(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102254113 A
(43)申请公布日 2011.11.23

(21)申请号 201110174482.9

(22)申请日 2011.06.27

(71)申请人 深圳市安之天信息技术有限公司
地址 518067 广东省深圳市南山区蛇口南海大道 1079 号花园城数码大厦 B 座 301B#

(72)发明人 肖梓航 李柏松

(51)Int.Cl.
G06F 21/00(2006.01)
H04L 29/06(2006.01)

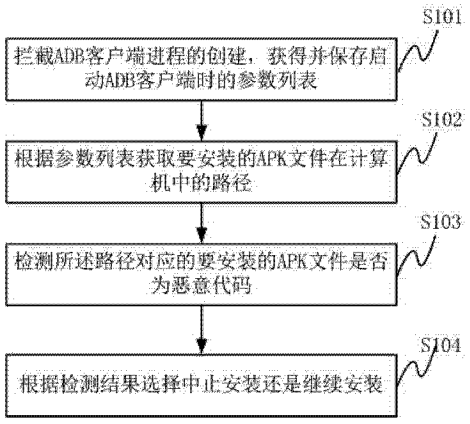
权利要求书 1 页 说明书 6 页 附图 1 页

(54)发明名称

一种检测和拦截移动终端恶意代码的方法及系统

(57)摘要

本发明公开了一种检测和拦截移动终端恶意代码的方法，包括：在计算机中拦截 ADB 客户端进程的创建，获得并保存启动 ADB 客户端时的参数列表；根据参数列表判断如果是安装 APK 文件(Android 应用程序)，则获取要安装的 APK 文件在计算机中的路径；检测所述路径对应的要安装的 APK 文件是否为恶意代码；根据检测结果选择终止安装或者继续安装。本发明还公开了一种在计算机中检测和拦截移动终端恶意代码的系统。本方法及系统不消耗手机系统资源，适用于目前所有类型的安装方法和第三方手机辅助软件，并可以有效抵御恶意代码通过计算机向智能手机传播。



CN 102254113 A

1. 一种检测和拦截移动终端恶意代码的方法，其特征在于，包括以下步骤：

在计算机中拦截 ADB 客户端进程的创建，获得并保存启动 ADB 客户端时的参数列表；

根据参数列表判断如果是安装 APK 文件（Android 应用程序），则获取要安装的 APK 文件在计算机中的路径；

检测所述路径对应的要安装的 APK 文件是否为恶意代码；

判断如果扫描结果为恶意代码则向用户报警并退出安装进程；如果扫描结果不是恶意代码则根据参数列表启动真正的 ADB 客户端进行 APK 文件安装。

2. 如权利要求 1 所述的检测和拦截移动终端恶意代码的方法，其特征在于，在计算机中拦截 ADB 客户端进程的创建的方法包括：替换 adb.exe 可执行文件、对 adb.exe 做映像劫持、钩挂创建进程相关的系统 API。

3. 如权利要求 1 所述的检测和拦截移动终端恶意代码的方法，其特征在于，根据参数列表判断如果不是安装 APK 文件，则根据参数列表启动真正的 ADB 客户端进行工作。

4. 如权利要求 1 所述的检测和拦截移动终端恶意代码的方法，其特征在于，获取要安装的 APK 文件在计算机中的路径的方法是解析参数列表中的 ADB 客户端的启动参数，第一个参数 install 对应的最后一个参数是文件路径。

5. 如权利要求 1 所述的检测和拦截移动终端恶意代码的方法，其特征在于，调用外部具有 Android APK 文件检测能力的恶意代码检测工具或模块检测所述路径对应的要安装的 APK 文件是否为恶意代码。

