WRITEUP - HALL OF FAME - RETO 20 H4F

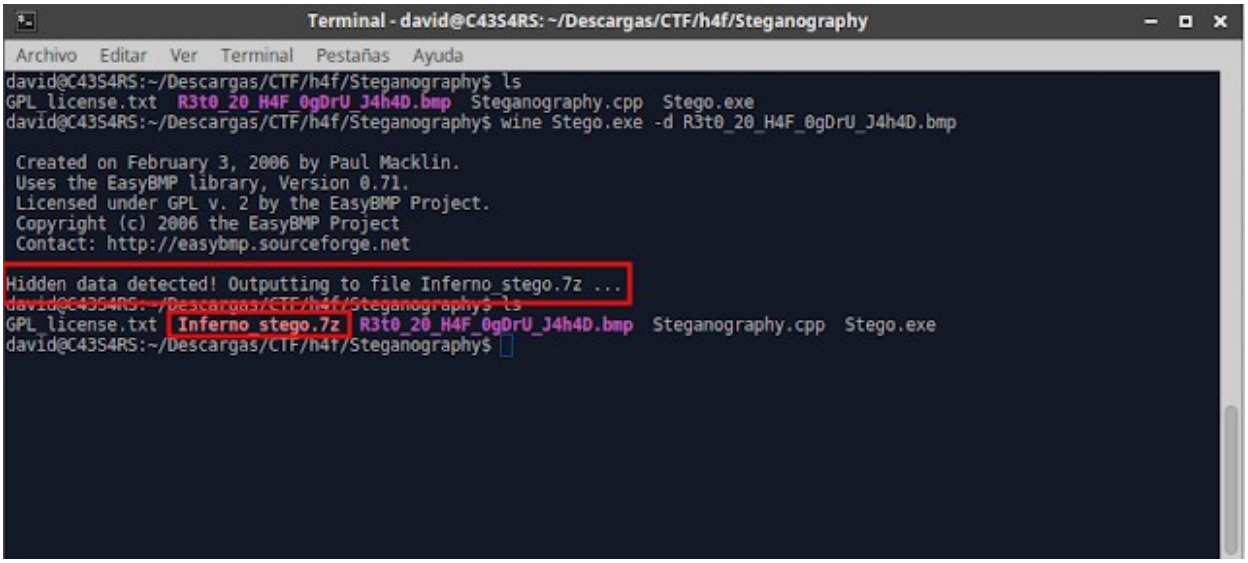
Los chic@s de Hackers4Fun nos traen el reto nº 20. En él, nos enfrentamos a una amenaza "infernal".

Manos a la obra

Descargamos el archivo .7z del enlace que nos proporcionan desde Twitter. Una vez descomprimida tendremos la siguiente imagen:

La analizamos con Exiftool pero nada raro, tras probar con varias herramientas de Stego, vemos

que una de ellas nos saca una nueva imagen.



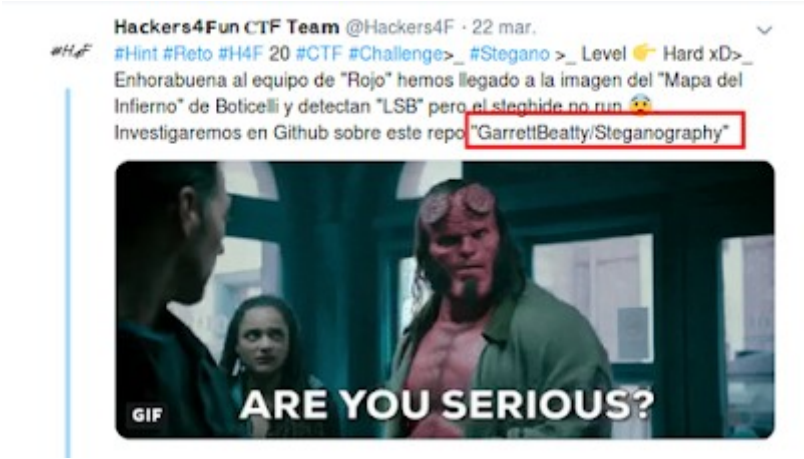
Obtenemos un archivo llamado "Inferno_stego.7z ", en el título tenemos una pista y es que volvemos a tener "stego" para rato...

No podemos ver la imagen, por lo que seguramente no se trate de un .jpg....

