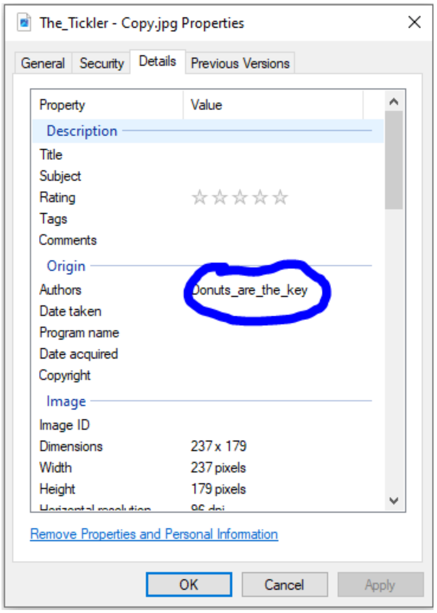
Overall Notes

- You will notice that the overall pattern for advancing through the floors was to find the hidden message, and use that message to unlock the .7z archive contained in the floor
- The .7z archive would then contain a celebratory gif, and the password/key to unlock the next floor
    - *Note here, I think it would've been better to acknowledge that the celebratory gif had nothing to do with the solution, apologies for that…*

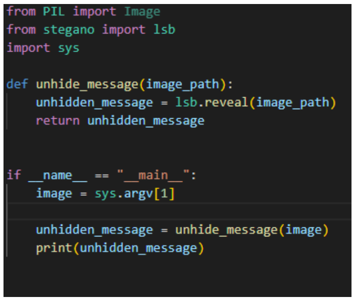
Floor 1 - The tickler

- This flag was meant to be a bit of an intro to get everyone warmed up and just utilized the EXIF data to hide the key
- The key for the .7z archive was hidden in the "Author" field of the image
    - 

The_Tickler - Copy.jpg Properties

Property	Value
Description	
Title	
Subject	
Rating	☆☆☆☆☆
Tags	
Comments	
Origin	
Authors	Donuts_are_the_key
Date taken	
Program name	
Date acquired	
Copyright	
Image	
Image ID	
Dimensions	237 x 179
Width	237 pixels
Height	179 pixels

Floor 2 - Lipp Cervas

- This flag utilized the Least Significant Bit technique to hide a message in the image
- A bit of sample Python code to decode this message might look something along the lines of this:
    - 

```python
from PIL import Image
from stegano import lsb
import sys

def unhide_message(image_path):
    unhidden_message = lsb.reveal(image_path)
    return unhidden_message


if __name__ == "__main__":
    image = sys.argv[1]

    unhidden_message = unhide_message(image)
    print(unhidden_message)
```

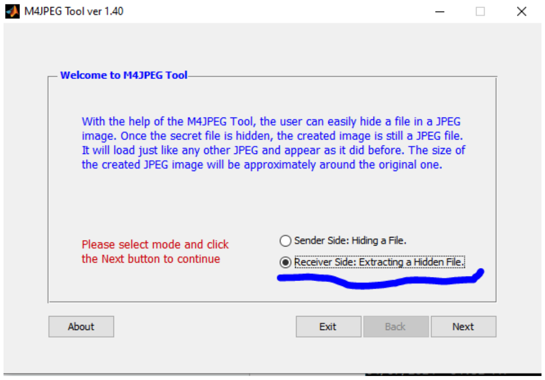
- This was a relatively simple method just using the **lsb** module from the **stegano** library for python to hide the message with lsb.hide(), and the lsb.reveal() function to reveal the message

Floor 3 - Filthy Phil

- This flag utilized the frames of a gif to hide a message
- The gif was split into frames and each letter of the message was then drawn onto the frame
- The way I had intended for people to solve this was to:
    - Recognize that there are random letters appearing in the top left corner
    - Either split the gif into frames, or convert it to mp4
        - https://ezgif.com/split
        - https://ezgif.com/gif-to-mp4
    - Then manually re-assemble the message frame by frame

Floor 4 - Master Udon

- This flag admittedly was significantly harder than the first 3
- The technique used here was using the Discrete Cosine Transform technique using **m4jpeg** to hide the message
    - I won't pretend like i understand the inner workings of this but I used the tool called **m4jpeg** to hide a text file within the but the plan here was just that everyone tries different techniques until they figure out which technique was used and/or they are very smart and know how to detect DCT in an image
- Then essentially once it was discovered that DCT was the technique, you would then be able to use the same tool I did to decode the message
    - ***The tool also utilizes a key, and the key needed here was KarateIsland***

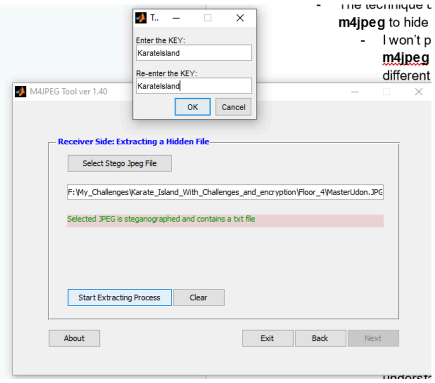

M4JPEG Tool ver 1.40

Welcome to M4JPEG Tool

With the help of the M4JPEG Tool, the user can easily hide a file in a JPEG image. Once the secret file is hidden, the created image is still a JPEG file. It will load just like any other JPEG and appear as it did before. The size of the created JPEG image will be approximately around the original one.

