

说明书

实现安全防护的方法、防火墙、终端及可读存储介质

本申请要求于 2012 年 4 月 26 日提交中国专利局、申请号为
5 201210126865.3、发明名称为“一种实现安全防护的方法及一种防火墙”的中国
专利申请优先权，其全部内容通过引用结合在本申请中。

技术领域

本发明涉及计算机领域，特别涉及一种实现安全防护的方法、防火墙、终
10 端及可读存储介质。

背景技术

随着手机的功能增多，手机已经越来越被人们应用于网络中。但是，随着
手机功能的增大，其网络安全性也开始受到威胁，因此，在手机中安装防火墙
15 已经成为一种趋势。

在近期的 Android（安卓）手机系统中，主要是通过系统内核中的
iptables/netfilter 模块实现防火墙功能。但是，在早期版本的 Android 固件中并不
存在 iptables/netfilter 模块，因此，很多厂商采用了在编译手机的 ROM（Read-Only
Memory，只读内存）时，将 iptables/netfilter 模块添加进去的方式，以使得手机
20 在此基础上实现防火墙功能。

在实现本发明的过程中，发明人发现现有技术至少存在以下问题：

现有技术仍不能保证所有手机均具有 iptables/netfilter 模块，在没有
iptables/netfilter 模块的手机上仍不能实现防火墙功能，而且，现有的处理方式
很容易产生兼容性问题。

25

发明内容

为了解决现有技术中存在的技术问题，本发明实施例提供了一种实现安全
防护的方法、防火墙、终端及可读存储介质。所述技术方案如下：

一种实现安全防护的方法，所述方法运行于终端上，所述方法包括：

30 防火墙将包含有第一函数的代码模块注入到目标进程中，所述第一函数用

于修改非法应用对应的安全包信息;

所述防火墙触发所述第一函数执行修改非法应用对应的安全包信息的操作,使所述非法应用因自身对应的安全包信息被修改而执行失败。

5 进一步地,所述目标进程为包管理服务 PackageManagerService 的宿主进程,所述安全包信息为所述 PackageManagerService 的成员变量。

进一步地,所述防火墙将包含有第一函数的代码模块注入到目标进程中,具体包括:

所述防火墙获取自身进程的第二函数地址 A1 和第一动态链接库的基地址
10 A2;

通过查看预设路径的文件获取所述目标进程的第一动态链接库的基地址 A3;

计算得到所述目标进程的第二函数地址 $A4=A3-A2+A1$;

将所述目标进程的寄存器地址修改为 A4,并使所述第二函数所调用的动态链接库的地址指向所述第一函数的所属的代码模块的地址;
15

其中,所述第一动态链接库为所述防火墙自身进程和所述目标进程共同调用的动态链接库,所述第二函数位于所述第一动态链接库中,所述寄存器用于记录所述目标进程当前运行的函数的地址。

进一步地,所述触发所述第一函数执行修改非法应用对应的安全包信息的操作,具体包括:
20

寻找所述目标进程当前所调用的第二动态链接库的基地址及所述第二动态链接库的重定向表;

在所述重定向表中,获取需要重定向的第三函数的地址;

获取所述第一函数的地址 A5;

25 将所述重定向表中所述第三函数的地址重定向为 A5;

通过调用所述第三函数触发所述第一函数根据用户传递的参数修改非法应用对应的安全包信息的操作;

其中,所述第一函数为对所述第三函数的重写,所述第三函数包含在所述第二动态链接库中。

30 进一步地,所述第一函数根据用户传递的参数修改非法应用对应的安全包信息之前,所述方法还包括:

所述防火墙向操作系统注册得到本地服务，所述本地服务用于等待并接收用户传递的参数。

一种防火墙，所述防火墙运行于终端上，所述防火墙包括：

5 注入模块，用于将包含有第一函数的代码模块注入到目标进程中，所述第一函数用于修改非法应用对应的安全包信息；

触发模块，用于触发所述第一函数执行修改非法应用对应的安全包信息的操作，使所述非法应用因自身对应的安全包信息被修改而执行失败。

10 进一步地，所述目标进程为包管理服务 PackageManagerService 的宿主进程，所述安全包信息为所述 PackageManagerService 的成员变量。

