갤럭시 스마트폰의 데이터 분석 및 추출 방법

조영호 ㅇ 황지민 ㅇ 조진성

경희대학교 컴퓨터공학과
namespace@khu.ac.kr ㅇ jim1286@khu.ac.kr ㅇ chojs@khu.ac.kr

How to analyze and extract data from Samsung Galaxy smartphones

Youngho Cho ㅇ Jimin Hwang ㅇ JinSung Cho

Department of Computer Science and Engineering, KyungHee University

요 약

스마트폰은 현재 삶의 필수품이라고 볼 수 있을 정도로 많은 사람들의 소유물이 되었다. 다만 이러한 스마트폰은 개인정보를 가장 많이 닮고 있는 도구이며 이 때문에 범죄의 표적이 되기도 한다. 본 논문에서는 디지털 포렌식을 위한 갤럭시 스마트폰에 대한 데이터분석 및 수집 방법을 제안한다. 제안하는 방법은 갤럭시 스마트폰 포렌식의 기초 데이터를 만들고, 결론적으로 안드로이드 운영체제에서 데이터를 수집, 분석하는 포렌식 도구를 개발 방법론을 제시한다.

1. 서 론

스마트폰이 등장하고 나서 스마트폰은 개인 정보를 가장 많이 담고 있는 도구가 되었고 이 때문에 범죄의 표적이 되기도 한다. 이와 반대로 범죄 활동을 증명하는 수단으로 스마트폰을 분석하는 기술 중에 하나인 디지털 포렌식 기술이 활용되고 있다. 그 역할을 해줄 수 있게 해주는 기술을 디지털 포렌식이라고 한다. 디지털 포렌식은 디지털 증거물을 분석하여 수사에 활용하고, 디지털 증거물의 증거 능력을 향상하기 위한 과학 수사 기법을 총칭하는 용어이다. 포렌식 기술은 앞으로 점점 스마트폰을 이용한 범죄에 필수적인 기술로 자리 잡을 예정이며, 더 많은 연구가 요구되고 있다.

포렌식 도구의 개발에 대한 요구는 꾸준히 늘고 있으며, 데이터를 복구하기 위해서는 연구가 필요하다고 판단이 되었다. 따라서 안드로이드 운영체제에서 데이터를 수집, 분석하는 포렌식 도구를 개발하려고 한다.

본 논문에서는 대표적인 스마트폰인 안드로이드 갤럭시를 대상의 디지털 포렌식 기술을 제안한다. 제안하는 기술은 안드로이드 장치와 통신하게 해주는 ADB와 안드로이드 내부 데이터에 접근 가능하게 해주는 루팅이다.

"본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학 사업의 연구결과로 수행되었음"
(2017-0-00093)

2. 기존 연구

2.1 ADB(Android Debug Bridge, 이하 ADB)

안드로이드 장치와 통신하여 디버깅 등의 작업을 진행할 수 있는 Commend line tool 안드로이드 SDK에도 포함되어 있으며 애플리케이션 설치, 디바이스 접속 및 관리, 파일 앱/다운로드, 시스템 log 출력, shell 접속 등이 가능하다. 따라서 접근이 제한되어 있는 애플리케이션의 데이터에 대해 복원, 추출이 가능하다.

2.2 루팅(Rooting, 이하 루팅)

루팅은 모바일 기기에서 구동되는 안드로이드 운영 체제 상에서 최상위 권한을 얻음으로 해당 기기의 생산자 또는 판매자 측에서 걸어 놓은 제약을 해제하는 행위를 가리키는 말이다.[1]

이 루팅을 통해서 데이터에 접근이 가능해지기 때문에 갤럭시 폰에 대한 루팅이 필요하다.

3. 문제 정의 및 방법에 대한 분석

루팅을 하지 않은 핸드폰에 대해 접근할 수 있는 파일 시스템에 한계점이 있다. 가장 개인정보가 많이 저장되

어 있는 시스템에 접근하기 위해선 권한이 부족한 상태이다.

항 목	파일 경로
커널 정보	/data/log/recovery_kernel_log.txt
리커버리 로그	/data/log/recovery_log.txt
전원 종료 로그	/data/log/poweroff_info.txt
전원 재시작 로그	/data/log/powerreset_info.txt
전원 부팅 로그	/data/log/rtc.log
통화중 단절 로그	/data/log/CallDropInfoLog.txt
앱 에러 로그	/data/log/dumpstate_app_error.txt.gz
공유기 연결 로그	/data/misc/wifi/wpa_supplicant.conf
블루투스 정보	/data/misc/bluetoothd/config
앱 실행 정보	/data/system/dmappmgr.db
설치된 앱 정보	/data/system/packages.xml
등록된 계정 정보	/data/system/users/0/accounts.db
자동 로그인 계정 정보	/data/system/registered_services/android.accounts.AccountAuthenticator.xm
자동 동기화앱 목록	/data/system/registered_services/android.content.SyncAdapter.xml

그림 1 안드로이드 모바일기기의 시스템 데이터 제목 및 파일 경로

[그림 1]은 기존 연구 중 안드로이드 모바일 기기의 시스템 데이터 파일 경로를 나타내고 있다[2]. 안드로이드 버전이 올라감에 따라 내부 데이터 접근이 더욱 어려워졌으며, ADB 툴을 이용하여 데이터를 추출하는 데 한계가 있다.

