

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2015 年 5 月 21 日 (21.05.2015)



(10) 国际公布号
WO 2015/070653 A1

- (51) 国际专利分类号:
G06F 21/56 (2013.01)
- (21) 国际申请号: PCT/CN2014/085187
- (22) 国际申请日: 2014 年 8 月 26 日 (26.08.2014)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:
201310575193.9 2013 年 11 月 15 日 (15.11.2013) CN
- (71) 申请人: 北京奇虎科技有限公司 (BEIJING QIHOO TECHNOLOGY COMPANY LIMITED) [CN/CN]; 中国北京市西城区新街口外大街 28 号 D 座 112 室 (德胜园区), Beijing 100088 (CN)。奇智软件 (北京) 有限公司 (QIZHI SOFTWARE (BEIJING) COMPANY LIMITED) [CN/CN]; 中国北京市朝阳区酒仙桥路 6 号院 2 号楼 B 座 2 层、3 层 301-306 室, Beijing 100015 (CN)。
- (72) 发明人: 胡中 (HU, Zhong); 中国北京市朝阳区酒仙桥路 6 号院 2 号楼, Beijing 100015 (CN)。王鑫 (WANG, Xin); 中国北京市朝阳区酒仙桥路 6 号院 2 号楼, Beijing 100015 (CN)。
- (74) 代理人: 北京华沛德权律师事务所 (BEIJING BRIGHT & RIGHT LAW FIRM); 中国北京市朝阳区朝外大街乙 12 号昆泰国际大厦 1008 室, Beijing 100020 (CN)。
- (81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。
- (84) 指定国 (除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ,

[见续页]

(54) Title: VIRUS PROTECTION METHOD AND DEVICE

(54) 发明名称: 病毒防御方法和装置

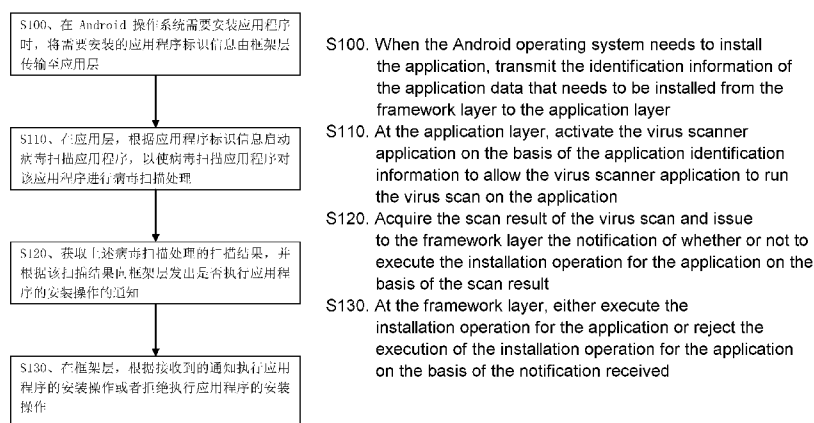


图 1 / Fig. 1

(57) Abstract: A virus protection method and device. The virus protection method comprises: when an Android operating system needs to install an application, transmitting identification information of the application that needs to be installed from a framework layer to an application layer; at the application layer, activating a virus scanner application on the basis of the application identification information to allow the virus scanner application to run a virus scan on the application; acquiring a scan result of the virus scan, issuing a notification to the framework layer of whether or not to execute an installation operation for the application on the basis of the scan result; and, at the framework layer, either executing the installation operation for the application or rejecting to execute the installation operation for the application on the basis of the notification.

(57) 摘要: 一种病毒防御方法和装置; 其中的病毒防御方法包括: 在 Android 操作系统需要安装应用程序时, 将需要安装的应用程序标识信息由框架层传输至应用层; 在应用层, 根据所述应用程序标识信息启动病毒扫描应用程序, 以使病毒扫描应用程序对所述应用程序进行病毒扫描处理; 获取所述病毒扫描处理的扫描结果, 并根据所述扫描结果向框架层发出是否执行所述应用程序的安装操作的通知; 在框架层, 根据所述通知执行所述应用程序的安装操作或者拒绝执行所述应用程序的安装操作。



WO 2015/070653 A1



BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

根据细则 4.17 的声明:

- 关于申请人有权申请并被授予专利(细则 4.17(ii))
- 发明人资格(细则 4.17(iv))

本国际公布:

- 包括国际检索报告(条约第 21 条(3))。

病毒防御方法和装置

技术领域

本发明涉及病毒防御技术，具体涉及一种病毒防御方法和装置。

背景技术

毒防御对于可以进行网络访问的终端设备来说是非常重要的。为了实现病毒防御，终端设备尤其是智能移动设备，通常都设置有安全管理应用程序，该安全管理应用程序所能够提供的的一个主要功能就是病毒查杀。

为了确保终端设备尽可能的不被病毒感染，现有的安全管理应用程序通常会提供应用程序安装的病毒查杀功能；也就是说，在终端设备安装完一应用程序后，安全管理应用程序立即对刚安装的应用程序进行病毒扫描及杀毒处理；一个具体的例子为：在基于Android操作系统的终端设备成功安装了一个应用程序之后，Android操作系统会针对该应用程序的成功安装发出相应的广播，安全管理应用程序在监听到该广播后，启动病毒扫描处理，以检测终端设备刚刚安装完成的应用程序是否携带有病毒；安全管理应用程序可以根据病毒扫描结果向用户提供相应的提示信息，从而安全管理应用程序可以根据用户输入的指示信息执行相应的操作；如安全管理应用程序向用户显示“安装的应用程序携带有病毒，是否卸载该应用程序”的提示信息，安全管理应用程序在接收到用户的卸载指示后，卸载刚安装的应用程序。

发明人在实现本发明过程中发现，上述现有的病毒防御方式并没有尽可能的确保终端设备的安全，具体的说，在Android操作系统中的框架层将应用程序成功安装在终端设备中之后，即使终端设备应用层中的安全管理应用程序扫描出该应用程序携带有病毒，并针对该应用程序执行了卸载处理，然而，由于框架层已经将该应用程序安装在终端设备中，因此该病毒已随着应用程序的成功安装而侵入了终端设备，从而在该应用程序被应用层的安全管理应用程序成功卸载之前，该应用程序携带的病毒很可能已经对终端设备造成了一定的危害。

发明内容

鉴于上述问题，提出了本发明以便提供一种克服上述问题或者至少部分地解决上述问题的病毒防御方法和相应的病毒防御装置。

依据本发明的一个方面，提供了一种病毒防御方法，所述方法包括：在Android操作系统需要安装应用程序时，将需要安装的应用程序标识信息由框

架层传输至应用层；在应用层，根据所述应用程序标识信息启动病毒扫描应用程序，以使病毒扫描应用程序对所述应用程序进行病毒扫描处理；获取所述病毒扫描处理的扫描结果，并根据所述扫描结果向框架层发出是否执行所述应用程序的安装操作的通知；在框架层，根据所述通知执行所述应用程序的安装操作或者拒绝执行所述应用程序的安装操作。

根据本发明的另一方面，提供了一种病毒防御装置，该装置包括：应用程序安装模块，适于在Android操作系统需要安装应用程序时，将需要安装的应用程序标识信息由框架层传输至应用层；启动病毒扫描模块，适于在应用层，根据所述应用程序标识信息启动病毒扫描应用程序，以使病毒扫描应用程序对所述应用程序进行病毒扫描处理；通知模块，适于获取所述病毒扫描处理的扫描结果，并根据所述扫描结果向框架层发出是否执行所述应用程序的安装操作的通知；所述应用程序安装模块还适于，在框架层，根据所述通知执行所述应用程序的安装操作或者拒绝执行所述应用程序的安装操作。

本发明的病毒防御方法和装置通过在框架层执行应用程序的安装操作之前，由框架层通知应用层利用病毒扫描应用程序对应用程序进行病毒扫描处理，这样，在应用程序携带有病毒的情况下，应用层可以通知框架层不再执行该应用程序的安装操作，使携带有病毒的应用程序可以不再被框架层安装在终端设备中，从而本发明实施例避免了携带病毒应用程序被框架层安装在终端设备中而给终端设备带来的潜在危害，提高了终端设备的安全性。