进一步地，所述注入模块，具体包括：

第一地址获取单元，用于获取所述防火墙自身进程的第二函数地址 A1 和第一动态链接库的基地址 A2；

15 第二地址获取单元，用于通过查看预设路径的文件获取所述目标进程的第一动态链接库的基地址 A3；

地址计算单元，用于计算得到所述目标进程的第二函数地址 $A4=A3-A2+A1$ ；

20 地址修改单元，用于将所述目标进程的寄存器地址修改为 A4，并使所述第二函数所调用的动态链接库的地址指向第一函数的所属的代码模块的地址；

其中，所述第一动态链接库为所述防火墙自身进程和所述目标进程共同调用的动态链接库，所述第二函数位于所述第一动态链接库中，所述寄存器用于记录所述目标进程当前运行的函数的地址。

进一步地，所述触发模块，具体包括：

25 获取单元，用于寻找所述目标进程当前所调用的第二动态链接库的基地址及所述第二动态链接库的重定向表；

第三地址获取单元，用于在所述重定向表中，获取需要重定向的第三函数的地址；

第四地址获取单元，用于获取所述第一函数的地址 A5；

30 重定向单元，用于将所述重定向表中所述第三函数的地址重定向为 A5；

触发单元，用于通过调用所述第三函数触发所述第一函数根据用户传递的

参数修改非法应用对应的安全包信息的操作;

其中,所述第一函数为对所述第三函数的重写,所述第三函数包含在所述第二动态链接库中。

进一步地,所述防火墙还包括:

- 5 通讯模块,用于在所述触发模块触发所述第一函数根据用户传递的参数修改非法应用对应的安全包信息之前,向操作系统注册得到本地服务,所述本地服务用于等待并接收用户传递的参数。

一种终端,其特征在于,该终端包括:

- 10 存储器;

一个或者一个以上的程序,存储于所述存储器中,且经配置由一个或者一个以上处理器执行,所述一个或者一个以上程序包含用于进行以下操作的指令:

- 15 将包含有第一函数的代码模块注入到目标进程中,所述第一函数用于修改非法应用对应的安全包信息;

触发所述第一函数执行修改所述非法应用对应的安全包信息的操作,使所述非法应用因自身对应的安全包信息被修改而执行失败。

进一步地,所述将包含有第一函数的代码模块注入到目标进程中,包括:

获取自身进程的第二函数地址 A1 和第一动态链接库的基地址 A2;

- 20 通过查看预设路径的文件获取所述目标进程的第一动态链接库的基地址 A3;

计算得到所述目标进程的第二函数地址 $A4=A3-A2+A1$;

将所述目标进程的寄存器地址修改为 A4,并使所述第二函数所调用的动态链接库的地址指向所述第一函数所属的代码模块的地址;

- 25 其中,所述第一动态链接库为所述防火墙自身进程和所述目标进程共同调用的动态链接库,所述第二函数位于所述第一动态链接库中,所述寄存器用于记录所述目标进程当前运行的函数的地址。

进一步地,所述触发所述第一函数执行修改非法应用对应的安全包信息的操作,包括:

- 30 寻找所述目标进程当前所调用的第二动态链接库的基地址及所述第二动态链接库的重定向表;

在所述重定向表中，获取需要重定向的第三函数的地址；

获取所述第一函数的地址 A5；

将所述重定向表中所述第三函数的地址重定向为 A5；

通过调用所述第三函数触发所述第一函数根据用户传递的参数修改非法

5 应用对应的安全包信息的操作；

其中，所述第一函数为对所述第三函数的重写，所述第三函数包含在所述第二动态链接库中。

10 一种可读存储介质，其特征在于，所述可读存储介质存储有一个或者一个以上程序，所述一个或者一个以上程序被一个或者一个以上的处理器用来执行，以实现安全防护的方法，所述方法包括：

