안드로이드 앱 취약점 점검 도구 개발

김상후, 조제경, 류재철
충남대학교 컴퓨터공학과
e-mail:whoyas2@cnu.ac.kr, aiking@hanmail.net, jcryou@cnu.ac.kr

Fuzzing Tool for Android Application Vulnerability

Sangwho Kim, Jegyeong Jo, Jaechul Ryou
Dept of Computer Engineering, Chungnam National University

요 약

2014년 가트너에서 조사된 통계 자료에 의하면 스마트폰 사용자 4명중 3명은 안드로이드인 것으로 나타났다. 즉, 안드로이드에서 취약점이 발생할 경우 다른 스마트폰 OS에서 취약점이 발생할 때보다 3배 이상의 피해가 예상된다고 할 수 있다. 따라서 안드로이드 환경에서 앱에 대한 취약점을 찾고 조치를 취해야하는 작업이 지속되어야 한다. 그러나 취약점을 찾고 조치를 취하기 위해 분석가는 많은 시간을 소모하는데 비해 앱의 증가 속도는 매우 빨라 취약점 점검을 위한 자동화 도구는 필수적일 수밖에 없다. 이에 본 연구는 안드로이드 환경에서 작동하는 앱을 대상으로 취약점 점검을 수행하는 도구를 개발하고 연구하였다.

1. 서론

2014년 2월 가트너(Gartner)에서 발표한 '2013년 스마트폰 OS별 판매량 및 시장점유율'에 따르면 안드로이드(Android)가 78.4%를 차지하고 있다. 안드로이드가 2012년 점유율이 66.4%였던 점을 고려하면 스마트폰 OS 점유율은 급상승하고 있다.

이러한 안드로이드에서 취약점이 악용되어 피해가 발생한다면 시장 점유율에 비춰볼 때 그 피해나 파급도가 무시 못 할 수준이 된다는 것은 자명한 사실이다.

이에 본 연구는 안드로이드 환경에서 작동하는 앱(APP : Application)을 대상으로 취약점 점검을 수행하는 도구를 개발하고 연구하였다.

2. 관련 연구

위에서 언급한 바와 같이 그 잠재적 위험성과 예상되는 피해를 미연에 방지하고자 안드로이드에서의 취약점 점검에 대한 방법 및 이를 자동화하기 위한 여러 연구들이 진행되어 왔다.

2.1. Droid Fuzzer

지난 MoMM 2013(The 11th International Conference on Advances in Mobile Computing and Multimedia)에서 공개된 Droid Fuzzer는 안드로이드 앱인 APK파일 내에서 AndroidManifest.xml을 통해 선별된 특정 Activity(이하 액티비티)에 비정상적인 데이터(Abnormal Data)를 입력해 크래쉬를 유발, 기록할 수 있는 프로그램이다.

해당 연구에서는 대상 앱의 액티비티에 MIME(Multipurpose Internet Mail Extensions) 데이터를 입력 값으로 사용하였다.

Droid Fuzzer는 크게 3가지 모듈로 구분되는데 Androidmanifest.xml에서 액티비티 정보를 추출하는 pretreatment module, 액티비티 정보를 인텐트-필터 태그(Intent-filter tag)로 분석해 특정 액티비티를 선별하고 대상 앱에 입력할 비정상적인 데이터를 생성하는 variation module, 마지막으로 특정 액티비티에 비정상적인 데이터를 입력하여 크래쉬 발생 여부 확인 및 기록하는 dynamic detection module이 있다.

입력 데이터의 필드 구분을 사전에 정의해서 데이터 포맷 프로필(Data Format Profile)을 작성할 수 있어 전략적으로 취약점 점검을 가능하게 하였다.

2.2. Intent Fuzzer

'블랙햇 USA 2009'에서 iSEC Partners가 Intent Sniffer, Intent Fuzzer를 발표한 바 있다.

Intent Sniffer는 대상 앱을 실행 시키면 액티비티의 Intent(이하 인텐트)를 볼 수 있게끔 제작된 앱이다. 다만 명시적인 broadcast Intent가 아닌 경우만 모니터링이 가능하다.

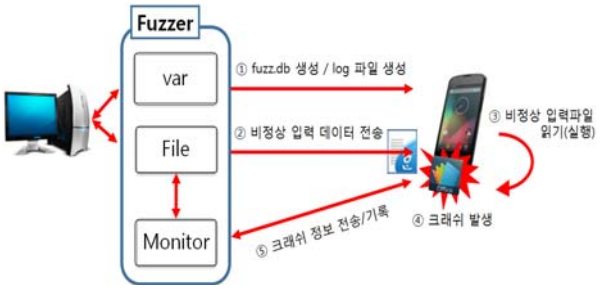
Intent Fuzzer는 대상 앱을 정해 Broadcasts, Services 등의 형태로 간단한 퍼징을 수행할 수 있게 제작된 앱이다.

2.3. 기존 연구와의 차이점

기존에 수행된 연구의 경우 안드로이드에서 제공하는 인텐트 메시지에 데이터를 추가할 수 있으며, 해당 데이터를 처리하는 과정에서 크래쉬를 유도한다. 하지만 제안하는 본 연구에서 인텐트 메시지는 앱을 호출하는데 사용하고, 실제 크래쉬를 유도하기 위해서 비정상적인 데이터를 전송하는 부분은 따로 파일을 생성하여 송신하도록 하였다.

