(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 103136471 B
(45)授权公告日 2015.12.16

(21)申请号 201110382248.5
(22)申请日 2011.11.25
(73)专利权人 中国科学院软件研究所
地址 100190 北京市海淀区中关村南四街4号
(72)发明人 焦四辈 苏璞睿 应凌云 杨轶
(74)专利代理机构 北京君尚知识产权代理事务所(普通合伙) 11200
代理人 余长江
(51)Int.Cl.
G06F 21/56(2013.01)

(56)对比文件
CN 1818823 A,2006.08.16, 说明书第2-5页.
CN 101013461 A,2007.08.08, 全文.
CN 101281571 A,2008.10.08, 全文.

审查员 王青

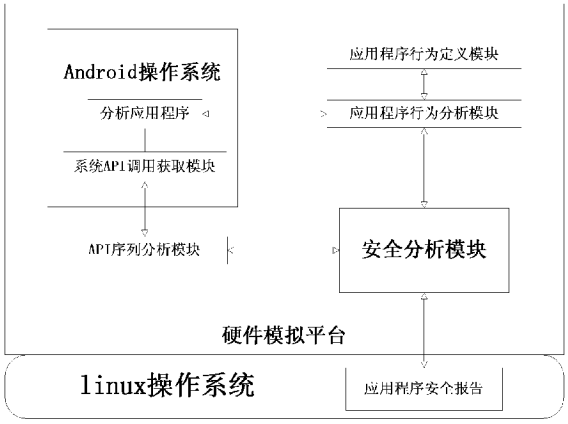
权利要求书2页 说明书5页 附图2页

(54)发明名称

一种恶意Android应用程序检测方法和系统

(57)摘要

本发明公开了一种恶意Android应用程序检测方法和系统,属于计算机软件技术领域。本方法为:1)将待测应用程序中的行为划分为若干类别;将待测应用程序每个按钮与一个或多个类别行为对应,得到应用程序按钮——行为模型;2)采集硬件模拟器执行按钮时的应用程序信息,识别出当前操作对应的按钮;根据应用程序按钮——行为模型得到该按钮要执行的操作行为;3)采集当前按钮触发的硬件模拟器底层API调用序列,根据API序列模型得到该按钮对应执行的操作行为;4)将步骤2)与步骤3)确定的操作行为进行对比,如果不一致则将该待测应用程序确定为恶意程序。本发明简化了分析复杂度,大大提高了分析检测的效率和准确性。



CN 103136471 B

1. 一种恶意 Android 应用程序检测方法，其步骤为：

1) 将待测 Andorid 应用程序中的行为划分为若干类别；将待测 Android 应用程序中每个页面上的每个按钮与一个或多个类别行为对应，得到应用程序按钮 ---- 行为模型；

2) 采集硬件模拟器执行按钮时的应用程序信息，识别出当前操作对应的按钮；根据所述应用程序按钮 ---- 行为模型得到该按钮要执行的操作行为；

3) 采集当前按钮触发的硬件模拟器底层 API 调用序列，根据 API 序列模型得到该按钮对应执行的操作行为；所述 API 序列模型记录 API 序列与步骤 1) 所划分行为类别的对应关系；

4) 将步骤 2) 确定的操作行为与步骤 3) 确定的操作行为进行对比，如果不一致则将该待测 Android 应用程序确定为恶意程序；

其中，采集所述 API 调用序列的方法为：首先实时监控硬件模拟器 CPU 运行，通过反汇编引擎反汇编硬件模拟器的 CPU 指令；然后根据反汇编的指令调用地址，在 API 表里查找对应的 API，记录调用的 API，得到所述 API 调用序列。

2. 如权利要求 1 所述的方法，其特征在于所述类别包括：联网，短信，访问地址薄，访问 sim 卡信息，访问多媒体数据，执行程序。

3. 如权利要求 1 所述的方法，其特征在于采用一 hash 表记录所述 API 调用序列的 API 地址和 API。

4. 如权利要求 1 或 2 所述的方法，其特征在于所述识别出当前操作对应的按钮的方法为：首先采集硬件模拟器中的各种点击、滑动以及晃动操作；然后通过模式匹配识别出目前操作为待测 Android 应用程序中哪个页面中的哪个按钮。

5. 如权利要求 1 或 2 所述的方法，其特征在于建立所述 API 序列模型的方法为：根据 Android 应用程序的开发文档确定 Android 应用程序中每一 API 的作用；然后对每一所述行为进行 API 序列验证，得到每一所述行为的 API 序列。

