SVEUČILIŠTE U SPLITU
FAKULTET ELEKTROTEHNIKE, STROJARSTVA I BRODOGRADNJE

IZVJEŠTAJ

LABORATORIJSKA VJEŽBA 4

Metapodaci datoteka

Danica Majić

Split, lipanj 2022.

U sklopu današnje vježbe analizirat će se metapodaci koji su prisutni u datotekama, a koje mogu dosta otkriti informacija o samoj datoteci te pomoći pri forenzičnoj analizi. Primjerice, ukoliko su fotografije slikane mobitelom, mogu sadržavati informacije o GPS koordinatama na kojoj je kreirana, model mobitela koji je bio korišten, rezolucija, ekspozicija itd.

1. ZADATAK

U direktoriju Download se nalaze dvije datoteke, jedna je tipa PDF, druga JPG datoteka. PDF dokument je enkriptiran, te je vaš zadatak dekriptirati PDF dokument, te izvući sve metapodatke iz njega, kao i metapodatke iz JPG datoteke.

> HINT: Lozinka kojom je PDF dokument enkriptiran odgovara sadržaju u ruci osobe koja se nalazi na Google Street View na GPS koordinati koja je sadržana u JPG datoteci.

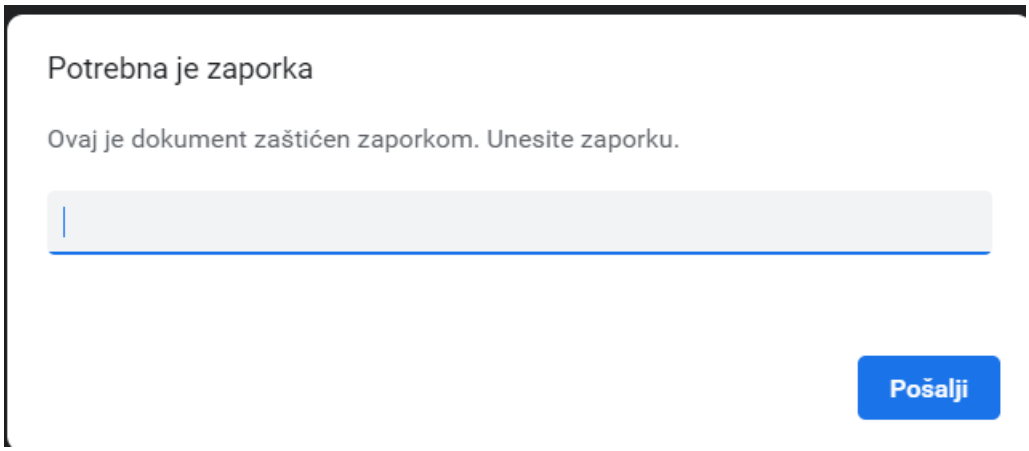
Za realizaciju vježbe koristite python programsko okruženje te ekstendirajte kod koji je naveden u nastavku. Možete pristupiti Google maps alatu iz pythona korištenjema biblioteke webbrowser, odnosno pozivom webbrowser.open_new_tab(url), gdje je url link na Google map koordinatu http://www.google.com/maps/place/**lat,long**. Pri tome su varijable lat i long rezultat konverzije GPS koordinate koji se može dobiti pozivom funkcije convertGPScoordinate.

```
import os, sys, optparse
from exif import Image
import webbrowser
from PyPDF2 import PdfFileReader, PdfFileWriter

def convertGPScoordinate(coordinate, coordinate_ref):
    decimal_degrees = coordinate[0] + \
                      coordinate[1] / 60 + \
                      coordinate[2] / 3600

    if coordinate_ref == "S" or coordinate_ref == "W":
        decimal_degrees = -decimal_degrees

    return decimal_degrees

def figMetaData(file_path):
    img_doc = Image(open(file_path, "rb"))
```

```
    if not img_doc.has_exif:
        sys.exit(f"Image does not contain EXIF data.")
    else:
        print(f"Image contains EXIF (version {img_doc.exif_version}) data.")

    print(f"{dir(img_doc)}\n")


def pdfMetaData(file_path):
    pdf_doc = PdfFileReader(open(path, "rb"))
    if pdf_doc.isEncrypted:
        try:
            if pdf_doc.decrypt("PASSWORD_GOES_HERE") != 1:
                sys.exit("target pdf document is encrypted")
        except:
            sys.exit("target pdf document is encrypted")

    pdfWriter = PdfFileWriter()
    for pageNum in range(pdf_doc.numPages):
        pdfWriter.addPage(pdf_doc.getPage(pageNum))
    resultPdf = open('decrypted_output.pdf', 'wb')
    pdfWriter.write(resultPdf)
    resultPdf.close()


if __name__ == "__main__":
    parser = optparse.OptionParser("Usage: python <script_name> -f <file>")
    parser.add_option("-f", dest="file", type="string", help="please provide full
path to the document")

    (options, args) = parser.parse_args()

    path = options.file
    if not path:
        print("please provide full path to the document")
        sys.exit(parser.usage)

    if any(path.endswith(ext) for ext in (".jpg", ".bmp", ".jpeg",)):
        figMetaData(path)
    elif path.endswith(".pdf"):
        pdfMetaData(path)
    else:
        print("File extension not supported/recognized... Make sure the file has
the correct extension...")
```