3. 취약점 점검 도구 개발 및 실험

본 연구에서는 안드로이드 환경에서 작동하는 앱을 대상으로 취약점을 점검하는 도구를 개발하려고 한다. 취약점 점검 대상 앱은 소스가 공개되지 않을 수 있기에 (그림 1)과 같이 블랙박스 테스트 방법 중 하나인 퍼징(Fuzzing)을 이용하였다.



(그림 1) 취약점 점검 도구 작동 과정

3.1. 취약점 점검 도구 개발

취약점 점검을 수행할 PC는 안드로이드에서 기본적으로 제공하는 도구인 ADB(Android Debug Bridge)를 이용하여 스마트폰과 통신을 수행하도록 한다. 따라서 스마트폰은 대상 앱만 설치되어 있으면 되고, 실행이 가능한 환경이면 작동하도록 구현하였다.

취약점 점검 도구는 Droid Fuzzer와 마찬가지로 입력 데이터 포맷 프로필(XML 파일 형태) 파일과 퍼징을 수행하기 위한 모듈들로 구성된다. 모듈은 크게 세 가지 부분으로 나누어지는데 그 중 var 모듈은 xml 형태의 입력 데이터 포맷 프로필 파일을 읽어들여 fuzz.db 파일을 생성하고, AAPT(Android Asset Packaging Tool)를 이용하여 대상 앱을 분석하는 역할을 수행한다. File 모듈은 실제로 퍼징을 수행하는 모듈로써 fuzz.db 파일로부터 퍼징을 위한 정보를 읽어들여 비정상적인 데이터 파일을 생성한다. 또한 퍼징을 수행하고자 하는 대상 앱에 인텐트 메시지(Intent Message)를 송신하여 비정상적인 데이터 파일을 읽어들이도록 한다. 송신된 인텐트 메시지에 의해 앱은 비정상적인 파일을 처리하게 되며 이때 Monitor 모듈이 비정상적인 파일 처리 과정에서 발생하는 환경 상태 정보(Memory 등)를 확인하고, 로그캣(logcat)을 이용하여 안드로이드 로그(Log) 내 크래쉬 발생 여부를 판단하게 된다.

3.2. 취약점 점검 실험

본 연구에서 취약점 점검 대상으로 문서를 읽고 쓸 수 있는 앱인 한컴뷰어와 폴라리스 오피스를 선정하였으며, 입력 데이터로는 한글 문서를 사용하였다. 입력데이터인 한글 문서는 OLE(Object Linking and Embedding) 포맷으로 데이터가 구성되어 있으며, OLE에서 Section의 마지막 ID 값으로 지정하는 '0xFE' 값을 크래쉬를 유도하기 위한 데이터로 활용하였다.

3.3. 취약점 점검 결과

본 연구에서 개발한 도구의 성능을 검증하기 위하여 12kb 크기의 한글 문서를 대상으로 수행하였다. 수행한 결과는 <표 1>과 같은 결과를 도출하였다.

한컴뷰어를 대상으로 총 12,288회 파일 퍼징을 수행하였으며 실행 결과 25개의 크래쉬를 확인할 수 있었다. 폴라리스 오피스를 대상으로 총 12,288회의 파일 퍼징을 수행하였으며 실행 결과 392개의 크래쉬를 확인할 수 있었다.

<표 1> 실험 결과

대상 앱	비정상적인 데이터 입력 및 실행 횟수	크래쉬 발생 수
한컴 뷰어	12,286	25
폴라리스 오피스	12,286	392

4. 결론 및 향후 연구 방향

본 연구에서 개발한 취약점 점검 도구로 대상 앱을 퍼징한 결과 크래쉬가 발생하는 것을 확인하였다. 하지만 한컴 뷰어의 경우 퍼징 횟수에 비해 크래쉬가 발생하는 확률이 1%도 안되었고, 폴라리스 오피스도 3%정도로 효율적이지 못했다. 이에 한글 문서의 변조 데이터 값으로 '0xFE'를 활용하기 위해 연구했던 OLE 분석 결과 등을 이용하여 차후 var 모듈에 들어갈 입력 데이터 포맷 프로필을 구체적으로 작성하여 크래쉬 발생 예상 구간만 퍼징을 수행할 수 있도록 향상시킬 계획이다. 또한 File 모듈에 다양한 앱 및 입력 파일 포맷에 대해서도 퍼징을 지원할 수 있도록 따로 OLE 파서(Parser) 등의 모듈을 추가할 예정이다. 따라서 퍼징 횟수 대비 크래쉬 발생 효율을 높여 보다 효과적인 취약점 점검 도구로 발전 할 수 있을 것으로 기대된다.

Acknowledgement

본 연구는 미래부가 지원한 2013년 정보통신·방송(ICT) 연구개발사업 및 한국연구재단-차세대정보 · 컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(No. 2010-0020726)

참고문헌

[1] Hui Ye, Shaoyin Cheng, Lanbo Zhang, Fan Jiang, "DroidFuzzer: Fuzzing the Android Apps with Intent-Filter Tag", MoMM 2013, 2013. 12

[2] Jesse Burns, "Exploratory Android Surgery", BlackHat USA 2009, 2009. 07

[3] Daniel Rentz, "Microsoft Compound Document File Format", http://www.openoffice.org/sc/compdocfileformat.pdf, 2007.08

[4] ㈜한글과컴퓨터, "한글 문서 파일 구조(Hwp Document File Formats", HwpBinarySpecification.pdf, 2010.12