

작성자	김경민
분석 일자	2024.05.24
작성 일자	2024.05.25
분석 대상	SM-F721N_Live.zip
문서 버전	2.0
작성자 E-mail	<a href="mailto:rlarudals877@gmail.com">rlarudals877@gmail.com</a>

0. 목차

1. 문제 .....3

2. 분석 도구 .....3

3. 환경 .....3


4. Write-Up.....4

5. Flag.....9

6. 별도 첨부 ..... 10

7. Reference ..... 13

### 1. 문제

URL	-
문제 내용	<p>Analyze provided Android live acquisition data and answer questions.</p> <p>1)What are the user's Google, YouTube, and Instagram account names? (30 points)</p> <p>2)What is the SSID and location (latitude, longitude coordinates) of the wireless network to which the evidence smartphone is connected? (30 points)</p> <p>3)Which photos taken with a smartphone have an edited EXIF timestamp? (30 points)</p> <p>4)Which photos uploaded to Instagram were not taken on the evidence smartphone? (30 points)</p> <p>5)What smartphone were the photo files found in question 4 taken on? (30 points)</p>
문제 파일	 SM-F721N_Live.zip
문제 유형	모바일 시스템 포렌식
난이도	2 / 3

### 2. 분석 도구

도구명	다운로드 링크	Version
Python	<a href="https://www.python.org/downloads/">https://www.python.org/downloads/</a>	3.12.0
DB Browser for SQLite	<a href="https://sqlitebrowser.org/">https://sqlitebrowser.org/</a>	3.12.2

### 3. 환경

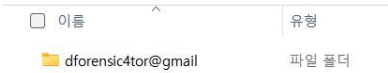
OS
Window 11 64-bit

## 4. Write-Up

파일명	SM-F721N_Live.zip
용량	3,799,608KB
SHA256	F2F4D387879E2CAF854DB8247C6D421B
Timestamp	2023-06-30 10:44:06

### 1. What are the user's Google, YouTube, and Instagram account names? (30points)

-WSM-F721N\_Live\data\com.google.android.apps.docs\app\_cello 라는 **구글 계정명으로 생성되는 폴더 경로**에서 계정명 "dforensic4tor@gmail.com" 확인할 수 있었다.



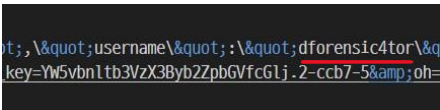
[사진 1] 구글 계정 확인

- WSM-F721N\_Live\data\com.google.android.youtube\shared\_prefs 경로에 있는 **youtube.xml**에서도 유튜브 계정을 확인할 수 있었다. 열어서 확인해보면 dforensic4tor@gmail.com 라는 유튜브 계정이 나와 있다.

```
name="com.google.android.libraries.youtube.innertube.cold_stored_timestamp"
value="1687474074054" />
<string name="offline_identity_nonce_mapping_DAxPt-x8oR8xcQq25DJQsg">
:KNQh5vQ-LxyRAW</string>
<long name="client_event_id_manager_client_count_identity_" value="16" />
<string name="user_account">dforensic4tor@gmail.com</string>
<string name="client_event_id_manager_event_id_for_identity_DAxPt-
8oR8xcQq25DJQsg">CPWxj-b61_8CFepG9QUds_8Kow%3D%3D</string>
<long
name="com.google.android.libraries.youtube.innertube.hot_stored_timestamp"
```

[사진 2] 유튜브 계정 확인

- 인스타그램 계정도 확인해 보면 data 파일에서 \com.instagram.android\shared\_prefs 경로에 있는 아티팩트 **com.instagram.android\_preferences.xml** 에서 찾을 수 있었다.



[사진 3] 인스타그램 계정 확인

[WHS-2] .iso

- 따라서 1번 문제에 대한 답은 밑에와 같다.