6. 一种检测和拦截移动终端恶意代码的系统，其特征在于，包括：

拦截模块，用于在计算机中拦截 ADB 客户端进程的创建，获得并保存启动 ADB 客户端时的参数列表；

解析模块，用于根据参数列表判断如果是安装 APK 文件（Android 应用程序），则获取要安装的 APK 文件在计算机中的路径；

检测模块，用于检测所述路径对应的要安装的 APK 文件是否为恶意代码；

处理模块，用于判断如果扫描结果为恶意代码则向用户报警并退出安装进程；如果扫描结果不是恶意代码则根据参数列表启动真正的 ADB 客户端进行 APK 文件安装。

7. 如权利要求 6 所述的检测和拦截移动终端恶意代码的系统，其特征在于，在计算机中拦截 ADB 客户端进程的创建的方法包括：替换 adb.exe 可执行文件、对 adb.exe 做映像劫持、钩挂创建进程相关的系统 API。

8. 如权利要求 6 所述的检测和拦截移动终端恶意代码的系统，其特征在于，解析模块具体用于根据参数列表判断如果不是安装 APK 文件，则通知处理模块根据参数列表启动真正的 ADB 客户端进行工作。

9. 如权利要求 6 所述的检测和拦截移动终端恶意代码的系统，其特征在于，解析模块具体还用于解析参数列表中的 ADB 客户端的启动参数，第一个参数 install 对应的最后一个参数是文件路径。

10. 如权利要求 6 所述的检测和拦截移动终端恶意代码的系统，其特征在于，检测模块具体用于调用外部具有 Android APK 文件检测能力的恶意代码检测工具或模块检测所述路径对应的要安装的 APK 文件是否为恶意代码。

一种检测和拦截移动终端恶意代码的方法及系统

技术领域

[0001] 本发明涉及移动终端恶意代码检测技术，特别是在计算机中检测和拦截移动终端恶意代码的方法及系统。

背景技术

[0002] 随着 Android 系统等智能手机平台的兴起，移动终端的恶意代码逐渐成为信息安全领域的又一重大威胁。目前，Android 中的恶意代码数量正呈现出爆炸式增长的趋势。从传播途径来看，绝大部分恶意代码通过软件下载站、非官方市场、手机论坛等渠道传播。用户从这些渠道下载伪装成正常应用程序的恶意代码到计算机中，然后使用辅助工具软件，通过 USB 数据线安装到 Android 手机之中。

[0003] 除了用户主动下载安装，从当前恶意代码的发展趋势来看，还将出现另一种混合型的攻击方式，即：计算机上的恶意代码（如木马等）在感染计算机后监控 USB 接口，一旦发现 Android 设备，就自动向该设备安装恶意代码。这种攻击不需要用户交互，具有一定的自动性；安装的恶意代码可以不显示在 Android 系统的应用程序列表中，用户无法觉察，具有一定的隐蔽性。因此，这类攻击方式具有极高的危害。

[0004] 目前对 Android 恶意代码检测一般部署在手机之中，即在手机中安装专用杀毒软件或安全保护软件。这类软件扫描、分析和判定手机中的应用程序，将占用一定的手机系统资源；杀毒软件中特征数据库的更新还将消耗网络流量。

[0005] 本发明从另一个角度，即用户在计算机中的为手机安装应用程序的过程中检测移动终端的恶意代码。

[0006] 事实上，从用户行为的角度，目前在计算机中的通过 USB 数据线为 Android 手机安装应用程序，有以下几种方法：

1. 使用 Android SDK（开发工具包）中的 ADB 命令行工具，通过 adb install 命令安装；

2. 使用第三方开发的手机辅助软件安装；

3. 双击 Android 应用程序的文件安装。

[0007] 经过分析发现，第 3 种方法实际上是第 2 种方法中所述辅助软件提供的附加功能；而第 2 种方法所述的辅助软件是通过第 1 种方法所述 ADB 命令行工具来安装应用程序的。

[0008] 因此，目前用户从计算机中的通过 USB 数据线向 Android 手机安装应用程序，最终都是使用 ADB 命令行工具的相应命令来实现。

[0009] ADB（Android Debug Bridge，Android 调试桥接器）是 Android SDK 中用于管理和控制 Android 设备的工具。它由三个运行部件组成：