上述说明仅是本发明技术方案的概述，为了能够更清楚了解本发明的技术手段，而可依照说明书的内容予以实施，并且为了让本发明的上述和其它目的、特征和优点能够更明显易懂，以下特举本发明的具体实施方式。

附图说明

通过阅读下文优选实施方式的详细描述，各种其他的优点和益处对于本领域普通技术人员将变得清楚明了。说明书附图仅用于示出优选实施方式的目的，而并不认为是对本发明的限制。而且在整个附图中，用相同的参考符号表示相同的部件。在附图中：

图1示出了根据本发明实施例一的病毒防御方法流程图；

图2示出了根据本发明实施例二的病毒防御方法流程图；

图3示出了根据本发明实施例三的病毒防御装置示意图；

图4示出了用于执行本发明的方法的通信设备的框图；以及

图5示出了用于保持或者携带实现根据本发明的方法的程序代码的存储单元示意图。

具体实施方式

下面将参照附图更详细地描述本公开的示例性实施例。虽然附图中显示了本公开的示例性实施例，然而应当理解，可以以各种形式实现本公开而不应被这里阐述的实施例所限制。相反，提供这些实施例是为了能够更透彻地理解本公开，并且能够将本公开的范围完整的传达给本领域的技术人员。

实施例一、病毒防御方法。下面结合图1对本实施例的方法进行说明。

图1中，S100、在Android操作系统需要安装应用程序时，将需要安装的应用程序标识信息由框架层传输至应用层。

具体的，本实施例可以在应用层中设置应用安装监听接口函数，在框架层中提供一服务（如安全服务），并为该安全服务设置安装前包检测接口函数；这样，需要安装的应用程序标识信息可以通过框架层的安全服务的安装前包检测接口函数传输给应用层的应用安装监听接口函数。

由于应用程序在安装时，Android操作系统会启动包管理服务，而包管理服务中的应用安装接口函数会被调用，因此，本实施例可以通过对现有的应用安装接口函数中的具体实现流程进行改造，以实现在执行应用程序的安装操作之前先对该应用程序进行病毒扫描，以避免终端设备安装携带有病毒的应用程序的目的。

本实施例可以在应用安装接口函数的开始位置处设置安装前包检测接口函数，这样，在应用安装接口函数被调用执行时，应用安装接口函数中的该安装前包检测接口函数中所包含的具体实现流程首先被执行，而应用安装接口函数中的应用程序的正常安装操作被中断。安装前包检测接口函数的一个主要作用是保存应用程序安装的现场数据（如保存操作系统传入至包管理服务的应用安装接口函数的各个参数信息），并将需要安装的应用程序的标识信息回传给应用层，以便于应用层对需要安装的应用程序进行病毒扫描处理。

本实施例可以通过回调接口的方式将需要安装的应用程序标识信息由框架层传输到应用层；具体的，应用层在框架层中注册应用安装监听器，从而应用层可以通过该应用安装侦听器从框架层获得需要安装的应用程序的标识信息。

一个具体的例子，应用层的安全管理应用程序（如360安全卫士等）可以通过用于设置监听器的类函数在框架层注册应用安装监听器，该应用安装监听器对应的应用安装监听接口函数会被框架层的安装前包检测接口函数所调用（该应用安装监听接口函数为回调函数），从而需要安装的应用程序标识由框架层传输至应用层。

本实施例可以将应用安装监听接口函数的名称设置于框架层的应用安装监听器集合中,这样,安装前包检测接口函数可以从应用安装监听器集合中获取到一个或者多个回调函数的名称(其中至少包含有应用安装监听接口函数的名称),安装前包检测接口函数通过获得的名称调用各回调函数,从而将需要

5 安装的应用程序标识信息传输给各回调函数。

上述应用程序标识信息是指可以唯一定位一应用程序的信息,该应用程序标识信息可以具体为应用程序的URI (Uniform Resource Identifier, 通用资源标识)等。

10 S110、在应用层,根据应用程序标识信息启动病毒扫描应用,以使病毒扫描应用程序对该需要安装的应用程序进行病毒扫描处理。

具体的,在应用层,应用安装监听接口函数可以将应用程序标识信息提供给终端设备中的病毒扫描应用程序(该病毒扫描应用程序也可以称为安全管理应用程序,如现有的杀毒应用程序等),并触发病毒扫描应用程序对该应用程序标识信息对应的应用程序进行病毒扫描。

15 S120、获取上述病毒扫描处理的扫描结果,并根据该扫描结果向框架层发出是否执行应用程序的安装操作的通知。

具体的,在病毒扫描应用程序(可以包含在安全管理应用程序中)对需要安装的应用程序病毒扫描完毕,产生并输出病毒扫描结果后,本实施例可以利用应用层中的应用安装恢复类函数或者中止应用安装类函数向框架层发出相

20 应的通知,以使框架层中的应用安装接口函数获知是否应执行该应用程序的安装操作。

一个具体的例子,安全管理应用程序在病毒扫描操作执行完毕后,调用应用层中的应用安装恢复类函数或者中止应用安装类函数,并根据病毒扫描结果设置应用层中的应用安装恢复类函数或者中止应用安装类函数的输入参数;应用层中的应用安装恢复类函数或者中止应用安装类函数再调用框架层的服务

25 提供的应用安装恢复类函数或者中止应用安装类函数,以通知框架层是否执行应用程序的安装操作。

在本实施例中,安全管理应用程序可以在病毒扫描应用程序成功执行了病毒扫描操作后,自动根据病毒扫描结果直接向框架层发出是否执行应用程序

30 安装操作的通知;一个具体的例子,在安全管理应用程序未发现需要安装的应用程序携带有病毒的情况下,安全管理应用程序直接调用应用层中的应用安装恢复类函数或者中止应用安装类函数,并通过应用层中的应用安装恢复类函数或中止应用安装类函数调用框架层的服务提供的应用安装恢复类函数或者中止应用安装类函数,来通知框架层执行该应用程序安装操作;另一个具体的例

子,在安全管理应用程序发现需要安装的应用程序携带有病毒的情况下,安全管理应用程序直接调用应用层中的应用安装恢复类函数或者中止应用安装类函数,并通过应用层中的应用安装恢复类函数或者中止应用安装类函数调用框架层的服务提供的应用安装恢复类函数或者中止应用安装类函数,来通知框架层不再执行该应用程序安装操作。

在本实施例中,安全管理应用程序也可以在病毒扫描应用程序成功执行了病毒扫描操作后,根据病毒扫描结果以及用户的指示信息向框架层发出是否执行应用程序的安装操作的通知;一个具体的例子,在安全管理应用程序发现应用程序携带有病毒的情况下,安全管理应用程序提示用户“扫描发现该应用程序携带有病毒,是否仍要安装该应用程序”,安全管理应用程序根据用户输入的指示信息来调用应用层中的应用安装恢复类函数或者中止应用安装类函数,并通过应用层中的应用安装恢复类函数或中止应用安装类函数调用框架层的服务提供的应用安装恢复类函数或者中止应用安装类函数,来通知框架层是否执行该应用程序安装操作。

S130、在框架层,根据接收到的通知执行应用程序的安装操作或者拒绝执行应用程序的安装操作。

具体的,框架层的服务提供的应用安装恢复类函数或者中止应用安装类函数在根据应用层中的应用安装恢复类函数或者中止应用安装类函数的输入参数确定出需要执行该应用程序的安装操作的情况下,应先获取前述保存的现场数据,将该现场数据提供给应用安装接口函数,并触发应用安装接口函数执行应用程序的安装操作,从而使应用程序成功安装于终端设备中;框架层的服务提供的应用安装恢复类函数或者中止应用安装类函数在根据应用层中的应用安装恢复类函数或者中止应用安装类函数的输入参数确定出不需要执行该应用程序的安装操作的情况下,可直接丢弃前述保存的现场数据,并不再触发应用安装接口函数执行应用程序的安装操作,从而应用程序不会被安装在终端设备中。

