Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.25, NO.1, Feb. 2015 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2015.25.1.31

안드로이드 기반 문서 어플리케이션의 퍼징 방법론 연구*

조 제 경,† 류 재 철‡
충남대학교

Method of Fuzzing Document Application Based on Android Devices*

Je-gyeong Jo,† Jae-cheol Ryou‡
Chung-Nam National University

요 약

최근 사이버공격의 형태가 다양해지면서 악성코드를 직접 유포하는 대신 문서나 멀티미디어 파일을 유포하고 그 파일을 처리하는 과정에서 발생되는 취약점을 활용하는 사례가 빈번하게 보고되고 있다. 이 공격은 문서를 편집하거나 멀티미디어 파일을 재생하는 소프트웨어가 특정한 입력을 처리하는 과정에서 내재된 취약점이 나타날 수 있다는 점에 근거한다. 비정상 데이터를 임의로 생성하여 입력하는 퍼징(Fuzzing) 기법은 이러한 취약점을 찾아내기 위한 것이다.

본 논문에서는 문서 어플리케이션에 대한 기존의 퍼징 도구가 PC 환경에서 동작하는 한계를 해결하기 위하여 모바일 환경에 적용할 수 있는 퍼징 도구를 제안한다. 제안된 퍼징 도구는 모바일 문서 어플리케이션에서의 취약점을 효과적으로 발견할 수 있으며, 이를 통하여 모바일 환경에서의 APT 공격에 대응하는 도구로도 유용하게 사용될 수 있다.

ABSTRACT

As the forms of cyberattacks become diverse, there has been reported another case of exploiting vulnerabilities revealed while processing either a document or multimedia file that was distributed for attacking purpose, which would replace the traditional method of distributing malwares directly. The attack is based upon the observation that the softwares such as document editer or multimedia player may reveal inherent vulnerabilities on some specific inputs. The fuzzing methods that provide invalid random inputs for test purpose could discover such exploits.

This paper suggests a new fuzzing method on document applications that could work in mobile environments, in order to resolve the drawback that the existing methods run only in PC environments. Our methods could effectively discover the exploits of mobile applications, and thus could be utilized as a means of dealing with APT attacks in mobile environments.

Keywords: SmartPhone, Android, Fuzzing, Document, Vulnerability

I. 서 론

모바일 시장 점유율이 높아지면서 기존 PC 어플리케이션 개발자 및 개발사가 모바일 어플리케이션으로 이동하기 시작하였다. 사용자는 모바일 솔루션의 급증으로 모바일 장치에 대한 의존도가 높아지게 되었으며 개인 정보뿐만 아니라 회사의 업무 정보까지 모바일 장치에 저장하기 시작하였다. 따라서 악성코드 개발자는 기존 PC 환경에서 수행하던 공격을 모바일까지 확대하고 있다. 이에 따라 모바일 환경에서의 보안은 매우 중요한 사안으로 떠오르고 있으며 특히 안드로이드 장치 대상 피싱 및 스미싱 공격은 기하학적으로 증가하였다. 이러한 모바일 대상 공격을 막기 위하여 기존 PC환경의 보안 솔루션이 모바일 환경으로 많이 이전

접수일(2014년 9월 19일), 수정일(2014년 12월 1일), 게재확정일(2015년 2월 10일)
* 본 연구는 한국연구재단 운영체제 안전성 연구과제(NRF-2014M3C4A7030648)의 일환으로 수행하였습니다.
† 주저자, oroi@cnu.ac.kr
‡ 교신저자, jcryou@home.cnu.ac.kr(Corresponding author)

하고 있다. 그 중 퍼징(Fuzzing)은 PC환경과 달리 고려해야할 부분이 다양하다. 모바일 장치는 PC와 달리 CPU가 낮고 메모리가 부족하여 PC 환경에서 작동하던 퍼징 도구를 그대로 사용할 수 없기 때문에 모바일 환경에 특화된 퍼징 도구가 필요하다.