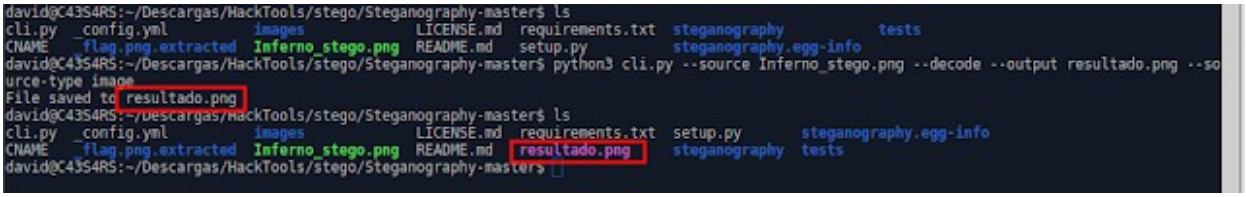
Efectivamente, el archivo es un PNG, renombramos el archivo y lo abrimos.

Perfecto! Otra foto! Pequeña...Pero foto! Si utilizamos Google Images, nos indica que se trata del Infierno de Dante con sus nueve círculos.

Tras probar con varias tools LSB, no se obtiene nada, tampoco funciona Steghide...Nos llega una pista desde Twitter:



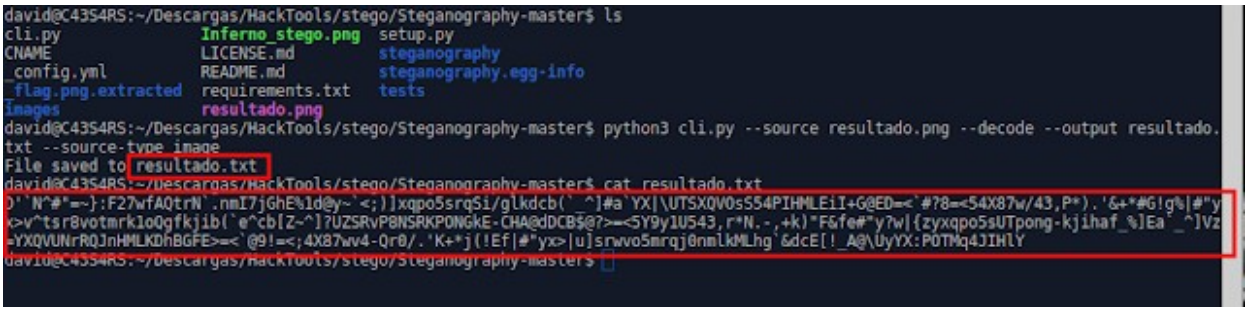
Probamos con esta herramienta:



Y nos desvela la nueva imagen:

Aunque no es muy legible, vemos que en la columna de la derecha pone "MALBOLGE", es el 8° circulo del Infierno de Dante.

Utilizaremos la misma técnica, pero esta vez llamaremos el archivo "resultado.txt " en vez de "resultado.png ".



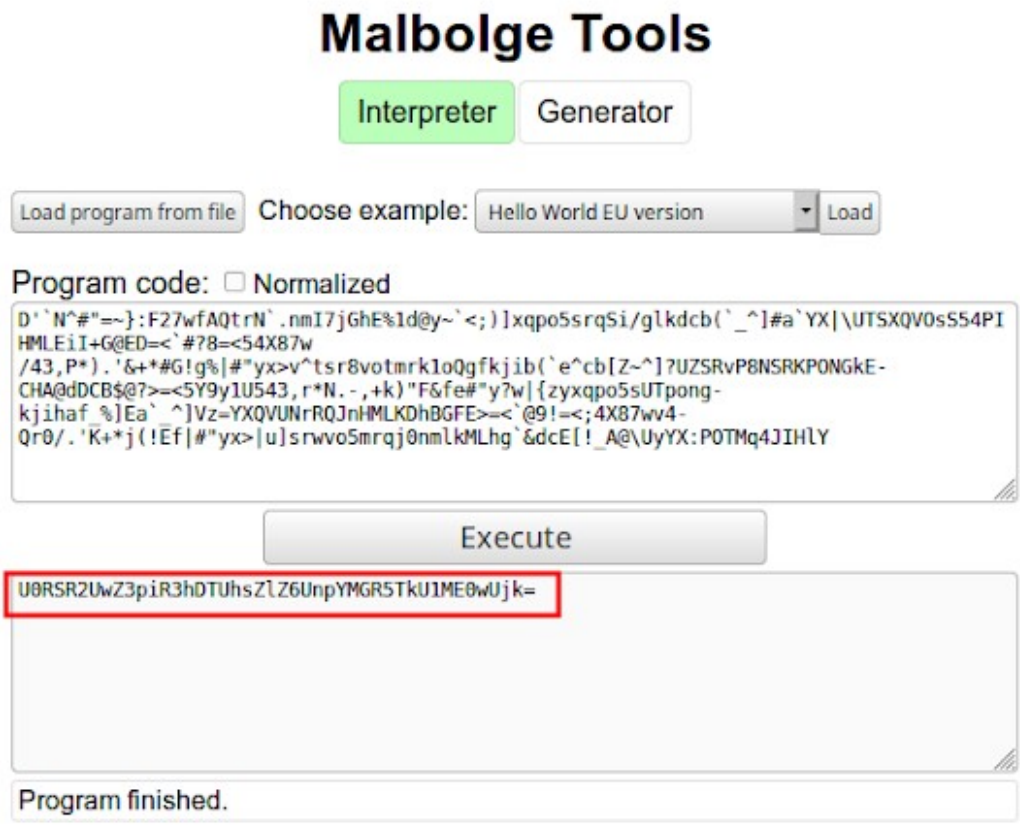
También escondía un .txt, si hacemos un "cat".... Nos suelta el siguiente string:

D`N^#"=~}:F27wfAQtrN`.nmI7jGhE%1d@y~`<;)]xqpo5srqSi/glkdcb(`_^]#a`YX|\UTSXQV
OsS54PIHMLEiI+G@ED=<`#?8=
<54X87w/43,P*).'&+*#G!g%|#"yx>v^tsr8votmrk1oQgfkjib(`e^cb[Z~^]?
UZSRvP8NSRKPONGkE-CHA@dDCB$@?>=<5Y9y1U543,r*N.-,+k)"F&fe#"y?w|
{zyxqpo5sUTpong-kjihaf_%]Ea`_^]Vz=YXQVUNrRQJnHMLKDhBGFE>=<`@9!=<;4X87wv4-
Qr0/.'K+*j(!Ef|#"yx>|u]srwvo5mrqj0nmlkMLhg`&dcE[!_A@\UyYX:POTMq4JIHlY

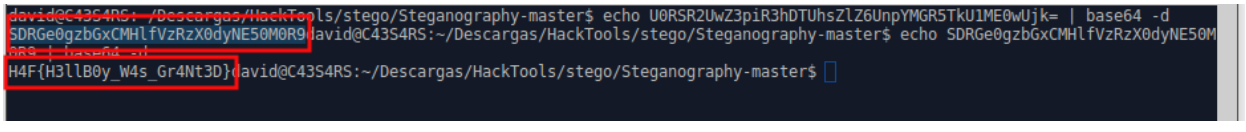
Si buscamos "Malbolge " en Google vemos que también hay un lenguaje de programación esotérico llamado "Malbolge " relacionado

Por lo que se aprecia en el ejemplo, observamos que el código es muy parecido y que no vamos mal encaminados.

Buscamos algún decoder de este lenguaje y encontramos este:



Obtenemos el siguiente string en base64, lo decodeamos:



Y por fin obtenemos la flag! H4F{H3llB0y_W4s_Gr4Nt3D}

Agradecer al equipo de Hackers4Fun por los retos tan entretenidos que se curran y por lo que aprendemos con ellos. Gracias y hasta la próxima!

POSTED IN CIFRADO, CRIPTOGRAFÍA, CTF, ESTEGANOGRAFÍA, STEGO ON 22:37 BY DAVID | LEAVE A COMMENT

0 comentarios:

Publicar un comentario

Introduce tu comentario...

Comentar como: Cuenta de

Publicar Vista previa

Suscribirse a: Enviar comentarios (Atom)

QUIERO SUSCRIBIRME

Email address... SUBM

ENTRADAS POPULARES

Haciendo un DUMP de la memoria RAM

En un análisis forense es de vital importancia conseguir una copia exacta del sistema que se vaya analizar incluyendo la memoria RAM, tenie...

Ingeniería Inversa a una APK maliciosa

Muy grande la jornada que echamos en la Hack & Beers Vol.2 sobre Mobile Security con muchos amigos. Agradeceros a todos, asistentes y ...

Scripts de geolocalización en Python

Buenas a todos, hoy os voy a hablar de como podemos montarnos unos script de geolocalización haciendo uso de los datos de los usuarios que ...

CATEGORÍAS

análisis forense (11) android (19) androidHacking (9) C43S4RS (17) Capture The Flag (2) Crypto (3) CTF (12) ethical (3) Exploiting (3) Forensics (9) hack&beers (8) hacking (39)

TODAS LAS ENTRADAS

▼ 2019 (1)

▼ marzo (1)

Writeup - Hall Of Fame - Reto 20 H4F

► 2018 (25)

► 2017 (7)

► 2016 (4)

► 2015 (9)

► 2014 (21)

► 2013 (9)

LICENCIA