***Descripción***

***Nombre:*** 4ugg13 (Related https://en.wikipedia.org/wiki/Covert_Affairs )
***Fecha de liberación:*** xx de septiembre de 2018
***Autor:*** 1v4n (https://twitter.com/1r0Dm48O // https://twitter.com/hackers4f )
***Categoría:*** Misc
***Puntuación:*** 200
***Dificultad:*** Medio-Bajo

Our Head of Technical Operations for DPD in the "agency" has received a file of an intercepted radio transmission on the 17855 kHz frequency, 16 meter band, from Arlington (Virginia). It is suspected that the well-known hacker "Tash" took advantage of the country of emission to transmit an encrypted message.

**Objetivo**

Formato de la flag: flag{texto}

**Herramientas utilizadas**

Versión 68.0.3440.106 (Build oficial) (64 bits) https://www.google.com/chrome/
file-5.34
UnZip 6.00 ftp://ftp.info-zip.org/pub/infozip/
fcrackzip v 1.0 http://www.goof.com/pcg/marc/
Stego-toolkit https://github.com/DominicBreuker/stego-toolkit
OpenSSL 1.1.0h 27 Mar 2018
https://md5online.org/md5-decrypt.html
https://gchq.github.io/CyberChef
https://github.com/aemkei/jsfuck
http://www.jsfuck.com/

**Resumen:**

Comenzamos por visitar el reto y descargamos el archivo adjunto *Reto_14_H4F_NLP_Augiie* sin extensión.

Pasamos a identificar el archivo con *$file* y nos arroja que estamos ante imagen PNG.

```
root@kali:~/Desktop/Hackers4F/Reto 14 NLP# file Reto_14_H4F_NLP_Augiie
Reto_14_H4F_NLP_Augiie: PNG image data, 425 x 252, 8-bit/color RGBA,
non-interlaced
root@kali:~/Desktop/Hackers4F/Reto 14 NLP#
```

Cambiamos la extensión a PNG con un tamaño de 2,2 MB (2.229.319 bytes) y con una dimensión de 425 x 252. Nos indica que puede la imagen pueden esconder algún secreto. Pasamos clonar y utilizar el script *check_png.sh* del Stego-toolkit https://github.com/DominicBreuker/stego-toolkit y nos arroja:

```
root@kali:~/Desktop/stego-toolkit/scripts# ./check_png.sh
Reto_14_H4F_NLP_Augiie.png

#################################
########## PNG CHECKER ##########
```

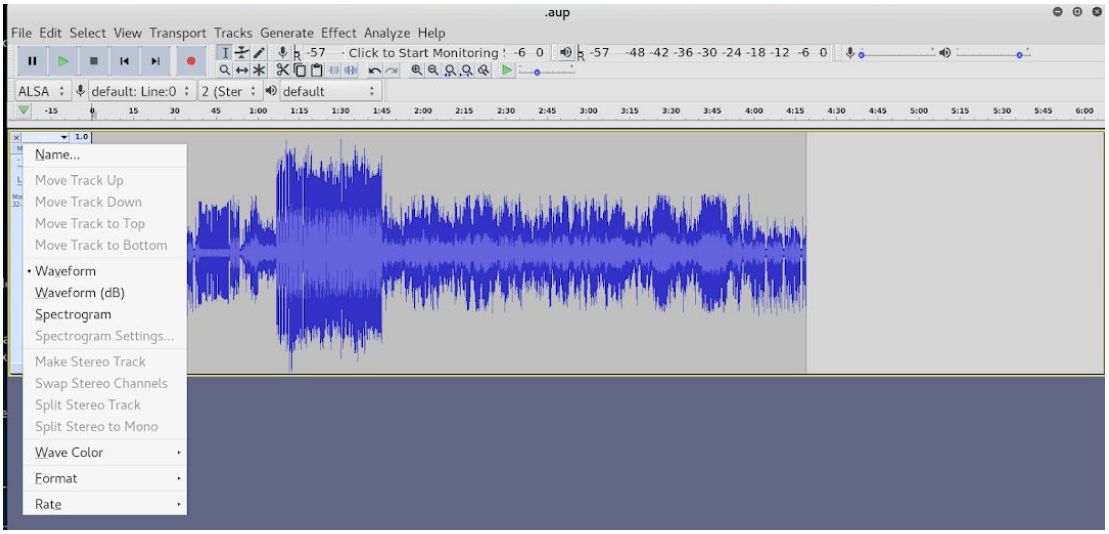
```
################################
Checking file Reto_14_H4F_NLP_Augiie.png

Reto_14_H4F_NLP_Augiie.png: PNG image data, 425 x 252, 8-bit/color RGBA,
non-interlaced
…

#########################
########## zsteg ##########
#########################

Watch out for red output. This tool shows lots of false positives...
[?] 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
...
                    ..
###############################
########## openstego ##########
###############################

Nothing found...

###############################
########## stegano-lsb ##########
###############################

Nothing found...

###################################
########## stegano-lsb-set ##########
###################################

Nothing found...

###############################
########## stegano-red ##########
###############################
./check_png.sh: línea 123: stegano-red: no se encontró la orden

#############################
########## LSBSteg  ##########
#############################
Nothing found...

################################
########## stegoVeritas ##########
################################
./check_png.sh: línea 143: stegoveritas.py: no se encontró la orden
```