따라서 본 연구에서는 모바일 환경 중 많은 사용자를 보유하고 있는 안드로이드 환경에서의 퍼징 도구를 설계하고 개발하여 안드로이드 환경에 대한 안전성을 높이고 안전한 모바일 환경을 구축하고자 한다.

II. 관련 연구

퍼징은 다양한 입력 값을 생성하여 오류를 유도하기에 입력 값 생성 방법이 매우 중요하다. 따라서 퍼징에서의 입력 값 생성 방식인 Mutation과 Generation 기법에 대해서 설명하고, 본 연구에서 Mutation 기법을 선택한 이유를 설명한다. 그리고 안드로이드 환경에서의 퍼징 기법에 관한 기존 연구인 DroidFuzzer와 Dynodroid에 대해 설명하고 본 연구와의 차이를 설명하고자 한다.

2.1 Generation and Mutation Fuzzing

퍼징은 비정상적인 값을 어플리케이션에 입력하여 메모리 충돌과 같은 비 정상적인 동작을 유도하고, 해당 충돌을 탐지하여 오류를 찾아내는 기법을 지칭한다. 이러한 오류를 찾아내는 과정을 자동화하기 위하여 여러 연구가 이루어지고 있으며 대표적인 데이터 입력 기법으로 Generation과 Mutation이 있다[1].

Generation 기법은 사용자가 지정한 길이만큼을 생성하여 어플리케이션에 입력하는 방식이다. 실제 입력되는 값을 알 수 없을 경우에 사용되며 1Bit부터 시작하여 무한대까지 늘어날 수 있기에 속도가 느리다는 단점이 있다.

Mutation 기법은 입력 값을 알고 있는 경우에 사용할 수 있다. 오류를 찾아내고자 하는 어플리케이션에 입력되는 값을 알고 있을 경우 해당 값의 범위 내에서 변경을 수행하기에 Generation 기법에 비해 빠르다는 장점이 있다.

문서 어플리케이션과 같이 특정 파일을 읽는 어플리케이션은 일반적으로 입력되는 파일을 알 수 있기에 대부분 Mutation 기법을 이용하여 퍼징을 수행한다. 특히 대부분 문서 파일은 OLE(공통 표준 문서 포맷) 형태를 가지고 있기에 OLE 구조 중 일부만 퍼징이 가능한 Mutation 기법이 적합하다.

2.2 DroidFuzzer

DroidFuzzer[2]는 2013년 MoMM이라 불리우는 컨퍼런스에서 발표된 퍼징 도구로써 안드로이드 어플리케이션의 "AndroidManifest.xml" 파일 기반으로 작동한다. "AndroidManifest.xml" 파일은 안드로이드 어플리케이션 설치파일인 APK 파일 내에 존재하며 안드로이드의 IPC 통신 방법 중 하나인 Intent 메시지를 송수신하기 위한 정보를 담고 있다. DroidFuzzer는 Intent 메시지에 비정상적인 데이터를 첨부하여 송신하며 대상 어플리케이션이 메일 프로토콜인 MIME를 사용하는 경우에 퍼징이 가능하도록 설계되어 있다.

DroidFuzzer는 "Pretretment", "Variation", "Dynamic Detection" 모듈로 구성되어 있으며 "Pretretment" 모듈은 "AndroidManifest.xml"을 분석하는 역할을 수행한다. "Variation"은 비정상 데이터를 생성하며 "AndroidManifest.xml" 파일로부터 "Intent-Filter Tag" 값을 분석하여 어떠한 데이터를 생성할지 결정하게 된다. "Dynamic Detection"은 비정상데이터를 Intent메시지에 첨부하여 송신하며 메모리 충돌 여부를 확인하는 과정을 거친다.

본 연구에서는 DroidFuzzer와 유사하게 Intent 메시지를 이용하여 어플리케이션을 실행하지만 비정상데이터는 DroidFuzzer와 다르게 문서 파일을 생성하여 수행한다. 따라서 DroidFuzzer는 Intent 메시지를 처리하는 과정에서 발생 가능한 보안 문제점을 찾기 위한 연구에 집중되어 있는 반면, 제안하는 본 연구에서는 문서 파일을 처리하기 위한 어플리케이션의 내부 기능에서 발생 가능한 보안 문제점을 찾기에 연구의 목적과 수행 대상 자체가 다르다는 것을 알 수 있다.

