안드로이드 앱 검수 자동화를 위한 인텐트 스펙 기반 테스팅 방법

윤성빈, 최지선○, 최광훈
연세대학교 원주캠퍼스 컴퓨터정보통신공학부
{biassb, giftchoi, kwanghoon.choi}@yonsei.ac.kr

An Intent Specification based Testing Method for Automating Android App Inspection

Sungbin Yoon, Jisun Choi○, Kwanghoon Choi
Computer&Telecommunications Engineering Division, Yonsei University, Wonju

요 약

안드로이드 앱은 인텐트를 이용해 컴포넌트를 호출하는 형태로 구성되어 있다. 안드로이드 컴포넌트 호출 시 인텐트의 데이터가 누락되거나 타입 오류가 있더라도 실행시간에 드러나는 인텐트 취약점이 있다. 본 논문은 안드로이드 앱의 설정 정보로부터 인텐트 스펙을 자동으로 생성해서 인텐트 취약점을 자동 테스트하는 검사 방법을 제안한다. 앱 스토어의 검수 과정 자동화에 이 방법을 적용할 수 있다.

1. 서 론

안드로이드 앱 스토어에 새로운 앱이 지속적으로 등록되고 주기적으로 업데이트되는 상황에서 모든 앱을 빠짐없이 검수하기 위해 자동화 과정이 필요하다.

안드로이드 앱은 액티비티, 서비스, 브로드캐스트 리시버와 같은 컴포넌트들을 포함하고, 인텐트라 부르는 메시지를 전달하여 각 컴포넌트들을 실행한다. 컴포넌트를 함수로 비유하면 인텐트는 함수의 인자이다.

이 논문은 인텐트를 잘못 사용하는 안드로이드 앱을 가려내는 검수 방법에 관해 연구한다. 안드로이드 앱의 컴포넌트에서 기대하는 인텐트와 이 컴포넌트를 호출할 때 실제로 주어진 인텐트 사이에 일관성이 없어 비정상 종료되는 취약점이 많다[1]. 컴포넌트를 호출할 때 사용하는 인텐트에 데이터가 빠져있거나 데이터 타입이 달라도 컴파일 시점에 검출하지 못하고 앱을 실행할 때 이 취약점이 비로소 드러나게 된다.

[그림 1]은 인텐트 정보를 이용해 제목과 내용을 보여주는 Note 액티비티다. 인텐트의 정보로 action은 *EDIT*와 *INSERT*를 기대한다. action이 *EDIT*인 경우는 *title, content*키에 *String* 타입의 extra를 추가로 기대한다.

이 예제는 잘못된 인텐트로 인한 취약점이 있다. 첫째, 인텐트의 action이 *null*이거나 *EDIT*, *INSERT*가 아니면 *NullPointException*이 발생한다. 둘째, action이 *EDIT*일 때 *title*, *content*키의 값이 *null*이거나 *String*타입이 아니면 *NullPointException*이 발생한다.

이 논문은 안드로이드 앱을 입력 받아 앞에서 설명한 인텐트 취약점이 있는지 여부를 검사하는 방법을 제안한다. 앱 스토어에서 검수해야 할 사항 중 하나인 인텐트 취약점을 자동으로 검사하는 방법으로 활용할 수 있다.

```
public class Note extends Activity {
  String title, content;
  void onCreate(Bundle savedInstanceState) {
    Intent intent = getIntent();
    String action = intent.getAction();
    if (Intent.ACTION_EDIT.equals(action)) {
      title = intent.getStringExtra("title");
      content = intent.getStringExtra("content");
    }
    else if (Intent.ACTION_INSERT.equals(action) {
      title = "new"; content = "memo";
    }
    // Display title, content
  }
}
```

그림 1 인텐트를 사용하는 액티비티

논문에서 제안한 방법은 크게 두 가지로 구성되어 있다. 첫째, 안드로이드 앱의 설정 파일에서 이 앱의 각 컴포넌트에서 기대하는 인텐트 구성을 추출해 인텐트 스펙[3]으로 자동 변환한다. 둘째, 안드로이드 유닛 테스팅 도구[4]에 이 인텐트 스펙을 전달하여 테스트 인텐트를 생성하고 해당 앱 컴포넌트에 전달해 자동으로 테스트한다.

