一种基于行为链的 Android 应用隐私窃取检测方法

王兆国[1],李城龙[2],张洛什[3],张际宝[3],关 毅[1],薛一波[4]
(1.哈尔滨工业大学计算机科学技术学院,黑龙江哈尔滨 150006;2.国家计算机网络应急技术处理协调中心,北京 100029;
3.哈尔滨理工大学计算机科学与技术学院,黑龙江哈尔滨 150080;4.清华信息科学与技术国家实验室,北京 100084)

摘 要: 本文针对 Android 应用中普遍存在的用户隐私窃取问题,提出了基于行为链的应用隐私窃取行为检测方法,该方法能细粒度地定位 Android 应用中存在的信息泄露源和信息泄露点,利用 WxShall 算法快速计算信息泄漏源和信息泄露点之间的可达性,自动化地追踪 Android 应用中存在的隐私信息传递路径,实现了对 Android 应用中隐私窃取行为的完整检测和分析.对 1259 款应用检测结果表明,本方法正确性超过 95.1%,算法复杂度仅为 WarShall 算法的 5.45%,检测效果优于 Androgurad 和 Kirin.

关键词: Android 系统; 行为链; 隐私窃取; 恶意软件检测
中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112 (2015)09-1750-06
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2015.09.011

A Privacy Stealing Detection Method Based on Behavior-Chain for Android Applications

WANG Zhao-guo[1], LI Cheng-long[2], ZHANG Luo-shi[3], ZHANG Ji-bao[3], GUAN Yi[1], XUE Yi-bo[4]
(1. *Harbin Institute of Technology, School of Computer Science and Technology, Harbin, Heilongjiang* 150006, *China*;
2. *National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing* 100029, *China*;
3. *Harbin Univ. of Sci. & Tech, Computer Science & Technology College, Harbin, Heilongjiang* 150080, *China*;
4. *Tsinghua National Lab for Information Sci. & Tech., Beijing* 100084, *China*)

Abstract: The increasing presence of Android privacy leakages poses a significant privacy risk for Android smartphone users. This paper proposes a privacy leakage detection method based on behavior chain, which can achieve the fine-grained location of the source and points of the information leakage. With the WxShall algorithm, we can calculate the accessibility between the leakage source and leakage points, and detect the transfer path of privacy in Android applications. The detections of 1259 Android applications show that the accuracy of this algorithm reaches 95.1% and the complexity accounts for 5.45% of WarShall algorithm. The results of the experiments demonstrate that the method is better than Androgurad and Kirin.

Key words: Android system; behavior chain; privacy leakage; malware detection

1 引言

移动互联网将移动通信和互联网二者结合起来,成为当今世界发展最快、市场潜力最大、前景最诱人的产业发展方向.中国移动互联网用户已达到 5 亿[1].2013 年 Google Play[2]已经突破百万种应用,国内机锋市场等平台也具有大量的安卓应用.这些应用在给人们带来巨大便利的同时,也带来很大的信息安全隐患和风险.Kaspersky 报告[3]指出:"目前针对恶意软件的样本抽查中,99%都是针对 Android 平台开发的".360 公司报告[4]称:"九成手机恶意软件涉及隐私窃取".因此,快速、准确地自动化检测规模庞大的 Android 应用,保障智能终端的隐私安全,是迫切需要解决的重要问题.

目前,用于检测用户隐私数据泄露的方法可分为两类:静态分析方法和动态分析方法.基于静态特征码的分析方法检测速度快,但是仅能检测已知病毒,具有延迟周期.动态分析方法准确度较高,但是性能开销很大,另外也无法覆盖程序中的所有代码路径,不能查全所有的泄漏路径.

为了克服上述两类方法的缺点,实现对 Android 平

收稿日期:2014-05-12;修回日期:2014-07-01;责任编辑:覃怀银
基金项目:国家科技支撑计划(No.2012BAH42B04)

台下的大规模应用的全部隐私窃取行为的自动化快速检测,本文提出了一种基于行为链的隐私泄露检测方法,该方法将调用关系转化为有向邻接矩阵,通过计算矩阵的可达性来定位隐私信息的窃取路径,进而检测是否有隐私窃取行为,有哪些隐私泄漏路径.该方法的优点一是可以快速地一次检全 Android 应用中的所有隐私泄漏路径;二是可以实现检测的自动化,大幅提高检测效率.