2.3 Monkey & Dynodroid

안드로이드 어플리케이션의 사용자 인터페이스(UI) 문제점을 찾는 도구로 활용되고 있는 "Monkey"에 대해 설명하고자 한다[3]. UI event를 생성하여 어플리케이션에 전달하며 UI Event를 처리하는 과정에서 발생가능한 문제점을 보여준다. 하지

만 복잡한 UI나 다중 계층의 UI를 사용하는 어플리케이션에 대해서는 좋은 결과를 나타내지 못하고 있다.

"Monkey" 도구를 보강하기 위하여 "Dynodroid"라는 연구가 진행이 되었다[4]. 해당 연구는 UI Event를 생성할 때 탭(taps)과 제스쳐(gestures), 입력창(text inputs)을 고려하도록 설계되었으며 기존 도구와 비교하여 뛰어난 성능을 보여주고 있다.

하지만 오픈소스인 "Monkey"와 조지아공과대학에서 연구한 "Dynodroid"는 안드로이드 환경의 UI Event를 이용한 것으로 문서 어플리케이션의 보안 문제점을 찾고자 하는 본 연구와는 크게 상이한 것을 확인할 수 있다.

2.4 Peach Fuzzer

Peach Fuzzer는 MIT 라이센스를 사용하며 무료로 배포되고 있는 도구로 편의성 및 지속성이 뛰어나고 교육 및 기술 지원을 제공한다[6]. 프레임워크 형태로 제공하기에 사용자가 모듈을 개발하여 적용할 수 있으며 특히 퍼징을 위한 규칙을 설정하기에 따라 뛰어난 성능을 보여주는 도구로 유명하다.

하지만 기본적으로 제공하는 기능으로는 문서 파일의 전체나 특정 일부만 가능할 뿐 문서의 특정 영역을 찾아내 필터링하는 기능을 제공하지않아 문서 어플리케이션을 퍼징하기에는 많은 문제점을 가지고 있다. 특히 안드로이드와 같은 모바일 기기를 위한 모니터링을 지원하지 않아 메모리 충돌과 같은 오류를 확인할 수가 없다는 단점을 가지고 있다.

III. 제안 연구 방법

기존 PC환경에서 취약점이 발생하는 어플리케이션은 매우 많다. 그 중에서 APT 공격에 많이 사용되는 문서 어플리케이션은 PC 뿐만 아니라 모바일 환경에도 존재하기에 본 연구에서는 모바일 문서 어플리케이션의 취약점을 연구하기 위한 퍼징 도구를 설계/개발하였다.

3.1 문서 어플리케이션 퍼징 도구

본 연구에서 제안하는 문서 어플리케이션 퍼징 도구는 총 4가지 기능을 가지도록 설계하였다. 사용자 설정 값을 읽어 들여 수행하기 위한 "Setting" 기능과 비정상적인 문서 파일을 생성하기 위한 "File" 기능, 대상 어플리케이션을 분석하고 실행하며 오류 정보를 확인하기 위한 "Monitor" 기능, 마지막으로 대부분 문서에서 많이 사용하는 공통 표준 문서를 분석하기 위한 "OLE" 기능이다.

기본적인 기능은 PC 환경 아래에서 작동하며 안드로이드 장치에는 비정상 파일의 입력, Intent 메시지, 메모리 충돌 여부의 확인이 이루어진다. 전체적인 도구의 흐름은 "Fig. 1."과 같다.