6. 一种恶意 Android 应用程序检测系统，其特征在于包括一硬件模拟器，一应用程序行为定义模块，一应用程序行为分析模块，一底层 API 获取模块，一 API 序列分析模块，一安全分析模块；其中：

所述硬件模拟器用于模拟 Android 应用程序运行环境；

所述应用程序行为定义模块用于将待测 Andorid 应用程序中的行为划分为若干类别，并将待测 Android 应用程序中每个页面上的每个按钮与一个或多个类别行为对应，建立应用程序按钮 ---- 行为模型；

所述应用程序行为分析模块用于从所述硬件模拟器中采集 Android 应用程序信息，识别出操作对应的按钮，并根据所述应用程序按钮 ---- 行为模型得到该按钮要执行的操作行为；

所述底层 API 获取模块用于从所述硬件模拟器中采集按钮触发的硬件模拟器底层 API 调用序列；

所述 API 序列分析模块用于根据获取的 API 调用序列，利用 API 序列模型得到该按钮对应执行的操作行为；所述 API 序列模型记录 API 序列与所述应用程序行为定义模块所划分行为类别的对应关系；

所述安全分析模块用于根据所述所述应用程序行为分析模块确定的操作行为与所述

API 序列分析模块确定的操作行为进行对比，如果不一致则确定待测 Android 应用程序为恶意程序；

其中，所述底层 API 获取模块获取所述 API 调用序列的方法为：首先实时监控硬件模拟器 CPU 运行，通过反汇编引擎反汇编硬件模拟器的 CPU 指令；然后根据反汇编的指令调用地址，在 API 表里查找对应的 API，记录调用的 API，得到所述 API 调用序列。

7. 如权利要求 6 所述的系统，其特征在于所述类别包括：联网，短信，访问地址薄，访问 sim 卡信息，访问多媒体数据，执行程序。

8. 如权利要求 6 或 7 所述的系统，其特征在于采用一 hash 表记录所述 API 调用序列的 API 地址和 API。

9. 如权利要求 6 或 7 所述的系统，其特征在于采集的所述 Android 应用程序信息包括所述操作是哪个页面中的哪个按钮。

一种恶意 Android 应用程序检测方法和系统

技术领域

[0001] 本发明主要涉及恶意 Android 应用程序检测技术，更确切的说是基于应用程序行为和底层 API 行为分析的恶意 Android 应用程序检测技术，属于计算机软件技术领域。

背景技术

[0002] 随着移动网络的不断发展，手机已经成为人类现代生活的不可或缺的一部分。而 Android 手机操作系统，占据了智能手机的半壁江山，并且出货量以每天 50 万部的速度增长。由于智能手机功能的多样性和复杂性，人们通过手机进行的工作也越来越多，不再局限于发短信、打电话，还可以玩游戏、上网、看视频、听音乐、购物等等，手机应用程序的数量也随之爆炸性的增长，Android 应用商店的程序数量已经突破 20 万个。手机应用程序增长的同时也带来了很多窃取用户信息的恶意软件。Juniper 网络公司发布的最新移动安全报告显示，Android 平台上的恶意软件数量激增了 400％。Android 应用程序商店已经成为恶意软件分布最多的移动智能平台。所以，对 Android 应用程序进行安全性分析是迫切需要的。但是由于软件数量众多，软件功能越来越复杂，且其恶意行为更加隐蔽，造成分析起来难度大、效率低。

[0003] 应用程序检测分析的时候，通常有两种基本方法：一种是静态分析，即静态反汇编程序代码，通过人工或者自动分析反汇编代码来分析程序安全性。一种是动态分析，即在程序运行过程中，获取其运行数据，分析其安全性。第一种方法，人工分析准确性较高，但是需要很强的专业知识，花费的人力物力很大，不适合大规模快速的安全分析；而自动分析误报和漏洞率相当高。第二种方法需要在软件运行过程中能实时的获取其运行数据，根据获取的大量数据进行分析，进而得出应用程序安全分析结果。因此，如何实时获取软件运行的数据，以及如何对这些数据进行分析，从而准确的得到应用程序检测分析结果，成为动态分析方法的研究的难点和热点。

发明内容

[0004] 针对上述问题，本发明的目的在于提供一种高效率且更加准确的恶意 Android 应用程序检测方法，利用该方法，通过简单设置，一个不具有专业分析知识的人也可以快速准确的对 Android 应用程序进行分析，确定是否为恶意程序。