将包含有第一函数的代码模块注入到目标进程中，所述第一函数用于修改非法应用对应的安全包信息；

15 触发所述第一函数执行修改所述非法应用对应的安全包信息的操作，使所述非法应用因自身对应的安全包信息被修改而执行失败。

进一步地，所述将包含有第一函数的代码模块注入到目标进程中，包括：

获取自身进程的第二函数地址 A1 和第一动态链接库的基地址 A2；

通过查看预设路径的文件获取所述目标进程的第一动态链接库的基地址 A3；

20 计算得到所述目标进程的第二函数地址 $A4=A3-A2+A1$ ；

将所述目标进程的寄存器地址修改为 A4，并使所述第二函数所调用的动态链接库的地址指向所述第一函数所属的代码模块的地址；

25 其中，所述第一动态链接库为所述防火墙自身进程和所述目标进程共同调用的动态链接库，所述第二函数位于所述第一动态链接库中，所述寄存器用于记录所述目标进程当前运行的函数的地址。

进一步地，所述触发所述第一函数执行修改非法应用对应的安全包信息的操作，包括：

寻找所述目标进程当前所调用的第二动态链接库的基地址及所述第二动态链接库的重定向表；

30 在所述重定向表中，获取需要重定向的第三函数的地址；

获取所述第一函数的地址 A5；

将所述重定向表中所述第三函数的地址重定向为 A5;

通过调用所述第三函数触发所述第一函数根据用户传递的参数修改非法应用对应的安全包信息的操作;

其中,所述第一函数为对所述第三函数的重写,所述第三函数包含在所述
5 第二动态链接库中。

本发明实施例通过将具有修改安全包信息功能的代码模块注入到目标进程中,对非法应用对应的安全包信息进行修改,使非法应用因自身对应的安全包信息被修改而执行失败,使得终端在不具有 iptable/netfilter 模块的情况下也
10 可以实现防火墙功能,并且不存在兼容性问题,不受固件版本的限制。

附图说明

为了更清楚地说明本发明实施例中的技术方案,下面将对实施例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
15 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。

图 1 是本发明实施例 1 中提供的一种实现安全防护的方法的流程示意图;

图 2 是本发明实施例 2 中提供的一种实现安全防护的方法的流程示意图;

图 3 是本发明实施例 3 中提供的一种防火墙的结构示意图;

20 图 4 是本发明实施例 3 中提供的一种防火墙中的注入模块的结构示意图;

图 5 是本发明实施例 3 中提供的一种防火墙中的触发模块的结构示意图;

图 6 是本发明实施例 3 中提供的一种防火墙的第二种结构示意图;

图 7 是本发明实施例 4 中提供的一种终端的结构示意图;

图 8 是本发明实施例 4 中提供的一种终端的结构示意图。

25

具体实施方式

为使本发明的目的、技术方案和优点更加清楚,下面将结合附图对本发明实施方式作进一步地详细描述。

实施例 1

30 如图 1 所示,本发明实施例提供了一种实现安全防护的方法,其中,该方法可通过软件进行实现,并将该软件安装入终端中,上述方法具体包括如下步

骤:

101、将包含有第一函数的代码模块注入到目标进程中;

其中, 第一函数用于修改非法应用对应的安全包信息。

102、触发第一函数执行修改非法应用对应的安全包信息的操作, 使非法
5 应用因自身对应的安全包信息被修改而执行失败。

本实施例提供了一种实现安全防护的方法, 使得在不具有 iptable/netfilter 模块的情况下也可以实现防火墙功能, 并且不存在兼容性问题, 不受固件版本的限制。

10

实施例 2

本实施例提供了一种实现安全防护的方法, 是在实施例 1 的基础之上进行的改进。

需要说明的是, 在本实施例中, 下述方法可通过软件实现, 例如防火墙软
15 件, 并将防火墙软件安装于终端中, 且本发明所提供的方法适用于基于 Linux 开发的系统中。