La tool zsteg detecta que existe un archivo de audio en MP3. Pasamos a extraerlo con $dd en el offset offset = 0x2589a que debemos pasarlo a decimal e indicarlo en el parámetro *skip=153754*, arrojándonos:

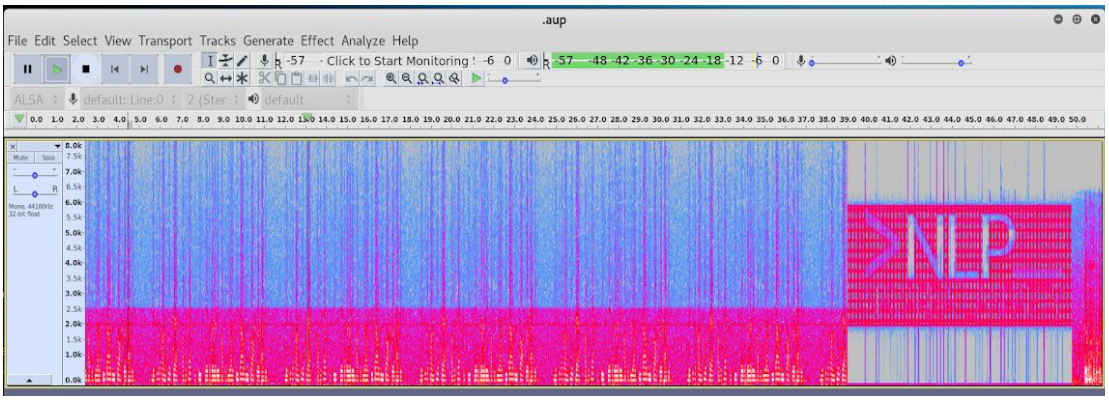
```
root@kali:~/Desktop/Hackers4F/Reto 14 NLP# dd if=Reto_14_H4F_NLP_Augiie
of=output bs=1 skip=153754
2075565+0 registros leídos
2075565+0 registros escritos
2075565 bytes (2,1 MB, 2,0 MiB) copied, 4,99504 s, 416 kB/s
root@kali:~/Desktop/Hackers4F/Reto 14 NLP#
```

```
root@kali:~/Desktop/Hackers4F/Reto 14 NLP# file output
output: Audio file with ID3 version 2.4.0, contains:MPEG ADTS, layer III, v1,
64 kbps, 44.1 kHz, Monaural
root@kali:~/Desktop/Hackers4F/Reto 14 NLP#
```

A partir de aquí vamos analizar el archivo con Audacity y el Espectrograma que genera:

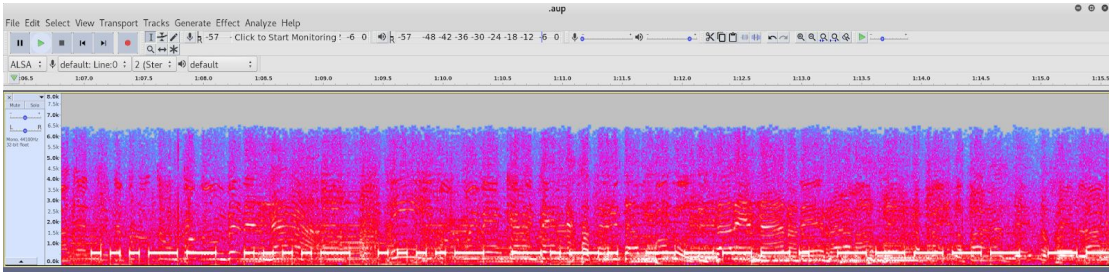



Observamos que la voz de una operadora se repite aprox. 7 veces y que posteriormente nos desvela el espectrograma del logo de la NavarraLanParty:



Pero hasta el segundo 00:50 de la reproducción empieza a escucharse la emisión de una radio que emite en castellano (y que suponemos que es la señal de RNE exterior).

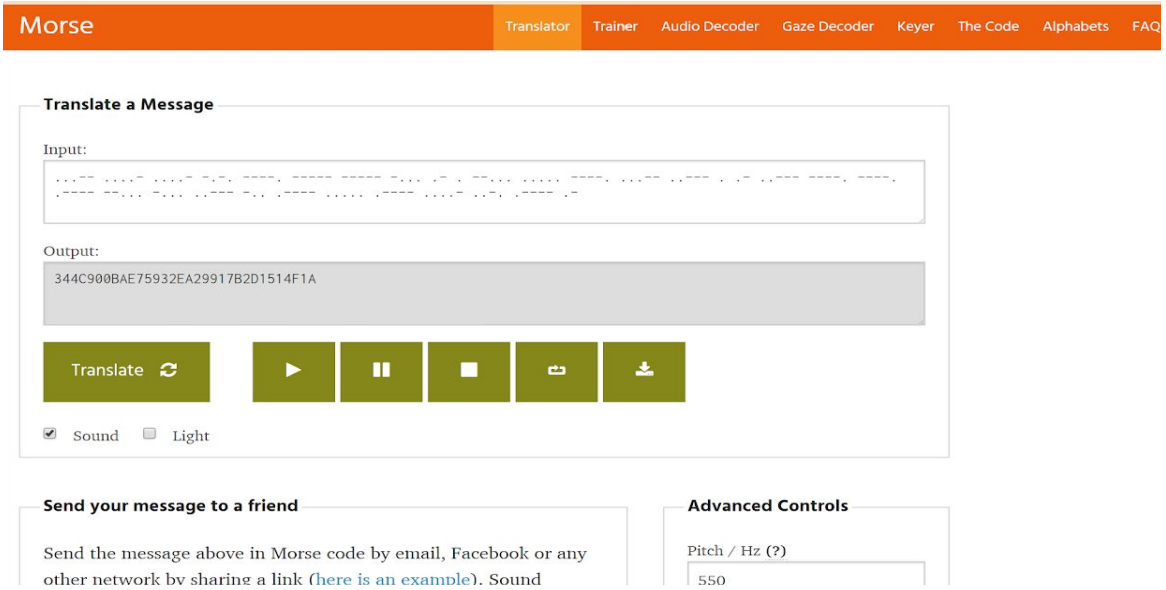
En el minuto 01:07 se escucha un claro código Morse que se plasma de la siguiente forma:



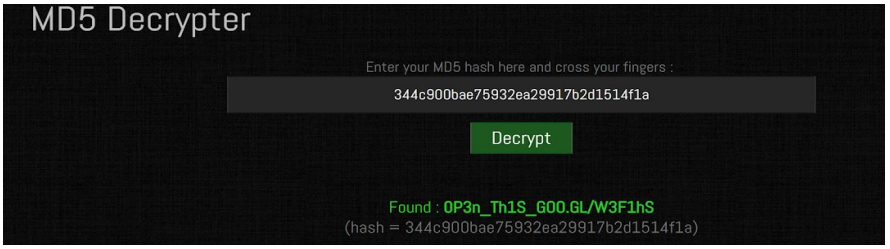

Se identifica el código Morse siguiente:

```
...-- ....- ....- -.-. ----. ----- ----- -... .- . --... ..... ----. ...-- ..---
. .- ..--- ----. ----. .---- --... -... ..--- -.. .---- ..... .---- ....- ..-.
.---- .-
```

