Informatyka śledcza Laboratorium nr 2

Raport – Nikodem Jakubowski

Zadanie 1 – Zadanie 1 – Montowanie pliku .E01 jako nośnik pamięci przy wykorzystaniu pakietu EwfTools.

Utworzenie katalogu „mnt/tmp”, wykorzystanie „ewfmount” oraz „losetup”. Następnie użycie „mount” dla „/dev/loop0”.

```
┌──(root㉿user)-[/home/user]
└─# ewfmount /home/user/Desktop/USB_4GB_Kingston.E01 mnt/tmp
ewfmount 20140814


┌──(root㉿user)-[/home/user]
└─# mmls mnt/tmp/ewf1
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

     Slot    Start        End          Length       Description
000:  Meta    0000000000   0000000000   0000000001   Primary Table (#0)
001:  ------  0000000000   0000000127   0000000128   Unallocated
002:  000:000 0000000128   0007581695   0007581568   Win95 FAT32 (0x0c)

┌──(root㉿user)-[/home/user]
└─# ls -la mnt/tmp/ewf1
-r--r--r-- 1 root root 3881828352 Oct 30 17:28 mnt/tmp/ewf1

┌──(root㉿user)-[/home/user]
└─# losetup -r -o 65536 /dev/loop0 mnt/tmp/ewf1

┌──(root㉿user)-[/home/user]
└─# mount /dev/loop0 /media/kali
mount: /media/kali: WARNING: source write-protected, mounted read-only.
```

Użycie „df -k” ujawnia zamontowany obraz „pendrive’a”.

```
┌──(root㉿user)-[/home/user]
└─# df -k
Filesystem     1K-blocks     Used Available Use% Mounted on
udev              966700        0    966700   0% /dev
tmpfs             201428     1012    200416   1% /run
/dev/sda1       24640544 15885344   7478172  68% /
tmpfs            1007136        0   1007136   0% /dev/shm
tmpfs               5120        0      5120   0% /run/lock
tmpfs             201424      112    201312   1% /run/user/1000
/dev/loop0       3787056    34744   3752312   1% /media/kali
```

Zadanie 2 – Wykonaj analizę zdjęć znajdujących się w zamontowanym obrazie (USB DISK). Zmodyfikuj metadane znajdujące się w plikach jpeg.

```
┌──(root㉿user)-[/media]
└─# cd kali

┌──(root㉿user)-[/media/kali]
└─# ls -a
.  ..  .Spotlight-V100  1  IMG_5609.JPG  IMG_5627.JPG  IMG_5753.JPG  IMG_6002.JPG  IMG_8064.JPG  text2.rar
```

Wybieram zdjęcia: IMG_5609.JPG, IMG_5627.JPG, IMG_5753.JPG, IMG_6002.JPG.

Przykładowe użycie „exiftool”.

```
┌──(root㉿user)-[/media/kali]
└─# exiftool IMG_5609.JPG
ExifTool Version Number         : 12.65
File Name                       : IMG_5609.JPG
Directory                       : .
File Size                       : 5.6 MB
File Modification Date/Time     : 2021:07:10 15:12:50+02:00
File Access Date/Time           : 2021:10:03 02:00:00+02:00
File Inode Change Date/Time     : 2021:07:10 15:12:50+02:00
File Permissions                : -rwxr-xr-x
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
Exif Byte Order                 : Big-endian (Motorola, MM)
Make                            : Apple
Camera Model Name               : iPhone XS
Orientation                     : Rotate 90 CW
X Resolution                    : 72
Y Resolution                    : 72
```

IMG_5609.JPG

Rozmiar: 5.6 MB

Data utworzenia: 2021:07:10 13:12:49

Urządzenie: Apple iPhone XS

Orientacja: Obrócone o 90 stopni w prawo (Rotate 90 CW)

Wersja oprogramowania: 14.6

ISO: 200

Ustawienie światła: 1/60

Flash: Wyłączony

Rozdzielczość: 4032x3024

Przesłona: f/1.8

Lokalizacja: na podstawie GPS - wąwóz w Kazimierzy Dolnej.

