iOS 애플리케이션 데이터 취약성에 관한 연구

이석준, 유형욱, 손태식
아주대학교 컴퓨터공학과
e-mail:wadeco@ajou.ac.kr

Research on the vulnerability of iOS Application Data

Seokjun Lee, Hyunguk Yoo, Taeshik Shon
Dept of Computer Engineering, Ajou University

요 약

최근 스마트폰 사용량이 급증함에 따라 사용자의 편의, 재미를 위한 수많은 애플리케이션이 출시되었으며 일상생활에서부터 비즈니스목적까지 다양한 분야에서 사용되고 있다. 스마트폰 애플리케이션은 필연적으로 내장메모리에 데이터를 저장하게 되는데, 그 데이터로는 대화기록, 개인정보, 유료구매 데이터 등이 있을 수 있다. 본 논문에서는 스마트폰 애플리케이션의 데이터 저장 및 로드 방식을 분석하여 어느 정도 데이터 보호가 되고 있는지, 데이터를 수정함에 따라 발생하는 문제는 무엇인지 등을 분석한 결과를 보일 것이다.

1. 서론

최근 스마트폰 사용량이 급증함에 따라 다양한 애플리케이션이 출시되었다. 스마트폰은 일반 휴대폰과 달리 Wi-Fi, 3G, LTE 등을 통한 데이터통신이 가능함에 따라, 통신사에서 제공하는 문자메시지를 서비스를 대체하기 위한 다양한 메신저형태의 애플리케이션들이 출시되기도 하였다. 또한, 사용자에게 즐거움을 줄 수 있도록 수많은 게임들이 스마트폰버전으로 출시되고, 최근 스마트폰 전용 게임들이 대거 출시됨에 따라, 생활에서 스마트폰의 사용시간 또한 급증하고 있는 추세이다.

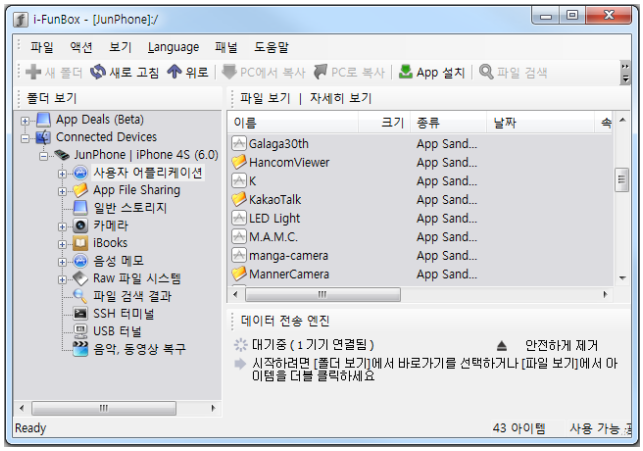
다양한 애플리케이션들이 동작하면, 애플리케이션 각각 관리하는 데이터가 생기게 되고, 스마트폰 OS에 따라 내부 메모리의 특정 위치 혹은 개발자가 지정한 위치에 데이터를 저장하게 된다. 특히 iOS같은 경우, 파일시스템 내 샌드박스 구조를 사용하여 애플리케이션 프로그램 및 데이터를 저장하는 공간이 지정되어 있으며, 그에 따라 손쉽게 파일을 빼내고, 수정하고, 다시 저장할 수 있다.

본 논문에서는 애플리케이션의 데이터를 분석하여 데이터 보호 수준에 대해 평가하고, '어떠한 경로로 유출 될 수 있는 것인가?'에 대해 분석한 결과를 보일 것이다. 논문의 2장에서는 실험대상에 대한 정의 및 사용도구에 대해 설명할 것이며, 3장에서는 메신저 애플리케이션 데이터의 유출 가능성에 대해 분석한 내용을 보이고, 4장에서는 게임 애플리케이션의 데이터 조작에 대한 분석 결과를 보일 것이며 5장에서 결론 및 향후 연구방향을 제시하였다.

2. 대상 시스템 설정 및 사용도구 소개

실험에 사용된 스마트폰은 iOS 6.0 버전이 설치된 iPhone4S를 사용하였으며, 탈옥되지 않은 상태이다. 또한, 데이터 유출의 위험성을 부각시키기 위해 비밀번호도 설정하고 실험하였다.

