인텐트 퍼즈 테스트 기반 안드로이드 테스트 케이스 자동 생성 방법

이승휘, 이화중, 최광훈
연세대학교 원주캠퍼스 컴퓨터정보통신공학부
{sunkus711,hj_tl22,kwanghoon.choi}@yonsei.ac.kr

An Automatic Generation Method for Android JUnit Testcases Using Intent Fuzz Testing

요 약

본 논문은 안드로이드 앱의 취약점으로 앱이 비정상 종료되는 사례를 인텐트 퍼즈 테스트로 찾아내고 그 테스트 케이스를 자동으로 생성하는 방법을 제안한다. 앱의 설정 정보를 바탕으로 자동 생성된 인텐트 명세를 활용하여 퍼즈 테스트를 진행하고, 비정상 종료되는 사례를 그 원인에 따라 그룹화한 다음, 각 원인 별 테스트 케이스를 자동으로 만든다. 이 방법은 개발자가 수작업으로 테스트 케이스를 만들 필요가 없고, 자동 생성된 이 테스트 케이스를 실행하면 안드로이드 앱이 반드시 비정상 종료되고, 생성된 테스트 케이스들은 서로 다른 앱 취약점을 공격하는 장점이 있다. 10개의 상용 앱으로 제안한 아이디어를 실험하여 이 방법의 활용 가능성을 논의한다.

1. 서 론

안드로이드 앱에 있어서 중요한 문제 중 하나는 인텐트 취약점으로 인한 앱의 비정상 종료이다. 인텐트란 안드로이드 앱에서 컴포넌트 간 통신을 위한 IPC메시지이고 컴포넌트를 메서드에, 메서드 인자를 인텐트에 비유할 수 있다. 이 때, 안드로이드 플랫폼에서 컴포넌트에서 요구하는 인텐트의 형태를 컴파일 시간에 검사하지 못하므로 인텐트 취약점이 발생하게 된다[1]. 예시로, 컴포넌트에서 요구하는 인텐트의 형태는 String 타입이지만 인텐트에 Integer 타입의 값을 지정하는 경우 런타임 예외가 발생한다.

이러한 인텐트 취약점을 검출하기 위해 인텐트 퍼저를 이용할 수 있다. 인텐트 퍼저란 랜덤 생성한 인텐트를 컴포넌트에 전달해서 앱이 비정상 종료되는지 테스트하는 도구이다. 인텐트 퍼즈 테스트에 관한 기존 연구[2][3]에서 상용 안드로이드 앱의 인텐트 취약점이 심각함을 보고한 바 있다. 인텐트에 랜덤으로 생성한 정보를 지정해 안드로이드 컴포넌트에게 전달하여 실험한 결과 332개의 액티비티 중 29(8.7%)개가 NullPointerException, ClassNotFoundException, IllegalArgumentException의 오류가 발생해 비정상으로 종료되었고 심지어 OS가 재부까지 되는 사례도 있었다.

인텐트 퍼저를 이용한 랜덤테스트의 결과를 자동으로 그룹화하여 분석하는 기존 연구가 있었다[4]. 먼저 인텐트 퍼저를 통해 각자의 방법으로 임의의 인텐트를 생성해서 안드로이드 앱이 실행 중에 비정상적으로 종료하지 않는지 테스트하고, 비정상 종료 시 얻은 로그를 분석하여 유사한 에러를 하나의 그룹으로 모아 카운팅을 하였다.

이렇게 발견된 인텐트 취약점으로부터 테스트 케이스를 생성하면 앱의 안전도가 높아질 뿐만 아니라 회귀테스트에 용이하다. 하지만 기존의 연구에서는 이 방법으로 찾아낸 인텐트 취약점으로부터 테스트 케이스를 생성하려면 사용자가 직접 에러 로그를 보고 생성해야만 했다.

본 논문에서는 이러한 테스트 케이스를 인텐트 퍼저를 통해 얻은 결과에서 자동으로 생성하는 방법을 제안한다. 인텐트 퍼저의 결과에서 에러 로그를 자동으로 그룹화하여 대표 로그를 분석한 뒤, 대표 로그를 기반으로 테스트 케이스를 자동으로 생성한다.

