Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.24, NO.2, Apr. 2014 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2014.24.2.345

안드로이드 스마트 기기 내의 애플리케이션 업데이트 정보 자동 추출 시스템*

김 형 환,† 김 도 현, 박 정 흠, 이 상 진‡
고려대학교 정보보호연구원

The Automatic Extraction System of Application Update Information in Android Smart Device*

Hyounghwan Kim,† Dohyun Kim, Jungheum Park, Sangjin Lee‡
Center for Information Security Technologies, Korea University

요 약

스마트기기 활용도가 높아지면서 스마트기기의 다양한 애플리케이션들이 개발되고 있다. 이 애플리케이션들은 디지털 포렌식 조사 관점에서 사용자의 행동과 관련된 중요한 데이터가 존재할 수 있기 때문에 애플리케이션에 대해 사전 분석이 진행되어야 한다. 하지만 애플리케이션들이 업데이트되면서 분석된 데이터 포맷이 변경되거나 새로운 형태로 생성되는 경우가 빈번히 발생하고 있다. 그래서 사전 분석된 모든 애플리케이션에 대하여 업데이트가 진행되었는지 일일이 확인하여야하고, 업데이트가 진행되었다면 변경된 데이터가 있는지에 대해서도 분석이 꼭 필요하다. 하지만 분석된 데이터를 지속적으로 반복 확인하는 작업은 많은 시간이 소요되므로 이를 효율적으로 대응할 수 있는 방법이 필요하다.

본 논문에서는 애플리케이션의 정보를 수집하여 업데이트 정보를 확인하고, 변경된 데이터에 대한 정보를 효율적으로 확인할 수 있는 자동화된 시스템을 제안한다.

ABSTRACT

As the utilization rate of smart device increases, various applications for smart device have been developed. Since these applications can contain important data related to user behaviors in digital forensic perspective, the analysis of them should be conducted in advance. However, lots of applications get to have new data format or type when they are updated. Therefore, whether the applications are updated or not should be checked one by one, and if they are, whether their data are changed should be also analyzed. But observing application data repeatedly is a time-consuming task, and that is why the effective method for dealing with this problem is needed.

This paper suggests the automatic system which gets updated information and checks changed data by collecting application information.

Keywords: Digital Forensics, Smartphone Forensics, Android Forensics, Android Application, Android Data Acquisition

I. 서 론

스마트기기의 수요가 높아지면서 사용자가 사용할 수 있는 스마트기기 전용 애플리케이션(application)들의 수가 꾸준히 증가하고 있다.[1] 이러한 많은 애플리케이션들은 다양한 사용자 데이터를 주로 SQLite DB와 xml, plist 형태로 저장한다. 이 데이터에

접수일(2013년 12월 19일), 수정일(2014년 1월 15일), 게재확정일(2014년 1월 16일)
* 본 연구는 2013년도 정부(미래창조과학부)의 재원으로 한국연구재단-공공복지안전사업의 지원을 받아 수행되었습니다. [2012M3A2A1051106]
† 주저자, timemachine@korea.ac.kr
‡ 교신저자, sangjin@korea.ac.kr(Corresponding author)

Table 1. The progress table of the smart device application updates 2013 (a-android, i- iOS,)

month/weeks / app	6/3	6/4	7/1	7/2	7/3	7/4	7/5	8/1	8/2	8/3	8/4	9/1	9/2	9/3	9/4	10/1
Drobox		i		a	a				i						a	
Evernote	a	i	a	a		a	a	i		i	a	a			a	
ezPDF	a		a		a										a	
Facebook	i			a, i				a	i						a, i	a, i
Joyn			a, i	a	a							a, i	a			
Jorte	a	a		a	i	a	a					a, i		i		a
Kakao Talk			a	a			a		i		i	a		a		
Me2Day					i	a		a	i			a	a	a	i	
Mypeople		i				i	a			a	i				a, i	
Naver Line	a, i				a	a	a, i	i			a, i			a	a, i	a, i
Skype			a		a			a	i					a, i		
Twitter				a, i	i	a, i		a, i	i		a	a, i			i	i

는 사용자의 행동에 관한 내용이 포함되어 있을 수 있기 때문에 포렌식 조사 시에 반드시 분석해야 할 대상이다.