在本实施例中, 以 Android 系统的终端为例进行说明, 但并不作为对本发明保护范围的限制。其中, Android 系统是基于 Linux 开发的移动操作系统, 但是其具备了比 Linux 粒度更细的权限安全机制。对于 Android 系统中的应用
20 (包括合法应用和非法应用), 应用的开发者需要预先在应用的配置文件里为自身申请权限。例如, 读写系统联系人、联网请求的权限。当应用编译时, 编译器会把权限信息、配置文件、资源文件和 Dex 文件打包在一起, 形成该应用在 Android 系统上可执行文件安装包, 具体可以为 APK 包 (AndroidPackage, Android 安装包)。其中, 资源文件包括文本字符串(strings)、颜色(colors)、数
25 组(arrays)、动画(anim)、布局(layout)、图像和图标(drawable)、音频视频(media)和其他使用的组件; Dex 文件是 Android 系统上可执行文件的类型。

在 Android 系统的终端上, 每当安装一个应用, 都要经过以下两个步骤:

A. 缓存该应用的安装包的信息, 即记录在 APK 包中的信息, 操作系统会
统一把安装包的信息记录在路径为 /data/system/ 下的 packages.list 和
30 packages.xml 两个文件里, packages.list 和 packages.xml 中保存着每个已经安装的 APK 的详尽信息。

B. 后台的 PackageManagerService (包管理服务) 服务, 会从 packages.list 和 packages.xml 中加载信息到内存中。

由于, 在 Android 系统的终端每启动一个应用, 都经由 ActivityManagerService —— PackageManagerService —— Zygote 三个关键环节, 因此, 从 PackageManagerService 传递给 Zygote 的参数 (uid、gid 和 gids), 也即安全包信息, 决定了应用在启动后所具备的权限。

其中, ActivityManagerService 用于对应用访问权限进行判定, 对应用生命周期进行管理;

PackageManagerService 是在系统启动阶段由 system_server 启动的一个 java 层服务, 用来管理 APK 文件, 决定应用启动后可以具备哪些功能;

Zygote 是一个虚拟机进程, 同时也是一个虚拟机实例的孵化器, 每当系统要求执行一个 Android 应用, Zygote 就会通过派生命令创建出一个子进程来执行该应用。

其中, uid: Android 系统的终端在安装一个应用, 就会为它分配一个 uid, 代表该应用的唯一标识。其中普通 Android 应用程序的 uid 是从 10000 开始分配, 10000 以下是系统进程的 uid。

gid: 对于普通应用来说, gid 等于 uid。由于每个应用的 uid 和 gid 都不相同, 因此不管是 native 层还是 java 层都能够达到保护私有数据的作用。

gids: 由框架在应用安装过程中生成, 与应用申请的具体权限相关。

本实施例提供了一种方法, 通过修改 PackageManagerService 中的安全包信息, 以使得非法的应用在运行时出错, 以结束非法应用, 实现了防火墙的功能。

本实施例所提供的方法, 具体通过安装于终端中的防火墙进行实现, 该防火墙可包括四个代码模块: 注入模块、控制模块、通信模块和启动模块, 其中, 注入模块用于将控制模块对应的代码注入到 PackageManagerService 的宿主进程 system_server 中, 启动模块用于触发控制模块的启动, 通信模块用于接收用户传递的参数, 并将用户传递的参数发送给控制模块, 控制模块用于修改 PackageManagerService 中的成员变量。

如图 2 所示, 一种实现安全防护的方法, 具体包括如下步骤:

201、将包含有第一函数的控制模块注入到目标进程中;

其中, 第一函数用于修改非法应用对应的安全包信息, 第一函数中具有预

定的逻辑，即包括有修改安全包信息的功能的代码，其从属于控制模块，第一函数通过对目标进程所执行的函数进行重写并加入具有修改安全包信息的功能的代码，目标进程具体的为 PackageManagerService 的宿主进程 system_server。