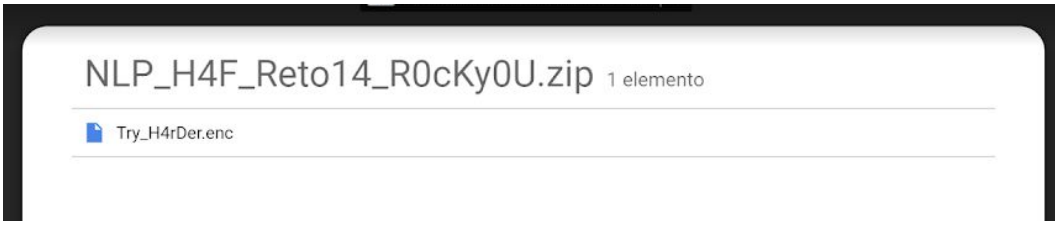
Utilizamos la herramienta online de https://morsecode.scphillips.com/translator.html para decodificar el Morse:




Nos encontramos un hash MD5 que pasamos a crackear a través de la herramienta online que nos arroja la siguiente **0P3n_Th1S_GOO.GL/W3F1hS**



Pasamos a visitar la dirección acortada de GOO.GL/W3F1hS y obtenemos un archivo comprimido *NLP_H4F_Reto14_R0cKy0U.zip*, que nos anticipa por su nombre que tengamos que hacer un ataque de fuerza bruta con rockyou.txt al archivo .zip comprimido.




Confirmamos que posee contraseña:

```
root@kali:~/Desktop/Hackers4F/Reto 14 NLP# unzip NLP_H4F_Reto14_R0cKyU0.zip
Archive:  NLP_H4F_Reto14_R0cKyU0.zip
[NLP_H4F_Reto14_R0cKyU0.zip] Try_H4rDer.enc password:
   skipping: Try_H4rDer.enc          incorrect password
root@kali:~/Desktop/Hackers4F/Reto 14 NLP#
```

Y ejecutamos el ataque de fuerza bruta al archivo .zip con fcrackzip arrojando la password `natasharock` en pocos segundos:

```
root@kali:~/Desktop/Hackers4F/Reto 14 NLP# fcrackzip -b -D -p rockyou.txt -u
NLP_H4F_Reto14_R0cKyU0.zip


PASSWORD FOUND!!!!: pw == natasharock
root@kali:~/Desktop/Hackers4F/Reto 14 NLP#
```

Por cierto nos encontramos con el nombre real de “Tash” que es Natasha como parte de la password. Pasamos a descomprimirlo y obtenemos un archivo codificado `Try_H4rDer.enc`:

```
root@kali:~/Desktop/Hackers4F/Reto 14 NLP# unzip NLP_H4F_Reto14_R0cKyU0.zip
Archive:  NLP_H4F_Reto14_R0cKyU0.zip
[NLP_H4F_Reto14_R0cKyU0.zip] Try_H4rDer.enc password:
  inflating: Try_H4rDer.enc
root@kali:~/Desktop/Hackers4F/Reto 14 NLP# file Try_H4rDer.enc
Try_H4rDer.enc: openssl enc'd data with salted password, base64 encoded
root@kali:~/Desktop/Hackers4F/Reto 14 NLP#
```

Observamos que con openssl podríamos decodificarlo, pero necesitamos una password. Vamos a analizar Try_H4rDer.enc y contiene una última línea de código añadida con la siguiente cadena:

```
KRUDGICLGN4SAMLTHIQG4YLWMFZHEYLMMFXHAYLSOR4Q====
```

Probamos a decodificar en base32 arrojando la password `navarralanparty`:

```
printf "KRUDGICLGN4SAMLTHIQG4YLWMFZHEYLMMFXHAYLSOR4Q====" | base32 -d
Th3 K3y 1s: navarralanparty
```