우선 iPhone4S 애플리케이션 데이터 파일 획득을 위해 윈도우즈 상에서 iPhone의 파일을 조회할 수 있는 무료 프로그램인 i-Funbox V1.99.958.697(www.i-funbox.com)을 사용하였다. i-Funbox는 GUI로 제작되었고, 윈도우즈 탐색기와 비슷한 뷰를 제공하여 사용하기 쉽다. 또한 아래 (그림 1)에서 보듯, 사용자 애플리케이션, 카메라, ibooks, 음성 메모 등 파일시스템에서 자주 접근 하게 되는 위치를 따로 링크하여 사용자에게 편의를 제공하고 있다.



(그림 1) i-Funbox 동작화면

데이터 분석을 위한 툴로는 메신저 애플리케이션의 DB 파일을 분석하기위해 SQLite Database Browser 2.0 b1를 사용하였다. SQLite는 파일하나로 데이터베이스를 관리하는 방식을 사용하여 소용량의 속도가 빠른 데이터베이스이다. SQLite Database Browser는 그 파일로부터 테이블 구조를 출력해 보여주는 GUI 툴이다. 또한 게임 애플리케이션 데이터의 복호화 및 수정을 수행하기 위해 HEX Workshop v5와 UltraEdit 18.0을 사용했다.

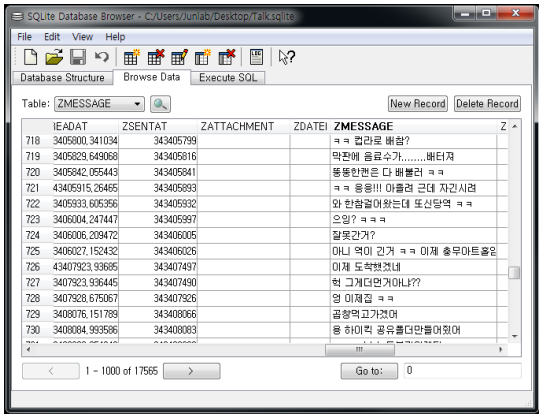
3. 메신저 애플리케이션 데이터 유출

스마트폰은 통신사의 문자메시지 이외에도 인터넷 데이터 통신을 사용하여 메시지를 전달하는 메신저 애플리케이션들이 다수 존재한다. 본 논문에서는 사용자가 가장 많은 두 애플리케이션인 '카카오톡', '마이피플'의 두 어플리케이션의 데이터를 확인하고, 유출 가능성에 대해 생각하였다.

3.1 메신저 애플리케이션 데이터 구조 및 조회

I-Funbox를 이용하여 사용자 어플리케이션-KakaoTalk 폴더를 확인해보면 Documents, KakaoTalk.app, Library 이 세 내부 폴더가 나타난다. Documents는 Apple사가 iTunes를 통해 사용자가 공식적으로 특정 앱에 파일을 넣고 뺄 수 있도록 허가된 폴더이다. KakaoTalk.app은 실제 프로그램 구동 시 필요한 데이터가 들어가 있으며, Library는 각 프로그램에서 데이터 저장 및 설정정보 관리 등에 사용하는 폴더이다.

KakaoTalk.app의 경우는 '/KakaoTalk/Library/Private Documents/Talk.sqlite'의 위치에 카카오톡 데이터베이스 파일이 저장되어있다. 위 파일은 스마트폰에 비밀번호가 설정된 상태에서도 파일을 복사할 수 있으며, 암호화가 되어있지 않기 때문에, SQLite Database Browser를 통해 내부 데이터를 확인할 수 있다. 다음 (그림 2)는 카카오톡 데이터베이스 파일로부터 주고받은 메시지 내용을 조회한 결과이다.



(그림 2) 카카오톡 데이터베이스의 메시지 부분

카카오톡의 데이터베이스에는 사용자의 기본정보 뿐 아니라 사용자의 모든 카카오톡 친구들의 이름, 닉네임, 전화번호 등이 노출되어 있다. 이는 악용 될 소지가 있으며, 데이터베이스의 구조만 파악하면 각 사용자별 고유 ID값을 통한 SQL문을 질의함으로써, 사용자별 대화를 추출해 낼 수도 있다.