논문의 구성은 다음과 같다. 2절에서 관련 연구에 대해서 설명하고, 3절에서는 인텐트 퍼즈 테스트의 결과에서 테스트 케이스를 자동으로 생성하는 방법에 대해 설명한다. 4절에서 제안된 방법으로 적용하여 실험한 결과를 설명한 다음, 5절에서 결론을 맺고 향후 연구를 논의한다.

2. 관련 연구

이 논문은 2014년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. NRF-2014R1A1A2053446)

안드로이드 앱의 취약점을 테스트할 때 낮은 비용으로 효과적인 테스트 케이스를 만들기가 쉽지 않다. 상호작용 기반의 안드로이드 앱 테스트 기법[5]에서 안드로이드 컴포넌트 간 상호작용 모델을 세워 테스트 케이스를 생성하여 랜덤 테스트 방법(Monkey 도구)과 테스트 커버리지를 비교하였다. 테스트 케이스를 자동으로 생성하지 않고 테스트 커버리지를 측정하였으나 잘 동작하는지 여부를 판단하지 않았다. 또한 랜덤 테스트 방법이 테스트 커버리지에 있어 효과적인 앱도 있음을 보고 하였다.

안드로이드 컴포넌트 간 통신의 테스트 케이스 생성과 모델링[6]에서는 앱을 분석하여 자동으로 테스트 케이스를 생성하는 방법과 실험 결과를 보였다. 인텐트와 인텐트 필터 커버리지에 대한 실험 결과로 안드로이드 컴포넌트 간 호출을 테스트했음을 보였지만 자동 테스트 케이스 생성 방법을 명확하게 이해하기 어렵다.

모바일 어플리케이션 테스트에 관한 연구[7]에서 논의한, 안드로이드 앱 테스트에 관한 다양한 연구를 살펴보면 인텐트 취약점으로 앱이 비정상 종료되는 테스트 케이스를 자동으로 만들어내는 연구는 찾을 수 없었다.

3. 안드로이드 앱의 인텐트 퍼즈 테스트 결과로 테스트 케이스를 자동으로 생성하는 방법

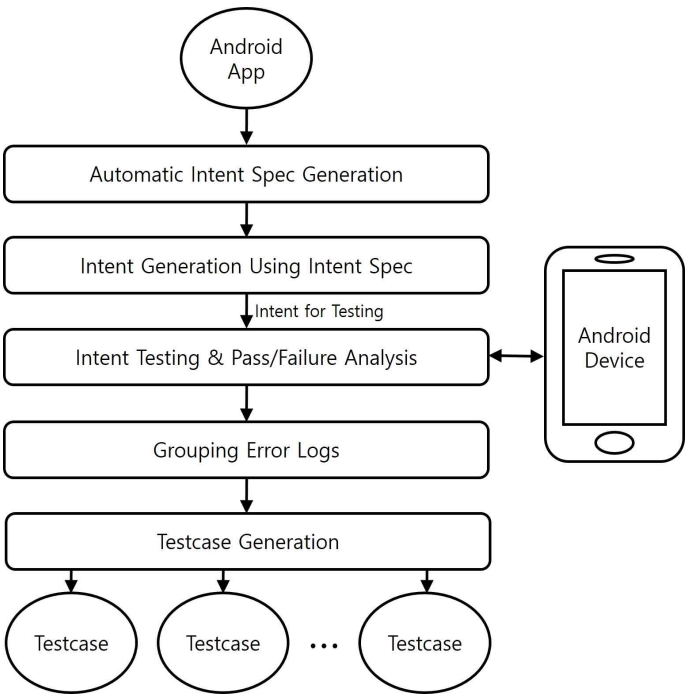

그림 1 인텐트 퍼저를 이용한 테스트 케이스 자동 생성
Fig. 1 Automatically Generating Testcases Using Intent Fuzz Testing

본 논문에서는 제안하는 안드로이드 테스트케이스 자동 생성 방법은 [그림 1]과 같은 단계를 거친다.