Liczba obiektywów: Dwa obiektywy (iPhone XS back dual camera 4.25mm f/1.8)

IMG_5627.JPG

Rozmiar: 4.4 MB

Data utworzenia: 2021:07:10 13:16:54

Urządzenie: Apple iPhone XS

Orientacja: Obrócone o 90 stopni w prawo (Rotate 90 CW)

Wersja oprogramowania: 14.6

ISO: 200

Ustawienie światła: 1/60

Flash: Wyłączony

Rozdzielczość: 4032x3024

Przesłona: f/1.8

Lokalizacja: na podstawie GPS – wąwóz w Kazimierzy Dolnej.

Liczba obiektywów: Dwa obiektywy (iPhone XS back dual camera 4.25mm f/1.8)

IMG_5753.JPG

Rozmiar: 5.4 MB

Data utworzenia: 2021:07:18 17:31:52

Urządzenie: Apple iPhone XS

Orientacja: Pozioma (normalna)

Wersja oprogramowania: 14.6

ISO: 25

Ustawienie światła: 1/4274

Flash: Wyłączony

Rozdzielczość: 4032x3024

Przesłona: f/1.8

Lokalizacja: na podstawie GPS - Ogród Krasińskich Warszawa.

Liczba obiektywów: Dwa obiektywy (iPhone XS back dual camera 4.25mm f/1.8)

IMG_6002.JPG

Rozmiar: 2.6 MB

Data utworzenia: 2021:07:24 20:00:15

Urządzenie: Apple iPhone XS

Orientacja: Pozioma (normalna)

Wersja oprogramowania: 14.6

ISO: 64

Ustawienie światła: 1/121

Flash: Wyłączony

Rozdzielczość: 4032x3024

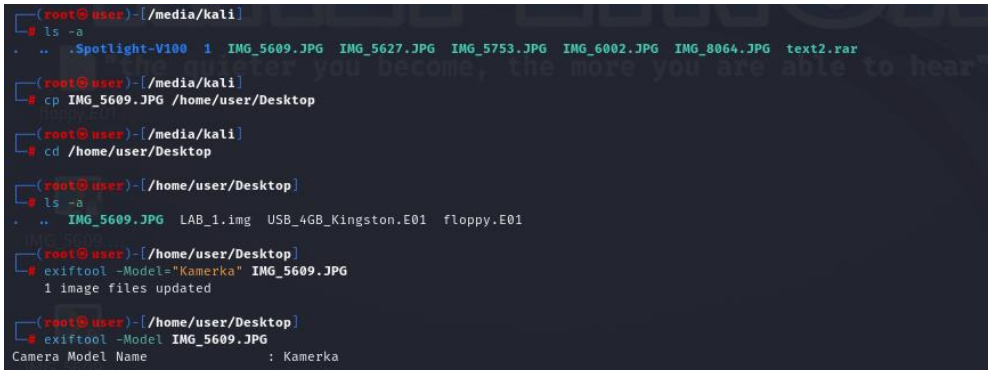
Przesłona: f/1.8

Lokalizacja: na podstawie GPS – Cypr.

Liczba obiektywów: Dwa obiektywy (iPhone XS back dual camera 4.25mm f/1.8)

Proszę o wybranie 5 dowolnych wartości oraz ich retusz.

Najpierw skopiowałem zdjęcie do innego folderu, następnie je modyfikowałem.



Zadanie 3 – Wykorzystując język programowania Python 3 sporządź prosty skrypt, który umożliwi wyświetlenie z konsoli linuxa podstawowych informacji z metadanych pliku jpg (np. czas wykonania zdjęcia).

Mój skrypt.

```
 GNU nano 7.2                                    img_reader.py
#!/usr/bin/env python

'''
Source: https://thepythoncode.com/article/extracting-image-metadata-in-python
'''

from __future__ import print_function
import argparse
from datetime import datetime as dt
import os
import sys

from PIL import Image
from PIL.ExifTags import TAGS

def get_jpg_metadata(file_path):
    try:
        with Image.open(file_path) as img:
            exif_data = img.getexif()
            if exif_data:
                print("Metadata for file:", file_path)
                for tag_id in exif_data:
                    # get the tag name, instead of a human-unreadable tag id
                    tag = TAGS.get(tag_id, tag_id)
                    data = exif_data.get(tag_id)
                    # decode bytes
                    if isinstance(data, bytes):
                        data = data.decode()
                    print(f"{tag:25}: {data}")
            else:
                print("No EXIF metadata found in the file.")
    except Exception as e:
        print(f"Error: {e}")

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description="Display basic metadata from a JPG file.")
    parser.add_argument("file_path", help="Path to the JPG file")
    args = parser.parse_args()

    get_jpg_metadata(args.file_path)
```

