

# ÍNDICE

— □ ×

0.- INTRODUCCIÓN			
1.- OSINT			
1.1.- Recibiste un correo	3	4.2.- Hermes	12
1.2.- Noticias importantes	4	5.- FORENSE	
1.3.- Todo H4ck3r tiene un blog	5	5.1.- Arde Troya	13
2.- MISCELLANEOUS		5.2.- Titanomachia	14
2.1.- Las Parcas - Cloto	6	6.- WEB	
2.2.- Las Parcas - Laquesis (1/2)	7	6.1.- Tártaro	15
2.3.- Las Parcas - Laquesis (2/2)	8	6.2.- Circe	16
3.- STEGO		6.3.- Caronte	17
3.1.- Cassandra	9	7.- RETOS OPCIONALES	
3.2.- Medusa	10	7.1.- Las parcas - Atropos (1/2)	18
4.- CRIPTOGRAFÍA		7.2.- Las parcas - Atropos (2/2)	19
4.1.- Minotauro	11	7.3.- Buscando a Matteo (1/2)	20
		7.4.- Buscando a Matteo (2/2)	21

# INTRODUCCIÓN

— □ ×

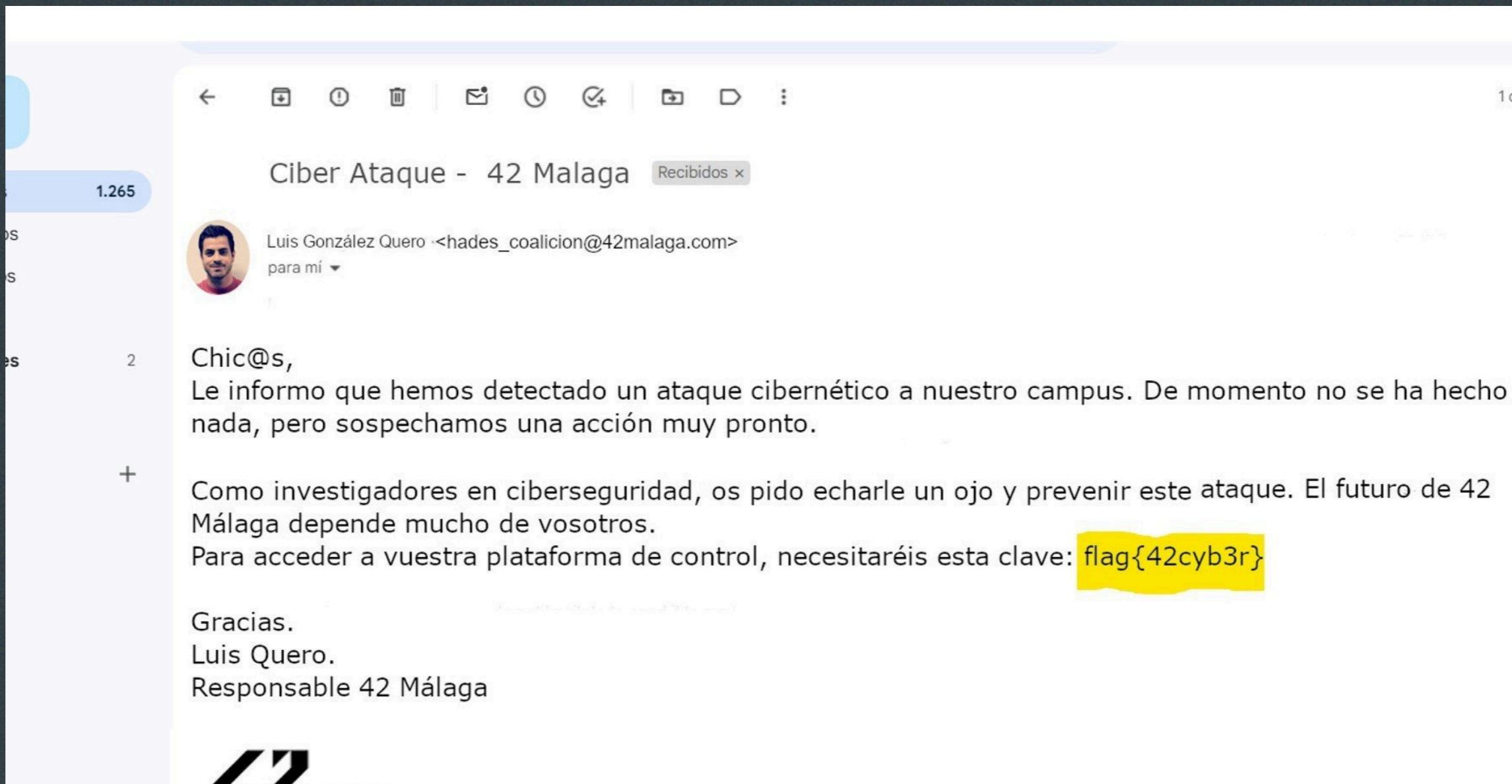
Un CTF representa un espacio de aprendizaje práctico y competitivo en el ámbito de la ciberseguridad, que permite a los participantes mejorar sus habilidades técnicas. Este documento tiene como objetivo proporcionar una visión general de lo que es un CTF, su importancia en el desarrollo de habilidades de ciberseguridad, y presentar una resolución detallada del mismo.

El CTF de Coaliciones contiene 16 retos de distintas temáticas (Web, Stego, Criptografía, Forense, OSINT, etc). La mitología griega gira alrededor de cada 'challenge' y nos acompaña a la vez que obtenemos las 'flags'. No se requieren conocimientos avanzados ni el uso de máquinas virtuales para su realización. La puntuación no siempre refleja la dificultad del reto. Está permitido el uso de 'Google' o cualquier 'chatbot' / 'IA'.



# OSINT > Recibiste un correo

En el primer reto de OSINT (proceso de recopilación y análisis de información de fuentes públicas para identificar amenazas, vulnerabilidades y otros datos relevantes para la seguridad informática) nos dan una imagen en formato '.jpg' en la que vemos una captura de un correo. Si leemos todo el contenido, podemos apreciar la flag. La introducimos y continuamos con el siguiente.