一个具体的例子,框架层中只存在一个应用安装接口函数,该应用安装接口函数的输入参数中包含有恢复安装操作的标志位,在该标志位表示恢复安装操作的情况下,应用安装接口函数在被调用时,不会再执行其中的安装前包检测接口函数,而在该标志位表示非恢复安装操作的情况下,应用安装接口函数在被调用时,应执行其中的安装前包检测接口函数。这样,框架层的服务提供的应用安装恢复类函数或者中止应用安装类函数在决定需要执行该应用程序的安装操作的情况下,调用该应用安装接口函数,将上述保存的现场数据提供给该应用安装接口函数,并将恢复安装操作的标志位设置为表征恢复安装操作

的取值,从而应用安装接口函数不再调用安装前包检测接口函数,而直接执行应用程序的安装操作,应用程序被安装在终端设备中;框架层的服务提供的应用安装恢复类函数或者中止应用安装类函数在决定不需要执行该应用程序的安装操作的情况下,直接丢弃前述保存的现场数据即可。

5 另一个具体的例子,框架层中同时存在有两个应用安装接口函数,这两个应用安装接口函数在函数名称上可以有所区别(如应用安装接口函数A和应用安装接口函数B),其中一个是经本发明改造的应用安装接口函数(即包含有安装前包检测接口函数),而另一个是未经本发明改造的应用安装接口函数(即现有的应用安装接口函数),该未经本发明改造的应用安装接口函数并不能直接
10 被外部调用(如不能被包管理服务直接调用),而只能被框架层的服务提供的应用安装恢复类函数或者中止应用安装类函数所调用。这样,框架层的服务提供的应用安装恢复类函数或中止应用安装类函数在决定需要执行该应用程序的安装操作的情况下,调用未经本发明改造的应用安装接口函数,并将上述保存的现场数据提供给未经本发明改造的应用安装接口函数,从而应用程序的
15 安装操作被执行,应用程序被安装在终端设备中;框架层的服务提供的应用安装恢复类函数或者中止应用安装类函数在决定不需要执行该应用程序的安装操作的情况下,直接丢弃前述保存的现场数据即可。

需要特别说明的是,本实施例在病毒扫描结果为应用程序携带有病毒的情况下,应用层(如安全管理应用程序)可以直接删除该应用程序;也可以先由
20 应用层(如安全管理应用程序)提示用户是否需要删除该应用程序,如果用户指示删除该应用程序,则应用层(如安全管理应用程序)再删除该应用程序,否则,应用层(如安全管理应用程序)不会执行删除该应用程序的操作。

由上述实施例的描述可知,本实施例可以在Android操作系统的框架(Framework)层中的应用安装接口函数中加入Hook(钩子),这样,中断了
25 原有的应用程序安装过程,应用安装接口函数先回调安全管理应用程序的接口函数,以通知应用层安装应用程序这一事件,然后,应用层可以根据病毒扫描结果,决定是否继续框架层的应用程序安装流程;从而可以禁止携带有病毒的应用程序被安装在终端设备中。

实施例二、病毒防御方法。下面结合图2以Android操作系统4.2.2为例,
30 对本实施例的方法进行说明。

在应用(App)层,利用Sdk(软件开发工具包)为安全管理应用程序(如360安全卫士)定义如表1和表2所示的类函数以及接口函数,以便于应用层与Framework(框架)层进行信息交互:

表1

接口	
QihooPreInstallListener	
方法（也即函数）	
Void 空类型	onPreInstall(Uri Uri) 是应用安装监听器对应的接口函数，在应用程序被安装前，Framework 层调用此回调函数，以通知应用层需要安装的应用程序的 Uri。

表 2

类	
QihooAppManager	
方法（也即函数）	
Boolean 布尔型	setPreInstallListener(Uri uri, QihooPreInstallListener listener) 用于向操作系统的框架层中注册应用安装监听器，这样，在应用被安装时，该应用安装监听器对应的接口函数会被 Framework 层回调。
Boolean 布尔型	resumeOrAbortInstall(String packageName, boolean abort) 用于通知框架层是否继续安装 App（应用程序）；它会调用框架层的 SecurityService 的 resumeOrAbortInstall() 接口函数。

在 Framework 层，新增一个服务，该服务的名称可以为 SecurityService（网络安全服务），该服务提供如下接口函数：

5 //在应用程序实际安装开始之前，调用表1所示的接口函数，以通知应用层的安全管理应用程序有个应用程序要被安装到终端设备中：

```
public boolean checkPkgPreInstall(Uri packageURI,
                                   IPackageInstallObserver observer,
                                   int flags,
10       String installerPackageName,
                                   VerificationParams verificationParams,
                                   ContainerEncryptionParams encryptionParams)
```

上述 packageURI、observer、flags、installerPackageName、VerificationParams 以及 encryptionParams 为应用程序的正常安装操作被中
15 断时，应保存的现场数据，且 packageURI、observer、flags、installerPackageName、VerificationParams 以及 encryptionParams 通常也是操作系统传入至包管理服务的信息。

//通知包管理服务 (PackageManagerService) 是否要安装指定的应用程序, 即是否触发实际的安装操作:

//uri: 表示应用程序的位置信息

//abort: true表示放弃安装该应用程序, false表示安装该应用程序。

5 public void resumeOrAbortInstall(Uri uri, boolean abort)

需要特别说明的是, 上述checkPkgPreInstall() 是本发明实施例在应用安装接口函数 (即installPackageWithVerificationAndEncryption() 接口函数) 的开始位置处新增的内容。也就是说, 本发明对现有的应用安装接口函数进行了改造, 下面以一个具体的例子对该改造进行详细说明。

10 设定现有的包管理服务中的应用安装接口函数为:

```
public void installPackageWithVerificationAndEncryption
    (Uri packageURI,
     IPackageInstallObserver observer,
     int flags,
15     String installerPackageName,
     VerificationParams verificationParams,
     ContainerEncryptionParams encryptionParams)
```

在本发明实施例中, 对上述现有的包管理服务中的应用安装接口函数进行了如下来两方面改造:

20 改造1、直接将上述现有的包管理服务中的应用安装接口函数重命名为installPackageWithVerificationAndEncryptionOrigin(), 即在保持函数中的内容没有发生任何变化的情况下, 在原应用安装接口函数的名称后面增加了Origin字样, 以表示该应用安装接口函数是原来的应用安装接口函数。

改造2、新增一个应用安装接口函数, 该新增的应用安装接口函数的函数
25 名字以及参数都和原应用安装接口函数完全相同, 只是函数中的内容发生了变化, 请参见如下描述:

```
public void installPackageWithVerificationAndEncryption
    (Uri packageURI,
     IPackageInstallObserver observer,
30     int flags, String installerPackageName,
     VerificationParams verificationParams,
     ContainerEncryptionParams encryptionParams)
```

// checkPkgPreInstall 主要用于检查一下是否有应用注册了应用安装监

听器等;

//如果没有应用注册应用安装监听器,则返回 false; 否则,回调应用安装监听器对应的接口函数 onPreInstall(), 以通知应用层对需要安装的应用程序进行病毒扫描。

```
5      boolean holdInstall = QSSecurityProxy.checkPkgPreInstall
        (packageURI, observer,
          flags, installerPackageName,
            verificationParams, encryptionParams);
      //如果 holdInstall 为 true, 表示应挂起当前应用程序的安装, 不执行
10  应用程序的安装操作。后续是否执行应用程序的安装应等待应用层调用框架层的
      resumeOrAbortInstall() 来决定。
      //如果 holdInstall 为 false, 表示说明发生了前述的情况, 也就没有必要中断安装了, 直接调用原安装接口。
      if (holdInstall == true) {
15      Slog.i(TAG, "Suspend install procedure for " +
packageURI.getPath());
      }else {
        installPackageWithVerificationAndEncryptionOrigin(packageURI,
          observer, flags, installerPackageName, verificationParams,
20      encryptionParams);-
      }
      }
      另外, 框架层中的 resumeOrAbortInstall() 的部分代码如下所示:
      public void resumeOrAbortInstall(Uri uri, boolean abort)
25      { //参数 abort 表示是否终止应用程序的安装
        .....
        .....
        try {
          if (abort == true) {//需要终止应用程序的安装, 直接回调通知操作系
30      统应用程序安装结果 (失败)
            installParams.observer.packageInstalled(null,
              PackageManager.INSTALL_FAILED_INTERNAL_ERROR);
          return;
        } else {//需要继续安装应用程序, 调用修改了函数名称的应用安装接口
```

