

基于多因素聚类选择的 Android 应用程序分类风险评估方法

超凡, 杨智, 杜学绘, 韩冰

(信息工程大学, 河南 郑州 450001)

摘要: 大多数现有的 Android 应用程序风险评估根据经验直接指定因素的权重, 通过统计少量因素的使用频率来计算安全风险。提出一种新的 Android 应用风险评估方法, 能够同时提供定量和定性评估。该方法融合系统权限、API 调用、Intent Filter 的 action 属性以及数据流等多种风险因素, 基于因素的风险分类与加成进行风险赋值, 基于层次聚类对因子子集进行权重分配。实验表明, 评估结果能够有效地反映 Android 应用程序的真实安全风险。

关键词: 安卓; 风险评估; 安全威胁; 静态分析; 层次聚类

中图分类号: TP309.2

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2021023

Classified risk assessment method of Android application based on multi-factor clustering selection

CHAO Fan, YANG Zhi, DU Xuehui, HAN Bing

Information Engineering University, Zhengzhou 450001, China

Abstract: Most existing risk assessments of Android applications directly assign weights to factors according to experience, and calculate security risks by counting the frequency statistics of few factors. A new method for risk assessment of Android applications is proposed, which can provide both quantitative and qualitative assessment. This method integrates multiple risk factors such as system permissions, API calls, the action properties of Intent Filter, and data flow. The risks of factors are assigned based on their risk classification and addition, and the weights of factor subsets are distributed based on hierarchical clustering. Experiments show that the assessment results can effectively reflect the real security risks of Android applications.

Keywords: Android, risk assessment, security threat, static analysis, hierarchical clustering

收稿日期: 2020-02-07; 修回日期: 2020-04-12

通信作者: 杨智, zynoh@163.com

基金项目: 国家重点研发计划 (2018YFB0803603); 国家自然科学基金 (61802436)

Foundation Items: The National Key R&D Program of China (2018YFB0803603), The National Natural Science Foundation of China (61802436)

论文引用格式: 超凡, 杨智, 杜学绘, 等. 基于多因素聚类选择的 Android 应用程序分类风险评估方法[J]. 网络与信息安全学报, 2021, 7(2): 161-173.

CHAO F, YANG Z, DU X H, et al. Classification risk assessment method of Android application based on multi-factor clustering selection [J]. Chinese Journal of Network and Information Security, 2021, 7(2): 161-173.

1 引言

典型的 Android 恶意软件检测只能将应用程序分类为良性、恶意或正常、异常,这样粗糙的二元指标无法详细表明应用程序安全面临的具体风险程度。在实际情况下,恶意软件与良性应用的界限通常不是特别鲜明,大量“正常”应用蕴含各种安全风险,可能给用户造成隐私泄露、资费消耗、系统破坏等损失。事实上,不存在完全安全的无风险应用,但风险更小的应用通常表明安全性更高,更值得信赖。安全风险评估是一种有效衡量应用程序安全风险的技术,可帮助市场与用户更好地选择较为安全的应用。

针对 Android 应用的风险评估目标存在多种说法,有风险评分和风险等级^[1]、安全信用指数^[2]、安全和隐私风险^[3]、可信度^[4]、隐私风险评分^[5]等。评估目标不一致,评估要素的选择也不同。在单一准则评估系统中,文献[1]使用申请的权限,文献[6-7]选择了调用的 API 接口。文献[8]将权限集分解为具有恶意倾向的组合权限、“溢权”问题和自定义权限,通过层次分析法来评估应用的风险值。多准则评估系统通常将权限与其他要素相结合,如文献[9]的准则为权限和系统 API 调用,文献[10]采用了权限、Intent action 以及是否使用第三方库。除了对应用程序进行静态分析,一些方法还从应用市场上寻找相关元数据,如文献[2]将权限、认证强度和第三方市场违规记录作为指标,文献[4]则结合了权限、市场收集质量、流行度。

有许多工作通过概率生成^[1]、信息熵^[6]、卡方检验^[9-10]、互信息^[11]等方法,计算要素与应用程序性质的相关性来衡量要素的风险值。文献[7]将 API 调用频率的客观风险与专家主观风险赋值相结合,给出要素的综合安全等级。文献[10]不仅考虑单个要素的风险,还额外考虑了某些危险要素组合的风险。文献[5]和 RiskMon^[12]利用机器学习技术构建回归模型,预测应用程序的风险评分。

挖掘应用程序的两种比较对象之间的关系也是一种有效的风险评估手段。文献[12]衡量用户的大致期望与应用实际行为之间的差异,WHYPER^[13]比较应用程序描述和权限使用之间

的差异,文献[14]挖掘应用类别与权限之间的关系,将同一类别中安全等级相同的应用聚集成种群。上述工作均是针对应用程序的安全风险评估,文献[15]在此基础上识别用户的隐私风险,进而通过大规模样本数据构建中国隐私风险指数体系。

根据以上信息,笔者认为大部分现有的 Android 程序风险评估研究存在以下不足:①评估因素的数量、种类较少;②简单地使用与因素频率相关的统计值来为其进行安全风险赋值;③根据经验判断因素权重,甚至根本不考虑因素权重。为此,本文提出一种 Android 应用程序的安全风险评估框架,从系统权限、API 调用、Intent Filter 的 action 属性以及数据流 4 个方面综合衡量应用程序的安全风险,给出定量和定性的评估结果。本文将所有因素根据操作对象进行分类,并通过对象与不同安全威胁之间的关系进行风险计算,由此得到安全风险值。此外,为了衡量 7 个不同的因素子集在应用总风险计算中所占的比重,本文设计了一种基于层次聚类的因素子集权重分配算法,并在此基础上,给出应用程序的安全风险计算方式。

2 风险因素

2.1 Android 应用程序的安全威胁

参考中国互联网协会反网络病毒联盟制定的《移动互联网恶意代码描述规范》^[16],本文将 Android 应用程序面临的安全威胁分为 8 种,并按危害程度进行排序,如表 1 所示。值得注意的是,这些安全威胁均在用户不知情或未授权的情况下发生。每个 Android 应用程序可能会面临不止一种安全威胁。

恶意软件要实施上述安全威胁,需要对相关事件进行监听或对重要资源进行请求。对于普通应用程序而言,开发人员需要为其功能实现进行相应的声明,虽然主观上不会故意对用户造成安全威胁,但此举同样会给应用程序带来风险。

2.2 风险举例

为了更好地阐述设计思路,本文以图 1 中某种风险行为的代码为例进行说明。该行为主要是从设备的 GPS 硬件读取地理位置,并将相关信息写入短信发送出去。如图 1(a)所示,这一操作首先