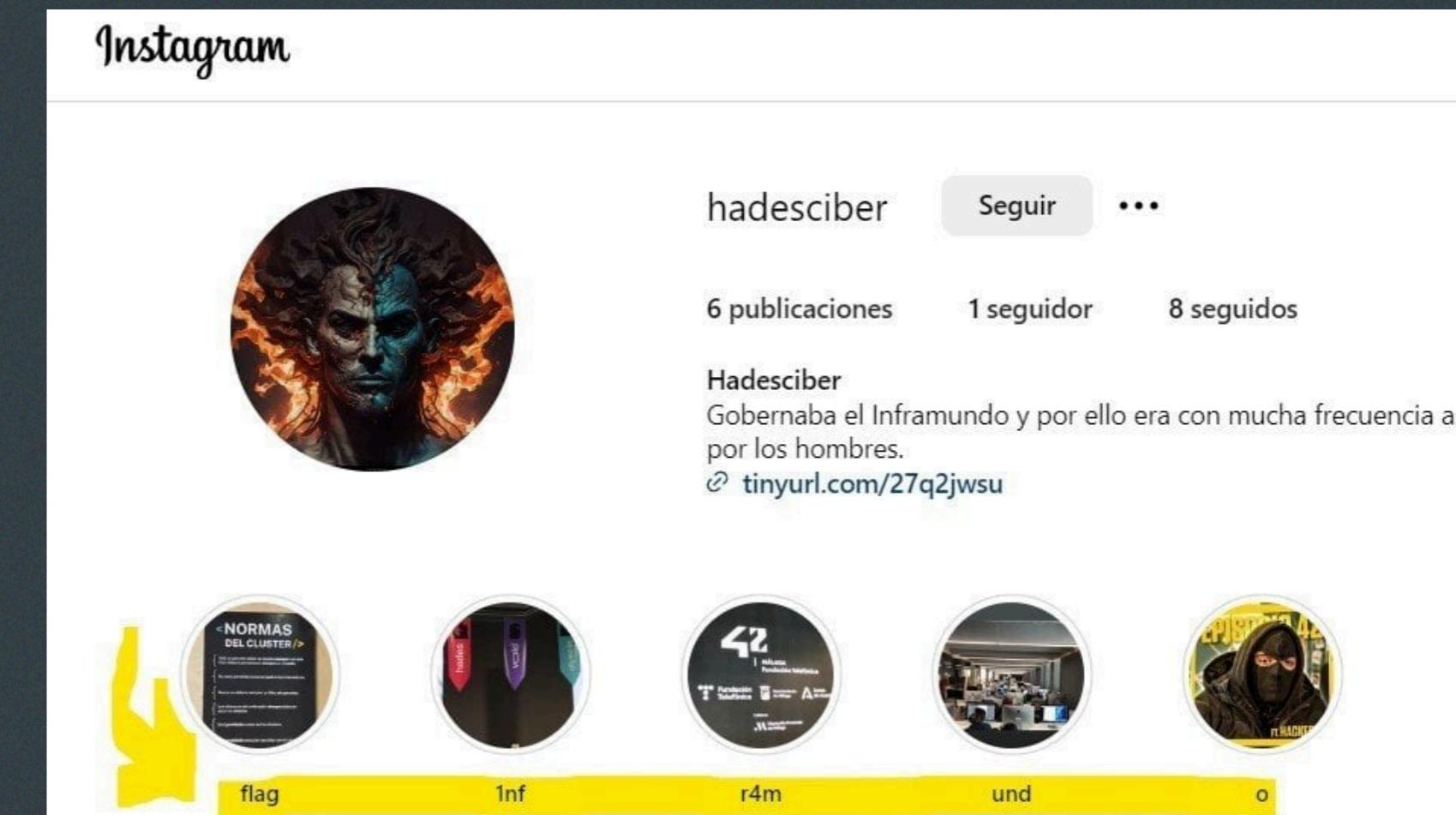
# OSINT > Noticias importantes

Para este reto, tenemos un archivo en formato 'PDF' que simula un periódico. En la segunda página, hay un apartado que habla sobre un "hackeo" a 42. Nos menciona una cuenta de Instagram. Si la buscamos, podemos ver la flag juntando el título de las historias del perfil.

## Hadesciber reivindica en hackeo de 42 Málaga

En su cuenta instagram @Hadesciber, un grupo de hacker se atribuyó el hackeo al campus de programación 42. De momento no se sabe ningún motivo y el campus no ha comunicado sobre el hecho.

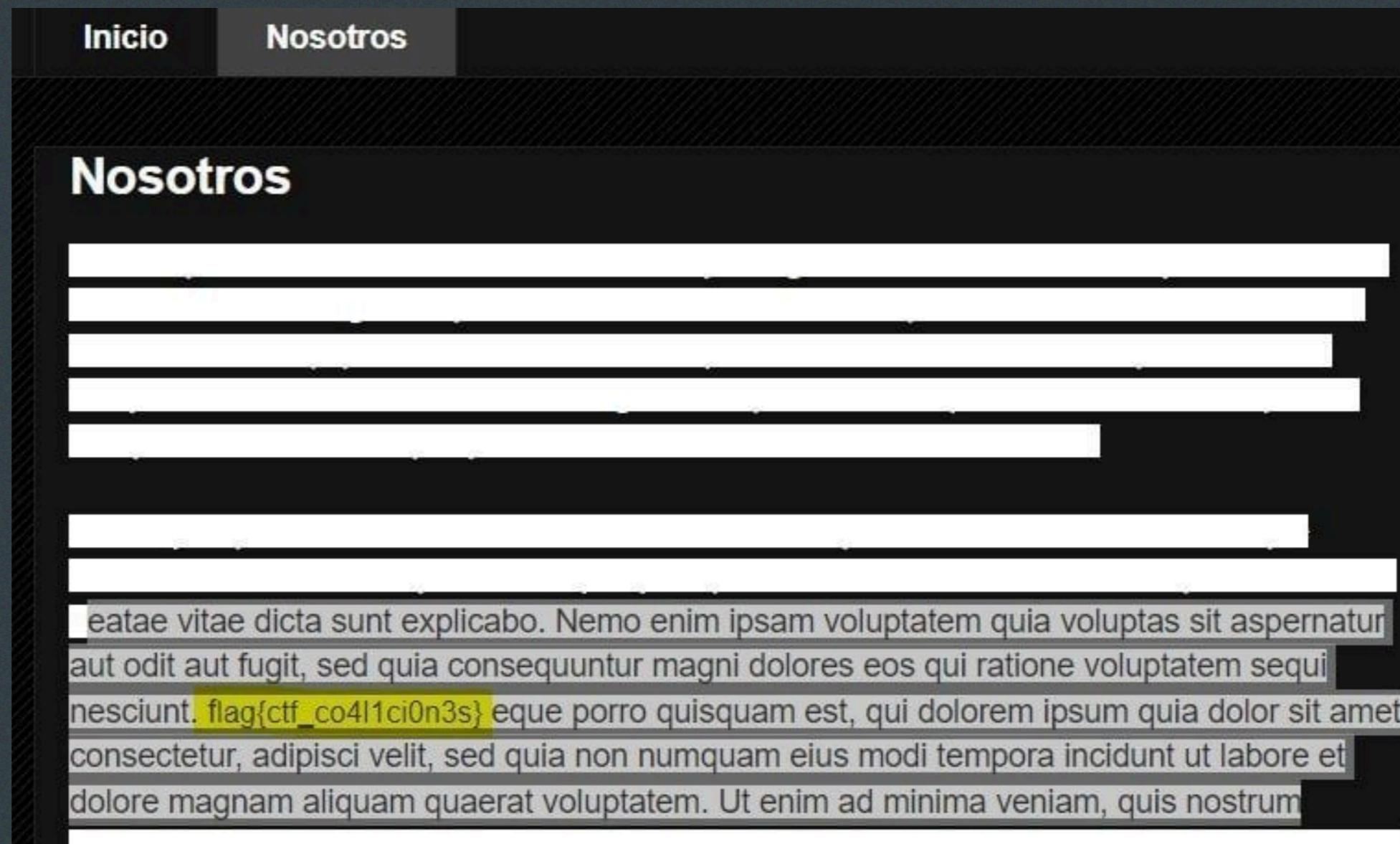
Últimamente, los ciberataques se multiplicaron, y la demanda de expertos en ciberseguridad también. Os tendremos informado próximamente cuando tendremos más informaciones.



# OSINT > Todo H4ck3r tiene un blog

- □ ×

En este challenge nos piden que visitemos la URL que hay en el perfil de Instagram, la cual nos lleva a la página de inicio de un 'Blog' donde no hay contenido relevante. Sin embargo, si visitamos la página de 'Nosotros', vemos dos párrafos en los que, si seleccionamos el texto con el ratón, podremos visualizar el contenido de manera clara, y en él estará la 'flag'.



# MISCELLANEOUS > Las Parcas - Cloto

— □ ×

Cambiamos de temática. La información necesaria la obtendremos en el perfil del 'Blog', que nos llevará a un repositorio de GitHub. Para obtener esta flag, tenemos que conectarnos a 'parcas.ddns.net' por el puerto '4242' usando el protocolo 'SSH'.

