

Actividad 5 - Seguridad en Redes

Profesor: René Guerrero Torres

Ayudante: Iván Zuñiga

Alumno: Reinaldo Pacheco Parra

1. Buscar ubicación a través de metadatos en imagen 1.jpg

Instrucciones: Indique toda la información que pueda entregar de la fotografía además de la geolocalización indicando la dirección exacta donde se sacó la fotografía , buscándola en google maps u otro aplicativo , etc.

- 1) Se descarga la fotografía 1.jpg y se arrastra en el software FOCA

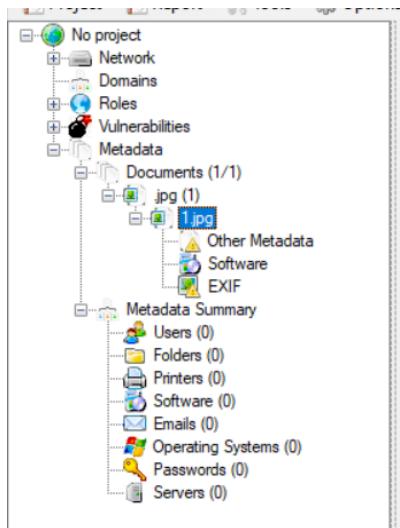
The screenshot shows the FOCA application window. At the top, there is a logo consisting of the letters 'FOCA' in a stylized font where each letter has a different color and shape. Below the logo is a search bar containing the query "site:site.com filetype:pdf". Underneath the search bar is a table with the following columns: Id, Type, URL, Download, Download Date, Size, Analyzed, and Modified Date. There is one row in the table:

Id	Type	URL	Download	Download Date	Size	Analyzed	Modified Date
0	jpg	C:\Users\Reinaldo\Downloads\1.jpg	•	11/17/2024 4:52:24...	2.78 MB	✗	-

- 2) Se selecciona la fotografía y se presiona en “Extract All Metadata” para obtener todos los metadatos

The screenshot shows the FOCA application window with the same search results table as before. A context menu is open over the first row of the table, specifically over the file "1.jpg". The menu options are: Download, Download All, Delete, Delete All, Extract Metadata, Extract All Metadata (which is highlighted with a blue selection bar), Analyze Metadata, Add file, Add folder, Add URLs from file, and Link.

- 3) Se visualiza que se obtuvieron los metadatos, para revisarlos, se presiona en EXIF para ver el archivo de metadatos



4) En el archivo EXIF se logran ver datos relevantes:

- El dispositivo con el que se tomó la fotografía fue un Apple iPhone SE de 2da generación.
- Se pueden ver las configuraciones de la cámara y resolución
- El día en que se tomó la fotografía fue el 29 de Octubre del 2021 a las 01:12:12 de la madrugada.

Exif Makernote	
Make	Apple
Model	iPhone SE (2nd generation)
Orientation	Right side, top (Rotate 90 CW)
X Resolution	72 dots per inches
Y Resolution	72 dots per inches
Resolution Unit	Inches
Software	14.7.1
Date/Time	2021:10:29 01:12:12
Tile Width	512
Tile Length	512
YCbCr Positioning	Center of pixel array
Exposure Time	1/15 sec
F-Number	F 1.8
Exposure Program	Program normal
ISO Speed Ratings	800
Exif Version	2.32
Date/Time Original	2021:10:29 01:12:12
Date/Time Digitized	2021:10:29 01:12:12

5) En la sección de GPS Exif Makernote se pueden visualizar datos de geolocalización los cuales son relevantes para detectar la ubicación en donde fue tomada la fotografía.

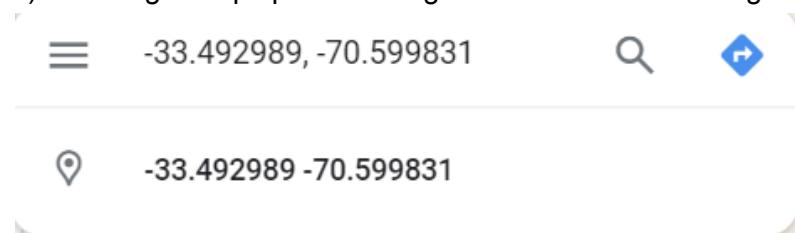
- GPS Latitude Ref: S
- GPS Latitude: 33°29'34.76
- GPS Longitude Ref: W
- GPS Longitude: 70°35'59.39

Las coordenadas corresponden a (33°29'34.76"S, 70°35'59.39"W)

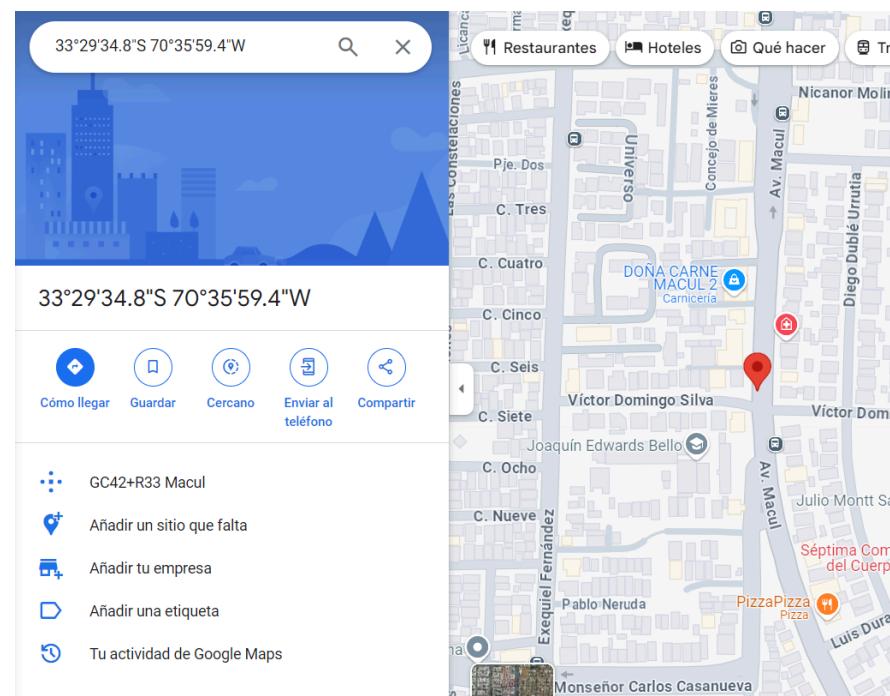
GPS Makernote

GPS Latitude Ref	S
GPS Latitude	33°29'34.76
GPS Longitude Ref	W
GPS Longitude	70°35'59.39
GPS Altitude Ref	Sea level
GPS Altitude	925307/1623 metres
GPS Speed Ref	kph
GPS Speed	9089/154206
GPS Img Direction Ref	True direction
GPS Img Direction	192858/1049 degrees
GPS Dest Bearing Ref	True direction
GPS Dest Bearing	192858/1049 degrees

6) En Google Maps podemos ingresar la información de geolocalización obtenida



7) La ubicación corresponde a la intersección entre la Av. Macul y Víctor Domingo Silva en la comuna de Macul



8) Luego en Google Street View se puede visualizar que la foto fue sacada en el panel de publicidad del paradero PD70-Avenida Macul / Esquina Víctor Diego Silva por lo cual la geolocalización es correcta.



Imagen original

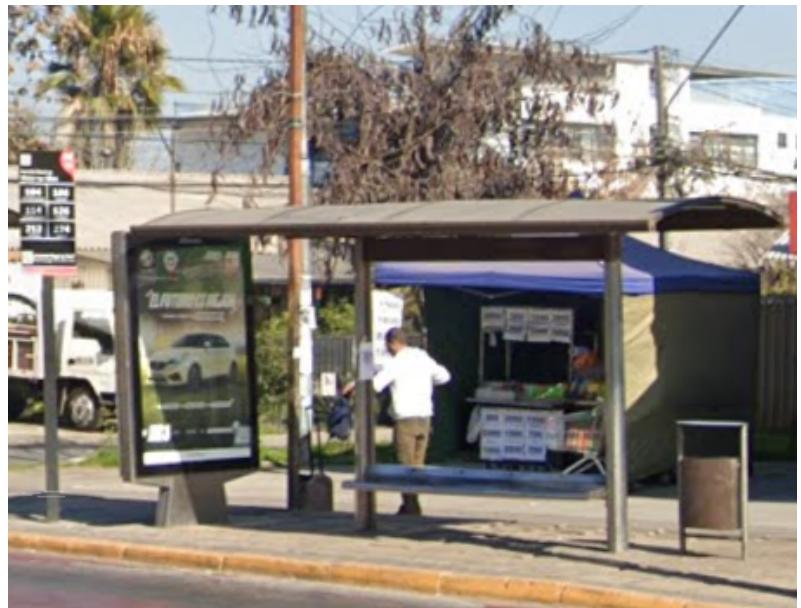
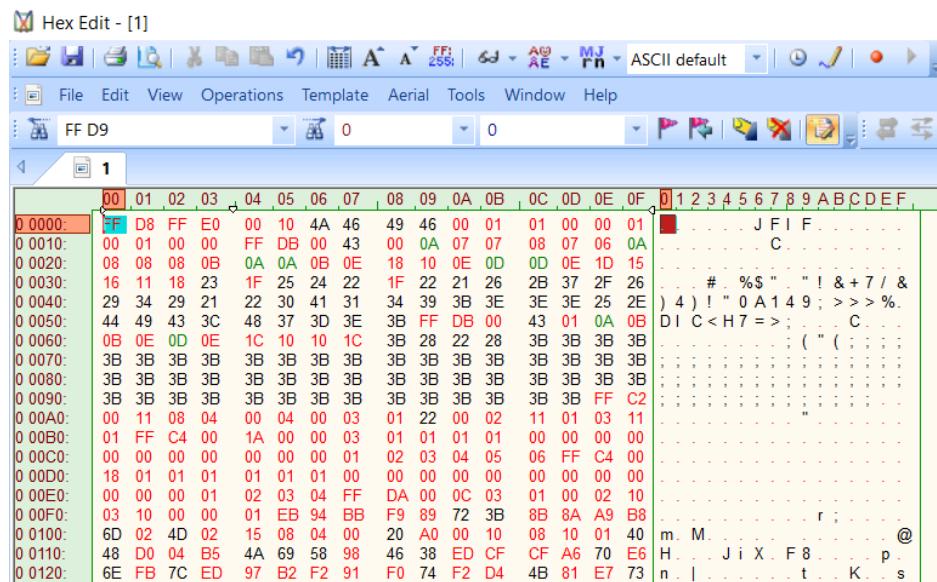
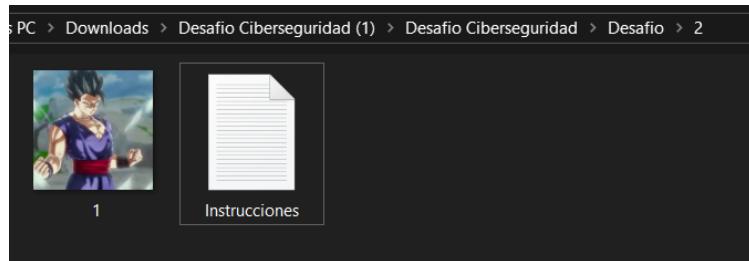