- 5 进一步地，将第一函数的控制模块注入到目标进程中，需要通过目标进程与防火墙自身进程共同调用的第二函数及第一动态链接库的地址计算得到第一动态链接库的基地址，以便将第一动态链接库的地址指向控制模块的地址，使得目标进程在执行任务时，可执行控制模块中的第一函数。

在本实施例中，由于需要使目标进程一定能够执行到被注入的控制模块，
10 因此需要使用防火墙自身进程和目标进程均需调用的动态链接库，在本实施例中以 lico.so 作为第一动态链接库，dlopen 函数作为第二函数为例进行说明，dlopen 函数为 lico.so 中的函数。

具体包括如下过程：

- 201-1、防火墙获取自身进程的 dlopen 函数的地址 addr_my_dlopen 以及
15 libc.so 的基地址 addr_my_base；

其中，dlopen 函数用于以指定模式打开指定的第一动态链接库，其中包含有欲打开的动态链接库的地址。

201-2、通过查看路径/proc/pid/maps，获取目标进程的 libc.so 的基地址
addr_other_base；

- 20 201-3、计算得到目标进程的 dlopen 函数的基地址 addr_other_dlopen；

其中， $\text{addr_other_dlopen} = \text{addr_other_base} - \text{addr_my_base} + \text{addr_my_dlopen}$ 。

通过找到第一动态链接库的基地址再加上第二函数的地址偏移量，即可在进程中找到第二函数的基地址。其中，第二函数的作用是用于打开动态链接库。

- 201-4、将目标进程的寄存器地址修改为 addr_other_dlopen，并使目标进程
25 的 dlopen 函数的地址指向控制模块的地址。

通过将目标进程的寄存器地址修改为第二函数的基地址，通过第二函数为入口，将第一函数的控制模块注入到目标进程。

- 其中，步骤 201-4 的操作通过 ptrace 函数实现，ptrace 函数为 linux 提供的具有修改寄存器地址、函数地址等功能的函数，寄存器用于存储目标进程当前
30 运行的函数的地址。

202、寻找目标进程当前调用的第二动态链接库的基地址及第二动态链接

库的重定向表;

203、在重定向表中, 获取需要重定向的第三函数的地址;

其中, 为了使对目标进程进行注入后, 使得目标进程执行第一函数后, 又不丧失原功能, 第一函数应为对第三函数的重写, 第三函数为目标进程需要调用的第二动态链接库中的函数。

204、获取第一函数的地址, 并将重定向表中的第三函数的地址重定向为第一函数的地址;

这个过程是为了在目标进程中执行第三函数时替换为执行第一函数, 触发第一函数来修改应用对应的安全包信息。

205、执行第三函数, 以触发第一函数中预定的逻辑;

其中, 由于第三函数在重定向表中的地址被修改, 因此, 当目标进程执行第三函数时, 实际转到第一函数的地址, 触发了第一函数中的代码所规定的逻辑。

需要说明的是, 步骤 202-205 的操作可通过 ptrace 函数进行实现。

206、通过通讯模块获取用户传递的参数;

具体地, 用户可通过防火墙的软件界面对手机中的应用进行控制, 软件界面接到用户的控制命令后, 将对应的参数传递给通讯模块, 通讯模块再将参数传递给控制模块, 其中, 上述参数即为用户传递的参数, 参数具体为用户设置应用的各种权限的权限信息。

通讯模块获取用户传递的参数, 具体包括:

防火墙向操作系统注册得到本地服务 (例如命名为 Local Service), 并使 Local Service 的宿主进程 Firewall_Service 处于等待软件界面连接的状态, 以接收用户传递的参数, 例如 on/off、app_name 等。

其中, 通讯模块获取 Firewall_Service 发送的用户传递的参数, 通过 Android 系统的 IPC (Inter-Process Communication, 进程间通信) ——binder 通讯实现。

207、根据用户传递的参数修改非法应用对应的安全包信息, 使非法应用因自身对应的安全包信息被修改导致执行失败;