表 1 安全威胁概述
Table 1 Overview of security threats

排序	安全威胁	行为描述
1	恶意扣费	通过隐秘执行、诱导用户点击等手段，订购收费业务或支付金额，造成用户经济损失
2	隐私窃取	获取有关用户隐私信息的内容
3	远程控制	能够受到远程控制，接收指令并进行相关操作
4	恶意传播	通过感染、复制、发送、下载等方式扩散恶意代码
5	资费消耗	通过打电话、发短信、发邮件、联网等方式消耗用户资费
6	系统破坏	通过篡改、感染、劫持、删除、终止进程等手段导致设备或设备中其他软件、用户文件等部分或完全无法正常使用
7	诱骗欺诈	通过篡改、伪造、劫持短信、彩信、邮件、通信录、通话记录、收藏夹、桌面等方式诱骗欺诈用户，达到不正当目的
8	流氓行为	对系统没有直接损害，也未对用户信息、资费造成侵害的其他恶意行为

```
protected void onCreate(Bundle savedInstanceState){
    ...
    L1: LocationManager locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
    L2: Location location = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
    L3: double lat = location.getLatitude();
    L4: double lng = location.getLongitude();
    L5: String loc = "纬度:" + lat + "\n 经度:" + lng;
}
public void onClick(View view){
    ...
    L6: SmsManager smsManager = SmsManager.getDefault();
    L7: smsManager.sendTextMessage(phoneNumber, null, loc, null, null);
}
```

(a) Activity 代码片段

```
...
<receiver ...>
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED"/>
        ...
    </intent-filter>
</receiver>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.SEND_SMS"/>
...
```

(b) AndroidManifest.xml 文件片段

图 1 风险行为举例

Figure 1 Example of risky behavior

需要利用 `getLastKnownLocation(LocationManager.GPS_PROVIDER)` (L2) 接口来访问用户的最新地理位置，从而得到此时的经纬度信息 (L3~L5)，然后利用 `sendTextMessage` 函数 (L7) 发送含有该信息的 SMS 消息。因此，存在一条从数据源 `getLastKnownLocation` 到泄露点 `sendTextMessage` 的数据流。除此以外，应用程序需要在配置文件 `AndroidManifest.xml` 中申请相应的权限，如图 1(b) 中 3 个权限分别用于访问 GPS 硬件，确保应

用在开机后自动运行和发送短信。为此，应用程序还需要为一个 `Broadcast Receiver` 组件声明 `action` 属性为 `android.intent.action.BOOT_COMPLETED` 的 `Intent Filter`，以监听系统启动广播并启动实施这一风险行为的 `Activity` 组件。

由此可见，Android 应用程序的风险行为与系统权限、API 调用、`Intent Filter` 的 `action` 属性以及数据流密切相关，因此本文将这 4 个方面作为风险评估的考虑因素。在确定评估因素之后，本文的

Android 应用风险评估系统需要围绕两个方面展开：一是如何为各个因素赋予安全风险值，二是判断不同种类因素对应用总风险造成的影响。

3 系统设计

本文的 Android 应用程序风险评估框架由以下 4 个部分组成，如图 2 所示。

1) 因素提取：给定 Android 应用程序，借助静态分析工具对 APK 文件进行分析，提取与风险评估相关的各种因素。

2) 因素风险赋值：考虑 Android 系统中 25 个安全事件的可操作对象，按其与 8 种安全威胁之间的关联来分别计算风险值。以这些对象为基准将提取的因素分类，由此得到各因素对应的风险值。

3) 因素子集权重分配：由于 7 个因素子集在风险评估中发挥的作用不同，采用一种基于层次聚类的权重分配算法来衡量各因素子集的权重。

4) 应用风险评估：根据因素的分布情况、所属对象的风险值与各因素子集的权重，可以对 Android 应用程序的安全风险进行定性和定量评估。

3.1 因素提取

为了获得 Android 应用程序的相关信息，本文借助静态分析工具 Androguard^[17]和 FlowDroid^[18]对 APK 文件进行分析，提取系统权限、API 调用、Intent Filter 的 action 属性以及数据流这 4 类因素。应用程序在 Android 系统中进行各种安全操作，最终都要通过调用系统 API 来具体实现，受到权限保护的 API 调用往往具有更高的风险性，应用程序需要在配置文件中申请相应的权限。组件通过注册 Intent Filter 来监听关注的事件，action 属性直接表明了要执行的动作名称，这些动作通常会引发一些安全风险。数据流由数据源和数据泄露点组成，分别称为 source 和 sink 函数，它们是一对 API 接口，不同的组合会产生不同程度的安全影响。具体的风险评估因素组成如表 2 所示。

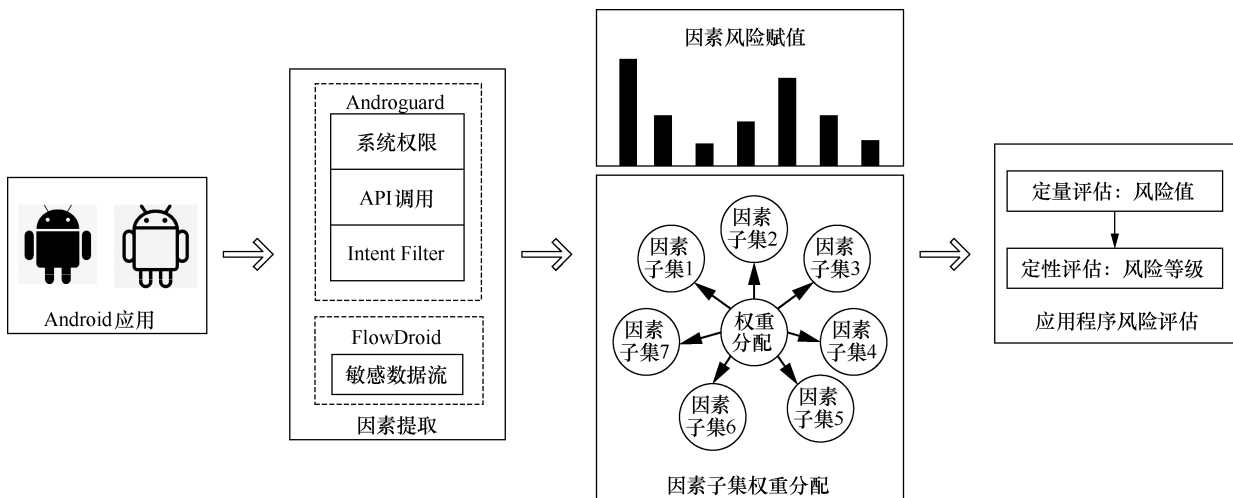


图 2 系统设计
Figure 2 System design

表 2 评估因素组成
Table 2 Composition of assessment factors

因素集	划分标准	因素子集
系统权限	申请的权限	①系统权限
API 调用	申请相关权限并以 API 形式使用	②系统权限的 API 使用点
	未申请相关权限并以 API 形式使用	③系统权限的 API 使用点
Intent Filter	Activity 组件声明	④系统 action
	Service 组件声明	⑤系统 action
	Broadcast Receiver 组件声明	⑥系统 action
数据流	FlowDroid 输出	⑦source-sink 对

