

Una-al-mes: La Casa de Papel (Episodio 1 - 1ª Parte)

Hemos conseguido entrar en la Fábrica Nacional de Moneda y Timbre. Pero una vez dentro, la lanza térmica que usaríamos para abrir la caja fuerte se ha roto. Debes descubrir los códigos para abrirla, y con ello conseguirás la contraseña para el zip del programa que genera la flag y el dinero ;).

Caja fuerte: <http://34.253.233.243/lacasadepapel/episodio1/puerta.php>

Info: La flag tiene el formato UAM{md5}

Resolución

Accedemos a la página, donde se nos solicitan dos códigos y un botón para comprobar los mismos. Damos un vistazo al código fuente, en el mismo, se hace mención en los comentarios a 1234/1234 y admin/admin, pero no dan resultado. Observamos que se “carga” un javascript (login.js) con código interesante.

```
var Password =
"unescape%28String.fromCharCode%252880%252C%2520108%252C%252097%252C%2
520110%2529%29:KZQWYZLOMNUWC===";
```

Esta variable se divide en dos partes, utilizando “:”, en los dos códigos que necesitamos.

```
Code1=unescape%28String.fromCharCode%252880%252C%2520108%252C%252097%2
52C%2520110%2529%29
```

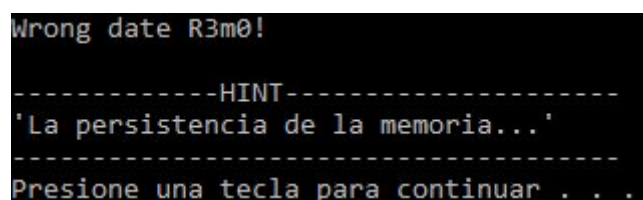
```
Code2 = KZQWYZLOMNUWC===
```

Desciframos código1. Doble decodificación URL y obtenemos `unescape(String.fromCharCode(80, 108, 97, 110))`. y estos códigos en la tabla ASCII se corresponden con la palabra **Plan**.

Desciframos código2. Por su terminación “parece” un base 64, pero realmente es un base32, obteniendo **Valencia**

Con estos código ya obtenemos el código para descomprimir el zip “**PR0F3S0R&R10**”

Descomprimos y obtenemos **episodio1.exe**, tras una primera ejecución obtenemos:



```
Wrong date R3m0!
-----HINT-----
'La persistencia de la memoria...'
-----
Presione una tecla para continuar . . .
```

Parece que tendremos que encontrar la fecha correcta. Vamos a darle un vistazo a los strings con rabin2. (rabin2 -z episodio1.exe)

```
000 0x00087288 0x00489088 4 5 (.rdata) ascii QZ^&
001 0x000872a5 0x004890a5 4 6 (.rdata) utf8 \a7|ç\r blocks=Basic Latin, Latin-1 Supplement
002 0x00087440 0x00489240 8 9 (.rdata) ascii 01/23/89
003 0x00087450 0x00489250 40 41 (.rdata) ascii \nCongratulation!!, Stealing Money $$$...
004 0x00087479 0x00489279 29 30 (.rdata) ascii \n-----
005 0x00087497 0x00489297 25 26 (.rdata) ascii \nStolen: 1.000.000.000 $
006 0x000874b1 0x004892b1 7 8 (.rdata) ascii \nFlag:
007 0x000874c0 0x004892c0 45 46 (.rdata) ascii \n.....\n
008 0x000874ee 0x004892ee 13 14 (.rdata) ascii System_Date:
009 0x000874fc 0x004892fc 17 18 (.rdata) ascii \nWrong date R3m0!
010 0x00087510 0x00489310 39 40 (.rdata) ascii \n-----HINT-----
011 0x00087538 0x00489338 35 36 (.rdata) ascii \n'La persistencia de la memoria...'
012 0x00087560 0x00489360 39 40 (.rdata) ascii \n-----
013 0x00087588 0x00489388 5 6 (.rdata) ascii pause
014 0x0008758e 0x0048938e 16 17 (.rdata) ascii 0123456789abcdef
```

Tenemos una cadena en formato de fecha, **01/23/89**. Cambiamos la fecha del sistema, probamos y BINGO!!!!

```
Congratulation!!, Stealing Money $$$...
-----
Stolen: 1.000.000.000 $
-----
Flag: e30f35ad8d9cb6efc0778539a669fa85
```

UAM{e30f35ad8d9cb6efc0778539a669fa85}

@bicacaro