# 基于 APK 文件抓取系统的匹配模块设计

高瑞华
(陕西省理工学校 陕西 西安 710054)

**摘要**：文中提出了一个 APK 抓取系统的设计思路，首先设计了该系统的整体框架，使用 Mysql5.5 实现系统的数据库，基于开源 python 网络爬虫框架 Scrapy，结合应用市场及 APK 的特性，联合使用 VirusTotal 和特征匹配检测的方法，重点设计了该 APK 文件抓取系统下基于模糊哈希算法的指纹匹配模块。为降低 APK 的恶意性，详细论述了匹配模块的功能、匹配算法、主要解决了“如何快速有效的检测恶意软件”的等问题，达到了设计要求。为后续研究提供了有力支撑。

**关键词**：APK 抓取；特征匹配；匹配度；模糊哈希算法
**中图分类号**：TP393.01 **文献标识码**：A **文章编号**：1674-6236(2016)03-0047-03

## Design of matching model based on APK file grabbing system

GAO Rui-hua
(*Shaanxi Technological School*, *Xi'an* 710054, *China*)

**Abstract:** This paper proposed a new design method for the APK capture system. The First designed the overall framework of the system using Mysql5.5 to setup the system´s database. Based on open source Python web crawler framework Scrapy as well as the characteristics of the market and application of APK, It accomplished the design of fingerprint matching module using Fuzzy Hashing algorithm. uring the design process, VirusTotal and feature matching method were also combined. It discussed function of the matching module and matching algorithm in detail, greatly improved the efficiency of malicious software detection. The new method meet the design requirement and provide a strong support for the future research.

**Key words:** APK capture; feature matching; compatibility; fuzzy hash algorithm

当前安卓应用市场鱼龙混杂，各个应用市场中存在较大比例的恶意应用程序，其原因是各个应用市场对发布的 APK 的检测方法存在差异和缺陷，因此面对众多应用市场，如何给用户提供更安全可靠的 APK，减小用户下载恶意 APK 的可能性，具有很高的研究和市场应用价值。

## 1 APK 文件抓取系统总体框架设计

无线通信技术、3G 及 4G 网络的发展深刻的改变着所有人的生活，搭载移动操作系统的智能手机的用户越来越多，其功能也日趋丰富和多元化。由于众多的开发人员开发出大量的 Android 应用程序，给 Android 用户带了很大的便利，但是同时也给用户带来很多的安全隐患，其中恶意应用程序的泛滥等问题严重威胁着 Android 用户的安全[1-2]。

针对目前市场恶意检测方案的缺陷，必须设置可靠性更高的检测模式，对应用市场中的应用程序进行抓取下载，进行恶意性分析后分别存储，以便下载用户的选择。该设计方案中包括搜索模块、跟踪模块、信息抓取模块、APK 下载模块、APK 解析模块、APK 特征信息提取模块、特征匹配模块、VirusTotal 检测模块和数据库存储模块，其中数据库存储模块包括市场元数据库、APK 元数据库、恶意 APK 库和非恶意 APK 库。

通过分析各个应用市场中恶意应用程序的安全现状，针对目前市场恶意检测方案的缺陷，设计方案如图 1 所示。

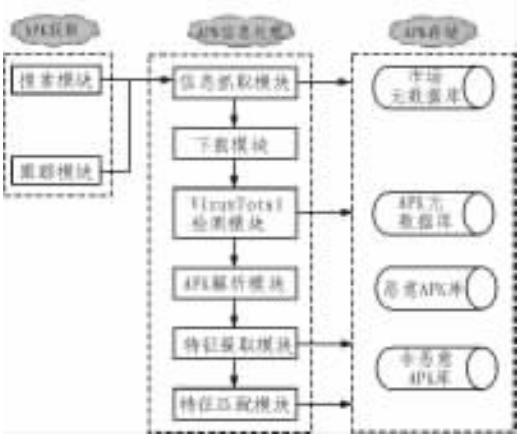


图 1 系统总体框架图
Fig. 1 The overall framework of the system

各个模块的功能如下：

1)搜索模块

**收稿日期**：2015-04-07 **稿件编号**：201504057
**作者简介**：高瑞华(1980—)，女，陕西米脂人，硕士，讲师。研究方向：网络与安全、信息控制。

该模块的作用是搜索新的 APK 文件,搜索各个应用市场中的 APK,并与已下载的 APK 对比,如果已经下载过,则不予处理,否则,交给后面的模块处理。

2)跟踪模块

该模块的作用是更新已下载的 APK,将已下载过的 APK 与应用市场上的对应的 APK 对比,如果已更新,则交给后面模块继续处理,否则不予处理。

3)信息抓取模块

该模块的作用是抓取符合条件的 APK 相关信息,并交给后续模块处理。

4)下载模块

该模块的作用是下载前面符合条件的 APK,并提供给后续模块分析处理。

5)VirusTotal 检测模块

该模块的作用是检测下载的 APK 文件的恶意性。通过调用病毒检测工具 VirusTotal,对 APK 进行分类存储。

6)APK 解析模块

该模块的作用是解析下载的 APK 文件,模块介绍了怎样解压缩应用程序安装包 APK, 然后遍历判断文件的类型,对不同的文件采用不同的方式进行解析, 最终得到配置文件、签名文件及 Java 源代码交给后面模块。