Imagen Google Street View

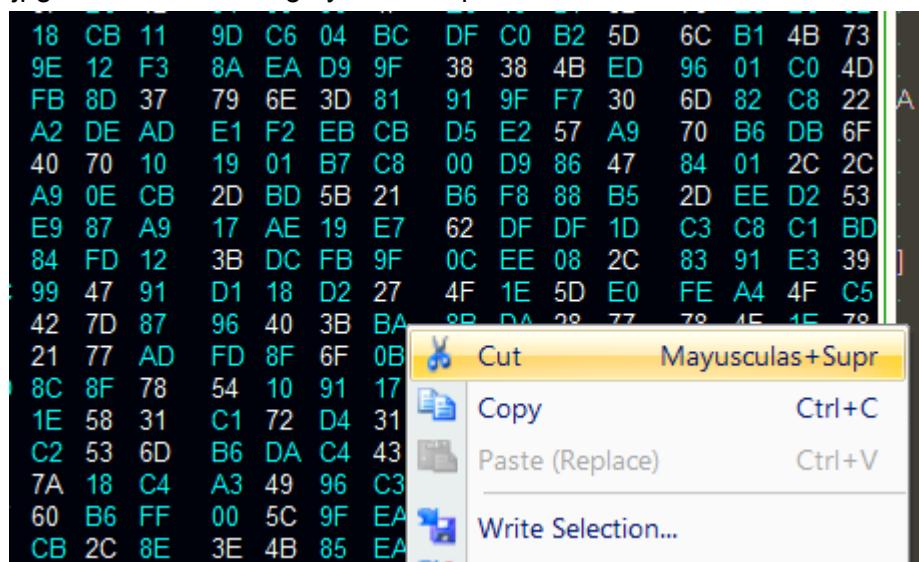
2. Extraer los archivos dentro de la fotografía 1.jpg

Instrucciones: Extraiga todos los archivos de la fotografía y documente todos los hallazgos encontrados.

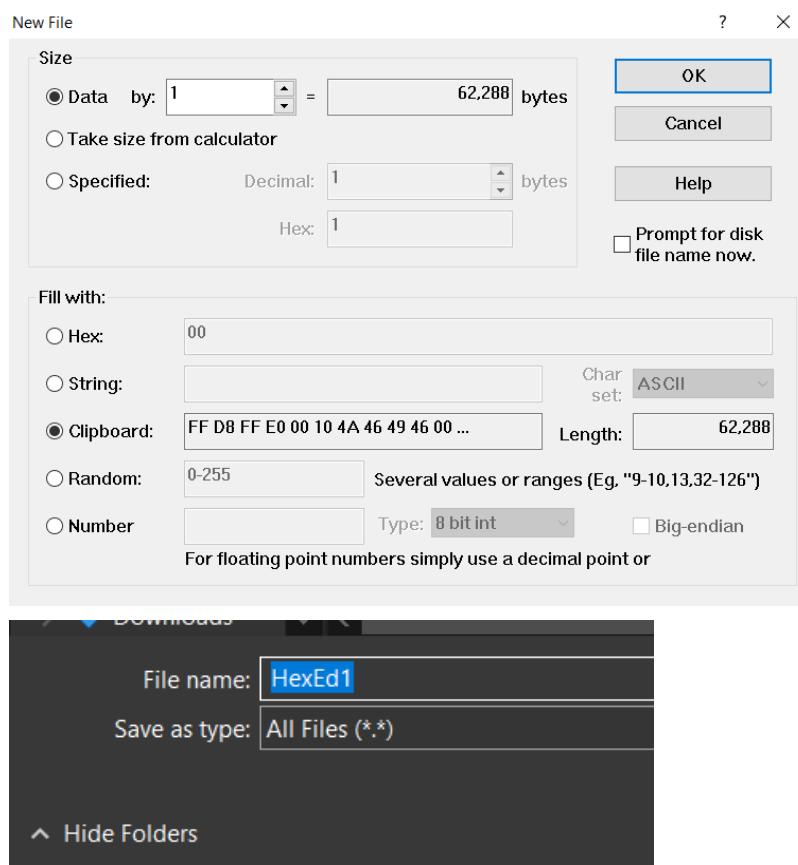
- 1) Se descarga la fotografía 1.jpg y se abre en la aplicación Hex Edit



2) Se ubica la cabecera de FF D8 y el fin FF D9 que corresponde a un archivo con formato jpg, se obtiene el rango y se corta para obtener el archivo.



3) Se guarda el rango obtenido entre la cabecera y el fin del archivo en un archivo con nombre “HexEd1.jpg”



4) En la aplicación PhotoFiltre11 se visualiza la imagen recortada, la cual corresponde a la imagen inicial.

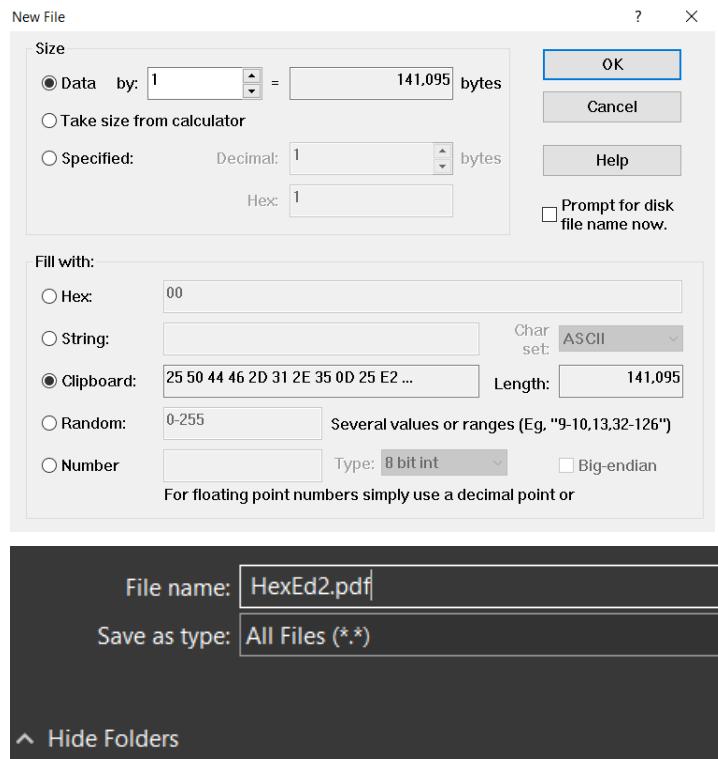


5) Luego de recortar la imagen inicial, se logra identificar como cabecera 25 50 44 46 2D que corresponde a un archivo en formato PDF

6) Se busca el final del archivo, el cual corresponde a 25 25 45 4F 46 y se copia el rango entre la cabecera y el fin del archivo para crear un nuevo archivo.

The screenshot shows a hex editor interface with a toolbar at the top and a status bar at the bottom. The main window displays a memory dump with rows labeled 2.2710 through 2.2780. A context menu is open over the row 2.2720, listing options like Cut, Copy, Paste (Replace), Write Selection..., Append Selection..., Append Same File, Calculate Using..., Find Selection, Highlight, Customize..., Options..., and Properties... . The status bar at the bottom shows file numbers 80, 24, 24, 89, S, H1, %, H1, \$.

7) Se crea un nuevo archivo con la información copiada agregando la extensión del tipo de archivo, en ese caso .pdf. Se asigna el nombre HexEd2.pdf y se guarda



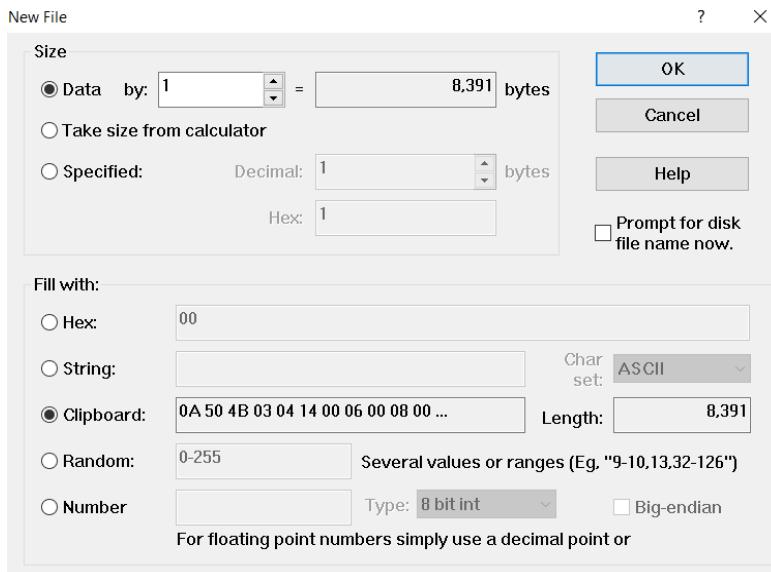
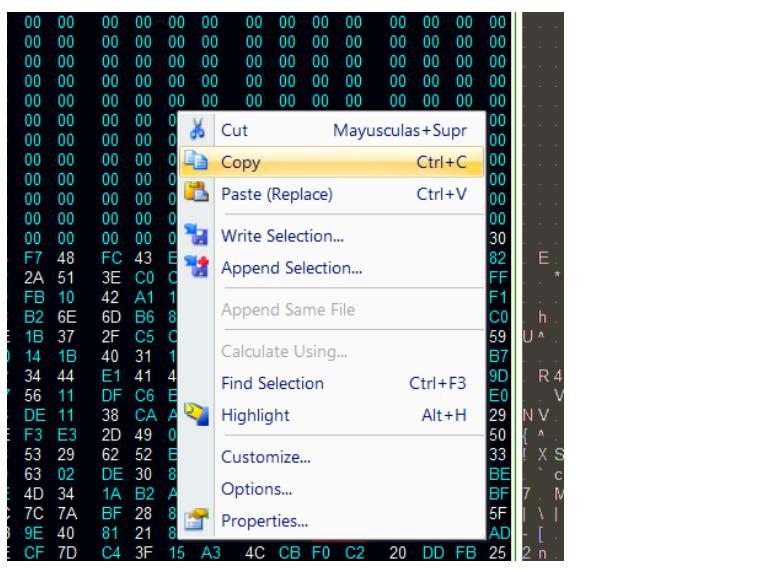
8) Se abre el archivo obtenido el cual corresponde a un pdf acerca del Convenio de Budapest

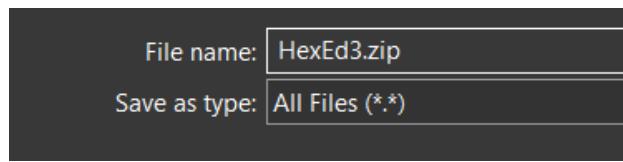
The image shows a Microsoft Edge browser window displaying a PDF document. The header of the PDF shows 'Today (9)' and the file details 'HexEd2' (PDF), '11/23/2024 2:42 AM', 'Microsoft Edge PD...', and '138 KB'. The PDF content is titled 'Convenio sobre la Ciberdelincuencia: Convenio de Budapest' and is dated 'Julio 2018'. It features a red header with the logo of the Biblioteca del Congreso Nacional de Chile / BNCN | Asesoría Técnica Parlamentaria. The document includes sections for 'Autor' (Verónica Barrios Achavar, email: vbarrios@bcn.cl, phone: (56 2) 22701884) and 'Resumen' (summary in Spanish). The summary discusses the Convention on Cybercrime (Budapest Convention) as an international agreement to combat cybercrime, its ratification by 60 states, and its incorporation into Chilean law. It highlights the Convention's objective of harmonizing legislation against cybercrimes and establishing effective cooperation and international assistance.

