3.3 Android应用漏洞检测模块

漏洞检测模块为静态检测中的第二部分，基于Soot对应用进行反编译以检测出其中可能存在的漏洞。另一方面，为了提高漏洞检测精度，本文提出了一种基于过程内数据流分析技术的变量行为记录算法，在漏洞检测中可以用于实现函数内变量到相关常量的追溯，提高检测准确率。

3.3.1 Android应用漏洞检测模块的工作流程

漏洞检测模块的完整工作流程如图7所示。首先使用Soot对APK进行初始化工作，输出的两个结果一个是Jimple中间代码，另一个则是控制流图，其中控制流图使用3.3.2小节中介绍的追溯算法为函数内的变量生成行为记录。初始化工作结束后，则开始进行漏洞特征匹配，最后输出检测结果。



图7 漏洞检测模块的工作流程

3.3.2 基于数据流分析的变量行为记录算法

在对漏洞特征进行匹配时，经常将特定的常量参数做为漏洞特征，然而应用开发中有很多情况是先将常量赋值到一个变量对象里，进行处理之后再作为参数传值到漏洞特征函数中，一个最简单的例子就是使用StringBuilder连接字符串，最后调用toString()方法将生成的字符串作为参数传给函数，对于这种情况，传统基于字符匹配的检测方案是无法匹配到特征的，本小节提出的算法则用于解决这种问题，算法基于过程内数据流分析生成变量的行为记录，之后基于此记录追溯变量的常量特征。具体算法流程如图8所示



图8 变量行为记录算法

在利用Soot生成控制流图后，使用Soot提供的过程内数据流分析框架来模拟完整控制结构下的语句遍历分析，并在此过程中基于传入的记录规则，详细记录所有变量的初始化、变量的方法和成员变量调用、变量赋值、变量值传递等所有变量相关的行为。在最终生成的变量行为记录中，即可清楚的获得变量的常量特征。

3.4 隐私泄露检测模块

在智能设备大量普及的当下，手机支付、手机社交都与移动设备紧密相关，可以说手机比电脑中存放着更多用户的隐私信息，这些信息一旦被应用恶意获取，就可能给用户带来无法挽回的损失。因此用户隐私安全性与Android应用的安全性是紧密相关的，隐私泄露检测模块便从隐私泄露的角度入手，进行Android应用安全性的评估

3.4.1 对Sources点及Sinks点的分析归类

Android系统中含有很多涉及到获取用户隐私的API(Sources点)以及隐秘的数据发送途径(Sinks点)，通过归类整理本文将其大致整理为6类，接着使用基于过程间数据流分析技术的污点传播算法，寻找从Sources点到Sinks点的传播路径，从而发现这些隐私泄露问题。

3.4.2 使用污点分析算法寻找传播路径

Android应用程序的运行方式不像其他程序一样有个固定的入口点，然后顺序执行下去，Android应用中每一个组件都有自己完整的生命周期，组件在运行过程中随时可能被触发调用，因此要对Android应用程序进行过程间数据流分析，就需要先为Android应用程序的生命周期建模，并构造一个虚拟的main方法，在方法内模拟触发所有组件，然后再对这个虚拟的main方法进行数据流分析，寻找Sources到Sinks间的路径。

模型使用FlowDroid[9]进行实现污点传播路径的寻找。FlowDroid是基于流分析技术的隐私泄露分析工具，它对Android应用程序的生命周期做了完整建模，并构建了一个虚拟的dummyMainMethod方法来模拟生命周期，在对应用内的Sources点及Sinks点做了标记后，使用过程间数据流分析技术寻找这些点间的传播路径。

本模型即基于FlowDroid所提供的分析接口以及3.1归类的Sources点与Sinks点进行Android应用中的隐私泄露分析。