1. ADB 客户端，是计算机中的一个命令行工具，向用户提供管理和控制功能，例如为设备安装指定的应用程序等；

2. ADB 服务端，是计算机中的一个后台进程，管理设备，并负责计算机与设备之间的数据交换；

3.ADB 守护进程 adbd，是运行在 Android 系统中的一个进程，它接收 ADB 服务端发来的数据并执行发来的指令。

[0010] 其中，ADB 客户端和 ADB 服务端在物理上存在于同一个可执行文件之中，在 Windows 系统中名为 adb.exe。但两者运行时是不同的进程，执行不同的代码。ADB 客户端负责与用户交互，执行完命令后就退出；而 ADB 服务端在第一次启动后就一直运行于计算机中。

[0011] Android 辅助软件或用户使用 adb.exe 命令行工具安装 Android 应用程序的调用方法如下：

adb.exe install [options] MyApp.apk

根据 Android 的规定，install 必须是第一个参数，要安装的 APK 文件必须是最后一个参数，两者之间可以是一些可选参数，也可以没有参数。

发明内容

[0012] 本发明针对在 Windows 平台下使用 ADB 客户端(即 adb.exe 命令行工具)向手机安装 Android 应用程序的方法特点，在安装前获得要安装文件在计算机中的路径，调用第三方检测模块检查该文件是否恶意代码。

[0013] 本发明提供了一种检测和拦截移动终端恶意代码的方法，包括：

在计算机中拦截 ADB 客户端进程的创建，获得并保存启动 ADB 客户端时的参数列表；

根据参数列表判断如果是安装 APK 文件(Android 应用程序)，则获取要安装的 APK 文件在计算机中的路径；

检测所述路径对应的要安装的 APK 文件是否为恶意代码；

判断如果扫描结果为恶意代码则向用户报警并退出安装进程；如果扫描结果不是恶意代码则根据参数列表启动真正的 ADB 客户端进行 APK 文件安装。

[0014] 进一步的，在计算机中拦截 ADB 客户端进程的创建的方法包括：替换 adb.exe 可执行文件、对 adb.exe 做映像劫持、钩挂创建进程相关的系统 API。

[0015] 进一步的，根据参数列表判断如果不是安装 APK 文件，则根据参数列表启动真正的 ADB 客户端进行工作。

[0016] 进一步的，获取要安装的 APK 文件在计算机中的路径的方法是解析参数列表中的 ADB 客户端的启动参数，第一个参数 install 对应的最后一个参数是文件路径。

[0017] 进一步的，调用外部具有 Android APK 文件检测能力的恶意代码检测工具或模块检测所述路径对应的要安装的 APK 文件是否为恶意代码。

[0018] 本发明还提供了一种检测和拦截移动终端恶意代码的系统，包括：

拦截模块，用于在计算机中拦截 ADB 客户端进程的创建，获得并保存启动 ADB 客户端时的参数列表；

解析模块，用于根据参数列表判断如果是安装 APK 文件(Android 应用程序)，则获取要安装的 APK 文件在计算机中的路径；

检测模块，用于检测所述路径对应的要安装的 APK 文件是否为恶意代码；

处理模块，用于判断如果扫描结果为恶意代码则向用户报警并退出安装进程；如果扫描结果不是恶意代码则根据参数列表启动真正的 ADB 客户端进行 APK 文件安装。

[0019] 进一步的，在计算机中拦截ADB客户端进程的创建的方法包括：替换adb.exe可执行文件、对adb.exe做映像劫持、钩挂创建进程相关的系统API。

[0020] 进一步的，解析模块具体用于根据参数列表判断如果不是安装APK文件，则通知处理模块根据参数列表启动真正的ADB客户端进行工作。

[0021] 进一步的，解析模块具体还用于解析参数列表中的ADB客户端的启动参数，第一个参数install对应的最后一个参数是文件路径。

[0022] 进一步的，检测模块具体用于调用外部具有Android APK文件检测能力的恶意代码检测工具或模块检测所述路径对应的要安装的APK文件是否为恶意代码。