또, 테스트 결과 타 기기의 카카오톡 데이터베이스 파일을 복사해 넣는 것만으로도 타인의 카카오톡 친구목록 및 대화기록을 자신의 기록인 것처럼 조회할 수 있다. 물론 단순 기록이기 때문에, 타인의 이름으로 메시지를 전송할 수는 없다.

'마이피플'의 경우, '카카오톡'과 비슷한 폴더구조를 갖고 있는데, '마이피플'은 '/마이피플/Documents/Air21-0.1.2.sqlite' 파일에 대화 기록 및 친구목록이 저장되어 있다.

3.2 데이터 유출의 위험성

데이터 유출 가능성을 생각해보면 다음과 같이 크게 두 가지 방법을 생각해볼 수 있다. 탈옥을 하지 않은 경우는 사용자가 조금만 주의하면 데이터 유출 가능성은 적어진다. 공공장소에서 잠깐 스마트폰을 방치하거나, 유료 휴대폰 충전소 같은 곳에 스마트폰 충전을 맡기는 경우, 위에 작성한 방법으로 데이터베이스 파일을 유출하는 경우 수분 내에 파일 복사가 가능하고, 스마트폰에 암호가 설정된 경우에도 아무 제약 없이 파일 복사가 가능하기 때문에, 사생활 침해당하지 않기 위해서 주의할 필요가 있다.

탈옥을 한 스마트폰의 경우 방식이 다른데, 데이터베이스 파일을 물리적인 접근을 통해 유출하는 방법 뿐 아니라, 탈옥 전용 앱스토어인 Cydia를 통해 악성 애플리케이션이 설치되면 각종 애플리케이션에 포함된 중요정보 데이터파일들이 유출될 가능성을 간과할 수 없다.

4. 게임 애플리케이션 데이터 조작

스마트폰으로 다양한 게임 애플리케이션을 설치해 즐길 수 있다. 그 중에는 게임 진행을 더 쉽게 하거나 더 나은 아바타를 위해 게임 내에서 유료 아이템을 결제해 사용할 수 있다. 하지만 상당수의 게임 데이터 파일은 암호화되어 있지 않거나, 약한 수준의 암호화가 되어 조작하는데 큰 어려움이 없다. 문제는 조작된 파일에 대한 검증을 수행하지 않는 애플리케이션이 매우 많다는 점이다. 최근 게임 산업이 발달함에 따라, 단순히 흥미 거리가 아닌 기업의 수익모델로도 사용되고 있는데, 이와 같이 조작이 가능한 경우, 게임회사에 상당한 금전적 손실을 가져올 수 있다. 4.1, 4.2에서 두 가지 예시를 들어 게임 데이터를 조작하는 실험 결과를 보일 것이다.

4.1 암호화되지 않는 게임 데이터의 조작

이번 절에서는 데이터를 암호화하지 않고 저장하는 한 야구게임의 골드를 조작하는 실험 결과를 보일 것이다. 본 게임은 세이브파일을 다수로 쪼개어 저장하였으나, 골드에

변화를 주었을 때 파일이 저장된 시간을 보고 판단하여, 조작이 가능하였다.

(그림 3) 야구게임 조작 전 화면 (251골드)

골드 값의 위치를 찾기 위해 10진수인 '251'을 16진수로 변환해 확인해보니 'FB'였다. 다음 (그림 4)는 세이브 파일에서 'FB'를 검색한 결과이다.

```
00000000h: 01 00 00 03 B5 CE BB EA BA A3 BE EE BD BA 00 00 ; ....     ..
00000010h: 00 00 BE C6 C0 CC BA C0 BE C6 C0 CC B8 B5 00 00 ; ..       ..
00000020h: B8 F0 BA F1 C5 AC 00 00 00 00 00 00 00 00 DB 07 ;    ........?
00000030h: 58 1B C4 09 D6 01 1E 00 00 00 00 00 00 00 00 00 ; X.??.........
00000040h: FB 00 00 00 0C 00 00 00 35 A6 05 3B 3A 01 00 00 ; ?.......5?;:...
```