Las credenciales las obtendremos en el mismo GitHub. Una vez dentro, podemos lanzar el comando 'ls' para ver el contenido del directorio. Entre todos los archivos, destaca uno llamado 'script.sh'. Al lanzarle un 'cat', podremos identificar rápidamente un script en 'bash' que parece que se ha usado para esconder la flag. En este interpretamos que se han creado 201 archivos ocultos, que la flag se ha introducido de manera aleatoria en uno de ellos y que el tamaño del archivo es mayor a 24 bytes y menor a 27. Usando 'ls -la', listaremos todos los archivos ocultos junto a la información necesaria para identificar al que estamos buscando. Hay 4 archivos con un tamaño entre 24 y 27 bytes. Podemos visualizarlos con 'cat' y probar cada uno. También podríamos usar 'cat .hidden\_flag\_\*' para visualizarlos todos y probar con cada uno, dado que no hay límite de intentos al probar las 'flags'.

```
cloto@hades42ctf:~$ cat .hidden_flag_*
flag{first_one_but_not_the_one}
flag{this_is_the_flag}
flag{this_also_could_be_the_flag}
flag{la_banda_del_patio}
flag{Mr.R0b0t}
flag{mi_mono_amedio_y_yo}

flag{cl0t0_th1s_w4s_e4sy}
flag{perrito_piloto}
flag{4tr0p0s_is_waiting}
cloto@hades42ctf:~$
```

@42malagaftef\_

```
cloto@hades42ctf:~$ cat .hidden_flag_59 .hidden_flag_77 .hidden_flag_190 .hidden_flag_167
flag{cl0t0_th1s_w4s_e4sy}
flag{4tr0p0s_is_waiting}
flag{mi_mono_amedio_y_yo}
flag{la_banda_del_patio}
cloto@hades42ctf:~$
```

Segundo reto de 'Miscellaneous', tendremos que realizar una conexión similar a la anterior pero con credenciales distintas.

Para hacer esto, podríamos salir de la sesión actual con 'exit' y volver a lanzar el comando 'ssh' completo con el nuevo login, o dentro de la misma sesión lanzar 'su laquesis' para cambiar a ese usuario. Una vez estemos conectados con el usuario laquesis y en su directorio '/home/laquesis', usando 'ls -a' llegaremos hasta un directorio llamado '.readme', el cual contiene un documento de texto llamado '.readme'. Podremos distinguir uno de otro usando 'file'. Este documento nos da la información necesaria para continuar; tenemos que buscar un archivo '.txt' cuyo nombre contiene 5 números además de letras, y está ubicado dentro de alguna carpeta del directorio '/usr'. Para lograr esto, podemos utilizar dos estrategias: buscar simplemente los archivos '.txt' que haya dentro de '/usr' o tratar de dar con el comando exacto que nos muestre ese archivo. Intentándolo con la primera opción, podríamos usar:'find /usr -type f -name "\*.[txt]". Este comando nos mostrará muchos archivos '.txt', y si queremos revisar sus nombres uno por uno para ver si alguno nos llama la atención, tardaríamos 2-3 minutos. El comando para sacar el archivo que nos indican en el 'readme' lo conseguiríamos jugando con los '\*' para sustituir parte del nombre que desconocemos:'find /usr -type f -name "\*[0-9]\*[0-9]\*[0-9]\*[0-9]\*[0-9]\*.txt".

```
laquesis@hades42ctf:~$ find /usr -type f -name "*.[txt]"
/usr/local/lib/python3.11/dist-packages/pyinstaller-6.3.0.dist-info/entry_points.txt
/usr/local/lib/python3.11/dist-packages/pyinstaller-6.3.0.dist-info/top_level.txt
/usr/local/lib/python3.11/dist-packages/pyinstaller-6.3.0.dist-info/COPYING.txt
/usr/local/lib/python3.11/dist-packages/altgraph-0.17.4.dist-info/top_level.txt
/usr/local/lib/python3.11/dist-packages/pyinstaller_hooks_contrib-2024.0.dist-info/entry_points.txt
/usr/local/lib/python3.11/dist-packages/pyinstaller_hooks_contrib-2024.0.dist-info/LICENSE
/usr/local/lib/python3.11/dist-packages/pyinstaller_hooks_contrib-2024.0.dist-info/top_level.txt
/usr/local/lib/python3.11/dist-packages/pyinstaller_hooks_contrib-2024.0.dist-info/LICENSE
/usr/lib/google-cloud-sdk/platform/gsutil/tests/test_data/test.txt
/usr/lib/google-cloud-sdk/platform/gsutil/vendored/boto/requirements.txt
/usr/lib/google-cloud-sdk/platform/gsutil/vendored/boto/boto/cacerts/cacerts.txt
/usr/lib/google-cloud-sdk/platform/gsutil/vendored/boto/tests/integration/s3/other.txt
/usr/lib/google-cloud-sdk/platform/gsutil/vendored/boto/requirements-py33.txt
/usr/lib/google-cloud-sdk/platform/gsutil/vendored/boto/requirements-docs.txt
/usr/lib/google-cloud-sdk/platform/gsutil/gslib/vendored/oauth2client/docs/requirements.txt
/usr/lib/google-cloud-sdk/platform/gsutil/gslib/data/cacerts.txt
/usr/lib/google-cloud-sdk/platform/gsutil/third_party/pyparsing/test/parsefiletest_input.txt
/usr/lib/google-cloud-sdk/platform/gsutil/third_party/pyparsing/requirements-dev.txt
/usr/lib/google-cloud-sdk/platform/gsutil/third_party/idna/tests/IdnaTestV2.txt
```

```
laquesis@hades42ctf:~$ find /usr -type f -name "*[0-9]*[0-9]*[0-9]*[0-9]*[0-9]*.txt"
/usr/lib/google-cloud-sdk/platform/gsutil/third_party/chardet/tests/iso-8859-9-turkish/wikitop_tr_ISO-8859-9.txt
/usr/lib/google-cloud-sdk/platform/bundledpythonunix/lib/python3.11/site-packages/.l4s_m0ir4s/l4_qu3_4s1gn4.txt
find: '/usr/share/polkit-1/rules.d': Permission denied
laquesis@hades42ctf:~$
```

En el directorio del archivo '.txt', encontramos dos archivos. Si ejecutamos el comando 'file' sobre ellos, podremos identificarlos. El primero es un documento de texto que nos proporciona información sobre cómo continuar y ofrece algunas pistas. El segundo parece ser un archivo encriptado, específicamente 'openssl enc'd data with salted password'. Ahora, debemos buscar cómo desencriptar este archivo. Realizando una búsqueda rápida, encontramos un comando que utiliza 'openssl' para desencriptarlo. Es importante tener en cuenta la pista que nos proporcionan en el documento de texto (hint: -des-ede3-cbc), que en realidad es el algoritmo utilizado para encriptar el archivo. El comando final sería: 'openssl enc -d -des-ede3-cbc -in l4\_qu3\_4s1gn4.txt'. Ahora nos pide una contraseña. Si revisamos nuevamente el documento de texto, menciona un 'email', el cual encontramos en la información del reto: Correo:lasparcas@hades42ctf.com. Probamos con este correo y finalmente logramos desencriptar el archivo y obtener la flag.

Otra opción sería volcar el contenido en otro documento, pero para ello tendríamos que crear una carpeta en '/tmp' y copiar el archivo allí con 'cp', dado que en la carpeta actual no tenemos permisos. El comando que necesitaríamos sería: 'openssl enc -d -des-ede3-cbc -in l4\_qu3\_4s1gn4.txt -out file.txt', lo abriríamos con 'cat' y dentro estaría la flag.

```
laquesis@hades42ctf:/usr/lib/google-cloud-sdk/platform/bundledpythonunix/lib/python3.11/site-packages/.14s_m0ir4s$ openssl enc -d -des-ede3-cbc -in l4_qu3_4s1gn4.txt
enter DES-EDE3-CBC decryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
flag{l4qu3s1s_d0s/tR3s_~}
laquesis@hades42ctf:/usr/lib/google-cloud-sdk/platform/bundledpythonunix/lib/python3.11/site-packages/.14s_m0ir4s$
```

