Looking through the source code reveals the executable uses **PyInstaller**, meaning we have some python to extract.

After using **pyinstxtractor** to extract the pyc, we find a pyc named img2bytes.pyc, exactly what we were looking for.

Using an online decompiler reveals the initial source code, and that the script converts an image into hexadecimal color values and formats it as an assembly-like data structure. We can easily reverse this:

```python
from PIL import Image
import re

def parse_hexels(filePath):
    with open(filePath, 'r') as f:
        lines = f.readlines()

    hexels = []
    pattern = re.compile(r'0([0-9a-fA-F]{6})h')

    for line in lines:
        matches = pattern.findall(line)
        hexels.extend(matches)

    width = len(matches) if matches else 0
    height = len(lines)

    return hexels, width, height

def hex_to_pixels(hexels):
    pixels = []
    for hexel in hexels:
        r = int(hexel[0:2], 16)
        g = int(hexel[2:4], 16)
        b = int(hexel[4:6], 16)
        pixels.append((r, g, b))

    return pixels

def rebuild_image(filePath, outputPath):
    hexels, width, height = parse_hexels(filePath)
    pixels = hex_to_pixels(hexels)

    if width == 0 or height == 0:
        print("Error: Unable to determine image dimensions.")
        return
    im = Image.new('RGB', (width, height))
    im.putdata(pixels)
    im.save(outputPath)
    print(f'Image saved as {outputPath}')

rebuild_image('asm_image.inc', 'output.png')
```

The generated image contains a number which is also the “message” mentioned in the flag format.

Made with love by: AndreiCat