하지만 애플리케이션들의 수가 증가함에 따라 디지털 포렌식 분석에 소요되는 시간이 크게 증가하고 있으며 대상을 분석하였더라도 각각의 애플리케이션들이 빈번하게 업데이트를 진행하고, 그 과정에서 사용자의 데이터가 저장되는 SQLite Table의 스키마 정보 및 xml, plist 구조가 변경되거나 삭제 혹은 새롭게 생성되기도 한다. 따라서 업데이트된 애플리케이션의 변경사항에 대하여 지속적으로 확인을 하여야 하는데 사전 분석된 애플리케이션의 수가 많은 경우 많은 시간이 소요될 수 있다. 또한 업데이트 확인이 되었더라도 디지털 포렌식 조사에 필요한 데이터에 변경사항이 있는지 재차 분석을 하여야 한다. 따라서 이러한 일련의 과정들을 자동적으로 대응할 수 있는 효율적인 방법이 필요한 시점이다

본 논문에서는 이러한 분석 과정들을 자동으로 대응할 수 있도록 개발된 시스템을 제안한다. 이는 모바일 스마트기기 애플리케이션 디지털 포렌식 분석에 소요되는 시간을 상당히 줄일 수 있다는 점에서 의미가 있으며, 본 연구 결과는 더 많은 스마트기기 애플리케이션들을 분석할 때 유용하게 활용될 수 있을 것이다.

II. 스마트기기의 애플리케이션 분석

2.1 애플리케이션의 업데이트 현황

스마트기기의 활용도가 높아지면서 생활에 유용한 애플리케이션들이 대거 등장하고 있고, 2013년 8월 기준 50만개 이상의 애플리케이션들이 안드로이드 마켓에 등록되어 있다.[2] 이러한 많은 애플리케이션들 중에 디지털 포렌식 조사에 의미가 있는 애플리케이션을 선별하여 분석을 진행하여야 하고, 분석된 애플리케이션의 업데이트에 대하여 지속적으로 대응해야 한다. **Table 1은** 2013년 6월부터 10월까지 16주 동안 "Kakao Talk", "Drobox", "Facebook", "Twitter", "Joyn", "NaverLine", "Skype", "Evernote", "Jorte", "Me2Day", "Mypeople", "ezPDF"의 업데이트 상황을 나타내는데, 12개의 애플리케이션이 16주 동안의 조사 기간 동안 iOS는 40건, Android는 61건의 업데이트가 진행된 것을 확인할 수 있다. 이는 실제로 애플리케이션들의 업데

user_id	message
	Click here to defi
23063285	Given your manifest elements above,
10552435	Here is a blog post I wrote up after helpir

user_id	message
10552435	6nbW6qxgsnESSfWM32+SEQ==
10552435	anPA1W8c+OsDzzf4Teg74g==

Fig.1. Encrypted data structure

이트가 상당히 빈번하게 일어남을 확인할 수 있다.

이처럼 업데이트가 진행될 경우에는 데이터의 구조가 변경되거나 Fig.1과 같이 데이터 내용 자체가 암호화 되는 경우도 있다. 뿐만 아니라 기존의 파일에 데이터가 없어지고 새로운 파일에 데이터가 쌓이기도 한다. 버전이 업데이트 되었지만 실제로 분석대상 데이터 파일들은 아무런 변화가 없는 경우도 있다. 이처럼 빈번하게 업데이트되는 모든 애플리케이션의 변경된 부분을 확인하는 작업은 디지털 포렌식 관점에서 꼭 필요한 분석 업무이다.

2.2 안드로이드 OS에서 애플리케이션 관리체계

2.2.1 애플리케이션 관리

안드로이드 스마트기기에서는 애플리케이션을 시스템 운영에 사용되는 시스템 애플리케이션과 사용자가 직접 다운받아 설치한 사용자 애플리케이션으로 구별하여 관리하고 있고, 애플리케이션의 실행파일인 apk 파일과 apk 파일을 실행하였을 때 생성되는 data 파일들을 각각 다른 파티션의 폴더에서 저장하여 관리한다.

