Pues aquí estamos nuevamente frente a un ctf muy muy divertido, por ello dar las gracias antes de nada.

Bien, en primer lugar arrancamos descargando un archivo png, ya arrancamos con una pista en la presentación del reto, algo de emisión de radio hay de por medio.

En princio no parace nada mas que una imagen pero toda vez que la abrimos en bruto "cat Reto_14_H4F_NLP_Augiie.png > bruto.txt" vemos que al final nos encontramos con una línea que nos dice algo

TAGwd3ckiwisdr.ddns.net_2018-08-3

Tras dar muchas vueltas investigando sobre el tema de las emisiones SDR via internet, interesante mundo ese, decido buscar alguna herramienta para determinar si el png contiene algo mas, tiramos de zsteg (https://github.com/zed-0xff/zsteg) interesante herramienta que tras lanzarla "zsteg Reto_14_H4F_NLP_Augiie.png" nos arroja lo siguiente

```
[?] 2075565 bytes of extra data after image end (IEND), offset = 0x2589a
extradata:0         .. file: Audio file with ID3 version 2.4.0, contains:MPEG ADTS, layer III, v1,  64 kbps, 44.1 kHz, Monaural
   00000000: 49 44 33 04 00 00 00 00  09 1b 54 49 54 32 00 00  |ID3.......TIT2..|
   00000010: 00 5a 00 00 00 77 64 33  63 6b 69 77 69 73 64 72  |.Z...wd3ckiwisdr|
   00000020: 2e 64 64 6e 73 2e 6e 65  74 5f 32 30 31 38 2d 30  |.ddns.net_2018-0|
   00000030: 38 2d 33 31 54 32 31 5f  34 38 5f 34 36 5a 5f 31  |8-31T21_48_46Z_1|
   00000040: 37 38 35 35 2e 30 30 5f  61 6d 5f 30 30 20 28 6f  |7855.00_am_00 (o|
   00000050: 6e 6c 69 6e 65 2d 61 75  64 69 6f 2d 63 6f 6e 76  |nline-audio-conv|
   00000060: 65 72 74 65 72 2e 63 6f  6d 29 2e 6d 70 33 54 58  |erter.com).mp3TX|
   00000070: 58 58 00 00 00 0c 00 00  00 54 52 41 43 4b 54 4f  |XX.......TRACKTO|
   00000080: 54 41 4c 00 54 43 4f 50  00 00 00 01 00 00 03 54  |TAL.TCOP.......T|
   00000090: 4c 41 4e 00 00 00 01 00  00 03 54 50 55 42 00 00  |LAN.......TPUB..|
   000000a0: 00 01 00 00 03 00 00 00  00 00 00 00 00 00 00 00  |................|
   000000b0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
   *
   00000100:
imagedata           .. file: MIPSEB MIPS-III ECOFF executable not stripped - version 1.0
b2,g,msb,xy         .. text: "j}h__?d3"
b2,bgr,lsb,xy       .. text: "E8t>n6co"
b2,bgr,msb,xy       .. text: "/l&Z8#ziG"
b3,g,lsb,xy         .. file: PGP symmetric key encrypted data -
b3,bgr,msb,xy       .. text: "Z9jq91u/"
b3,abgr,msb,xy      .. text: "wpy?{xgt"
b4,b,lsb,xy         .. text: "ECDEffUd\n"
```

b4,bgr,msb,xy .. text: ";xKpDl,."
b4,rgba,lsb,xy .. text: "@/Q/A/1/A"

oh! Un mp3! abrimos nuevamente el archivo en bruto y nos vamos a buscar las cabeceras de un mp3 "ID3" y nos encontramos esto "IEND®B`,ID3 " donde acaba el png empieza un mp3!

Seleccionamos datos en bruto, copiamos y pegamos y tenemos un rico mp3. Lo escuchamos y tiene partes que suenan "raro". Siguiente paso, abrirlo en audacity e intentar encontrar algo. Analizamos el espectro y nos encontramos a mitad del audio una porción a caso hecho en la que se puede leer "NLP" y poco después se escucha una mensaje en morse, la subimos a https://morsecode.scphillips.com/labs/audio-decoder-adaptive/ pero el no consigue extraerlo, así que trabajamos un poco el spectro, unos filtros bajos y altos, cambiamos a blanco y negro, regulamos los decibelios para dejar el morse lo mas limpio posible y a mano obtenemos:

...-- ....- ....- -.-. ----. ----- ----- -... .- . --... ..... ----. ...-- ..--- . .- ..--- ----. ----. .---- --... -... ..--- -.. .---- ..... .---- ....- ..-. .---- .-

que una vez decodificado nos da:

344C900BAE75932EA29917B2D1514F1A

Tiramos de hash-identifier y nos dice que es podría ser un MD5, aunque a simple vista algo se intuía. Buscamos por la web y en https://md5online.es//descifrar-md5.html
nos arroja como resultado:

0P3n_Th1S_GOO.GL/W3F1hS

Que abra goo.gl/W3F1hS jejej si fuera de otro pasaría el enlace por mil filtros antes de abrir pero viene de alguien de confianza con quien algún día puede que comparta birra xD

Nos descargamos un fichero zip "NLP_H4F_Reto14_R0cKy0U.zip" que ya en su nombre nos indica que tenemos que tirar de la wordlist para abrirlo, ergo

fcrackzip -u -D -p /usr/share/wordlists/rockyou.txt NLP_H4F_Reto14_R0cKy0U.zip y nos da como resultado

PASSWORD FOUND!!!!: pw == natasharock

Descomprimimos y tenemos un fichero mas! Bien! esto no acaba aquí!

Un archivo Try_H4rDer.enc, file Try_H4rDer.enc nos dice que

Try_H4rDer.enc: openssl enc'd data with salted password, base64 encoded

Con lo cual, abrimos en bruto el archivo y al final nos encontramos esto,

KRUDGICLGN4SAMLTHIQG4YLWMFZHEYLMMFXHAYLSOR4Q==== que decodificamos en base64 y nos da error, pensemos un poco... ==== será base32? si!

Th3 K3y 1s: navarralanparty

Tenemos un archivo cifrado con openssl/base64... señor google (mi cerebro carece de datos) buscamos y nos revela que tenemos que usar:

openssl enc -base64 -d -in Try_H4rDer.enc -out secret.txt -k navarralanparty

Y el resultado es una locura que me sonaba de algún reto pero no conseguía recordarlo y tras dar vueltas me echan un cable y me dicen que esto

![]+[])[+!+[]]+(![]+[])[!+[]+!+[]]+(!![]+[])[!+[]+! (parte del archivo de 70.6 kb solo de signos) es un código codificado con JsFuck, Fuck! digo yo! XDD

Tras buscar encuentro

http://codertab.com/JsUnFuck y obtenemos flag{NLP_H4F_Y34h_M1ngU5}

ouuuu yea the flag in the house!

Muy muy divertida e instructiva! Gracias a Hackers4Fun CTF Team por el trabajazo!!!!

Nos vemos en las siguientes!

eternaln00b