MISCELLANEOUS
FWHIBBIT CTF TEAM

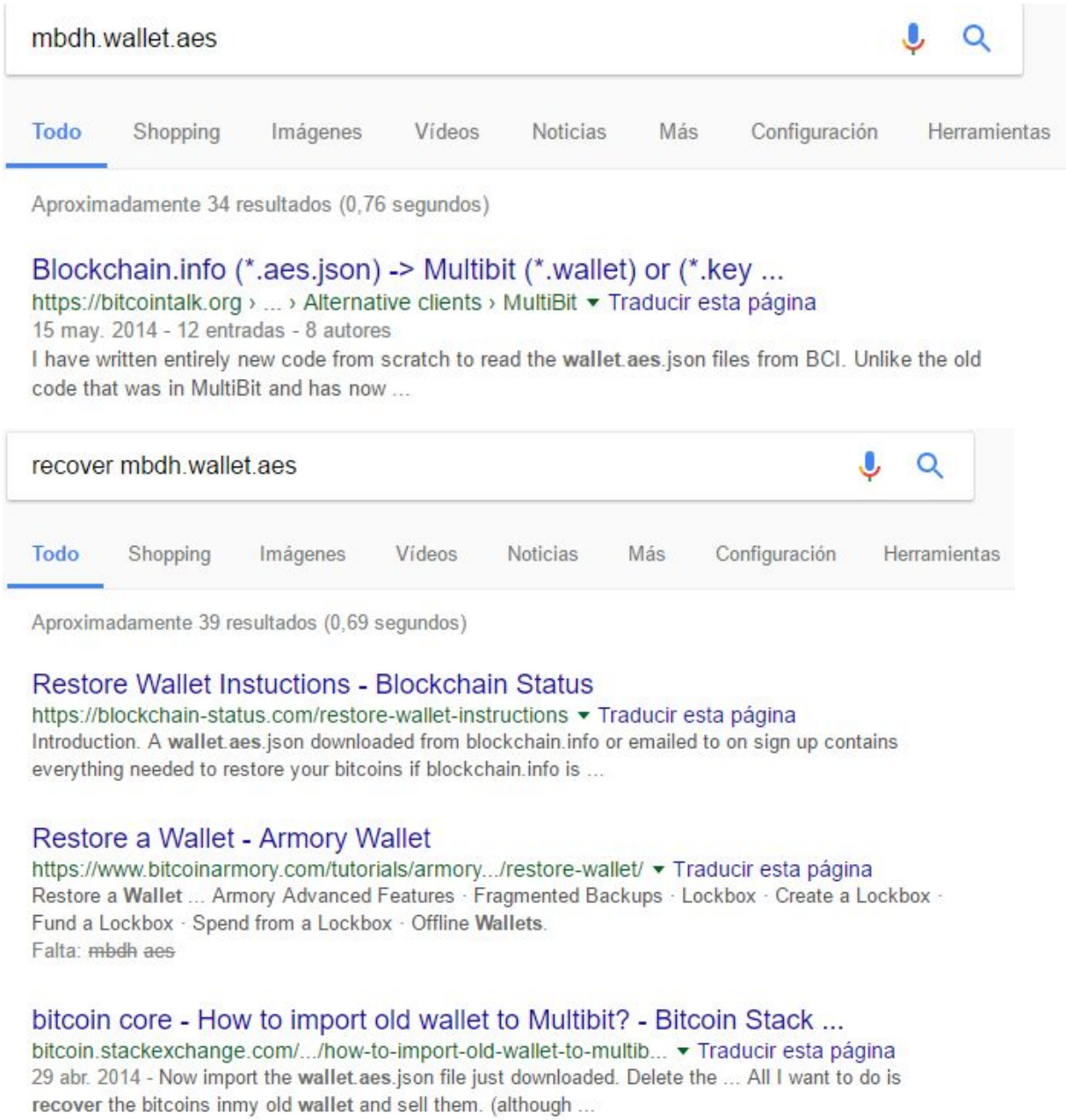
New Gold

Points: 350
Country: Brazil
Attatchment: https://mega.nz/#!0112RSqB!G4mDijp-vMQSaVcjr-ITP7KQgSqYEKcUqKcsCyZJ2Bk
Description: We lead a life of luxury and waste, but there are some rabbits trying to hide information in their brand new gold, and that's unnaceptable!