시스템 애플리케이션의 apk 파일은 시스템 파티션 영역인 "/system/app/"에 저장하고, 사용자 애플리케이션은 데이터 파티션 영역의 "/data/app/"에 저장하여 관리하고 있다. 시스템 파티션에 저장된 시스템 애플리케이션들은 임의로 변경할 수 없고, 기기가 포맷되었을 때 사용자 애플리케이션은 모두 삭제되지만 시스템 애플리케이션은 삭제되지 않는다.

이 애플리케이션들은 안드로이드 기기 내부에 공통으로 포함되어 있는 "packages.list", "packages.xml" 파일에 기록되어 관리되는데 설치된 목록들과 애플리케이션들의 상세한 정보들을 확인할 수 있다.

Table 2. The information packages.list

Value	Example
PackageName	com.kakao.talk
Flag	1 or 0
DataPath	/data/data/com.kakao.talk/
UserID	10104

2.2.1.1 packages.list

packages.list는 기기에 설치된 모든 사용자 애플리케이션의 설치 목록을 저장하고 있는 파일이다. 시스템 애플리케이션에 대한 설치 목록도 기록되어 있지만 모든 시스템 애플리케이션에 대한 기록은 되어있지 않기 때문에 사용자 애플리케이션에 대한 설치목록을 확인할 때 유용하며 "/data/system/" 혹은 "/dbdata/system/"에 저장되어 있다.

이 파일에는 Table 2와 같이 PackagesName과 Flag, DataPath,UserID값을 추출할 수 있다. Flag는 AndroidManifest.xml의 〈application〉 태그에서 수집된 정보로써 다양한 설정 값을 가지고 있다. 본 연구에서 이 Flag 값은 설치된 애플리케이션이 시스템 애플리케이션인지 다운로드한 사용자 애플리케이션이지를 구분하는데 사용된다. Datapath는 설치된 사용자 애플리케이션이 설치될 때 생성되는 SQLite, png, xml, os등의 데이터들이 저장되는 경로이다. UserID는 각 애플리이션들을 식별하는 고유번호이다.

2.2.1.2 packages.xml

packages.xml은 기기에 설치된 시스템 애플리케이션과 사용자 애플리케이션 전체에 대한 상세 정보가 수록되어 있으며, "/data/system/" 혹은 "/dbdata/system/"에 저장되어 있다. 이 파일에는 apk 파일의 경로, 패키지명, 애플리케이션과 관련된 시간정보 및 퍼미션정보를 포함한 다양한 정보가 저장되어 있고, 본 연구는 그 정보 중 디지털 포렌식 조사

Table 3. The information packages.xml

Value	Example
PackageName	com.google.android.location
apk path	/data/app/com.kakao.talk-1.apk
Flag	1 or 0
ft	141df09e670
if	141df09fbda
ut	142ce48facb
version code	125
permission	android.permission.WRITE_EXTERNAL_STORAGE

관점에서 유용하게 사용할 수 있는 PackageName, apk path, Flag, ft, if, ut, version code, permission 정보에 대해 설명한다.

packages.xml에 저장되어 있는 Flag, PackageName, UserID값은 packages.list에 저장된 값과 동일하며 packages.list에 존재하는 DataPath값은 유일하게 package.xml에 없는 정보이다. 그 외의 정보들을 **Table 3**에 정리하였다. **Table 3**의 apkpath는 설치된 애플리케이션의 실행파일인 apk 파일이 저장되는 경로이고, ft, it, ut는 각각 시간 정보를 나타내는 정보이다. ft는 "timestamp in hex format", it는 "timestamp in hex format of first time installation", 그리고 ut는 "timestamp in hex format of last update"로 정의할 수 있으며 애플리케이션의 사용시간과 설치시간 그리고 버전 업데이트 시간을 알 수 있다. 이 시간 정보는 16진수인 hex 값을 UnixTime으로 변환하여 정보를 확인할 수 있다. VersionCode는 정수로 이루어진 애플리케이션의 개발버전을 기록한 정보이다. permission은 각 애플리케이션들의 안드로이드 기기에 대한 퍼미션 정보를 저장한다.