本文的主要贡献如下:

(1)提出了基于行为链的 Android 应用隐私窃取行为检测方法;

(2)提出了 WxShall 可达矩阵的计算算法,通常情况下,该算法与经典的 Warshall 算法相比,可减少 95% 的计算量,并且可一次性检全隐私泄露路径;

(3)对 1259 个安卓应用进行实际测试,发现超过 70.7% 的应用存在隐私泄露问题.

2　相关工作

本节详细介绍国内外相关研究工作.

(1)静态分析方法

静态检测方法以基于权限的检测方法[5]为主要代表.2009 年 Enck 等人提出了 Kirin 检测系统,增强了 Android 权限模型[6],指出了敏感权限的组合可能导致隐私泄露问题;2010 年,Barrera 等人提出了用数据挖掘的方法分析权限间的相关度,从而找出一些与隐私泄露相关度很高的权限集[7];除了基于权限的分析方法,学术界还提出了基于控制流的分析方法和基于数据流的分析方法[8],用于隐私窃取行为的检测.2012 年,Arden O 等人[9]提出了基于语言设计的信息流控制方法,通过对现有编程语言进行扩展、对关键信息添加标签,进而对信息流进行监控实现安全防护.

(2)动态分析方法

在动态分析方法方面,除了传统的沙盒技术[10]和动态污点跟踪技术[11]等之外,2010 年,Nauman M 等人基于 Apex 方法[12]提出动态管理权限的方法保证安全.2011 年,Zhou Y 等人基于 TISSA[13]方法提出动态的管理权限方法解决隐私窃取行为;Enck 等人于 2014 年提出了 TaintDroid 的动态检测机制[14],但 TaintDroid 性能开销较大,且无法覆盖到程序中的所有代码路径.2011 年 Beresford 等人提出了改进的 MockDroid[15]方法.2011 年,Hornyack 等人提出 AppFence 方法[16],在动态运行时通过粗粒度和细粒度两种方式监测敏感数据的传播,并在不影响程序正确运行的基础上,最大程度地保护用户隐私.

3　基于行为链的 Android 应用隐私窃取行为的检测方法

本节首先定义相关概念明确解决对象,接着以一个案例揭示隐私窃取行为的过程,提出基于隐私窃取行为链的检测方法,并详细说明实现过程.

3.1　隐私窃取的相关定义

目前隐私信息、隐私窃取行为、隐私窃取行为检测还没有统一的标准定义,为了减少二义性首先定义相关概念.

定义 1　隐私信息　移动终端中存储的联系人信息、短信内容、彩信内容、通话记录、存储文件、账号密码信息、GPS 定位信息、Wi-Fi 定位信息、通话内容、已安装应用信息、按键信息、用户日历数据、当前任务信息、设备信息等相关信息通称为隐私信息.

定义 2　隐私窃取行为　隐私窃取行为是指在用户没有许可或不知情的情况下,有意获取用户的隐私信息,可能用于直接或间接地获取某种不当利益.

定义 3　隐私窃取行为检测　隐私窃取行为检测是指通过某种或者多种方法的组合来检测 Android 应用中是否存在隐私窃取行为.

3.2　隐私窃取行为的特点

为了展示隐私窃取行为的特点,通过对一款用于窃取用户隐私信息的恶意应用 Santander.apk[17]的深入分析进行说明.该款软件主要是在办理网上银行业务时,通过动态获取临时密码来保障账户安全,然而其假借保障安全之名,实则暗地窃取用户敏感信息.通过对该 app 反编译得到的关键源码如算法 1 所示.

算法 1　AndroidManifest 源码

```
1  Class MainApplication
2    public void onCreate(){
3      imei = Settings.getImei(this);//窃取用户设备隐私信息
4      imsi = Settings.getimsi(this);//窃取用户通讯录
5       pendingintent = PendingIntent.getBroadcast(this,0x193dac6a,new
            Intent(this),0);
6      alarmmanager = (AlarmManager)getSystemService("alarm");}//定
       时启动
7    public void onStart(Inten paramIntent,int paramInt){
8      if ((localBundle != null) && (localBundle.get("key") != =
           null)){
9        new Thread(new ThreadOperation(this,1,null)).start();   }//
         接受控制指令 1
10   }
```

其中,第 3、4 行中包含 getDeviceID 和 getLine1Number 为获取用户的设备信息和通讯号码;第 9 行向远端服务器,发送截获到的短信内容、设备信息和通讯号码.由上述分析可知,该恶意软件窃取了用户设

备信息、通讯号码、短信等,存在严重的隐私窃取行为,对用户的安全产生了极大的危害和风险.