(그림 4) 세이브 파일 내 골드 값 위치

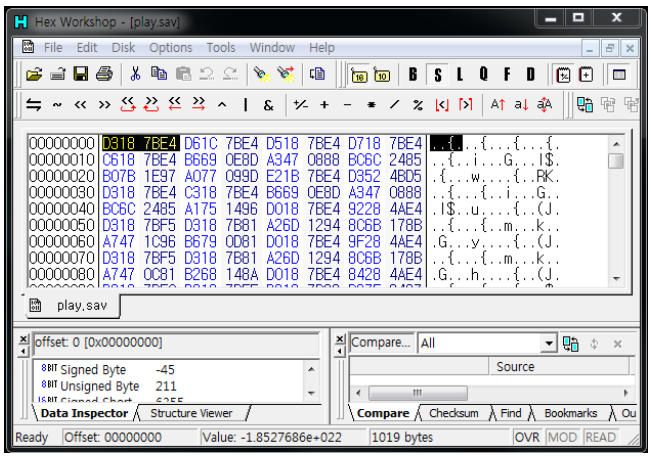
세이브파일의 0x040-0x043까지 Little Endian으로 골드 값을 표현하고 있음을 짐작할 수 있다. 실험을 위해 10000000의 HEX값인 989680을 Little Endian으로 하여 '80 96 98'로 수정한 뒤 다시 스마트폰에 집어넣어 게임을 재 구동하였다.

(그림 5) 야구게임 조작 후 화면 (10000000골드)

위의 (그림 5)와 같이 데이터 조작이 매우 간단함을 알 수 있다. 위 게임에서 골드는 유료이며, 네트워크를 통해 아이템을 사고팔수 있다. 이렇게 조작이 쉬운 경우 유료로 게임을 이용하는 사용자의 상대적 피해 및 게임회사의 금전적인 손실을 야기할 수 있다.

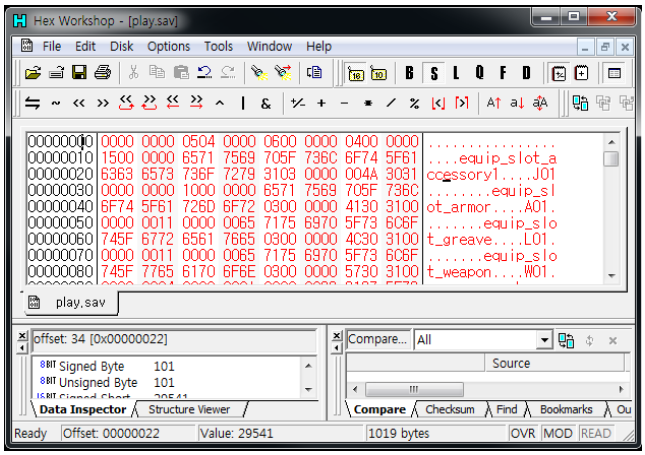
4.2 암호화된 게임의 데이터 조작

이번 절에서는 약한 암호화가 적용된 게임 세이브 데이터를 복호화 하여 조작해보는 실험을 하였다. 아래 (그림 6)은 i-Funbox를 통해 복사해 온 세이브데이터 play.sav 파일 을 Hex Workshop으로 로드했다.



(그림 6) 암호화 된 세이브데이터

얼핏 보면 우측에 ASCII 값들도 비슷한 패턴이 반복됨을 보이고 있고, 좌측에 Hex 코드도 유심히 살펴보면 맨 첫 4바이트의 D318 7BE4를 기준으로 조금씩 변동이 있음을 알 수 있다. 따라서 Hex Workshop의 기능 중 XOR을 이용해 4바이트 단위로 D318 7BE4를 전체에 XOR을 수행하면 아래 (그림 7)과 같이 복호화 되어 실제 데이터가 나타남을 알 수 있다. 맨 윗줄의 5, 6 번째 바이트가 Little Endian으로 저장된 게임머니의 값이며, 그 아래 줄부터는 ASCII 값에서 보이듯이 J01, A01, 등과 같이 아이템 코드가 존재하며, 이를 수정하고 다시 동일 키로 암호화하여 스마트폰에 파일을 저장하면, 실제 게임에 조작한 아이템이 적용되어있음을 알 수 있다.