3.2 因素风险赋值

根据 Android 安全领域的相关研究, 本文列举了 25 个 Android 系统中与安全事件相关的可操作对象, 涉及 Android 移动安全的方方面面。本文提取的因素与这些对象息息相关, 可以依据对象将相关因素进行分类, 以便更好地为因素赋予安全风险值。

本文主要借鉴《移动互联网恶意代码描述规范》来确定因素风险值。表 1 将 8 种安全威胁按危害程度由大到小进行排序, 本文依次赋予 8-1 的危害等级, 并基于对象与安全威胁之间的关系来判断对象的安全风险。同一操作对象可能受到不止一种安全威胁, 本文将排名最高的安全威胁称为对象的主安全威胁, 它直接决定了对象的风险等级。除此以外, 其余的相关安全威胁也要在对象的安全风险中有所体现。具体来说, 主安全

威胁的危害等级为对象的风险等级; 计算其余安全威胁的危害等级之和占排名在主安全威胁之后所有安全威胁危害等级之和的比重, 称为风险加成。

定义 1 安全威胁集合 $T_i = \{t_1^i, t_2^i, \dots, t_{n_i}^i\}$ 描述与第 i 个对象相关的安全威胁, 其中, t_j^i ($1 \leq j \leq n_i$) 为集合中第 j 个安全威胁的序号。因此, 该对象的风险等级为 $9 - t_1^i$, 风险加成计算公式为

$$RB_i = \frac{(9 - t_2^i) + \dots + (9 - t_{n_i}^i)}{[9 - (t_1^i + 1)] + [9 - (t_1^i + 2)] + \dots + 1} = \frac{2[9(n_i - 1) - (t_2^i + \dots + t_{n_i}^i)]}{(8 - t_1^i)(9 - t_1^i)} \quad (1)$$

根据《移动互联网恶意代码描述规范》中对各安全威胁典型行为的具体描述, 本文将各操作对象对应的安全威胁列举如下, 并给出风险计算结果, 如表 3 所示。

表 3 安全操作对象风险赋值
Table 3 Risk assignment of objects related to security operation

对象/安全威胁	1	2	3	4	5	6	7	8	风险等级	风险加成	风险值
BOOKMARK		√					√	√	7	3/21	80
CALENDAR		√							7	0	78
CALL_LOG		√					√		7	2/21	79
CONTACT		√					√		7	2/21	79
STORAGE		√		√					7	5/21	80
CAMERA		√							7	0	78
LOCATION		√							7	0	78
MIC		√							7	0	78
SENSOR		√							7	0	78
BLUETOOTH				√					5	0	56
MESSAGE	√	√		√	√		√		8	18/28	96
NETWORK	√	√	√	√	√	√	√	√	8	28/28	100
NFC				√					5	0	56
PHONE	√				√				8	4/28	91
SYNC		√							7	0	78
ACCOUNT		√							7	0	78
ID		√							7	0	78
LOG		√							7	0	78
SETTING						√			3	0	33
DEVICE						√		√	3	1/3	37
FILESYSTEM						√	√		3	2/3	41
PACKAGE				√			√	√	5	3/10	59
TASK			√			√		√	6	4/15	70
TEST						√			3	0	33
TOOL				√		√			5	3/10	59

可以观察到, 这些安全威胁均离不开 NETWORK 对象, 它在构成风险行为方面有着最广泛的使用, 因此风险值具有最高的 100 分。相比之下, SETTING 和 TEST 对象仅与系统破坏有关, 而系统破坏的危害程度排名第六, 相对来说危害较小, 因此这两个对象的风险值最低。将因素依据对象进行分类, 系统权限、API 调用、Intent Filter action 的风险值即为所属对象的风险值, 数据流的风险值取决于对应 source 和 sink API 中的较高风险值。

3.3 因素子集权重分配

本文根据因素与安全威胁之间的联系来评定安全风险, 无论是权限、API 调用、Intent Filter action 还是数据流, 只要属于同一对象都具有相同的安全风险值。例如, 与 BLUETOOTH 对象相关的因素有: ①系统权限 BLUETOOTH; ②受 BLUETOOTH 权限保护的 API 调用, 如 android.bluetooth.BluetoothAdapter 类的 disable 函数, 起到关闭蓝牙的效果; ③Intent Filter 的 action 属性 android.bluetooth.adapter.action.STATE_CHANGED, 供 Broadcast Receiver 组件声明, 监听蓝牙的状态变化; ④读取蓝牙数据或将信息从蓝牙泄露出去的数据流, 即 source 或 sink 函数为受 BLUETOOTH 保护的 API 调用。除数据流需权衡 source 与 sink 两方面风险外, 其他因素均享有与 BLUETOOTH 对象相同的风险值 55.6。

然而, 不同种类因素具有不同的性质, 而某些因素集也可以进一步细分, 因此每种因素的实际影响不能一概而论。本文为表 2 中 7 个因素子集设计了一种权重分配算法, 根据依次移去某一因素子集后数据集层次聚类产生的偏差对其进行权重赋值。

3.4 应用程序风险评估

为了更好地进行 Android 应用程序风险评估, 本文通过矩阵的形式来表达风险计算的过程。首先, 为每个应用程序建立因素分布矩阵, 以便描述其中包含因素所属的子集及对象类别的分布情况。然后, 将因素分布矩阵进行归一化, 即可得到便于计算的因素频率分布矩阵。此外, 需要将 3.2 节给出的对象风险值以及 3.3 节求得的因素子集权重转化为向量的形式。

定义 2 因素分布矩阵 $F = (f_{ij})^{m \times n}$ 描述各因素子集中因素的分布情况, 其中 m 为因素子集的数量, n 为对象数量, 元素 f_{ij} 为第 i 个因素子集中含有属于第 j 个对象类别的因素个数 ($1 \leq i \leq m, 1 \leq j \leq n$)。在本文的工作中, $m=7, n=25$ 。

定义 3 因素频率分布矩阵 $Q = (q_{ij})^{m \times n}$ 描述各因素子集中因素的频率分布, 其中

$$q_{ij} = \frac{f_{ij}}{\sum_{j=1}^n f_{ij}} \quad (2)$$

定义 4 对象风险向量 $O = [o_1, o_2, \dots, o_n]$ 是所有对象的安全风险值组成的集合, 其中 n 为对象数量, 元素 o_i 代表第 i 个对象的安全风险值 ($1 \leq i \leq n$)。