[0023] 本发明的有益效果如下：

本发明是一种在计算机中检测移动终端恶意代码的方法，与传统在移动终端之中检测的方法相比，不占用和消耗手机的系统资源，包括CPU、内存、存储空间、网络流量、电量等。此外，因为计算机具有更强大的计算能力，可以配合复杂深入的细粒度检测方法。

[0024] 本发明适用面广，对通过计算机中的各种Android手机辅助软件来向手机安装应用程序的方法，均能有效拦截、检测和中止；可以有效发现和阻止前文所述新型混合攻击，防止恶意代码通过计算机感染智能手机；相比于计算机中的杀毒软件对文件和系统的实时监控，本发明的检测仅由安装动作触发，不占用大量计算机资源，精确高效。

附图说明

[0025] 为了更清楚地说明本发明或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本发明中记载的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

[0026] 图1为本发明检测和拦截移动终端恶意代码的方法流程图；

图2为本发明检测和拦截移动终端恶意代码的方法实施例流程图；

图3为本发明检测和拦截移动终端恶意代码的系统示意图。

具体实施方式

[0027] 为了使本技术领域的人员更好地理解本发明实施例中的技术方案，并使本发明的上述目的、特征和优点能够更加明显易懂，下面结合附图对本发明中技术方案作进一步详细的说明。

[0028] 本发明提供一种检测和拦截移动终端恶意代码的方法及系统，在计算机中的拦截向Android智能手机安装应用程序的进程，通过其启动参数列表获得要安装程序在计算机中的路径，调用外部恶意代码检测模块检测要安装的应用程序是否恶意代码。如果是恶意代码，则向用户提示并中止安装。该方法及系统不消耗手机系统资源，适用于目前所有类型的安装方法和第三方手机辅助软件，并可以有效抵御恶意代码通过计算机向智能手机传播。

[0029] 首先介绍本发明提供的一种检测和拦截移动终端恶意代码的方法，具体步骤如图1所示：

S101、在计算机中拦截ADB客户端进程的创建，获得并保存启动ADB客户端时的参数列

表；

在计算机中拦截 ADB 客户端进程的创建的方法包括：替换 adb.exe 可执行文件、对 adb.exe 做映像劫持、钩挂创建进程相关的系统 API。

[0030] S102、根据参数列表判断如果是安装 APK 文件（Android 应用程序），则获取要安装的 APK 文件在计算机中的路径；

根据参数列表判断如果不是安装 APK 文件，则根据参数列表启动真正的 ADB 客户端进行工作。

[0031] 获取要安装的 APK 文件在计算机中的路径的方法是解析参数列表中的 ADB 客户端的启动参数，第一个参数 install 对应的最后一个参数是文件路径。

[0032] S103、检测所述路径对应的要安装的 APK 文件是否为恶意代码；

调用外部具有 Android APK 文件检测能力的恶意代码检测工具或模块检测所述路径对应的要安装的 APK 文件是否为恶意代码。

[0033] S104、判断如果扫描结果为恶意代码则向用户报警并退出安装进程；如果扫描结果不是恶意代码则根据参数列表启动真正的 ADB 客户端进行 APK 文件安装。

[0034] 下面给出本发明的具体实现方法，如图 2 所示：

S201、拦截 ADB 客户端

拦截是使得 ADB 客户端启动时，会先执行本发明后续步骤的实现代码。后文将实现了本发明后续步骤的可执行文件称之为劫持文件。本发明可以使用三种拦截发明中任何一种，其实现方法如下：