[0005] 根据以上目的，实现本发明的一个具体的方案，其系统结构示意图如图 1 所示：至少包括一个硬件模拟器，一个应用程序行为定义模块，一个应用程序行为分析模块，一个底层 API(application program interface 应用程序接口）获取模块，一个 API 序列分析模块，一个安全分析模块。硬件模拟器模拟 Android 运行环境，应用程序行为分析模块和 API 序列分析模块从硬件模拟器中采集信息，安全分析模块判断应用程序的安全性。在这个最简模式下，正常的 Android 应用程序处理过程包括如下步骤：

[0006] 1) 应用程序行为定义模块。首先，将 Andorid 中的行为分为 6 个类别：联网，短信，访问地址薄，访问 sim 卡信息，访问多媒体数据，执行程序。其次，把需要分析的应用程序，

每个页面上的每个按钮，跟 Android 中的 6 类行为对应起来，一个按钮可以对应多个行为。最后，把对应关系建模，建立应用程序按钮 ---- 行为模型。按钮是待分析程序页面上原有的，每个按钮有独一无二的 id，通过获取按钮的 id 即可得到相应的按钮。

[0007] 2) 应用程序行为分析模块采集硬件模拟器中的应用程序信息，应用程序信息包括：确定目前操作的是哪个页面中的哪个按钮，有些点击可能没有涉及按钮，就过滤掉；这部分采用了模式匹配技术。具体的方法如下：首先，在应用程序定义模块中，记录了应用程序每个按钮对应的行为。应用程序行为分析模块，采集硬件模拟器中的各种点击、滑动以及晃动操作，通过模式匹配，即模式识别技术，识别出目前操作对应的按钮，进而根据所建的应用程序按钮 ---- 行为模型得到即将进行的操作行为。

[0008] 3)API 获取模块，采集硬件模拟器底层的 API 序列，这部分采用了反汇编、API 函数识别等技术。具体的方法如下：首先，实时监控硬件模拟器 CPU 运行，通过反汇编引擎反汇编硬件模拟器的 CPU 指令，然后根据反汇编的指令调用地址，在 API 表里查找对应的 API，记录这些 API 调用。

[0009] 4)API 序列分析模块，这部分采用 API 序列识别技术。具体的方法如下：首先，建立 API 序列模型，该模型实现 API 序列和事件一一对应，建立模型的方法是：根据 Android 开发文档，得知 API 的作用，根据经验，对每个事件（即所划分的行为）进行 API 序列验证。例如，函数 open/read 可以用来打开驱动或者文件，进行读写，实现读取文件内容或者是读取驱动内容的功能，如果 open 的参数是无线通信模块，读取的内容被短信处理程序调用，可以得知进行的是接收短信操作。检测的方法是：检测到的 API 序列是 open、read、copy，检查 open 的参数，确认其打开的是什么，接着检查 read 的参数，确认其返回值，然后检查 copy 的参数，确认 read 出来的数据复制到了哪个进程空间里，如果 open 打开的是无线通信模块，copy 将 read 读取的内容复制到了短信进程空间中，可以判断这一系列操作是接收短信。其次，API 序列分析模块对 API 获取模块采集硬件模拟器的所有 API 调用进行分析，根据 API 序列模型发现 API 序列所对应的事件，进而识别出硬件模拟器正在进行的操作。

[0010] 5) 安全分析模块，根据应用程序行为分析模块和 API 序列分析模块得出的结果，判断应用程序的安全性。具体的方法如下：应用程序行为分析模块得出应用程序即将进行的操作，安全分析模块得到该数据后，然后将其与 API 序列分析模块的结果对比，如果两者符合，则证明应用程序的底层操作符合其功能描述，如果不符合，则说明应用程序底层进行了不在其功能描述范围内的操作，确定为恶意程序；最后生成安全分析报告。

[0011] 与现有技术相比，本发明的优点在于：

[0012] 本发明使用硬件模拟器运行 Android，可以完整的获取 Android 运行的所有数据，保证分析工作的完整性、准确性、可靠性和真实性。通过应用程序分析模块和底层 API 序列分析模块对比，直接检测应用程序是否进行了不符合功能描述的操作，进而发现恶意行为。对分析人员来说，简化了分析复杂度，大大提高了分析检测的效率和准确性。

附图说明

[0013] 图 1 为最简化模式下恶意 Android 应用程序检测系统的结构示意图；

[0014] 图 2 为最简化模式下恶意 Android 应用程序检测方法的流程图。

具体实施方式

[0015] 如图 2 所示，为最简化模式下的恶意 Android 应用程序检测方法的实现。具体的实现方式如下：

[0016] 1) 应用程序行为定义，具体的数据使用如下格式：

[0017]

```
struct program
{
  char *page;
  char *button;
  int function[6];
}
```

