



작성자	김경민
분석 일자	2024.05.22
작성 일자	2024.05.23
분석 대상	Slack Off
문서 버전	1.0
작성자 E-mail	rlarudals877@gmail.com

0. 목차

1. 문제3

2. 분석 도구3

3. 환경3


4. Write-Up.....4

5. Flag..... 11

6. 별도 첨부 12

7. Reference 13

1. 문제

URL	
문제 내용	<p>Analyze the following through some data of the emulator extracted from the employee's PC.</p> <p>1.What are the names of mobile games installed by the user and their initial execution time? (40 points)</p> <p>2.What is the user’s account of Google Play and an IP address used to run the games? (40 points)</p> <p>3.Identify the payment history in the games and then list them in chronological order. (80 points)</p> <p>4.Check out the character ID, level, VIP level, and combat power of the game character played first by the user. (40 points)</p>
문제 파일	<div>  <p>Slack Off.zip</p> </div>
문제 유형	모바일 포렌식
난이도	3 / 3

2. 분석 도구

도구명	다운로드 링크	Version
FTK Imager	https://accessdata.com/product-download	4.7
DB Browser for SQLite	https://sqlitebrowser.org/	3.12.2
Python	https://www.python.org/downloads/	3.12.0

3. 환경

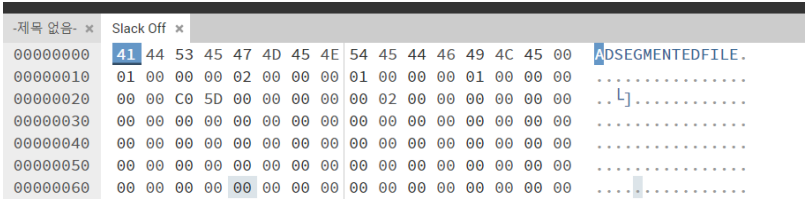
OS
Window 11 64bit

4. Write-Up

파일명	Slack Off
용량	424,932KB
SHA256	1DCBE63E9DCEAE8C92637DB3F6474F5D
Timestamp	2021-05-12 01:56:20

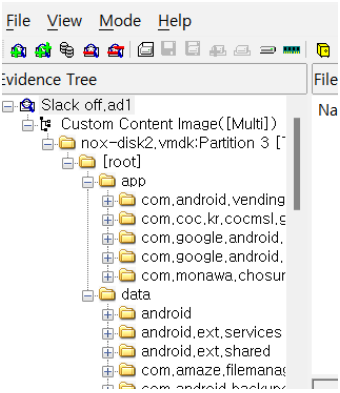
1. What are the names of mobile games installed by the user and their initial execution time?
(40 points)

- 일단 헥스로(<https://hexed.it>) 주어진 파일 확장자를 확인하였다.



[사진 1] 헥스로 Slack Off 파일 시그니처 확인하기

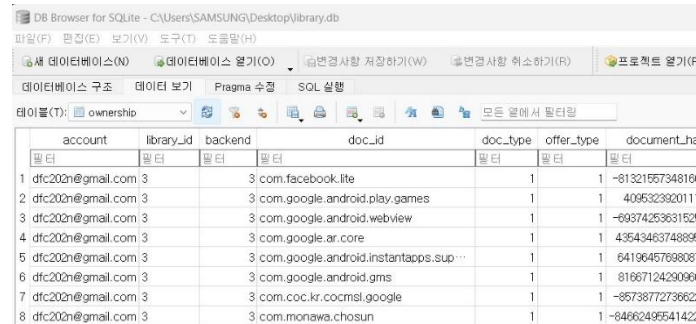
- 주어진 시그니처는 ADSEGMENTEDFILE 임으로 Access Data 에서 지원하는 File Format 으로 FTK Imager 를 통해 Dump 를 수행했을 때 확장자를 AD1 으로 진행할 경우 만들어지는 파일이다. 따라서 주어진 이미지에 확장자 .ad1 를 추가해 FTK Imager 에 업로드 후 분석을 시작했다. 폴더 디렉터리 구조를 확인해 보면 일반적으로 안드로이드에서 사용되는 android. ~ 폴더를 확인할 수 있으며 이를 통해 안드로이드 구조의 이미지임을 추측할 수 있다.