9) Ahora queda como cabecera un archivo con extensión PK por lo cual se toma desde la cabecera y se busca el fin del archivo

	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
0000:	0A	50	4B	03	04	14	00	06	00	08	00	00	00	21	00	62	P	K	!	.	b					
0020:	EE	9D	68	5E	01	00	00	90	04	00	00	13	00	08	02	5B	.	h	^	[.	.						
0030:	43	6F	6E	74	65	6E	74	5F	54	79	70	65	73	5D	2E	78	C	o	n	t	e	n	t	_	T	y	p	e	s]	.	x
0040:	6D	6C	20	A2	04	02	28	A0	00	02	00	00	00	00	00	00	m	l	(.
0050:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
0060:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
0070:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00			
0080:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00			
0090:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00			
00A0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00			
00B0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00			
00C0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00			
00D0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00			
00E0:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00			

10) Se identifica el fin del archivo y se copia el rango entre el inicio y fin para crear un nuevo archivo el cual se guarda como "HexEd3" con la extensión .zip la cual corresponde al código PK en ASCII





11) Luego, de guardar el archivo, se extrae y se genera una carpeta que contiene archivos en formato .xml

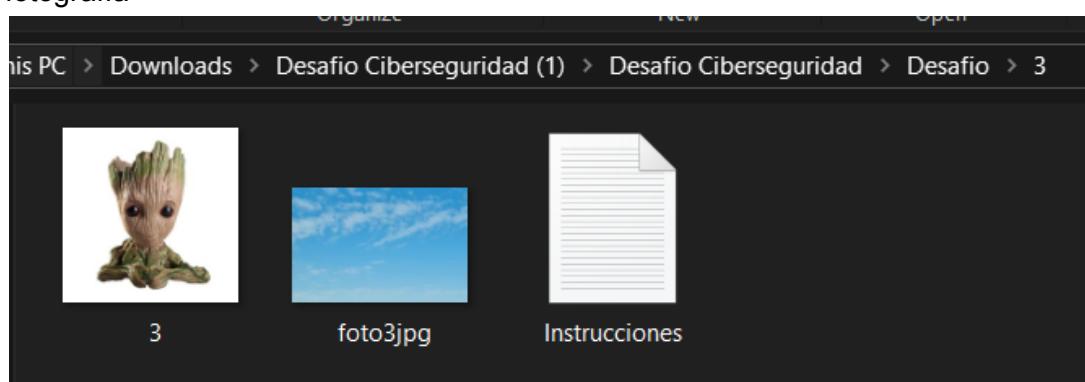
Name	Date modified	Type	Size
HexEd3	11/23/2024 2:59 AM	WinRAR ZIP archive	9 KB
HexEd3	11/23/2024 2:59 AM	File folder	

Name	Date modified	Type	Size
_rels	11/23/2024 2:59 AM	File folder	
docProps	11/23/2024 2:59 AM	File folder	
xl	11/23/2024 2:59 AM	File folder	
[Content_Types]		XML Document	2 KB

3. Reconstrucción de una foto con FOCA

Instrucciones: Reconstruya la foto y documente los hallazgos encontrados

1) Se descarga la foto, en este caso se utilizará foto3.jpg, ya que si se abre con la opción de propiedades, se puede identificar que contiene metadatos para la reconstrucción de la fotografía



2) Se abre el software FOCA y se agrega la imagen foto3.jpg

A screenshot of the FOCA application interface. At the top, there is a search bar with the query "site:site.com filetype:pdf". Below the search bar is a table with columns: Id, Type, URL, Download, Download Date, Size, and Analyzed. There is one entry in the table:

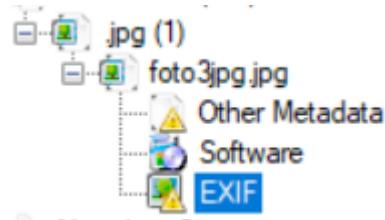
Id	Type	URL	Download	Download Date	Size	Analyzed
0	jpg	C:\Users\Reinaldo\Downloads\Desafio Ciberseguridad (...)	.	11/23/2024 3:06:55...	164.74 KB	x

The URL column shows a truncated path: "C:\Users\Reinaldo\Downloads\Desafio Ciberseguridad (1)\Desafio Ciberseguridad\Desafio\3\foto3jpg.jpg".

3) Se selecciona y se presiona la opción de “Extract All Metadata” para obtener los metadatos de la imagen

A screenshot of the FOCA application interface showing a context menu for the selected file. The menu options are: Download, Download All, Delete, Delete All, Extract Metadata, Extract All Metadata (which is highlighted), Analyze Metadata, and Add file.

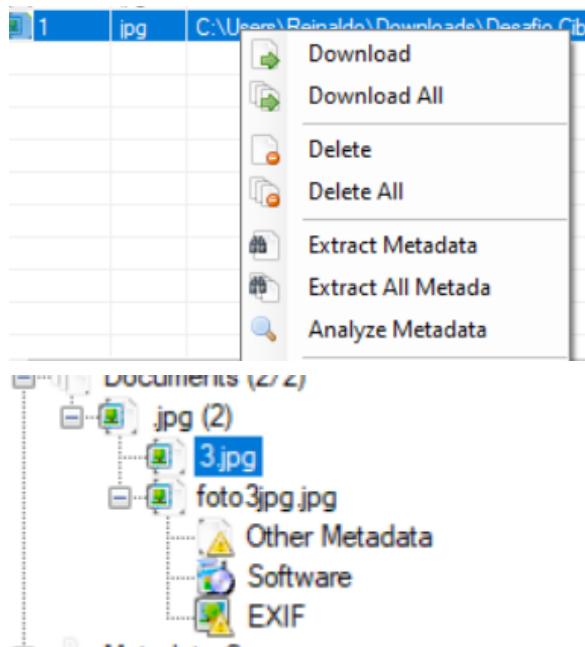
4) Luego, se selecciona la opción EXIF para visualizar los metadatos con mayor detalle



5) En la sección Thumbnail, se obtiene la imagen reconstruida

A screenshot of the EXIF viewer showing the reconstructed thumbnail image. The thumbnail is labeled "Thumbnail" and "Picture". The image itself is a photograph of a coastal scene with a blue sky, white clouds, and a road or railway track leading towards the horizon.

6) La imagen 3.jpg que tiene a Groot se ingresó a la aplicación FOCA, pero no contiene metadatos por lo que no se puede reconstruir



4. Cifrado simétrico y asimétrico de nombre y apellido

Instrucciones: Demuestre cómo cifrar y descifrar con un protocolo de cifrado simétrico y asimétrico , documente el proceso.

Probar con su nombre y apellido.

4.1 Cifrado simétrico con el algoritmo AES

- 1) Se inicia la máquina Kali con la interfaz de red en modo NAT
- 2) Luego, en la terminal se configura el teclado en el formato Latinoamericano con el comando:

```
setxkbmap -layout latam
```

```
(kali㉿kali)-[~]
$ setxkbmap -layout latam
               me
(kali㉿kali)-[~]
$ █
```

- 3) Se crea un directorio de trabajo con el comando mkdir (make-directory) y cd (change-directory) para acceder a él con los comandos:

```
mkdir cifrado
cd cifrado
```

```

└──(kali㉿kali)-[~]
$ mkdir cifrado

└──(kali㉿kali)-[~]
$ cd cifrado

└──(kali㉿kali)-[~/cifrado]
$ ls
private.key

```

4) Se genera una llave privada de 2048 bits con el comando:

`openssl genrsa -out private.key 2048`

```

└──(kali㉿kali)-[~/cifrado]
$ ls -l
total 4
-rw——— 1 kali kali 1704 Nov 23 11:39 private.key

└──(kali㉿kali)-[~/cifrado]    algoritmo_dh
$ ls

```

5) Se visualiza el archivo `private.key` con el comando `cat`

```

└──(kali㉿kali)-[~/cifrado]
$ cat private.key
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwgSjAgEAAoIBAQCpxuM7ELZJ9Vlb
bvLZ6zLXfdQGrIZ6OwHiXEM6SC/AbZD0QRW09MmenVNxm3r3mc0uoAZClH1SHEs5
1Rh86qJLwJGAoajEg/ALZDYkezTx+MHln/vB5Hwz6f+csJEbAxV+ypyn7TUmlWEWA
PVywMqngok8Lgw/tzn8FCo38HHGLsLT84QcPDEzPRKG3RGJoVI78/xmwGxXjMSQb
YPbxTfc7TE4iUTEfok08DtX3oqQW8EXKXmd+Sdtc8S0LnU0/Po+xbDWSVcPY97w
X8vA8dJT6Zalwg/bykd2b1mcn30SoXlrLALc154x0IFxGY0zBx7ew8iwl/A69QVG
SenG4l2hAgMBAAECggAE+dwpCYuePxotKLskteueDTRg0H9GCNR1Ulvzl4pQiz
YSTC6KdPfpGsl2RxCteIwiTvVvLtq6i227WImhcv+82RwWYye7I0JuxoDz2FPDER
Gl334rK/f56ZYcIEnl8q2qM1BoL1e0S0zo/LGWUh1ryVM7ZOhxoZziTUoeVRHTKN
+8jgYKTErB24lMuF7qU+TbZwtXGb18wYWVWzYMDrcQia0N1bazcQMFxVzzG1N9wZU
LmDqn8WsQtLSAs1QF/oAesnRDVYUxg5lsNqjFF0JtnhqcZussm3IVLpb9E7vxAI8
aE/LDwemWYB6KKkZ0Zdr0Q00EvcXgonYZM0wGdPK0QKBgQDfIu7/WG9qEz9eKDUm
MHPTB9XmtEoW4dG6kEl5R6PEwILWBxiVG75RZCpmLew2yXitghkocIpCd2xNvzv
5iuTAiRtchhqlPiQAKUP7QIBLXhHbo/y0kIlvt1nCJ8BC/U1IT7MmbJQPDBoUrM
ullDEB+2htwlFigm/ml1U2sTbwKBgQDCyBoVYPeImSH7Y9lj0HYauqdf8vzHvpUN
IfyqrqZpSY2xIyR8Jl4zQXt+wEiVIUMcN6bjlNi4ySE0IvbolWvrVz17gCl3kUXpm
LGzs3LTzfCgcWRaVLJvPCafjx53apD7Z4WF53rQPE8AbQCwH6qEFcftuJ00ycDX
w2J7EHfX7wKBgDOrmzx19hbSLjJXjTcysR+FCptLdHyn/jD9iIYVGM905n6Mu2Dp
vgjsuvsK6j6CANvm2zqaRWPLbRLEcP38JnX8YQMXEKbdrtp1M9DStuK8/dwmu7L
FrZuf/xUE4g0T/Bw1zp0CwlBK8uImwWGWQil7XmtN3Bqj1YJJSpDxAoGBAJTJ
4ia6/bGDzk+IMHFstiso1d4xTycSPNl00uwfo/efsMCzzC2kZxCPMD0u4IIImXg
WZrNzNAehqnctCr/nMFTW0fs41gbXVjbMUGmPpuYBdPRwcxl7Qc0Kwl2XIC94jzx
TVijRgkRCJAc9/V8J1BPh27VylMzICFAd9b7xAFaOGAH4bPoeKJjzru6TM9J2Z+
UHUL8Y7xSjbDiOLLv8PI/ewaxbf1u1eX06N5uydDHIsaHTexU/SPci8otoceFH3q
YKQ5mlx83all0Z7Gly0gkrZFU7K27psua7t7Et7NE97PerVRMkSfx5cPZupnX5p0
B/vB2DK4XQhy8tJ7SfvHX4Y=
-----END PRIVATE KEY-----