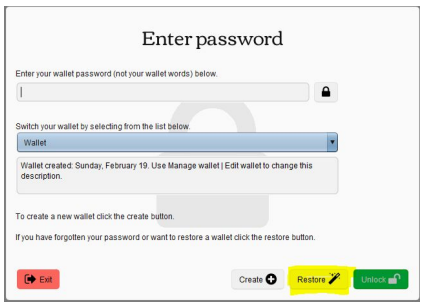
Find out what's going on.

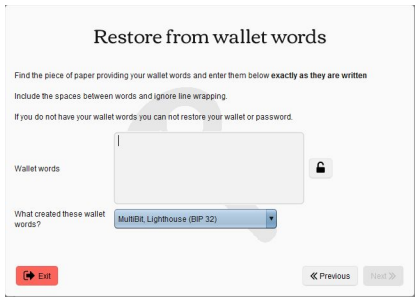
Link: https://mega.nz/#!0112RSqB!G4mDijp-vMQSaVcjr-ITP7KQgSqYEKcUqKcsCyZJ2Bk

Se nos proporciona un fichero llamado *mbhd.wallet.aes*, haciendo una búsqueda rápida en Google obtenemos lo siguiente:



Entonces sabemos que tenemos un wallet de MultiBit. Descargamos su cliente y lo instalamos. Lo abrimos e intentamos restaurar el wallet con *Restore -> I want to restore a wallet*





En la última captura se ve que se nos pide las wallet words, que suelen ser 12, 18 o 24 palabras.

Buscando (de nuevo) con nuestro amigo Google como se puede restaurar wallets y cómo obtener la "*master password"* y las otras 12 palabras (que es el número de palabras para las carteras de MultiBit) encontramos un github muy interesante:

- https://github.com/FuzzyHobbit/btc-recover

"*btcrecover is an open source Bitcoin wallet password and seed recovery tool. It is designed for the case where you already know most of your password or seed, but need assistance in trying different possible combinations*"

Entre todos los tipos de wallet vemos que da soporte a "*MultiBit Classic and MultiBit HD"* que es lo que estamos buscando. Recomiendo leer su guía de uso (pero no es necesario leerlo todo en nuestro caso, solo el apartado de "Running btcrecover"):

- https://github.com/FuzzyHobbit/btc-recover/blob/master/TUTORIAL.md#btcrecover-tutorial

Ejemplo de uso:

```
python btcrecover.py --wallet wallet.dat --tokenlist tokens.txt
--other-options...
```

En nuestro caso vamos a pasarle un diccionario. ¿Cuál es el diccionario más famoso? El *rockyou.txt*. Lo podemos descargar de aqui:

- https://wiki.skullsecurity.org/index.php?title=Passwords

Ejecutamos lo siguiente:

```
python btcrecover.py --wallet mbhd.wallet.aes --passwordlist rockyou.txt
```

Al ser muchas passwords que probar y dependiendo de la máquina donde lo corremos puede que ponga como *estimated time* **DÍAS**, tranquilos, dejarlo corriendo y en cuestión de minutos saca la password. En mi caso obtuve lo siguiente como salida (nos os preocupeis de los warnings que vayan apareciendo):