需要说明的是, 当非法应用的安全包信息 (例如 gid) 被修改后, 非法应用在执行时, 不具有合法的 gid, 则不能正常运行。

进一步的, 还可以对合法应用进行修改, 限制合法应用的权限。

本实施例提供了一种实现安全防护的方法, 使得在不具有 iptable/netfilter

模块的情况下也可以实现防火墙功能，并且不存在兼容性问题，不受固件版本的限制。

实施例 3

5 如图 3 所示，本实施例提供了一种防火墙，该防火墙包括：

注入模块 301，用于将含有第一函数的代码模块注入到目标进程中，该第一函数用于修改非法应用对应的安全包信息；

触发模块 302，用于触发第一函数执行修改非法应用对应的安全包信息的操作，使非法应用因自身对应的安全包信息被修改而执行失败。

10

进一步地，目标进程为包管理服务 PackageManagerService 的宿主进程，安全包信息为 PackageManagerService 的成员变量。

进一步地，如图 4 所示，注入模块 301，具体包括：

15 第一地址获取单元 3011，用于获取防火墙自身进程的第二函数地址 A1 和第一动态链接库的基地址 A2；

第二地址获取单元 3012，用于通过查看预设路径的文件获取目标进程的第一动态链接库的基地址 A3；

地址计算单元 3013，用于计算得到目标进程的第二函数地址 $A4=A3-A2+A1$ ；

20 地址修改单元 3014，用于将目标进程的寄存器地址修改为 A4，并使第二函数所调用的动态链接库的地址指向第一函数的所属的代码模块的地址；

其中，第一动态链接库为防火墙自身进程和目标进程共同调用的动态链接库，第二函数位于第一动态链接库中，寄存器用于记录目标进程当前运行的函数的地址。

25 进一步地，如图 5 所示，触发模块 302，具体包括：

获取单元 3021，用于寻找目标进程当前所调用的第二动态链接库的基地址及第二动态链接库的重定向表；

第三地址获取单元 3022，用于在重定向表中，获取需要重定向的第三函数的地址；

30 第四地址获取单元 3023，用于获取第一函数的地址 A5；

重定向单元 3024，用于将重定向表中第三函数的地址重定向为 A5；

触发单元 3025, 用于通过调用第三函数触发第一函数根据用户传递的参数修改非法应用对应的安全包信息的操作;

其中, 第一函数为对第三函数的重写, 第三函数包含在第二动态链接库中。

进一步地, 如图 6 所述, 上述防火墙还包括:

- 5 通讯模块 303, 用于在触发模块 302 触发第一函数根据用户传递的参数修改非法应用对应的安全包信息之前, 向操作系统注册得到本地服务, 本地服务用于等待并接收用户传递的参数。

- 10 本实施例提供了一种防火墙, 使得在不具有 iptable/netfilter 模块的情况下也可以实现防火墙功能, 并且不存在兼容性问题, 不受固件版本的限制。

实施例 4

如图 7 所示, 本实施例提供了一种终端,

该终端 1200 包括:

- 15 存储器 1201 和至少一个处理器 1202, 其中, 存储器 1201 存储软件程序以及模块, 如本发明实施例中的防火墙所对应的程序指令及模块, 以及实现本发明实施例中实现安全防护过程中生成的数据, 处理器 1202 通过运行存储在存储器 1201 的软件程序以及模块, 从而执行各种功能应用以及数据处理, 即实现本发明中的安全防护功能。

- 20 其中, 所述处理器 1202 被配置为执行如下显示图片的操作:

防火墙将包含有第一函数的代码模块注入到目标进程中, 所述第一函数用于修改非法应用对应的安全包信息;

所述防火墙触发所述第一函数执行修改所述非法应用对应的安全包信息的操作, 使所述非法应用因自身对应的安全包信息被修改而执行失败。

- 25 其中, 所述目标进程为包管理服务 PackageManagerService 的宿主进程, 所述安全包信息为所述 PackageManagerService 的成员变量。

其中, 所述防火墙将包含有第一函数的代码模块注入到目标进程中, 具体包括:

- 30 所述防火墙获取自身进程的第二函数地址 A1 和第一动态链接库的基地址 A2;