```

6) Se crea un archivo de texto con un mensaje en el editor nano con el comando:

`nano mensaje.txt`

```

└──(kali㉿kali)-[~/cifrado]
$ nano mensaje.txt

```

```
File Actions Edit View Help  
GNU nano 8.1  
Reinaldo Pacheco
```

7) Se realiza el cifrado del archivo usando la llave privada creada con el algoritmo DES con el comando:

```
openssl enc -in mensaje.txt -out mensaje.enc -e -des-cbc -k private.key -pbkdf2
```

```
[(kali㉿kali)-[~/cifrado]]  
└─$ openssl enc -in mensaje.txt -out mensaje.enc -e -des-cbc -k private.key -pbkdf2  
[(kali㉿kali)-[~/cifrado]]  
└─$ ls -l  
total 12  
-rw-rw-r-- 1 kali kali 40 Nov 23 11:49 mensaje.enc  
-rw-rw-r-- 1 kali kali 17 Nov 23 11:41 mensaje.txt  
-rw----- 1 kali kali 1704 Nov 23 11:39 private.key
```

8) Se visualiza el contenido del archivo cifrado mensaje.enc con el comando cat mensaje.enc

```
[(kali㉿kali)-[~/cifrado]]  
└─$ cat mensaje.enc  
Salted__♦!F♦d♦d♦4#U♦|= 6♦B♦u♦!K  
[(kali㉿kali)-[~/cifrado]]  
└─$ █
```

9) Para descifrar el archivo, se usa el comando:

```
openssl enc -in mensaje.enc -out mensaje2.txt -d -des-cbc -k private.key -pbkdf2
```

```
[(kali㉿kali)-[~/cifrado]]  
└─$ openssl enc -in mensaje.enc -out mensaje2.txt -d -des-cbc -k private.key -pbkdf2  
[(kali㉿kali)-[~/cifrado]]  
└─$ ls -l  
total 16  
-rw-rw-r-- 1 kali kali 17 Nov 23 11:52 mensaje2.txt  
-rw-rw-r-- 1 kali kali 40 Nov 23 11:49 mensaje.enc  
-rw-rw-r-- 1 kali kali 17 Nov 23 11:41 mensaje.txt  
-rw----- 1 kali kali 1704 Nov 23 11:39 private.key
```

10) Se visualiza el contenido del archivo generado con el comando cat cat mensaje2.txt

```
[(kali㉿kali)-[~/cifrado]]  
└─$ cat mensaje2.txt  
Reinaldo Pacheco  
[(kali㉿kali)-[~/cifrado]]  
└─$ █
```

4.2 Cifrado asimétrico con el algoritmo RSA

- 1) Se crea y accede una nueva carpeta con el comando mkdir (make-directory) y cd (change-directory) para acceder. Se usan los comandos:

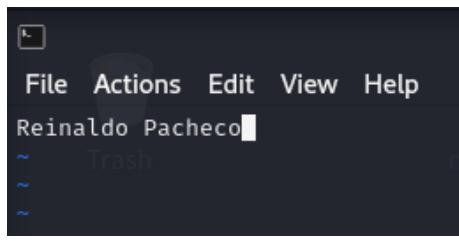
```
mkdir rsa
```

```
cd rsa
```

```
(kali㉿kali)-[~]
$ mkdir rsa
(kali㉿kali)-[~]
$ cd rsa
(kali㉿kali)-[~/rsa]
$
```

- 2) Se crea un archivo de texto que contiene el nombre y apellido con el comando:

```
vi mensaje.txt
```



```
(kali㉿kali)-[~/rsa]
$ vi mensaje.txt
(kali㉿kali)-[~/rsa]
$
```

- 3) Se genera una llave privada usando RSA con el comando

```
openssl genrsa -out private.key 2048
```

Se visualiza con el comando

```
ls -l
```

```
(kali㉿kali)-[~/rsa]
$ openssl genrsa -out private.key 2048
(kali㉿kali)-[~/rsa]
$ ls -l
total 8
  -rw-rw-r-- 1 kali kali 17 Nov 23 12:07 mensaje.txt
  -rw----- 1 kali kali 1700 Nov 23 12:09 private.key
(kali㉿kali)-[~/rsa]
$
```

- 4) Se genera una llave pública para “public.key” con el comando:

```
openssl rsa -in private.key -pubout -out public.key
```

Se visualiza con

```
ls -l
```

```
[(kali㉿kali)-~/rsa]$ openssl rsa -in private.key -pubout -out public.key
writing RSA key
[(kali㉿kali)-~/rsa]$ ls -l
total 12
-rw-rw-r-- 1 kali kali 17 Nov 23 12:07 mensaje.txt
-rw----- 1 kali kali 1700 Nov 23 12:09 private.key
-rw-rw-r-- 1 kali kali 451 Nov 23 12:10 public.key
[(kali㉿kali)-~/rsa]$
```

5) Se genera una firma digital para el usuario con el siguiente comando:

```
openssl dgst -sha1 -sign private.key -out signed.sha1 mensaje.txt
```

Se visualiza con

```
ls -l
```

```
[(kali㉿kali)-~/rsa]$ openssl dgst -sha1 -sign private.key -out signed.sha1 mensaje.txt
[(kali㉿kali)-~/rsa]$ ls -l
total 16
-rw-rw-r-- 1 kali kali 17 Nov 23 12:07 mensaje.txt
-rw----- 1 kali kali 1700 Nov 23 12:09 private.key
-rw-rw-r-- 1 kali kali 451 Nov 23 12:10 public.key
-rw-rw-r-- 1 kali kali 256 Nov 23 12:13 signed.sha1
```

6) Se valida la firma digital generada con el siguiente comando:

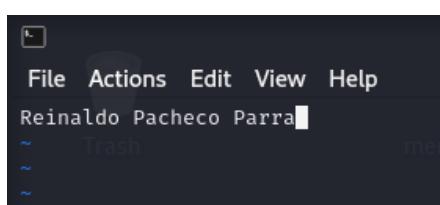
```
openssl dgst -sha1 -verify public.key -signature signed.sha1
mensaje.txt
```

```
[(kali㉿kali)-~/rsa]$ openssl dgst -sha1 -verify public.key -signature signed.sha1 mensaje.txt
Verified OK
```

7) Se realiza una modificación al documento firmado con el comando:

```
vi mensaje.txt
```

```
[(kali㉿kali)-~/rsa]$ vi mensaje.txt
```



Se visualiza la modificación realizada con el comando:

```
cat mensaje.txt
```

```
(kali㉿kali)-[~/rsa]
$ cat mensaje.txt
Reinaldo Pacheco Parra
```

8) Se realiza la validación de la firma digital con el comando:

```
openssl dgst -sha1 -verify public.key -signature signed.sha1
mensaje.txt
```

```
(kali㉿kali)-[~/rsa]
$ openssl dgst -sha1 -verify public.key -signature signed.sha1 mensaje.txt
Verification failure
40375DDB7F7F0000:error:02000068:rsa routines:ossl_rsa_verify:bad signature:../crypto/rsa/rsa_sign.c:426:
40375DDB7F7F0000:error:1C880004:Provider routines:rsa_verify:RSA lib:../providers/implementations/signature/rsa_sig
.c:801:
```

La verificación falla debido a que el mensaje inicial fue modificado.

5. Comparar 3 archivos diferentes a través de funciones hash

Instrucciones: Aplique y compare funciones hash (md5-sha) a los archivos y documente todos los hallazgos encontrados en cada archivo.

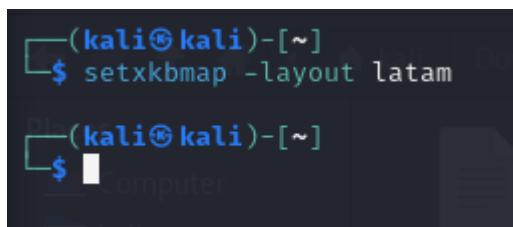
1) Se inicia la máquina Kali con la interfaz de red en modo NAT

2) Se descargan los 3 archivos sonic.jpg, sonic1.jpg y sonic2.jpg



3) Luego, en la terminal se configura el teclado en el formato Latinoamericano con el comando:

```
setxkbmap -layout latam
```



4) A través de la terminal, accedemos al directorio en donde se encuentran los archivos, en este caso es /home/kali/Downloads/Desafio Ciberseguridad/Desafio/5/ por lo que se usará el siguiente comando:

```
cd "/home/kali/Downloads/Desafio Ciberseguridad/Desafio/5"
```

Se visualizan los archivos con el comando:

```
ls -l
```

```
(kali㉿kali)-[~]
$ cd "/home/kali/Downloads/Desafio Ciberseguridad/Desafio/5"
Places
(kali㉿kali)-[~/Downloads/Desafio Ciberseguridad/Desafio/5]
$ ls -l
total 1060
-rw-rw-r-- 1 kali kali 120 Sep 29 2023 Instrucciones.txt
-rw-rw-r-- 1 kali kali 385217 Sep 29 2023 sonic1.jpg
-rw-rw-r-- 1 kali kali 449268 Sep 29 2023 sonic2.jpg
-rw-rw-r-- 1 kali kali 241302 Sep 29 2023 sonic.jpg

(kali㉿kali)-[~/Downloads/Desafio Ciberseguridad/Desafio/5]
$
```

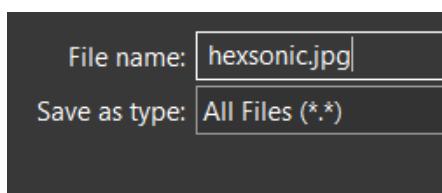
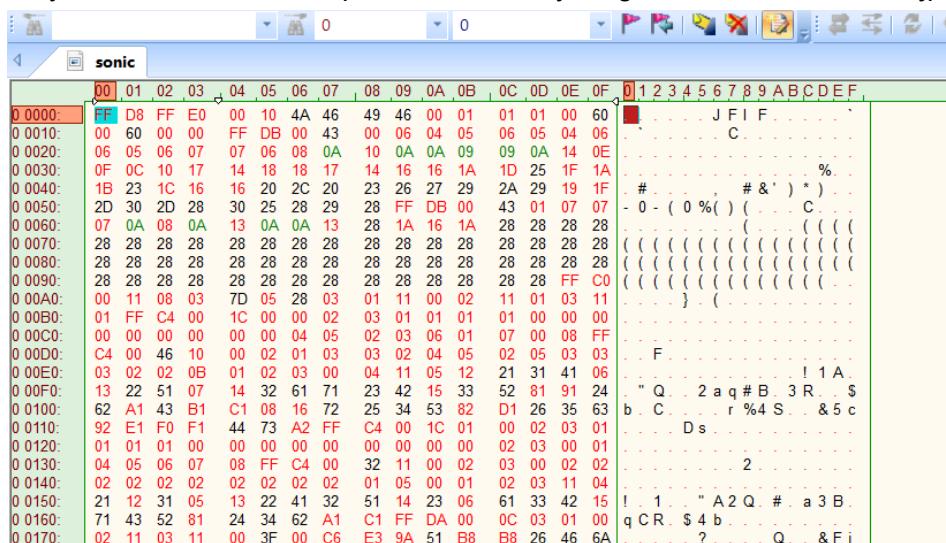
5) Ahora, para comparar los archivos a través de una función hash se aplica el siguiente comando:

`md5sum sonic1.jpg sonic2.jpg sonic.jpg`

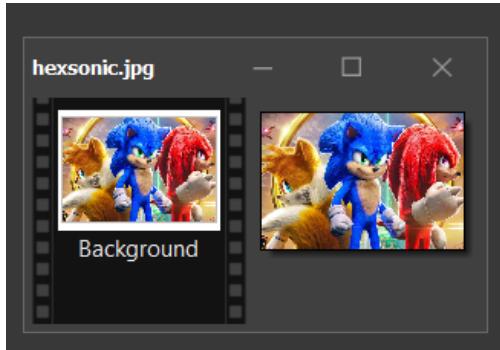
```
(kali㉿kali)-[~/Downloads/Desafio Ciberseguridad/Desafio/5]
$ md5sum sonic1.jpg sonic2.jpg sonic.jpg
752cdb4449be0a5731cd68dd9129d06f  sonic1.jpg
8eeb3c954608428f57e118849577c244  sonic2.jpg
620d254677d67f37feb1d89f00be7cc2  sonic.jpg
```

Se puede evidenciar que las 3 imágenes tienen hashes diferentes, por lo que los archivos son diferentes.

