선정 이유

모바일 포렌식을 처음 접해서 우선 가볍게 파악하기위해 해외논문보다 dbpia에서 인용수가 높은걸로 골라봄

내용 요약

기존에 연구된 모바일 포렌식 데이터 획득 기법은 안드로이드 7.0 누가(Android 7.0 Nougat) 이상의 버전에서 한계점이 존재

제조사 제공 기본 백업용 앱을 루팅없이 저장된 중요 데이터(메시지,연락처,통화기록 등) 획득하는 방법

하드웨어 기반 -> 칩오프(메모리 물리적으로), JTAG(인터페이스 논리적으로)

칩오프: PCB에서 플래시 메모리 분해 후 데이터 비트단위로 복제

-> 디스크가 암호화 되어있어 데이터 추출 어려움

JTAG:PCB에서 JTAG 인터페이스에 연결하여 플래시 메모리 데이터 획득

JTAG란 임베디드시스템 개발 시 사용하는 디버깅 장비

주로 RiffBox, Octopus Box 같은 툴을 사용해 연결

-> 최근 JTAG인터페이스를 숨기거나 제거하여 출시하거나 추가적으로 보안 기능 적용

소프트웨어 기반-> 논리적, 물리적

논리적: 활성 데이터 획득 기법, ADB Backup 기능에 기반한 획득 기법,Android Content Provider를 이용한 획득 기법, 백업 기능을 이용한 획득 기법 등

활성 데이터 획득 기법: 내부 저장소 데이터 단순 복사

ADB Backup 기반 획득 기법: 안드로이드 내 내장된 백업 기능 활용

백업 파일 생성 위해 adb backup -apk -shared -all –f TargetDevice.ab 명령 실행

Android Debug Extractor로 tar형식으로 변환 가능

디렉토리의 Android 폴더가 제외되어 일부 데이터만 백업 가능

Android Content Provider 기반 획득법: 안드로이드 앱 간 데이터 공유 기술을 이용해 데이터 획득

Content Provider를 사용해 데이터 공유

URI(Uniform Resource Identifier)을 사용해 접근 가능, 삭제된 데이터 획득은 불가능

PC용 백업 프로그램을 이용한 데이터 획득 기법: Samsung Kies 프로그램을 이용, 백업 파일 내부 데이터를 분석하여 타입별 시그니처 기반으로 데이터 카빙

현재는 암호화 적용으로 힘듦

물리적:루팅에 기반한 획득 기법, bootloader의 취약점을 이용한 기법, custom recovery image를 이용한 기법

루팅에 기반한 획득 기법: usb 케이블 연결로 플래시메모리의 전체 데이터를 읽어 획득

usb 디버깅 모드 활성화 -> ADB Shell로 명령 -> 복제 이미지 생성

무결성 손실 위험

<백업 앱 기반 데이터 획득 절차>

1. 획득 가능 데이터 분석

텍스트, 폰트, 번호, 영수증이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

제조사별 백업용 앱, 백업 가능 주요 데이터

2. 백업용 앱을 이용한 데이터 획득 과정

①백업 앱 실행 및 백업 대상 데이터 선택

데이터 획득 대상의 스마트폰의 백업용 모바일 앱을 실행한 후 전송 대상 데이터를 선택

②백업파일 생성

백업용 모바일 앱은 시스템 앱의 앱 데이터에 저장된 정보를 추출하여 압축한 후 암호화하여 백업용 모바일 앱의 앱 데이터 영역에 저장(system권한, /data/user/0/com.sec.android.easyMover/fil es/). 암호화 시에는 “AES/CBC/PCKS5Padding” 암호 알고리즘을 사용하여 백업 파일을 암호화하고 세션 키는 xml파일 형태로 공유

③수신용 폰으로 생성파일 전송

암호화된 압축파일은 무선(Wifi) 또는 USB 연결을 통해 수신용 스마트폰의 Userdata 파티션의 내부 메모리 영역에 전송(/sdcard/Smart Switch/tmp/)

④백업 파일 전송 및 복호화

전송된 암호화된 압축파일은 공유된 세션키를 이용하여 복호화

⑤압축 해제 후 데이터 복원

복원과정에 서 생성된 암호화가 해제된 임시 파일은 작업 종료 후 모두 삭제됨

획득 대상 폰 -> 수신용 폰 데이터 전송, 복원 과정에서 임시 생성된 파일을 중간에 가로채는 방식으로 데이터 획득 가능

임시 생성 파일은 루팅 필요 없고 일반 사용자 권한으로 접근 가능텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

수신용 폰에 임시파일 저장되는 경로

<실험>

텍스트, 폰트, 번호, 스크린샷이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.대상: 메시지 데이터

획득, 수신용 폰 모드 비행기모드로 데이터 유입 차단

획득 대상에서 1. 설정 -> 클라우드 및 계정 -> Smart Switch (전송방법은 USB) -> 보내기

수신용에서 같은 설정으로 수신 대기상태 설정 후 개발자모드 활성화 -> USB 디버깅 활성화 -> PC와 USB 연결 후 PC에서 ADB Devices 명령으로 연결 확인

1. 설정 -> 클라우드 및 계정 -> Smart Switch (전송방법은 USB) -> 받기 (송신 기기: 안드로이드)

연결 후 메시지 항목 선택 후 전송

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

adb pull 위치 명령어를 사용하여 경로에 임시 생성 데이터 획득

edb, zip 파일 획득

(데이터 복구)

압축 해제 시 첨부되어 송수신된 jpeg 파일 확인 가능

SQLite의 비할당 영역 삭제된 레코드 복구 기법을 이용하여 Leaf 페이지에 있는 삭제 메시지 존재 여부 분석

먼저 Sqlite의 헤더 페이지 분석 -> Leaf (첫 바이트가 0x0D)탐색

Leaf는 바이트 오프셋 2-3에 2바이트 크기로 첫번째 비할당 블록의 오프셋 값 저장함 각각의 블록은 바로 다음의 비할당 블록의 오프셋을 상위 2바이트에 저장하는 방식으로 비할당 블록의 체인을 구성

비할당 블록 체인을 통해 탐색 시 삭제된 SMS 내용 확인 가능

무결성 검증 -> 해시값

MD5sum 툴을 사용해 원본 데이터와 백업 앱으로 획득한 데이터 해시값 비교

원본 데이터 획득을 위해서 루팅 가능한 폰으로 확인 결과 해시값 동일

대상: 주소록 (LG V30 기종)

비행기모드 -> 모바일 스위치 앱 실행 -> 저장위치 SD카드 -> 백업시 lbf 파일 하나로 백업 파일 생성

Hxd로 시그니처 기반 카빙 -> 원본 파일 추출 가능

Zip 파일 시그니처 헤더 시작 오프셋~Footer의 오프셋까지 데이터 카빙 기법으로 추출 시 단일 압축파일 획득 가능(주소록 저장된 DB파일)

Sqlite DB 분석법으로 분석결과 삭제한 연락처 포함 전체 정보 획득 가능

활용 방안

1. 증거 데이터 수집의 즉시성
2. 동일 제조사 스마트폰에 적용 가능
3. 삭제 데이터 복구 지원