定义 5 权重分布向量 $W = [w_1, w_2, \dots, w_m]$ 是所有因素子集的权重组成的集合, 其中 m 为因素子集数量, 元素 w_i 代表第 i 个因素子集的权重 ($1 \leq i \leq m$)。

由因素频率分布矩阵和对象风险向量可计算因素子集风险向量

$$R^T = QO^T = \begin{bmatrix} q_{11} & \cdots & q_{1n} \\ \vdots & \ddots & \vdots \\ q_{m1} & \cdots & q_{mn} \end{bmatrix} \begin{bmatrix} o_1 \\ \vdots \\ o_n \end{bmatrix} = \begin{bmatrix} r_1 \\ \vdots \\ r_m \end{bmatrix} \quad (3)$$

其中, r_i 代表第 i 个因素子集的安全风险值 ($1 \leq i \leq m$)。

进一步, 由因素子集风险向量和权重分布向量可计算应用程序的安全风险值

$$S = WR^T = [w_1 \quad \cdots \quad w_m] \begin{bmatrix} r_1 \\ \vdots \\ r_m \end{bmatrix} \quad (4)$$

最终得到的 S 是一个百分制的数值, 即定量评估结果。

参考 GB/T 20984-2007《信息安全技术信息安全风险评估规范》^[19], 本文将 Android 应用程序的安全风险划分为 5 个等级, 分别标识为很低、低、中等、高、很高, 并定义风险值与等级的关系: 当 $S \in [0, 20]$ 时, 风险等级为 1; 当 $S \in (20, 40]$ 时, 风险等级为 2; 当 $S \in (40, 60]$ 时, 风险等级为 3; 当 $S \in (60, 80]$ 时, 风险等级为 4; 当 $S \in (80,$

100]时, 风险等级为 5。

4 基于层次聚类的因素子集权重分配

上文阐述了 Android 应用程序的安全风险评估框架, 涉及的 7 个因素子集都发挥了一定的作用。然而, 各因素子集对安全风险评估的贡献是不同的, 因此有必要设计一个权重分配算法来衡量它们的重要性。本文算法的主要思想是, 因素子集的权重能够反映它对安全风险评估结果的影响程度, 而这一影响可以量化为完整因素集与缺少该因素子集时, 应用程序安全风险评分造成的聚类效果之差。下面依次介绍层次聚类算法、最优聚类选择、聚类相似度计算方法, 并在此基础上引出本文的权重分配算法。图 3 是因素子集权重分配的流程示意, 最终可以得到因素子集的权重分布向量。

4.1 层次聚类算法

聚类是一种重要的无监督机器学习方法, 用于挖掘大量样本之间未知的内在关联。层次聚类是一种常见的聚类原型, 具有操作简单、对噪声数据不敏感、不依赖于初始值的选择等优点。本文采用一种称为聚合嵌套 (AGNES, agglomerative nesting) [20] 的凝聚式层次聚类方法, 基本步骤如下: ①将每个样本都视为一个单独的类簇; ②计算类簇之间的相似度; ③在所有类簇中, 将具有最大相似度的两个类簇合并成一个新类簇; ④计算新类簇与其他类簇之间的相似度; ⑤重复第③、④步直至达到终止条件。下面给出算法的实现细节。

(1) 算法输入: 根据 3.4 节给出的公式, 分别计算数据集中每个样本的因素子集风险向量

R , 进一步得到数据集风险分布矩阵作为 AGNES 算法的输入。由于需要衡量因素子集在安全风险评估中发挥的作用, 因而以因素子集的安全风险值为特征, 由此聚集的类簇是各因素子集安全风险值类似的样本团体。

定义 6 数据集风险分布矩阵 $D = (d_{ij})^{s \times m}$ 描述各样本中因素子集的风险分布情况, 其中, s 为数据集中的样本数量, m 为因素子集的数量, 元素 d_{ij} 代表第 i 个样本中第 j 个因素子集的安全风险值 ($1 \leq i \leq s, 1 \leq j \leq m$)。

(2) 类簇相似度: 本文用类簇之间的距离来衡量它们的相似度, 距离越小, 相似度越大。类簇之间的距离应当以某两个样本之间的距离来表示, 本文采用一种称为豪斯多夫距离的计算方法, 可以度量两个类簇间的最大不匹配程度。式(5)和式(6)给出类簇 A 、 B 之间的豪斯多夫距离

$$HD(A, B) = \max(\text{hd}(A, B), \text{hd}(B, A)) \quad (5)$$

$$\text{hd}(A, B) = \max_{a \in A} \min_{b \in B} d(a, b) \quad (6)$$

其中, $d(a, b)$ 代表样本 a 、 b 之间的距离, 本文采用欧氏距离。

$$d(a, b) = \|a - b\|_2 = \sqrt{\sum_{i=1}^m |a_i - b_i|^2} \quad (7)$$

定义 7 类簇距离矩阵 $E_k = (e_{ij})^{n_k \times n_k}$ 描述第 k 层两两类簇之间的相似度, 其中, k 为生成新类簇的迭代次数, n_k 为此时的类簇数量, 元素 e_{ij} 代表第 i 个类簇与第 j 个类簇之间的距离。当进行迭代时, 从当前矩阵中寻找大于 0 的最小元素, 将对应的两个类簇合并成一个整体, 并计算新的距离来替代与原先两个类簇有关的元素。

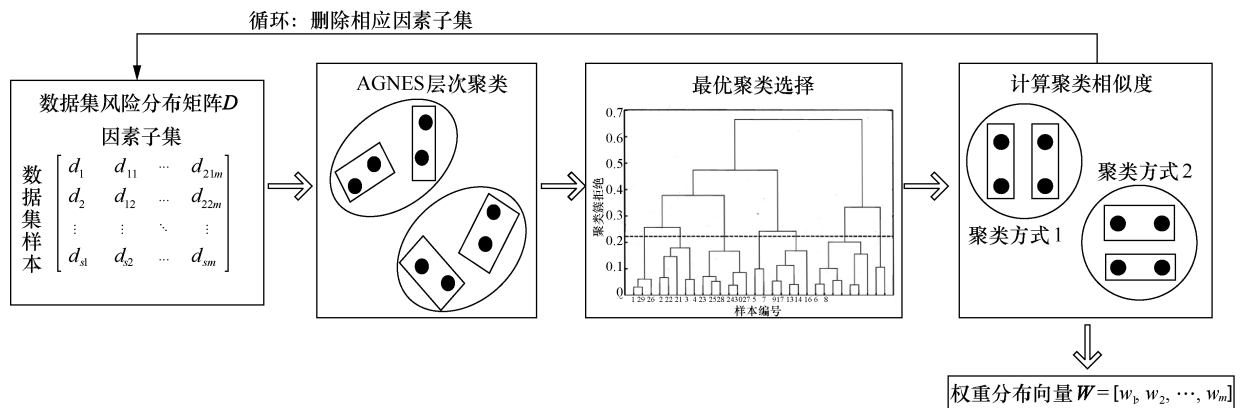


图3 因素子集权重分配流程
Figure 3 Weight distribution process of factor subsets

(3) 最大相似度不唯一的处理：当最大相似度不唯一时，列举出所有具有最大相似度的样本对，然后随机选择一对样本合并成一个类簇，并将涉及这两个样本的所有样本对删除。重复此过程直至没有样本对存在。