6) Usamos Hex Edit para visualizar la imagen sonic.jpg la cual contiene una cabecera FF D8 y un final FF D9, se copia el contenido y se guarda como hexsonic.jpg



7) Al abrir el archivo en PhotoFiltre11 se puede visualizar la imagen inicial

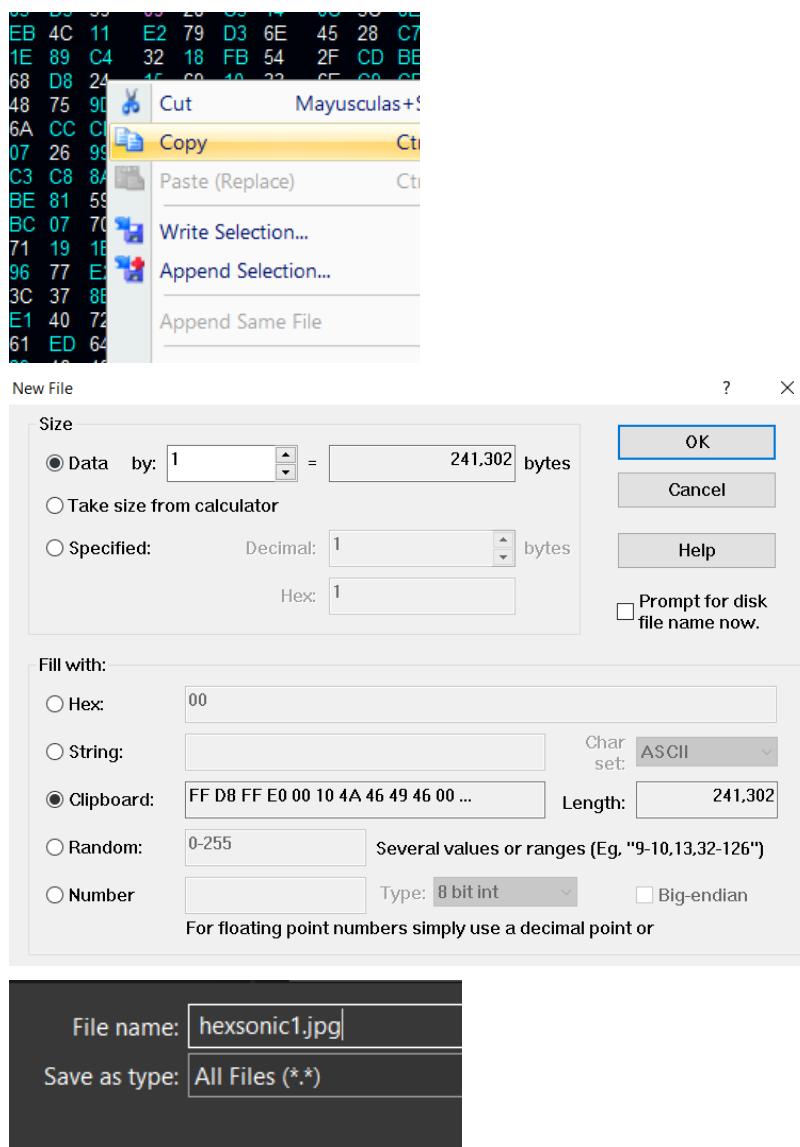


8) Se identifica que dentro de la imagen sonic.jpg solo existe la imagen extraída ya que el inicio y el final corresponde a una cabecera FF D8 y una cola FF D9, no hay más archivos adjuntados

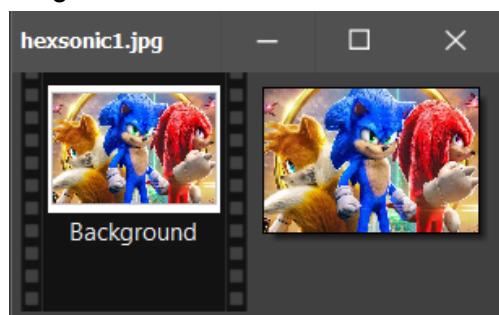
66	34	8B	63	A
85	6F	1E	11	9
1C	D5	B5	86	
C9	82	91	D8	5
61	94	10	00	
21	FF	D9		

9) Ahora se abre la imagen sonic1.jpg en Hex Edit la cual también tiene una cabecera FF D8 y una cola FF D9 que corresponde al inicio y final de un archivo en formato .jpg

10) Se copia el contenido entre la cabecera y la cola del archivo y se crea un nuevo archivo llamado hexsonic1 con la extensión .ipq que corresponde al tipo de archivo obtenido.

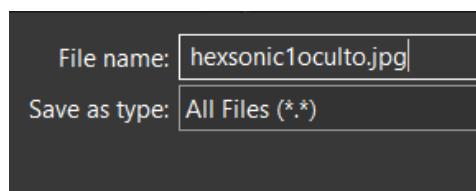
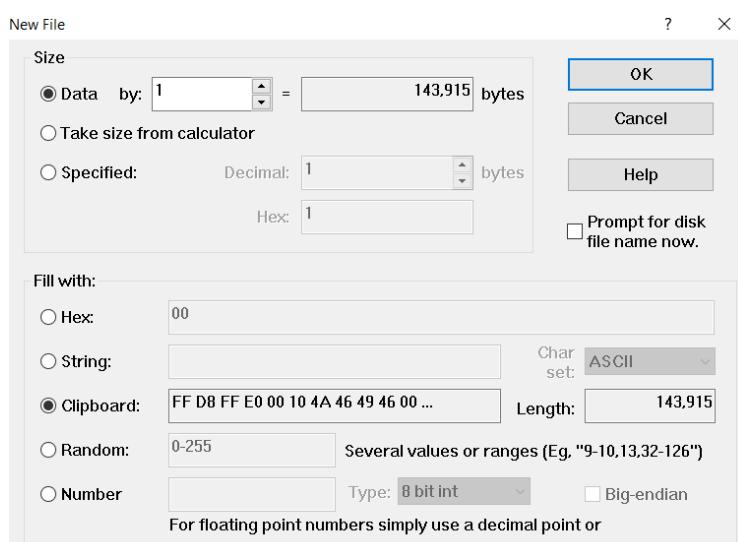
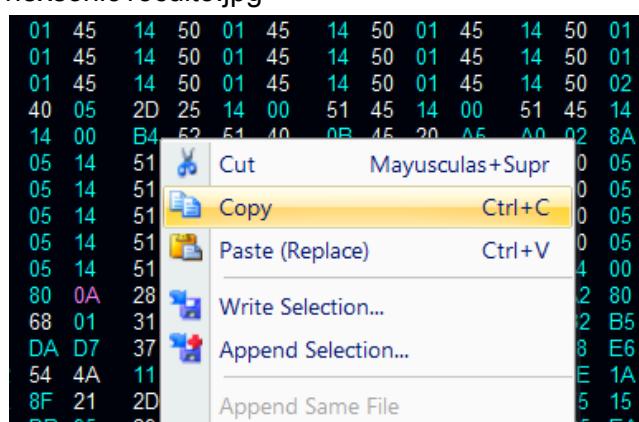


11) Se abre la imagen hexsonic1.jpg en la aplicación PhotoFiltre11 la cual corresponde a la imagen inicial

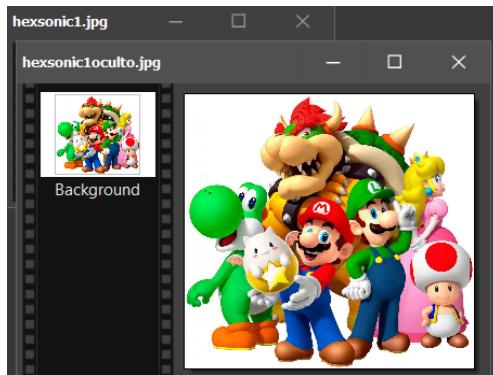


12) Luego, en Hex Edit, se puede ver que después de la imagen anterior, en la cabecera existe otro archivo con cabecera FF D8 y fin FF D9 lo que indica que hay otro archivo oculto que también corresponde a un .jpg

13) Se realiza el mismo procedimiento con la nueva imagen, copiando el contenido del nuevo archivo desde la cabecera FF D8 hasta el fin FF D9 y se guarda como hexsonic1oculto.jpg



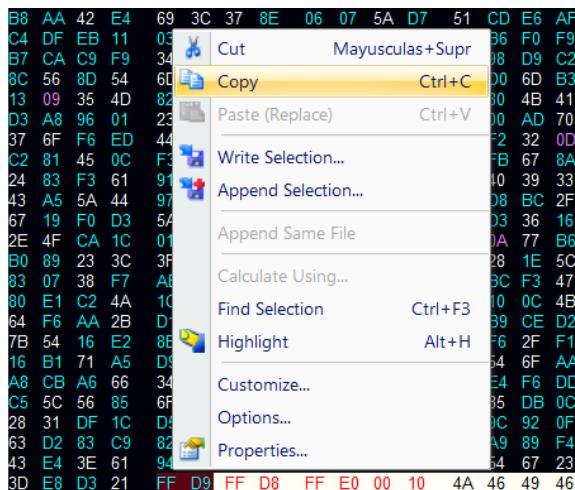
14) Se abre la nueva imagen guardada hexsonic1oculto.jpg en la aplicación PhotoFiltre11 y se puede visualizar una nueva imagen de los personajes del videojuego Mario Bros



