece a “admin7” y password con el valor del array _0xb404 en la posición 1 que pertenece a la variable _0xb346.

Si el usuario y contraseña son correctos hace un llamado al método getURL pasándole como parámetro la variable _0xb204 que corresponde al URL que va a obtener y el valor del array _0xb404 en la posición 4 que corresponde a “_blank” (Para abrir el documento en otra ventana).

En estos momentos podemos escoger dos caminos, pero ambos llegarán al mismo resultado.

El primer camino es averiguar la contraseña del usuario conociendo el valor de la variable _0xb346. El segundo es averiguar el valor de la variable _0xb204 para obtener la dirección suponiendo que la contraseña es correcta.

La ventaja en este punto es que ActionScript y JavaScript son ambas implementaciones del estándar [ECMA-262](#), de aquí que su sintaxis sea similar. Por lo que entonces podemos crear un documento HTML y en él, por medio de la etiqueta script, simular el comportamiento del código que tenemos.

En la función encode, debemos cambiar el nombre de la función “chr” por “String.fromCharCode”, comentamos las últimas líneas donde se encuentra la comparación, y digitamos los respectivos document.write donde vamos a mostrar el contenido de las variables.

```
document.write("La clave es: " + _0xb404[1]);  
document.write("<br />");  
document.write("El documento es: " + _0xb204);
```

El resultado es el siguiente:

La clave es: HpYs2k1O

El documento es: <http://sites.google.com/site/h4ckpl4y3s/faff625d3d5f937fcabb9dd5ffdc0238.zip>

Procedemos a descargar el documento, el cual es un archivo ZIP con contraseña, usamos a [google](#) para encontrar la primera contraseña que corresponde al texto plano “caenigma” del hash MD5 “faff625d3d5f937fcabb9dd5ffdc0238”

Al extraer el archivo que contiene el ZIP, nos encontramos con otro del mismo nombre y de nuevo con contraseña, pero no es problema ya que si observamos bien el código del ActionScript, hay una variable que no está siendo usada

```
google.com/site/h4ckp14y3s/', 'sZiPpwd', '_blank'];
```

Usamos “SZiPpWd” como contraseña y obtenemos un archivo llamado solucion.jpg



Usamos el servicio de [TinEye](https://www.tineye.com/) para encontrar la imagen original



JPEG, 400x207, 64.0 KB

244 Results

Searched over **1.7559 billion** images in 0.137 seconds.
for file: solucion.jpg

These results expire in 72 hours. [Why?](#)

[Share a success story!](#)

TinEye is **free** to use for non-commercial purposes.

[Download](#) the official TinEye extension for Firefox
with right-click functionality!



[Compare](#) | [Link](#)

JPEG Image
400x207, 28.0 KB

makenanasday.skynetblogs.be

[dyn003_original_400_207_pjpeg_2524271...](#)

<http://makenanasday.skynetblogs.be/archive-mont...>

Descargamos la imagen original y notamos una pequeña diferencia de peso

 solucion.jpg	1.558 KB	IrfanView JPG File
 dyn003_original_400_207_pj...	28 KB	IrfanView JPG File

Hacemos uso del ejecutable “compare” de la grandiosa herramienta [ImageMagick](#) para comparar pixel por pixel en busca de diferencias en los pixeles, aunque ya con la diferencia de peso podemos afirmar que el cambio no está en los pixeles, sino que se debe a algo que fue agregado al final del mismo.

```
C:\WINDOWS\system32\cmd.exe
C:\RetoCajaFuerte>compare solucion.jpg dyn003_original_400_207_pjpeg_2524271_b49c497fea542c2709c90f6e95142aa7.jpg salida.png
C:\RetoCajaFuerte>_
```

Y como lo afirmé anteriormente, la diferencia no se encuentra en los pixeles (Si existen diferencias en los pixeles entonces la herramienta lo indica cambiando el respectivo pixel a color rojo)



Comparamos binariamente los archivos

Solucion.jpg

00006db0	A7 87 E9 60 0A B0 9A C6 C2 8C 5A E8 E9 D5 E2 05	50é'. "0EÂ0Zéé0â.
00006dc0	A3 A5 6A 78 F8 78 D6 BC 70 05 58 89 76 ED AC 85	zYjxex0%p.X0v1-0
00006dd0	13 5C B5 03 AE 8D 52 20 3A C1 1A 68 68 0D 78 60	. \p.00R :Á.hh.x`
00006de0	4D 8D 36 3D EC 30 D6 BE D7 25 7F 2A 3F F1 70 62	M06=i00%*%0*?ñpb
00006df0	41 22 CE 18 62 56 58 64 F3 2A F5 76 24 13 A8 81	A"î.bVXdó*ðv\$. "0
00006e00	C7 48 03 85 30 01 FF D9 FF FB 52 C4 00 00 0A 28	CH.00.ÿÛyüRÄ... (
00006e10	2D 40 B4 F3 00 21 87 48 29 C3 36 10 01 FF FF FA	-0'ó.!0H)Ä6..ÿÿú
00006e20	B1 91 E3 CA 53 B1 93 B0 90 08 62 10 8B 10 B0 33	#0âÊS±0"0.b.0."3
00006e30	81 9C 30 D1 E2 C0 69 EF FB C1 F2 FC 1F 07 C1 00	000ÑâÀiüâü..Á.