函数

```

IPackageManager pm = QSServiceUtil.getPm();
pm.installPackageWithVerificationAndEncryptionOrigin(installParams.
packageURI, installParams.observer, installParams.flags, -
5  installParams.installerPackageName,
installParams.verificationParams,
installParams.encryptionParams);
}
} catch (RemoteException re) {
10  Log.e(TAG, re.getMessage(), re);
}
}
}

```

图 2 中, 首先, 安全管理应用程序可以调用 QihooAppManager 中的类函数

15 setPreInstallListener() 来在框架层中注册应用安装监听器; 当需要安装应用程序时, 操作系统调用包管理服务提供的 installPackageWithVerificationAndEncryption() 接口函数, 该接口函数的开始位置处设置有安全服务提供的 checkPkgPreInstall() 接口函数, checkPkgPreInstall() 接口函数在保存现场数据后, 根据上述注册的应用安装

20 监听器回调 onPreInstall(Uri Uri), 这样, 框架层并没有执行实际的安装操作, 而是先通过前述的应用安装监听器通知安全管理应用程序这一安装事件, 从而安全管理应用程序可以根据 onPreInstall(Uri Uri) 的调用马上对需要安装的应用程序进行病毒扫描处理, 然后, 安全管理应用程序调用应用层的类函数 resumeOrAbortInstall(), 该类函数中的第二个参数的具体取值是安全管理

25 应用程序根据病毒扫描结果来确定的, 该类函数 resumeOrAbortInstall() 通过调用框架层的网络安全服务提供的 resumeOrAbortInstall() 来通知框架层是否执行应用程序的安装操作。在框架层的网络安全服务提供的 resumeOrAbortInstall() 判断出需要执行应用程序的安装操作的情况下, 调用未经本发明改造的 installPackageWithVerificationAndEncryptionOrigin()

30 接口函数, 并将上述保存下来的现场数据提供给 installPackageWithVerificationAndEncryptionOrigin() 接口函数, installPackageWithVerificationAndEncryptionOrigin() 接口函数执行应用程序的安装操作, 如执行图 2 中示出的 sendMessage(INIT-COPY) 等操作; 在框架层的网络安全服务提供的 resumeOrAbortInstall() 判断出不需要执行

应用程序的安装操作的情况下，框架层的网络安全服务提供的 resumeOrAbortInstall() 可以直接将上述保存下来的现场数据丢弃，以结束应用程序的安装操作。

图 2 中最右侧的虚线表示，现有的应用安装接口函数在被调用后会直接执行应用程序的安装操作，如执行图 2 中示出的 sendMessage (INIT_COPY) 等操作，本实施例完全改变了现有的应用程序的安装操作。

由上述描述可知，本发明的实施例对 Android 操作系统的 Framework (框架) 层中的应用安装接口函数进行了修改，在应用安装接口函数的现有安装流程中插入了 hook (钩子)，这样，在应用安装接口函数执行应用程序的安装操作之前，会先由病毒扫描应用对需要安装的应用程序进行病毒扫描，从而本发明实施例可以针对携带有病毒的应用程序不再执行安装操作，使携带有病毒的应用程序可以不再被安装在终端设备中，最终本发明实施例避免了携带病毒应用程序被安装在终端设备中而给终端设备带来的潜在危害。

实施例三、病毒防御装置。下面结合图 3 对该装置进行详细说明。

图 2 中示出的装置可以设置于计算机、智能移动电话或者平板电脑等智能电子设备中；且该装置主要包括：应用程序安装模块 300、启动病毒扫描模块 310、通知模块 320、监听器注册模块 330 和应用程序清除模块 340；其中，应用程序安装模块 300 与监听器注册模块 330 连接，启动病毒扫描模块 310 与应用程序安装模块 300 和通知模块 320 分别连接，且通知模块 320 还与应用程序清除模块 340 连接。

应用程序安装模块 300 主要适于在 Android 操作系统需要安装应用程序时，将需要安装的应用程序标识信息由框架层传输至应用层。

具体的，本实施例可以在应用层中设置应用安装监听接口函数，在框架层中提供一服务（如安全服务），并为该安全服务设置安装前包检测接口函数，该安装前包检测接口函数设置于应用程序安装模块 300 中；这样，应用程序安装模块 300 可以将需要安装的应用程序标识信息通过框架层的安全服务的安装前包检测接口函数传输给应用层的应用安装监听接口函数。

由于应用程序在安装时，Android 操作系统会启动包管理服务，而包管理服务中的应用安装接口函数（该应用安装接口函数属于应用程序安装模块 300）会被调用，因此，本实施例可以通过对现有的应用安装接口函数中的具体实现流程进行改造，以实现在执行应用程序的安装操作之前先对该应用程序进行病毒扫描，以避免终端设备安装携带有病毒的应用程序的目的。

本实施例可以在应用安装接口函数的开始位置处设置安装前包检测接口函数，这样，在应用安装接口函数被调用执行时，应用安装接口函数中的该安

装前包检测接口函数中所包含的具体实现流程首先被执行,而应用安装接口函数中的应用程序的正常安装操作被中断。安装前包检测接口函数的一个主要作用是保存应用程序安装的现场数据(如保存操作系统传入至包管理服务的应用安装接口函数的各个参数信息),并将需要安装的应用程序的标识信息回传给应用层,以便于应用层对需要安装的应用程序进行病毒扫描处理。

本实施例可以通过回调接口的方式将需要安装的应用程序标识信息由框架层传输到应用层;具体的,应用层中的监听器注册模块 330 在框架层中注册应用安装监听器,从而应用层中的启动病毒扫描模块 310 可以通过该应用安装侦听器从框架层获得需要安装的应用程序的标识信息。

一个具体的例子,应用层的安全管理应用程序(如 360 安全卫士等)中的监听器注册模块 330 可以通过用于设置监听器的类函数在框架层注册应用安装监听器,该应用安装监听器对应的应用安装监听接口函数会被框架层的应用程序安装模块 300 中的安装前包检测接口函数所调用,从而需要安装的应用程序标识由框架层传输至应用层。

监听器注册模块 330 可以将应用安装监听接口函数的名称设置于框架层的应用安装监听器集合中,这样,应用程序安装模块 300 中的安装前包检测接口函数可以从应用安装监听器集合中获取到一个或者多个回调函数的名称(其中至少包含有应用安装监听接口函数的名称),安装前包检测接口函数通过获得的名称调用各回调函数,从而应用程序安装模块 300 将需要安装的应用程序标识信息传输给各回调函数。

上述应用程序标识信息是指可以唯一定位一应用程序的信息,该应用程序标识信息可以具体为应用程序的 URI 等。

启动病毒扫描模块 310 主要适于在应用层,根据应用程序标识信息启动病毒扫描应用程序,以使病毒扫描应用程序对需要安装的应用程序进行病毒扫描处理。

具体的,在应用层,启动病毒扫描模块 310 中的应用安装监听接口函数可以将应用程序标识信息提供给终端设备中的病毒扫描应用程序(该病毒扫描应用程序也可以称为安全管理应用程序,如现有的杀毒应用程序等),并触发病毒扫描应用程序对该应用程序标识信息对应的应用程序进行病毒扫描。

通知模块 320 主要适于获取病毒扫描处理的扫描结果,并根据扫描结果向框架层发出是否执行应用程序的安装操作的通知。

具体的,在病毒扫描应用程序(也即安全管理应用程序)对需要安装的应用程序病毒扫描完毕,产生并输出病毒扫描结果后,通知模块 320 可以利用应用层中的应用安装恢复类函数或者中止应用安装类函数向框架层发出相应的

