CNCERT

2018 中国网络安全年会
CNCERT Annual Conference

基于频繁模式和加权朴素贝叶斯的安卓恶意应用检测方法

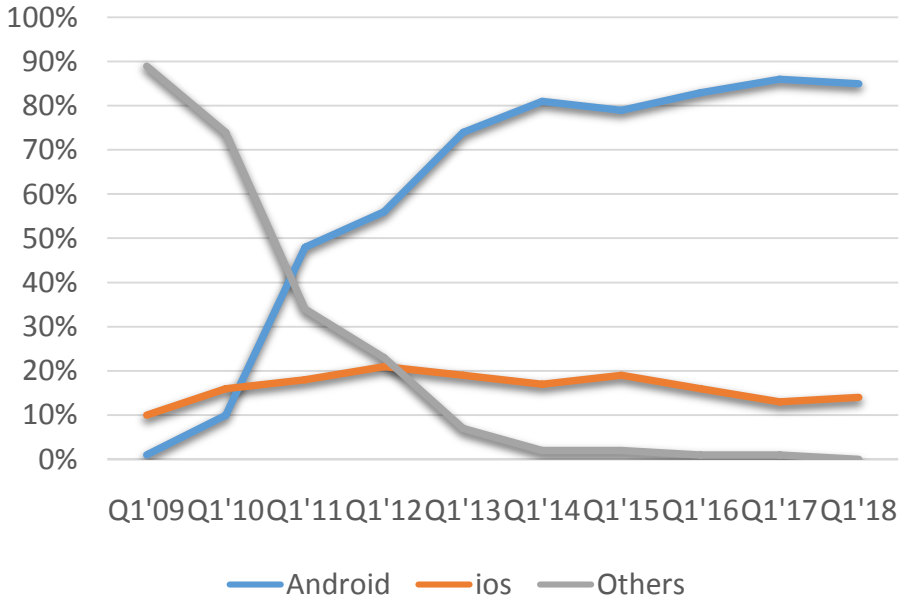
报告人：李经纬

湖南鼎源蓝剑信息科技有限公司 技术总监

北京大学软件安全研究小组 组长

背景

2009-2018安卓系统全球市场份额

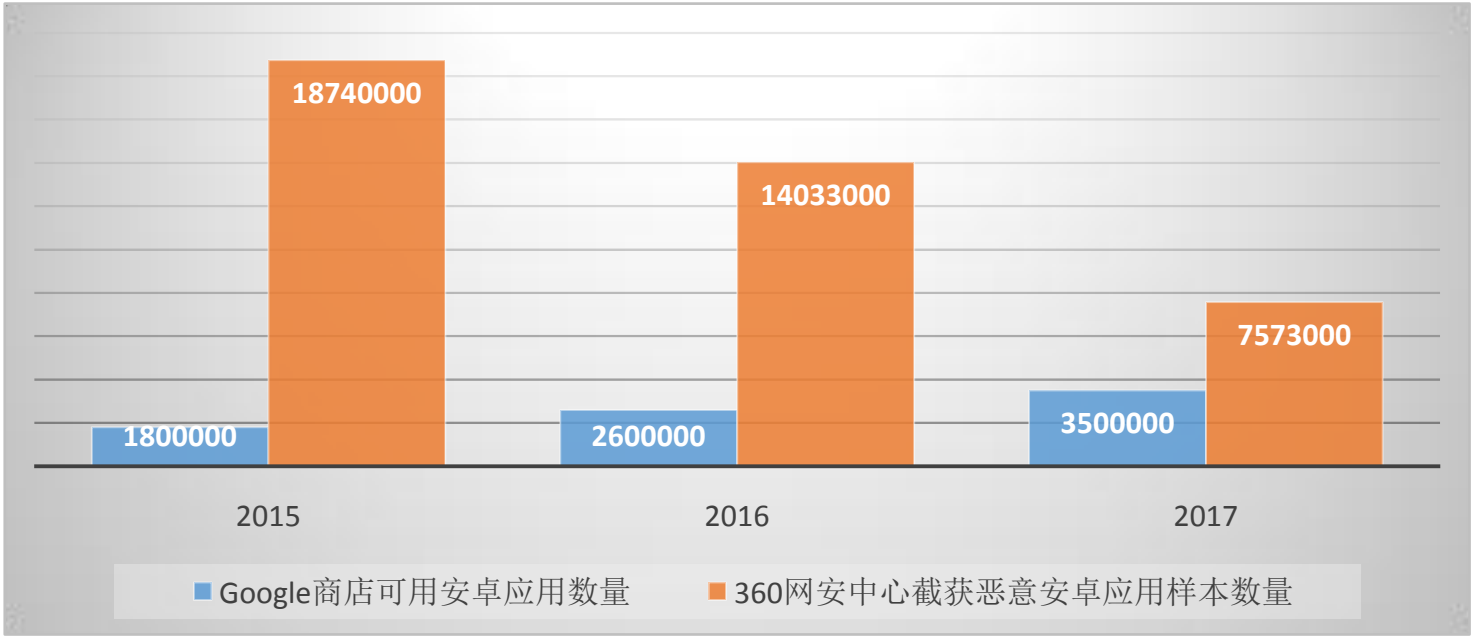


- 从2008年谷歌正式发布安卓系统1.0到2011年第一季度，仅仅历经三年时间Android系统市场份额就达到了48%，并超过Symbian跃居全球首位。

中国智能手机市场份额

- 截止2017年8月，仅Android系统在中国市场的占有率就已超过80%。
- 日前，据市场调研机构Gartner公布的2018年第一度Smart Phone Market Report显示，安卓的市场份额已经达到85.9%，远超ios所占的14.1%市场份额。

背景



根据360于2018年上半年发布的研究报告显示，截止至2017年12月高达93.94%的Android手机存在安全漏洞，360互联网安全中心2017全年累计截获了Android平台新增恶意软件样本757.3万个，监测到感染恶意软件Android用户2.14亿人次。

安卓恶意应用检测方法

● 静态检测

基于权限、API调用：找出在恶意软件中请求频繁的但是在正常软件中却请求很少的权限组合，由此自动产生出规则集合，用于进行恶意软件识别。这些方法都基于找出恶意应用中较多的权限组合或模式，但是这种方法过于绝对，这些权限的组合或模式往往也会出现在正常应用中，导致较高的误报率。