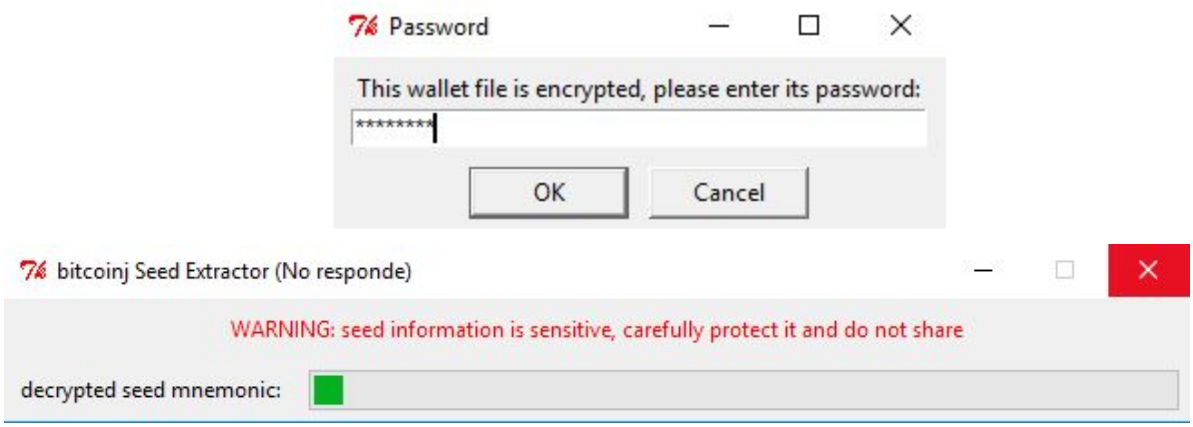
16662 of 14329857 [----------] 0:15:10, ETA: 9 days, 1:22:07
Password found: 'monkey08'

De acuerdo, ¿pero no se necesitaban 12 palabras para restaurar una wallet? Porque en el cliente de MultiBit escribiendo la password obtenida no deja darle a Next. Haciendo otra búsqueda en Google (cómo no) y leyendo algunos foros llegamos al siguiente Github:

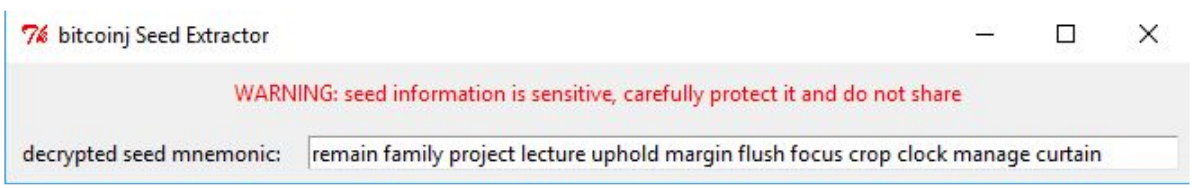
- https://github.com/gurnec/decrypt_bitcoinj_seed

"*a simple Python script which decrypts and displays the seed mnemonic from from a bitcoinj-based HD wallet file*"

Lo ejecutamos (es un .pyw) e introducimos la password que hemos obtenido previamente (*monkey08*):