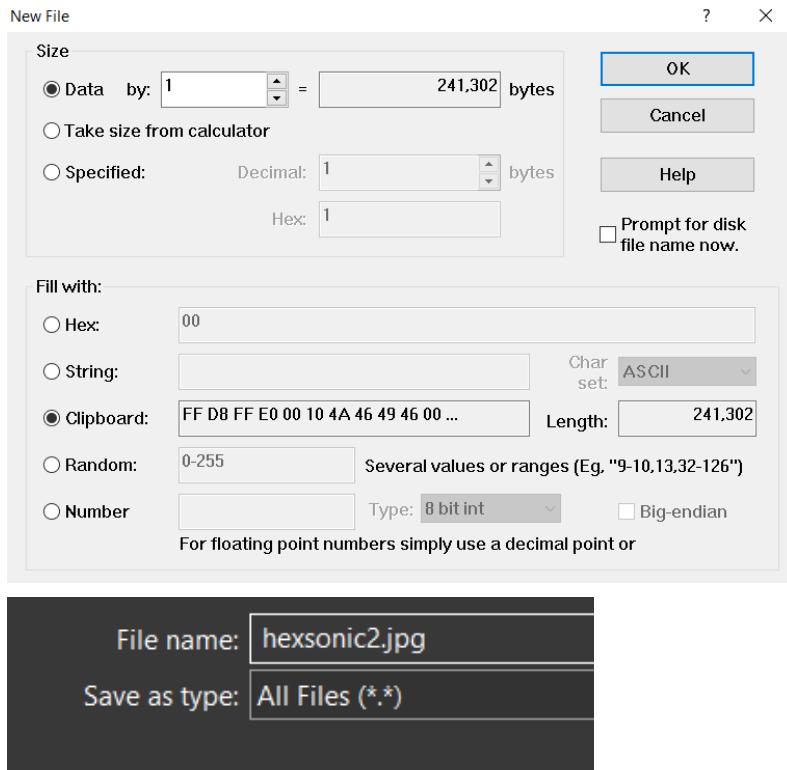
15) Ahora analizaremos la última imagen, sonic2.jpg la cual también tiene una cabecera FF D8, se busca el fin del archivo que será un FF D9 para extraer la primera imagen

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F				
0.0000:	FF	D8	FF	E0	00	10	4A	46	49	46	00	01	01	01	00	60		J	F	I	F			
0.0010:	00	60	00	00	FF	DB	00	43	00	06	04	05	06	05	04	06		C		
0.0020:	06	05	06	07	07	06	08	0A	10	0A	0A	09	09	0A	14	0E			
0.0030:	0F	0C	10	17	14	18	18	17	14	16	16	1A	1D	25	1F	1A				
0.0040:	1B	23	1C	16	16	20	2C	20	23	26	27	29	2A	29	19	1F		#		
0.0050:	2D	30	2D	28	30	25	28	29	28	FF	DB	00	43	01	07	07		-	0	-	(0	%)	(0	%)	(0	%)					
0.0060:	07	0A	08	0A	13	13	0A	0A	13	28	1A	16	1A	28	28	28			
0.0070:	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28			
0.0080:	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28			
0.0090:	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28			
0.00A0:	00	11	08	03	7D	05	28	03	01	11	00	02	11	01	03	11			
0.00B0:	01	FF	C4	00	1C	00	00	02	03	01	01	01	01	00	00	00			
0.00C0:	00	00	00	00	00	00	04	05	02	03	06	01	07	00	08	FF			
0.00D0:	C4	00	46	10	00	02	01	03	03	02	04	05	02	05	03	03		F	
0.00E0:	03	02	02	0B	01	02	03	00	04	11	05	12	21	31	41	06			
0.00F0:	13	22	51	07	14	32	61	71	23	42	15	33	52	81	91	24		"	Q	.	2	a	q	#	B	.	3	R	.	\$	
0.0100:	62	A1	43	B1	C1	08	16	72	25	34	53	82	D1	26	35	63		b	.	C	.	r	%	4	S	.	&	5	c	
0.0110:	92	E1	F0	F1	44	73	A2	FF	C4	00	1C	01	00	02	03	01		D	s
0.0120:	01	01	01	00	00	00	00	00	00	00	00	00	00	02	03	00	01		

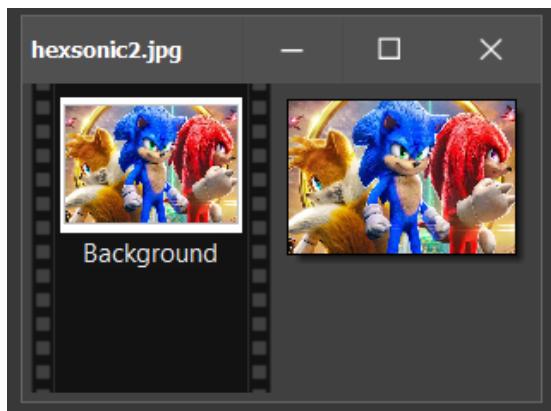
16) Se copia el contenido del primer archivo desde FF D8 hasta FF D9 para crear un nuevo archivo con extensión .jpg



17) Se guarda el primer archivo encontrado como hexsonic2.jpg



18) Se abre la imagen en la aplicación PhotoFiltre11 obteniendo la imagen inicial de sonic



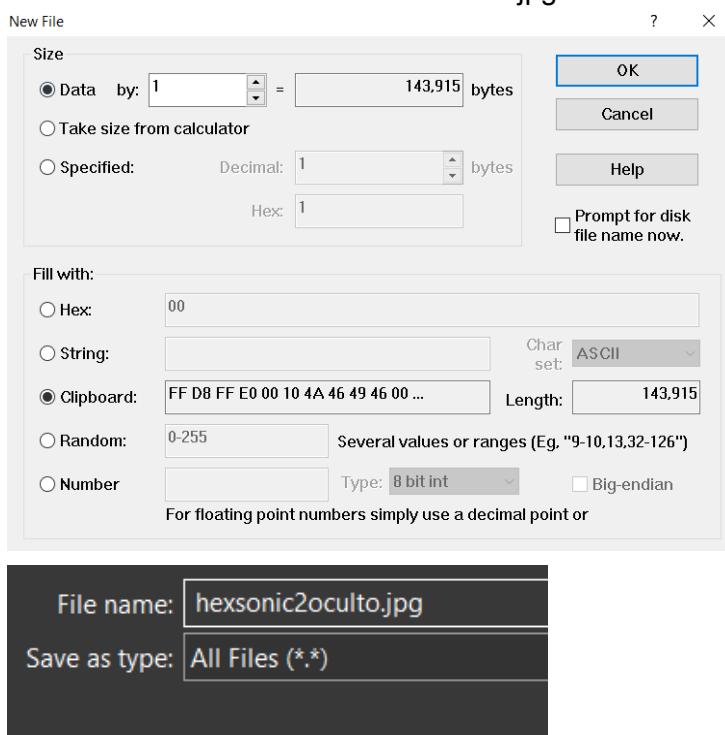
19) Ahora, en la imagen original sonic2.jpg queda una nueva cabecera que corresponde a una imagen .jpg

Se busca el final y se realiza el proceso de copiar el contenido desde FF D8 hasta FF D9.

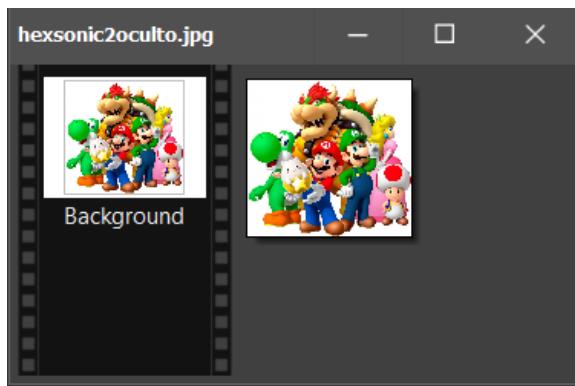
25	14	00	B4	52	51	40	0B
00	5A	29	29	68	00	A2	92
8A	4A	00	5A	29	28	CD	00
00	7F	FF	D9	FF	D8	FF	E0
01	00	00	01				
00	00	49	49				
00	00	FF	E				
61	64	6F	62				
30	2F	00	30				
69	6E	3D	24				
20	4D	70	41				



20) Se crea un nuevo archivo y se guarda con el contenido copiado creando un nuevo archivo con el nombre hexsonic2oculto.jpg

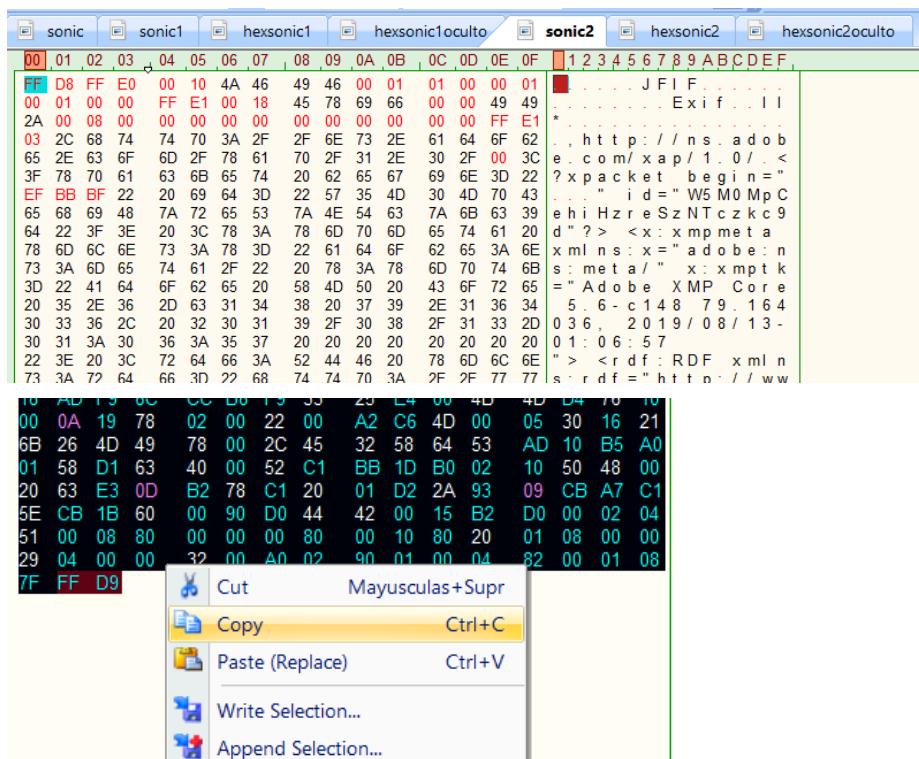


21) Se abre la imagen en PhotoFiltre11 y se identifica una imagen de los personajes de Mario Bros