基于签名：静态污点跟踪，结合内部组件调用，通过语义提取来产生签名。但是他们的方法仅对于几个恶意软件家族的检测比较有效，对于具有新签名的恶意软件难以检测。

基于组件：通过反编译应用程序，提取权限、资源和字节码等重要内容，进行程序安全性的评估。

安卓恶意应用检测方法

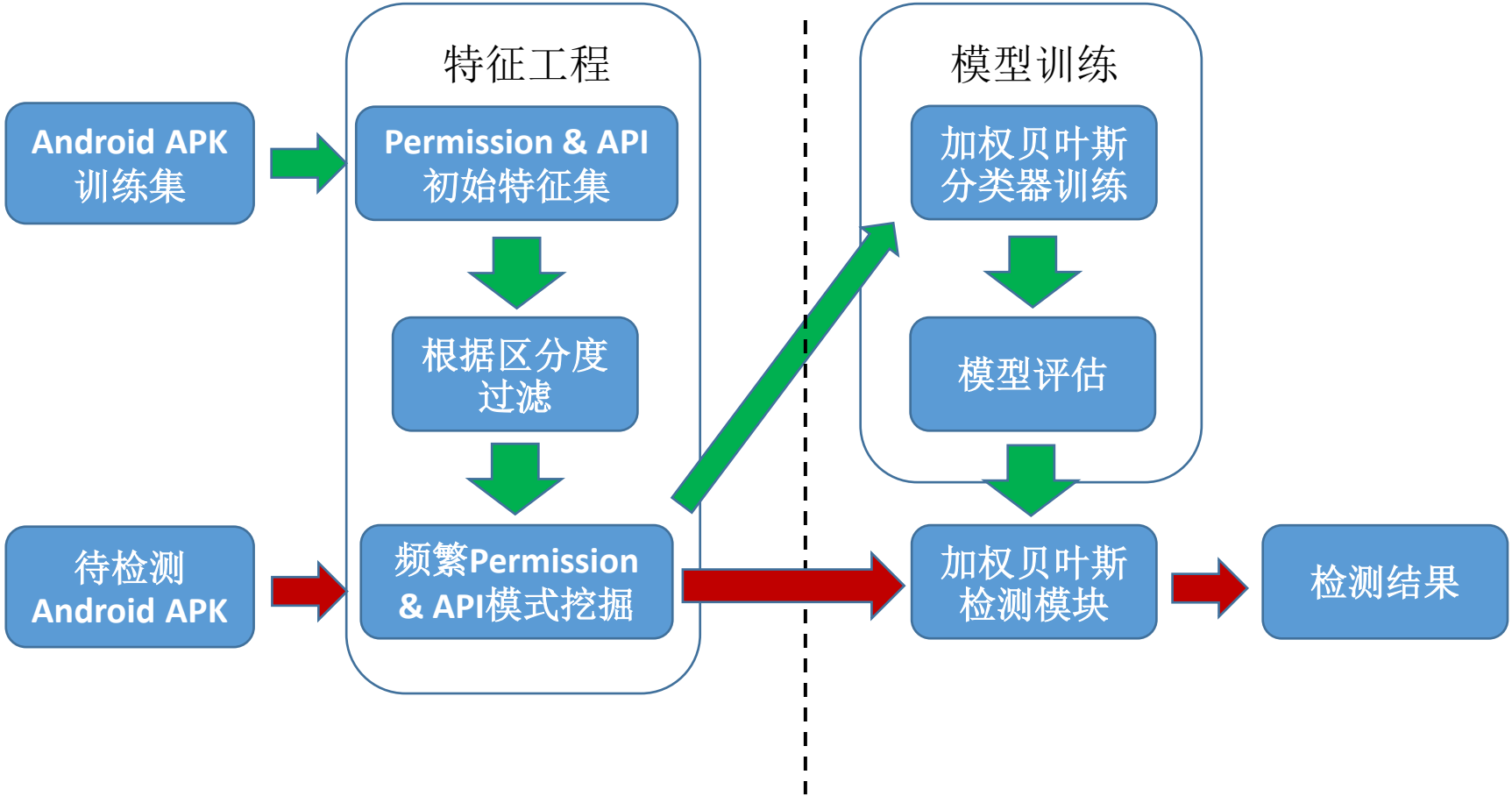
● 动态检测

基于行为：行为检测是指检测特定的恶意行为，如敏感的数据泄露，发送短信 / 电子邮件，无需用户同意的语音通话，可 以准确地检测到特定功能。目前大多数动态检测方法采用 的都是行为检测 , 以下举例说明几种行为检测方法。

基于轮廓：恶意应用程序可能会利用有限的硬件资源，从安卓系统收集正常应用和恶意应用的 参数(如CPU 的使用、内存利用率统计、网络流量模式、电池的使用和系统调用)。

缺点：时效性不强、对运行环境要求高

检测方法流程图


特征工程
模型训练
Android APK
训练集
Permission & API
初始特征集
根据区分度
过滤
加权贝叶斯
分类器训练
模型评估
待检测
Android APK
频繁Permission
& API模式挖掘
加权贝叶斯
检测模块
检测结果

数据集来源及初始的权限&API调用提取

● 数据集来源

获取了1000个来自VirusShare的恶意应用，收集了1000个来自谷歌商店的正常应用，将以上样本按一定比例作为训练集与测试集。

● 权限&API调用提取

本文使用androguard静态分析工具，它默认使用ded反编译逆向工具来进行反编译，利用androguard中的get_permissions等方法提取权限列表和敏感API的调用信息。