# STEGO > Cassandra

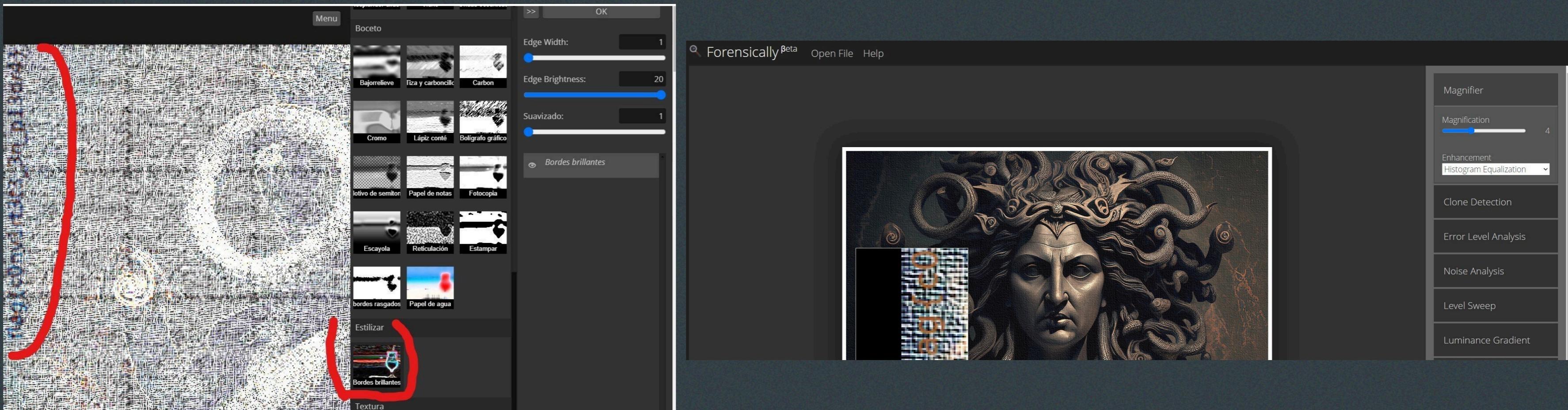
Desbloqueamos otro tema, este nos redirige al Github mencionado previamente en el cual hay una imagen que tenemos que descargar, llamada: 'mito-de-cassandra.jpg'. En el mismo reto ya nos dan una pista cuando nos indican el formato de esta flag-> formato: flag{m3t4d4t0s}. Sabiendo esto, tenemos que buscar una página web que nos muestre los metadatos de un archivo, en este caso usaremos: 'https://www.metadata2go.com', dado que es muy intuitiva y sencilla de usar. Una vez analizamos los metadatos de la imagen, podemos distinguir un string idéntico en distintos apartados. Si hemos visto algún 'base64' antes, lo podemos relacionar a simple vista. En nuestro caso, dado que queremos averiguar de qué se trata, lo pasaremos por la herramienta: 'https://gchq.github.io/CyberChef/'. Usaremos esta porque tiene infinidad de opciones y no requiere conocimientos para su uso, además cuenta con una 'varita mágica' que básicamente es un identificador de cifrado, que muchas veces nos hace todo el trabajo con un simple 'click'. Como podemos ver en la captura, podremos obtener la flag, usando la 'varita mágica' x3, que lo que hace es decodificar el texto 3 veces 'from base64'.

The screenshot shows two main sections. On the left is the 'METADATA2GO.com' interface, displaying a file list for 'mito-de-cassandra.json'. It contains three entries: 'image\_description' with value 'V20xNGFGb3pkRzlaVjFKc1kzazFiR041TlhGaU1rWjRaRmRzZFdauIBUMD0='; 'xp\_title' with value 'V20xNGFGb3pkRzlaVjFKc1kzazFiR041TlhGaU1rWjRaRmRzZFdauIBUMD0=' (highlighted in yellow); and 'padding' with value '(Binary data 268 bytes)'. On the right is the 'CyberChef' tool interface, which has processed the 'xp\_title' value. It shows three stages of Base64 decoding:

- Input:** V20xNGFGb3pkRzlaVjFKc1kzazFiR041TlhGaU1rWjRaRmRzZFdauIBUMD0=
- Decoding Stage 1:** Recipe: From Base64, Alphabet: A-Za-z0-9-\_ (checkbox checked), Output: V20xNGFGb3pkRzlaVjFKc1kzazFiR041TlhGaU1rWjRaRmRzZFdauIBUMD0=
- Decoding Stage 2:** Recipe: From Base64, Alphabet: A-Za-z0-9+= (checkbox checked), Output: V20xNGFGb3pkRzlaVjFKc1kzazFiR041TlhGaU1rWjRaRmRzZFdauIBUMD0=
- Decoding Stage 3:** Recipe: From Base64, Alphabet: A-Za-z0-9+= (checkbox checked), Output: flag{hadess.es.joaquin}

# STEGO > Medusa

Continuamos con el siguiente reto de esteganografía (práctica de ocultar información dentro de otro mensaje u objeto para evitar su detección), de nuevo en el formato de flag nos dan otra pista-> formato: flag{imagen\_negrativa}. Si hacemos una rápida búsqueda sobre esto, sabremos que tenemos que buscar algo que haya en la imagen invirtiendo los colores de la misma, para ello podemos usar: '<https://fixthephoto.com/es/gimp-online.html>', o alguna herramienta como '<https://29a.ch/photo-forensics>', ambas online y no requieren instalación previa. En 'Gimp-online', tendremos que probar con los filtros disponibles, nos iríamos a 'Filtro > Galería de filtros... > Bordes brillantes'. En nuestro caso sabremos que es ese porque tras probar con todos es en el que más se distinguen las palabras que están en la esquina superior izquierda. Usando la otra página, en la primera herramienta 'magnifier', ya podríamos visualizar la flag moviendo el ratón por toda la imagen hasta llegar a la esquina superior izquierda.



Pasamos a un tema nuevo 'Criptografía' (práctica de desarrollar y utilizar algoritmos codificados para proteger y ocultar la información transmitida, asegurando que solo puedan ser leídas por aquellos con permiso y capacidad de descifrarla).

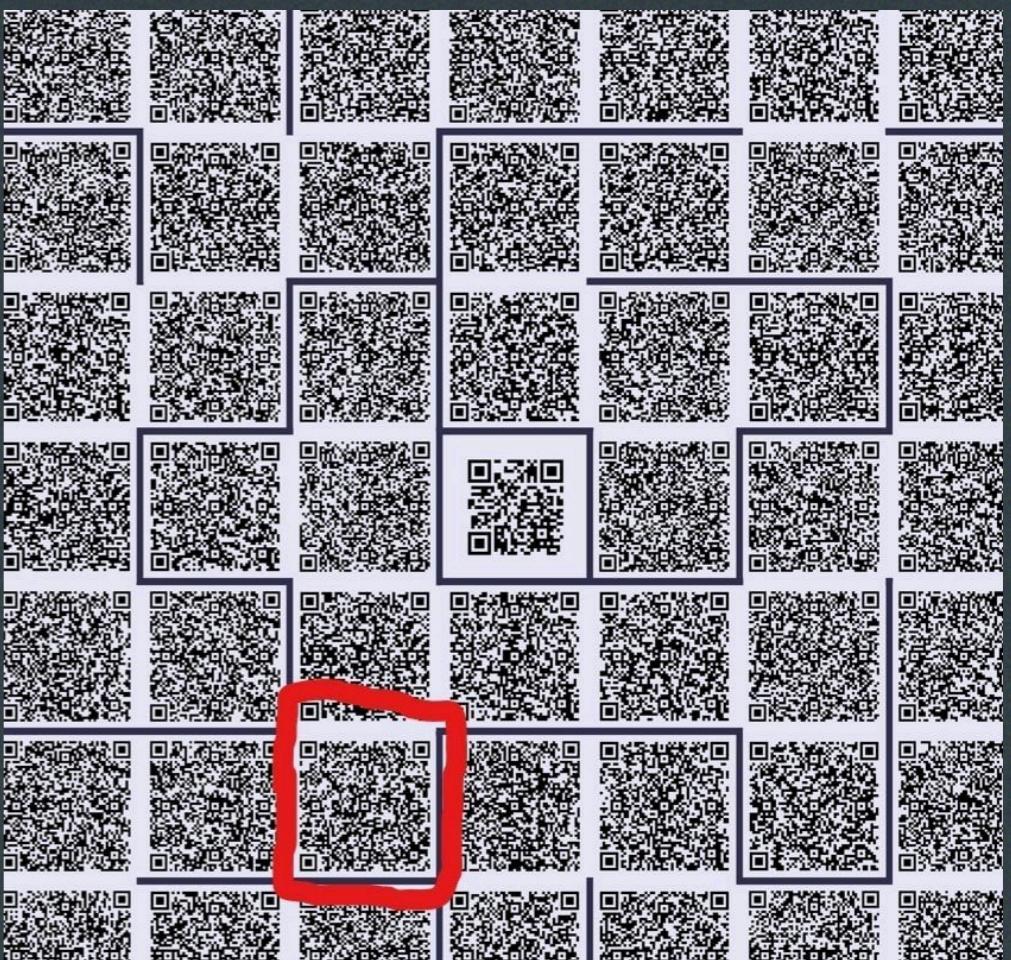
Para resolver este reto nos dan unas instrucciones que, a simple vista, viendo en qué posición están las mayúsculas, sabremos que está al revés. Además, nos dan una imagen que tendremos que descargar, se trata de un laberinto repleto de 'QRs' y para llegar al 'QR' correcto tendremos que darle la vuelta al texto. Utilizaré esta página: '<https://www.topster.es/texto/umdrehen.html>'. Ya tenemos las instrucciones totalmente legibles, ahora solo tenemos que seguir cada paso que indica. Cuando tengamos el correcto, lo escaneamos con un lector de QRs, y nos dará unas coordenadas en 'Google Maps', esa es nuestra flag.

