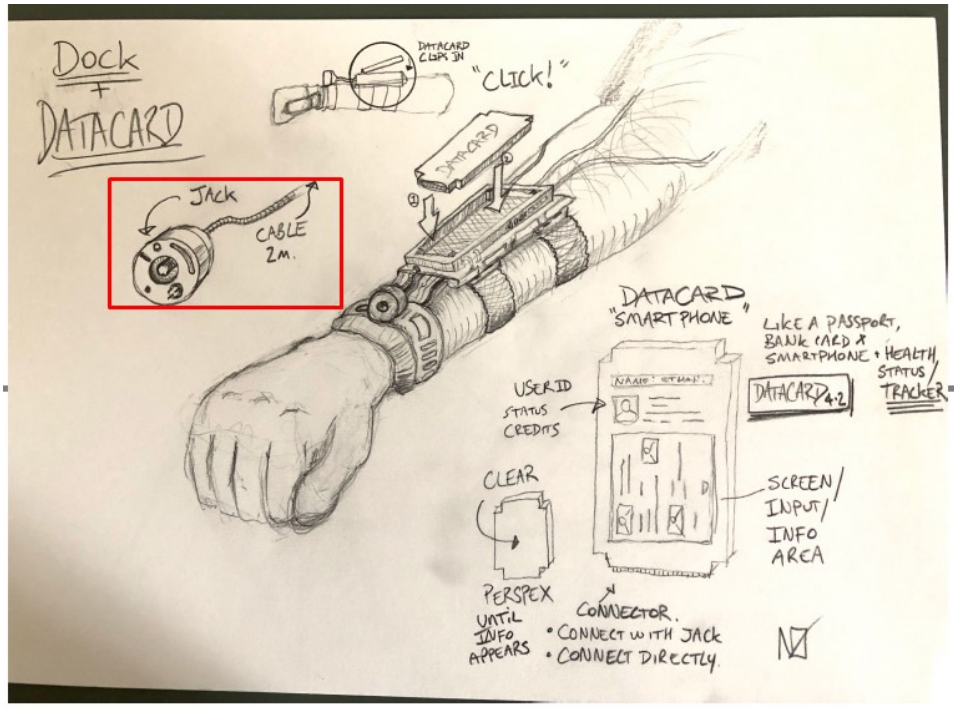
Mission Name

LazarusForensics

History Background

The Level 7 citizen is plugged into the jack, but he does not have the DATACARD:



Claire finally gets from the citizen the following evidence.

- Attempts to Change Vaccine Records
- Bank Records
- Code Snippets

Once the forensic procedure has finished, Claire obtains information about where ETHAN lives. Block 44 - SUFU, as this is the address that appears to Claire when Ethan is connected to the Level 7 Dock. Again, depending on how the evidence is provided, it will determine the difficulty of the mission.

Technical High-Level Overview

Player must carry out several recovery task to get files from the provided evidence. In this case, Citizen´s dock has information inside itself and this information is provided to the player. The goal of this mission would be to recover an SQLite database located inside Citizen´s dock.

IN GAME INFORMATION

Short Mission Description

Claire is jacking to Citizen and getting a full dump of him. Your goal is to get de exact date of the COVID19 vaccine injected.

Mission Description

Player must carry out several recovery task to get files from the provided evidence. In this case, Citizen´s dock has information inside. Claire is jacking to Citizen and getting a full dump of him. Your goal is to get de exact date of the **COVID19** vaccine injected. Insert: YYYY-MM-DD HH:MM:SS

Tools

- Photorec for recovering files: https://www.cgsecurity.org/testdisk-7.2-WIP.linux26-x86_64.tar.bz2
- Wxhexeditor: https://howtoinstall.co/es/wxhexeditor
- Whatsapp Parser Tool set: https://github.com/B16f00t/whapa
- Exiftool: https://www.poftut.com/how-to-install-and-use-exiftool-in-linux-windows-kali-ubuntu-mint-with-examples/
- SQLite viewer tool to read the database extracted.

Credentials

- **N/A**

Questions

Which is the first two hex bytes of the 7Z file?

- 377A

Which is the Primary Platform of the JPG file extracted?

- Microsoft Corporation

How many bytes were needed to decrypt the database?

- 158

Hints

1. Check header and footer of the evidence provided.
2. Extract the JPG file based on the footer.
3. Decrypt the content of the 7Z file based on the JPC picture.

Location

- SYLVARCON | ANUTM DISTRICT | LEVEL 7 CLIENT'S APARTMENT

Write Up

First step would be to identify header and footer of the evidence provided. To accomplish this, use wxHexEditor:

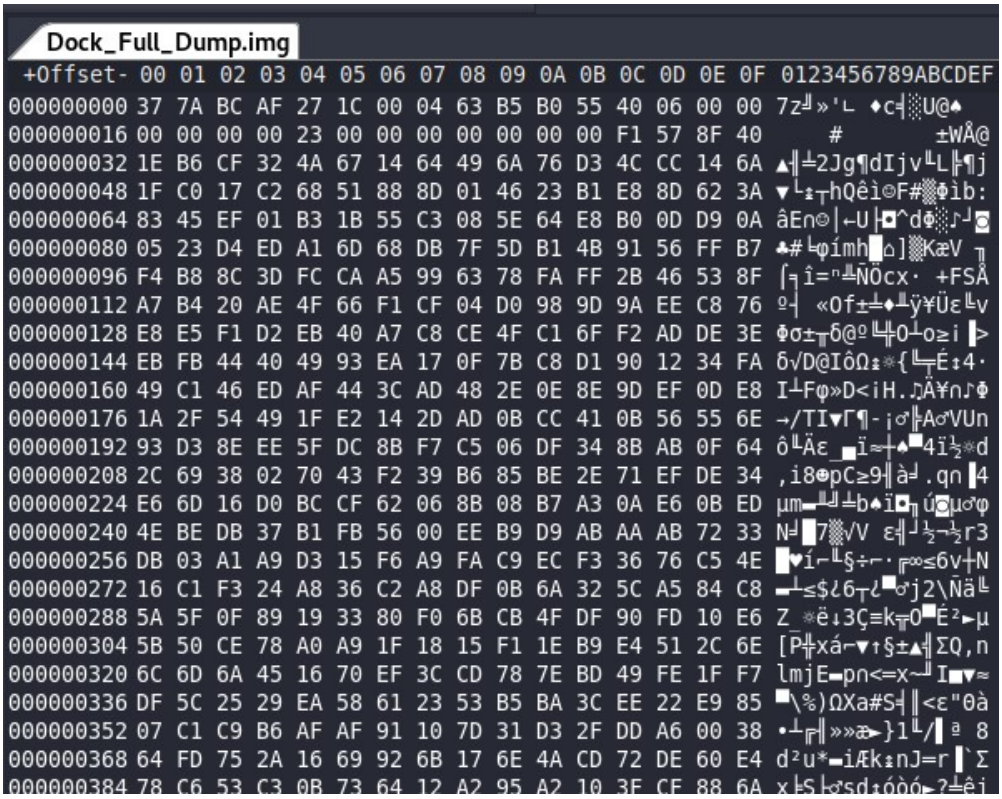


Figure 1

Above image represents the header of the evidence. It´s a 7Z file.

The following image represents the footer of the evidence:

Figure 2

The footer has FF D9 values. It´s necessary to discover which file is, so searching on Google would be easy to find it:

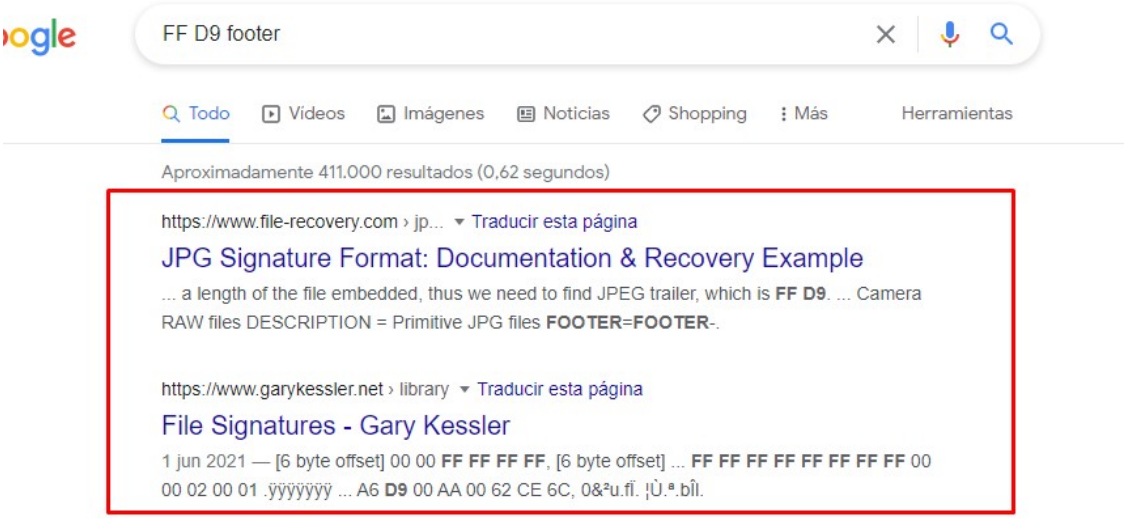


Figure 3

To sum up: we´ve found out two files: 7Z file and JPG file. To extract them, it would be essential to use a tool like Photorec. Once downloaded Photorec for Linux, we have the following binaries:

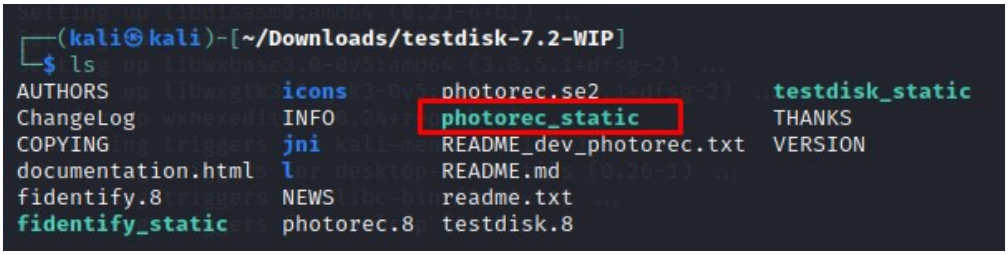


Figure 4

Photorec_static is the main binary to execute. ./photorec_static Dock_Full_Dump.img

Once executed, let´s forward to the recovery phase, selecting:

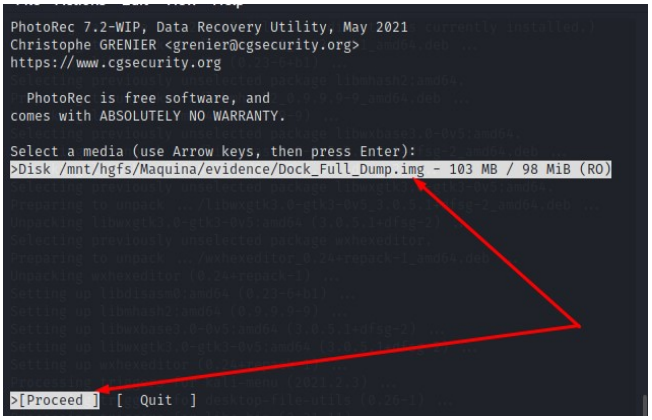


Figure 5

And now, file Opt:

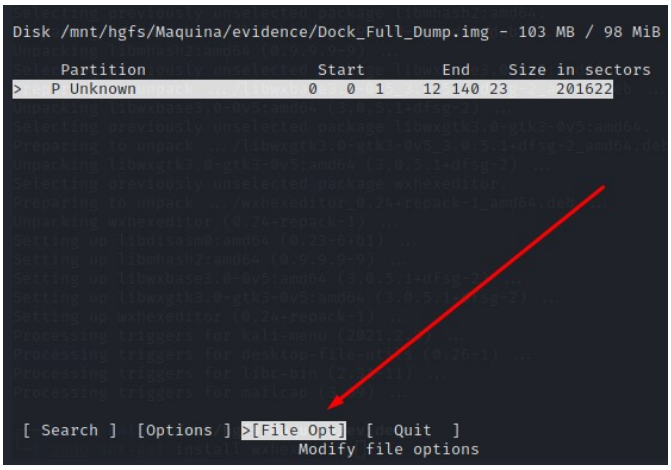


Figure 6

Leave all selected files, in order to recovery the whole content:

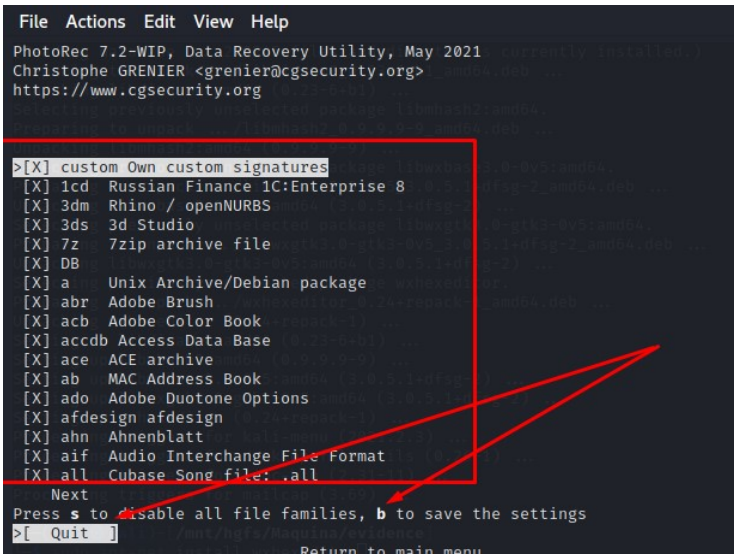


Figure 7

Select b, to save settings and then Quit to return to main window. Then, leave P Unknown option and select Search:

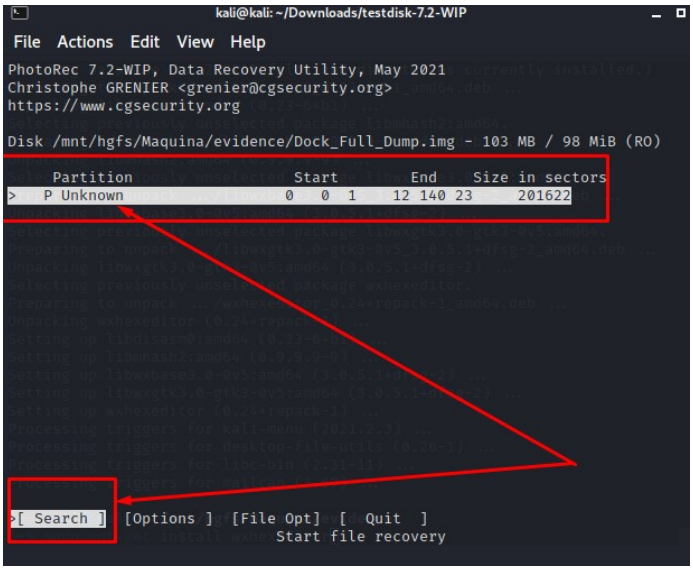


Figure 8
Figure 9

Select other:

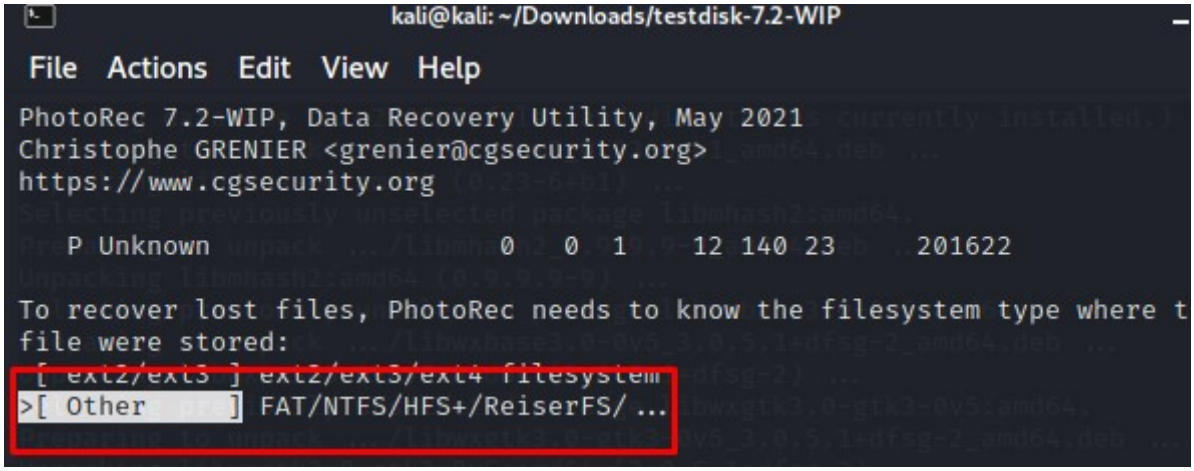


Figure 10

And finally, player must select folder to extract recovered files:

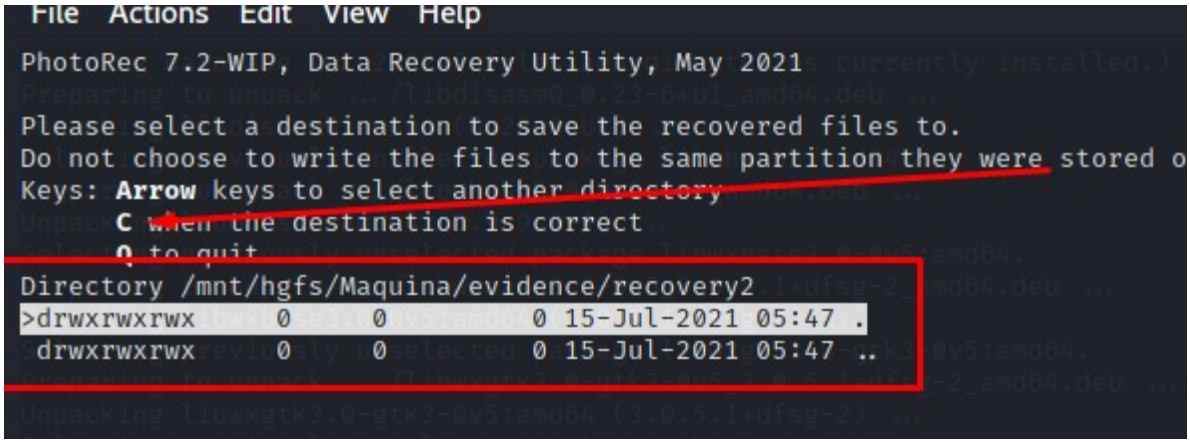


Figure 11

Last step would be select C to confirm the folder. After Photorec finishes, Quit it. In the same folder that you must analyse recovered files. 7Z file was recovered but the JPG wasn´t recovered:

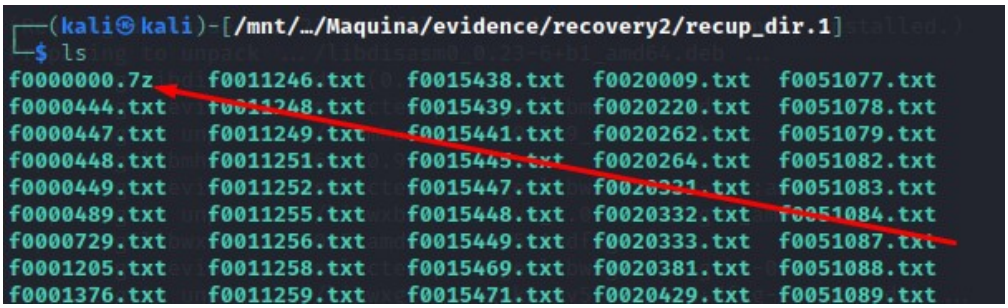


Figure 12

To extract the JPG file, it´s mandatory to use wxHexEditor, to locate the header of the JPG file. Searching on Internet, you will find: https://www.garykessler.net/library/file_sigs.html

FF D8 ÿØ
JPE, JPEG, JPG Generic JPEGimage file
Trailer: FF D9 (ÿÙ)

NOTES on JPEG file headers: The proper JPEG header is the two-byte sequence, 0xFF-D8, aka *Start of Image (SOI)* marker. JPEG files end with the two-byte sequence, 0xFF-D9, aka *End of Image (EOI)* marker.

Between the SOI and EOI, JPEG files are composed of *segments*. Segments start with a two-byte *Segment Tag* followed by a two-byte *Segment Length* field and then a zero-terminated string identifier (i.e., a character string followed by a 0x00), as shown below with the JFIF, Exif, and SPIFF segments.