III. 애플리케이션의 정보 수집

3.1 애플리케이션의 정보 수집 대상

안드로이드 스마트기기에 설치된 애플리케이션들은 일정 시간마다 해당하는 서버와 통신을 하여 현재의 버전을 체크하여 업데이트를 진행하거나 설치된 기기가 재부팅 될 때 설치된 애플리케이션 버전을 서버와 비교하여 업데이트의 유무를 사용자에게 알려준다. 따라서 지속적으로 파일을 추출할 수 있는 안드로이드 스마트기기를 안드로이드 업데이트 확인전용 서버로 지정하고, 컴퓨터와 연결하여 일정 시간마다 재부팅하여 애플리케이션의 업데이트에 관련된 파일을 추출하여 분석한다면 업데이트에 대해 자동으로 대응할 수 있게 된다. 여기서 업데이트와 관련된 파일은 packages.list 파일과 packages.xml 파일 그리고 build.prop 파일이다.

Table 4. The information extracted from build.prop

Value	Example
Device Model	SHV-E160S
Device Brand	samsung
Device Version	E160SKSJMH2
NetOperator	skt-kr

build.prop는 "/system/" 경로에 존재하는 스마트기기와 관련된 정보가 기록된 파일이다. 기기의 기종 혹은 안드로이드 버전에 따라 시스템 애플리케이션의 종류와 저장되는 데이터들의 경로가 다를 수 있기 때문에 다양한 애플리케이션의 정보를 수집하고 분류하기 위해선 기기 자체의 정보를 확인할 필요가 있다. build.prop에는 기기에 대한 장치정보 및 dalvik vm[3]의 설정 정보등 많은 정보가 기록되어 있다. 그 중에 본 연구에서는 **Table 4**와 같이 Device Model, Device Brand, Device Version, NetOperator의 정보만을 추출하는데, 기기의 모델명과 브랜드, 안드로이드의 버전 그리고 통신사 정보를 확인할 수 있다.

그리고 업데이트를 확인하기 위하여 수집되는 데이터를 저장하여 분류할 때 좀 더 직관적으로 분석할 수 있도록 "aapt.exe(android SDK를 설치하면 "android-sdk-window/platform-tools/" 폴더에 존재하는 파일. apk 파일의 다양한 정보를 출력해주는 tool)" 도구를 이용하여 애플리케이션 apk파일에서 추가적인 데이터를 추출할 수 있다. 데이터는 퍼미션 정보 및 패키지명등 다양한 내용이 수록되어 있지만 packages.list, packages.xml 과 중복된 내용은 제외하고 name, versionName, icon정보만 추출하여 저장한다. name은 en, ko, ca, ja, be 등 각국의 언어로 정의되어 있어, 해당하는 애플리케이션이 어떠한 작동을 하는지 직관적로 알 수 있다. icon은 애플리케이션의 대표 이미지(png) 파일이 저장되어 있는 경로를 나타낸다. versionName은 앞서 설명된 정수로 표현되는 개발버전인 version code를 텍스트 형태로 표현한 개발버전이다.

이렇게 수집된 데이터들은 모두 Server에 데이터베이스화 하여 저장한다.

3.2 애플리케이션 정보 수집 절차

스마트기기 내에 있는 packages.list와 packages.xml, build.prop, apk 파일을 일정시간 간격으로 추출하여 동일한 패키지명의 버전 정보를 비교해서, 업데이트 정보를 확인할 수 있다. 이 파일들은 adb(Android Debug Bridge)의 "adb

reboot", "adb pull" 명령어를 이용하여 파일을 추출할 수 있고, 그 프로세스는 다음과 같다.