1）替换 adb.exe 可执行文件

通过查询注册表、查询环境变量、ADB 服务端进程可执行文件路径、文件夹遍历等方法，找到计算机中的 adb.exe 可执行文件，将其重命名为 adb_.exe，然后将劫持文件拷贝至同目录下并命名为 adb.exe。这样，其他程序对 adb.exe 的调用，实际上就启动了劫持文件。

[0035] 2）对 adb.exe 做映像劫持

Windows 操作系统中，可以设置命令行程序的默认调试器，设置后，启动该应用程序将直接启动其调试器，并将其启动参数传给调试器。映像劫持即利用这一系统机制，将劫持文件设置为要劫持目标的调试器。

[0036] 具体操作方法是，在 Windows 系统的注册表中，进入下列路径：

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options

在其中创建名为 adb.exe 的注册表子键，在该子键下新建名为 debugger 的字符串值，将其值设置为劫持文件的路径，即完成了映像劫持。

[0037] 完成后，任何程序调用 adb.exe，系统会直接启动其调试器，即劫持文件。

[0038] 3）钩挂（hook）用于创建进程的系统 API

API 钩挂是 Windows 系统中已经成熟的一种技术，有多种方法实现，例如微软的 Detours 开发库。可以用任何一种实现方法，钩挂用于创建进程的 Windows 系统 API（例如 CreateProcess、CreateProcessA、CreateProcessW 等）。钩挂后，每次该 API 被调用时，判断其启动的进程是否 ADB 客户端进程，如果不是，则放行；如果是，则转而启动钩挂文件而不是真正启动 ADB 客户端进程。

[0039] S202、获得启动参数

根据 S201 步骤中拦截方法的不同，获得 ADB 客户端启动时的参数列表的方法也有所不同，具体而言：

当使用替换 adb.exe 可执行文件的方法时，应用程序启动 ADB 客户端的所有参数都被传递给劫持文件，因此只需取出当前程序的所有启动参数并保存即可；

当对 adb.exe 做映像劫持时，系统会将 adb.exe 的路径和其启动参数依次传递给劫持文件，因此，取出当前程序的所有启动参数，移除掉第一个参数之后即可；

当钩挂创建进程的系统 API 时，需要对 CreateProcess 等 API 的前两个参数综合判断，取出其中非空的一个，将其指向的字符串用空格分隔，移除掉第一个参数后，即得到 ADB 客户端的所有启动参数。

[0040] S203、判断是否安装

根据 S202 步骤得到的启动参数列表，采取大小写无关的比较判断其第一个参数是否等于“install”，如果等于，则是启动 ADB 客户端用于安装，转至 S204；否则，启动 ADB 客户端用于其它操作，不是用于安装，转至 S208。

[0041] S204、获得安装文件

根据 S202 步骤得到的启动参数列表，取得其中最后一个参数，将其作为安装文件的路径保存。

[0042] S205、调用检测模块

本步骤涉及在 PC 端具有对 Android 平台恶意代码检测能力的外部检测模块，例如杀毒软件等。将 S204 步骤获得的安装文件路径通过 Windows 消息发送给该检测模块，并等待其结果输出。获得检测结果后，转至 S206。

[0043] S206、判断是否恶意

解析 S205 步骤所获得的由外部检测模块对安装文件的检测结果，如果结果显示该文件是恶意代码，则转至 S207；否则，转至 S208。

[0044] S207、报警、退出

使用弹出对话框（MessageBox）将要安装的文件路径和外部检测模块检测结果显示，报告给用户。用户确认后，退出当前进程，从而中止了此次安装过程。

[0045] S208、启动 ADB 客户端

在 S201 步中使用不同的拦截方法，本步骤对应有不同的实现：

1）使用替换 adb.exe 可执行文件的方法

使用已被重命名的 adb_.exe 可执行文件，使用 S202 步骤中获得的启动参数列表，通过 CreateProcess() 系统调用启动真正的 ADB 客户端程序。