通过对隐私窃取过程分析,可以得到该款恶意软件隐私窃取的调用流程图,如图 1 所示.图中节点表示函数,有向边表示函数之间调用关系,右侧方框节点表示涉嫌隐私窃取函数,左侧方框节点表示涉嫌隐私泄露函数,关联节点和有向边表示传递的关键路径.由此可见,隐私窃取行为按先后顺序包括三个过程:隐私数据窃取、隐私数据传递、隐私数据泄漏,大多数的隐私窃取行为也都符合这个行为特征.

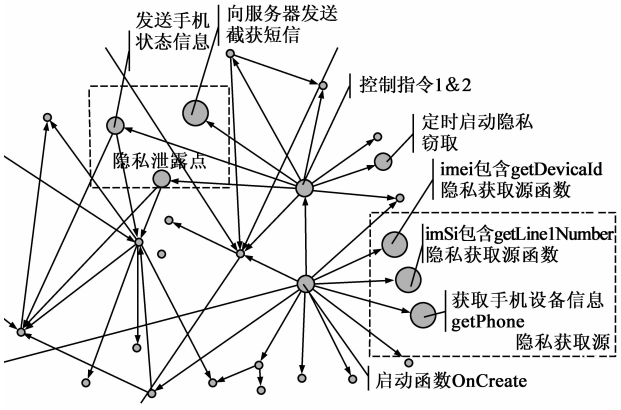

图1 Santander.apk隐私窃取调用流程图

3.3 隐私窃取行为链

将隐私窃取行为的三个过程进一步抽象可以得到隐私窃取行为状态.每一个状态对应一个行为过程,状态之间存在着严格的时间序列,状态之间构成了一个完整的链,可称之为“隐私窃取行为链”.这个“隐私窃取行为链”可精确刻画 Android 应用的隐私窃取过程.

3.4 基于行为链的检测方法

对应隐私窃取行为链状态,将检测过程分为三个环节:隐私数据窃取检测、隐私数据泄露检测、隐私传递路径检测.

(1)相关术语和符号

为了更好地表述隐私窃取行为链,我们定义了一些术语和符号,如表 1 所示.

表 1 隐私窃取行为链术语和符号

符号	解释
$F_{(n)}=\{F_{(1)},F_{(2)}\cdots,F_{(n)}\}$	应用中使用的函数集合(“函数”指 java 语言中 method)
$C_{(i;j)}:F_{(i)}\rightarrow F_{(j)}$	函数 i 到函数 j 的调用关系
$F_{(S)}$	隐私获取过程中使用的函数集合
$F_{(T)}$	隐私泄露过程中使用的函数集合
$P_{(S\rightarrow T)}$	$P_{(S\rightarrow T)}$指隐私数据从获取到泄露的传播路径,即隐私窃取行为链

(2)隐私信息窃取检测

隐私信息窃取检测的目标是检测 Android 应用中是否获取用户隐私信息 $F_{(S)}$,主要检测以下隐私信息并获取使用的函数:读取联系人信息、读取短信/彩信内容、获取通话记录、读取账号密码信息、位置信息等,通过对 ADK 中 API 的统计形成共 11 大类,191 种隐私获取函数.

(3)隐私泄露点检测

隐私泄露点检测是检测隐私信息泄露过程使用的函数集合,主要包含 Internet 链接、Wi-Fi 蓝牙和短消息等泄露方式,共 118 种隐私泄露函数.

(4)隐私信息传递路径检测

隐私信息获取和隐私信息泄露为隐私窃取行为链的始末,更重要的是分析隐私信息的传递过程,即隐私信息行为链 $P_{(S\rightarrow T)}$.在一个应用中可能存在多条传递路径,如何能把所有的传递路径快速地找到,是该方法的重点和难点.为了解决该问题,这里引入调用关系 $C_{(i;j)}$与布尔矩阵 $\boldsymbol{P}$ 来求解隐私信息传递路径,如定义 4 所示.