通知,以使框架层中的应用安装接口函数获知是否应执行该应用程序的安装操作。

一个具体的例子,通知模块 320 在病毒扫描应用程序执行了病毒扫描操作后,调用应用层中的应用安装恢复类函数或者中止应用安装类函数,并根据病毒扫描结果设置应用层中的应用安装恢复类函数或者中止应用安装类函数的输入参数;应用层中的应用安装恢复类函数或者中止应用安装类函数再调用框架层的服务提供的应用安装恢复类函数或者中止应用安装类函数,以通知框架层是否执行应用程序的安装操作。

在本实施例中,通知模块 320 在病毒扫描应用程序成功执行了病毒扫描操作后,自动根据病毒扫描结果直接向框架层发出是否执行应用程序的安装操作的通知;一个具体的例子,在病毒扫描应用程序未发现需要安装的应用程序携带有病毒的情况下,通知模块 320 直接调用应用层中的应用安装恢复类函数或者中止应用安装类函数,并通过应用层中的应用安装恢复类函数或者中止应用安装类函数调用框架层的服务提供的应用安装恢复类函数或者中止应用安装类函数,来通知框架层执行该应用程序安装操作;另一个具体的例子,在病毒扫描应用程序发现需要安装的应用程序携带有病毒的情况下,病毒扫描应用程序直接调用应用层中的应用安装恢复类函数或者中止应用安装类函数,并通过应用层中的应用安装恢复类函数或者中止应用安装类函数调用框架层的服务提供的应用安装恢复类函数或者中止应用安装类函数,来通知框架层不再执行该应用程序安装操作。

在本实施例中,通知模块 320 也可以在病毒扫描应用程序成功执行了病毒扫描操作后,根据病毒扫描结果以及用户的指示信息向框架层发出是否执行应用程序的安装操作的通知;一个具体的例子,在病毒扫描应用程序发现应用程序携带有病毒的情况下,通知模块 320 提示用户“扫描发现该应用程序携带有病毒,是否仍要安装该应用程序”,通知模块 320 根据用户输入的指示信息调用应用层中的应用安装恢复类函数或者中止应用安装类函数,并通过应用层中的应用安装恢复类函数或者中止应用安装类函数调用框架层的服务提供的应用安装恢复类函数或者中止应用安装类函数,来通知框架层是否执行该应用程序安装操作。

应用程序安装模块 300 还主要适于根据通知模块 320 的通知执行应用程序的安装操作或者拒绝执行应用程序的安装操作。

具体的,框架层的服务提供的应用安装恢复类函数或者中止应用安装类函数(属于应用程序安装模块 300)在根据应用层中的应用安装恢复类函数或者中止应用安装类函数的输入参数确定出需要执行该应用程序的安装操作的情

况下,应先获取前述保存的现场数据,将该现场数据提供给应用安装接口函数,并触发应用安装接口函数执行应用程序的安装操作,从而使应用程序成功安装于终端设备中;框架层的服务提供的应用安装恢复类函数或者中止应用安装类函数在根据应用层中的应用安装恢复类函数或者中止应用安装类函数的输入参数确定出不需要执行该应用程序的安装操作的情况下,可直接丢弃前述保存的现场数据,并不再触发应用安装接口函数执行应用程序的安装操作,从而应用程序不会被安装在终端设备中。

一个具体的例子,应用程序安装模块 300 中只存在一个应用安装接口函数,该应用安装接口函数的输入参数中包含有恢复安装操作的标志位,在该标志位表示恢复安装操作的情况下,应用安装接口函数在被调用时,不会再执行其中的安装前包检测接口函数,而在该标志位表示非恢复安装操作的情况下,应用安装接口函数在被调用时,应执行其中的安装前包检测接口函数。这样,应用程序安装模块 300 中的应用安装恢复类函数或者中止应用安装类函数在决定需要执行该应用程序的安装操作的情况下,调用该应用安装接口函数,将上述保存的现场数据提供给该应用安装接口函数,并将恢复安装操作的标志位设置为表征恢复安装操作的取值,从而应用安装接口函数不再调用安装前包检测接口函数,而直接执行应用程序的安装操作,应用程序被安装在终端设备中;框架层的服务提供的应用安装恢复类函数或者中止应用安装类函数在决定不需要执行该应用程序的安装操作的情况下,直接丢弃前述保存的现场数据即可。

另一个具体的例子,应用程序安装模块 300 中同时存在有两个应用安装接口函数,这两个应用安装接口函数在函数名称上可以有所区别(如应用安装接口函数 A 和应用安装接口函数 B),其中一个是经本发明改造的应用安装接口函数(即包含有安装前包检测接口函数),而另一个是未经本发明改造的应用安装接口函数(即现有的应用安装接口函数),该未经本发明改造的应用安装接口函数并不能直接被外部调用(如不能被包管理服务直接调用),而只能被框架层的服务提供的应用安装恢复类函数或者中止应用安装类函数(属于应用程序安装模块 300)所调用。这样,框架层的服务提供的应用安装恢复类函数或者中止应用安装类函数在决定需要执行该应用程序的安装操作的情况下,调用未经本发明改造的应用安装接口函数,并将上述保存的现场数据提供给未经本发明改造的应用安装接口函数,从而应用程序的安装操作被执行,该应用程序被安装在终端设备中;框架层的服务提供的应用安装恢复类函数或者中止应用安装类函数在决定不需要执行该应用程序的安装操作的情况下,直接丢弃前述保存的现场数据即可。

本实施例在病毒扫描结果为应用程序携带有病毒的情况下,设置于应用层中的应用程序清除模块 340 可以直接删除该应用程序;也可以先由应用程序清除模块 340 提示用户是否需要删除该应用程序,如果用户指示删除该应用程序,则应用程序清除模块 340 再删除该应用程序,否则,应用程序清除模块 340 不会执行删除该应用程序的操作。

在此提供的算法和显示不与任何特定计算机、虚拟系统或者其它设备固有相关。各种通用系统也可以与基于在此的示教一起使用。根据上面的描述,构造这类系统所要求的结构是显而易见的。此外,本发明也不针对任何特定编程语言。应当明白,可以利用各种编程语言实现在此描述的本发明的内容,并且上面对特定语言所做的描述是为了披露本发明的最佳实施方式。

在此处所提供的说明书中,说明了大量具体细节。然而,能够理解,本发明的实施例可以在没有这些具体细节的情况下实践。在一些实例中,并未详细示出公知的方法、结构和技术,以便不模糊对本说明书的理解。

类似地,应当理解,为了精简本公开并帮助理解各个发明方面中的一个或多个,在上面对本发明的示例性实施例的描述中,本发明的各个特征有时被一起分组到单个实施例、图、或者对其的描述中。然而,并不应将该公开的方法解释成反映如下意图:即所要求保护的本发明要求比在每个权利要求中所明确记载的特征更多的特征。更确切地说,如下面的权利要求书所反映的那样,发明方面在于少于前面公开的单个实施例的所有特征。因此,遵循具体实施方式的权利要求书由此明确地并入该具体实施方式,其中每个权利要求本身都作为本发明的单独实施例。

本领域那些技术人员可以理解,可以对实施例中的设备中的模块进行自适应性地改变并且把它们设置在与该实施例不同的一个或多个设备中。可以把实施例中的模块或单元或组件组合成一个模块或单元或组件,以及此外可以把它分成多个子模块或子单元或子组件。除了这样的特征和/或过程或者单元中的至少一些是相互排斥之外,可以采用任何组合对本说明书(包括伴随的权利要求、摘要和附图)中公开的所有特征以及如此公开的任何方法或者设备的所有过程或单元进行组合。除非另外明确陈述,本说明书(包括伴随的权利要求、摘要和附图)中公开的每个特征可以由提供相同、等同或相似目的的替代特征来代替。

此外,本领域的技术人员能够理解,尽管在此所述的一些实施例包括其它实施例中所包括的某些特征而不是其它特征,但是不同实施例的特征的组合意味着处于本发明的范围之内并且形成不同的实施例。例如,在下面的权利要求书中,所要求保护的实施例的任意之一都可以以任意的组合方式来使用。