Python skriptu pozivate python <script_name> -f <file>, gdje je <script_name> ime .py skripte, a <file> ime PDF ili JPG datoteke.

REZULTAT:



Lozinka je banana.

Ovo je lazni dokument, nemojte ga gledati

```
PS C:\Users\DanicaMajić\Downloads\lab4\lab4> python zad.py -f hotel.jpeg
Image contains EXIF (version 0221) data.
48.37325
-123.58688888888888
['_exif_ifd_pointer', '_gps_ifd_pointer', '_segments', 'aperture_value', 'color_space', 'components_configuration', 'custom_re
ndered', 'datetime', 'datetime_digitized', 'datetime_original', 'delete', 'delete_all', 'exif_version', 'exposure_mode', 'expo
sure_program', 'exposure_time', 'f_number', 'flash', 'flashpix_version', 'focal_length', 'get', 'get_all', 'get_file', 'get_th
umbnail', 'gps_altitude', 'gps_altitude_ref', 'gps_datestamp', 'gps_img_direction', 'gps_img_direction_ref', 'gps_latitude', '
gps_latitude_ref', 'gps_longitude', 'gps_longitude_ref', 'gps_timestamp', 'has_exif', 'list_all', 'make', 'metering_mode', 'mo
del', 'orientation', 'photographic_sensitivity', 'pixel_x_dimension', 'pixel_y_dimension', 'resolution_unit', 'scene_capture_t
ype', 'sensing_method', 'shutter_speed_value', 'software', 'white_balance', 'x_resolution', 'y_resolution']

PS C:\Users\DanicaMajić\Downloads\lab4\lab4>
```

2. ZADATAK

Cilj ove vježbe je pokazati kako dvije datoteke kreirane u na različitim uređajima imaju iste hash otiske ukoliko je njihov sadržaj identičan. Također ćemo pokazati iako je sadržaj datoteke identičan (razlikuje se po kapitalizaciji) hash otisak će u tom slučaju biti isti.

- U Notepad-u kreirajte *text* dokument
- U dokument upišite vrijednost test te sačuvajte datoteku pod nazivom test.txt
- Kreirajte novi dokument te u njega upišite vrijednost Test te ga sačuvajte pod nazivom test1.txt
- korištenjem Python programa izračunajte hash vrijednosti navedenih datoteka
- Trebali biste dobiti ove rezultate:

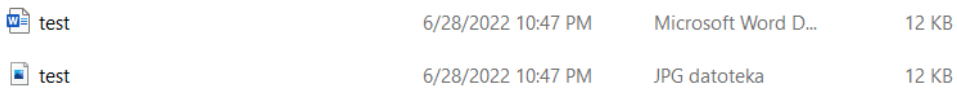
```
--Test.txt--
MD5: 098f6bcd4621d373cade4e832627b4f6
SHA1: a94a8fe5ccb19ba61c4c0873d391e987982fbbd3
SHA256: 9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08

--Test1.txt--
MD5: 0cbc6611f5540bd0809a388dc95a615b
SHA1: 640ab2bae07bedc4c163f679a746f7ab7fb5d1fa
SHA256: 532eaabd9574880dbf76b9b8cc00832c20a6ec113d682299550d7a6e0f345e25
```