Przykładowe działanie.

```
┌──(root㉿user)-[/home/user/Desktop]
└─# ./img_reader.py IMG_5609.JPG
Metadata for file: IMG_5609.JPG
GPSInfo                   : 2156
ResolutionUnit            : 2
ExifOffset                : 268
ImageDescription          : Wakacje
Make                      : Apple
Model                     : Kamerka
Software                  : Psujemyyy
Orientation               : 6
DateTime                  : 2021:07:10 13:12:49
YCbCrPositioning          : 1
Copyright                 : Mojeee
XResolution               : 72.0
YResolution               : 72.0
HostComputer              : iPhone XS
```

Zadanie 4 – W trakcie analizy śledczej może pojawić się potrzeba przełamania zabezpieczenia w postaci hasła np. rar. Przy użyciu programu Rarcrack można obejść proste zabezpieczenia i pozyskać dane z archiwum.

```
┌──(root㉿user)-[/home/user/Desktop]
└─# rarcrack --type rar text2.rar
RarCrack! 0.2 by David Zoltan Kedves (kedazo@gmail.com)

INFO: the specified archive type: rar
INFO: cracking text2.rar, status file: text2.rar.xml
Probing: '3n' [89 pwds/sec]
Probing: '7L' [90 pwds/sec]
```

Później zmieniłem maskę w pliku konfiguracyjnym z rozszerzeniem „.xml”, licząc na to że hasło jest proste, żeby skrócić czas „brute force”.

```
  GNU nano 7.2                                          text2.rar.xml *
<?xml version="1.0" encoding="UTF-8"?>
<rarcrack>
  <abc>abcdefghijklmnopqrstuvwxyz</abc>
  <current>a</current>
  <good_password/>
</rarcrack>
```

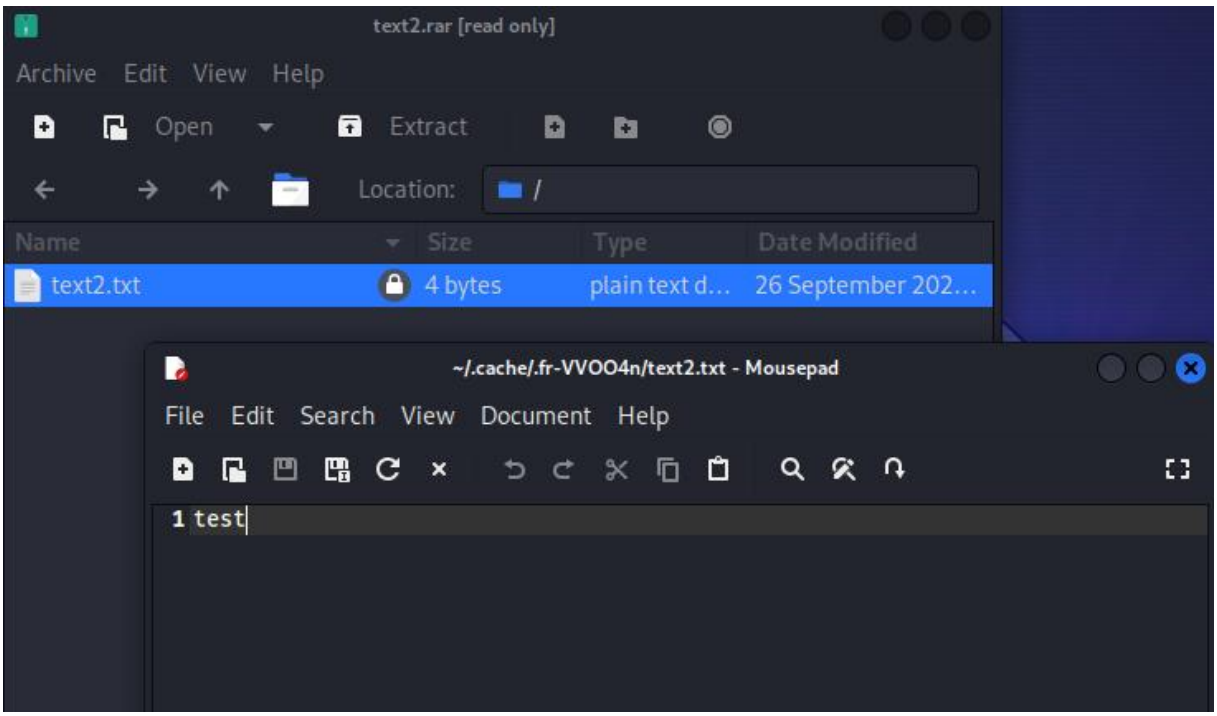
```
┌──(root㉿user)-[/home/user/Desktop]
└─# rarcrack --type rar --threads 32 text2.rar
RarCrack! 0.2 by David Zoltan Kedves (kedazo@gmail.com)

INFO: number of threads adjusted to 12
INFO: the specified archive type: rar
INFO: cracking text2.rar, status file: text2.rar.xml
INFO: Resuming cracking from password: 'byb'
Probing: 'cip' [87 pwds/sec]
Probing: 'csw' [89 pwds/sec]
Probing: 'ddb' [88 pwds/sec]
Probing: 'dnn' [90 pwds/sec]
Probing: 'dxr' [88 pwds/sec]
Probing: 'ehv' [88 pwds/sec]
Probing: 'esc' [89 pwds/sec]
Probing: 'fcj' [89 pwds/sec]
Probing: 'fmn' [88 pwds/sec]
Probing: 'fwv' [89 pwds/sec]
Probing: 'ghb' [88 pwds/sec]
Probing: 'gri' [89 pwds/sec]
Probing: 'gxc' [50 pwds/sec]
Probing: 'hhj' [89 pwds/sec]
Probing: 'hrp' [88 pwds/sec]
Probing: 'ibr' [87 pwds/sec]
```