[사진 2] 주어진 파일 확장자 변경 및 FTK Imager에 업로드

[WHS-2] .iso

- 1번 문제는 사용자가 설치한 게임의 게임명과 처음으로 시작한 시간을 확인하는 문제이다. 이를 해결하기 위해 **WdataWcom.android.vendingWdatabasesWlibrary.db** 경로에 있는 파일을 확인했다.



	account	library_id	backend	doc_id	doc_type	offer_type	document_ha
1	dfc202n@gmail.com	3	3	com.facebook.lite	1	1	-81321557348160
2	dfc202n@gmail.com	3	3	com.google.android.play.games	1	1	4095323920111
3	dfc202n@gmail.com	3	3	com.google.android.webview	1	1	-69374253631525
4	dfc202n@gmail.com	3	3	com.google.ar.core	1	1	43543463748895
5	dfc202n@gmail.com	3	3	com.google.android.instantapps.sup...	1	1	64196457698087
6	dfc202n@gmail.com	3	3	com.google.android.gms	1	1	8166712429096
7	dfc202n@gmail.com	3	3	com.coc.kr.cocmsl.google	1	1	-8573877273662
8	dfc202n@gmail.com	3	3	com.monawa.chosun	1	1	-84662495541422

[사진 3] library.db로 게임명 확인하기

- 파일을 열었을 때 게임으로 볼 수 있는 doc_id를 확인하면 7번과 8번이었다. 이를 검색해보면 7행의 com.coc.kr.cocmsl.google은 멸망록: 14일간의 종말 MMORPG 이라는 게임명을 가지고 있으며 com.monawa.chosun 는 조선협객전M 이라는 게임명을 가지고 있다.
- 게임들의 최초 실행 시간을 확인하기 위해 **com.google.android.gms.measurement.prefs.xml** 파일을 열었다. 실행시간은 "first_open_time"에 나와있다.

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<map>
  <boolean name="allow_remote_dynamite" value="false" />
  <string
    name="gmp_app_id">1:78016880137:android:24fb08d8f1da30750362d8</string>
  <string name="app_instance_id">51fc59ab4de3b9e30ed55a73165de85c</string>
  <boolean name="has_been_opened" value="true" />
  <long name="first_open_time" value="1618235801203" />
  <boolean name="deferred_analytics_collection" value="false" />
  <boolean name="use_service" value="true" />
  <boolean name="app_backgrounded" value="false" />
  <long name="health_monitor:start" value="1619366436690" />
  <long name="last_pause_time" value="1619430160064" />
  <long name="last_upload" value="1618235801296" />
  <boolean name="start_new_session" value="true" />
  <string name="previous_os_version">7.1.2</string>
</map>
```

[사진 4] 멸망록 게임의 실행시간을 포함한 xml

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<map>
  <boolean name="allow_remote_dynamite" value="false" />
  <string
    name="gmp_app_id">1:752736922095:android:ee37843e821608e249ee1a</string>
  <string name="app_instance_id">46fa6b19cc4d3cb24f7e5ab31dc17044</string>
  <boolean name="has_been_opened" value="true" />
  <long name="first_open_time" value="1619430269268" />
  <boolean name="deferred_analytics_collection" value="false" />
  <boolean name="use_service" value="true" />
  <boolean name="use_dynamite_api" value="true" />
  <boolean name="app_backgrounded" value="false" />
  <long name="health_monitor:start" value="1619703955609" />
  <long name="last_pause_time" value="1619743948018" />
  <long name="last_upload" value="1619430269321" />
  <boolean name="start_new_session" value="false" />
  <string name="previous_os_version">7.1.2</string>
```