[0046] 2）使用对 adb.exe 做映像劫持的方法

临时修改注册表中，HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\adb.exe 键中字符串值 debugger 的名称为 nodebugger；

正常使用 adb.exe 可执行文件，使用 S202 步骤中获得的启动参数列表，通过 CreateProcess() 系统调用启动真正的 ADB 客户端程序；

将注册表中，上述子键中名为 nodebugger 的字符串值修改为 debugger。

[0047] 3）钩挂（hook）用于创建进程的系统 API

正常使用 adb.exe 可执行文件，使用 S202 步骤中获得的启动参数列表，通过 API 钩挂时保存的原始 CreateProcess 系统调用，启动 ADB 客户端程序。

[0048] 本发明还提供了一种检测和拦截移动终端恶意代码的系统，如图 3 所示，具体包括：

拦截模块 301，用于在计算机中拦截 ADB 客户端进程的创建，获得并保存启动 ADB 客户端时的参数列表；

解析模块 302，用于根据参数列表判断如果是安装 APK 文件（Android 应用程序），则获取要安装的 APK 文件在计算机中的路径；

检测模块 303，用于检测所述路径对应的要安装的 APK 文件是否为恶意代码；

处理模块 304，用于判断如果扫描结果为恶意代码则向用户报警并退出安装进程；如果扫描结果不是恶意代码则根据参数列表启动真正的 ADB 客户端进行 APK 文件安装。

[0049] 其中，拦截模块 301 在计算机中拦截 ADB 客户端进程的创建的方法包括：替换 adb.exe 可执行文件、对 adb.exe 做映像劫持、钩挂创建进程相关的系统 API。

[0050] 解析模块 302 具体用于根据参数列表判断如果不是安装 APK 文件，则通知处理模块 304 根据参数列表启动真正的 ADB 客户端进行工作。

[0051] 解析模块 302 具体还用于解析参数列表中的 ADB 客户端的启动参数，第一个参数 install 对应的最后一个参数是文件路径。

[0052] 检测模块 303 具体用于调用外部具有 Android APK 文件检测能力的恶意代码检测工具或模块检测所述路径对应的要安装的 APK 文件是否为恶意代码。

[0053] 虽然通过实施例描绘了本发明，本领域普通技术人员知道，本发明有许多变形和变化而不脱离本发明的精神，希望所附的权利要求包括这些变形和变化而不脱离本发明的精神。