본 장에서는 2가지 문제를 제시한다. 첫째는 루팅 없이 갤럭시 스마트폰에 접근할 수 있는 데이터는 한정되어 있다는 점이다. 두 번째 문제는 ADB를 통해 수집할 수 있는 데이터를 수집해본 결과, 실제 데이터에 접근할 수 없도록 암호화가 되어 있는 파일들이 존재한다는 점이다.

첫 번째 문제를 해결하기 위해 여러 가지 해결 방안을 고민해볼 수 있다. 루팅을 통해 루트 권한을 획득하는 방법, 물리적으로 메모리를 추출하여 파일시스템에 접근하는 방법, Smart Switch(삼성 스마트폰 백업 프로그램)의 데이터를 하이재킹하여 데이터를 수집하는 방법 등을 제시할 수 있다.

루팅을 통해 루트 권한을 획득할 경우 가장 쉽게 루트 권한을 취득할 수 있다는 장점이 있다. 다만 안드로이드 버전 업에 따라 루팅을 할 수 있는 커널이 업데이트 될 때까지 기다려야 한다는 단점이 있으며, 또한 추후 포렌식 진행 시 데이터의 무결성에 문제가 생길 수도 있다.

두 번째로 물리적으로 메모리를 추출하여 파일시스템에 접근하는 방법은 추가 작업이 필요 없이 메모리를 추출하면 되기 때문에 시간적으로 가장 빠르다는 장점이 있다. 다만 PUF(Physical Unclonable Function)이 적용되어 있는 갤럭시 스마트폰의 경우 물리적인 추출을 진행할 경우 데이터가 모두 손상될 수 있다는 단점이 있다.

마지막으로 Smart Switch(삼성 스마트폰 백업 프로그램)의 데이터를 하이재킹할 경우, 물리적, 논리적 손상 없이 데이터를 추출할 수 있다는 장점이 있다. 다만 실제로 하이재킹을 위한 연구를 추가 진행해야 한다는 단점이 있다.

아래 [표 1]은 스마트폰 데이터 접근 방안별 장단점을 정리한 표이다.

표 1. 스마트폰 데이터 접근 방안별 장단점

방안	장점	단점
루팅	쉽게 루트권한 취득 가능	추후 진행 연구의 데이터 무결성 위협
물리적 추출	시간적으로 가장 빠르며, 쉽게 파일시스템 접근 가능	PUF의 존재 유무에 따라 데이터 손상 가능
Smart Switch	물리적, 논리적 손상 없이 데이터 추출 가능	추가 연구 필요

두 번째 문제를 해결하기 위해 분석 대상 스마트폰을 추가 분석하여 실제로 얻어낼 수 있는 데이터들을 파악할 수 있다. 예를 들면 기본 인터넷 브라우저의 즐겨찾기 등의 사용자가 쉽게 접근할 수 있는 데이터를 기준으로 암호화되어 있는 데이터에 접근하여 추가 분석을 진행한 뒤 암호화되어 있는 데이터를 복호화 할 수 있다.

4. 데이터 분석 및 추출 결과

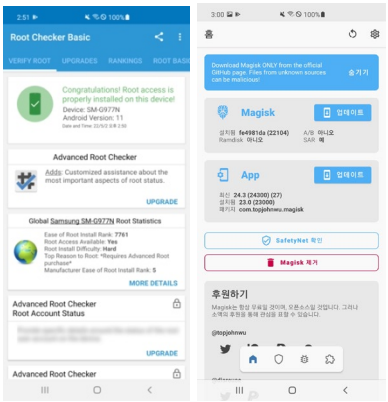

그림 2. 안드로이드 루팅 결과

Magisk를 활용하여 대상 기기를 루팅 하였다. 루팅은 Magisk를 활용하여 루팅 하였으며, Magisk의 경우 시스템 파티션은 그대로 두고 부트 파티션을 수정하여 작동한다.[3] 위 [그림 2]는 루팅 완료된 후 루트 권한을 획득한 모습이다.