[사진 5] 조선협객전M의 실행시간을 포함한 xml

[WHS-2] .iso

- 이때 파이썬 코드를 이용해서 간단하게 실행시간을 확인할 수 있다. 주어진 밀리초 단위의 타임스탬프를 변수에 저장하고 그 변수를 초단위로 나눈 다음 타임스탬프를 UTC 시간대의 datetime 객체로 변환해서 시간대를 고려하여 GMT+09:00 으로 변환하고 날짜와 시간을 출력하도록 했다. GMT+09:00 으로 한 이유는 우리나라가 9시간 빠르기 때문이다.

```
D:\forensic > 2021 DFC > 204 - Slack Off 이거 > import datetime.py > ...
1 import pytz
2 import datetime
3
4 first_open_time = 1619430769268
5
6 first_open_time_sec = first_open_time / 1000
7
8 first_open_datetime_utc = datetime.datetime.fromtimestamp(first_open_time_sec)
9
10 jst = pytz.timezone('Asia/Tokyo')
11 first_open_datetime_gmt9 = first_open_datetime_utc.replace(tzinfo=pytz.utc).astimezone(jst)
12
13 print(first_open_datetime_gmt9.strftime('%Y년 %m월 %d일 %A %p %I:%M:%S %Z'))
14
```

[Done] exited with code=0 in 0.297 seconds

[Running] python -u "d:\forensic\2021 DFC\204 - Slack Off 이거\import datetime.py"
d:\forensic\2021 DFC\204 - Slack Off 이거\import datetime.py:10: DeprecationWarning: datetime.datetime.fromtimestamp() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.fromtimestamp(timestamp, datetime.UTC).
first_open_datetime_utc = datetime.datetime.fromtimestamp(first_open_time_sec)
2021년 04월 26일 Monday PM 06:44:29

[사진 6] 파이썬을 이용해서 시간 구하기

- 각 파일에 나와 있는 정보를 추출해보면 문제에 대한 해답은 밑에와 같다.

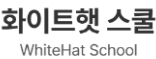
멸망록: 14일간의 종말 MMORPG	조선협객전M
2021년 4월 12일 월요일 오후 10:56:41 GMT+09:00	2021년 4월 26일 월요일 오후 6:44:29 GMT+09:00

2. What is the user’s account of Google Play and an IP address used to run the games? (40 points)

- 일단 사용자 계정부터 확인해보았다. 주어진 **localappstate.db** 파일 내 **appstate** 테이블중 **account** 칼럼을 확인해 보면 dfc202n@gmail.com 이라는 구글 계정을 확인할 수 있다.

3)	0	0	NULL	NULL
4)	0	0	NULL	NULL
5)	0	1618235240909	dfc202n@gmail.com	Google Play 서비스
6)	0	1619430224815	dfc202n@gmail.com	조선협객전M
7)	0	1618235486023	dfc202n@gmail.com	멸망록: 14일간의 종말 MMORPG
8)	0	0	dfc202n@gmail.com	Instant Apps

[사진 7] 구글계정 확인하기



- 이후 로그 데이터를 확인하여 IP주소를 확인하기 위하여 멸망록 게임의 로그를 확인했다. 그 결과, 사용자가 멸망록 게임을 하기 위해 이용한 계정과 IP 주소를 확인할 수 있었다. 멸망록 게임 로그에서 게임에 사용된 ip 주소는 218.50.143.55이다.

[illegible]

[사진 8] 게임에 사용된 ip주소

- 따라서 이 문제에 대한 해답은 밑에와 같다.