基于区分度的权限&API调用过滤

- 我们要选取的特征集合中的元素应该是在恶意应用中频繁出现且在非恶意应用中较少出现的，或在恶意应用中较少出现且在非恶意应用中频繁出现的。
- 设恶意应用样本总量为$|A_{mal}|$，正常应用的样本总量为$|A_{nor}|$
- 具有特征f_i的恶意应用样本数量为$|f_{i,mal}|$
- 具有特征f_i的恶意应用样本数量为$|f_{i,nor}|$
- 特征f_i在恶意应用中出现的频率为$F_{i,mal}=|f_{i,mal}| / |A_{mal}|$
- 特征f_i在正常应用中出现的频率为$F_{i,nor}=|f_{i,nor}| / |A_{nor}|$
- 则区分度$dis(f_i) = 1 - \frac{\min\{F_{i,mal}, F_{i,ben}\}}{\max\{F_{i,mal}, F_{i,ben}\}}$

筛选发现经常出现在恶意应用中的权限和API

Permission	API
WRITE_EXTERNAL_STORAGE	Timer;->schedule
READ_SMS	NetworkInfo;->toString
WRITE_SMS	DataOutputStream;->writeBytes
SEND_SMS	Socket;->getSoLinger
RECEIVE_SMS	Runtime;->exec
READ_CONTACTS	System;->setErr
WRITE_APN_SETTING	DexClassLoader;->LoadClass
CALL_PHONE	ContextImpl;->getSystemService
READ_PHONE_STATE	Intent;->setAction
INSTALL_PACKAGES	CoontextWrapper;->registerReceiver
…	…

频繁特征模式挖掘

朴素贝叶斯：$P(C|F_1,F_2,\dots,F_n)=\frac{P(C)P(F_1,F_2,\dots,F_n|C)}{P(F_1,F_2,\dots,F_n)}$

$$=\frac{P(C)\prod_{i=1}^{n}P(F_i|C,F_1,F_2,\dots,F_{i-1})}{P(F_1,F_2,\dots,F_n)}\propto\frac{P(C)\prod_{i=1}^{n}P(F_i|C)}{P(F_1,F_2,\dots,F_n)}$$

$P(F_i|C,F_1,F_2,\dots,F_{i-1})$难以计算，$P(F_i|C)$却容易得多，假设特征间相互独立，使$P(F_i|C,F_1,F_2,\dots,F_{i-1})=P(F_i|C)$。