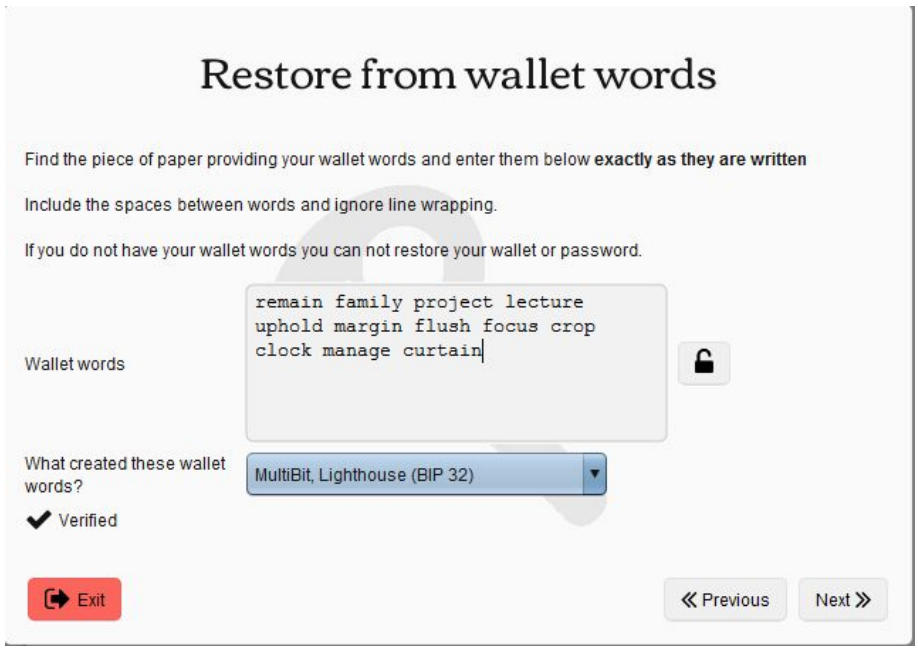


Resultado:



Palabras obtenidas (12): remain family project lecture uphold margin flush focus crop clock manage curtain

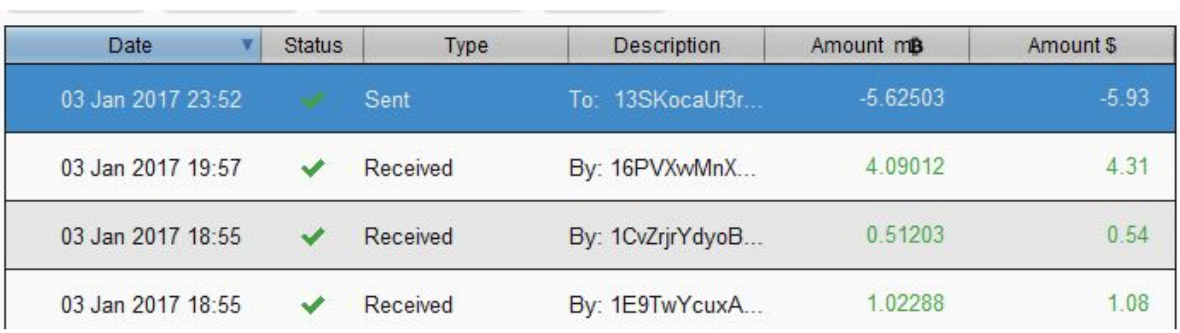
Ahora con esas palabras podemos restaurar el wallet:



(No tengo pantallazos de las siguientes pantallas porque tenía un backup guardado, pero te pide un Datestamp que se supone que estaba apuntado junto con las 12 palabras pero ignorad ese campo, introducir una contraseña nueva y listo)

Dejar unos minutos para que el cliente se sincronice:

Una vez sincronizado, vamos a la pestaña de “Payments” a la izquierda y observamos lo siguiente:



Date	Status	Type	Description	Amount mB	Amount $
03 Jan 2017 23:52	✔	Sent	To: 13SKocaUf3r...	-5.62503	-5.93
03 Jan 2017 19:57	✔	Received	By: 16PVXwMnX...	4.09012	4.31
03 Jan 2017 18:55	✔	Received	By: 1CvZrjrYdyoB...	0.51203	0.54
03 Jan 2017 18:55	✔	Received	By: 1E9TwYcuxA...	1.02288	1.08

(Se puede exportar la tabla de transacciones a un .csv con el botón de Export para luego poder copiar y pegar las direcciones de wallets)

Tenemos 4 transacciones en total, 1 envío y 3 recibos. Os dejo una tabla formateada con la información relevante:

Fecha	Tipo	Dirección	Cantidad
03-01-2017 23:52	Envío	13SKocaUf3rCgiFissrWcwd4E97SXVWooY	-5.62
03-01-2017 19:57	Recibido	16PVXvvMnXKSiKXccdYikHZJMyNNBMQHDg	+4.09
03-01-2017 18:55	Recibido	1CvZrjrYdyoBYbBksPh8qEg8WJx5aeZ6MF	+0.51
03-01-2017 18:55	Recibido	1E9TwYcuxAVPYYgkqTbbPRG4W7RXJzLk3j	+1.02

Es importante saber que cuando se crea un wallet de bitcoin normalmente se crean varias direcciones de bitcoin (información no relevante para nuestro caso, pero es bueno saberlo).

En mi caso tenía la hint, que era mirar el blockchain, y se puede utilizar cualquier página para ello. Yo he utilizado las siguientes dos:

- https://blockchain.info/es/home
- https://bitcoinchain.com/

¿Y ahora qué se hace? Intenté mirar tanto para adelante como para atrás por si veía algo sospechoso. Demasiada información, los pagos se partían en muchas carteras, luego se juntaba, muchas transacciones, algunas direcciones *Script Hash,* etc. Muchas cosas y no sabes qué tienes que mirar.

OFFTOPIC: muy interesante lo de *Script Hash* (no lo conocía) que tienen una pinta así los scripts (TLDR: direcciones que empiezan por 3 en vez de 1 que son las normales y que para gastar necesitan un script):

- https://en.bitcoin.it/wiki/Pay_to_script_hash

Después de pegarme horas viendo carteras y transacciones (sin ver nada obviamente y con historias muy curiosas como ver IPs de por allí que retransmiten las transacciones, llegando a entrar en sus dominios, viendo el whois e incluso llegando a

descargar algún que otro .vbs con sentencias SQL, en fin, totally OFFTOPIC y out of scope), “alguien” me preguntó: “se puede esconder informacion en el blockchain?”, mi respuesta fue: “O_O”.

Investigando cómo ocultar información en el blockchain se llega a un artículo y un post de reddit muy interesantes:

- http://www.righto.com/2014/02/ascii-bernanke-wikileaks-photographs.html
- https://www.reddit.com/r/WhereIsAssange/comments/5e4o70/tutorial_how_to_decode_btc_blockchain_hidden/

Probé muchos de los ejemplos mencionados para entender cómo funcionaba y me quedé impresionado, jamás había oído de ello. Para demostrarlo voy a explicar el ejemplo del primer artículo.

Tenemos la transacción 8881a937a437ff6ce83be3a89d77ea88ee12315f37f7ef0dd3742c30eef92dba, y vemos la información de la transacción en hexadecimal en la siguiente página (en la url lleva la transacción):

- https://blockexplorer.com/api/rawtx/8881a937a437ff6ce83be3a89d77ea88ee12315f37f7ef0dd3742c30eef92dba

Obviando la primera parte *{"rawtx":"* y *"}* del final, lo convertimos a ASCII utilizando por ejemplo http://www.rapidtables.com/convert/number/hex-to-ascii.htm y obtenemos “algo” que es “legible” (o al menos algunas partes):

> There is not¬|v©hing like returning ¬|v©to a place
> that re¬|v©mains unchanged to f¬|v©ind the ways in
> w¬|v©hich you yourself ha¬|v©ve altered.â
> -Ne¬|v©lson Mandela

Esto se puede hacer con un bloque entero del blockchain. Siguiendo el ejemplo anterior, volvemos a https://blockchain.info/tx/8881a937a437ff6ce83be3a89d77ea88ee12315f37f7ef0dd3742c30eef92dba , vamos al “Resumen”, “Incluídas en el bloque” y le damos. A la derecha donde pone “hash”, copiamos la hash “000000000000000048832c5b83c4fa63f0d40f973f71f43d1334673ca7d450f19” y nos vamos a https://blockexplorer.com/api/rawblock/000000000000000048832c5b83c4fa63f0d40f973f71f43d1334673ca7d450f19 (fijarse en le URL, en este caso /rawblock) y obtenemos “mucha” (o demasiada) información. Ahora al igual que antes, lo convertimos a ASCII y fin, porque leerse todo eso es… :pirate:

Sabiendo esto, empezamos abriendo las 4 direcciones que tenemos:

- https://blockchain.info/address/13SKocaUf3rCgiFissrWcwd4E97SXVWooY
- https://blockchain.info/address/16PVXvvMnXKSiKXccdYikHZJMyNNBMQHDg
- https://blockchain.info/address/1CvZrjrYdyoBYbBksPh8qEg8WJx5aeZ6MF
- https://blockchain.info/address/1E9TwYcuxAVPYYgkqTbbPRG4W7RXJzLk3j

En cada link podemos ver que hay 2 transacciones, un envío y un recibo. Ahora con lo aprendido de los artículos anteriores, y utilizando el script que se menciona al final del post de reddit, empezamos a mirar si hay algo interesante (o simplemente con http://www.rapidtables.com/convert/number/hex-to-ascii.htm, yo me decanté por el script porque estaba un poco cansado).

- http://gateway.glop.me/ipfs/QmSU67Ei3TerNe32CcZTgd48jKqsVvBTgera1qBWFjKK9V/jean.py

El script se utiliza de la siguiente forma:

```
python [nombredelscript].py [transaccion]
```

Con las transacciones de las direcciones 13SKocaUf3rCgiFissrWcwd4E97SXVWooY, 1E9TwYcuxAVPYYgkqTbbPRG4W7RXJzLk3j, 1CvZrjrYdyoBYbBksPh8qEg8WJx5aeZ6MF no se obtiene nada interesante. En cambio con la dirección 16PVXvvMnXKSiKXccdYikHZJMyNNBMQHDg, y en cocnreto con la transacción de recibo 809d96b01e04916a4a9e2ef836682f27ec36829eaef8e65df0db73fe60a307b4 obtenemos algo interesante (salida del script):

```
aXR7Q3J5cHQwY1VycjNuYzEzc180cjNfdGgzX0Z1dHVyM30=
  Óı╗D^▓EA¿®!¹]▒MÕGøwôÉÔ½YÜ/kóÝN°;¢M9ùù;Ö4.ó/ÂO│p&y÷▒ƒ┐Ê«
```

La primera línea es un base64, lo convertimos y obtenemos: **it{Crypt0cUrr3nc13s_4r3_th3_Futur3}**

Investigando por curiosidad las otras transacciones para obtener el flag completo (aún sabiendo el formato) no obtuve gran cosa. Al final haciéndolo a mano (con la misma transacción) y reconstruyendo base64 que estaba partido al decodificar el hexadecimal tenemos ZndoaWJiaXR7Q3J5cHQwY1VycjNuYzEzc180cjNfdGgzX0Z1dHVyM30=, que al convertirlo obtenemos **fwhibbit{Crypt0cUrr3nc13s_4r3_th3_Futur3}**.

FLAG: fwhibbit{Crypt0cUrr3nc13s_4r3_th3_Futur3}

DISCLAIMER: no soy un experto en el mundo del bitcoin, tengo conocimientos básicos, por lo tanto, hay muchas cosas que no las llamo con los términos apropiados. Durante la resolución de este reto he ido cambiando entre una máquina Windows y Linux por el tema de librerías de python y otros tipos de software (que me daban problemas).

Para acabar me gustaría dar las gracias a la organización de fwhibbit por brindar la oportunidad de participar en este CTF en el cual he aprendido mucho (incluso demasiado, aunque eso nunca puede pasar ^_^). Gracias a todos los admin por aguantarme y apoyarme para que siguiese con los retos. Este reto era de la categoría de Misc, y creo que es el reto con el cual más he aprendido (junto con los 3 retos de Dev).

-SoA aka ruso