2장에서 기존 연구 결과를 비교 살펴보고, 3장에서

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

인텐트 스펙 기반 안드로이드 앱 자동 검사 방법을 제안한다. 4장에서 상용 앱에 이 방법을 적용하여 테스트한 실험 결과를 제시하고, 5장에서 결론을 맺는다.

2. 관련 연구

첫째, 인텐트 취약점을 지적한 두 가지 연구 사례와 비교한다. 랜덤 생성한 인텐트를 컴포넌트에 전달해서 앱이 비정상 종료되는지 테스트하는 랜덤 인텐트 테스트 도구 Intent fuzzer에 대한 연구가 있다[1]. 앱 바이너리 코드를 분석하여 잠재적으로 인텐트 취약점이 있는 부분을 리포트하는 도구 ComDroid[2]가 있다.

기존 연구에서는 테스트 인텐트를 무작위로 만들거나 정적 분석의 정확성이 낮은 분석 정보에 따라 테스트 인텐트를 만들었지만, 이 연구에서는 안드로이드 앱 설정 파일에서 각 컴포넌트에서 요구하는 인텐트 구성을 자동으로 추출해서 테스트 인텐트 데이터를 만드는 차이점이 있다. 제안한 방법은 자동화가 쉽고 컴포넌트가 기대하는 인텐트 모양을 복잡한 분석 없이 간단하게 파악하는 장점이 있다.

둘째, 인텐트 취약점에 대응하는 두 가지 기존 연구와 차이점을 살펴본다. 프로그래머가 인텐트 스펙을 직접 작성해 각 컴포넌트 시작 전에 전달된 인텐트를 검사하는 방법이 있다[3]. 인텐트 스펙이 주어졌을 때 이 모양대로 테스트 인텐트 데이터를 생성해 자동 테스트하는 안드로이드 유닛 테스트 도구에 대한 연구가 있다[4].

기존 연구[3,4]에서는 개발자가 인텐트 스펙을 작성하는 가정을 했으나, 이 연구에서는 안드로이드 바이너리 앱의 설정 파일에서 인텐트 스펙을 자동으로 만드는 아이디어를 제안한다.

3. 인텐트 스펙 자동 생성 및 인텐트 스펙 기반 테스팅

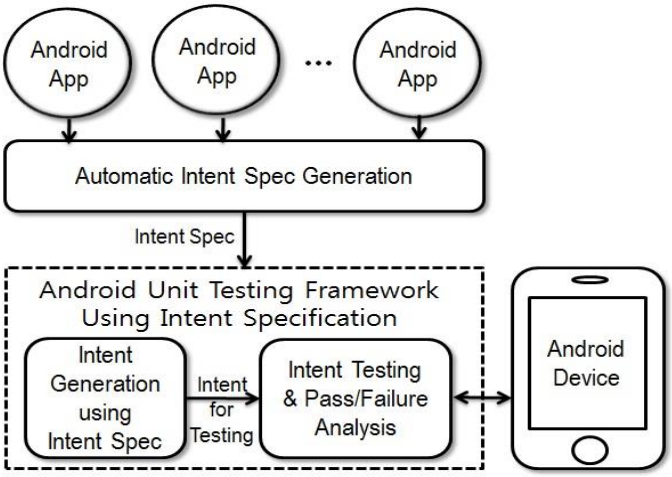

그림 2 인텐트 스펙 자동 생성 및 테스팅

이 논문에서 제안한 인텐트 스펙 기반 안드로이드 앱 자동 검사 방법은 [그림 2]와 같다. 안드로이드 바이너리 앱으로부터 인텐트 스펙을 자동으로 만들어내는 부분과 이 인텐트 스펙에 따라 테스트 데이터를 만들어 인텐트 테스팅을 진행하는 것으로 구성되어 있다.