● Volteado horizontal (Escritura de espejo) Ejemplo: retspoT  
○ Volteado hacia atrás, vertical Ejemplo: r̄etspoT  
○ Texto tachado Ejemplo: Topster

sert rajab setna nis on ,adreiuqzi al aicah sosap sod rad sebeD .ojaba aicah sod y sod adreiuqzi al aicah arohA .sod ebuS .rartne la ohceh sah euq sosap ed daditnac amsim al ahcered al aicah aunitnoC .ojaba aicah ohceh sah euq abirra aicah sosap ed daditnac amsim al eV .sosap sod aznava y ahcered al a arig ,ogeul .onu rajab setna nis on ,adreiuqzi al aicah sosap sert rad sebeD .ojaba aicah onu y onu adreiuqzi al aicah arohA .sert ebuS .rartne la ohceh sah euq sosap ed daditnac amsim al ahcered al aicah aunitnoC .ojaba aicah sosap ed daditnac amsim al eV .onu aznava y ahcered al a arig ,abirra aicah sosap sod noc otnirebal la artnE

Girar texto

Entra al laberinto con dos pasos hacia arriba, gira a la derecha y avanza uno. Ve la misma cantidad de pasos hacia abajo. Continua hacia la derecha la misma cantidad de pasos que has hecho al entrar. Sube tres. Ahora hacia la izquierda uno y uno hacia abajo. Debes dar tres pasos hacia la izquierda, no sin antes bajar uno. Luego, gira a la derecha y avanza dos pasos. Ve la misma cantidad de pasos hacia arriba que has hecho hacia abajo. Continua hacia la derecha la misma cantidad de pasos que has hecho al entrar. Sube dos. Ahora hacia la izquierda dos y dos hacia abajo. Debes dar dos pasos hacia la izquierda, no sin antes bajar tres



# CRPTOGRAFIA > Hermes

— □ ×

En el segundo reto de 'cripto', nos dan un archivo '.txt' cuyo contenido parece estar cifrado. Además, añaden una página que podríamos usar: '<https://www.boxentriq.com>'. En ella hay una herramienta muy útil que se llama 'cipher identifier', en la cual copiaremos el texto y nos indicará que podría tratarse de 'Vigenère Cipher'. Esta misma página nos redireccionará a su propia herramienta que tiene para descifrar este algoritmo, copiamos el texto y clicamos en el recuadro 'Auto solve without key'. Nos mostrará un texto en el que está la flag.

The screenshot shows the Boxentriq homepage with several tools listed:

- Upide down text**: An icon of a document with text rotated 180 degrees.
- Word unscrambler**: An icon of a document with text that appears to be jumbled or unscrambled.
- Analysis Tools**: A large heading.
- Binary analysis**: An icon of a binary file.
- Cipher identifier**: An icon of a document with a yellow highlight.
- Hex analysis**: An icon of a hex dump.
- Text analysis**: An icon of a document with text.

Below the tools, it says: "These tools will help you identify the types of encodings and ciphers used."

The screenshot shows the 'Vigenère Tool' page on Boxentriq:

- Text Input:** Zuycg aoeatk... (ciphertext)
- Buttons:** Copy, Paste, Text Options...
- Key Input:** Type key here...
- Mode Selection:** Standard Mode (dropdown), English (dropdown)
- Tool Buttons:** Decode, Encode, Auto Solve (without key) (highlighted in green), Instructions
- Auto Solve Options:** Min Key Length: 3, Max Key Length: 10, Iterations: 100, Max Results: 10, Spacing Mode: Automatic (dropdown)
- Auto Solve results:**

Score	Key	Text
23366	malaga	nunca aceptes la forma en que las cosas se han hecho siempre como la unica forma en que pueden hacerse se que estan tratando de tener el ciberataque no voy robar mucho si me dejan en paz los perdonare te parece si flag <b>negociamos</b> las negociaciones suelen ir mejor cuando existe la confianza

# FORENSE > Arde Troya

Pasamos a otro ámbito, 'forense', el cual se refiere al análisis forense digital. Este tipo de análisis implica examinar datos digitales para recopilar evidencias que puedan ser utilizadas en investigaciones de ciberseguridad. Para este challenge nos dan una imagen, en la que nos indican que debemos buscar y un archivo '.pcapng' (formato de archivo de captura de paquetes de red que se utiliza para almacenar datos de tráfico de red), el cual debemos analizar para dar con una IP en concreto.

Podríamos usar 2 herramientas online, estas dos en específico no requieren registrarse: 1.- <https://apackets.com/>. 2.-<https://app.packetsafari.com/>. 'Apackets' es un poco más sencilla de usar y separa todos los datos por categoría, en nuestro caso solo necesitamos ir a 'Connections'. 'Packetsafari' nos muestra la información más detallada, desde la pantalla de 'log' podemos distinguir todas las IPs capturadas. Probamos con la que se repite tras haber realizado múltiples conexiones y obtendremos la flag.

log											Start typing to search for fields...								
Total: 271	Displayed: 271	/	General	TCP	DNS	TLS	Misc	Performance	DHCP	Infrastructure	STP	ARP	WiFi	ICMP	IP	SMB	HTTP	Errors	
Number	Time	Delta time	Source	Destination	Protocol	Length	Info												
17	10.405674000	0.230782000	10.18.207.131	10.18.207.131	TCP	56	56042 → 21 [SYN] Seq												
18	10.405735000	0.000061000	10.18.207.131	10.18.207.131	TCP	56	21 → 56042 [SYN, ACK]												
19	10.405810000	0.000075000	10.18.207.131	10.18.207.131	TCP	44	56042 → 21 [ACK] Seq												
20	10.406414000	0.000604000	127.0.0.1	127.0.0.1	TLSv1.2	328	Application Data												

The screenshot shows the Apackets interface with the following details:

- Left Sidebar:** A-Packets, Browse Captures (log.pcapng selected), Data Overview, Credentials 2, DNS 4, HTTP Headers, Connections 6, Open Ports 7 (highlighted), SSL/TLS, Pictures, HTTP, SMB, Servers, Documents, Network 14, FTP 4, Telnet, SSDP.
- Right Panel:** A tree view of network entities:
  - From IP or DNS: 10.18.207.131 (highlighted)
  - 127.0.0.1
  - 192.168.64.1
  - 192.168.44.1
  - 10.18.207.131 (highlighted again)
  - 192.168.56.1