(4) 终止条件：设定生成的类簇数量或限定类簇间的最大距离是两种常见的终止条件，但是都很难在算法执行之前人工确定。因此，本文完整地执行算法流程直到所有类簇聚合成一个整体，再判断最合适的聚类层次。

(5) 算法输出：执行上述步骤，最终可以得到一个树状的聚类层次结构图。每一层上的类簇组成、类簇数量以及类簇之间的距离都有所体现，可以基于实际需要选择限制条件，获取此时的社区结构。

4.2 最优聚类选择

由 AGNES 算法的输出可以得到，数据集在不同层次上有着不同的聚类结构。为了确定最优的聚类方式，本文以层次为单位考察聚类结果，分别度量其性能。一个好的聚类划分，应当保证属于同一簇的样本相似度较高，而不同簇样本的相似度较低。本文采用一种称为 DB 指数^[20]的内部指标，在没有外部参考模型的情况下衡量簇内紧密程度和簇间分散程度。

定义 8 聚类簇划分向量 $C_k = \{C_1^k, C_2^k, \dots, C_{n_k}^k\}$ 描述第 k 层的聚类情况，其中， n_k 为此时的聚类簇数量，元素 $C_r^k = \{x_1^{kr}, x_2^{kr}, \dots, x_{n_r}^{kr}\} (1 \leq r \leq n_k)$ 描述第 r 个类簇的样本组成，则类簇 C_r^k 内的样本间平均距离为

$$\text{avg}(C_r^k) = \frac{2}{n_{kr}(n_{kr}-1)} \sum_{i=1, j>i}^{n_{kr}-1} d(x_i^{kr}, x_j^{kr}) \quad (8)$$

其中， n_{kr} 表示类簇 C_r^k 中的样本个数， C_r^k 的中心点 μ_r^k 为

$$\mu_r^k = \frac{1}{n_{kr}} \sum_{i=1}^{n_{kr}} x_i^{kr} \quad (9)$$

因此，第 k 层聚类的 DB 指数可以表达为

$$\text{DBI}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \max_{j \neq i} \left(\frac{\text{avg}(C_i^k) + \text{avg}(C_j^k)}{d(\mu_i^k, \mu_j^k)} \right) \quad (10)$$

DB 指数的值越小，聚类效果越好。比较不

同层次的 DB 指数，具有最小值的聚类方式即为最优聚类。

4.3 聚类相似度计算

本节给出同一数据集的两种聚类方式之间的相似度计算方法，以便衡量删除某一因素子集后对聚类结果产生的影响，从而推断该因素的重要性。若数据集 $X = \{x_1, x_2, \dots, x_s\}$ 存在两个不同的聚类簇划分 A, B ，那么根据是否在 A, B 中属于同一类簇，数据集中任意两个样本之间可以存在 4 种关系。若样本 x_i 在 A 中所属类簇为 $\lambda_A(x_i)$ ，在 B 中所属类簇为 $\lambda_B(x_i)$ ，则不同关系对的集合可以表达成

$$\text{SS} = \{(x_i, x_j) \mid \lambda_A(x_i) = \lambda_A(x_j), \lambda_B(x_i) = \lambda_B(x_j), 1 \leq i < j \leq s\} \quad (11)$$

$$\text{SD} = \{(x_i, x_j) \mid \lambda_A(x_i) = \lambda_A(x_j), \lambda_B(x_i) \neq \lambda_B(x_j), 1 \leq i < j \leq s\} \quad (12)$$

$$\text{DS} = \{(x_i, x_j) \mid \lambda_A(x_i) \neq \lambda_A(x_j), \lambda_B(x_i) = \lambda_B(x_j), 1 \leq i < j \leq s\} \quad (13)$$

$$\text{DD} = \{(x_i, x_j) \mid \lambda_A(x_i) \neq \lambda_A(x_j), \lambda_B(x_i) \neq \lambda_B(x_j), 1 \leq i < j \leq s\} \quad (14)$$

FM 指数^[20]可用于评判两种聚类方式之间的相似程度，它的计算公式为

$$\text{FMI} = \sqrt{\frac{|\text{SS}|}{|\text{SS}| + |\text{SD}|} \cdot \frac{|\text{SS}|}{|\text{SS}| + |\text{DS}|}} \quad (15)$$

FM 指数的值位于 [0,1] 内，其值越大，说明两者越相似。

4.4 权重分配算法

以层次聚类算法、最优聚类选择和聚类相似度计算为基础，算法 1 给出了基于层次聚类的因子子集权重分配算法，主要依据因子子集在以安全风险值为标准的聚类划分中发挥的作用来评定权重。

算法 1 因素的权重分配算法

输入 D_0 ——数据集风险分布矩阵

输出 W ——权重分布向量

Procedure Factor_WeightDistribution

begin

$\text{CT}_0 \leftarrow \text{AGNES}(D_0)$

$\text{OC}_0 \leftarrow \text{OptimalCluster}(\text{CT}_0)$

```

for( $i=1, i<m+1, i++$ ) do
  begin
     $D_i \leftarrow \text{Delete}(D_0, i)$ 
     $CT_i \leftarrow \text{AGNES}(D_i)$ 
     $OC_i \leftarrow \text{OptimalCluster}(CT_i)$ 
     $FMI \leftarrow \text{Similarity}(OC_0, OC_i)$ 
     $g_i \leftarrow 1-FMI$ 
  end
   $W \leftarrow \text{Normalization}(G)$ 
end