7)特征提取模块

该模块的作用是从解析模块解析的配置文件和签名文件中提取 APK 文件的特征属性,提供给后续模块。

8)特征匹配模块

文中探讨的模块, 作用是对 Java 源代码指令序列用模糊哈希算法,生成指纹,并与恶意代码特征库匹配,通过检测应用程序的源代码和恶意代码的相似度, 判定应用程序的恶意性。

9)数据库存储模块

此框架设计了 4 个数据库, 分别为市场元数据库、APK 元数据库、恶意 APK 库和非恶意 APK 库。分别存储 APK 市场信息、APK 文件信息、恶意 APK 文件和非恶意 APK 文件。

文中研究的对象是特征匹配模块,下面具体阐述这一模块的设计过程。

## 2 匹配模块功能

匹配模块对 Java 源代码指令序列用模糊哈希算法,生成指纹,并与恶意代码特征库匹配,通过检测应用程序的源代码和恶意代码的相似度,判定应用程序的恶意性。

初始时,AndRadar 需要一系列已知的应用软件样本,这些样本可以是恶意的软件或者其他良性软件,并把这些样本软件称为种子。由于 AndRadar 的动态性,实时在线的特点,使得 AndRadar 相对于静态分析有持续不断的分析软件的行为特点。软件种子可以来自于最新的被标记为恶意软件的软件库,可以是杀毒软件扫描的恶意软件,也可以是良性的软件[3-4]。这些软件样本将被用来和应用市场上的软件进行匹配,按匹配度进行相关处理。

## 3 APK 文件的匹配算法

种子应用程序和应用市场中的 APK 进行匹配时,有 4 种方式对两个软件进行相关度匹配,4 种不同的标识符分别为包名、指纹鉴别法、方法签名和哈希值[5-6]。并根据匹配组合分为 4 个匹配等级,如表 1 所示。

**表 1 基于 4 种标识符的不同的匹配度**

**Tab. 1 The different matching degree based on four identifier**

| APK identifer | Match level |
|---|---|
| MD5 | Perfect match |
| Package name | Weak match |
| Package name, fingerprint | Strong match |
| Package name, method signatures | Strong match |
| Package name, fingerprint, method signatures | Very strong match |

对应用市场而言, 如官方市场 Google Play、Appchina、Anzhi、Wandoujia 或者 Coolapk, 均使用包名作为市场内部的参考,在查询时也是非常直观的,这是因为包名会作为搜索该应用程序的一部分出现在应用程序的 URL 中。其他应用市场使用不同的内部识别标识符,而是使用更加细致的搜索程序,因此,在搜索页面按分隔符分离出包名,丢弃常见的部分如“.com”。一旦包名出现在搜索页面,一次搜索将被视为结束,否则会继续搜索各个应用市场并返回搜索结果。

最终,根据软件发布者的习惯,应用程序可能在被发布到其他检测的应用市场之前就已经出现在种子中了, 因此,搜索模块将定期访问所有的应用市场搜索目标应用程序,不管这些应用程序是否已经在种子样本程序中。

## 4 基于模糊哈希算法的指纹匹配模块

恶意 Android 应用程序检测是本文的重要内容, 如何快速有效的检测恶意软件是研究的重点。传统的恶意检测一般采用字符串逐个对比的匹配法[7],这种方法效率低下,文中拟将 APK 的 classes.dex 文件的源码生成指纹,再将指纹和已知的恶意源码指纹进行比对。指纹生成即数据压缩,利用算法函数把大容量文件压缩成一个字符串, 计算字符串的相似性。当前,通常采用哈希算法生成指纹,但是哈希算法对输入很敏感,一旦检测对象的恶意代码有微小的改动,哈希算法就失效了,另外恶意应用程序的源码是不断变化的,这时用传统的哈希算法生成指纹检测同样失效。文中采用模糊哈希算法来解决这些问题。

分析 classes.dex 文件后, 文件包含可执行的字节码 Dalvik,将其反编译后就得到 java 源码,恶意应用程序会在源码中添加相应的命令序列,匹配模块将待检测应用程序的文件 classes.dex 同恶意代码特征库进行对比,以判断程序的安全性。本模块采用模糊哈希算法(fuzzy hashing),将待检测应用程序源码同恶意源码进行指纹匹配,利用相似度判断应用程序的恶意性。模块流程图如图 2 所示。