# FORENSE > Titanomachia

— □ ×

Último reto de 'Forense', en este nos dan un archivo comprimido '.zip', en el que tenemos 1 documento de texto que parece ser una 'wordlist' y 1 archivo comprimido que requiere una contraseña para extraer su contenido. Seguramente tendrás que utilizar alguna palabra de la wordlist para ello. La forma más rápida para poder abrir el archivo sería usar un script que haga fuerza bruta y pruebe con cada palabra de la 'wordlist' hasta dar con la correcta. Hay muchos scripts en Github que hacen esto o incluso podríamos pedirle a algún 'chatbot' que nos haga uno. Una vez consigamos extraer el contenido nos da un documento en el que hay una especie de strings con puntos, y si lo googleamos nos dice que es 'braille'. Entonces tenemos que buscar un traductor de 'braille' a texto, usaremos este: '<https://abcbraille.com/braille>'. Copiamos y pegamos, y nos dará la esperada flag.

```
-zsh  
briferna@MacBook-Pro-de-Brian Archivo % python3 zip-cracker.py titano.zip 42.txt  
¡Contraseña encontrada: Atenea Archivos extraídos correctamente.  
briferna@MacBook-Pro-de-Brian Archivo % ls  
42.txt      anonymus      titano.zip      zip-cracker.py  
briferna@MacBook-Pro-de-Brian Archivo % cat anonymus  
.:;::.. :;:::;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;:;  
briferna@MacBook-Pro-de-Brian Archivo %
```

The screenshot shows the ABCBraille website interface. At the top, there is a navigation bar with links: ABCBraille, Image to Text, Contraction Lookup, Text to Braille, and Braille to Text. Below the navigation bar, there is a dropdown menu labeled "Language table: UEB Grade 2 - en-ueb-g2.ctb". The main area is titled "Braille to Text Translation". It features a text input field containing Braille characters (dots) and a "Style" button. Below the input field is a large text area displaying the translated text. A blue button at the bottom of this area is labeled "Translate Braille". At the very bottom of the page, there is a footer with the text "flag: c4s1-m3-p1ll4n".

# WEB > Tártaro

Llegamos a la recta final, los retos de esta temática implican explorar y encontrar vulnerabilidades en aplicaciones web. En este nos dan una dirección: '<https://tartaro-y-circe-hadesctfs-projects.vercel.app>'. Al entrar, podemos ver un cuadro de 'log-in' para iniciar sesión, con lo que tendremos que buscar credenciales. Para ello, clicaremos con el botón derecho en la página y seleccionaremos 'inspeccionar'. A continuación, nos iremos a 'fuentes' y nos mostrará los archivos que componen la página. El que nos interesa es 'scriptindex.js', el cual realiza una validación básica del inicio de sesión y nos redirige a una página específica. Los credenciales que buscamos los encontraremos en el condicional 'if'. Una vez iniciemos sesión, nos mostrará una supuesta 'flag' en mitad de la pantalla, pero si tratamos de validarla no la aceptará. La flag correcta la encontramos en el código fuente de la página.

The screenshot shows the browser's developer tools with the 'Elements' tab selected. On the left, the file structure is shown with 'scriptindex.js' highlighted. The main pane displays the code of 'scriptindex.js':

```
document.getElementById('loginForm').onsubmit = function(event) {
    event.preventDefault();
    var uname = document.querySelector('input[name="uname"]').value;
    var psw = document.querySelector('input[name="psw"]').value;
    if(uname === 'joacking' && psw === 'DMBANV') {
        window.location.href = '../hades3/hades3.html';
    } else {
```

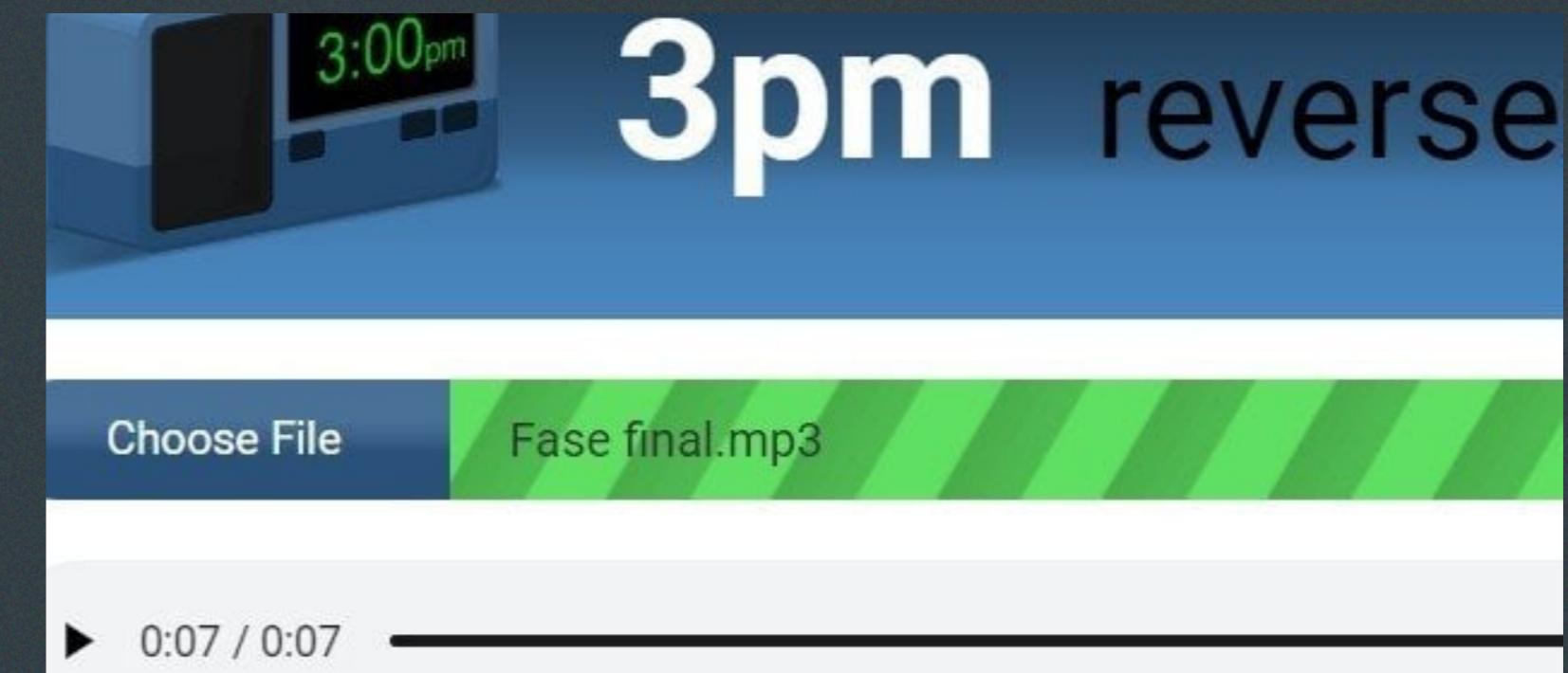
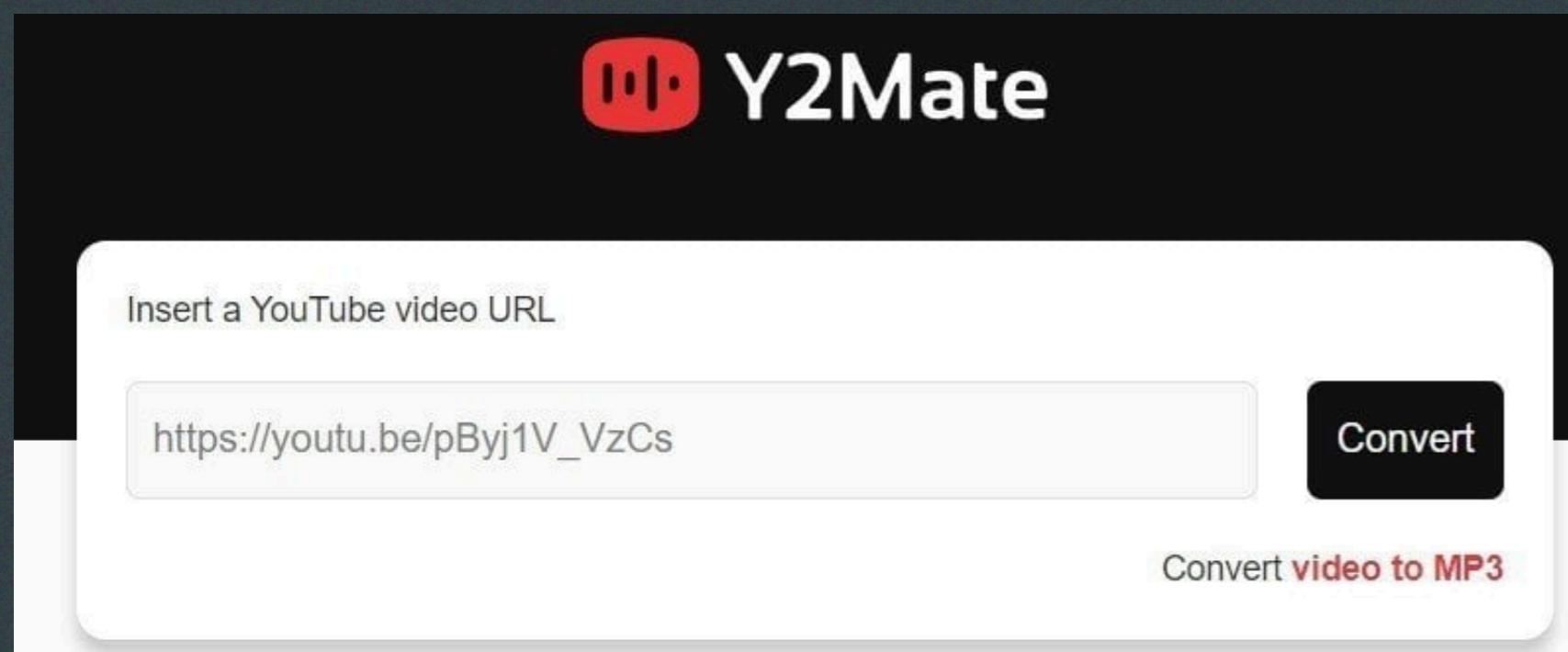
The screenshot shows the browser's developer tools with the 'Elements' tab selected. The code pane displays the full HTML source of the page:

```
<!DOCTYPE html>
<html lang="en">
...<head> == $0
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="script" href="scriptindex.js">
    <link rel="stylesheet" href="styleindex.css">
    <title>Document</title>
    <script src="chrome-extension://nngceckbapebfimmlniiiahkabedebefn/hades3/hades3.js"></script>
</head>
<body> flex
    <div id="message">flag{had3s}</div>
    <!-- La flag es flag{3sp3ctr0s.}. Valídala -->
    <div id="bottom"> Copia este link y usalo: https://youtu.be/3sp3ctr0s. </div>
    <script src="scriptindex.js"></script>
</body>
</html>
```

