Forencics

Writeup Report

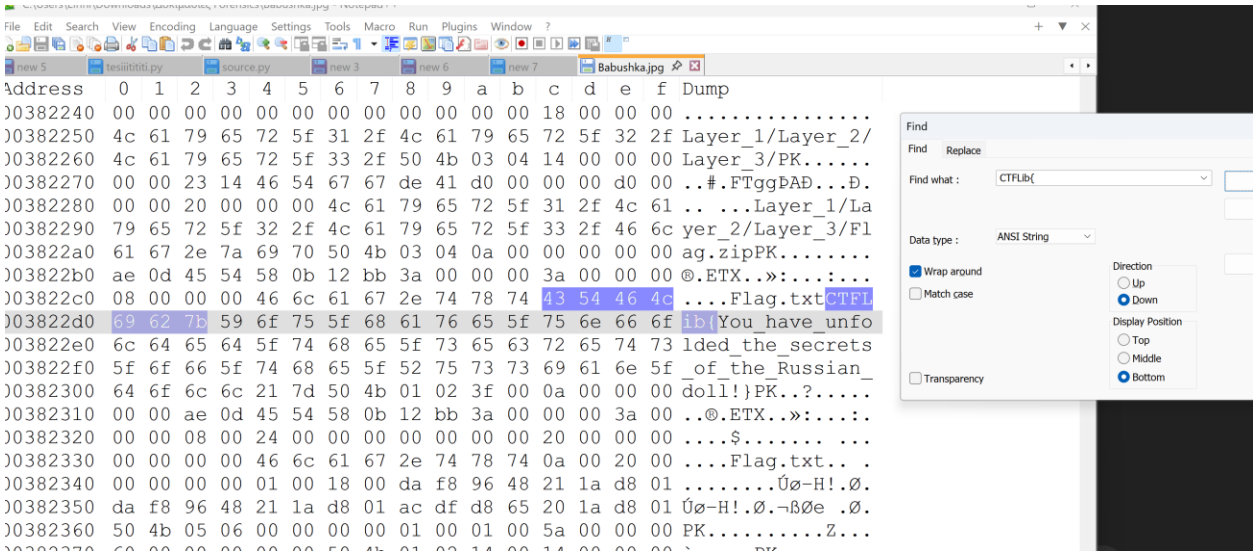
1. **Babushka**

Challenge Description

We were given a zipped file named forensics, which contained an image called Babushka.jpg. The challenge hinted that, much like a real Russian nesting doll, this image would contain multiple layers of hidden information. Our goal: to find the hidden flag of the format CTFLib{...}.

Inspection (Hex Analysis)

a. The zipped file ‘forencics’ has been downloaded.
b. The image Babushka.jpg has been opened via Notepad++ Hex editor plugin.
c. By clicking View in Hex in the text editor plugins, we see the raw hexadecimal representation of the image data. Since the challenge describes the CTFLib{..} would be hidden inside, we search for this ANSI string within the hex dump. Next up, the results show



…that the string is indeed found, and also there is the message inside the Flag.txt that shows the encrypted message, CTFLib{You_have_unfolded_the_secrets_of_the_Russian_doll!}

The hex dump revealed references to Flag.txt and the text. This confirmed that there was a hidden flag embedded in the image.

So there is one more step to be made.

d. An internal structure is noticed, indicating an embedded ZIP archive: Layer1/Layer2/Layer3/Flag.zipPK.
e. Locating the ZIP's starting signature (50 4B 03 04 in hex) and its ending signature (50 4B 05 06), which marks the end of the central directory record,
    1. copying the whole raw byte block and
    2. pasting it in a new file as a binary paste,
    3. saving the file as bab.zip, we finally get the result.

    Using the archive tool 7-zip, we open the bab.zip and inside we're now able to see the Flag.txt which inside contains the flag clearly now seen :
    CTFLib{You_have_unfolded_the_secrets_of_the_Russian_doll!}

Conclusion

In this challenge, the Babushka.jpg file was modified to contain a nested ZIP archive. By analyzing the image in a hex editor and extracting the embedded archive, we revealed the hidden **Flag.txt** file. Ultimately, we recovered the flag:

CTFLib{You_have_unfolded_the_secrets_of_the_Russian_doll!}

2. Hacked

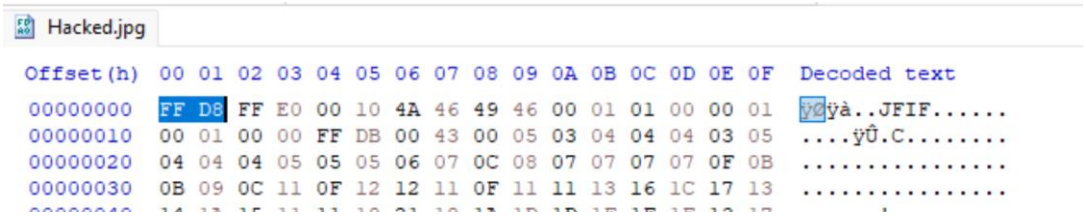
Challenge Description

We were given an image file named Hacked.jpg that looked normal at first glance. Our objective was to investigate whether it contained any hidden data or had been otherwise tampered with, and to find a flag in the format CTFLib{...} if present.