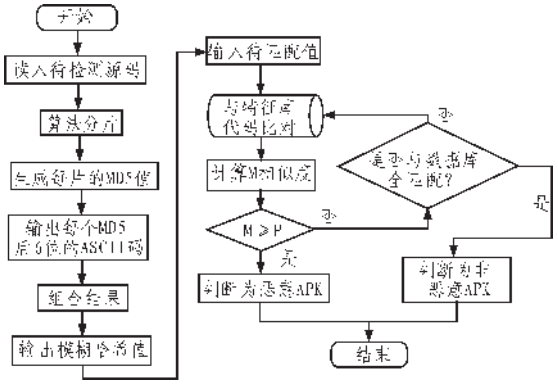


图 2　匹配模块流程图

Fig. 2　Matching module flow chart

在文中采用模糊哈希生成指纹的算法中，先将文件分成许多的小片段，分别对这些片段映射成哈希值，然后整合这些哈希值，最终生成指纹信息。首先选取合适的触发值：

$$b_{init}=b_{min}2^{\left[\log_2\left(\frac{n}{sb_{min}}\right)\right]} \tag{1}$$

式中：$S$ 为哈希串最大的长度；$n$ 为文本长度；$b_{min}$ 为最小触发值。

再校验触发条件：

$$rolling\_value\%b_{init}=b_{init}-1 \tag{2}$$

$$rolling\_value\%2b_{init}=2b_{init}-1 \tag{3}$$

式中：rolling_value——滚动哈希值；$b_{init}$ 为触发值。

将用 Alder-32 算法计算出的滚动哈希值 rolling_value 与前面计算出的触发值 $b_{init}$ 比较，直到满足上面等式。同理根据等式，计算出触发值 $2b_{init}$，确定了触发值后，通过 FNV 算法计算出哈希值。并取哈希值的 6 个低有效位的一个 base64 码。最终得到指纹的两部分：一部分是基于触发值 $b_{init}$ 的 base64 码，一部分是基于触发值 $2b_{init}$ 的 base64 码。

在得到指纹信息后，通过比较相似度的方法判断两个 APK 是否相似，文中采用加权编辑距离判断相似性。有以下计算公式：

$$ed(S_1,S_2)=i+d+3c+5w \tag{4}$$

$$c+w\leqslant\min(l_1,l_2) \tag{5}$$

$$i+d=|l_1-l_2| \tag{6}$$

$$M=\left[1-\frac{S*ed(S_1,S_2)}{64(l_1+l_2)}\right]*100 \tag{7}$$

式中：$S_1$ 为指纹 1；$S_2$ 为指纹 2；$ed(S_1,S_2)$ 为 $S_1$ 与 $S_2$ 之间的编辑距离；$i$ 为插入操作的次数；$d$ 为删除操作次数；$c$ 为替换的次数；$w$ 为交换的次数；$\min(l_1,l_2)$ 为 $l_1$ 和 $l_2$ 的最小值；$l_1$ 为指纹 $S_1$ 的长度；$l_2$ 为指纹 $S_2$ 的长度；$M$ 为匹配指数。

先判断 $S_1$ 到 $S_2$ 需要的最少次数，对不同操作给出一个权值，将结果相加，得到加权编辑距离。在算匹配指数时，将加权编辑距离除以 $S_1$、$S_2$ 的长度和，再映射到 0 到 100 之间的一个整数值之间。其中，$S$ 是指纹最大长度默认值是 64，$M$ 是匹配指数。如果 $M$ 值越接近 100，则两个文本的相似度越高。

## 5　结束语

文中主要对 APK 文件抓取系统中匹配模块进行了详细的设计，首先给出了 APK 文件抓取系统总体框架，之后详细描述了匹配模块功能，APK 匹配算法，并采用模糊哈希算法实现对匹配模块的设计，极大的提高了 APK 抓取系统的可靠性。

**参考文献：**

[1] Racic R，Ma D，Chen H. Exploiting mms vulnerabilities to stealthily exhaust mobile phone′s battery [C]//Securecomm and Workshops，2006: 1-10.

[2] 刘泽衡. 基于 Android 智能手机的安全检测系统的研究与实现[D]. 哈尔滨：哈尔滨工业大学，2011.

[3] Inoue D，Eto M，Yoshioka K，et al. nicter: An incident analysis system toward binding network monitoring with malware analysis [C]//Information Security Threats Data Collection and Sharing，2008. WISTDCS′08. WOMBAT Workshop on. IEEE，2008: 58-66.

[4] Inoue D，Yoshioka K，Eto M，et al. Malware behavior analysis in isolated miniature network for revealing malware′s network activity[C]//Communications，2008. ICC′08. IEEE International Conference on. IEEE，2008:1715-1721.

[5] Bayer U，Habibi I，Balzarotti D，et al. A view on current malware behaviors [C]//USENIX workshop on large-scale exploits and emergent threats (LEET).2009.

[6] Bayer U，Moser A，Kruegel C，et al. Dynamic analysis of malicious code[J]. Journal in Computer Virology，2006，2(1): 67-77.

[7] Van Randwyk J，Chiang K，Lloyd L，et al. Farm: An automated malware analysis environment [C]//Security Technology，2008. ICCST 2008. 42nd Annual IEEE International Carnahan Conference on. IEEE，2008:321-325.