3.1 인텐트 스펙 자동 생성 방법

안드로이드 앱은 ZIP 압축 형식의 APK 확장자를 갖는 바이너리 파일이고, 압축을 풀면 앱 코드와 함께 앱 설정 파일(AndroidManifest.xml)을 포함한다. 이 설정 파일은 앱에 포함된 컴포넌트들 목록과 각 컴포넌트가 기대하는 인텐트 구성(Intent Filter)가 기술되어 있다. 예를 들어, [그림 1]의 Note 액티비티에 대해 설정된 인텐트 구성은 [그림 3]과 같다.

```
<intent-filter>
 <action name="android.intent.action.EDIT" />
 <category name="android.intent.category.DEFAULT" />
</intent-filter>
<intent-filter>
 <action name="android.intent.action.INSERT" />
 <category name="android.intent.category.DEFAULT" />
</intent-filter>
```

그림 3 Note 액티비티[그림 1]의 인텐트 구성 정보

안드로이드 앱을 작성할 때 [그림 3]과 같은 인텐트 구성을 포함한 앱 설정을 프로그래머가 작성하고, 앱을 모바일 디바이스에 설치할 때 안드로이드 플랫폼이 이 설정 정보를 관리한다.

이 논문에서 제안한 자동 인텐트 스펙 생성 방법을 이용하여 [그림 3]과 같은 인텐트 구성으로부터 자동으로 [그림 4]와 같은 인텐트 스펙을 만든다.

```
    { act = android.intent.action.EDIT
      cat=[android.intent.category.DEFAULT] }
||  { act = android.intent.action.INSERT
      cat=[android.intent.category.DEFAULT] }
```

그림 4 [그림 3]으로부터 자동 생성한 인텐트 스펙

인텐트 스펙은 인텐트에 포함되어야 할 데이터 또는 데이터 타입이 무엇인지 기술한다. 인텐트 스펙과 문법에 대한 자세한 설명은 [3,4]를 참고한다.

[그림 4]는 Note 액티비티가 기대하는 인텐트를 표현하는 인텐트 스펙이다. 이 인텐트는 action(act)이 *EDIT*이거나 *INSERT*이고, action이 *EDIT*인 경우는 DEFAULT 카테고리 속성과 노트 마임타입(mime-type)을 갖는다. action이 INSERT인 경우도 유사하게 설명할 수 있다.

3.2 인텐트 스펙 기반 안드로이드 유닛 테스트

앞에서 설명한 방법에 따라 생성한 인텐트 스펙에

따라 테스트용 인텐트를 생성하고 해당 앱을 테스트한다.

먼저, 인텐트 스펙이 주어지면 이 스펙에 부합하는 정도에 따라 세 가지 종류의 테스트용 인텐트를 생성한다.

1) Compatible : 인텐트 스펙의 필드와 값에 맞춰 생성하되 새로운 필드도 추가할 수 있다.

2) Shape-Compatible : 인텐트 스펙의 모든 필드는 갖추되 필드 값은 랜덤으로 생성한다. 그리고 새로운 필드도 추가한다.

3) Random : 인텐트 스펙과 무관하게 랜덤 생성된 인텐트다.

[그림 5]는 [그림 4]의 인텐트 스펙으로 생성된 테스트용 인텐트의 예를 보여준다.

```
▶ Compatible
{ act = android.intent.action.EDIT
 cat = android.intent.category.DEFAULT ... }
▶ Shape-Compatible
{ act = android.intent.action.ADD
 cat = [ttoIjEWJnpk, vYQEpERvvb] ... }
▶ Random
{ dat = tel:123 cat = [ttoIjEWJnpk, vYQEpERvvb, xpWj_Q,
 android.intent.category.APP_CALENDAR] ... }
```

그림 5 인텐트 스펙으로 생성된 랜덤 인텐트

이렇게 생성한 인텐트를 안드로이드 디바이스의 해당 앱에 전달해 실행하고 앱의 정상 동작 또는 비정상 종료를 안드로이드 유닛 테스트 도구를 통해 자동 판단한다.

