

Forum:[Hack Forums](#)

Thread: [Monopoly Market EXIF data scraper](#)

BigFruity - August 27, 2021, 2:37 pm

Code `#!/usr/bin/env python3`

```
import requests
from exif import Image
```

```
LOCALHOST = "127.0.0.1"
#PORT = str(9050) # Edit if needed.
PORT = "9150"
proxies = {
'http': 'socks5h://' + LOCALHOST + ':' + PORT,
'https': 'socks5h://' + LOCALHOST + ':' + PORT
}
```

```
# Tiffs break that are out of official format.
```

```
# The market:
# 4jm77gv7h36xfvnslizyavt2anheditededited.onion
```

```
# Order of operations:
# 1. Get user input.
# 2. Scrape the user market photo to local drive.
# 3. Process the local market photo on the local drive. (Save all file information to separate
corresponding text files that accompany the photo name)
```

```
user = input('Input Market Photo: ')
name = user.split('/')[1]
exifile = name.split('.')[0] + '.exif'
```

```
def write_image(url):
    r = requests.get(str(user), proxies=proxies)
    with open(name, 'wb') as image_file:
        image_file.write(r.content)
        image_file.flush()
```

```
def write_exif(data):
    with open(exifile, 'a+') as image_file:
        image_file.write(data + '\n')
        image_file.flush()
```

```
def main():
    write_image(user)

    my_image = Image(name)
    exifs = my_image.list_all()

    if len(exifs) > 0:
```

```
print("This image has exif metadata.")
```

```
if 'gps_longitude_ref' in exifs:
```

```
    print('gps longitude reference:', str(my_image.gps_longitude_ref))  
    write_exif('gps longitude reference: ' + str(my_image.gps_longitude_ref))
```

```
if 'gps_altitude' in exifs:
```

```
    print('gps altitude:', str(my_image.gps_altitude))  
    write_exif('gps altitude: ' + str(my_image.gps_altitude))
```

```
if 'gps_timestamp' in exifs:
```

```
    print('gps timestamp:', str(my_image.gps_timestamp))  
    write_exif('gps timestamp: ' + str(my_image.gps_timestamp))
```

```
if 'gps_datestamp' in exifs:
```

```
    print('gps datestamp:', str(my_image.gps_datestamp))  
    write_exif('gps datestamp: ' + str(my_image.gps_datestamp))
```

```
if 'gps_altitude_ref' in exifs:
```

```
    print('gps altitude reference:', str(my_image.gps_altitude_ref))  
    write_exif('gps altitude reference: ' + str(my_image.gps_altitude_ref))
```

```
if 'gps_latitude_ref' in exifs:
```

```
    print('gps latitude reference:', str(my_image.gps_latitude_ref))  
    write_exif('gps latitude reference: ' + str(my_image.gps_latitude_ref))
```

```
if 'gps_latitude' in exifs:
```

```
    print('gps latitude:', str(my_image.gps_latitude))  
    write_exif('gps latitude: ' + str(my_image.gps_latitude))
```

```
if 'gps_longitude' in exifs:
```

```
    print('gps longitude:', str(my_image.gps_longitude))  
    write_exif('gps longitude: ' + str(my_image.gps_longitude))
```

```
if 'model' in exifs:
```

```
    print('model:', str(my_image.model))  
    write_exif('phone model: ' + str(my_image.model))
```

```
if 'make' in exifs:
```

```
    print('make:', str(my_image.make))  
    write_exif('phone make: ' + str(my_image.make))
```

```
if 'software' in exifs:
```

```
    print('software:', str(my_image.software))  
    write_exif('software: ' + str(my_image.software))
```

```
if 'datetime' in exifs:
```

```
    print('datetime:', str(my_image.datetime))  
    write_exif('date and time: ' + str(my_image.datetime))
```

```
if 'gps_dop' in exifs:
```

```
    print('gps_dop:', str(my_image.gps_dop))  
    write_exif('GPS DOP: ' + str(my_image.gps_dop))
```

```
elif len(exifs) < 1:
    print("This image has no exif metadata.")

if __name__ == '__main__':
    try:
        main()
    except Exception:
        print('JPEG Bytes are Out Of Order.')
        print('This is not an actual JPEG.')
        print('You may need to manually check the image.')
```

SAMPLE data pulled:

```
Code gps longitude reference: W
gps altitude: 95.0
gps timestamp: (13.0, 32.0, 31.0)
gps datestamp: 2021:03:13
gps altitude reference: GpsAltitudeRef.ABOVE_SEA_LEVEL
gps latitude reference: N
gps latitude: (36.0, 37.0, 30.0)
gps longitude: (4.0, 30.0, 6.0)
phone model: SM-G930F
phone make: samsung
software: G930FXXS8ETC3
date and time: 2021:03:13 14:33:19
gps longitude reference: W
gps altitude: 95.0
gps timestamp: (13.0, 32.0, 31.0)
gps datestamp: 2021:03:13
gps altitude reference: GpsAltitudeRef.ABOVE_SEA_LEVEL
gps latitude reference: N
gps latitude: (36.0, 37.0, 30.0)
gps longitude: (4.0, 30.0, 6.0)
phone model: SM-G930F
phone make: samsung
software: G930FXXS8ETC3
date and time: 2021:03:13 14:33:19
```

Even onoin sites can have leaks, too. Be smart about your OPSEC think things through, they found DPR they'll come find you, it's true.

czarnylab - August 30, 2021, 11:40 am

DPR had shitty OPSEC.

The site should be scrapping exif, that's terrible. This should be a common sense for those marketplaces.

That's easy like:

Code pip install exif_delete
and

```
Code python exif_delete.py /path/to/my/image.jpg
```

or

```
Code # Uses the Python Imaging Library
```

```
# `pip install Pillow` works too
```

```
from PIL import Image
```

```
image_filename = "picture_with_EXIF.jpg"
```

```
image_file = open('image_filename')
```

```
image = Image.open(image_file)
```

```
# next 3 lines strip exif
```

```
image_data = list(image.getdata())
```

```
image_without_exif = Image.new(image.mode, image.size)
```

```
image_without_exif.putdata(image_data)
```

```
image_without_exif.save(u"clean_{}".format(image_filename))
```

guardian - August 30, 2021, 11:44 am

(Aug 30, 2021 11:40 AM)czarnylab Wrote: [\[CITE\]](#) DPR had shitty OPSEC.

The site should be scrapping exif, that's terrible. This should be a common sense for those marketplaces.

I would even say that people that don't delete their exif data should not be allowed to be vendors on those marketplaces. Personally wouldn't rely on them being able to package the shipment stealthy enough.

BigFruity - September 2, 2021, 9:09 am

(Aug 30, 2021 11:40 AM)czarnylab Wrote: [\[CITE\]](#) DPR had shitty OPSEC.

The site should be scrapping exif, that's terrible. This should be a common sense for those marketplaces.

That's easy like:

```
Code pip install exif_delete
```

```
and
```

```
Code python exif_delete.py /path/to/my/image.jpg
```

Thank you for that, I learned something new in the way of removing EXIF from images.

When this code was written I shit myself at what was pulled from the image. I thought that the market would have stripped this data itself.

I had intended to contact admins to alert them because their rules are explicit for vendors, but they make contact hard for obvious reasons.

I do agree that all images should be auto sanitized, as in this should be a seamless function that the vendor doesn't see. But then again, too much hand holding might lead to a comfort level that results in getting V&.