Segment Tags of the form 0x-FF-Ex (where x = 0..F) are referred to as APP0-APP15, and contain application-specific information. The most commonly seen APP segments at the beginning of a JPEG file are APP0 and APP1 although others are also seen. Some additional tags are shown below:

- 0xFF-D8-FF-E0 — Standard JPEG/JFIF file, as shown below.
- 0xFF-D8-FF-E1 — Standard JPEG file with Exif metadata, as shown below.
- 0xFF-D8-FF-E2 — Canon Camera Image File Format (CIFF) JPEG file (formerly used by some EOS and Powershot cameras).
- 0xFF-D8-FF-E8 — Still Picture Interchange File Format (SPIFF), as shown below.

FF D8 FF E0 xx xx 4A 46 49 46 00 ÿØÿà..JF IF.
JFIF, JPE, JPEG, JPG JPEG/JFIF graphics file
Trailer: FF D9 (ÿÙ)

FF D8 FF E1 xx xx 45 78 69 66 00 ÿØÿá..Ex if.

Figure 13

We don´t know anything about the JPG file, so the main pattern to find the header would be: FF D8 FF. Use wxHexEditor to find it: Select the magnifying glass, and then insert the pattern, find all:

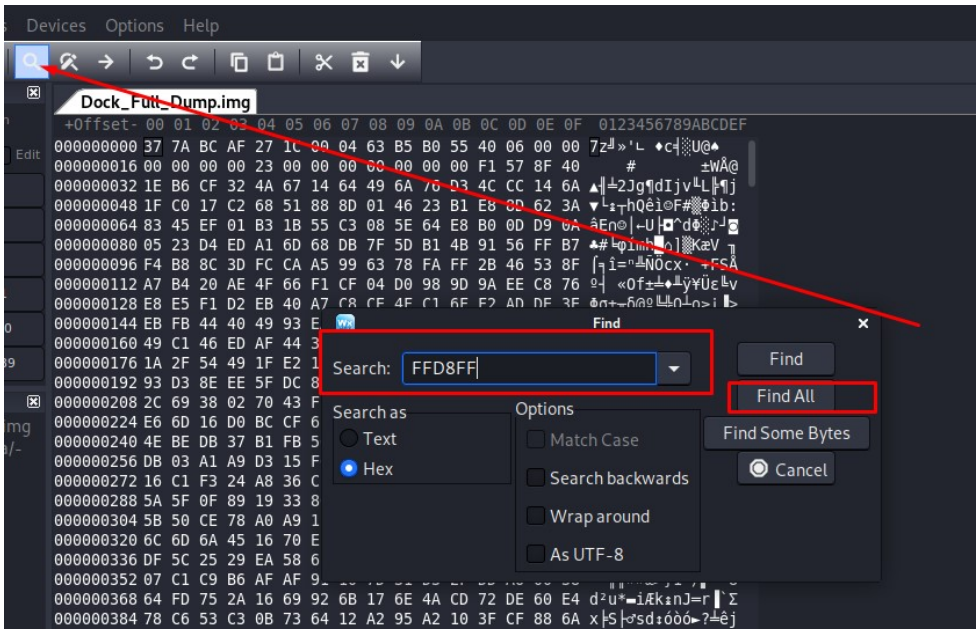


Figure 14

Finally found 334 matches:

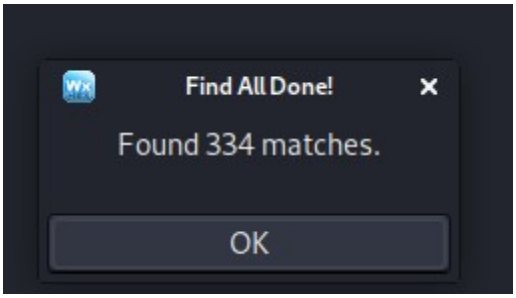


Figure 15

The key to extract the specific JPG file, would be to locate the last match. This means that the last match is the header of the file. (Keeping in mind, the footer is JPG file, and the footer of the evidence provided). On Search results windows you will find the results: OFFSET 103125280

