



PÁGINA PRINCIPAL WRITE UPS

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

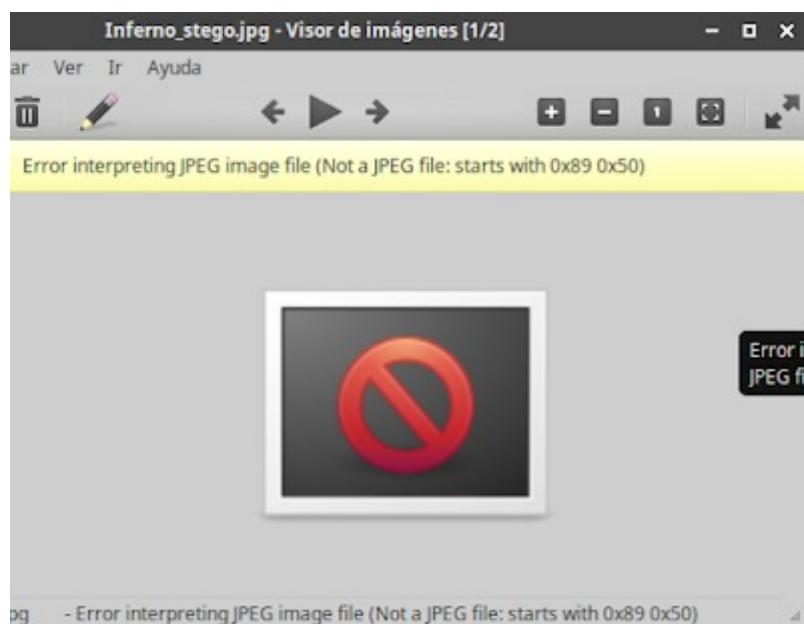
que una de ellas nos saca una nueva imagen.

```
Terminal - david@C43S4RS: ~/Descargas/CTF/h4f/Steganography
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
david@C43S4RS:~/Descargas/CTF/h4f/Steganography$ ls
GPL license.txt  R3t0_20_H4F_0gDrU_J4h4D.bmp  Steganography.cpp  Stego.exe
david@C43S4RS:~/Descargas/CTF/h4f/Steganography$ wine Stego.exe -d R3t0_20_H4F_0gDrU_J4h4D.bmp

Created on February 3, 2006 by Paul Macklin.
Uses the EasyBMP library, Version 0.71.
Licensed under GPL v. 2 by the EasyBMP Project.
Copyright (c) 2006 the EasyBMP Project
Contact: http://easybmp.sourceforge.net

Hidden data detected! Outputting to file Inferno_stego.7z ...
david@C43S4RS:~/Descargas/CTF/h4f/Steganography$ ls
GPL license.txt  Inferno_stego.7z  R3t0_20_H4F_0gDrU_J4h4D.bmp  Steganography.cpp  Stego.exe
david@C43S4RS:~/Descargas/CTF/h4f/Steganography$
```

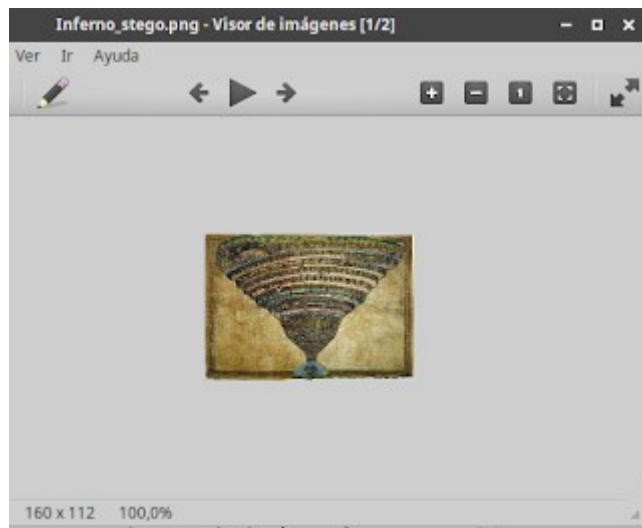
Obtenemos un archivo llamado "Inferno_stego.7z", en el título tenemos una pista y es que volvemos a tener "stego" para rato...



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

```
david@C43S4RS:~/Descargas/CTF/h4f/Steganography$ ls
GPL license.txt  Inferno_stego.7z  Inferno_stego.jpg  R3t0_20_H4F_0gDrU_J4h4D.bmp  Steganography.cpp  Stego.exe
david@C43S4RS:~/Descargas/CTF/h4f/Steganography$ file Inferno_stego.jpg
Inferno_stego.jpg: PNG image data, 160 x 112, 8-bit/color RGB, non-interlaced
david@C43S4RS:~/Descargas/CTF/h4f/Steganography$
```

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:

```
david@C43S4RS:~/Descargas/HackTools/stego/Steganography-masters$ ls
cli.py  config.yml  images  LICENSE.md  requirements.txt  steganography  tests
CNAME  flag.png.extracted  Inferno_stego.png  README.md  setup.py  steganography.egg-info
david@C43S4RS:~/Descargas/HackTools/stego/Steganography-masters$ python3 cli.py --source Inferno_stego.png --decode --output resultado.png --source-type image
File saved to resultado.png
david@C43S4RS:~/Descargas/HackTools/stego/Steganography-masters$ ls
cli.py  config.yml  images  LICENSE.md  requirements.txt  setup.py  steganography.egg-info
CNAME  flag.png.extracted  Inferno_stego.png  README.md  resultado.png  steganography  tests
david@C43S4RS:~/Descargas/HackTools/stego/Steganography-masters$
```