[0018] 其中 page 表示应用程序的一个操作界面，button 表示操作界面上的一个按钮，function 表示操作界面上 button 的功能，具体的功能如下几种：

[0019] #define 1 联网

[0020] #define 2 短信

[0021] #define 3 访问地址薄

[0022] #define 4 访问 sim 卡信息

[0023] #define 5 访问多媒体数据

[0024] #define 6 执行程序

[0025] 这部分内容由用户辅助定义，即对一个应用程序内的所有按钮进行功能定义。

[0026] 2) 执行应用程序。

[0027] 在受控制的 Android 操作系统中，运行待分析的应用程序，用户依次点击应用程序中的每个按钮，对其进行分析。

[0028] 3) 应用程序行为分析。

[0029] 获取应用程序的运行数据，按照应用程序行为定义模块建立的模型进行模式匹配，得到当前的按钮即将进行的操作。实现代码如下：

[0030] // 获取当前所在页面

[0031] page = getcurrentpage() ;

[0032] // 获取点击的按钮

[0033] button = getcurrentbutton() ;

[0034] // 匹配应用程序行为定义模块建立的模型

[0035] function = getfunction(page, button) ;

[0036] 最后得到当前所在页面，点击的按钮对应的操作行为。

[0037] 4)Android 底层 API 获取。

[0038] 实时的监控模拟 CPU 运行，用反汇编引擎解析 CPU 指令，根据反汇编的指令调用地址，找到对应的 API，然后记录这些 API 调用。

[0039] 创建 hash 表，存储 API 地址和 API，方便查询。用以下代码实现。

[0040]

```
//如果反汇编的cpu指令中，调用地址的hash值为1，说明该地址指向某一API
if ( hash [demisser(cpuEIP)] == 1)
{
    //通过地址获取API
    apiname = getAPIbyaddress( demisser(cpuEIP) );
    //创建API序列，存储API调用
    apilist[i] = apiname;
}
```

[0041] 5)API序列分析

[0042] 建立API序列模型,使用以下格式：

[0043]

```
struct apilist
{
    char *api[5];      //API序列，最长为6
    char *function;    //记录API序列作用
    int count;      //记录API序列长度
}
struct apilistfind
{
    apilist api;       //存储api序列
    char *function;
    int count;         //存储和API序列中的API匹配的个数
}
```

[0044] 根据模型,对获取的API序列进行匹配,分析API序列进行的操作。代码如下：

[0045]

```
//获取的API中，跟API序列匹配
if( apilistfind.count == apilistfind.api.count )
{
    apilistfind.function = apilistfind.api.function;
}
```

[0046] 6)安全分析模块,根据应用程序分析模块和API序列分析模块得出的结果,判断应用程序的安全性。

[0047] 某一页面上,触摸一个按钮后,应用程序分析模块和API序列分析模块得到分析

结果,传递过来,判断代码如下

[0048]

```
//判断两个模块的分析结果是否一致，如果一致，继续分析，如果不一致，记录
if ( programfunction != apifunction )
{
    //记录该操作
    save(page, buttion, programfunction, apifunction);
}
```