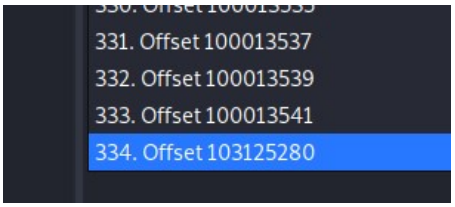


Figure 16

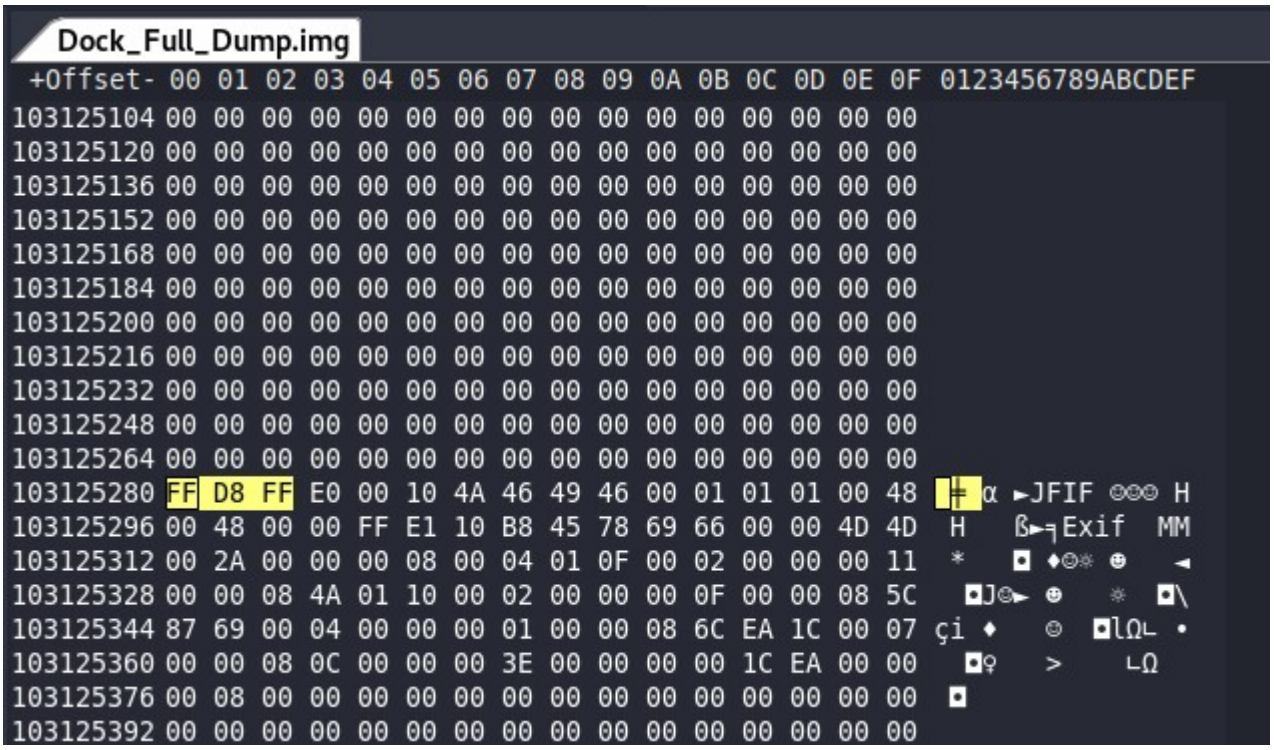


Figure 17

On this offset, right click and select:

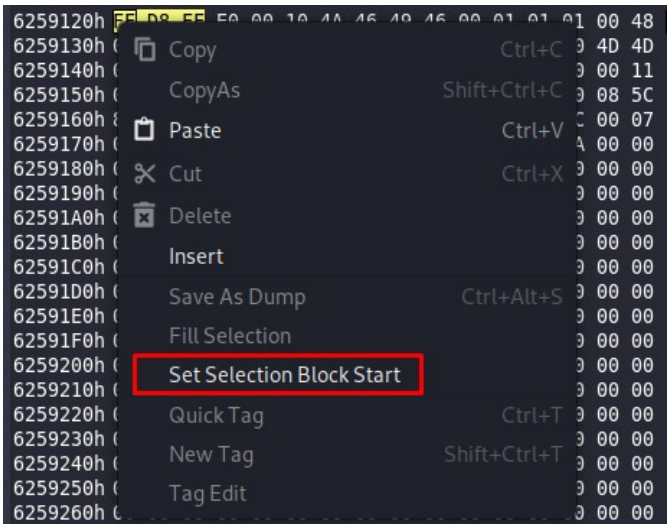


Figure 18

Go to the end of the file, and repeat the step:

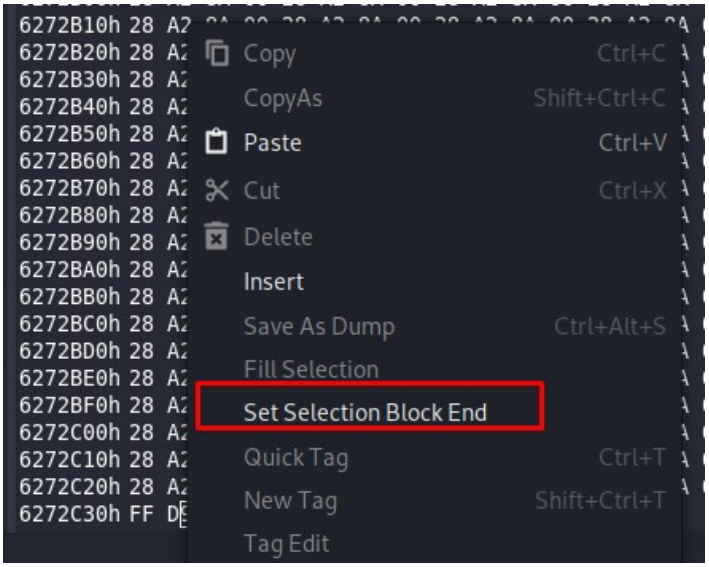


Figure 19

You will able to see the whole JPG file select, ready to extract from the evidence provided:

Figure 20

Finally extract the JPG file, selecting Edit -> Save as dump. There we, We´ve got it!