Y nos desvela la nueva imagen:



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

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

```
david@C43S4RS:~/Descargas/HackTools/stego/Steganography-master$ ls
cli.py          Inferno_stego.png  setup.py
CNAME           LICENSE.md         steganography
_config.yml     README.md         steganography.egg-info
_flag.png.extracted requirements.txt    tests
images          resultado.png
david@C43S4RS:~/Descargas/HackTools/stego/Steganography-master$ python3 cli.py --source resultado.png --decode --output resultado.txt --source-type image
File saved to resultado.txt
david@C43S4RS:~/Descargas/HackTools/stego/Steganography-master$ cat resultado.txt
D`N^#"=~}:F27wfAQtrN`.nml7jGhE%1d@y~`<:;)]xqpo5srqSi/glkdcB(`_^]#a`YX|\UTSXQV0s54PIHMLEiI+G@ED=<`#?8=<54X87w/43,P*).`&+*#G!g%|#`y
>v^tsr8votmrk1oQgfkjib(`e^cb[Z~^]?UZSRvP8NSRKPONGkE-CHA@dDCB$@?>=<5Y9y1U543,r*N.-,+k)"F&fe#"y?w|{zyxqpo5sUTpong-kjihaf_%]Ea`_^]Vz
-YXQVUNrRQJnHMLKDhBGFE>=<`@9!<=<4X87wv4-Qr0/.`K+*j(!Ef|#`yx>|u]srwvo5mrqj0nmlkMLhg`&dcE[!_A@UyYX:POTMq4JIHIY
david@C43S4RS:~/Descargas/HackTools/stego/Steganography-master$
```

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

```
D`N^#"=~}:F27wfAQtrN`.nml7jGhE%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:POTMq4JIHIY
```

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

Malbolge - Wikipedia

<https://en.wikipedia.org/wiki/Malbolge> ▼ Traducir esta página

Malbolge is a public domain esoteric programming language invented by Ben Olmstead in 1998, named after the eighth circle of hell in Dante's *Inferno*, the ...

Designed by: Ben Olmstead **Filename extensions:** mal,.mb

First appeared: 1998 **Typing discipline:** Untyped

Malbolge es un lenguaje de programación esotérico de dominio público desarrollado por Ben Olmstead en 1998. Se llamó así por el octavo círculo del infierno en *La Divina Comedia*, escrito por Dante.

Malbolge es peculiar porque se diseñó para ser el lenguaje más difícil. Sin embargo, varios de los trucos utilizados para hacerlo difícil de entender pueden ser evitados.

Código de ejemplo Hello World:¹

```
(=<`$9]7<SYXz7wT.3,+0/o'K%$H''-D|#z@)b='{^LxB%$Xmrkpohm-kNi;gsedcba`_^}|ZYXWVUTSRQPONMLKJIHGFEDCBA@?>=<;:9876543s+0<olm
```

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](#):

Malbolge Tools

Interpreter

Generator

Load program from file

Choose example: Hello World EU version

Load

Program code: ☐ Normalized

```
D`N^#"==}:F27wfaQtrN`.nmI7jGhE%ld@y~`<;)lxqpo5srqSi/glkdcB(`_^]#a`YX|\UTSXQV0s554PI
HMLEiI+G@ED=<`#78=<54X87w
/43,P*).`&+*#G!g%|#`yx>v^tsr8votmrk1oQgfkjib(`e^cb[Z~^]?UZSRvP8NSRKPONGkE-
CHA@dDCB$@?>=<5Y9y1U543,r*N.-,+k)"F&fe#"y?w|{zyxqpo5sUTpong-
kji haf %]Ea`^]Vz=YXQVUNrRQJnHMLKDhBGFE>=<@9!<4X87wv4-
Qr0/.`K+*j(!Ef|#`yx>|u]srwvo5mrqj0nmlkMLhg`&dcE(!_A@UyYX:P0TMq4JIHLY
```

Execute

U0RSR2UwZ3piR3hDTUhsZlZ6UnpYMGR5TkU1ME0wUjk=

Program finished.

Obtenemos el siguiente string en base64, lo decodeamos:

```
david@C4364R6:~/Descargas/HackTools/stego/Steganography-master$ echo U0RSR2UwZ3piR3hDTUhsZlZ6UnpYMGR5TkU1ME0wUjk= | base64 -d
SDRGe0gzbgxCMHlfVzRzX0dyNE50M0R9
david@C4354RS:~/Descargas/HackTools/stego/Steganography-master$ echo SDRGe0gzbgxCMHlfVzRzX0dyNE50M0R9 | base64 -d
H4F{H3lLB0y_W4s_Gr4Nt3D}
david@C4354RS:~/Descargas/HackTools/stego/Steganography-master$
```

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](#)

[Página principal](#)

[Entrada antigua](#)

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



C43S4RS is licensed under a [Creative Commons Reconocimiento–NoComercial 3.0 España License](#).