- 첫째, 안드로이드 프로젝트 또는 APK 파일의 앱 설정 파일(AndroidManifest.xml)로부터 자동으로 인텐트 명세[8][9][10]를 생성한다.
- 둘째, 생성한 인텐트 명세를 이용하되 부족한 정보는 랜덤 값으로 채워 넣어 테스트용 인텐트를 만든다.
- 셋째, 테스트용 인텐트를 ADB 명령어로 변환하고 ADB 인터페이스로 안드로이드 기기에 전달하여 안드로이드 앱을 실행하고 비정상 종료 되는지를 확인한다.
- 넷째, 앞의 인텐트 테스트 로그에서 비정상 종료(fail)되는 경우에서 유사성을 비교하여 동일한 원인이라 판단하는 에러 로그를 그룹화하고 대표 로그를 찾는다.
- 다섯째, 각 비정상 종료 경우를 대표하는 에러 로그를 만들어 낸 인텐트를 참고하여 안드로이드 JUnit 테스트 케이스 코드를 생성한다.

4. 실험

10개의 상용 앱에 이 논문에서 제안한 방법을 적용하여 실험하였다.

표 1 실험 결과
Table 1 Result

앱이름	인텐트 개수	fail 개수	TC 개수	테스트 시간	그룹화 시간	전체 시간
Between	2340	0	0	198분	10초	199분
Clean Master	2400	4	4	204분	9초	205분
Face Book	8640	0	0	732분	13초	732분
HoHo	4440	231	15	379분	485초	388분
Insta gram	1140	0	0	96분	2초	96분
Kakao Story	2580	77	13	219분	118초	222분
Kakao Talk	5460	182	24	691분	550초	702분
LINE	2040	105	11	204분	162초	208분
Melon	3600	137	11	885분	796초	899분
모두의 마블	480	28	2	45분	26초	46분

[표 1]의 첫 번째 칼럼은 실험한 안드로이드 앱의 이름이고, 두 번째 칼럼은 각 앱의 인텐트 명세로부터 생성한 테스트용 인텐트 개수, 세 번째 칼럼은 테스트 도중 인텐

트 취약점으로 인해 비정상 종료된 경우의 개수, 네 번째 칼럼은 비정상 종료된 에러 로그를 그룹화하여 만들어진 그룹으로부터 생성한 테스트케이스의 개수, 다섯 번째 칼럼은 테스트용 인텐트를 테스트하는데에 걸린 시간, 여섯 번째 칼럼은 그룹화에 걸린 시간, 마지막 칼럼은 각 앱의 테스트에 걸린 전체 시간이다.

전체 10개의 앱에서 33120개의 테스트 인텐트를 생성하여 테스트한 결과 766개의 에러가 발생하였으며, 이를 그룹화하여 얻은 대표 에러 로그의 개수는 80개였다. 764개의 에러 중 NullPointExceptionError가 572개(74.9%), UnsupportedOperationExceptionError가 186개(24.3%), 불명 에러가 6개(0.8%)였다. 80개의 대표 에러 로그 중 NullPointExceptionError가 57개(71.3%), UnsupportedOperationExceptionError가 19개(23.7%), 불명 에러가 4개(5%)였다.

대표 에러 로그 하나당 하나의 테스트케이스를 자동 생성하였으며 [그림 2]는 그 중 일부이다.

```
public void testcase1 ()
{
  Intent intent = new Intent();
  intent.setClassName("vStudio.Android.Camera360",
  "com.pinguo.camera360.save.processer.PhotoProcessService");
  intent.setAction("com.android.action.Edit");
  try { intent.setData(Uri.parse("7w45C:CKR")); }
  catch (Throwable t) { }
  intent.addCategory("qu_YtVG");
  intent.addCategory("vhKbe");
  startService(intent);
}
```