```

给定完整因子集对应的数据集风险分布矩阵 D_0 ，经层次聚类算法 AGNES 与最优聚类选择算法 OptimalCluster 计算得到的最优聚类簇划分表示为 OC_0 。然后进入循环，在第 i 次迭代时删除 D_0 的第 i 列，表明第 i 个因子子集不参与聚类划分，余下的因子子集构成数据集风险分布矩阵 D_i ，与之对应的最优聚类簇划分为 OC_i ，其中 $1 \leq i \leq m$ ， m 为 D_0 的列数（即因子子集个数）。Similarity(OC_0, OC_i) 给出完整因子集导致的聚类划分与缺少第 i 个因子子集导致的聚类划分之间的相似程度 FMI，它与该因子子集的重要性呈负相关，且 $FMI \in [0, 1]$ ，因此，用 $1-FMI$ 来量化该因子子集的重要性。循环体结束后求得因子子集重要性向量 $G = [g_1, g_2, \dots, g_m]$ ，并据此通过 Normalization 函数的归一化来分配权重。对于 $1 \leq i \leq m$ ，有

$$w_i = \frac{g_i}{\sum_{i=1}^m g_i} \quad (16)$$

即可得到因子子集的权重分布向量 W 。

5 实验分析

为了说明本文方法的有效性，将其与静态分析工具 Androguard 进行比较。实验结果表明，Androguard 的风险评估存在明显不合理之处，而本文方法更能反映 Android 应用程序的实际风险。

5.1 实验数据集

为了更好地观察良性应用与恶意软件的安全风险，本文收集了 Android Malware Genome Project^[21]的恶意软件，并在华为应用市场上下载

了各个类别的流行 Android 应用程序。同时，为确保良性样本真实可靠，本文将市场应用发送到 VirusTotal^[22]进行检查。当 VirusTotal 的所有反病毒扫描程序均未检测到异常时，该应用程序被认为是良性样本。最终，本文的数据集由包含 49 个家族的 1 260 个恶意软件，以及涉及华为市场 17 个类别的 1 340 个良性应用组成，共计 2 600 个实验样本。

5.2 权重分布

本文提取数据集中样本的相关因素，并采用基于层次聚类的权重分配算法来计算各因子子集的权重，以识别各因子子集对 Android 应用程序总风险评估的重要性。该方法将因子子集的影响量化为完整因素集与缺少该因子子集时安全风险评分的样本聚类效果差别，主要通过层次聚类、最优聚类选择、聚类相似度计算、归一化等步骤完成。算法利用 python 语言编程实现，运行结果如表 4 所示。可以看到，权限及其保护的 API 调用在风险评估中具有较为重要的作用，此外，Broadcast Receiver 关注的广播事件也是风险评估的关键因素。

表 4 权重分布
Table 4 Weight distribution

因子子集	权重分配
1	0.155
2	0.188
3	0.149
4	0.132
5	0.091
6	0.179
7	0.106

5.3 样本风险评估

Androguard 的 Androrisk.py 模块可以对应用程序的 APK 文件进行风险评分，通过 Fuzzy 算法从 DEX 代码、APK 文件、权限 3 个方面进行计算。相比之下，本文选择的评估因素种类更为丰富，不同类因素对总风险影响的权重分配更为合理，而 Androguard 简单地统计了一些孤立的因素且没有分配权重，导致其风险评估模型较为粗糙。

(1) 恶意样本

在本文的数据集中，共有 122 个样本属于恶

意软件家族 BaseBridge, Androguard 对它们的平均风险评分为 62, 并且存在大量达到 91 最高分的家族成员, 但有 6 个样本的风险评分为 0。在这些 0 分的恶意样本中, 以 SHA-1 哈希值为 7c0af89dd083b97db3dd70b7a2329c4a21a2c592 的 APK 文件为例, 表 5 说明了其中存在的风险因素, 并给出风险计算结果。

根据表 5 中的数据, 可以计算因子集风险向量 $R = [37, 37, 100, 59, 0, 0, 76]$, 得到该恶意样本的安全风险值 $S = 43$, 风险等级为 3。BaseBridge 家族的恶意软件通常会在用户不知情的情况下进行以下行为: ①将与设备 ID 有关的信息发送到远程服务器; ②通过拨号、发短信等手段订购增值服务, 并拦截运营商的确认短信; ③杀死设备中运行的安全软件进程。与 BaseBridge 家族的其他成员相比, 该样本的恶意行为简单得多, 主要是使屏幕关闭之后仍然在后台运行, 通过联网接收远程服务器的指令实施操作。因此, 本文的评估框架判定该恶意样本的安全风险值为 43 (中等风险) 是比较合理的。

(2) 良性样本

图 4 显示了华为应用市场分类下良性样本的 Androguard 安全风险评分情况。可以看到, Androguard 对许多良性应用给出 100 分的风险值, 几乎涵盖各个类别。Androguard 的风险评定范围为 0~100, 从这个意义上说 100 是理论上才会达到的风险值, 将如此多的良性应用判定为最高风险值是不太科学的。

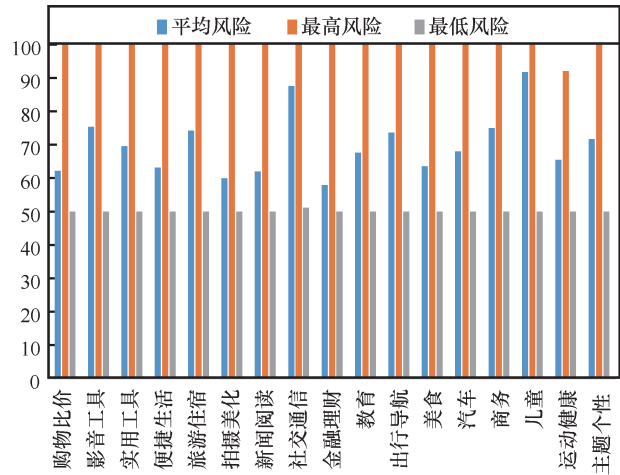


图 4 良性应用的 Androguard 风险评分
Figure 4 Risk score distribution of benign applications given by Androguard

在这些 100 分的良性样本中, 以 SHA-1 哈希值为 ff5a77e5dce38bbc2e9f46eb031bb435d8c05240 的 APK 文件为例说明其中存在的风险因素, 并给出风险计算结果。由于该样本具有较多的风险因素, 本文在表 6 中分 3 个子表显示, 并且对部分因素进行了概括。

根据表 6 中的数据, 可以计算因子集风险向量 $R = [70, 80, 0, 77, 0, 55, 56]$, 得到该良性样本的安全风险值 $S = 52$, 风险等级为 3。与表 5 相比, 该样本因功能需要而具有更多的风险因素, 但相对来说因子集的风险分布较为平均, 因此本文的评估框架判定它的风险值为 52, 略高于前一样本, 但同样是中等风险应用。通过以上两个例子可以说明, 本文的风险评估结果有侧重地受到所有因子集的影响, 综合考虑了应用程序中各个