Po kilku minutach, nie przyniosło to rezultatów, więc zacząłem się obawiać, że hasło może mieć związek z dużymi literami lub liczbami … Zresetowałem ustawienia, po dłuższej chwili okazało się, że hasło to: AGH.

```
Probing: 'zAV' [89 pwds/sec]
Probing: 'zFh' [90 pwds/sec]
Probing: 'zJC' [89 pwds/sec]
Probing: 'zNV' [89 pwds/sec]
Probing: 'zSh' [90 pwds/sec]
Probing: 'zWH' [91 pwds/sec]
Probing: 'A10' [89 pwds/sec]
Probing: 'A5n' [90 pwds/sec]
Probing: 'A9F' [88 pwds/sec]
Probing: 'Ae0' [89 pwds/sec]
Probing: 'Aik' [89 pwds/sec]
Probing: 'AmC' [88 pwds/sec]
Probing: 'AqV' [89 pwds/sec]
Probing: 'Avg' [89 pwds/sec]
Probing: 'Azu' [87 pwds/sec]
Probing: 'ADM' [88 pwds/sec]
GOOD: password cracked: 'AGH'
```

Zawartość pliku text2.txt.



Zadanie 5 – Odmontuj wirtualny nośnik /dev/loop0.

Odmontowanie obrazu dysku, sprawdzenie czy dalej figuruje w „df -k”.

```
┌──(root㉿user)-[/home/user]
└─# df -k
Filesystem     1K-blocks     Used Available Use% Mounted on
udev              966700        0    966700   0% /dev
tmpfs             201428     1012    200416   1% /run
/dev/sda1       24640544 15873324   7490192  68% /
tmpfs            1007136        0   1007136   0% /dev/shm
tmpfs               5120        0      5120   0% /run/lock
tmpfs             201424      112    201312   1% /run/user/1000
/dev/loop0       3787056    34744   3752312   1% /media/kali

┌──(root㉿user)-[/home/user]
└─# umount /dev/loop0

┌──(root㉿user)-[/home/user]
└─# df -k
Filesystem     1K-blocks     Used Available Use% Mounted on
udev              966700        0    966700   0% /dev
tmpfs             201428     1012    200416   1% /run
/dev/sda1       24640544 15873324   7490192  68% /
tmpfs            1007136        0   1007136   0% /dev/shm
tmpfs               5120        0      5120   0% /run/lock
tmpfs             201424      112    201312   1% /run/user/1000
```

Nie ma go, udało się odmontować. Potwierdzenie w „lsblk”. Nie ma żadnych MOUNTPOINTS.

```
┌──(root㉿user)-[/home/user/Desktop]
└─# lsblk -a
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0    7:0    0  3.6G  1 loop
```