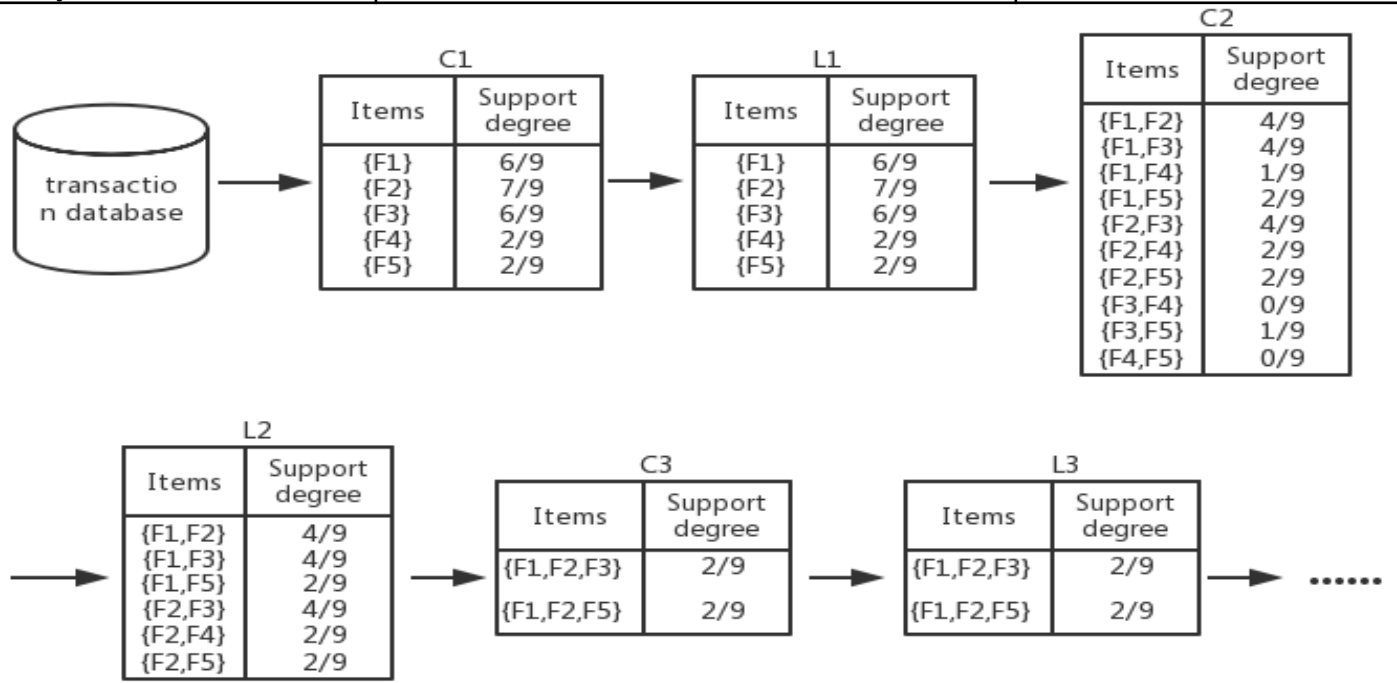
- 直接应用朴素贝叶斯分类器进行分类并不恰当
- 朴素贝叶斯的前提条件为所有特征之间相互独立
- 权限和API调用之间往往具有很强的相关性

如何解决？

频繁特征模式挖掘

- 通过挖掘权限&API调用频繁模式，以频繁模式来表示特征间的相关性。
- 以频繁模式作为特征应用朴素贝叶斯算法。
- 一定程度上弥补权限和API调用之间不独立的问题，克服朴素贝叶斯的假设条件与这种分类情况产生的冲突。

ItemID	Permission&API Item	APKID	Permission&API Feature Itemset	APKID	Permission&API Feature Itemset
F1	WRITE_EXTERNAL_STORAGE	APK001	F1,F2,F5	APK006	F2,F3
F2	RECEIVE_BOOT_COMPLETED	APK002	F2,F4	APK007	F1,F3
F3	NetworkInfo;->toString	APK003	F2,F3	APK008	F1,F2,F3,F5
F4	ActivityManager;->getProcessMemoryInfo	APK004	F1,F2,F4	APK009	F1,F2,F3
F5	DataOutputStream;->writeBytes	APK005	F1,F3		



加权的朴素贝叶斯算法——可靠性

在频繁模式挖掘的过程中我们发现一个频繁模式中包含项的数量越多它的支持度也会越低，随之其先验概率也就越小，但是一个频繁模式中的数量越多往往意味着根据它进行的判断越准确可靠，因此我们对朴素贝叶斯进行加权改进。

$$reli(items_i) = \frac{|Items_i|}{\max\{|Items_k|\}}, Items_k \in D$$

$$P(Malware|D) = \frac{P(Malware)\prod_{k=1}^{n} P(Items_i|Malware)}{P(D)}$$

$$P(Malware|D) = \frac{P(Malware)\prod_{k=1}^{n} [P(Items_i|Malware)\cdot reli(items_i)]}{P(D)}$$

过去基于权限、API调用的静态检测存在的问题

- 安卓应用中的权限和API调用能够有效的反应一个安卓应用的行为模式，以往的研究者大多数只考虑单一的权限或API特征，并未考虑权限、API调用这些特征之间存在着一定的关联性和模式。