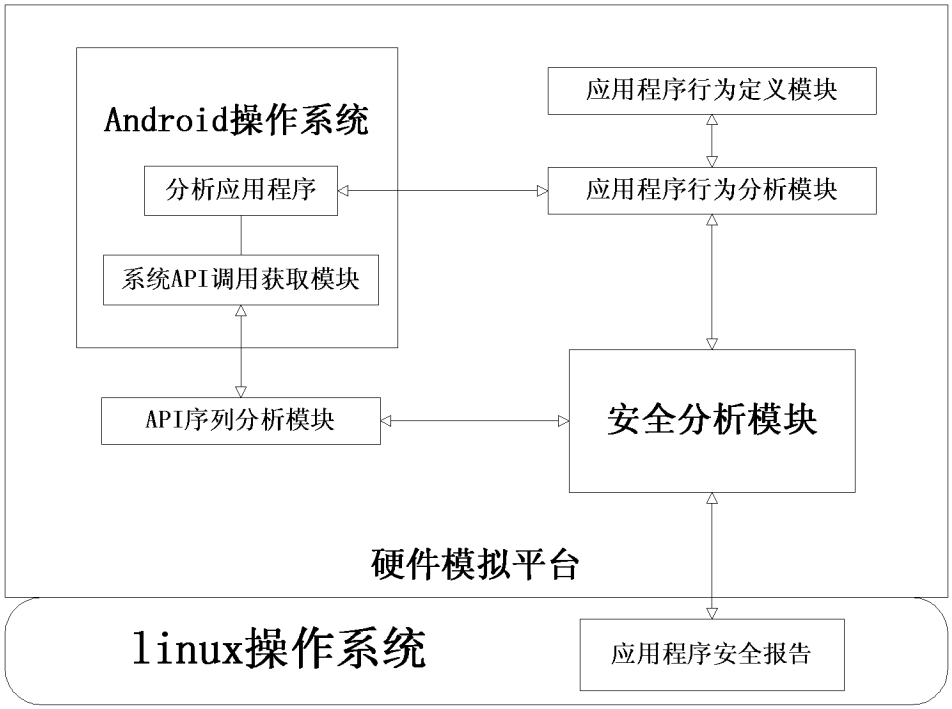

Android操作系统
分析应用程序
系统API调用获取模块
应用程序行为定义模块
应用程序行为分析模块
API序列分析模块
安全分析模块
硬件模拟平台
linux操作系统
应用程序安全报告

图 1

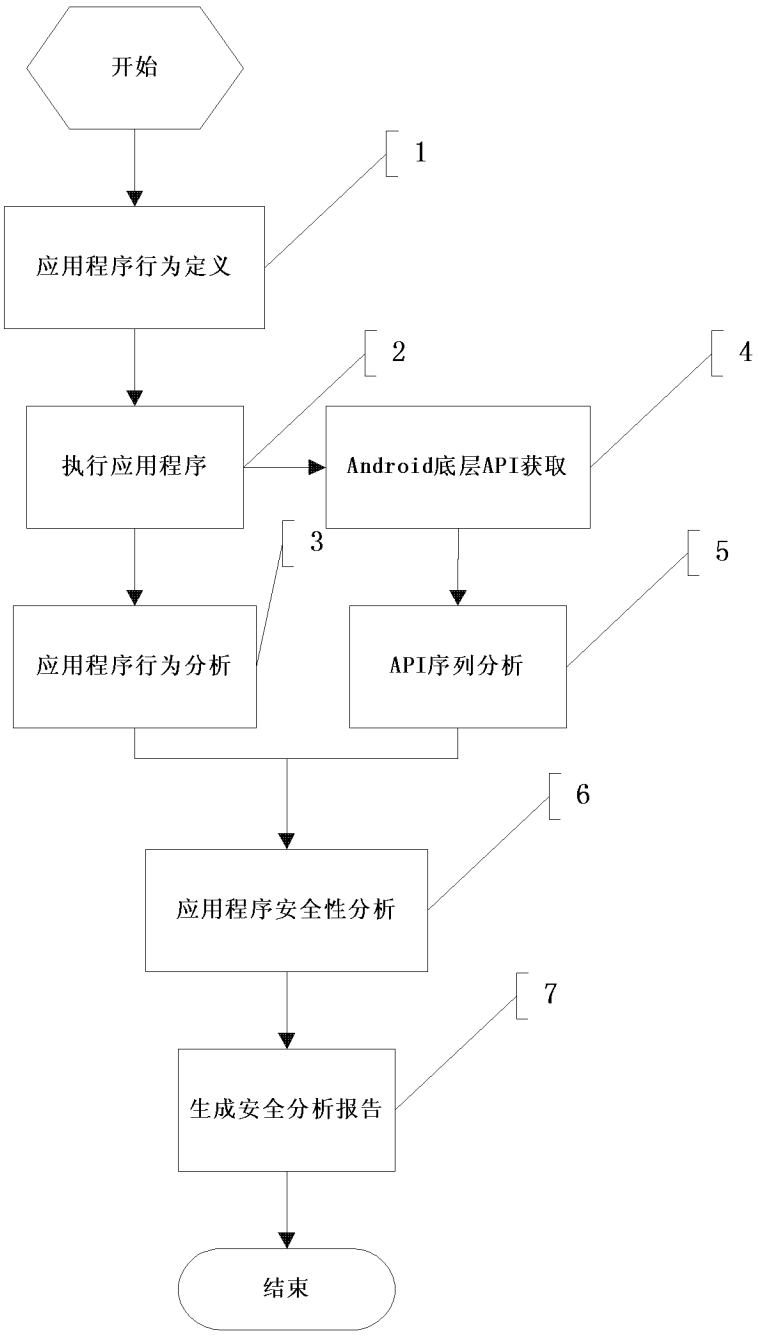

开始
1
应用程序行为定义
2
4
执行应用程序
Android底层API获取
3
5
应用程序行为分析
API序列分析
6
应用程序安全性分析
7
生成安全分析报告
结束

图 2