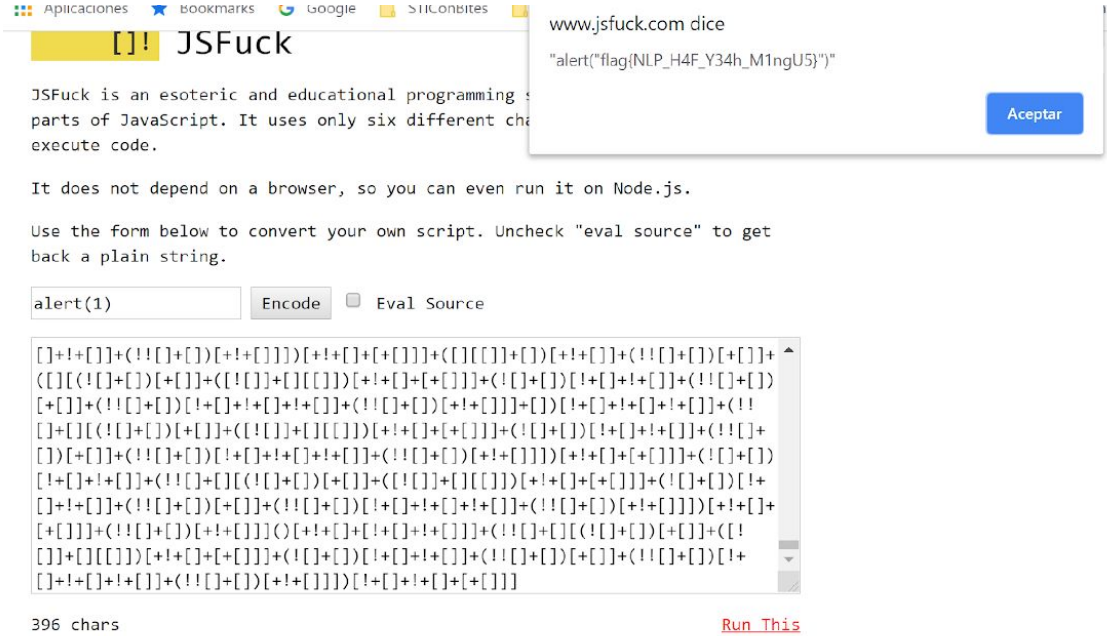
Ahora ya podemos pasar a decodificar el mensaje con openssl y vemos que hay un error:

```
root@kali:~/Desktop/Hackers4F/Reto 14 NLP# openssl enc -aes-256-cbc -d -a -in
Try_H4rDer.enc -out output
enter aes-256-cbc decryption password:
bad decrypt
139736640700608:error:0606506D:digital envelope
routines:EVP_DecryptFinal_ex:wrong final block
length:../crypto/evp/evp_enc.c:525:
root@kali:~/Desktop/Hackers4F/Reto 14 NLP#
```

Debemos eliminar la última línea de condigo adicional que nos aportó la password `navarralanparty` y volvemos a repetir el proceso:

```
root@kali:~/Desktop/Hackers4F/Reto 14 NLP# openssl enc -aes-256-cbc -d -a -in
Try_H4rDer.enc -out output
enter aes-256-cbc decryption password:
root@kali:~/Desktop/Hackers4F/Reto 14 NLP# cat output
(![]+[])[+!+[]]+(![]+[])[!+[]+!+[]]+(!![]+[]+!+[]+!+[]]+(!![]+[])[+!+[]]]+[])[!+[]
...
!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]])[!+[]+!+[]+[
+[]]]
```

Nos encontramos con una cadena que pertenece a un código JavaScript ofuscado en JSFuck (http://www.jsfuck.com/ ) encontramos una herramienta online que desofusca.




Y la solución es ***flag{NLP_H4F_Y34h_M1ngU5}***

Autor: 1v4n a.k.a. @1r0Dm448O
Twitter: https://twitter.com/Hackers4f // https://twitter.com/1r0Dm448O