① 안드로이드 기기와 컴퓨터 연결
② 기기와 연결된 PC에서 "adb reboot" 명령어로 재부팅
③ "adb pull" 명령어로 build.prop을 추출하여 기기정보 확인
④ "adb pull" 명령어로 packages.xml 추출하여 packagesName과 VersionCode를 비교하여 업데이트 확인
⑤ "adb pull" 명령어로 업데이트된 데이터의 apk 파일 및 애플리케이션 관련 데이터 추출
⑥ "aapt.exe" 명령어로 apk 파일 정보 확인 및 저장
⑦ 일정 시간 마다 반복

이렇게 총 7단계의 프로세스로 안드로이드 애플리케이션의 정보를 확인할 수 있는 파일들을 추출할 수 있고, 디지털 포렌식 조사 관점에서 **Table 5**의 항목과 같이 기기의 정보들과 기기에 설치된 애플리케이션들의 의미 있는 정보를 수집할 수 있다. 기기의 정보를 수집하는 이유는 서버로 사용되는 스마트기기가 변경되었을 경우에 기기내의 애플리케이션들을 분류하기 위함이다.

Table 5. The data collected

종류	수집항목
Device	기기 명
	상표
	버전
	통신사
Application	앱의 명
	패키지 명
	코드버전
	코드명
	Database경로(Data File)
	Apk 경로(apk File)
	Icon 경로
	플래그 값
	최초 설치시간
	마지막 업데이트시간
	사용시간
	안드로이드 사용권한

3.3 수집된 파일의 정보 분석

데이터 파티션 영역에서 수집된 파일들은 패키지 명으로된 폴더 아래 데이터들을 분류하여 저장되는데, 이 파일들 중 업데이트되어 데이터가 변경된 파일만을 구별하는 방법이 필요하다.

본 연구에서는 변경된 파일을 구별하기 위해 수집된 모든 파일에 대한 hash 값을 생성하여 업데이트에 대한 변경 사항을 확인한다.

특히 특정한 데이터 값이 변경되거나 암호화 되어 있는 xml과 같은 텍스트 파일에 유용하게 대응할 수 있는데, 그 이유는 수많은 텍스트 파일들을 일일이 확인하여 업데이트가 되었는지, 변경된 부분이 있는지 확인하는 작업은 실제로 많은 시간이 소요되고, 파일의 변경된 내용을 발견하지 못할 수 도 있다. 그리고 분석을 진행하였는데 변경된 내용이 없어 시간 낭비가 될 수 있기 때문에 hash값으로 변경된 파일만을 구별하는 시스템은 상당히 유용할 수 있다.

또한, 변경된 부분을 발견하지 못한 상태로, 다음 버전으로 업데이트가 진행 되는 경우 변경된 애플리케이션의 파일을 분석하는데 더 많은 시간이 소비 될 수 있기 때문에 이러한 시스템은 매우 효율적으로 사용 될 수 있다.

그리고 변경된 파일이 SQLite일 경우에는 좀 더 상세한 업데이트 정보를 확인할 수 있다. 그 방법은 모든 테이블과 그 테이블에 해당하는 컬럼 정보를 수집하고, 업데이트가 발생하였을 때 이전 버전의 데이터베이스 정보와 업데이트된 데이터베이스를 비교하면 어떤 데이터베이스의 어떠한 테이블이 변경되었다는 것을 알 수 있다.

IV. 애플리케이션 업데이트 분석 시스템

4.1 애플리케이션 업데이트 정보 자동 추출 시스템

AU(Application Update Analysis System)는 주기적으로 애플리케이션의 데이터를 수집하고, 버전별로 관리하며 수집된 데이터들을 가공하여 애플리케이션의 업데이트에 대한 상세정보를 사용자에게 제공하는 애플리케이션 업데이트 분석 시스템이다.