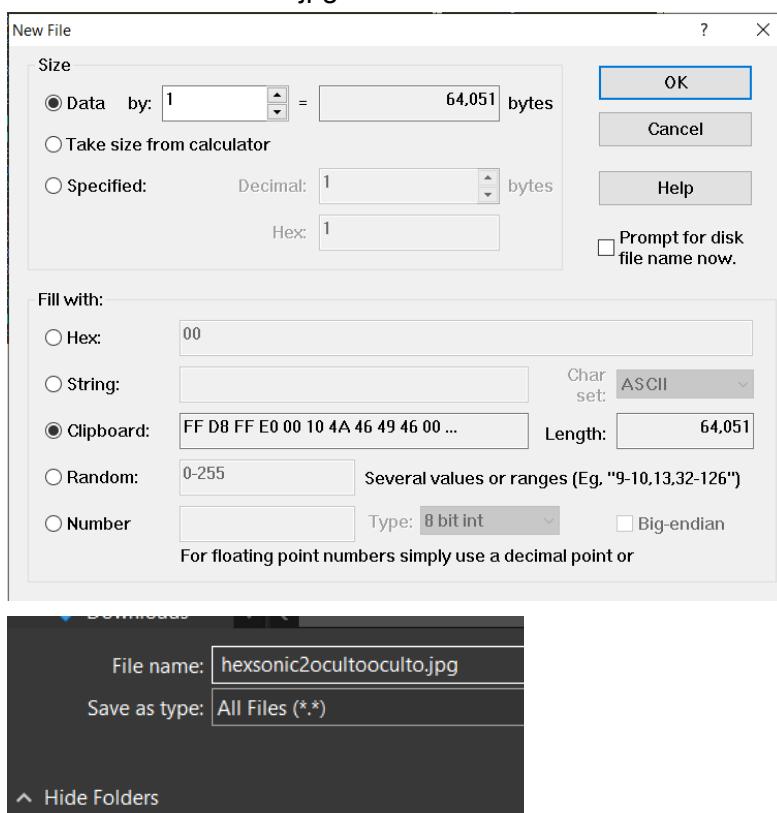


22) Luego de extraer la imagen, se puede ver que existe una nueva cabecera FF D8 en el archivo sonic2, lo cual corresponde a otro archivo en formato .jpg

Se realiza el mismo procedimiento para copiar el contenido desde la cabecera FF D8 hasta el final FF D9



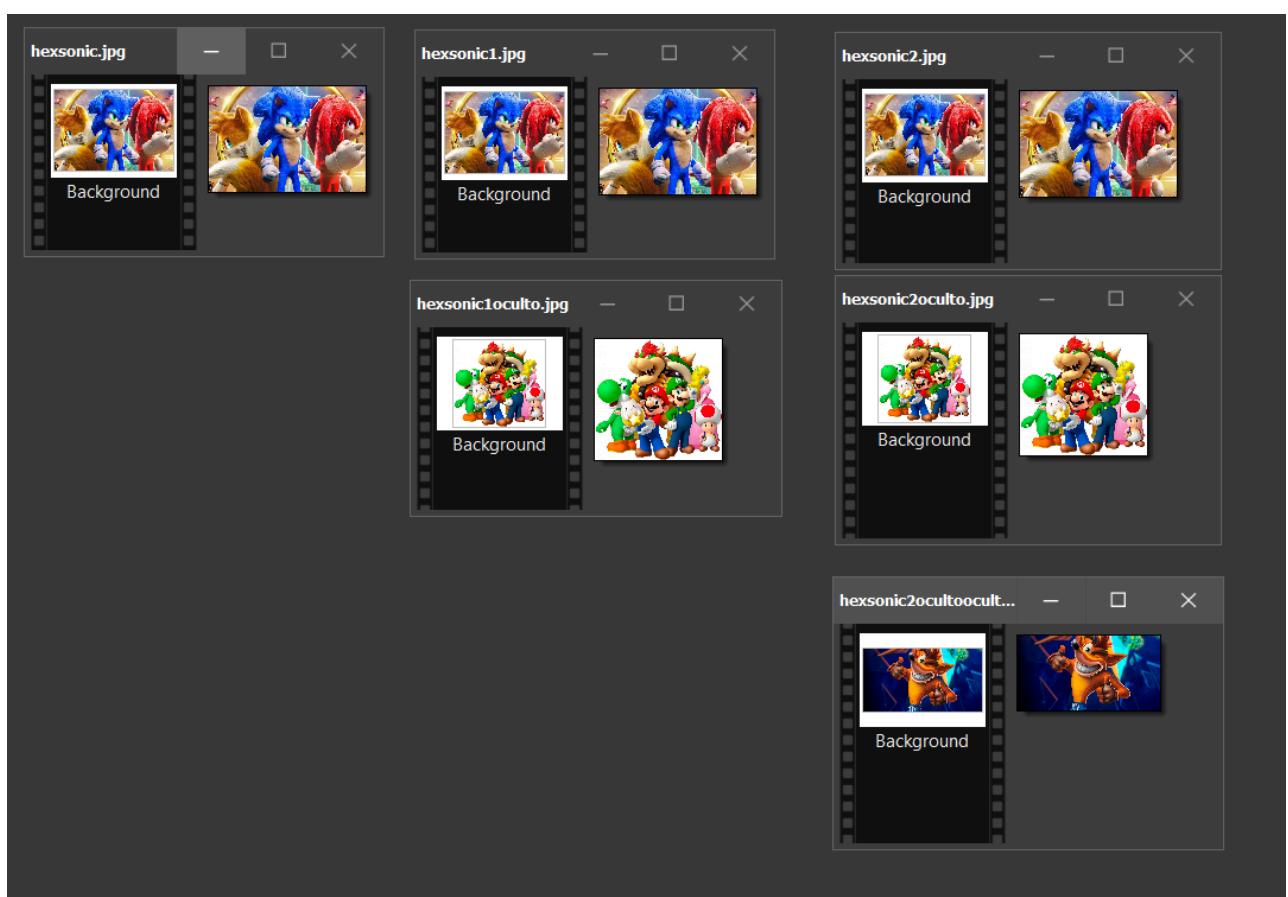
23) Se crea un nuevo archivo con el contenido copiado y se guarda como hexonic2ocultooculto.jpg



24) Se abre la imagen en la aplicación PhotoFiltre11 en donde se puede identificar una nueva imagen, en este caso de Crash Bandicoot



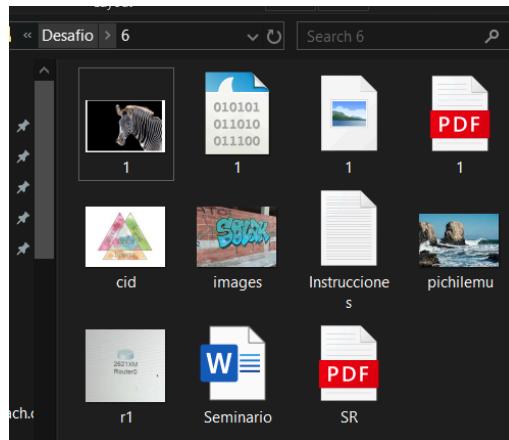
25) Finalmente se compara el contenido de las 3 fotografías (sonic.jpg, sonic1.jpg y sonic2.jpg) que inicialmente parecían iguales, pero contenían archivos ocultos dentro de ellos.



6. Resolver el desafío

Instrucciones: resolver el desafío , documente los hallazgos.

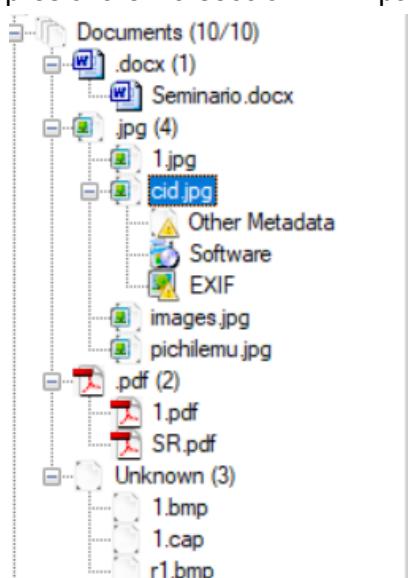
- 1) Inicialmente, se descargan todos los archivos y se utiliza el programa FOCA para identificar si alguno de los archivos tiene metadatos que puedan ser relevantes



- 2) Se agregan todos los archivos en el programa y se presiona en “Extract All Metadata”

Id	Type	URL	Download	Download Date	Size	Analyzed	Modified Date
0		C:\Users\Reinaldo\Downloads\Desafio Ciberseguridad (.	•	11/23/2024 3:57:23...	1.71 MB	x	-
1		C:\Users\Reinaldo\Downloads\Desafio Ciberseguridad (.	•	11/23/2024 3:57:35...	9.42 KB	x	-
2	jpg	C:\Users\Reinaldo\Downloads\Desafio Ciberseguridad (.	•	11/23/2024 3:57:36...	211.72 KB	x	-
3	pdf	C:\Users\Reinaldo\Downloads\Desafio Ciberseguridad (.	•	11/23/2024 3:57:37...	79.34 KB	x	-
4	jpg	C:\Users\Reinaldo\Downloads\Desafio Ciberseguridad (.	•	11/23/2024 3:57:40...	96.86 KB	x	-
5	jpg	C:\Users\Reinaldo\Downloads\Desafio Ciberseguridad (.	•	11/23/2024 3:57:41...	11.72 KB	x	-
6	jpg	C:\Users\Reinaldo\Downloads\Desafio Ciberseguridad (.	•	11/23/2024 3:57:42...	6.32 KB	x	-
7	docx	C:\Users\Reinaldo\Downloads\Desafio Ciberseguridad (.	•	11/23/2024 3:57:43...	247.25 KB	x	-
8		C:\Users\Reinaldo\Downloads\Desafio Ciberseguridad (.		11/23/2024 3:57:46...	669 KB	x	-
9	pdf	C:\Users\Reinaldo\Downloads\Desafio Ciberseguridad (.		11/23/2024 3:57:47...	70.38 KB	x	-

- 3) Se identifica que el único archivo que contiene metadatos es cid.jpg por lo que se presiona en la sección EXIF para verlos



4) No se encuentra información relevante para la resolución del desafío

Attribute	Value
Exif Makernote	
Orientation	Top, left side (Horizontal / nomal)
X Resolution	72 dots per inches
Y Resolution	72 dots per inches
Resolution Unit	Inches
Software	Adobe Photoshop CC 2017 (Macintosh)
Date/Time	2019.08.15 11:14:17
Color Space	sRGB
Exif Image Width	800 pixels
Exif Image Height	600 pixels
Compression	JPEG (old-style)
Thumbnail Offset	310 bytes
Thumbnail Length	4153 bytes
Thumbnail Data	[4153 bytes of thumbnail data]
Iptc Makernote	
Directory Version	0
Thumbnail	

5) Ahora, se procede a descifrar las contraseñas de los archivos 1.pdf, SR.pdf y Seminario.docx por lo cual se inicia la máquina virtual de Kali y se accede a la carpeta run de John The Ripper, para ello se ejecuta el siguiente comando:

```
cd john  
cd run
```

```
└─(kali㉿kali)-[~]  
└─$ cd john  
└─(kali㉿kali)-[~/john]  
└─$ ls  
CONTRIBUTING.md  doc  LICENSE  README.md  requirements.txt  run  shell.nix  src  
└─(kali㉿kali)-[~/john]  
└─$ cd run
```

6) Se guardan los archivos a descifrar en la carpeta run.