本发明的各个部件实施例可以以硬件实现,或者以在一个或者多个处理器上运行的软件模块实现,或者以它们的组合实现。本领域的技术人员应当理解,可以在实践中使用微处理器或者数字信号处理器(DSP)来实现根据本发明实施例的电子设备中的病毒防御装置中的一些或者全部部件的一些或者全部功能。本发明还可以实现为用于执行这里所描述的方法的一部分或者全部的设备或者装置程序(例如,计算机程序和计算机程序产品)。这样的实现本发明的程序可以存储在计算机可读介质上,或者可以具有一个或者多个信号的形式。这样的信号可以从因特网网站上下载得到,或者在载体信号上提供,或者以任何其他形式提供。

例如,图4示出了可以实现本发明的病毒防御方法的通信设备。该通信设备传统上包括处理器410和以存储器420形式的计算机程序产品或者计算机可读介质。存储器420可以是诸如闪存、EEPROM(电可擦除可编程只读存储器)、EPROM、硬盘或者ROM之类的电子存储器。存储器420具有用于执行上述方法中的任何方法步骤的程序代码431的存储空间430。例如,用于程序代码的存储空间430可以包括分别用于实现上面的方法中的各种步骤的各个程序代码431。这些程序代码可以从一个或者多个计算机程序产品中读出或者写入到这一个或者多个计算机程序产品中。这些计算机程序产品包括诸如硬盘,紧致盘(CD)、存储卡或者软盘之类的程序代码载体。这样的计算机程序产品通常为如参考图5所述的便携式或者固定存储单元。该存储单元可以具有与图4的通信设备中的存储器420类似布置的存储段或者存储空间等。程序代码可以例如以适当形式进行压缩。通常,存储单元包括用于执行根据本发明的方法步骤的程序431',即可以由例如诸如410之类的处理器读取的代码,这些代码当由通信设备运行时,导致该通信设备执行上面所描述的方法中的各个步骤。

应该注意的是,上述实施例对本发明进行说明而不是对本发明进行限制,并且本领域技术人员在不脱离所附权利要求的范围的情况下可设计出替换实施例。在权利要求中,不应将位于括号之间的任何参考符号构造成对权利要求的限制。单词“包含”不排除存在未列在权利要求中的元件或步骤。位于元件之前的单词“一”或“一个”不排除存在多个这样的元件。本发明可以借助于包括有若干不同元件的硬件以及借助于适当编程的计算机来实现。在列举了若干装置的单元权利要求中,这些装置中的若干个可以是通过同一个硬件项来具体体现。单词第一、第二以及第三等的使用不表示任何顺序。可将这些单词解释为名称。

权 利 要 求

1、一种病毒防御方法，其包括：

在Android操作系统需要安装应用程序时，将需要安装的应用程序标识信息由框架层传输至应用层；

5 在应用层，根据所述应用程序标识信息启动病毒扫描应用程序，以使病毒扫描应用程序对所述应用程序进行病毒扫描处理；

获取所述病毒扫描处理的扫描结果，并根据所述扫描结果向框架层发出是否执行所述应用程序的安装操作的通知；

10 在框架层，根据所述通知执行所述应用程序的安装操作或者拒绝执行所述应用程序的安装操作。

2、如权利要求1所述的方法，其中，所述应用层中设置有应用安装监听接口函数，所述框架层中设置有安全服务的安装前包检测接口函数；

且所述在Android操作系统需要安装应用程序时，将需要安装的应用程序标识信息由框架层传输至应用层包括：

15 在Android操作系统的包管理服务中的应用安装接口函数被调用的情况下，所述应用安装接口函数通过所述安装前包检测接口函数将需要安装的应用程序标识信息传输至应用安装监听接口函数，所述安装前包检测接口函数保存应用程序安装的现场数据，且所述应用安装接口函数停止安装操作。

20 3、如权利要求2所述的方法，其中，所述安装前包检测接口函数设置于所述应用安装接口函数的开始位置处，且所述应用安装接口函数通过所述安装前包检测接口函数将需要安装的应用程序标识信息传输至应用安装监听接口函数包括：

所述应用安装接口函数被调用时，所述应用安装接口函数调用所述安装前包检测接口函数；

25 所述安装前包检测接口函数从应用安装监听器集合中获取所有回调函数的名称，且所述应用安装监听器集合中包含有应用安装监听接口函数的名称；

所述安装前包检测接口函数根据所述回调函数的名称向所有回调函数传输需要安装的应用程序标识信息。

4、如权利要求3所述的方法，其中，所述方法还包括：

30 在框架层的应用安装监听器集合中设置应用安装监听接口函数的名称。

5、如权利要求1所述的方法，其中，所述应用程序标识信息包括：应用程序的URI。

6、如权利要求1所述的方法，其中，所述根据所述扫描结果向框架层发出是否执行所述应用程序的安装操作的通知包括：

根据所述扫描结果利用应用层中的应用安装恢复类函数或者中止应用安装类函数调用框架层的服务提供的应用安装恢复接口函数或者中止应用安装接口函数，以向框架层发出是否执行所述应用程序的安装操作的通知。

7、如权利要求1至6中任一权利要求所述的方法，其中，所述根据所述扫描结果向框架层发出是否执行所述应用程序的安装操作的通知包括：

根据所述病毒扫描结果直接向框架层发出是否执行所述应用程序的安装操作的通知；或者

在病毒扫描结果为应用程序携带有病毒的情况下，将所述病毒扫描结果呈现给用户，并根据用户输入的指示信息向框架层发出是否执行所述应用程序的安装操作的通知。

8、如权利要求1至6中任一权利要求所述的方法，其中，所述方法还包括：

在所述病毒扫描结果为所述应用程序携带有病毒的情况下，直接删除所述应用程序；或者

在所述病毒扫描结果为所述应用程序携带有病毒的情况下，向用户呈现是否删除应用程序的信息，并根据用户输入的指示信息删除/不删除所述应用程序。

9、一种病毒防御装置，其包括：

应用程序安装模块，适于在Android操作系统需要安装应用程序时，将需要安装的应用程序标识信息由框架层传输至应用层；

启动病毒扫描模块，适于在应用层，根据所述应用程序标识信息启动病毒扫描应用程序，以使病毒扫描应用程序对所述应用程序进行病毒扫描处理；

通知模块，适于获取所述病毒扫描处理的扫描结果，并根据所述扫描结果向框架层发出是否执行所述应用程序的安装操作的通知；

所述应用程序安装模块还适于，在框架层，根据所述通知执行所述应用程序的安装操作或者拒绝执行所述应用程序的安装操作。

10、如权利要求9所述的装置，其中，所述应用层中设置有应用安装监听接口函数，所述应用程序安装模块中设置有包管理服务中的应用安装接口函数以及安全服务的安装前包检测接口函数；

且所述在Android操作系统需要安装应用程序时，将需要安装的应用程序标识信息由框架层传输至应用层包括：

在Android操作系统的包管理服务中的应用安装接口函数被调用的情况下，所述应用安装接口函数通过所述安装前包检测接口函数将需要安装的应用程序标识信息传输至应用安装监听接口函数，所述安装前包检测接口函数保存应用程序安装的现场数据，且所述应用安装接口函数停止安装操作。

11、如权利要求10所述的装置，其中，所述安装前包检测接口函数设置于所述应用安装接口函数的开始位置处，且所述应用安装接口函数通过所述安装前包检测接口函数将需要安装的应用程序标识信息传输至应用安装监听接口函数包括：