단계 ① : 퍼징 설정(XML) 파일 생성(사용자)
단계 ② : 설정 파일을 이용한 퍼징 도구 실행 (PC, Python)
단계 ③ : 안드로이드 어플리케이션 파일(APK) 분석 (Package 및 Activity 정보 추출)
단계 ④ : 비정상 데이터 입력을 위한 문서 파일 생성
단계 ⑤ : 생성한 비정상 문서 파일을 안드로이드 장치로 전송
단계 ⑥ : 대상 어플리케이션에 비정상 문서 파일을 열기 위한 Intent 메시지 전송
단계 ⑦ : 대상 어플리케이션의 메모리 충돌 여부 확인
단계 ⑧ : 설정 파일의 내용에 맞추어 단계 ④부터 단계 ⑦까지 반복 수행

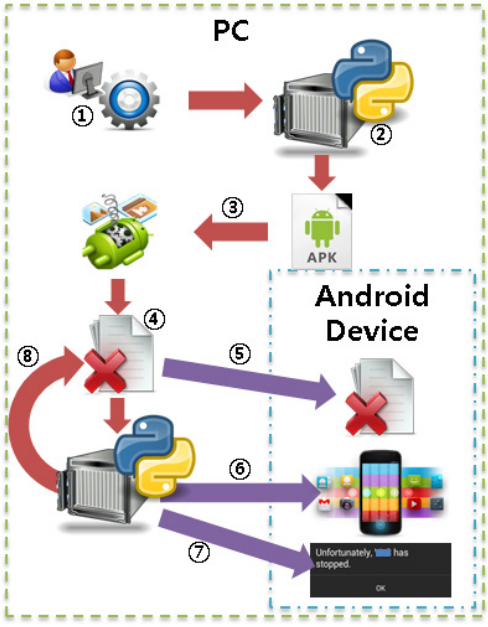

Fig.1. Flow of whole fuzzing system

3.2 Setting : 사용자 환경 설정 처리 기능

본 연구에서는 퍼징 도구가 작동하기 위한 다양한 정보를 XML 형태의 설정 파일로 저장하였다. 설정

파일 자체는 XML을 이용하지만 저장 및 처리는 Android 내부에서 사용하는 DB와 동일한 SQLite3를 이용하여 추후 Android 환경 내에서 모든 과정을 수행할 수 있도록 고려하였다[5].

주요 설정 정보로는 비정상 입력데이터를 생성하기 위한 원본 문서 파일의 경로와 비정상 데이터 생성 범위, 퍼징 수행 전 실행할 프로그램의 경로, 퍼징 수행 후 실행할 프로그램의 경로 등이다.

특히 기존의 퍼징 도구와 다르게 안드로이드 SDK의 adb와 aapt 파일에 대한 경로, 안드로이드 어플리케이션의 Activity 이름을 지정할 수 있도록 하였다. 안드로이드 어플리케이션은 시작 시 여러 Activity를 통하여 시작하는 코드의 위치를 지정 할 수 있다. 따라서 퍼징시 원하는 Activity를 지정함으로써 안드로이드 환경에서의 퍼징을 효율적으로 수행할 수 있다.

또한 퍼징을 수행하고자 하는 안드로이드 장치를 지정할 수 있는 기능을 제공한다. 퍼징은 시간이 많이 소요되기 때문에 여러 안드로이드 장치를 이용할 경우 빠르게 수행할 수 있다. 하나의 PC에 여러 안드로이드 장치를 연결할 수 있기 때문에 장치를 지정하는 기능은 안드로이드 퍼징 도구의 필수 기능이라고 할 수 있다.

설정 처리 기능에서는 앞서 설정한 내용을 가지고 안드로이드 어플리케이션의 package 이름 등 다양한 안드로이드 어플리케이션 정보를 추출하여 저장하게 되며 저장한 정보를 토대로 퍼징을 수행하게 된다.

3.3 File : 비정상 문서 파일 생성 기능

본 연구에서 제안하는 퍼징 도구는 안드로이드 어플리케이션에서 읽어 들이기 위한 비정상 문서 파일을 생성해야하기에 설정 기능에서 확인한 원본 파일을 기준으로 새로운 파일을 생성하게 된다. 설정 기능에서 기존 값 변경을 위한 시작 지점과 끝 지점을 지정하면 본 기능에서는 시작 지점부터 특정 bit 값으로 변경하며 비정상 문서 파일을 생성하게 된다.