게임에 사용된 계정	별망록 게임에 사용된 ip주소
dfc202n@gmail.com	218.50.143.55

3. Identify the payment history in the games and then list them in chronological order. (80 points)

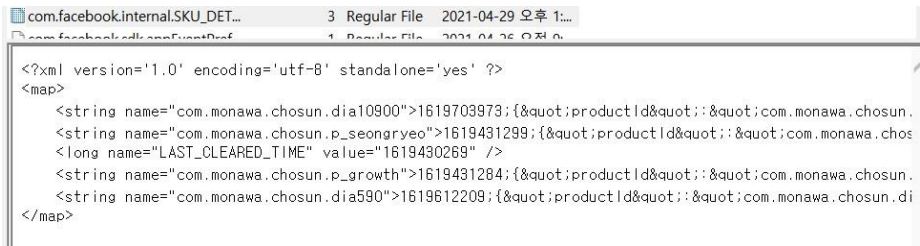
- 결제 내역을 확인하기 위해 결제 내역이 담겨 있는 파일처럼 보이는 **event_journal**을 열어서 확인해 보았다. 해당 이벤트는 어플리케이션 내부에서 상품을 구매했을 때 발생하는 "InAppPurchase" 이벤트로 사용자의 구매와 관련한 시간과 수량, 가격, 상품 정보 등이 기록되어 있음을 확인할 수 있었다. 따라서 InAppPurchase 이벤트가 일어난 행은 문제에서 말한 게임에서 구매한 상품과 그 정보가 포함되어 있을 가능성이 높다고 생각했다.

<pre>code": "AppLaunch", "page": {"ref": {"app_version": "com.moneta.chosun"}}, track_id": 2 code": "InAppPurchase", "page": {"ref": {"app_version": "0.12", "identifier": "com.mona code": "lnAppPurchase", "page": "@c449556", "price": "55000.0", "quantity": "1"}}, "t code": "Purchase", "page": {"ref": {"app_version": "0.12", "identifier": "com.mona code": "lnAppPurchase", "page": "@c449556", "price": "55000.0", "quantity": "1"}}, "t code": "Purchase", "page": {"ref": {"app_version": "0.12", "identifier": "com.mona code": "lnAppPurchase", "page": "@592b443", "price": "55000.0", "quantity": "1"}}, "t code": "Purchase", "page": {"ref": {"app_version": "0.12", "identifier": "com.mona code": "lnAppPurchase", "page": "@592b443", "price": "55000.0", "quantity": "1"}}, "t code": "Purchase", "page": {"ref": {"app_version": "0.12", "identifier": "com.mona</pre>	com.kakao.ad.sig 1619430269567 { " 1619430269568 { " 1619430269567 { " 1619430269568 { "
--	--

[그림 9] 왼쪽부터 InAPPPurchase 이벤트가 발생한 행, 그때의 가격과 수량, 구매 시각

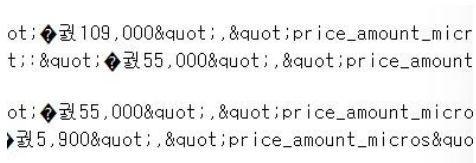
[WHS-2] .iso

- 이때 1619431285240, 1619431299999, 1619612210100, 1619703974755 때에 InAppPurchase 이벤트가 일어난 것을 알 수 있었다. 그러나 상품명을 알 수가 없어서 com.facebook.internal.SKU_DETAILS.xml 을 참고해서 **구매 목록과 상품의 가격을 확인했다.** 참고로 SKU 의 뜻은 소비자가 구매할 수 있는 제품 또는 서비스의 고유 식별자이다.



```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
  <string name="com.monawa.chosun.dia10900">1619703973;{"product_id":"com.monawa.chosun.dia10900","price_amount_micro":109000}</string>
  <string name="com.monawa.chosun.p_seongryeo">1619431299;{"product_id":"com.monawa.chosun.p_seongryeo","price_amount_micro":55000}</string>
  <long name="LAST_CLEARED_TIME" value="1619430269" />
  <string name="com.monawa.chosun.p_growth">1619431284;{"product_id":"com.monawa.chosun.p_growth","price_amount_micro":5900}</string>
  <string name="com.monawa.chosun.dia590">1619612209;{"product_id":"com.monawa.chosun.dia590","price_amount_micro":109000}</string>
</map>
```