5 所述应用安装接口函数被调用时，所述应用安装接口函数调用所述安装前包检测接口函数；

 所述安装前包检测接口函数从应用安装监听器集合中获取所有回调函数的名称，且所述应用安装监听器集合中包含有应用安装监听接口函数的名称；

10 所述安装前包检测接口函数根据所述回调函数的名称向所有回调函数传输需要安装的应用程序标识信息。

12、如权利要求11所述的装置，其中，所述装置还包括：

 监听器注册模块，适于在框架层的应用安装监听器集合中设置应用安装监听接口函数的名称。

15 13、如权利要求9所述的装置，其中，所述应用程序标识信息包括：应用程序的URI。

14、如权利要求9所述的装置，其中，所述根据所述扫描结果向框架层发出是否执行所述应用程序的安装操作的通知包括：

20 通知模块根据所述扫描结果利用应用层中的应用安装恢复类函数或者中止应用安装类函数调用框架层的服务提供的应用安装恢复接口函数或者中止应用安装接口函数，以向框架层发出是否执行所述应用程序的安装操作的通知。

15、如权利要求9至14中任一权利要求所述的装置，其中，所述根据所述扫描结果向框架层发出是否执行所述应用程序的安装操作的通知包括：

25 通知模块根据所述病毒扫描结果直接向框架层发出是否执行所述应用程序的安装操作的通知；或者

 通知模块在病毒扫描结果为应用程序携带有病毒的情况下，将所述病毒扫描结果呈现给用户，并根据用户输入的指示信息向框架层发出是否执行所述应用程序的安装操作的通知。

30 16、如权利要求9至14中任一权利要求所述的装置，其中，所述装置还包括：

 应用程序清除模块，适于在所述病毒扫描结果为所述应用程序携带有病毒的情况下，直接删除所述应用程序；或者

 适于在所述病毒扫描结果为所述应用程序携带有病毒的情况下，向用户呈现是否删除应用程序的信息，并根据用户输入的指示信息删除/不删除所述应

用程序。

17、一种计算机程序，包括计算机可读代码，当通信设备运行所述计算机可读代码时，导致权利要求1-8中的任一项权利要求所述的方法被执行。

18、一种计算机可读介质，其中存储了如权利要求17所述的计算机程序。

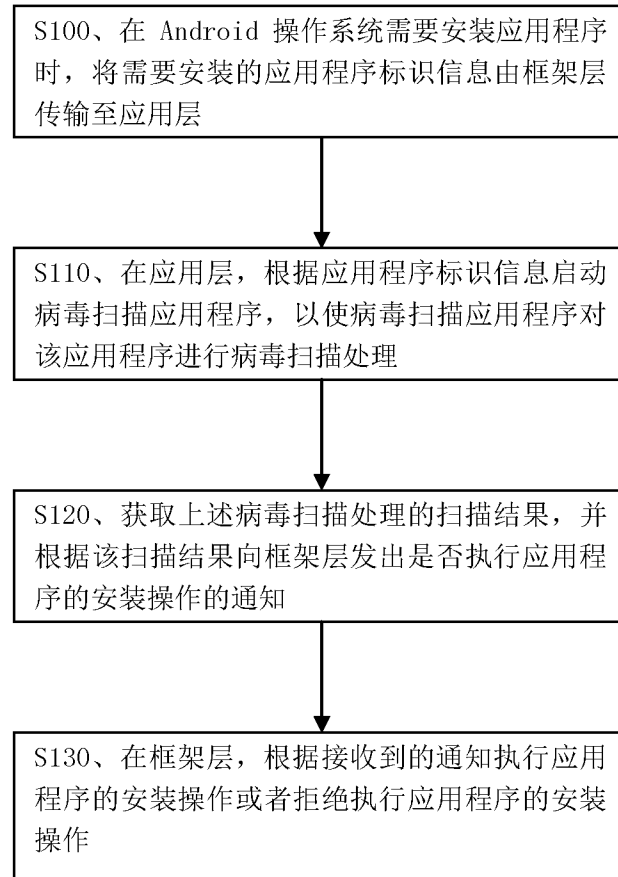


图 1

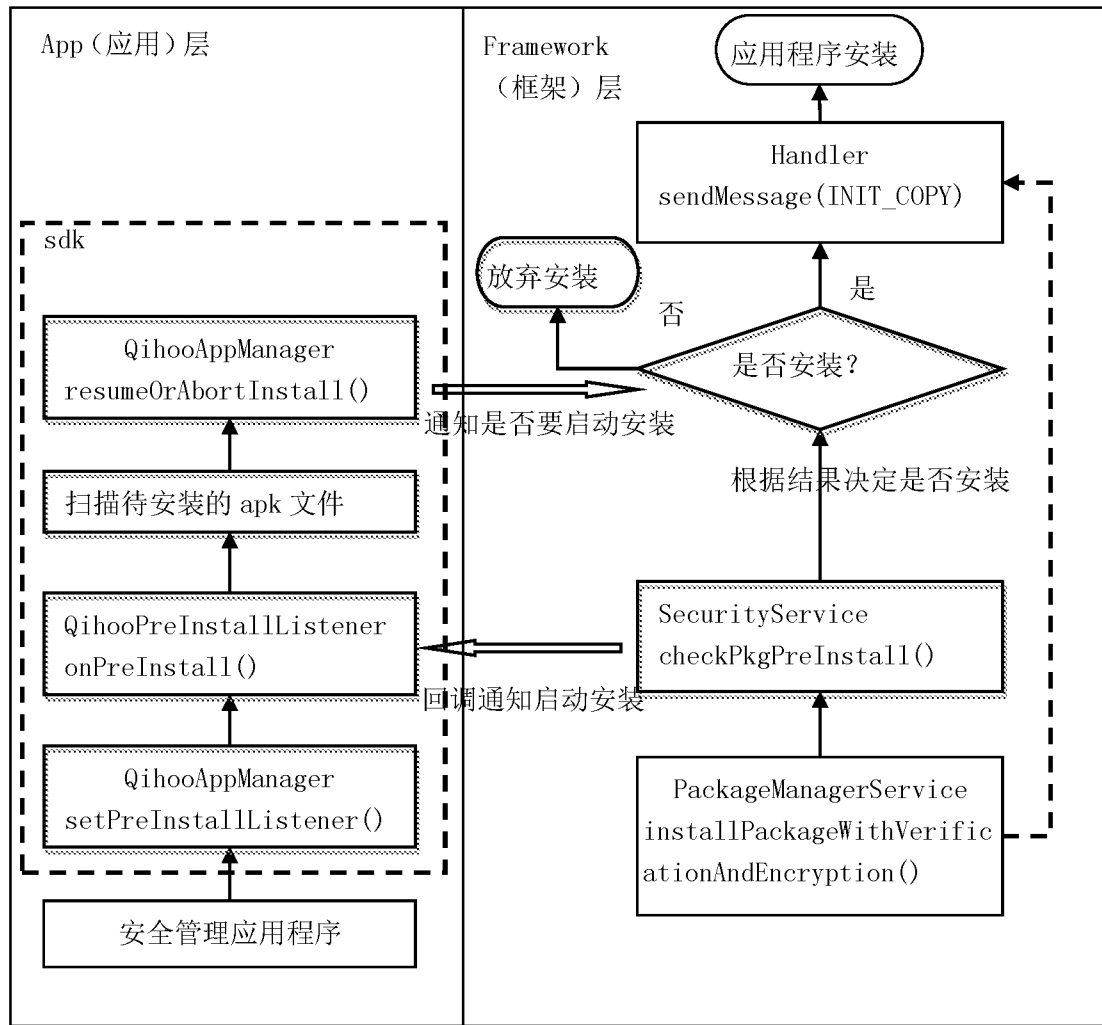


图 2

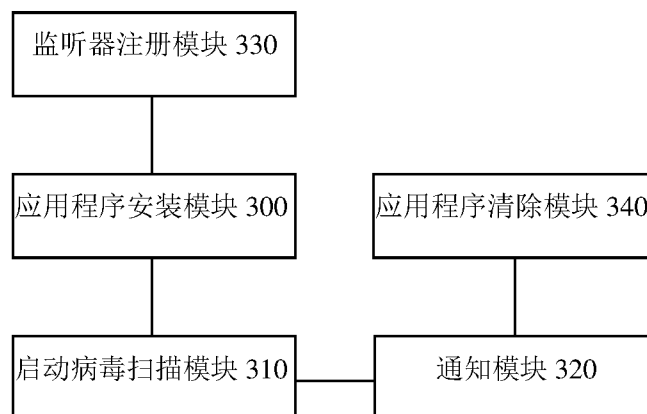


图 3

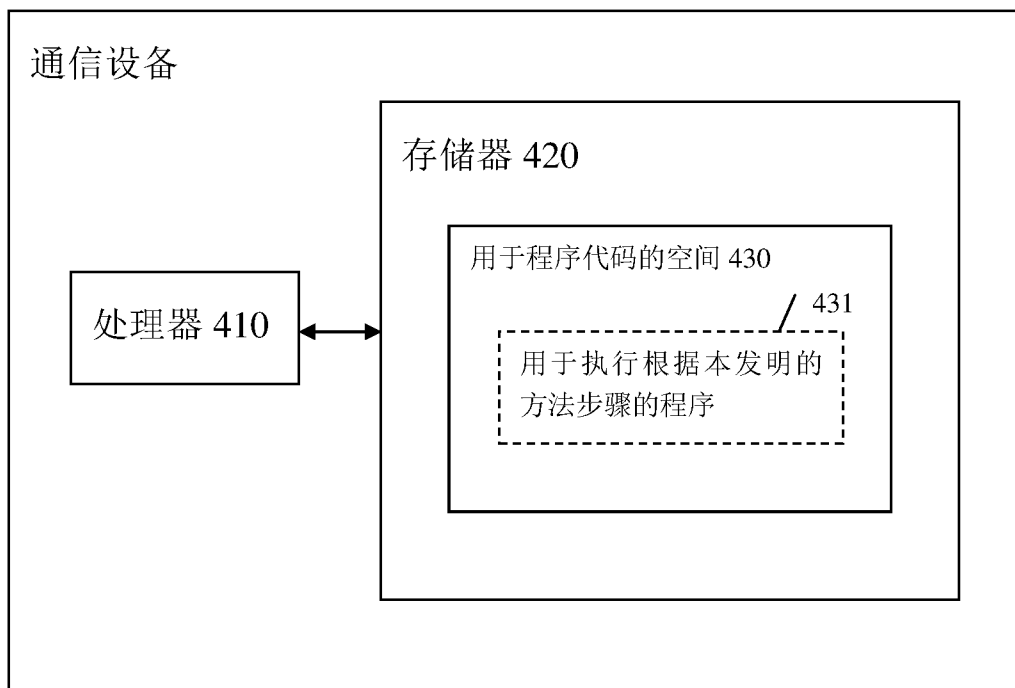


图 4

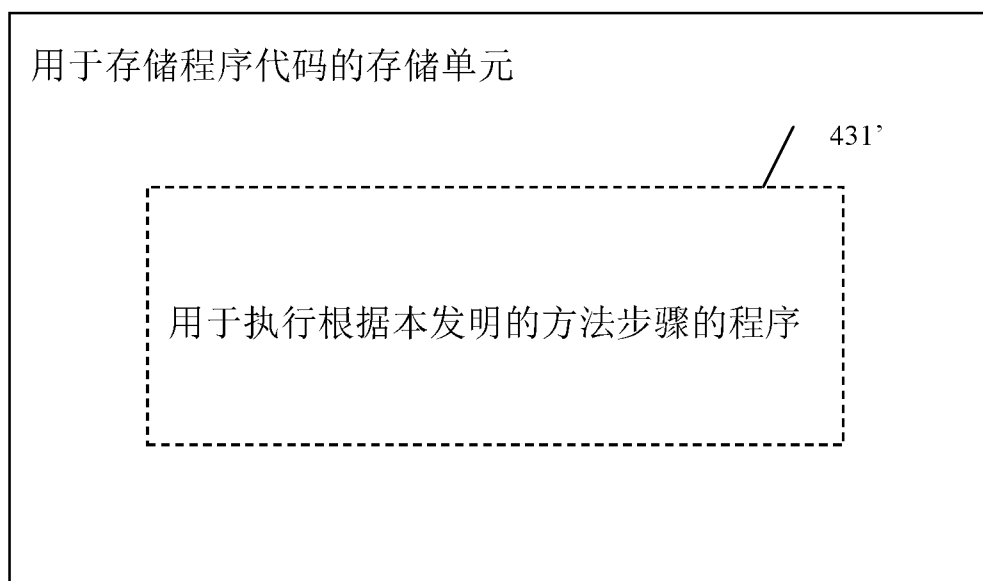


图 5

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2014/085187

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/56 (2013.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT, CNKI, WPI, EPODOC: attack, threaten, application layer, frame layer, malice, frame, android, install+, layer, virus, Malware

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WANF, Feifei, "Study on Detection and Protection Techniques of Mobile Phone Malicious Code under the Android Platform", CHINA MASTER'S THESES FULL-TEXT DATABASE (ELECTRONIC JOURNALS), no. 10, 15 October 2012 (15.10.2012), page 5, 3 rd line from the bottom to page 7, line 6 and page 18, line 1 to page 19, line 9	1-18
PX	CN 103577757 A (BEIJING QIHOO TECHNOLOGY CO., LTD. et al.), 12 February 2014 (12.02.2014), claims 1-10, and description, paragraphs 0014-0022 and 0162	1-18
A	KR 20070073361 A (CODEZEN. CO., LTD.), 10 July 2007 (10.07.2007), the whole document	1-18
A	CN 102663285 A (QIZHI SOFTWARE (BEIJING) CO., LTD.), 12 September 2012 (12.09.2012), the whole document	1-18