定义 4 设 $G_c=(V,E)$是一个 n 阶的简单有向图,定义有向图 G_c的有向关系矩阵 $\boldsymbol{P}=(\boldsymbol{p}_{ij})_{n\times n}$如式(1)所示,本文称 $\boldsymbol{P}$ 是图 G_c的可达矩阵:

$$P_{i,j}=\begin{cases}1, F_{(i)}\text{到 }F_{(j)}\text{存在非零有向通路}\\0,\text{其他}\end{cases}\tag{1}$$

(5) $\boldsymbol{W}$ 可达矩阵与可达向量

为了实现自动化隐私窃取行为链的检测,本文提出的 WxShall 算法.通过计算可得集合 $F_L=\{F_1,F_2,\cdots,F_{b-1},F_b\}$(其中 $a+b=m$, F_a表示检测的隐私获取函数, F_b表示检测隐私泄露使用的函数)中每一元素与其它节点的可达性,最终生成可达关系矩阵 $\boldsymbol{Q}=(\boldsymbol{q}_{\alpha,\beta})_{n\times n}$.本文将该矩阵称为 $\boldsymbol{W}$ 可达矩阵,如式(2)所示(其中 $\alpha\in\{1,2,\cdots a\}$, $\beta\in\{1,2,\cdots b\}$, $a+b=m$):

$$q_{\alpha,\beta}=\begin{cases}1, & F_aF_\beta\text{ 路径}\\0, & \text{其他}\end{cases}\tag{2}$$

由 $\boldsymbol{W}$ 可达矩阵可以得到 Android 应用中使用的每一个隐私获取函数和隐私泄露函数对应的可达向量 $\boldsymbol{V}=(\boldsymbol{V}_1,\boldsymbol{V}_2,\cdots,\boldsymbol{V}_m)$,其中 $\boldsymbol{V}$ 对应于 $\boldsymbol{W}$ 可达矩阵的行,表达函数与其它函数的存在的调用关系信息.

接下来利用 WxShall 算法计算隐私获取源和隐私泄露点与其它函数节点的可达性,其伪代码如算法 2 所示:

算法 2 WxShall 算法

```
1  BEGIN
2  初始化函数调用关系 X_m,初始化二维数组 G,确定函数数量 n;
3  while 未满足停止条件 do
4     for 关系 X_m(i) do
5        根据调用关系构建 G;
```

```
    end for
初始化敏感函数库 W_u;
for W_u(i) do
  检索是否存在敏感函数,确定敏感函数数量 m;
end for
for m(i) do
  for n(j) do
    if G[i,j]存在可达路径 then
      for n(k) do
        if G[j,k]存在可达路径 && G[i,k]不存在可达路径
        && k 小于 j then
          for n(l) do
            G[i,l] = G[i,l] or G[k,l];
          end for
        end if
        else
          G[i,k] = G[i,k] or G[j,k];
        end else
      end for
    end if
  end for
end for
输出匹配结果
END
```

按照表 2 所示 WxShall 算法流程,对该算法的时间复杂和空间复杂度分析,得到 WxShall 可达矩阵算法最差时间复杂度 $mO(|n|^2)$、最优时间复杂度 $m\Omega(|n|^2)$、最差空间复杂度 $mO(|n|^2)$.

4　实验与评估

为了验证基于隐私窃取行为链的检测方法的有效性和正确性,本文对比其它 Android 恶意应用检测方法,实验证明了本文提出的检测方法效果更优.

4.1　实验环境与实验组成

为了更好表述本文的实验过程,下面简述实验环境、实验样本与实验组成.

4.1.1　实验环境与实验样本

实验在内存 8GB、处理器为 Intel i3 2120CPU @ 3.3GHz 的机器上完成.检测算法由 Python 语言实现,爬虫程序由 Java 语言实现.实验样本使用文献[18]中提供的 1259 款典型恶意应用.

4.1.2　实验组成

实验由 3 部分组成:

(1)检测文献[18]已经标定结果的 697 款恶意应用,并与其标定结果对比,证明本文检测方法的正确性;

(2)分析 WarShall 算法与本文 WxShall 算法的时间/空间复杂度,证明本文提出算法的有效性与可行性;

(3)对文献[18]提供的 1259 个恶意应用,综合对比本文方法与 Androguard(基于静态特征码)、Kirin(基于静态权限),证明本文方法在解决隐私窃取问题方面的优势.

4.2　实验评估

对同一数据样本,对比分析了本文方法与文献[18]、Androguard 和 Kirin 检测方法的实验结果.实验结论的详细评估在正确性验证、有效性验证与可用性验证中详细说明.为了对实验进行评估,使用如下的度量指标:

正确率 = 检测结果正确数量/检测总数量 × 100%

漏检率 = 应该检测出、但未检测出数量/检测总数量 × 100%

误报率 = 错误检测数量/检测总数量 × 100%

4.2.1　正确性验证

本文的检测结果与文献[18]中对恶意软件的标定结果对比(共 697 款已标定结论的恶意应用),共有 34 处检测结果不一致,检测结果如表 2 所示.