4. 실험

상용 안드로이드 앱 10개를 선정하여 이 논문에서 제안한 인텐트 취약점 테스트 방법을 적용하여 실험하였다.

표 1 인텐트 스펙 생성 및 테스트 실험결과

안드로이드앱	자동 생성한 인텐트 스펙	테스트한 인텐트 개수	발견한 오류 개수
비트윈	38	1140	2
CGV앱	121	3630	11
페이스북앱	153	4560	0
네이버앱	108	3240	0
모두의마블	8	240	0
문+리더	6	180	0
아프리카TV	17	510	2
연세앱	7	210	0
우리은행앱	12	360	0
카카오톡	91	2730	12

표 1의 첫 번째 칼럼은 실험한 안드로이드 앱 이름이고, 두 번째 칼럼은 각 앱의 설정 정보로부터 자동 생성한 인텐트 스펙의 개수, 세 번째 칼럼은 이 인텐트 스펙으로부터 생성한 테스트용 인텐트 개수, 마지막 칼럼은 각 앱에서 인텐트 취약점이 있어 비정상 종료한 경우의 전체 개수이다.

요약하자면, 전체 10개의 앱에서 561개 인텐트 스펙을 자동으로 생성하고 이 스펙에 따라 16830개 테스트 인텐트 데이터를 만들어 적용한 결과 4개 앱에서 인텐트 취약점으로 비정상 종료하는 것을 발견하였다.

실험에서 관찰한 앱의 비정상 종료 상황에서 널 참조 예외, 런타임 예외, 불법 인자 예외가 발생하였다. 모두 Java 언어의 비 검사 예외(Unchecked Exception)로 컴파일 시점에 반드시 처리하지 않아도 되는 예외들이다.

인텐트 1개를 테스트하는데 걸리는 시간은 대략 5초이고 전체 16830개 인텐트를 모두 테스트하는데 약 24시간 소요되었다. 이 모든 과정을 자동으로 수행하였다. 다만, 안드로이드 유닛 테스트 도구에서 자동으로 분류한 오류들 중에서 동일한 오류들을 그룹으로 묶어 가려내는 작업은 수작업으로 진행하였다.

5. 결론

안드로이드 앱에 포함된 앱 설정 정보로부터 자동으로 인텐트 스펙을 생성하는 방법을 제안하였고, 안드로이드 유닛 테스트 도구에 이 인텐트 스펙으로부터 생성한 인텐트로 테스트하는 방법을 제안하였다. 상용 앱 10개에 적용해 실험한 결과 인텐트 취약점이 있는 앱을 찾을 수 있었다.

참고문헌

[1] Raimondas Sasnauskas and John Regehr, “Intent Fuzzer: crafting intents of death”, In Proc. of the Joint Int’l Workshop on Dynamic Analysis and Software and System Performance Testing, Debugging, and Analytics, 2014.

[2] Erika Chin, Adrienne porter Felt, Kate Greenwood, and David Wagner, “Analyzing inter-application communication in android”, In Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services, pages 239-252, ACM, New York, NY, USA, 2011.

[3] 고명필, 최광훈, 창병모, “강건한 안드로이드 앱을 위한 실행시간 인텐트 스펙 검사 방법,” 소프트웨어공학 학술대회, 2015 년.

[4] 고명필, 최광훈, 창병모, “인텐트 스펙 기반 안드로이드 유닛 테스팅 프레임워크 설계와 구현,” 한국컴퓨터종합학술대회(KCC), 2015 년.