AU는 **Fig.2**의 프로세스로 동작한다. 스마트기기와 연결 후 AU를 구동시키면 Databases Server와

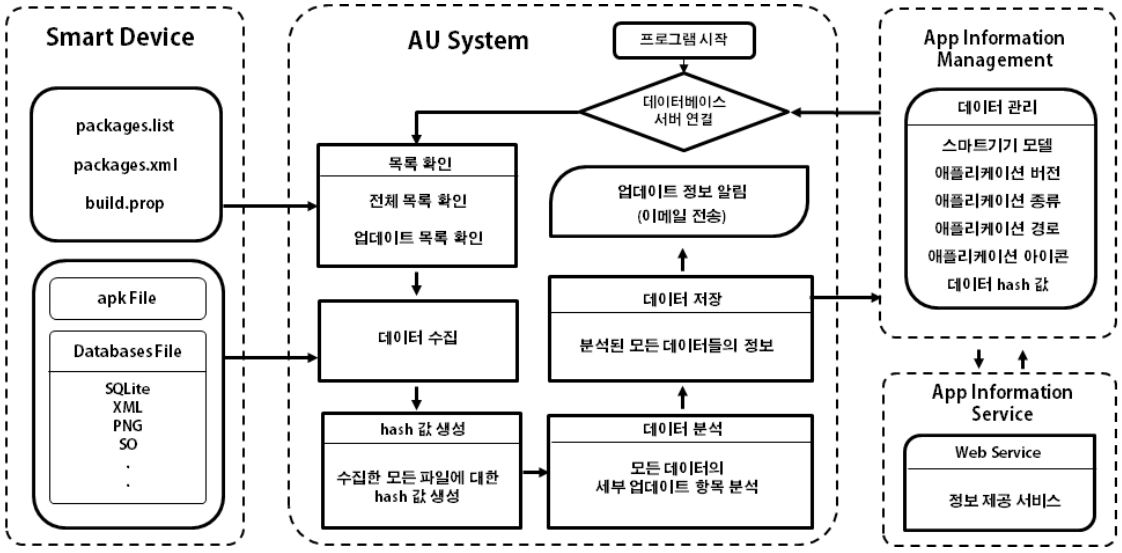

Fig. 2. Process of Application Update Analysis System

연동하여 기존에 분석된 애플리케이션들의 버전정보를 확인하고, 프로세스가 종료될 때 까지 섹션을 유지한다. 그리고 기기에 새로 설치된 애플리케이션 및 업데이트된 애플리케이션 목록을 확인하기 위해 기기에서 build.prop, package.list와 package.xml를 수집하고 분석한다. apk 파일과 apk파일을 실행시켰을 때 생성되는 데이터 파일들을 수집한다.

수집한 데이터들을 대상으로 hash값을 생성하고 비교하여 업데이트에 대한 상세한 정보를 확인할 수 있다. 또한, 애플리케이션 종류와 버전, 데이터베이스와 테이블을 선택하면 레코드 정보를 확인할 수 있고, SQLite Table의 실제 데이터도 확인이 가능하다.

이러한 일련의 과정에서 업데이트된 애플리케이션을 탐지하면 관리자에게 이메일을 전송하여 업데이트된 상세정보를 좀더 편리하게 확인할 수 있다.

4.2 애플리케이션 정보 관리

AU에서 생성된 애플리케이션의 정보는 기기의 모델, 애플리케이션의 종류, 애플리케이션의 버전별로 분류되어 앱의 명, 패키지 명, 코드버전, 코드 명, Database경로, Apk 경로, Icon 경로, 플래그 값, 최초 설치시간, 마지막 업데이트 시간, 사용시간, 안드로이드 사용 권한에 대한 상세 내용을 Database Server에 저장되어, AU에서 업데이트 정보를 요청하였을 때 관련된 정보를 제공한다.

또한 애플리케이션의 수집된 정보는 웹 서비스를 통해 외부에 제공하기도 한다. 현재 DFRC RDS 서비스의 모바일 항목으로 애플리케이션의 종류별 서비스가 되고 있으며, 상세 항목은 업데이트 될 예정이다.