表 2　正确性验证结果

检测 APP 总数	697
漏检数量	14
误报	20
漏检率	2.0%
误报率	2.87%
正确率	95.1%

由此可见基于隐私窃取行为链的检测方法准确率超过 95%,证明了本方法正确性.

4.2.2　有效性验证

基于隐私窃取行为链检测方法,其系统的性能瓶颈在于可达矩阵的求解过程,因此本节通过对比可达矩阵求解算法的时间复杂和空间复杂度来评估本方法的有效性.

文献[19]介绍了经典 Warshall 算法用以计算二元关系的传递闭包运算,并给出该算法的最差时间复杂度为 $O(|V|^3)$、最优时间复杂度为 $\Omega(|V|^3)$、最差空间复杂度为 $\Theta(|V|^2)$.依据上述分析如图 2 所示,Warshall 和 WxShall 两种算法的复杂度与一款软件中函数的总数量、敏感源函数数量有关.当一款软件中函数的总数量越多时,WxShall 算法的优越性越明显.通过统计分析,覆盖 96% 的 Android 应用的敏感源函数数量(包含获取和泄露)为 30 以下,80% 的 Android 应用使用函数数量在 550 以上.因此,设置敏感源函数数量为 30,函数数量 550 时,WxShall 算法计算次数仅为 Warshall 算法计算量的 5.45%.通常情况下,待检测的 Android 应用的数量均大大超出 550 个,此时 WxShall 算法优势更加明显.由此可见 WxShall 算法更适用于自动化恶意行为分析,证明了本文方法的有效性.

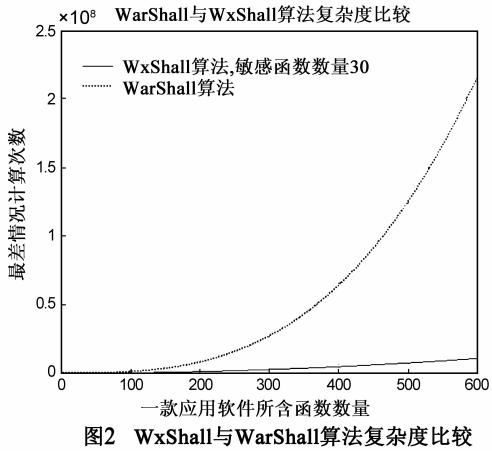

图2 WxShall与WarShall算法复杂度比较

4.2.3 可用性验证

可用性验证结果如表 3 所示. Kirin[6]方法基于 9 条权限规则,检测率仅仅为 2.87%,因此可见基于权限的模型误报和漏报严重.

表 3 可用性验证结果

	Kirin	Androgurad
总检测数量	1259	1259
正确检测数量	36	755
正确检测率	2.87%	59.96%

而 Androguard[20]检测方法可检测出 755 款恶意应用,检测率为 59.96%.但是只能检测已知的恶意应用,且无法明确标记出隐私窃取行为和所属类别,所以在隐私窃取行为检测中,Androguard 效果不佳.由此可见,本文方法相比 Androguard、Kirin 方法更适合 Android 应用软件的隐私窃取检测.

5 结论

随着移动互联网快速发展,Android 应用隐私泄露问题日益严重.为了快速有效地检测 Android 应用的隐私窃取行为,本文首次提出了基于行为链的检测方法,该方法具有很好有效性和可用性,能够准确识别 Android 应用存在的隐私窃取行为.本文对 WarShall 算法进行了优化,提出了 WxShall 算法.WxShall 算法一次计算便可检测出包含表 2 中的 11 类隐私获取源、表 3 中的 4 类隐私泄露点、118 种 API 函数复杂组合情况下的隐私信息窃取行为,可有效解决复杂情况的隐私窃取路径识别问题.

未来将一方面进行更多应用的检测,并跟其它方法进行详细的比较;另一方面,将进一步优化算法和系统,将隐私泄露模型扩展到其它恶意行为的检测与分析.

参考文献

[1] CNNIC. 第 31 次中国互联网络发展状况统计报告[EB/OL]. http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/201403/t20140305_46239htm,2014-03-05.

[2] Google Play. Google Play Store [EB/OL]. https://play.google.com/store,2014-07-01.