생성한 파일은 안드로이드 장치로 전송하며 저장된 위치를 안드로이드 어플리케이션에 전달함으로써 비정상 데이터 입력을 수행할 수 있게 된다.

3.4 Monitor : 안드로이드 어플리케이션/장치 관리 기능

본 기능은 안드로이드 환경을 고려한 기능을 가지고 있으며 어플리케이션의 실행/종료, 안드로이드 장치의 로그 분석 등을 수행하게 된다.

안드로이드 장치의 경우 어플리케이션 시작과 종료를 위해서 Intent 메시지를 보내는 기능이 필요하며 Activity Manager를 사용하여 수행한다. Activity Manager는 Intent 메시지를 위해 Action 값과 Package 이름, Activity 이름을 옵션으로 전달할 수 있으며 특히 Data 필드에 열고자하는 문서 파일의 경로를 전달할 수 있다. 문서 파일의 경로가 Data 필드에 저장되어 Intent 메시지가 생성될 경우 안드로이드 어플리케이션은 해당 필드의 값을 자동으로 읽어 들인다. 해당 경로의 문서파일은 “File” 기능에서 생성한 비정상 문서 파일로써 안드로이드 어플리케이션이 자동으로 종료될 수도 있지만 주로 문서 파일 열기 실패 메시지가 출력되기에 Activity Manager를 통한 강제 종료를 수행하게 된다.

안드로이드 어플리케이션의 시작과 종료 이외에도 주요 기능으로써 안드로이드 장치에 대한 로그를 분석하는 기능이 존재한다. 안드로이드 운영체제는 안드로이드 장치 내에서 작동하는 하드웨어 및 소프트웨어에 대한 로그를 생성하고 있으며 이는 Logcat이라고 불리는 서비스를 통하여 조회할 수 있도록 되어 있다. 본 연구에서는 해당 서비스를 통하여 안드로이드 장치의 로그를 읽어 들이며 메모리 충돌과 관련된 문자열을 분석하여 비정상 문서에 의한 메모리 충돌 발생 여부를 판별하게 된다.

“Monitor” 기능의 주요한 부분 중 하나로는 메모리 점검 기능이 있을 수 있다. 안드로이드 환경에서 수행하는 퍼징은 기존 PC 환경과 다르게 장치의 메모리를 주기적으로 확인하여야 하며 비정상적인 종료로 인하여 할당된 메모리가 해제되지 않았을 경우 어플리케이션 자체가 실행되지 않을 수 있기 때문이다. 따라서 여유메모리가 일정 값 이하일 경우 장치의 재부팅을 수행하게 되며 부팅이 완료되기를 기다렸다가 퍼징을 재개하게 된다.

3.5 OLE : 공통 문서 규격을 위한 기능

우리나라에서 많이 사용하는 “한글과 컴퓨터”의 문서 규격부터 시작하여 Microsoft Office에 이르기까지 많은 문서 및 실행 파일들이 OLE 형태를 가지고 있다. OLE는 특정 사이즈를 기준으로 영역을 나누어 저장하고자 하는 데이터를 분산하여 저장함으로써 효율성을 극대화 시키고자 하였다. 따라서 파일 내 특정 영역을

수정하였을 경우 안드로이드 어플리케이션에 영향을 미치는 부분일수도 있지만 지정된 크기에 의한 빈 공간에 쓰이는 경우가 발생할 수 있다. 특히 하나의 Bit만 수정하는 것이 아닌 여러 Byte를 이용하여 비정상 데이터를 생성할 경우 지정된 크기의 경계선 상에서 발생 가능한 문제를 해결하기 위해 OLE를 지원하여야 한다. 특히 2Byte 이상의 데이터를 변경할 경우 1Byte는 OLE 내의 값을 변경하고 1Byte는 OLE 영역이 아닌 곳을 바꿀 수 있기 때문이다.