[사진 10] 구매 목록 확인



```
ot: ₩109,000;{"product_id":"com.monawa.chosun.dia10900","price_amount_micro":109000}</string>
t;{"product_id":"com.monawa.chosun.p_seongryeo","price_amount_micro":55000}</string>
ot: ₩55,000;{"product_id":"com.monawa.chosun.p_growth","price_amount_micro":5900}</string>
₩5,900;{"product_id":"com.monawa.chosun.dia590","price_amount_micro":109000}</string>
```

[사진 11] 상품 가격 확인

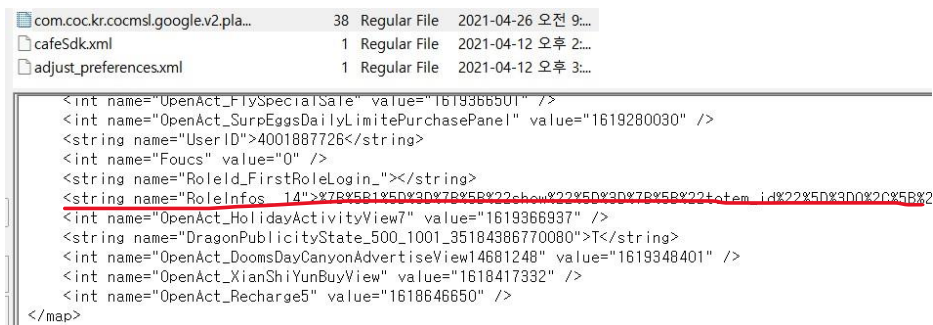
- 따라서 가격 비교를 통해 1619612210100 때는 dia590을 구매했고 1619703974755 때는 dia10900을 구매했다는 것을 알 수 있었다. 또한 가격이 같은 나머지 상품들은 event_journal에서 나온 구매 시각과 위의 xml 파일에서 나온 구매 시각을 비교해서 구분했다. 그 결과 1619431285240 때는 p_growth를 구매했다는 것을 확인했고 1619431299999 때는 p_seongryeo를 구매했다는 것을 확인할 수 있었다.

- 앞에서 설명한 파이썬 코드를 이용해 구매 날짜와 시간을 바꿔주었고 현재 파일은 한글이 깨져서 나와 직접 홈페이지에 들어가서 상품명을 확인해주었다. 그 결과, 이 문제에 대한 답은 밑에와 같다.

구매 시각	상품명	가격
2021-04-26 19:01:24	성장 패키지	₩ 55,000
2021-04-26 19:01:39	승려 장비 패키지	₩ 55,000
2021-04-28 21:16:49	590다이아몬드	₩ 5,900
2021-04-29 22:46:13	10900 다이아몬드	₩ 109,000

4. Check out the character ID, level, VIP level, and combat power of the game character played first by the user. (40 points)

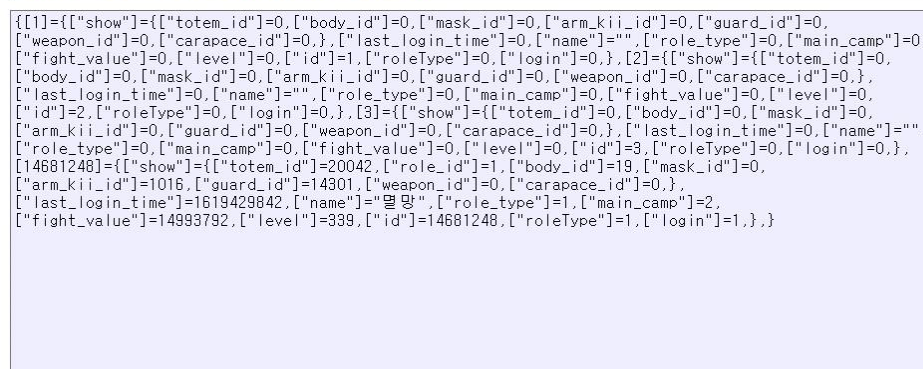
- 사용자의 캐릭터 정보를 확인하기 위해 **com.coc.kr.cocmsl.google.v2.playerprefs.xml** 파일을 열어보았다.