Google Account	YouTube Account	Instagram Account
dforensic4tor@gmail.com	dforensic4tor@gmail.com	dforensic4tor

**2. What is the SSID and location (latitude, longitude coordinates) of the wireless network to which the evidence smartphone is connected? (30points)**

- SSID는 기기가 특정 네트워크를 찾아 연결할 수 있게 해주는 역할을 한다. WSM-F721N\_Live\data\_backup\WABR\WIFICONFIG의 **semiconfigurations.json**, **WifiConfigStore.xml**을 열어서 확인해두면 SSID와 위치 정보 값을 알 수 있다.

```
[{"semwificonfig":{"configKey":"W\"JWMarriott
W\"OWE","networkScore":4,"location":{"latitude":1000,"longitude":1000}},{"configKey":"W\"JWMarrio
ttW\"NONE","networkScore":4,"location":{"latitude":8.1656823,"longitude":98.2952157}}]]
```

[사진 4] semiconfigurations.json에 있는 위치 정보

```
network={
  ssid="JWMarriott"
  key_mgmt=OWE
}
```

[사진 5] WifiConfigStore.xml에 있는 SSID 정보

- 따라서 2번 문제에 대한 해답은 밑에와 같다.

SSID	Location	
JWMarriott	Latitude - 8.1656823	Longitude – 98.2952157

**3. Which photos taken with a smartphone have an edited EXIF timestamp? (30points)**

- 이 부분은 파이썬 코드를 이용하여 DCIM -> Camera 안에 있는 사진들의 **EXIF의 타임 스탬프를 출력하고 생성시각과 편집된 시각을 비교**했다. 참고로 DCIM은 디지털 카메라 이미지의 약자로 스마트폰으로 찍은 사진은 거의 DCIM 폴더에 저장된다.

## [WHS-2] .iso

- 문제 풀이에 쓰인 **파이썬 코드**는 **[별도 첨부 1]**를 확인하면 된다. 이 코드는 DCIM -> Camera 디렉토리 내의 이미지 파일에서 EXIF 메타데이터를 추출하여 해당 이미지의 생성, 수정, 원본 날짜를 출력한다.
- `extract_exif_timestamps` 함수에서는 이미지 파일의 경로를 입력받아, EXIF 데이터를 로드하고 생성 날짜, 수정 날짜, 원본 날짜를 추출한다. 해당 데이터가 존재하면 문자열로 변환하고, 존재하지 않으면 'N/A'를 반환한다. `process_images` 함수에서는 주어진 디렉토리 내의 모든 .jpg 파일 경로를 가져와서 각 파일에 대해 `extract_exif_timestamps` 함수를 호출한다. 그리고 각 이미지 파일의 경로와 추출된 EXIF 날짜 정보를 출력한다.
- 출력된 타임 스탬프를 확인해본 결과, 20230607\_125642.jpg에서 **이미지 생성 시간이 편집 시간보다 느리다는 점**을 발견할 수 있었다. 이를 통해 이 파일이 문제에서 요구하는 EXIF가 편집된 파일임을 알 수 있었다.

```
File: D://forensic//DCIM//Camera\20230605_181351.jpg
Create Date (DateTimeDigitized): 2023:06:05 18:13:50
Date/Time Original: 2023:06:05 18:13:50
Modify Date: 2023:06:05 18:13:50

File: D://forensic//DCIM//Camera\20230607_125642.jpg
Create Date (DateTimeDigitized): 2023:06:07 12:56:42
Date/Time Original: 2023:06:06 12:56:00
Modify Date: 2023:06:06 12:56:00

File: D://forensic//DCIM//Camera\20230607_155850.jpg
Create Date (DateTimeDigitized): 2023:06:07 15:58:50
Date/Time Original: 2023:06:07 15:58:50
Modify Date: 2023:06:07 15:58:50
```

### [사진 6] 출력 결과 일부

- 따라서 3번 문제에 대한 해답은 20230607\_125642.jpg 이다.

## 4. Which photos uploaded to Instagram were not taken on the evidence smartphone? (30 points)

- 앞서 1번 문제에서 구한 인스타그램 계정을 실제로 들어가면 어떤 사진을 업로드 했는지 알 수 있다.