따라서 문서 어플리케이션 퍼징을 위해서 OLE 기능의 지원은 필수이며 본 연구에서는 OLE의 구조 중 데이터를 저장하는 Directory 영역의 이름을 가지고 퍼징을 수행할 수 있도록 하였다. 따라서 대용량의 문서 파일을 가지고 수행할 경우 영역을 효율적으로 나누어 수행할 수 있게 되며 여러 안드로이드 장치에 분산하여 수행할 수 있게 된다.

IV. 실 험

본 연구에서는 앞서 제안한 방법대로 퍼징 도구를 개발하였으며 Python과 SQLite3, Android SDK R22 버전을 이용하여 수행하였다. 수행한 환경은 Windows 7, XP, Apple OSX Leopard이었으며 모든 플랫폼에서 아무 문제없이 잘 작동하는 것을 확인하였다.

본 연구의 실험에서는 2개의 안드로이드 어플리케이션에 대해 실험하였다. 하나는 "Polaris Office" 였으며, 다른 하나는 "Hancom Viewer" 였다. 두 개의 프로그램은 안드로이드 앱스토어에서 비즈니스 분야 3위, 6위 자리를 차지하고 있으며, 문서 뷰어로써는 1위, 2위를 차지하고 있다. 따라서 본 연구의 실험에서 발견된 취약점이 악의적인 목적으로 이용될 경우 큰 파장을 일으킬 것으로 예상된다.

또한 실험 대상인 문서 어플리케이션에 퍼징을 수행할 기준 파일은 PC환경에서의 한글 어플리케이션에서 일반적인 문자열을 입력하여 만든 12Kb 크기의 파일이며 "한글과 컴퓨터"사에서 지정한 문서 파일 규격을 사용하였다.

본 연구에서 개발한 도구를 이용하여 퍼징을 수행하였을 경우 "Hancom Viewer"에서는 총 26개의 메모리 충돌이 탐지되었으며 "Polaris Office"에서는 392개의 메모리 충돌이 탐지되었다. 실험 전체에 의한 결과는 "Table 2."와 같이 정리할 수 있다.

위 결과는 12,288 Byte의 한글 파일 전체에 대해 수행한 결과이며 본 연구에서 지원하는 OLE 기능을 이용하였을 경우에는 "Table 3."과 같이 적은 횟수의 시도로 많은 메모리 충돌을 탐지할 수 있었다. 적은 횟수의 시도로 인하여 퍼징 시간 역시 줄어들었으며, 실험 대상이 된 12,288Byte 파일에서는 1/3 보다 적은 시도횟수와 시간 소모로 다수의 취약점을 발견 할 수 있었다.

본 실험에서는 OLE 영역만을 수행하기에 발견된 취약점 수는 전체 퍼징 시도에 비해 상대적으로 적게 발견되었으나 많은 시간을 단축할 수 있었다.

Table 1. Development and Implement environment of fuzzing system

Development	OS	Windows 7
	Tools	Python 2.7, android sdk r22
	Device	Samsung Note 2
Implement	OS	Windows 7, XP, Ubuntu, OSX
	Tools	Python 2.7, android sdk r22
	Device	Samsung Note 2

Table 2. Fuzzing result of Android applications

Target APP	Hancom Viewer	Infraware Polaris Office 5
Target File(size)	HWP file format (12,288 byte)	
Run Time	41h 44m 01s	41h 07m 00s
Run Count	12,288	12,288
Found Crash	26	392

Table 3. Fuzzing result with OLE supported

Target APP	Hancom Viewer	
Target File(size)	HWP file format (12,288 byte)	
Fuzzing Range	Full	Directory Stream ("docinfo")
Run Time	41h 44m 01s	13h 05m 23s
Run Count	12,288	3,860
Found Crash	26	4

본 실험과 타 퍼징 도구와의 성능 비교를 통하여 본 연구의 우수성을 확실하게 표현하고자 하였으나 기존의 퍼징 도구 중 문서 어플리케이션에 사용이 가능한 경우(Peach, Sully 등)에는 모바일 기기에 적용이 불가능하며, 모바일 기기에 적용이 가능한 퍼징 도구(DroidFuzzer, Dynodroid 등)는 문서 어플리케이션에 대한 퍼징을 지원하지 않아 성능 비교가 불가능하였다. 따라서 본 연구의 우수성을 성능 비교를 통하여 나타 낼 수는 없지만 기존에 없던 독창적인 연구로써 그 우수성을 표현할 수가 있다.