4.3 애플리케이션 업데이트 정보 자동 추출 시스템 UI

애플리케이션 업데이트 정보 자동 추출 시스템인 AU의 구동하는 모습은 **Fig.3**과 같다. **Fig.3**의 ①은 분석중인 애플리케이션의 이름과 버전 그리고 아이콘을 나타낸다. ②는 설치된 전체 애플리케이션 중에 업데이트가 있는 애플리케이션을 검색하는 진행과정을 프로그래스바로 출력한다. ③은 업데이트된 애플리케이션의 기본 정보를 로그형태로 출력하고, 관리자의 이메일로 해당하는 업데이트의 상세정보를 전송한다. ④부터 ⑦까지는 각각 애플리케이션의 SQLite에 관련된 정보를 상세히 출력하며, 각각 애플리케이션의 이름, 버전, 데이터베이스명, 테이블명을 선택할 수 있도록 하여 원하는 데이터가 형식이 변했는지 관리자가 직접 분석 할 수 도 있다. ⑧은 해당 데이터베이스의 테이블 타입을 보여주며 ⑨는 해당 데이터베이스의 실제 데이터들을 출력하여 암호화의 유무 및 다양한 분석을 진행 할 수 있다.

V. 결 론

스마트기기의 사용이 많아지고 사용자가 사용하는 애플리케이션들의 종류가 많아지면서 디지털 포렌식

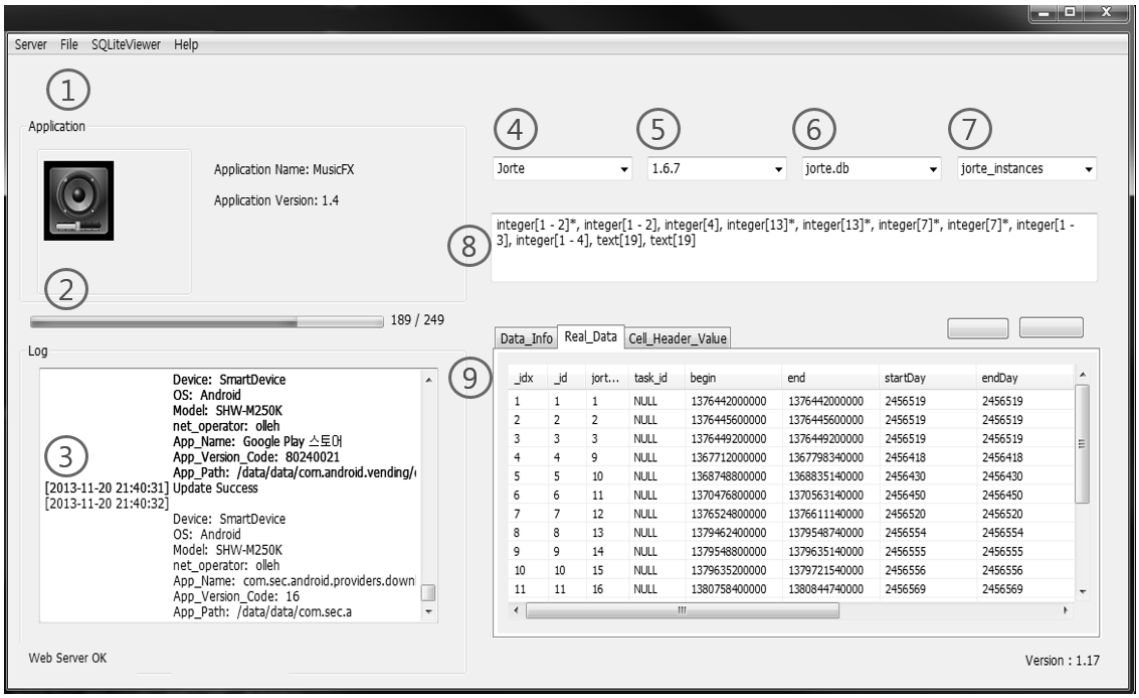

Fig. 3. Application Update Analysis System

관점에서 분석해야하는 애플리케이션도 증가하고 있다. 이러한 시점에서 스마트기기에 대한 디지털 포렌식 분석을 하고 있는 인원은 한정되어 있지만 분석해야하는 대상이 증가하고 있는 것은 매우 염려되는 상황이다. 또한 이미 분석된 대상이 업데이트되어 반복적으로 재분석을 진행하는 것은 매우 비효율적이지만 이러한 애플리케이션의 업데이트에 관련된 연구가 부족한 실정이다.