(그림 7) 복호화 된 세이브데이터

이러한 간단한 XOR암호화 방식은 1차적으로 데이터를 감출 수 있고, 속도가 빠르다는 장점으로 사용되고 있다. 하지만 위 게임에서처럼 어느 정도 눈썰미를 가진 악성이용자라면 XOR의 Key값이 계속 바뀌어도 바로 Key를 알아낼 수 있을 것이다.

다수의 애플리케이션이 위와 같은 방식을 사용하고 있으며, 국내 출시되지 않은 게임의 대사 파일 등도 위와 같이 간단한 XOR 암호화를 사용하고 있어서, 누군가 한글화를 수행한다면, 국내 심의를 받지 않은 외국 게임이 한글판으로 유통될 가능성도 존재한다.

5. 결론 및 향후 연구방향

본 논문에서는 스마트폰 애플리케이션의 데이터 보호에 대해 분석한 결과를 보였다. 메신저 애플리케이션 및 게임 애플리케이션의 데이터들은 보안처리가 안된 부분이 상당수이며, 그 파일들로부터 정보를 얻고, 조작하는 것이 크게 어렵지 않음을 알 수 있었다.

현재 스마트폰 애플리케이션의 데이터 관리현황은 다음 <표 1> 과 같은 유형으로 분류해 볼 수 있다.

<표 1> 애플리케이션 데이터 저장방식에 따른 분류

저장 방식	저장 형태	이슈사항
데이터 평문저장	데이터를 DB파일 혹은 애플리케이션 제작사가 관리하고자 하는 형태로 구성된 파일로 저장한 형태	- 카카오톡, 마이피플 등과 같은 유형 - 악성이용자가 파일만 습득하면 데이터가 쉽게 유출됨 - 평문 저장은 피하는 것이 좋음
데이터 암호문저장	데이터보호를 위해 저장데이터에 암호화를 적용해 파일로 저장한 형태	- 키가 노출되면 평문과 다를 바 없고, 주로 키도 데이터와 함께 내부 저장소에 저장됨 - 주로 XOR을 사용
데이터 암호문저장 (검증포함)	데이터에 조작이 가해지면 검증할 수 있는 저장 형태	- 데이터 조작 시, CRC 등을 이용해 검출이 가능함 - 타임스탬프를 이용해 데이터파일 수정 시간을 비교해 검증하므로 조작이 어려움
데이터를 서버에 관리	데이터 자체를 서버에서 관리하는 형태	- 네트워크에 접속되어야 데이터에 접근할 수 있어 안정성은 높으나, 애플리케이션 사용성을 위해 잘 사용하지 않는 방법임

메신저 데이터베이스로부터 개인정보가 유출될 수 있고, 다양한 애플리케이션의 유료 아이템 부분을 데이터파일 조작으로 얻는다면 기업의 경제적 피해 및 선의의 피해자가 발생할 것이다. 이를 막기 위해서는 개발자입장에서 퍼포먼스가 좀 떨어지고, 한 단계 더 작업을 하더라도 좀 더 높은 수준의 데이터 보호를 할 필요가 있다.

향후에는 애플리케이션 데이터 보호 방안에 대한 연구 및 데이터 조작 시 데이터 조작 검증 방안에 대한 연구가 필요할 것으로 예상된다.

참고문헌

[1] i-Funbox 사용법, http://blog.i-funbox.com/?cat=8

[2] 류재철, “iOS 탈옥 취약점 분석”, 충남대학교 정보보호연구실, 2012.08.31

[3] 김기영, 강동호, “개방형 모바일 환경에서 스마트폰 보안기술”, 정보보호학회지 제19권 제5호, 2009. 10, 21-28

[4] 김명신, “스마트폰기반의 강화된 모바일 오피스 인증 및 보안시스템”, 숭실대학교 석사학위논문, 2012

[5] 구본민 외 4명, “Android & iOS 기반 스마트폰의 디지털 증거 수집 및 분석”, 정보보호학회논문지 제21권 제1호, 2011.2

[6] 김기연, 조성제, “스마트폰 보안 취약점 동향”, 한국정보과학회 한국컴퓨터종합학술대회 논문집 제 37권 제2호, 2010.11