ScanDal: 안드로이드 개인정보 누출 분석기에서 악성 앱 분석기로 확장하기

김진영 윤용호 이승중
서울대학교 프로그래밍 연구실

ROSAEC center
Research On Software Analysis for Error-free Computing

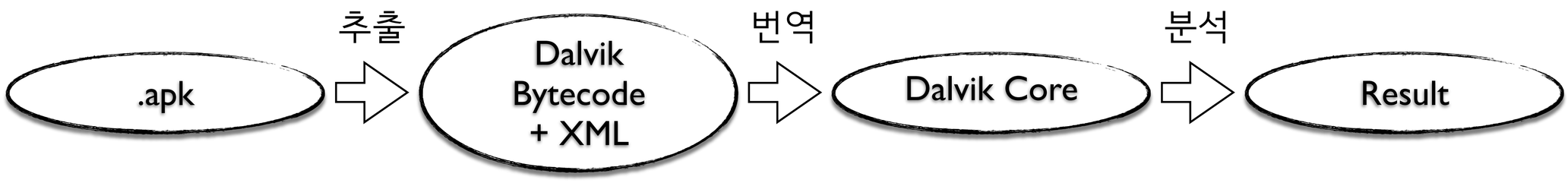
개인정보 누출 분석기 (MoST'12)

문제

- 앱이 사용자의 민감한 개인 정보를 외부로 내보내지 않는가?
  - 단말기 식별번호, 위치정보 등 많은 누출 사례

해결 방법

- ScanDal: 요약해석 기반의 정적 분석기

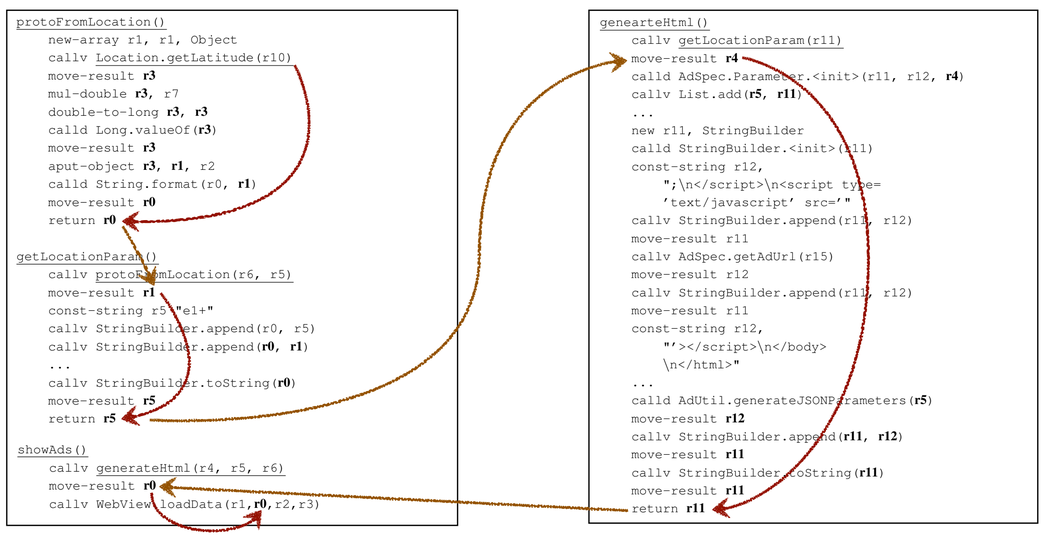


- 앱 패키지 파일을 입력으로 (.apk)
  - 소스 코드 없이 분석
  - Dalvik VM Bytecode가 대상언어
  - 코드에 없는 XML 정보도 이용
- 분석용 중간 언어 Dalvik Core로 번역
  - 200여개 Dalvik 명령을 10여개 핵심 명령으로
  - 앱 구성요소의 실행흐름은 명시되어 있지 않음
  - 2단계 실행모델: 초기화 - (이벤트/콜백)*
- 요약실행으로 분석 대상 성질 확인
  - 실행의미, 요약실행의미 정의
  - Dalvik Core 언어에 대해 안전
  - 고정점 계산 결과를 보고 확인

- 개인정보 누출의 흐름을 분석
  - 개인정보 Source API로부터
    - 위치정보 - `getLastKnownLocation()`, ...
    - 단말기 식별번호 - `getDeviceId()`, ...
  - Sink API로 나가는 흐름이 있는지
    - 네트워크 - `loadURL()`, ...
    - 네트워크/파일 - `write()`, ...
    - SMS - `sendTextMessage()`, ...
- 요약실행으로 어떻게 분석?
  - 값을 만드는 데 기여한 개인정보 Source API들을 안전하게 기록

- ScanDal의 기존 틀을 유지
  - 특정 악성 앱에 특화되지 않는 일반화된 분석
  - 이를 위해 요약실행기와 후처리기를 분리
- 악성 앱을 유연하게 감지할 수 있도록
  - 악성 앱의 행동을 질의 형태로 받은 뒤
  - 이를 요약실행의 고정점 계산 결과를 이용해 확인
- 다양한 악성 행동을 감지할 수 있도록
  - 악성 행동을 기술할 수 있는 언어 정의
  - 기술된 악성 행동 포함 여부 확인하는 후처리기 설계

예제



- 개인정보 누출
  - Source와 Sink를 기술하고 그 사이에 흐름이 있음을 기술
- 사용자 몰래 문자메세지 전송
  - 문자메세지 전송을 위해 불려야 하는 이벤트를 기술
  - 해당 이벤트 처리 과정 중 유저 이벤트가 없음을 기술
- Exploit을 이용한 앱의 루트 권한 상승
  - 알려진 Exploit 행동의 핵심을 기술

할 일

- 현재의 앱 실행모델은 지나치게 단순
  - 모든 이벤트를 하나의 묶음으로 처리
  - 액티비티간의 순서 및 액티비티 생애주기 (lifecycle) 고려 없음
- 개선 가능성 충분
  - 2단계 실행모델에서 최대한 개선한 결과 수행시간 18% 감소

- 액티비티 고려하도록 실행모델 개선
  - 2단계가 아닌 액티비티 단위로
  - 문법, 실행의미, 요약실행의미 수정, 인텐트 분석 필요

- 1260개의 악성 앱 확보 (http://malgenomeproject.org/)
  - 살펴보는 중
- 다양한 악성 행동을 기술할 수 있는 언어 정의
- 기술된 악성 행동 포함 여부 확인하는 후처리기 설계

- 일부 악성 행동은 JNI (Java Native Interface)를 사용
  - 권한 상승을 위한 Exploit
  - 필요하다면 ARM 기계어 코드 포함하도록 분석기 확장