表 5 示例恶意样本的风险因素
Table 5 Risk factors of malicious example

因子子集	因素	数量	对象类别	风险值
1	android.permission.WAKE_LOCK	1	DEVICE	37
2	WAKE_LOCK 对应 API	3	DEVICE	37
3	INTERNET 对应 API	3	NETWORK	100
4	android.intent.action.MAIN	1	PACKAGE	59
7	STORAGE → LOG、STORAGE → SETTING、STORAGE → PACKAGE	13	STORAGE	80
	FILESYSTEM → LOG、PACKAGE → LOG	10	LOG	78
	FILESYSTEM → SETTING	1	FILESYSTEM	41
	FILESYSTEM → PACKAGE、PACKAGE → SETTING	3	PACKAGE	59

表 6 示例良性样本的风险因素
Table 6 Risk Factors of benign example

(a) 因子子集 1 中的相关因素				
因素	数量	对象类别	风险值	
READ_CALENDAR 等权限	2	CALENDAR	78	
READ_CALL_LOG	1	CALL_LOG	79	
READ_CONTACTS	1	CONTACT	79	
READ_EXTERNAL_STORAGE 等权限	2	STORAGE	80	
CAMERA	1	CAMERA	78	
ACCESS_COARSE_LOCATION 等权限	4	LOCATION	78	
RECORD_AUDIO 等权限	2	MIC	78	
USE_FINGERPRINT	1	SENSOR	78	
BLUETOOTH 等权限	2	BLUETOOTH	56	
INTERNET 等权限	4	NETWORK	100	
CALL_PHONE	1	PHONE	91	
READ_PHONE_STATE	1	ID	78	
READ_LOGS	1	LOG	78	
WRITE_SETTINGS	1	SETTING	33	
WAKE_LOCK 等权限	3	DEVICE	37	
MOUNT_UNMOUNT_FILESYSTEMS	1	FILESYSTEM	41	
READ_APP_BADGE 等权限	4	PACKAGE	59	
GET_TASKS 等权限	3	TASK	70	
FLASHLIGHT 等权限	2	TOOL	59	
(b) 因子子集 2 中的相关因素				
因素	数量	对象类别	风险值	
READ_CONTACTS 对应 API	4	CONTACT	79	
ACCESS_COARSE_LOCATION 等权限对应 API	4	LOCATION	78	
BLUETOOTH 对应 API	33	BLUETOOTH	56	
ACCESS_NETWORK_STATE 等权限对应 API	79	NETWORK	100	
READ_PHONE_STATE 对应 API	12	ID	78	
READ_LOGS 对应 API	1	LOG	78	
WRITE_SETTINGS 对应 API	10	SETTING	33	
WAKE_LOCK 对应 API	4	DEVICE	37	
VIBRATE 对应 API	1	TOOL	59	
(c) 因子子集 4、6、7 中的相关因素				
因子子集	因素	数量	对象类别	风险值
4	android.intent.action.VIEW	5	STORAGE	80
	android.intent.action.MAIN	1	PACKAGE	59
6	android.intent.action.USER_PRESENT	1	DEVICE	37
	android.intent.action.HOTFIX_UPDATE_SUCCESS_ACTION	1	PACKAGE	59
	android.intent.action.BOOT_COMPLETED	1	TASK	70
7	FILESYSTEM → FILESYSTEM	19	FILESYSTEM	41
	PACKAGE → PACKAGE、FILESYSTEM → PACKAGE、PACKAGE → FILESYSTEM	102	PACKAGE	59

因素的分布情况,因此可以更合理地给出风险评分,不会出现较为极端的情况。

5.4 性能开销

本文在一台搭载 6 核 1.10 GHz 的 Intel Core i7-10710U 处理器和 16 GB 内存的笔记本电脑上进行实验,运行时间开销主要分为因素提取、矩阵的生成与计算、权重分配 3 部分。首先,在静态分析工具 Androguard 和 FlowDroid 的帮助下,平均为每个应用程序提取因素的时间为 23.7 s。然后,分类并生成对应的因素分布矩阵,据此计算出因素频率分布矩阵、因素子集风险向量,平均用时 4.8 s。其次,根据数据集所有样本的因素子集风险向量运行基于层次聚类的权重分配算法。这一过程涉及大量计算与比较,在运行了 56 h 之后,得到了权重分布向量。权重分配算法只需运行一次。此后,对于任意应用程序而言,只需经过因素提取、矩阵生成、风险计算步骤即可得到风险评估结果,瓶颈主要在于因素提取。图 5 表明在给定的单个样本评估时间内,数据集有多少应用程序可以完成风险评估结果。

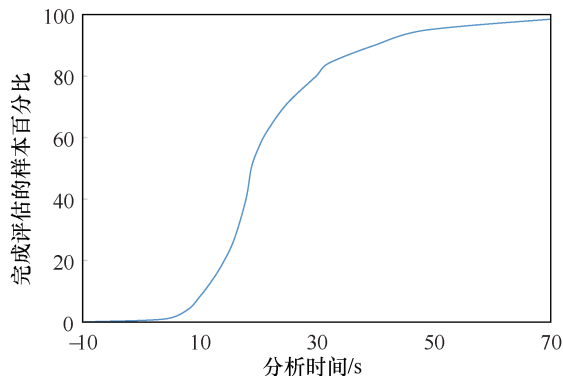


图 5 样本风险评估耗时分布

Figure 5 Time distribution of risk assessment for the dataset

6 结束语

评估 Android 应用程序造成的风险是移动安全领域中一个非常重要的研究方向。本文提出一种基于系统权限、API 调用、Intent Filter 的 action 属性以及数据流等多因素的 Android 应用程序风险评估方法,能够给出定量的风险值和定性的风险等级,具有较强的可扩展性。将评估因素进一步细分为 7 个因子集,全面地展示了应用程序的风险来源。根据与安全事件有关的操作对象将

因素进行分类,参考《移动互联网恶意代码描述规范》并结合对象与安全威胁的关系来计算对象风险,以便为因素赋予安全风险值。为了识别不同因子集对应用风险评估的影响,本文提出一种基于层次聚类的权重分配算法,使评估结果更加合理。实验表明,本文的评估方法能够有效地反映出 Android 应用的真实风险,可帮助用户选择风险更低的应用。

参考文献:

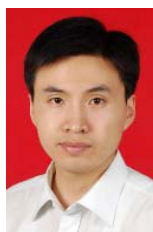
- [1] PENG H, GATES C, SARMA B, et al. Using probabilistic generative models for ranking risks of Android Apps[C]//The 19th ACM Conference on Computer and Communications Security (CCS). 2012: 241-252.
- [2] 徐君锋, 王嘉捷, 朱克雷, 等. 基于 AHP 的安卓应用安全信用指数度量方法[J]. 清华大学学报(自然科学版), 2018, 58(2): 131-136.
XU J F, WANG J J, ZHU K L, et al. Credit index measurement method for Android application security based on AHP[J]. Journal of Tsinghua University(Science and Technology), 2018, 58(2): 131-136.
- [3] RAHMAN A, PRADHAN P, PARTHO A, et al. Predicting Android application security and privacy risk with static code metrics[C]//2017 IEEE International Conference on Mobile Software Engineering and Systems. 2017: 149-153.
- [4] DINI G, MARTINELLI F, MATYEUCCI I, et al. Risk analysis of Android applications: a user-centric solution[J]. Future Generation Computer Systems, 2018, 80: 505-518.
- [5] 张贤贤. 安卓应用隐私评分技术的研究与实现[D]. 北京: 北京邮电大学, 2018.
ZHANG X X. Research and implementation of Android application privacy rating technology[D]. Beijing: Beijing University of Posts and Telecommunications, 2018.
- [6] 李舟军, 吴春明, 王啸. 基于沙盒的 Android 应用风险行为分析与评估[J]. 清华大学学报(自然科学版), 2016, 56(5): 453-460.
LI Z J, WU C M, WANG X. Analysis and assessment of Android application's risk behavior based on sandbox[J]. Journal of Tsinghua University (Science and Technology), 2016, 56(5): 453-460.
- [7] 余小秋. 基于行为分析的 Android 应用程序安全评估技术研究及系统设计[D]. 北京: 北京邮电大学, 2015.
YU X Q. The research and system design of Android application's safety assessment based on behavior analysis[D]. Beijing: Beijing University of Posts and Telecommunications, 2015.
- [8] 卜同同, 曹天杰. 基于权限的 Android 应用风险评估方法[J]. 计算机应用, 2019, 39(1): 131-135.
BU T T, CAO T J. Risk assessment method of Android application based on permission[J]. Journal of Computer Applications, 2019, 39(1): 131-135.
- [9] 王家琰. Android 应用软件安全检测与风险评估方法研究[D]. 郑州: 战略支援部队信息工程大学, 2018.
WANG J Y. Research on Android application security detection and

- risk assessment method[D]. Zhengzhou: Information Engineering University, 2018.
- [10] 黄心依, 曾凡平. 基于多重对应分析的 Android 应用安全等级评估[J]. 电子技术, 2016, 8(18): 72-78.
HUANG X Y, ZENG F P. The security level assessment model for Android application based on multiple correspondence analysis[J]. Electronic Technology, 2016, 8(18): 72-78.
- [11] 魏娟. Android 应用风险评估系统的设计与实现[D]. 成都: 电子科技大学, 2015.
WEI J. The design and implementation of application risk assessment system based on Android[D]. Chengdu: University of Electronic Science and Technology of China, 2015.
- [12] JING Y, AHN G J, ZHAO Z, et al. RiskMon: Continuous and automated risk assessment of mobile applications[C]//The 4th ACM Conference on Data and Application Security and Privacy. 2014: 99-110.
- [13] PANDITA R, XIAO X, YANG W, et al. WHYPER: Towards automating risk assessment of mobile applications[C]//The 22th USENIX Security Symposium. 2013: 527-542.
- [14] 肖智婕. 面向种群的 Android 安全风险评估和恶意应用检测[D]. 武汉: 武汉科技大学, 2019.
XIAO Z J. Population-based Android security risk assessment and malicious application detection[D]. Wuhan: Wuhan University of Science and Technology, 2019.
- [15] 孟小峰, 朱敏杰, 刘俊旭. 基于 Android 权限机制的应用安全检测方法[J]. 信息安全研究, 2019, 5(9): 778-788.
MENG X F, ZHU M J, LIU J X. Quantitative research on privacy risk of large-scale mobile users[J]. Journal of Information Security Research, 2019, 5(9): 778-788.
- [16] 中国互联网协会反网络病毒联盟. 移动互联网恶意代码描述规范[R]. 2011.
Anti Network-Virus Alliance of China. Specification for mobile Internet malicious code[R]. 2011.
- [17] DESNOS A. Androguard documentation, release 3.3.5[R]. 2019.
- [18] ARZT S, RASTHOFER S, FRITZ C, et al. FlowDroid: precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for Android apps[C]//The 35th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI). 2014: 259-269.
- [19] GB/T 20984-2007, 信息安全技术信息安全风险评估规范[S].
GB/T 20984-2007, Information security technology-risk assessment specification for information security[S].
- [20] 周志华. 机器学习[M]. 北京: 清华大学出版社, 2016.
ZHOU Z H. Machine learning[M]. Beijing: Tsinghua University Press, 2016.
- [21] ZHOU Y J, JIANG X X. Dissecting Android malware: characterization and evolution[C]//The 33rd IEEE Symposium on Security and Privacy (Oakland). 2012: 95-109.
- [22] GoogleGroups. VirusTotal[EB].

[作者简介]



超凡 (1995-), 女, 江苏启东人, 信息工程大学硕士生, 主要研究方向为信息安全、代码信息流分析。



杨智 (1975-), 男, 河南开封人, 博士, 信息工程大学副教授, 主要研究方向为操作系统安全、云计算安全、隐私保护。



杜学绘 (1968-), 女, 河南新乡人, 博士, 信息工程大学教授、博士生导师, 主要研究方向为空间信息网络、云计算安全。



韩冰 (1978-), 女, 河南民权人, 信息工程大学讲师, 主要研究方向为网络空间信息管理和评估。

《网络与信息安全学报》

第二届编辑委员会

主 任 委 员：方滨兴

常务副主任委员：李风华

副 主 任 委 员：李 晖 俞能海 Kui Ren[USA] 刘建伟

邹德清 杜学绘 刘华鲁

委 员：（按姓氏笔画排序）

丁 勇	马建峰	王 标	王 滨	王布宏
王连海	王良民	王国胤	王明生	王美琴
牛 犇	仇洪冰	田有亮	田俊峰	史国振
付章杰	冯 涛	朱建明	延志伟	仲 盛
向 涛	刘 哲	刘 渊	刘西蒙	刘哲理
闫 峥	孙东红	杨宏宇	肖 甫	肖 亮
吴 铤	吴晓平	邱卫东	何道敬	沈 剑
沈 超	张小松	张云勇	张红旗	张宏莉
张建标	张振江	陆月明	陈立全	陈兴蜀
陈性元	陈雷霆	苗春雨	林 鹏	易东山
赵 波	赵琉涛	咸鹤群	禹 勇	姚志强
秦 拯	贾 焰	殷丽华	翁 健	黄欣沂
曹春杰	彭国军	蒋文保	韩伟力	韩劲松
程 光	阚海斌			

David Naccache [FRA] Gang Qu [USA] Jinjun Chen [AUS]

Willy Susilo [AUS] Yier Jin [USA]