[사진 7] dforensic4tor의 게시물

- 문제를 해결하기 위해서는 위의 게시물 중 사용자의 핸드폰으로 찍지 않은 사진을 찾아야 한다. 먼저 사용자의 필터를 확인해주었다. Contents.db 파일을 열어보면 알 수 있다. **사용자의 필터는 SM-F721N**이고 폰 기종은 삼성이다.

	domain	key1	data1	key2	data2
	필터	필터	필터	필터	필터
1	Version	Code	33	Release	13
2	Make	Maker	SAMSUNG	Model	SM-F721N
3	AndroidID	ID	7542463adffcc856		
4	SerialNo	ID	R5CT82DUL7N		
5	Timezone	ID	Asia/Seoul		

[사진 8] 스마트폰 기종 확인하기

- 따라서 앞에서 나온 필터를 가지고 DCIM -> Camera 디렉토리에서 SM-F721N 필터로 찍지 않은 사진을 찾아주면 문제를 풀 수 있다. 여기서는 파이썬 스크립트를 이용했다. **파이썬 스크립트는 [별도 첨부 2]**에서 확인하면 된다.

- get\_exif\_data 함수에서는 이미지 파일의 경로를 입력받아, 해당 이미지의 EXIF 데이터를 추출한다. EXIF 데이터에서 'Model' 태그(카메라 모델)를 찾아 반환하고 없으면 None을 반환한다. find\_different\_camera 함수에서는 주어진 디렉토리 내의 모든 .jpg 파일 목록을 가져와서 각 파일에 대해 get\_exif\_data 함수를 호출하여 카메라 모델을 확인한다. 카메라 모델이 'SM-F721N'이 아닌 첫 번째 이미지를 찾으면, 해당 이미지의 경로와 카메라 모델을 반환하도록 했다. 모든 이미지가 'SM-F721N'으로 촬영된 경우 None을 반환하도록 했다.

- 코드를 실행한 결과 해당 스마트폰 모델로 촬영되지 않은 이미지 파일은 20230609\_042440.jpg이다.

```
[Running] python -u "d:\forensic\DCIM\Camera\SM-F721N.py"
i find the file that pictured other model:
file path: D://forensic//DCIM//Camera\20230609_042440.jpg
camera model: gnullSM-F721NnullF7

[Done] exited with code=0 in 0.955 seconds
```

[사진 9] 다른 카메라로 찍은 사진 파일 발견

- 이를 통해 4번 문제의 답은 20230609\_042440.jpg임을 알 수 있다.

5. What smartphone were the photo files found in question 4 taken on? (30 points)

- 이번에는 4번 문제에서 나온 파일이 무슨 핸드폰 모델로 촬영되었는지 확인해야 한다. 처음에는 앞선 문제와 같이 필터로 찾아 모델 정보를 확인하려고 했으나 자꾸 Null 오류가 생겨서 **렌즈 모델 정보를** 출력하여서 스마트폰 기종을 확인하기로 변경했다.
- 역시 파이썬을 이용해서 진행했고 **파이썬 스크립트는** **[별도 첨부 3]**에서 확인하면 된다. 로직은 4번 문제의 스크립트와 비슷하다.
- 코드를 실행한 결과 해당 사진을 찍은 스마트폰 모델은 iPhone 12 Pro임을 확인할 수 있었다.

```
[Running] python -u "d:\forensic\DCIM\Camera\othermodel.py"
#####: iPhone 12 Pro back triple camera 4.2mm f/1.6
i#####: gnullSM-F721NnullF7

[Done] exited with code=0 in 0.225 seconds
```

[사진 10] 렌즈 모델을 통한 스마트폰 기종 확인 결과

- 따라서 5번 문제의 답은 iPhone 12 Pro이다.



## 5. Flag

1.

Google Account	YouTube Account	Instagram Account
dforensic4tor@gmail.com	dforensic4tor@gmail.com	dforensic4tor

2.

SSID	Location	
JWMarriott	Latitude - 8.1656823	Longitude – 98.2952157

3.

20230607\_125642.jpg

4.