본 논문에서는 스마트기기에 설치된 애플리케이션의 업데이트를 감지하는 방법과 업데이트된 파일을 실시간으로 분석하고, 분석된 정보들을 데이터베이스화하여 업데이트에 대응하는 프로세스를 제안하였다.

특히, 수작업으로 업데이트에 대한 정보를 확인하고, 변경된 파일에 대응하던 작업을 모두 자동화된 시스템에서 처리하여, 관련된 정보를 이메일을 통한 알림시스템으로 관리자가 즉각 확인 할 수 있는 점은 디지털 포렌식 분석을 진행함에 있어 매우 편리한 기능이 될 것이며, 분석 시간도 상당히 단축시킬 수 있을 것으로 기대된다.

향후 분석되어 저장된 데이터베이스의 활용도를 더 향상한다면 스마트기기에 대한 디지털 포렌식을 진행함에 많은 도움을 줄 수 있을 것이다.

References

[1] M.Guido, "Automated identification of installed malicious Android applicatons," vol. 10, Digital Investigation, pp. 96-104, Oct. 2013

[2] http://www.appbrain.com/stats/android-market-app-categories

[3] Sohail Khan, "Analysis of Dalvik Virtual Machine and Class Path Library Constrained," Security Engineering Research Group, Institute of Management SciencesPeshawar, Pakistan, Nov. 2009

[4] Namheun Son, "A study of user data integrity during acquisition of Android devices," Digital Investigationm, vol. 10, pp, 3-11, Aug. 2013

[5] Timothy Vidas, "Toward a general collection methodology for Android devices," Digital Investigation, vol. 8, pp. 14-24, Aug. 2011

〔6〕 Steve Mead, "Unique File Identification in the National Software Reference Library," Digital Investigation, vol. 8, no. 3, pp. 138-150, Sep. 2006
〔7〕 Ralf D. Brown, "Reconstructing corrupt DEFLATEd files," Digital Investigation, vol. 8, pp. 125-131. Aug. 2011
〔8〕 Steve Mead, "Unique File Identification in the National Software Reference Library," Digital Investigation, vol. 3, No. 3, pp. 138-150, Sep. 2006
〔9〕 Jungheum Park and Hyunji Chung, Sangjin Lee, "Forensic analysis techniques for fragmented flash memory pages in smartphones," Digital Investigation, vol. 9, No. 2, pp. 109 - 118, Nov. 2012

〈저 자 소 개〉

김 형 환 (Hyounghwan Kim) 학생회원
2011년 2월: 경남대학교 e-비즈니스학과 학사
2011년 3월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 디지털 포렌식, 모바일 포렌식

김 도 현 (Dohyun Kim) 학생회원
2011년 2월: 서울과학기술대학교 정보통신대학 컴퓨터공학 공학사
2011년 3월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 디지털 포렌식, 모바일 포렌식, 파일 시스템

박 정 흠 (Jungheum Park) 학생회원
2007년 2월: 한양대학교 정보통신대학 컴퓨터전공 공학사
2007년 3월~2009년 2월: 고려대학교 정보경영공학전문대학원 공학석사
2009년 3월~현재: 고려대학교 정보보호대학원 박사과정
〈관심분야〉 디지털 포렌식, 안티-안티 포렌식

이 상 진 (Sangjin Lee) 종신회원
1987년 2월: 고려대학교 수학과 학사
1989년 2월: 고려대학교 수학과 석사
1994년 8월: 고려대학교 수학과 박사
1989년 10월~1999년 2월: ETRI 선임 연구원
1999년 3월~2001년 8월: 고려대학교 자연과학대학 조교수
2001년 9월~현재: 고려대학교 정보보호대학원 교수
2008년 3월~현재: 고려대학교 디지털포렌식연구센터 센터장
〈관심분야〉 디지털 포렌식, 심층 암호, 해쉬 함수