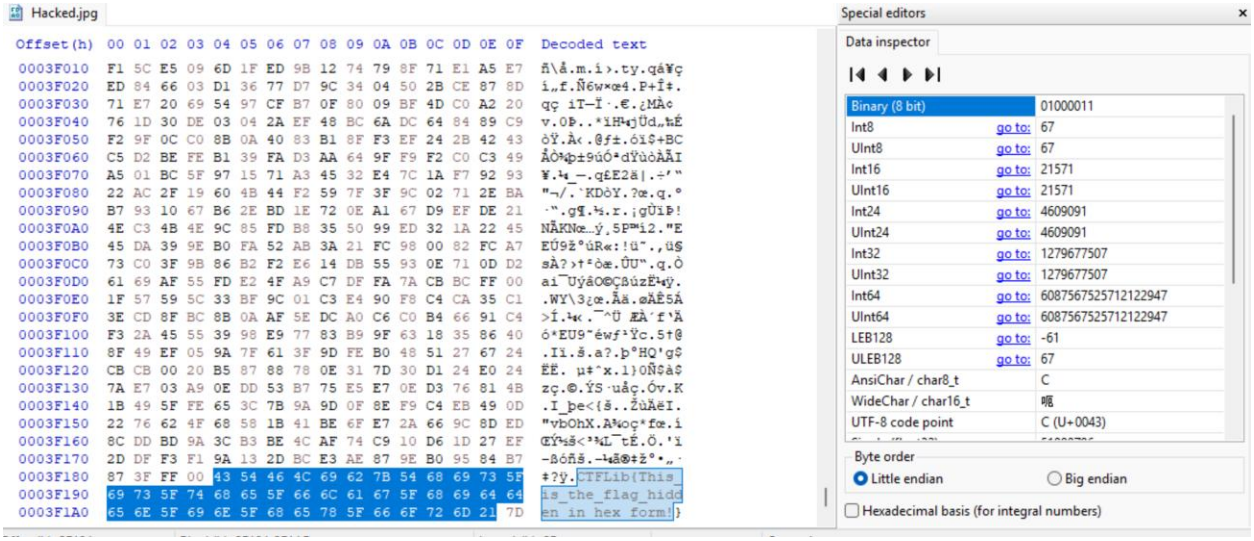
Hex Analysis with HxD

a. For this challenge we use the tool **HxD** (a Windows-based hex editor) and open the Hacked.jpg. We're able to see the bytes of the file image.
b. We search again for the String CTFLib{ in ANSI to see if there is somewhere in the file.

c. A standard JPG file typically ends with the bytes FF D9 and the file actually we see that indeed starts the standard way with FF D8



.. but it doesn't really end with the expected bytes. Instead :



d. we observed that the file's ending bytes include FF 00 followed immediately by an ASCII string, the **CTFLib{This_is_the_flag_hidden_in_hex_form!}** so it is definitely modified.
e. Extra data (the hidden flag) has been intentionally appended after what would be the normal image content. The image appears normal while hiding a secret message, so it's indeed Hacked.

Conclusion

By analyzing Hacked.jpg in a hex editor, we discovered that the image's data does not conclude with the usual FF D9 marker. Instead, extra data beginning with FF 00 is appended, which contains the hidden flag:

CTFLib{This_is_the_flag_hidden_in_hex_form!}

..confirming the "Hacked" nature of the challenge.