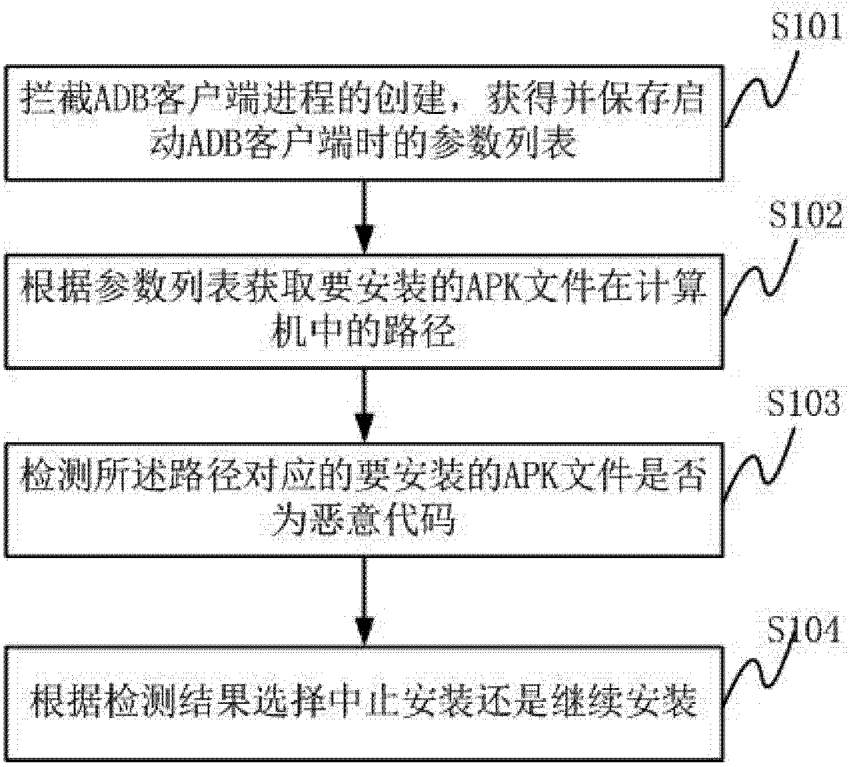

S101
拦截ADB客户端进程的创建，获得并保存启动ADB客户端时的参数列表
S102
根据参数列表获取要安装的APK文件在计算机中的路径
S103
检测所述路径对应的要安装的APK文件是否为恶意代码
S104
根据检测结果选择中止安装还是继续安装

图 1

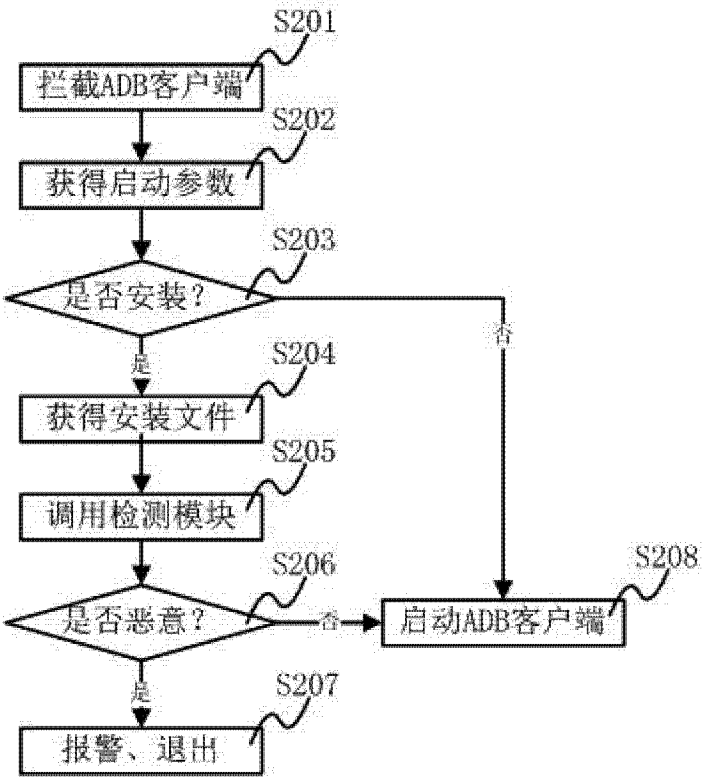

S201
拦截ADB客户端
S202
获得启动参数
S203
是否安装？
否
是
S204
获得安装文件
S205
调用检测模块
S206
是否恶意？
否
是
S207
报警、退出
S208
启动ADB客户端

图 2

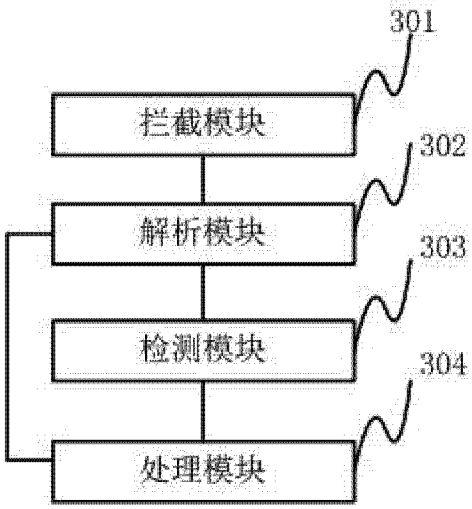

301
拦截模块
302
解析模块
303
检测模块
304
处理模块

图 3