- 一些学者也试图找出恶意应用内权限特征之间具有的组合模式，但是根据这种组合模式进行恶意性的检测又过于绝对。比如通过频繁模式挖掘寻找组合模式，由于很多正常安卓应用也具有恶意应用的频繁模式，但是频繁模式挖掘具有太过绝对往往会将这类正常应用归类为恶意应用，使检测的准确率大大降低。

- 朴素贝叶斯分类器具有所有特征不相关的前提条件，这一假设在安卓恶意应用的检测中并不适用，往往很多特征之间的相关性极强，所以有也很大的改进和提升空间。

● 创新点

对具有较高区分度的权限和API调用进行频繁模式挖掘，克服正常安卓应用也具有恶意应用特征的问题，提高频繁模式挖掘效率。

通过频繁模式挖掘得到权限特征和API特征组成的频繁模式用以消除特征间相关关系对朴素贝叶斯分类器影响。

通过基于频繁模式的“权限&API”模式特征进行朴素贝叶斯分类模型的训练，消除频繁模式挖掘进行分类时太过绝对的缺点。

对朴素贝叶斯进行加权，提高分效果好的频繁模式的权重。

频繁模式挖掘+朴素贝叶斯算法

- **检测速度快**

贝叶斯分类非常适合用来过滤大量应用程序数据集，因为它一旦经过训练就可以执行相对较快的分类，计算开销较低。

- **配合效果好**

由频繁模式的挖掘过程可知，某一频繁特征模式的支持度即为该频繁特征模式的先验概率，可直接应用于朴素贝叶斯算法中，由此也可见频繁模式与朴素贝叶斯具有良好的适配性。

- **引入概率的思想**

用概率的思想，弥补频繁模式挖掘过于绝对，非黑即白的缺陷。

实验一

对于特征属性我们利用了区分度进行了筛选，去除了区分度不佳的特征属性，降低一些特征属性对于分类结果的影响。因此实验一用来检测我们采用的特征处理方法的性能。我们首先比较了未进行特征处理的朴素贝叶斯算法性能和进行了特征处理之后的朴素贝叶斯算法性能。

Algorithm	Evaluation index	Pre-filter	Post-filter
Na ïve Bayes	ACC(%)	72.75	82.34
	FP(%)	20.41	16.53
	Runtime(s)	143.18	72.83
Random Forest	ACC(%)	70.59	81.96
	FP(%)	23.63	14.23
	Runtime(s)	203.69	79.38
SVM	ACC(%)	73.02	81.96
	FP(%)	18.63	14.23
	Runtime(s)	194.18	86.06

实验二

为了评估基于频繁模式的加权贝叶斯方法的性能，实验对于以筛选后的普通权限和API调用为特征的朴素贝叶斯算法，以频繁权限&API调用模式为特征的朴素贝叶斯算法和基于频繁模式的加权贝叶斯的检测结果进行了比较。

Evaluation index	**Naïve Bayes based on pre-filtering permission&API features**	**Naïve Bayes based on post-filtering permission&API features**	**Weighted naive Bayes based on frequent permission&API patterns**
ACC(%)	72.75	82.34	88.69047619047618
FP(%)	20.41	16.53	12.3
Runtime(s)	143.18	72.83	80.65

结论

本文首次将频繁模式与朴素贝叶斯相结合，将概率的思想与频繁模式的思想相结合，既克服了朴素贝叶斯分类器对于特征无关假设的缺点，又克服了根据频繁模式分类太过绝对的缺点。提出了一种基于频繁模式和加权朴素贝叶斯的安卓系统恶意应用的检测方法，对权限特征和API特征进行深层次的频繁模式挖掘，并将频繁项集的可靠性与朴素贝叶斯算法相结合得出一种加权的朴素贝叶斯算法，最后将具有权限相关性的频繁权限&API项集特征基于加权的朴素贝叶斯算法进行分类，实验结果证明了本文提出方法的有效性和正确性。

2018
感谢大家的聆听

Thank you very much & best regards.