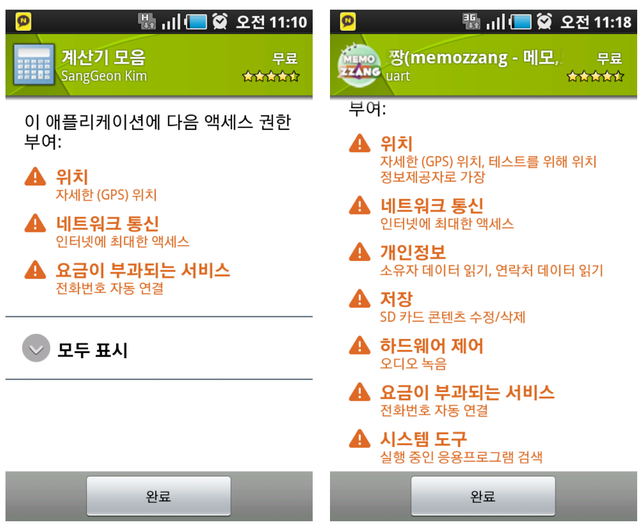
ROSAEC center
Research On Software Analysis for Error-free Computing
소프트웨어 무결점 연구센터 NRF ERC

서울대학교
프로그래밍 연구실

안드로이드 앱 개인정보 유출 여부 분석

김진영 윤용호 이승중

1. 출발점 : 앱들이 수상하다



- 액세스 권한 남용
- 이용자 몰래 나쁜 일을 하는 악성 앱들
- 카메라, SMS, 전화번호부, 위치정보 등을 인터넷, SMS등을 통해 유출 가능

2. 기존의 연구

TaintDroid(동적 분석)
- 개인정보의 소스에 Taint, 싱크를 감시
- OS 내부에 감시 코드 삽입
- 14%의 퍼포먼스 오버헤드 (배터리 추가 소모)

우리의 정적 분석은?
- 개인정보의 소스-싱크 데이터 흐름 분석
- 개별 앱의 코드를 분석, OS와 무관
- 정확도는 조금 떨어지나 오버헤드가 없음

3. 개인정보의 소스 & 싱크

소스	싱크

카메라
`takePicture()`
	인터넷
	`getOutputStream().write()`
	
	SMS
	`sendTextMessage()`

SMS, 전화번호부 `getContentResolver().query()`	
휴대폰 식별번호 `getDeviceID()`	
위치정보 `getLatitude(), getLongitude()`	

4. Dalvik 핵심 언어

move,
move/from16,
move/16, move-wide, move-wide/from16,
...
200여 개의
Davlik Instructions

Data Commands	Control Commands
move	call-direct
istype	call-virtual
new	return
get	throw
put	jmpnz
	skip

원본 바이트코드

```
move-result-object v0
iget-object v1, v3, DDayInfo.calcType
if-eqz v1, 001a // +0012
```

핵심 언어 표현

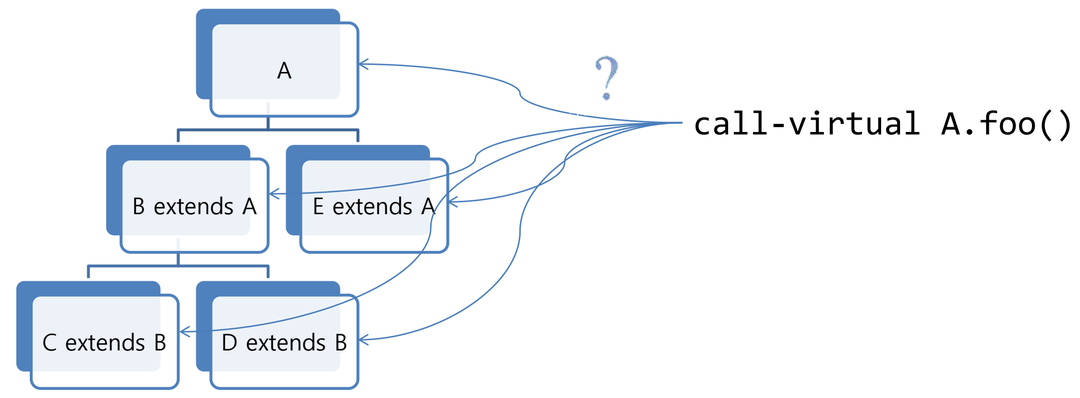
```
move r0 rret
get r1, r3 DDayInfo.calcType
jmpnz r1≠0 001a
```

```
iget-object v1, v3, DDayInfo.calcType
const-string v2, "1" // string@0011
invoke-virtual {v1, v2}, String.equals
```