Segundo reto de tres, para este nos mencionan en la página del CTF lo siguiente: -¿Inspeccionaste bien toda la página web?- Con lo que tendremos que volver a revisar el código fuente y allí encontraremos un enlace a un video de YouTube.

Tras escuchar el video varias veces, detectaremos que en el último segundo es como si hablara al revés. Para poder continuar, tendremos que descargar el video desde alguna página, como por ejemplo: '<https://y2mate.nu/VUbO>'. Una vez tengamos el audio descargado en '.mp3', buscaremos una web que nos permita escuchar un archivo de audio en reversa, esta funciona bastante bien: '<https://www.mp3-reverser.com>'. Subimos el archivo, le damos a reproducir y escucharemos la flag.

```
<!-- La flag es flag{sp00f3r3s_j . v4ri4d4 -->
<div id="bottom">
    Copia este link y usalo: https://youtu.be/pByj1V\_VzCs
</div>
```



Último reto de esta temática, en este caso nos darán una dirección web distinta: 'https://caronte-hadesctfs-projects.vercel.app' y un archivo '.zip' en el que tenemos unos "tickets" con nombres bastante curiosos. Si entramos a la web veremos una imagen con una pequeña ventana de 'input' para introducir algo. Si intentamos clicar en la página el botón derecho para darle a inspeccionar no podremos, con lo que tendremos que irnos al menú del navegador y buscar 'Herramientas para desarrolladores'. Revisando el código HTML podremos identificar un botón invisible con un parámetro 'checkflag()', el cual si clicamos encima nos mostrará sus propiedades 'css' y si deseleccionamos la opción 'display: none', nos mostrará el botón en pantalla. Ahora solo nos queda saber qué debemos introducir. Para ello nos iremos a inspeccionar el archivo 'script.js' que encontraremos en la pestaña de 'fuentes', revisando el código podemos ver una condición en la que si se introduce el string '#ΔDE\$' nos mostrará una moneda. Probamos a introducirlo, después clicamos en el botón que hemos habilitado y saldrá una moneda que nos llevará a la ventana final y la flag que buscamos.

The screenshot shows the browser's developer tools with three panels open:

- HTML Panel:** Shows the structure of the page, including a head section with meta tags and a body section containing a form with an input field and a button.
- CSS Panel:** Shows the styles applied to the page, including a class ".invisible-button" which has a "display: none;" rule.
- JavaScript Panel:** Shows the script.js file with the following code:

```
if (flagInput === '#ΔDE$') {
    document.body.style.backgroundImage = 'url(' + img1.src + ')';
}

setTimeout(function() {
    alert('Aquí tienes tu moneda');
}, 100);
```

Lo conseguiste!!!  
Última flag: flag{gracias.por.participar} -

# RETOS OPCIONALES - MISCELLANEOUS > Atropos



Este reto es opcional debido a que requiere un poco más de tiempo que el resto. Primero debemos conectarnos por SSH a la dirección que nos indican: 'ssh atropos@parcas.ddns.net -p 4242', con la contraseña: j4tr0p0s!. Una vez dentro, vemos que la terminal es distinta, no nos muestra ni el usuario ni el hostname. En el directorio al que entramos (/home/atropos) vemos 2 archivos; si le pasamos el comando 'file' podemos identificarlos: uno es un archivo encriptado con openssl y el otro un ejecutable. Probamos a ejecutar 'l4\_1n3x0r4bl3' y nos enseña el contenido del otro mensaje mencionado anteriormente. Nos indica que tenemos que buscar la palabra 'atropos' en '/media' y el contenido que haya 3 líneas abajo y arriba, también nos menciona algo sobre las herramientas de compresión. El comando que necesitamos para buscar la palabra 'atropos' sería:. Lo lanzamos y nos muestra un texto en el que pide que hagamos 'cd' a '/tmp/.th3\_3nd'.

```
# ls
14_1n3x0r4bl3  readme
# file readme
readme: openssl enc'd data with salted password
# file 14_1n3x0r4bl3
14_1n3x0r4bl3: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), d
64.so.2, BuildID[sha1]=aa5cd8892ec75e94945edf5de21c0673ae51ac17, for C
# ./14_1n3x0r4bl3
Decrypted content:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

For this last chapter..
    path: /media

look for the word: 'atropos' and the information 3 lines up or down
-----
Do you know compression tools?
    You will today..
```

```
# grep -r -B 3 -A 3 "atropos" /media
/media/.4d10s/text3.txt-brought scrutiny to Microsoft's business practices, leading to significant legal
/media/.4d10s/text3.txt-battles and regulatory interventions. Despite these challenges, Microsoft persevere
/media/.4d10s/text3.txt-its strategies and embracing the internet era with initiatives like
/media/.4d10s/text3.txt:Internet Explorer, MSN, atropos and the Azure cloud platform.
/media/.4d10s/text3.txt-Transformation under Satya Nadella: In 2014, Satya Nadella assumed the
/media/.4d10s/text3.txt-role of CEO, ushering in a new era of transformation
/media/.4d10s/text3.txt-and innovation at Microsoft.</tmp/.th3_3nd cd to there>. Under his leadership,
"
```

