Kratos: Discovering Inconsistent Security Policy Enforcement in the Android Framework

MAR 1ST, 2016

论文下载：http://web.eecs.umich.edu/~alfchen/roy_ndss16.pdf

Abstract

静态工具，在系统framework层自动寻找访问控制相关的权限泄漏的漏洞。可以分析AOSP或其他定制的Android系统。Android中的访问控制包括：权限检查，UID检查，包名检查，线程状态检查（前台还是后台）。

Methodology

Preprocessing

输入是framework的java编译出的class文件(java编译或odex→smali→dex→class)。通过分析，找到所有系统服务暴露出给APP调用的接口。通过ServcieManager可以找到所有系统服务。之后在服务里找暴露的接口及为保护的receiver。

特别的，全部由native实现的service不多（camera,media player,audio flinger,sound trigger），都是人工分析。

Call Graph Construction

因为是对整个framework构建call graph(Spark),所以采用轻量级的context-insensitive。从上一步每一个入口开始构建图，最后汇总在一起。用到PScout来解决IPC相关的部分。

Call Graph Annotation

标记每条路所做的的安全检查。

Inconsistency Detection

这步最关键，从每个入口开始，生成子图，确定从这个入口到达某个节点所做的所有安全检查。之后两两对比，如果调用了相同的方法缺做了不同的安全检查，就认为存在问题。但这样会误报很多。这里需要做一些处理来减少误报。1）把所有类分为两种：只有系统回调用的类和其他。如果相同的部分不是只有系统才能调用的类，就是不敏感的，当作误报。2）每个类都有成对的功能相反的方法。例如get和set。start和stop.等等，如果对比的两个入口是这种的，当作误报。3）将相关的系统服务进行分类。两个不同类别的入口重复了，当作误报。例如power manager service和SMS servcie的两个入口重复，当作误报。

Result

TABLE I. STATISTICS OF THE SIX CODEBASES IN OUR EVALUATION. WE ONLY CONSIDER SERVICES IMPLEMENTED IN JAVA.

Codebase	# Services	# Service Interfaces		# Class Files
		# AIDL Methods	# Broadcast Receivers	
Android 4.4	70	1,010	26	14,901
Android 5.0	89	1,483	28	33,110
Android 5.1	89	1,510	31	33,433
Android M Preview	89	1,490	31	35,431
AT&T HTC One (Android 4.4.2)	85	1,868	35	17,879
T-Mobile Samsung Galaxy Note 3 (Android 4.4.2)	159	2,463	64	171,306

TABLE II. TIME CONSUMED IN EACH ANALYSIS STEP OF KRATOS (IN SECONDS)

Codebase	Preprocessing	CG Construction	CG Annotation	Inconsistency Detection
Android 4.4	95.4	23.4	8.6	470.3
Android 5.0	137.1	25.0	10.53	496.4
Android 5.1	209.0	22.2	14.6	445.9
Android M Preview	141.6	21.6	9.7	482.3
AT&T HTC One (Android 4.4.2)	110.8	29.1	16.0	655.8
T-Mobile Samsung Galaxy Note 3 (Android 4.4.2)	306.9	57.5	50.7	1273.7

TABLE III. OVERALL RESULTS OF KRATOS. THE NUMBERS OF EXPLOITABLE INCONSISTENCIES, TRUE POSITIVES AND FALSE POSITIVES ARE CONCLUDED BY MANUAL ANALYSIS.

Codebase	# Inconsistencies	# TP	# FP	Precision	# Exploitable
Android 4.4	21	16	5	76.2%	8
Android 5.0	61	50	11	82.0%	11
Android 5.1	63	49	14	77.8%	10
Android M	73	58	15	79.5%	8
AT&T HTC One (Android 4.4.2)	29	20	9	69.0%	8
T-Mobile Samsung Galaxy Note 3 (Android 4.4.2)	128	102	26	79.7%	10

- 误报原因：
  - 调用了相同敏感方法但参数不一样。
  - 遇到虚函数时，采用的是保守的方法，找到所有可能的路径，就会带来误报。
  - 两个功能类似的方法，但不对等。例如deleteHost() and deleteAllHosts()。
- 漏洞不一定能利用：
  - 例如两个接口，需要权限不一致，但都需要系统权限，因此没法用三方APP调用。
  - 两个接口需要权限不一致，但权限少的那个需要传入一个参数，这个参数需要那个缺少的权限才能拿到。
  - 复杂的逻辑难以构造攻击。
- 漏洞的一些特点：
  - 大部分是隐藏的接口，但反射可以调用。
  - 不同的服务可能提供相同功能的API，且要求权限不一致，这属于设计上的失误，功能多余。