dyn003_original_400_207_pjpeg_2524271_b49c497fea542c2709c90f6e95142aa7.jpg

00006db0	A7 87 E9 60 0A B0 9A C6 C2 8C 5A E8 E9 D5 E2 05	SQé\."0EÂZèéÔâ.
00006dc0	A3 A5 6A 78 F8 78 D6 BC 70 05 58 89 76 ED AC 85	£Ÿjxøx0%p.X0v1-0
00006dd0	13 5C B5 03 AE 8D 52 20 3A C1 1A 68 68 0D 78 60	.\µ.0QR :Â.hh.x`
00006de0	4D 8D 36 3D EC 30 D6 BE D7 25 7F 2A 3F F1 70 62	M06=i00%×%0*?ñpb
00006df0	41 22 CE 18 62 56 58 64 F3 2A F5 76 24 13 A8 81	A"Î.bVXdó*ôv\$. "0
00006e00	C7 48 03 85 30 01 FF D9 A0 46 1D F0 F5 BA 00 00	ÇH.00.ÿÛ F.88°..
00006e10	14 AE F4 41 36 4C 26 8E 34 97 0D 58 D9 BF C5 9A	.00A6L&040.XÛ¿ÂD
00006e20	FF FF D8 FF E0 00 10 4A 46 49 46 00 01 02 01 00	ÿÿ0ÿà..JFIF.....
00006e30	48 00 48 00 00 FF ED 0B 3E 50 68 6F 74 6F 73 68	H.H..ÿi.>Photosh

Observamos que las diferencias se empiezan a producir a partir del marcador del final de la imagen (EOI). Con nuestro editor hexadecimal preferido enviamos todo lo que está a partir del FFh D9h a un nuevo archivo al que llamaremos desconocido.hex.



Usamos la herramienta [TrID](#) para hacer un reconocimiento general sobre qué tipo de archivo es el que estamos tratando. En Linux la alternativa a usar sería el comando file.

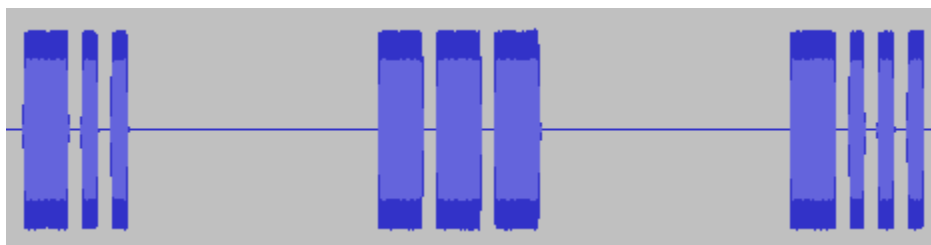
```
C:\RetoCajaFuerte>trid desconocido.hex

TrID/32 - File Identifier v2.02 - (C) 2003-06 By M.Pontello
Definitions found: 3879
Analyzing...

Collecting data from file: desconocido.hex
100.0% (.MP3) MP3 audio (1000/1)

C:\RetoCajaFuerte>
```

Renombramos nuestro archivo desconocido a desconocido.mp3 y escuchamos esa melodiosa armonía de beeeeps, no hay que ser un genio para saber que se trata de código Morse, así que podemos o analizar las ondas con un excelente programa llamado [Audacity](#) (Las ondas largas equivalen a “-” y las cortas a “.”)



O usar un programa llamado [MRP40 Morse Decoder](#) que se encarga del trabajo sucio




















La frase resultante es *“ni nos domaron, ni nos doblaron, ni nos van a domesticar - marcelino camacho, fundador CC.OO.”*

Referencias

- [ECMA-262](#), Estándar para lenguaje scripting.
- [Código Morse](#), método para transmitir información textual en forma de sonido.

Utilidades

Nombre	Atributo	Objetivo
Flare	 	Decompilador de ActionScript.
ImageMagick	  	Suite para crear, editar y componer imágenes.
TinEye	  	Buscador inverso online de imágenes.
TrID	 	Reconocimiento de tipos de archivos.
Audacity	  	Creación y edición de sonido.
UltraCompare	 	Comparación profesional de archivos.
MRP-40	 	Decodificación de ondas de sonido.