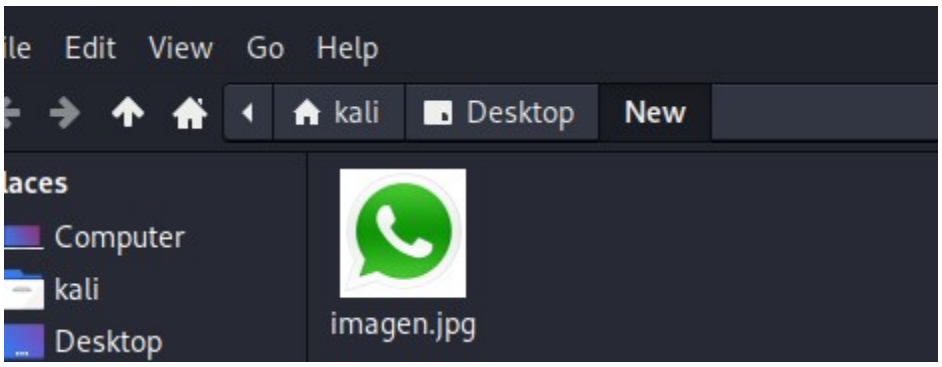


Figure 21

It´s a whatspp picture. This is the key to the following steps. A good one forensicator will test medata of the JPG files, using exiftool tool: exiftool imagen.jpg

```
File Permissions                : rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : inches
X Resolution                    : 72
Y Resolution                    : 72
Exif Byte Order                 : Big-endian (Motorola, MM)
Make                            : This is the clue
Camera Model Name               : file and file2
Warning                         : Invalid EXIF text encoding for UserComment
User Comment                    : cmp3.9.27.3q4 0×dcdf0d34
Padding                         : (Binary data 2060 bytes, use -b option to e
xtract)
```

Figure 22

Pay attention on the above results: "This is the clue" and "file and file2". If you compare these clues with the content of the 7Z file, you will guess, that the image extracted is related to the 7Z file:

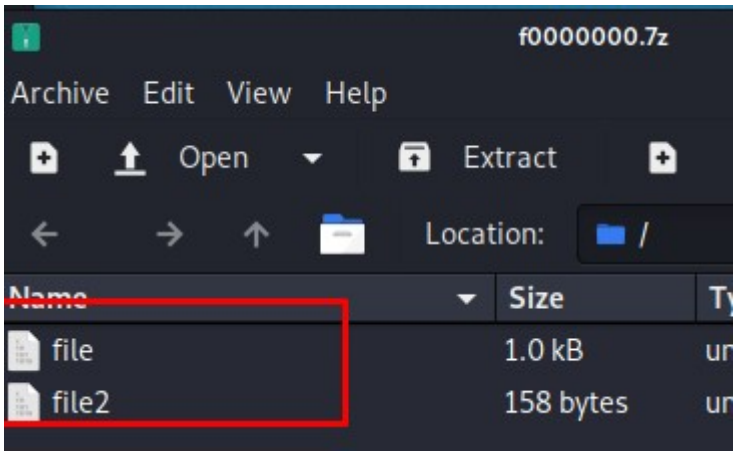


Figure 23

At this point, player has:

- File
- File 2
- Whatsapp JPG file pointing to file1 and file 2

File it´s an SQLite Database encrypted using the steps like WhatsApp encryption. File2 , it´s the key to decrypt the SQLite database.

File2:

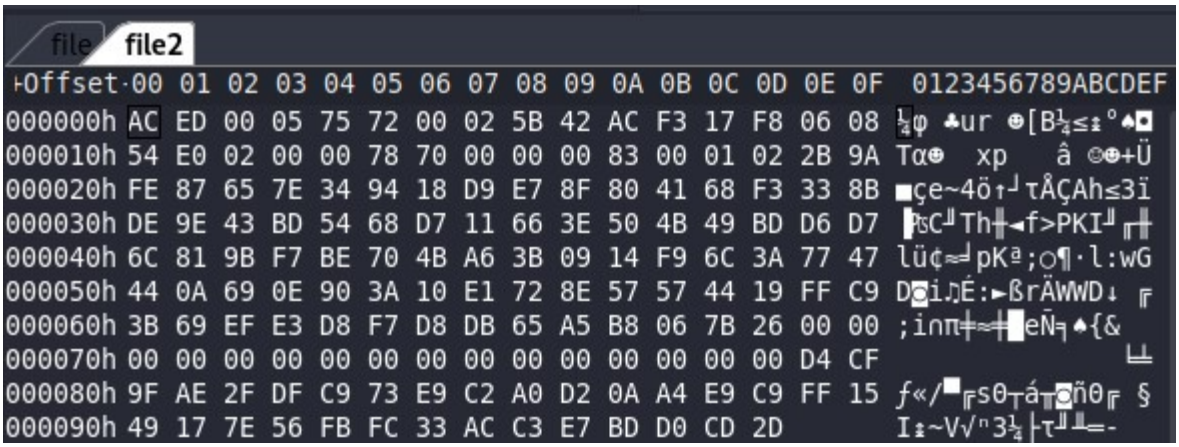


Figure 24

File:

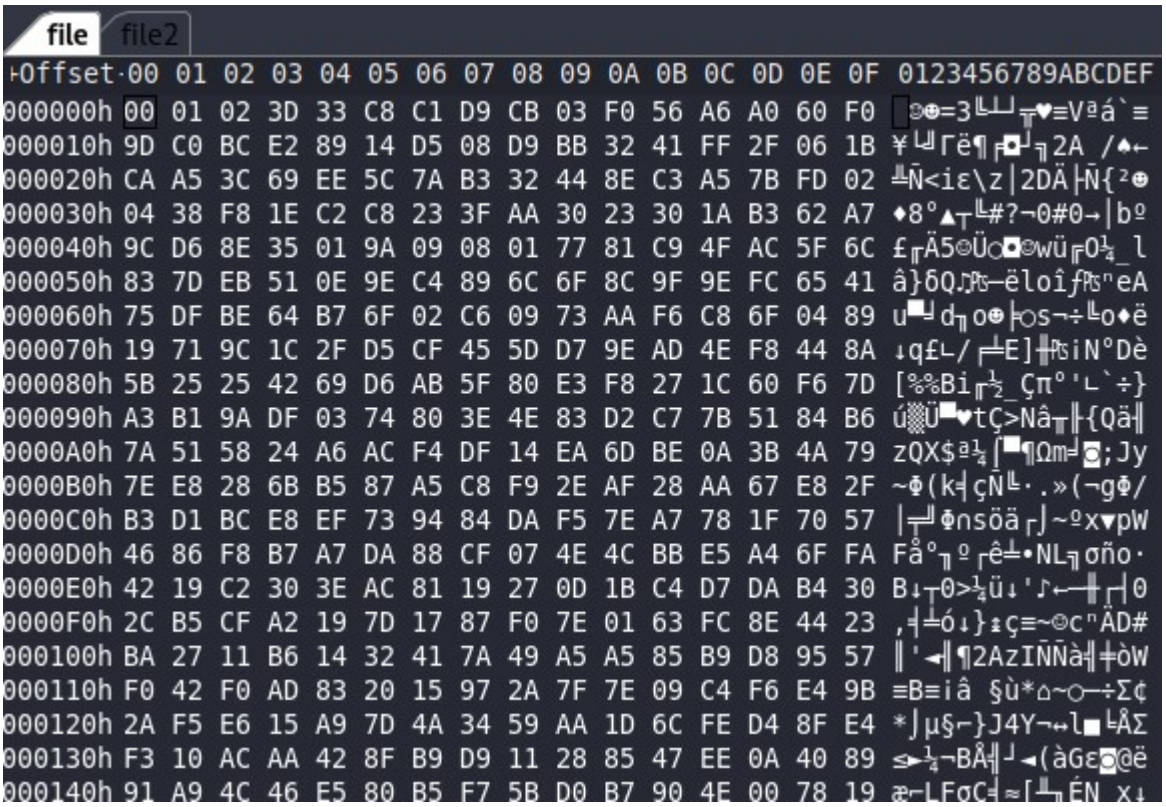


Figure 25

In this phase of the challenge, player could use a user friendly tool like WhatsAPP parsers tools. The command to launch the tool would be: python3 whapa-gui.py

To use this tool, player must try to rename the file for the following options:

- file.crypt14
- file.crypt13
- file.crypt12

And so on… The correct option is **file.crypt12.** The ending of the file, indicate the type of the encryption. There is no other way to find out unless you try manually.

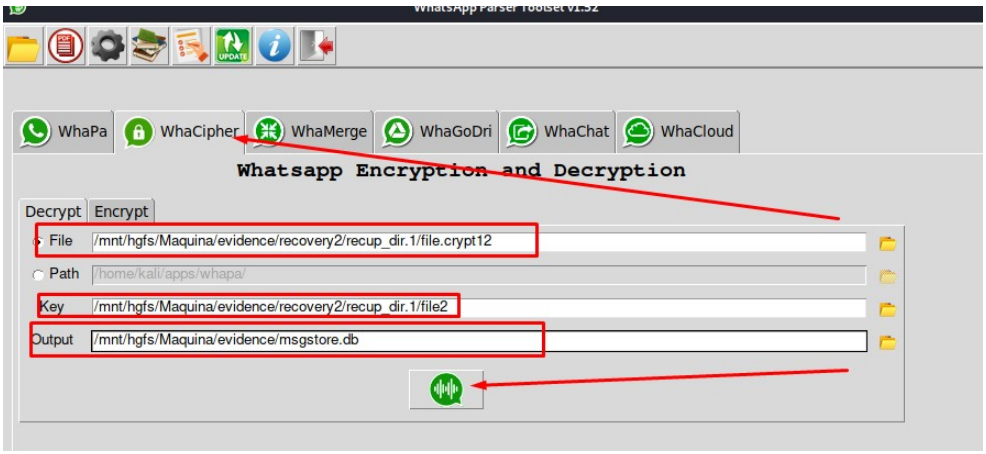


Figure 26

Select WhaCipher, locate the renamed file "file.crypt12", locate the key to decrypt and the ouput folder. Finally check the last button. Open an SQLite Data browser on Kali, and select the unencrypted database:

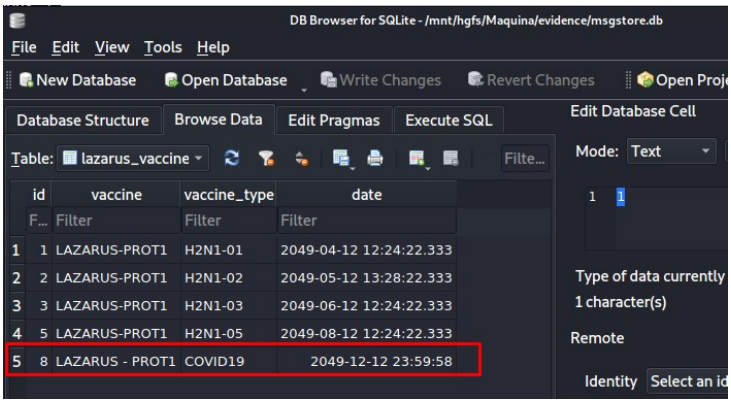


Figure 27

The flag would be: 2049-12-12 23:59:58

Flag

flag{2049-12-12 23:59:58}