루팅 후 데이터를 취득하기 위해 ADB Shell을 활용하여 데이터를 추출하였다. 추출 데이터 대상은 /data 및 /system의 하위 폴더 모두이다. 다만 ADB pull 명령어를 이용하여 한 번에 추출할 수 있는 파일의 한계가 있

어 파이썬 프로그램을 활용하여 자동화하였다. /system 폴더 이하 700개의 디렉토리, 5574개의 파일을 추출하였으며, /data 폴더 이하 12142개의 디렉토리, 23121개의 파일을 추출하였다.

추출된 주요 파일들에 대하여 로그 분석을 진행하였으며, 루트 권한을 획득한 후 접근할 수 있던 로그 파일들에는 별다른 암호화는 진행되어 있지 않았다. 분석 대상 로그파일은 사용자의 기록을 담고 있는 로그 파일을 위주로 분석하였으며, 이를 통해 사용자의 앱 사용 내역이나 동선 등을 추적할 수 있는 파일을 기준으로 선정하였다. 아래 [표 2]는 주요 파일과 어떤 데이터를 포함하고 있는지를 정리한 결과이다.

표 2. 주요 로그 파일 분석 결과

파일명	저장 데이터
power_off_reset_reason.txt	전원 관련 로그 저장
qtables.json	네트워크 접근 기록 저장
Iwc_dump.txt	WIFI 접근 기록 저장
subBuffer.log	블루투스 접근 기록 저장
CallContent.log	전화 기록 저장
Recovery_history.log	복구 기록 저장
Settingsprovider.txt	핸드폰의 세팅 변경 기록 저장
Shutdown_profile.n.txt (n은 1이상 숫자)	기기의 종료 기록 저장
Packages.xml	어플리케이션의 목록 및 각각의 권한 저장
n/Settings_config.xml (n은 0이상 숫자)	사용자별 기기 세팅 정보 저장

데이터베이스 파일의 분석을 진행하였다. 안드로이드는 SQLite DB를 활용하여 데이터베이스를 관리하고 있어 해당 파일들의 뷰 프로그램을 개발하여 분석을 진행하였다. 데이터베이스의 주요 파일들 또한 로그파일과 마찬가지로 사용자의 앱 사용내역이나 동선 등을 추적할 수 있는 파일을 기준으로 작성하였으며, 주요 데이터베이스 파일별 저장 데이터를 정리한 결과는 [표 3]과 같다.

표 3. 주요 데이터베이스 파일 분석 결과

파일명	저장 데이터
Audioservice_sec.db	어플리케이션의 오디오 권한 저장
ClipboardimageTable.db	클립보드에 있는 스크린샷 파일 목록 저장
Displaysolution_setting.db	화면 권한 설정 저장
Enterprise.db	기기의 전반적인 설정 저장
Gamemanager.db	게임 매니저에 관한 앱과 로그 저장
Locksettings.db	화면 잠금에 대한 정보 저장
Notification.db	알림들의 로그 저장
Pda.db	단말기 정보 저장
Pkgpredictions.db	패키지 정보 저장
Psitracker.db	Psi 측정정보 저장
Recoverablekeystore.db	클라우드 정보 저장
wifihistory.db	와이파이 사용 기록 저장

5. 결론 및 향후연구

연구 기기가 공기계였기 때문에 통화, 문자 등의 기능을 테스트하기 어려웠다. 또한 통신사의 유심(USIM)에 기록되는 데이터들도 있다고 기존 연구에서 보았지만 접근해보지는 못했다는 단점이 있다. 또한 기본 로그 이외의 어플리케이션에서 제공하는 데이터베이스 혹은 로그들은 암호화가 되어있어 접근할 수 없었다. 이번 연구를 통해 추출한 파일들을 자세하게 분석할 필요가 있다고 생각한다.

향후에는 파일시스템에 직접 접근하는 방법을 제시한 만큼 삭제된 파일들을 복구할 수 있는 방안 또한 추가할 예정이다. 안드로이드의 경우 바이너리 데이터를 활용하여 파일 카빙을 진행할 수 있다[4]. 파일 시스템이 안드로이드 버전에 따라 변경되기 때문에 최적화된 복구 방법론 모색이 필요하다.

참고 문헌

[1] 오정훈, 이상진. (2012). 안드로이드 스마트폰 포렌식 분석 방법에 관한 연구. 한국디지털포렌식학회 디지털 포렌식 연구 제9호

[2] 김도현, 이상진. (2016). 모바일 포렌식 동향. 한국정보보호학회논문지 제26권 제5호

[3] https://www.xda-developers.com/how-to-install-magisk/

[4] 방승규,전상준,김도현,이상진. (2016). HFS+ 저널 파일 파싱 알고리즘을 이용한 삭제된 파일 복구 기법 향상 방안. 정보처리학회논문지 p.463~p.470