20230609\_042440.jpg

5.

iPhone 12 Pro

## 6. 별도 첨부

```
inspect EXIF.py > ...
1  from PIL import Image
2  import piexif
3  import glob
4
5  def extract_exif_timestamps(image_path):
6      img = Image.open(image_path)
7      try:
8          exif_data = piexif.load(img.info.get('exif', b''))
9          date_time_original = exif_data['Exif'].get(piexif.ExifIFD.DateTimeOriginal)
10         date_time_digitized = exif_data['Exif'].get(piexif.ExifIFD.DateTimeDigitized)
11         date_time_modified = exif_data['0th'].get(piexif.ImageIFD.DateTime)
12
13         date_time_original = date_time_original.decode('utf-8') if date_time_original else 'N/A'
14         date_time_digitized = date_time_digitized.decode('utf-8') if date_time_digitized else 'N/A'
15         date_time_modified = date_time_modified.decode('utf-8') if date_time_modified else 'N/A'
16
17         return date_time_original, date_time_digitized, date_time_modified
18     except Exception as e:
19         print(f"Error reading EXIF data from {image_path}: {e}")
20         return 'N/A', 'N/A', 'N/A'
21
22 def process_images(directory):
23     image_paths = glob.glob(f"{directory}/*.jpg")
24     for image_path in image_paths:
25         date_time_original, date_time_digitized, date_time_modified = extract_exif_timestamps(image_path)
26         print(f"File: {image_path}")
27         print(f"Create Date (DateTimeDigitized): {date_time_digitized}")
28         print(f"Date/Time Original: {date_time_original}")
29         print(f"Modify Date: {date_time_modified}")
30         print()
31
32 if __name__ == "__main__":
33     directory = "D://forensic//DCIM//Camera"
34     process_images(directory)
35
```

[별도 첨부 1] EXIF를 이용하여 생성시각과 수정된 시각을 출력하는 프로그램



```
SM-F721N.py > ...
from PIL import Image
from PIL.ExifTags import TAGS

def get_exif_data(image_path):

    image = Image.open(image_path)

    exif_data = image._getexif()

    if exif_data:
        for tag, value in exif_data.items():
            tag_name = TAGS.get(tag, tag)
            if tag_name == 'Model':
                return value
    return None

def find_different_camera(images_dir):
    import os
    image_files = [f for f in os.listdir(images_dir) if f.endswith('.jpg')]

    for image_file in image_files:
        image_path = os.path.join(images_dir, image_file)
        camera_model = get_exif_data(image_path)
        if camera_model != 'SM-F721N':
            return image_path, camera_model

    return None, None

images_directory = "D://forensic//DCIM//Camera"
different_image, different_camera_model = find_different_camera(images_directory)

if different_image:
    print("i find the file that pictured other model:")
    print("file path:", different_image)
    print("camera model:", different_camera_model)
else:
    print("other camera's model is 'SM-F721N'")
```

[별도 첨부 2] 지정된 필터 모델(SM-F721N)과 다른 모델로 촬영된 이미지를 찾는 프로그램



```
othermodel.py > get_exif_data
1 from PIL import Image
2 from PIL.ExifTags import TAGS
3
4 def get_exif_data(image_path):
5     image = Image.open(image_path)
6     exif_data = image._getexif()
7     if exif_data:
8         exif_info = {}
9         for tag, value in exif_data.items():
10             tag_name = TAGS.get(tag, tag)
11             if value is not None:
12                 exif_info[tag_name] = value
13         return exif_info
14     return None
15
16 image_path = "D://forensic//DCIM//Camera//20230609_042440.jpg"
17
18 exif_info = get_exif_data(image_path)
19
20 if exif_info:
21     lens_model = exif_info.get('LensModel')
22     camera_model = exif_info.get('Model')
23     if lens_model is not None:
24         print("렌즈 모델 정보:", lens_model)
25     else:
26         print("이미지 파일에서 렌즈 모델 정보를 찾을 수 없음")
27
28     if camera_model is not None:
29         print("카메라 모델 정보:", camera_model)
30     else:
31         print("이미지 파일에서 카메라 모델 정보를 찾을 수 없음")
32 else:
33     print("이미지 파일에서 Exif 데이터를 찾을 수 없음")
```

[별도 첨부 3] 이미지 파일에서 EXIF 데이터를 추출하여 렌즈와 필터 정보를 출력하는 프로그램

## 7. Reference

- [URL]