3. **Information**

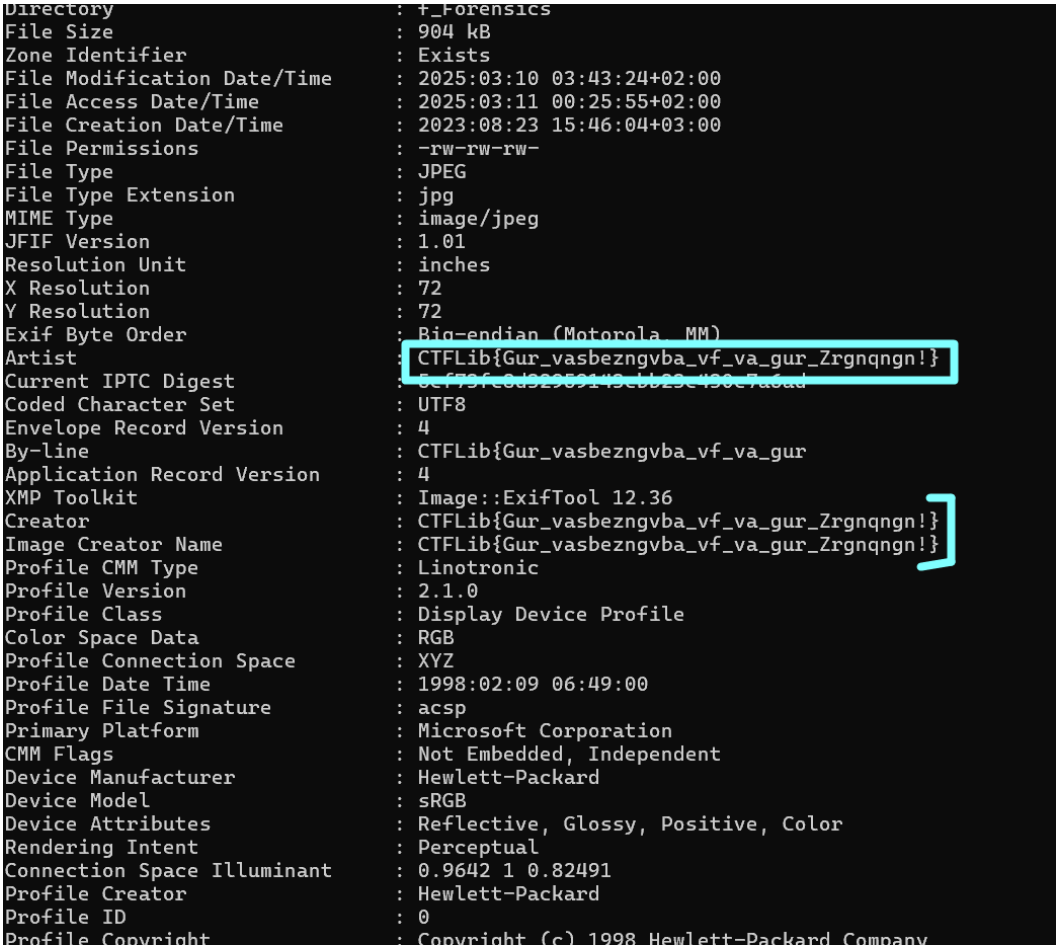
Challenge Description

We have the image Information.jpg that appears to show an adorable missing dog. The hint suggests we should look for details about the dog's owner, implying the crucial data may be hidden in the image's metadata.

Metadata Analysis

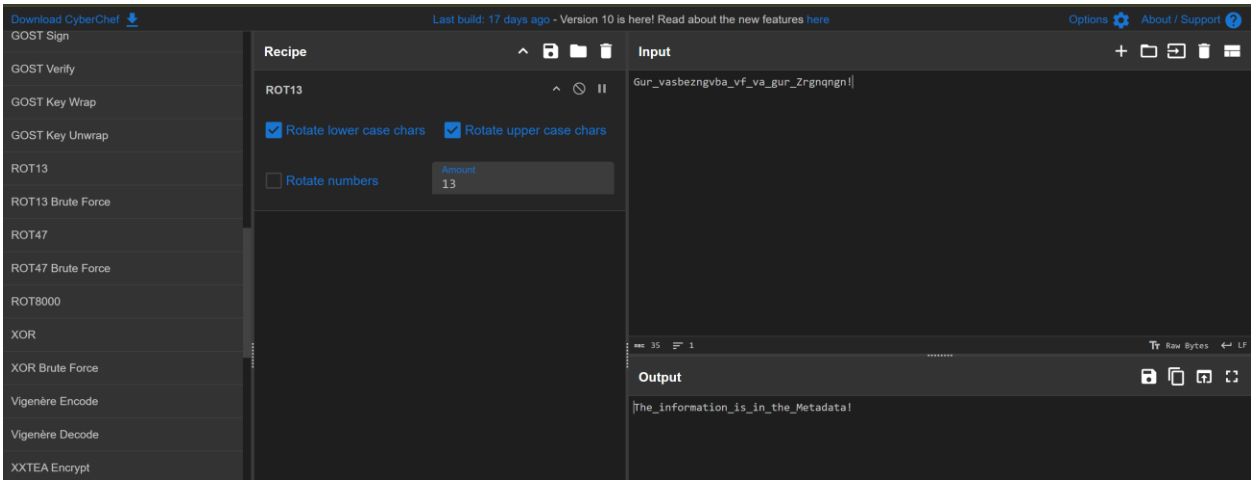
a. For this challenge we use the ExifTool and we run the **exiftool Information.jpg** in command line. (there is a small report about how the exiftool was directly used from cmd).

b. In the **Artist**, **Creator**, or **Image Creator** fields, we discovered a string that looked encoded or encrypted:
CTFLib{Gur_vasbezngvba_vf_va_gur_Zrgnqngn!}



There is not an encryption suspicion because headers and structure of the file follows the expected. So the message is just encoded.

c. Using the Cyber Chef from https://gchq.github.io/CyberChef/ we try to decode the message by experimenting with multiple common encryption/encoding schemes, ROT13 was the one that gave us the result:



d. There isn't a name in the end. We only found the message:
The_information_is_in_the_Metadata!
The challenge is not done; This is not the name of the owner. We should try find its owner through physical means… 🔎

Conclusion

Using **ExifTool**, we examined the metadata of Information.jpg and discovered a ROT13-encoded string in the Artist/Creator/Image Creator field. After trying common cipher methods, ROT13 seem to correctly decode the message. This revealed the final flag:

CTFLib{The_information_is_in_the_Metadata!}