그림 2 자동 생성된 테스트케이스 예

Fig. 2 Automatically Generated Testcase example

이 논문에서 제안한 방법의 장점은 안드로이드 JUnit 테스트 코드를 생성하는 전 과정이 자동화 가능한 것이다. 개발자가 테스트 케이스를 수작업으로 만드는 비용을 줄일 수 있다. 또한 자동으로 생성된 테스트 코드를 실행하면 해당 앱이 비정상 종료되기 때문에 매우 중요한 테스트 케이스이다. 단점은 전 과정을 자동화하기 위해서 인텐트 명세에서 부족한 정보를 랜덤하게 채워 넣다 보니 앱의 실행과 무관한 테스트 인텐트를 생성할 가능성이 높다는 것이다. 기존 연구 [1,2,3,4,8,9,10]에서 보고된 바와 같이 인텐트 퍼즈 테스트 방법이 인텐트 취약점을 찾는데 매우 효과적이었다. 기존 연구 결과와 동일한 이유로, 실험 결과에서와 같이 의미 있는 안드로이드 JUnit 테스트 케이스를 생성할 수 있었다. 이렇게 생성한 케이스를 활용하여, 개발자는 앱 개발 과정 중에 회귀 테스트를 쉽게 적용할 수 있을 것이다.

5. 결론

안드로이드 앱의 인텐트 퍼즈 테스트 결과로부터 얻은 에러 로그에서 대표 로그를 분석하여 테스트 케이스를 자동으로 생성하는 방법을 제안하였다. 상용 안드로이드 앱에 제안한 방법을 적용하는 실험을 통해 적절한 테스트 케이스를 큰 비용없이 자동으로 생성할 수 있음을 보였다.

참고 문헌

[1] Amiya K, Maji, Fahad A. Arshad, Saurabh Bagchi, and Jan S. Rellermeyer, “An empirical study of the robustness of Inter-component Communication in Android”, In proceedings of the 2012 42nd Annual IEEE/IFIP International comference on Dependable Systems and Networks (DSN) (DSN’12), IEEE Computer society, Washington, DC, USA, 1-12, 2012

[2] J. Burns, Intent Fuzzer, [Online]. Available: https://www.isecpartners.com-/tools/mobile-security/intent-fuzzer.aspx, iSEC Partners, 2009.

[3] R. Sasnauskas and J. Regehr, "Intent Fuzzer: Crafting Intents of Death," Proc. of the 2014 Joint International Workshop on Dynamic Analysis (WODA) and Software and System Performance Testing, Debugging, and Analytics (PERTEA), pp. 1-5, San Jose, CA, 2014, ACM.

[4] 김현순, 윤성빈, 최지선, 고명필, 최광훈, 안드로이드 앱의 랜덤 인텐트 테스트에서 동일한 에러 로그를 자동으로 그룹화하는 방법, 한국정보처리학회 추계학술발표대회, 제주한라대학교, 2015년10월30-31일.

[5] 이정욱, 채흥석, 상호작용 기반의 안드로이드 앱 테스트 기법, 정보과학회논문지: 소프트웨어 및 응용, 제40권, 제5호, pp.263-276, 2013년5월.

[6] Ajay Kumar Jha, Sunghee Lee, Woo Jin Lee, Modeling and Test Case Generation of Inter-Component Communication in Android, 2nd ACM International Conference on Mobile Software Engineering and Systems, pp113-116, 2015.

[7] Samer Zein, Norsaremah Salleh, John Grundy, A Systematic Mapping Study of Mobile Application Testing Techniques, The Journal of Systems and Software, Vol.117, pp.334-356, 2016.

[8] 고명필, 최광훈, 창병모, 강건한 안드로이드 앱을 위한 실행시간 인텐트 스펙 검사 방법, 한국 소프트웨어공학 학술대회, 강원도 평창 한화리조트, 2015년 1월28-30일.

[9] 고명필, 최광훈, 창병모, 강건한 안드로이드 어플리케이션 개발을 위한 실행시간 인텐트 명세 검사 기법, , 정보과학회논문지, 43권, 2호, pp212-221, 2016년 2월.

[10] 고명필, 최광훈, 창병모, 인텐트 스펙 기반 안드로이드 유닛 테스팅 프레임워크 설계와 구현, KCC 2015, 제주대학교, 2015년 6월24-26일.