Please select mode and click the Next button to continue

○ Sender Side: Hiding a File.
● Receiver Side: Extracting a Hidden File.


T..

Enter the KEY: KarateIsland

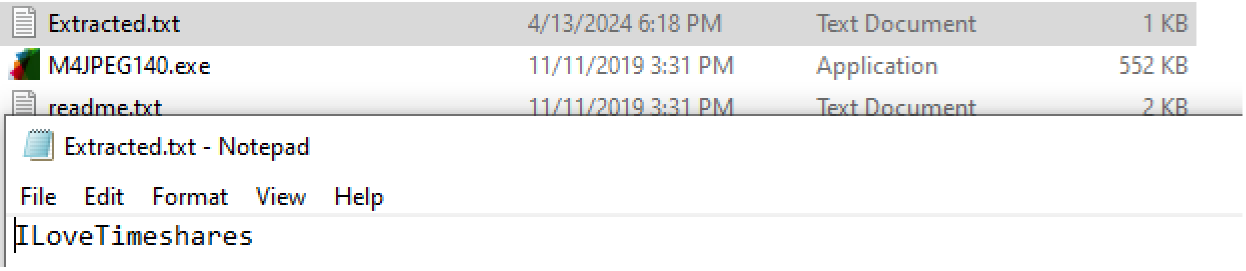
Re-enter the KEY: KarateIsland

M4JPEG Tool ver 1.40

Receiver Side: Extracting a Hidden File

F:\My_Challenges\Karate_Island_With_Challenges_and_encryption\Floor_4\MasterUdon.JPG

Selected JPEG is steganographed and contains a txt file



Extracted.txt	4/13/2024 6:18 PM	Text Document	1 KB
M4JPEG140.exe	11/11/2019 3:31 PM	Application	552 KB
readme.txt	11/11/2019 3:31 PM	Text Document	2 KB

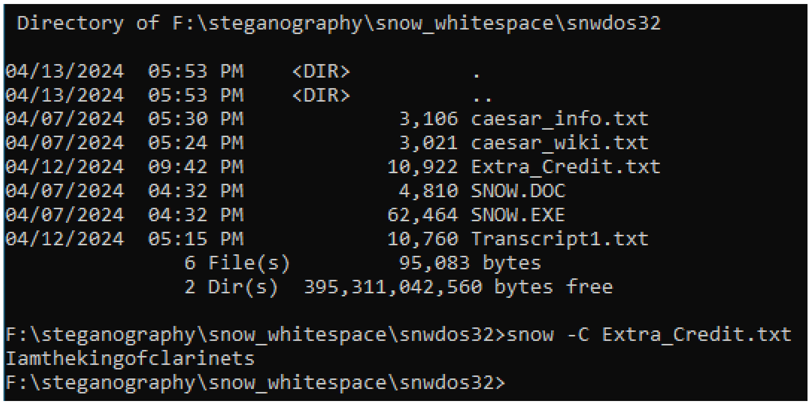
Extracted.txt - Notepad

```
ILoveTimeshares
```

- **M4jpeg tool:** https://digitnet.github.io/m4jpeg-tool/
    - I would highly recommend reading up on how this tool, I definitely will be, to understand how this works more in depth

Extra Credit Floor

- This flag was a relatively basic one where I utilized a tool called **snow** to hide a message in the whitespace in the Transcript.txt file
  - Note: there are probably other tools you could use to hide/find this message but this one works pretty well for Windows (not sure about other operating systems).



```
 Directory of F:\steganography\snow_whitespace\snwdos32

04/13/2024  05:53 PM    <DIR>          .
04/13/2024  05:53 PM    <DIR>          ..
04/07/2024  05:30 PM             3,106 caesar_info.txt
04/07/2024  05:24 PM             3,021 caesar_wiki.txt
04/12/2024  09:42 PM            10,922 Extra_Credit.txt
04/07/2024  04:32 PM             4,810 SNOW.DOC
04/07/2024  04:32 PM            62,464 SNOW.EXE
04/12/2024  05:15 PM            10,760 Transcript1.txt
               6 File(s)         95,083 bytes
               2 Dir(s)  395,311,042,560 bytes free

F:\steganography\snow_whitespace\snwdos32>snow -C Extra_Credit.txt
Iamthekingofclarinets
F:\steganography\snow_whitespace\snwdos32>
```

- **Snow whitespace-tool:** https://darkside.com.au/snow/