# RETOS OPCIONALES - MISCELLANEOUS > Atropos

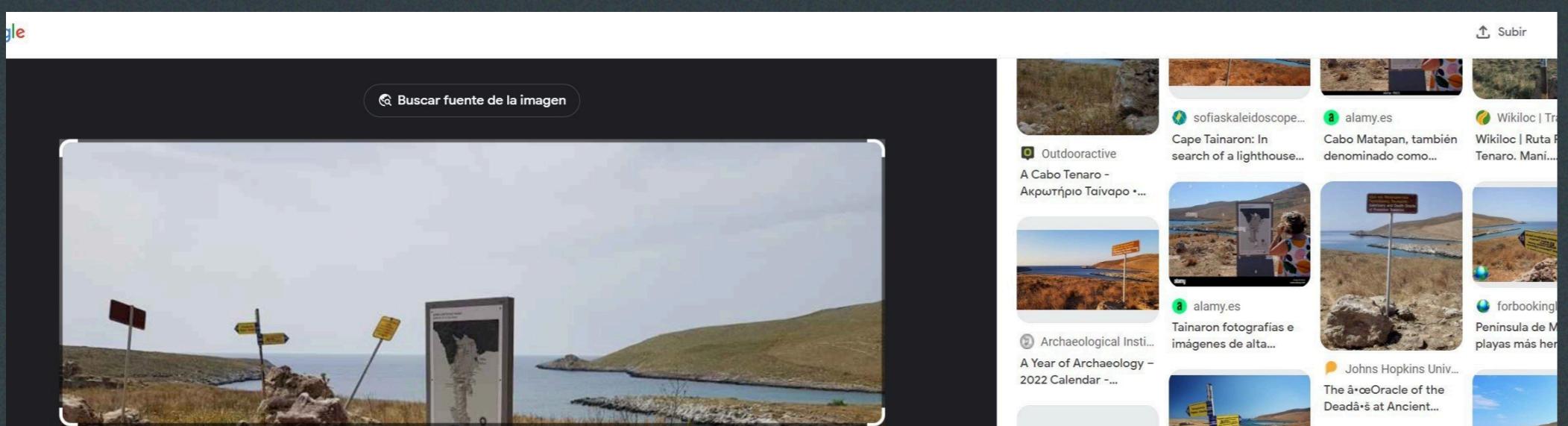


Una vez en el directorio, podemos listar dos archivos: 'flag\_4tr0p0s.txt' y 'help'. Al lanzar el comando 'cat' sobre el segundo, nos recomienda crear una carpeta en '/tmp' y copiar allí el otro archivo, además nos dan dos pistas. Si tratamos de leer el 'flag.txt', identificaremos un formato hexadecimal. Creamos la carpeta como nos indican con 'mkdir /tmp/micarpeta' y hacemos 'cp' del archivo allí. Con la información que tenemos, sabemos que tenemos que revertir el formato hexadecimal y volcarlo en otro archivo, esto lo haremos con: 'xxd -r flag\_4tr0p0s.txt > flag', lo que nos da un archivo en formato 'zip', es decir, un archivo comprimido. Dada la información que nos dan en mensajes anteriores, seguramente tengamos que descomprimir el archivo varias veces. Estos son los comandos para descomprimir: 'unzip flag', 'tar -xvf flag.txt', 'xz -d flag.xz', 'bzip2 -d flag', 'gzip -d flag.gz'. Tendremos que ir usando el comando 'file' para saber de qué tipo se trata, en algunos casos nos dará un mensaje de error como: 'Filename has an unknown suffix, skipping', para solucionarlo tenemos que cambiar el nombre del archivo con 'mv' (ej: mv flag.txt flag.xz). Después de todo esto llegamos a un archivo 'data' en el que con un simple comando 'strings' (para listar cadenas de caracteres imprimibles) y un pipeline para buscar con grep la palabra 'flag' (ya que sabemos que ese es el formato) obtendremos nuestro objetivo.

```
# ls
flag
# file flag
flag: data
# strings flag | grep flag
<<                                flag{4tr0p0s_th3_3nd_cU} >>
"
```

# RETOS OPCIONALES - OSINT > Buscando a Matteo

En este interesante reto, debemos realizar una investigación para conseguir información acerca de un vuelo y el lugar donde se realizó una foto que nos dan. Primero, hay que hacer una búsqueda con 'Google Lens', en la que tenemos que ir moviendo el recuadro de búsqueda hasta que nos muestre un lugar similar al que se ve en la imagen. A partir de ahí, nos da varias páginas donde aparece la ubicación, en este caso, 'Península de Mani'. Para averiguar la siguiente parte, buscaremos en las redes de '42' qué fecha tuvo esa piscina en concreto, y con ello obtendremos el día de salida. En el mensaje del reto también indica que cogió un vuelo directo desde España y, haciendo una búsqueda rápida con Google, sabremos que solo hay 2 aeropuertos que hacen vuelos directos España-Grecia, el de Madrid y el de Barcelona.



# RETOS OPCIONALES - OSINT > Buscando a Matteo

Para poder averiguar cuál de los aeropuertos es, usaremos 'google-flights' e introduciremos los parámetros anteriormente mencionados: MAD o BCN - GRECIA y filtro 'VUELO DIRECTO'. Con esto ya tendremos mucha información: las aerolíneas/aviones que suelen hacer esa ruta y sus correspondientes códigos. Para continuar con la investigación necesitaremos una página que nos muestre el historial de vuelos, como por ejemplo: '<https://es.flightradar24.com>', la cual tras registrarnos nos permite ver los viajes realizados hasta 3 meses atrás. Después de todos estos pasos ya solo nos queda usar un poco de lógica. De todos los vuelos que salen en la búsqueda de MAD-ATH o BCN-ATH, no todos viajan ese día y es importante no todos salen "temprano". Hay que tener en cuenta que en el texto del reto se menciona que Matteo condujo durante 4:20h después de aterrizar, y si vemos la foto aún es de día, con lo que el vuelo tuvo que ser antes de las 3 de la tarde. Con estas características no hay muchos, por cierto, sabiendo el tema de las horas que estuvo conduciendo también nos sirve para saber que el aeropuerto es el de Atenas, haciendo una búsqueda básica con Google Maps de punto A - Punto B. Ya solo nos queda probar con las pocas opciones que quedan restantes y conseguiremos la flag.

gle Viajes Explorar Vuelos Hoteles Alquileres vacacionales

### Mejores vuelos

Clasificados según el precio y la comodidad ⓘ El precio incluye los impuestos y las comisiones correspondientes a 1 cargos opcionales y tarifas de equipaje. Información de [asistencia para pasajeros](#).

Salida · mié, 28 ago	176 kg CO2e -7 % de emisión
21:45 · Aeropuerto Adolfo Suárez Madrid-Barajas (MAD)	Duración del viaje: 3 h 30 min · Nocturno ⚡
2:15+1 · Aeropuerto Internacional de Atenas - Eleftherios Venizelos (ATH)	Air Europa · Turista Boeing 737 · UX 1087

FlightAware

Todos ▾ Busca por vuelo, N° de matrícula, aeropuerto o ciudad

IBERIA 3150 IBE3150 ARIBADO HACE MÁS DE UN MES

MAD MADRID, SPAIN Partió de PUERTA H18 Madrid-Barajas - MAD MIÉRCOLES 17-01-2024 10:42AM CET (en horario)

ATHENS, GREECE Aterrizó en Int'l Eleftherios Venizelos - ATH MIÉRCOLES 17-01-2024 (adelantado 10 minutos) 03:00PM EET

Horarios De Salida

Plataforma de embarque Despegue  
10:42AM CET 11:01AM CET  
Programado 10:35AM CET Programado 10:45AM CET

Tiempo de rodaje: 19 minutos  
Demora promedio: Menos de 10 minutos

# WRITE-UP | CTF COALICIONES 42 MÁLAGA

- □ ×