A	US 2010107160 A1 (NOVELL, INC.), 29 April 2010 (29.04.2010), the whole document	1-18

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier application or patent but published on or after the international filing date	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&" document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search 06 November 2014 (06.11.2014)	Date of mailing of the international search report 25 November 2014 (25.11.2014)
Name and mailing address of the ISA/CN: State Intellectual Property Office of the P. R. China No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088, China Facsimile No.: (86-10) 62019451	Authorized officer YAO, Jie Telephone No.: (86-10) 61648090

International application No.
PCT/CN2014/085187

Form PCT/ISA/210 (patent family annex) (July 2009)

国际检索报告

国际申请号

PCT/CN2014/085187

A. 主题的分类

G06F 21/56(2013.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))

CNPAT, CNKI, WPI, EPDOC: 攻击, 威胁, 安卓, 病毒, 应用层, 框架层, 恶意, frame, android, install+, layer, virus, Malware

C. 相关文件

类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
X	王菲飞, “基于Android平台的手机恶意代码检测与防护技术研究” 中国优秀硕士学位论文全文数据库(电子期刊), 第2012年第10期卷, 2012年 10月 15日 (2012 - 10 - 15), 第5页倒数第3行-第7页第6行, 第18页第1行-第19页第9行	1-18
PX	CN 103577757 A (北京奇虎科技有限公司 等) 2014年 2月 12日 (2014 - 02 - 12) 权利要求1-10, 说明书第0014-0022, 0162段	1-18
A	KR 20070073361 A (CODEZEN. CO., LTD.) 2007年 7月 10日 (2007 - 07 - 10) 全文	1-18
A	CN 102663285 A (奇智软件北京有限公司) 2012年 9月 12日 (2012 - 09 - 12) 全文	1-18
A	US 2010107160 A1 (NOVELL, INC.) 2010年 4月 29日 (2010 - 04 - 29) 全文	1-18

☐ 其余文件在C栏的续页中列出。☒ 见同族专利附件。

* 引用文件的具体类型:

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的)

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期

2014年 11月 06日

国际检索报告邮寄日期

2014年 11月 25日

ISA/CN的名称和邮寄地址

中华人民共和国国家知识产权局(ISA/CN)
北京市海淀区蓟门桥西土城路6号
100088 中国

传真号 (86-10)62019451

授权官员

姚杰

电话号码 (86-10)61648090

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2014/085187

检索报告引用的专利文件			公布日 (年/月/日)	同族专利	公布日 (年/月/日)
CN	103577757	A	2014年 2月 12日	无	
KR	20070073361	A	2007年 7月 10日	无	
CN	102663285	A	2012年 9月 12日	WO 2013139216 A1	2013年 9月 26日
US	2010107160	A1	2010年 4月 29日	无	