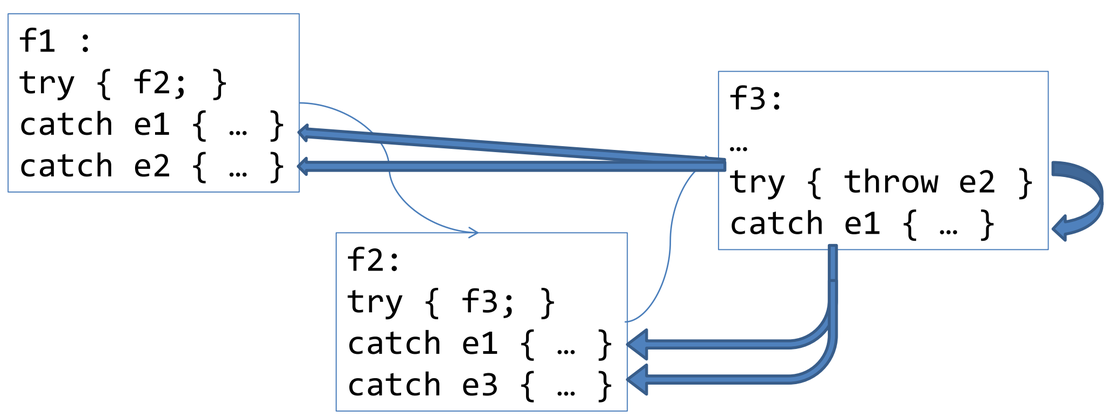
```
get r1 r3 DDayInfo.calcType
move r2 “1”
call-virtual String.equals {v1,v2}
```

5. Dalvik 코드 정적 분석의 특징

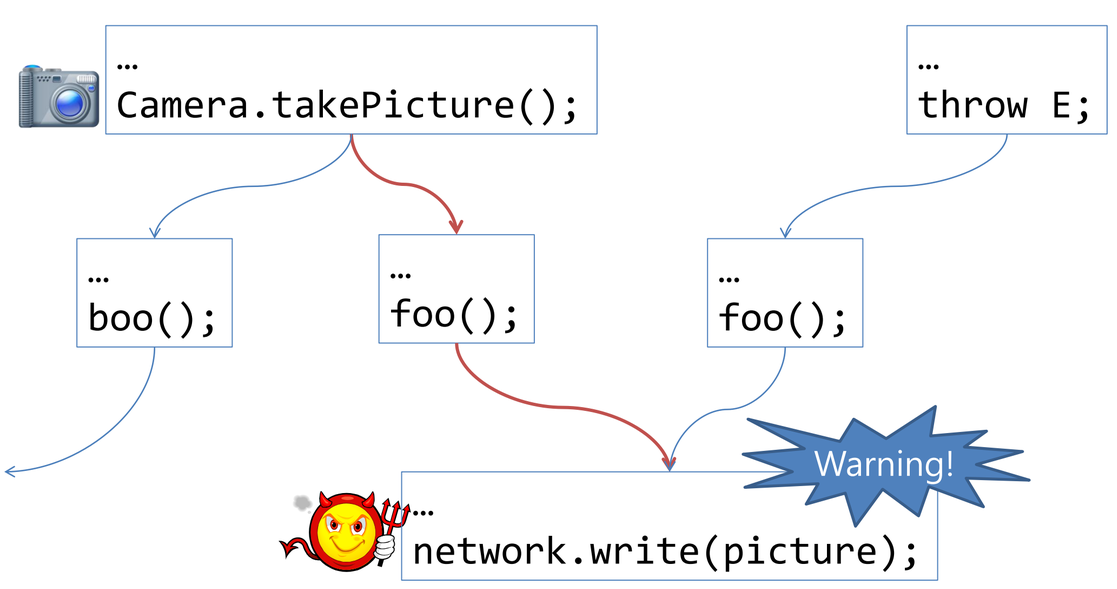
Virtual 메소드 호출



예외 처리



6. 분석 예



7. 진행 상황

개인정보 소스 & 싱크 정리

Dalvik Executable 구조 분석

핵심 언어 정의 및 바이트코드 → 핵심언어 변환 설계

실제 실행의 의미 정의

바이트코드를 핵심언어로 변환하는 전단부 작성

요약 실행의 의미 정의와 안전성 증명

분석기 작성

한 일

하고 있는 일