Primero se descifrará el archivo 1.pdf para lo cual se crea un hash del archivo 1.pdf con el comando:

```
./pdf2john.pl 1.pdf > 1.hash
```

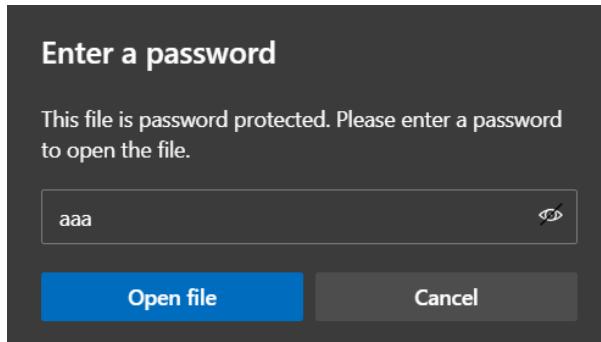
```
└─(kali㉿kali)-[~/john/run]  
└─$ ./pdf2john.pl 1.pdf > 1.hash
```

7) Después se ejecuta el comando john 1.hash para obtener la contraseña del archivo john 1.hash

```
└─(kali㉿kali)-[~/john/run]  
└─$ john 1.hash  
Using default input encoding: UTF-8  
Loaded 1 password hash (PDF [MD5 SHA2 RC4/AES 32/64])  
Cost 1 (revision) is 3 for all loaded hashes  
Will run 4 OpenMP threads  
Proceeding with single, rules:Single  
Press 'q' or Ctrl-C to abort, almost any other key for status  aruba2john.py  
Almost done: Processing the remaining buffered candidate passwords, if any.  
Proceeding with wordlist:/usr/share/john/password.lst  
aaa          (1.pdf)  
1g 0:00:00:00 DONE 2/3 (2024-11-23 15:06) 9.090g/s 70636p/s 70636c/s 70636C/s marisol.. ford  
Use the "--show --format=PDF" options to display all of the cracked passwords reliably  
Session completed.
```

En amarillo se puede ver que la contraseña del archivo 1.pdf es aaa

8) Se ingresa la contraseña para abrir el archivo 1.pdf



Se abre el pdf y se obtiene la información de que la clave está en los metadatos

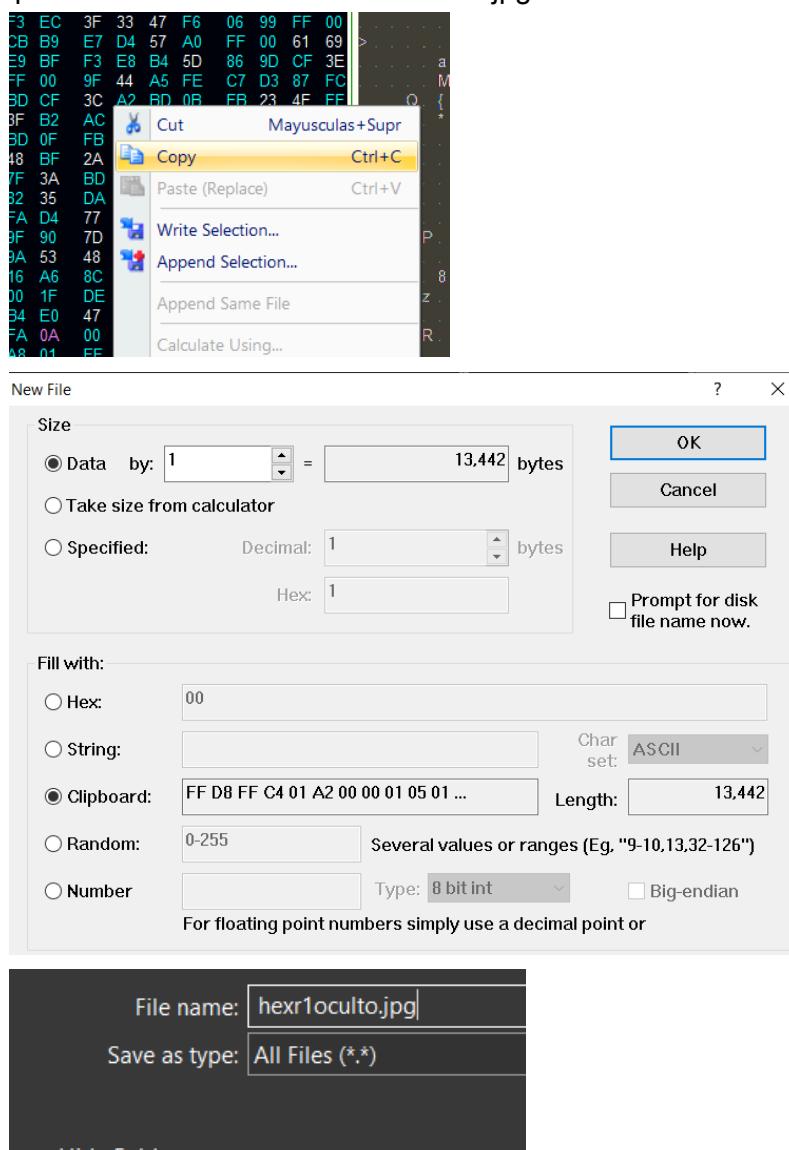


La clave esta en la metadato para el desafío final.

9) Ahora, se abre la imagen r1 (la que contiene el router) en Hex Edit la cual contiene un archivo oculto que comienza en FF D8 y finaliza en FF D9 lo que corresponde a un archivo .jpg

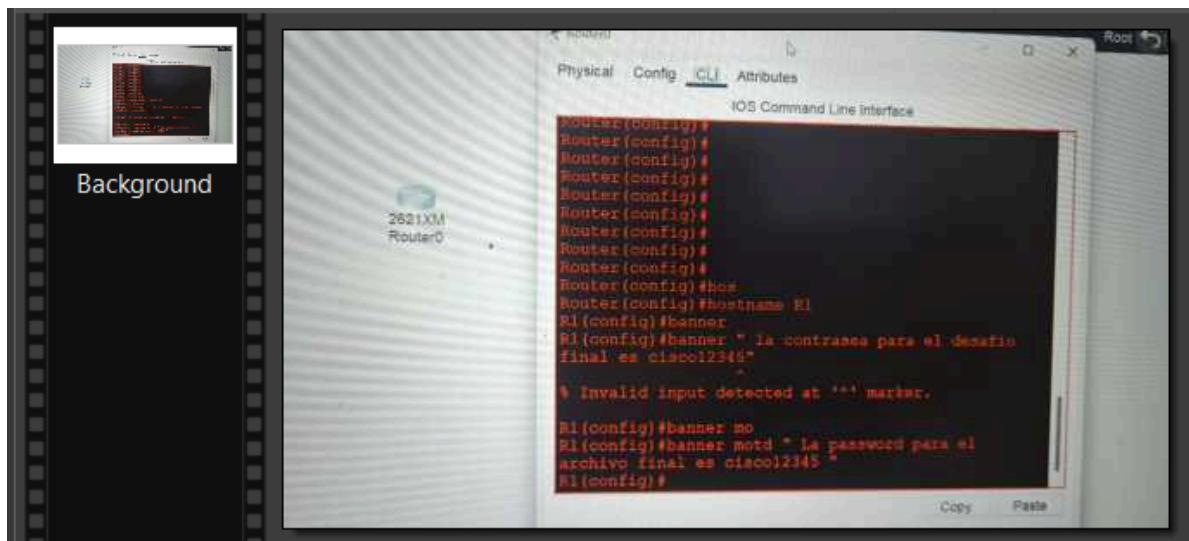
0 0460:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	00 00 06 00
0 0470:	00 01 04 00 01 00 00 00 00 02 00 00 01 01 04 00	
0 0480:	01 00 00 00 20 01 00 00 03 01 03 00 01 00 00 00	
0 0490:	06 00 00 00 00 12 01 03 00 01 00 00 00 01 00 00 00	
0 04A0:	01 02 04 00 01 00 00 00 B0 04 00 00 02 02 04 00	
0 04B0:	01 00 00 00 82 34 00 00 00 00 00 FF D8 FF C4 4	
0 04C0:	01 A2 00 00 01 05 01 01 01 01 01 01 00 00 00 00	
0 04D0:	00 00 00 00 01 02 03 04 05 06 07 08 09 0A 0B 01	
0 04E0:	00 03 01 01 01 01 01 01 01 01 00 00 00 00 00 00	
0 04F0:	00 01 02 03 04 05 06 07 08 09 0A 0B 10 00 02 01	
0 0500:	03 03 02 04 03 05 05 04 04 00 00 01 7D 01 02 03 }	
0 0510:	00 04 11 05 12 21 31 41 06 13 51 61 07 22 71 14 ! 1 A Q a " q .	
0 0520:	32 81 91 A1 08 23 42 B1 C1 15 52 D1 F0 24 33 62 2 # B R \$ 3 b	
0 0530:	72 82 09 0A 16 17 18 19 1A 25 26 27 28 29 2A 34 r %& () * 4	
0 0540:	35 36 37 38 39 3A 43 44 45 46 47 48 49 4A 53 54 5 6 7 8 9 : C D E F G H I J S T	
0 0550:	55 56 57 58 59 5A 63 64 65 66 67 68 69 6A 73 74 U V W X Y Z c d e f g h i j s t	
0 0560:	75 76 77 78 7A 82 85 86 87 88 89 8A 92 93	UVWXYZcdefghijst

10) Se copia el contenido entre la cabecera y el fin del archivo para crear un nuevo archivo que tendrá el nombre de hexr1oculto.jpg

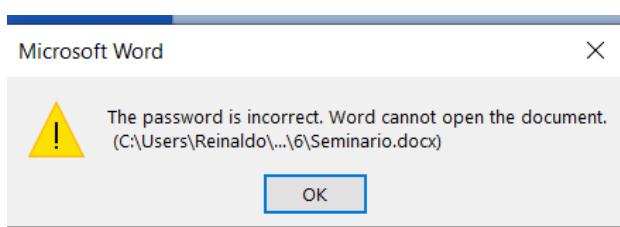
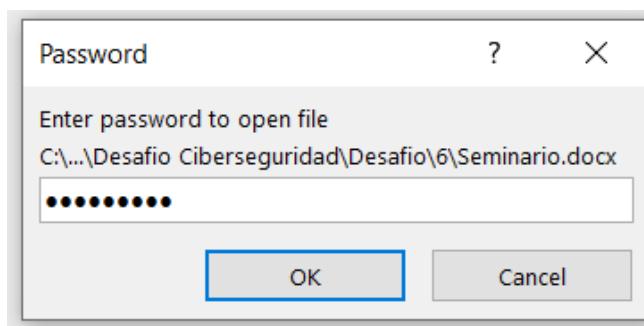


11) Se abre la imagen generada hexr1oculto.jpg en la aplicación PhotoFiltre11 lo cual resulta en una extensión de la imagen inicial del Router.

En la terminal aparece un mensaje que dice “la contraseña para el desafío final es cisco12345”

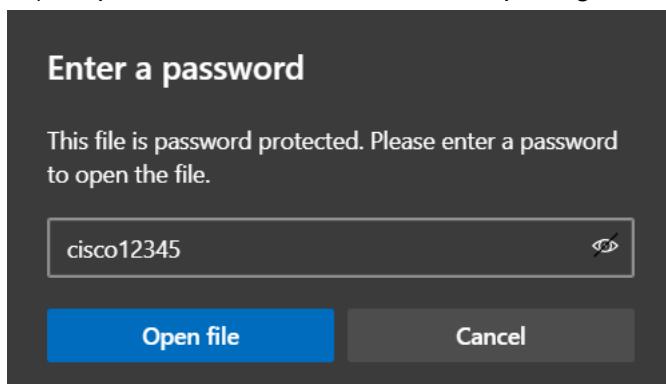


- 12) Se ingresa la contraseña cisco12345 en los archivos restantes, primero en Seminario.docx



En el archivo Seminario.docx dice que la contraseña es incorrecta

- 13) Se prueba ahora en el archivo SR.pdf ingresando la contraseña cisco12345



14) La contraseña es correcta y se logra abrir el pdf que muestra una imagen indicando que se completó el desafío.