V. 결 론

실험 결과에서 볼 수 있듯이 본 연구의 퍼징 도구는 문서를 보기위해 만들어진 안드로이드 어플리케이션에서 다수의 메모리 충돌 취약점을 발견할 수 있다는 것을 증명하였다. 또한 본 연구에서 지원하는 문서 규격에 최적화된 기능을 통하여 퍼징에 수행되는 시간을 획기적으로 줄이는 것을 확인하였다.

하지만 본 연구에서 제안하는 퍼징 도구는 아직까지 메모리 충돌을 통해 확인한 취약점이 서비스 거부 형태의 취약점인지 임의의 코드를 실행할 수 있는 취약점인지 확인 할 수가 없다. 따라서 본 연구의 향후 주제로 취약점 종류를 판단할 수 있는 기능을 연구하고자한다. 또한 PDF 파일 역시 문서 어플리케이션으로 많이 사용하고 있으며 스크립트 형태의 파일 저장 방식을 사용하기에 추후 PDF 스크립트 파싱 엔진 연구를 통한 발전 가능성 고려하고 있다.

본 연구에서는 두 개의 안드로이드 어플리케이션을 대상으로 수행하였지만 안드로이드 앱스토어에서 가장 많은 인기를 얻고 있는 문서 어플리케이션이기 때문에 그 파급력은 매우 크다고 판단된다. 또한 "Polraris Office"의 경우 삼성 스마트폰에 기본으로 설치되는 어플리케이션인 만큼 취약점에 의한 파급력은 매우 크다고 볼 수 있다.

APT 공격과 같은 공격에 대응하기 위해서는 문서 어플리케이션과 같은 업무와 관련된 어플리케이션에 대한 취약점 분석 자동화 도구는 필수라고 할 수 있다. 이에 본 연구와 같은 모바일 환경에서의 취약점 분석 자동화 도구의 발전으로 안전한 모바일 환경을 구축하여야 한다.

References

[1] "Fuzzing - Mutation vs. Generation," http://resources.infosecinstitute.com/fuzzing-mutation-vs-generation/

[2] Hui Ye1, Shaoyin Cheng, Lanbo Zhang and Fan Jiang, "DroidFuzzer: Fuzzing the Android Apps with Intent-Filter Tag," International Conference on Advances in Mobile Computing & Multimedia, pp. 68-74, Dec. 2013.

[3] "UI/Application Exerciser Monkey." http://developer.android.com/tools/help/monkey.html

[4] Aravind MacHiry, Rohan Tahiliani and Mayur Naik, "Dynodroid: An Input Generation System for Android Apps," ESEC/FSE 2013 Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering, pp 224-234, Aug, 2013

[5] Android.Database, "Android Developers," http://developer.android.com/reference/android/database/package-summary.html

[6] Korea Internet & Security Agency, "A Study on Major Domestic S/W Vulnerability Discovery and Analysis Method," 2012.10

〈저자소개〉

조 제 경 (Je-gyeong Jo) 학생회원
2006년 2월: 한신대학교 정보시스템공학 졸업
2008년 8월: 한신대학교 컴퓨터정보학 석사
2014년 3월~현재: 충남대학교 컴퓨터공학 박사과정
〈관심분야〉 정보보호, 시스템보안, 네트워크보안

류 재 철 (Jae-cheol Ryou) 종신회원
1985년 2월: 한양대학교 산업공학과 졸업
1988년 5월: Iowa State University 전산학 석사
1990년 12월: Northwestern University 전산학 박사
1991년 2월~현재: 충남대학교 컴퓨터공학과 교수
〈관심분야〉 정보보호, 네트워크보안, 암호학, 보안프로토콜