[사진 12] 캐릭터 정보 확인하기

- 이 파일에 대한 내용을 살펴보면 **RoleInfos**는 캐릭터에 대한 데이터가 저장되어 있고 이는 **WorldChatData_500_1001_14_14681248**와 더불어 URL Encoding 된 형태로 저장되어 있음을 확인할 수 있었다.
- 따라서 RolenInfo 부분을 <https://meyerweb.com/eric/tools/dencoder/> 을 이용해서 디코딩을 해주었다. 이를 통해 캐릭터의 아이디와(14681248), 캐릭터 이름(멸망), 레벨(339) 전투력(14993792)인 것을 알 수 있었다.

URL Decoder/Encoder

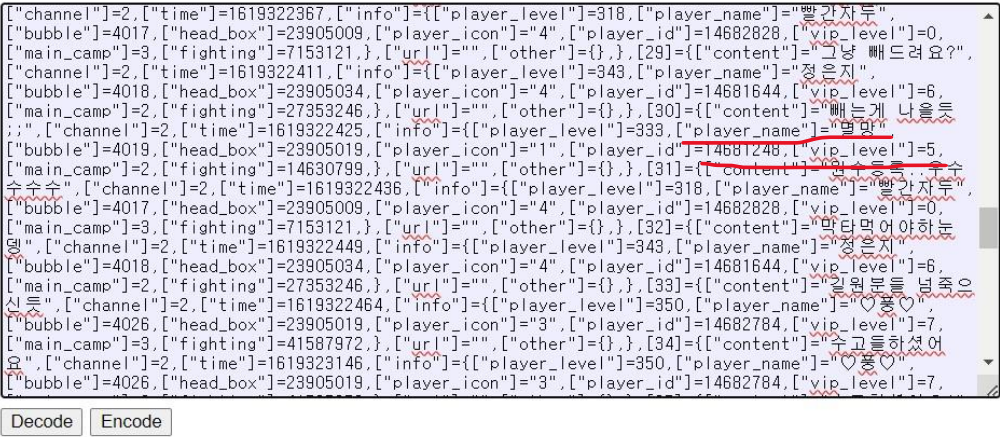


[사진 13] RolenInfo 디코딩 결과

[WHS-2] .iso

- 같은 방식으로 WorldChatData 부분도 디코딩 해주면 vip 레벨이 5임을 알 수 있다.

URL Decoder/Encoder



[사진 14] 디코딩 결과

- 따라서 이 문제에 대한 해답은 밑에와 같다.

아이디	이름	레벨	전투력	Vip 레벨
14681248	별망	339	14993792	5

5. Flag

1.

멀망록: 14일간의 종말 MMORPG	조선협객전M
2021년 4월 12일 월요일 오후 10:56:41 GMT+09:00	2021년 4월 26일 월요일 오후 6:44:29 GMT+09:00

2.

게임에 사용된 계정	멀망록 게임에 사용된 ip주소
dfc202n@gmail.com	218.50.143.55

3.

구매 시각	상품명	가격
2021-04-26 19:01:24	성장 패키지	₩ 55,000
2021-04-26 19:01:39	승려 장비 패키지	₩ 55,000
2021-04-28 21:16:49	590다이아몬드	₩ 5,900
2021-04-29 22:46:13	10900 다이아몬드	₩ 109,000

4.

아이디	이름	레벨	전투력	Vip 레벨
14681248	멀망	339	14993792	5

6. 별도 첨부

7. Reference

- [URL]