[3] Kasperskylb. Careless Majority More Than Half Use No Security Software on Their Android-based Devices[EB/OL]. http://www.kaspersky.com/about/news/press/2013/careless_majority_more_than_half_use_no_security_software_on_their_android_based_devices,2013-09-27.

[4] 李燕博. 360 手机安全中心:九成以上 Android 恶意软件窃取用户隐[EB/EL]. http://news.xinhuanet.com/tech/2013-08/12/c_125151578.htm,2013-8-12.

[5] Felt A P, Chin E, Hanna S, et al. Android permissions demystified[A]. Proceedings of the 18th ACM Conference on Computer and Communications Security[C]. New York: ACM, 2011. 627-638.

[6] Enck W, Ongtang M, McDaniel P. On lightweight mobile phone application certification [A]. Proceedings of the 16th ACM Conference on Computer and Communications Security [C]. New York: ACM, 2009. 235 – 245.

[7] Barrera D, Kayacik H G, van Oorschot P C, et al. A methodology for empirical analysis of permission-based security models and its application to Android [A]. Proceedings of the 17th ACM Conference on Computer and Communications Security [C]. New York: ACM, 2010. 73 – 84.

[8] Grace M, Zhou Y, Zhang Q, et al. Riskranker: scalable and accurate zero-day Android malware detection[A]. Proceedings of The 10th International Conference on Mobile Systems, Applications and Services[C]. New York: ACM, 2012. 281 – 294.

[9] Arden O, George M D, Liu J, et al. Sharing mobile code securely with information flow control[A]. Security and Privacy (SP), 2012 IEEE Symposium on IEEE[C]. San Francisco: IEEE, 2012. 191 – 205.

[10] Blasing T, Batyuk L, Schmidt A D, et al. An Android application Sandbox system for suspicious software detection [A]. Malicious and Unwanted Software (MALWARE), 2010 5th IEEE International Conference on [C]. Nancy, Lorraine: IEEE, 2010. 55-62.

[11] 杨广亮,龚晓锐,等. 一个面向 Android 的隐私泄露检测系统[J]. Computer Engineering, 2012, 38(23): 1 – 6.

[12] Nauman M, Khan S, Zhang X Apex: Extending Android permission model and enforcement with user-defined runtime constraints[A]. Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security [C]. New York: ACM, 2010. 328 – 332.

[13] Zhou Y, Zhang X, et al. Trust and Trustworthy Computing [M]. Springer Berlin Heidelberg: Springer, 2011. 93 – 107.
[14] W Enck, P Gilbert, B Chun, L Cox, J Jung, P Mc Daniel, A Sheth. TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones[A]. Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation[C]. Berkeley: USENIX, 2010. 1 – 6.
[15] Beresford A R, Rice A, Skehin N, et al. MockDroid: Trading privacy for application functionality on smartphones[A]. Proceedings of the 12th Workshop on Mobile Computing Systems and Applications[C]. New York: ACM, 2011. 49 – 54.
[16] Hornyack P, Han S, Jung J, et al. These aren't the Droids you're looking for: Retrofitting Android to protect data from imperious applications[A]. Proceedings of the 18th ACM Conference on Computer and Communications Security[C]. New York: ACM, 2011. 639 – 652.
[17] Santander. Resolva On-line [EB/OL]. http://www.santander.com.br/, 2014-03-14.
[18] Zhou Y, Jiang X. Dissecting Android malware: characterization and evolution[A]. Security and Privacy (SP), 2012 IEEE Symposium on[C]. San Francisco: IEEE, 2012. 95 – 109.
[19] Warshalhttp. Floyd-Warshall Algorithm [EB/OL]. https://en.wikipedia.org/wiki/Floyd%E2%80%93Warshall_algorithm, 2014-06-04.
[20] ANDROGUARD. Androguard Tool[EB/OL]. https://code.google.com/p/androguard/, 2014-06-04.

作者简介

王兆国　男,1986 年 4 月出生于黑龙江七台河市.现在为哈尔滨工业大学计算机科学与技术学院博士研究生.主要研究方向为移动互联网安全,网络与信息安全.
E-mail: wangzhaoguo@tsinghua.edu.cn

李城龙　男,1985 年 7 月出生于陕西西安.2007 年、2012 年在清华大学计算机科学与技术学院获得本科和博士学位.现在工作在国家计算机网络应急技术处理协调中心(CNCERT/CC).主要研究方向信息安全和移动互联网安全.
E-mail: lichenglong@cert.org.cn