Dobiveni rezultati:

```
PS C:\Users\DanicaMajić\Downloads\lab2(1)\lab2> python zad2.py
---------zad2\test.txt--------
SHA 256
9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08
MD 5
098f6bcd4621d373cade4e832627b4f6
SHA 1
a94a8fe5ccb19ba61c4c0873d391e987982fbbd3
---------zad2\test1.txt--------
SHA 256
532eaabd9574880dbf76b9b8cc00832c20a6ec113d682299550d7a6e0f345e25
MD 5
0cbc6611f5540bd0809a388dc95a615b
SHA 1
640ab2bae07bedc4c163f679a746f7ab7fb5d1fa
PS C:\Users\DanicaMajić\Downloads\lab2(1)\lab2>
```

3. ZADATAK

a) Kreirajte u **Word** programu datoteku te u nju upišite neki sadržaj. Sačuvajte dokument u ekstenziji .docx pod nazivom test na računalu (test.docx). Nakon toga, napravite kopiju word dokumenta te joj promijenite naziv i ekstenziju tako da ime bude identično originalnom dokumentu, dok joj je ekstenzija jednaka ekstenziji slike .jpg (test.jpg). Hoće li hash otisak (**MD5 i SHA1**) obaju dokumenata biti isti.



RIJEŠENJE: Hash otisak (MD5 I SHA1) su isti.

```
PS C:\Users\DanicaMajić\Downloads\lab2(1)\lab2> python zad3.py
---------files\test.docx--------
MD 5
29e29e66fc9636598d28d97ed46c7c1c
SHA 1
78671f9029405fa7956f4a9fedd93b66422f468e
---------files\test.jpg--------
MD 5
29e29e66fc9636598d28d97ed46c7c1c
SHA 1
78671f9029405fa7956f4a9fedd93b66422f468e
```

b) Pretpostavimo da je tvrtka prijavila problem korporativne špijunaže u kojem smatraju

da im je ukraden/kopiran tekstualni dokument u PDF-u iznimne važnosti. Budući da tvrtka ne želi otkriti sadržaj dokumenta, forenzični istražitelj dobiva u uvid hash otisak dokumenta:

c15e32d27635f248c1c8b66bb012850e5b342119

```
Found: Dokaz\Secret_file_52.jpg
c15e32d27635f248c1c8b66bb012850e5b342119
c15e32d27635f248c1c8b66bb012850e5b342119
```

Također, sa računala osumnjičene osobe ste izuzeli niz dokumenata koji bi mogli ukazivati na potencijalni dokaz. Dokumente u datoteci Dokaz.zip možete preuzeti iz direktorija Download. Raspakirajte dokumente i napravite analizu te navedite o kojem se dokumentu radi.

KOD:

```python
import glob
import hashlib

BLOCK_SIZE = 65536

filenames=glob.glob('files/test.*', recursive = True)
for filename in filenames:
    print("----------"+filename+"--------")
    sha256_hash = hashlib.sha256()
    md5_hash = hashlib.md5()
    sha1_hash=hashlib.sha1()
    with open(filename, 'rb') as f:
        fb = f.read(BLOCK_SIZE)
        while len(fb) > 0:
            sha1_hash.update(fb)
            sha256_hash.update(fb)
            md5_hash.update(fb)
            fb = f.read(BLOCK_SIZE)
    print("MD 5")
    print (md5_hash.hexdigest())
    print("SHA 1")
    print (sha1_hash.hexdigest())
```

```
print("-----------------------------------------------")
filenames=glob.glob('Dokaz/*', recursive = True)
for filename in filenames:
    sha1_hash=hashlib.sha1()
    with open(filename, 'rb') as f:
        fb = f.read(BLOCK_SIZE)
        while len(fb) > 0:
            sha1_hash.update(fb)
            fb = f.read(BLOCK_SIZE)
    if sha1_hash.hexdigest()=="c15e32d27635f248c1c8b66bb012850e5b342119":
        print(f"\nFound: {filename}")
        print (sha1_hash.hexdigest())
        print("c15e32d27635f248c1c8b66bb012850e5b342119")
```