通过查看预设路径的文件获取所述目标进程的第一动态链接库的基地址

A3;

计算得到所述目标进程的第二函数地址 $A4=A3-A2+A1$;

将所述目标进程的寄存器地址修改为 A4, 并使所述第二函数所调用的动态链接库的地址指向所述第一函数的所属的代码模块的地址;

5 其中, 所述第一动态链接库为所述防火墙自身进程和所述目标进程共同调用的动态链接库, 所述第二函数位于所述第一动态链接库中, 所述寄存器用于记录所述目标进程当前运行的函数的地址。

其中, 所述触发所述第一函数执行修改非法应用对应的安全包信息的操作, 具体包括:

10 寻找所述目标进程当前所调用的第二动态链接库的基地址及所述第二动态链接库的重定向表;

在所述重定向表中, 获取需要重定向的第三函数的地址;

获取所述第一函数的地址 A5;

将所述重定向表中所述第三函数的地址重定向为 A5;

15 通过调用所述第三函数触发所述第一函数根据用户传递的参数修改非法应用对应的安全包信息的操作;

其中, 所述第一函数为对所述第三函数的重写, 所述第三函数包含在所述第二动态链接库中。

20 其中, 所述第一函数根据用户传递的参数修改非法应用对应的安全包信息之前, 所述方法还包括:

所述防火墙向操作系统注册得到本地服务, 所述本地服务用于等待并接收用户传递的参数。

优选的, 如图 8 所示, 该终端 1200 还为以下各单元构成, 包括:

25 RF (Radio Frequency, 射频) 电路 110、一个或一个以上计算机可读存储介质的存储器 120、输入单元 130、显示单元 140、传感器 150、音频电路 160、传输模块 170、一个或者一个以上处理核心的处理器 180、以及电源 190 等部件。

本领域技术人员可以理解, 图 7 中示出的终端结构并不构成对终端的限定, 30 可以包括比图示更多或更少的部件, 或者组合某些部件, 或者不同的部件布置。其中:

RF 电路 110 可用于收发信息或通话过程中, 信号的接收和发送, 特别地, 将基站的下行信息接收后, 交由一个或者一个以上处理器 180 处理; 另外, 将涉及上行的数据发送给基站。通常, RF 电路 110 包括但不限于天线、至少一个放大器、调谐器、一个或多个振荡器、用户身份模块 (SIM) 卡、收发信机、
5 耦合器、LNA (Low Noise Amplifier, 低噪声放大器)、双工器等。此外, RF 电路 110 还可以通过无线通信与网络和其他设备通信实现。所述无线通信可以使用任一通信标准或协议, 包括但不限于 GSM(Global System of Mobile communication, 全球移动通讯系统)、GPRS(General Packet Radio Service, 通用分组无线服务)、CDMA(Code Division Multiple Access, 码分多址)、
10 WCDMA(Wideband Code Division Multiple Access, 宽带码分多址)、LTE(Long Term Evolution, 长期演进)、电子邮件、SMS(Short Messaging Service, 短消息服务)等。

存储器 120 可用于存储软件程序以及模块, 如本发明实施例中的防火墙所对应的程序指令及模块, 以及实现本发明实施例中实现安全防护过程中生成的数据, 处理器 180 通过运行存储在存储器 120 的软件程序以及模块, 从而执行
15 各种功能应用以及数据处理, 即实现本发明中的安全防护功能。存储器 120 可主要包括存储程序区和存储数据区, 其中, 存储程序区可存储操作系统、至少一个功能所需的应用程序 (比如声音播放功能、图像播放功能等) 等; 存储数据区可存储根据终端 1200 的使用所创建的数据 (比如音频数据、电话本等)
20 等。此外, 存储器 120 可以包括高速随机存取存储器, 还可以包括非易失性存储器, 例如至少一个磁盘存储器件、闪存器件、或其他易失性固态存储器件。相应地, 存储器 120 还可以包括存储器控制器, 以提供处理器 180 和输入单元 130 对存储器 120 的访问。

输入单元 130 可用于接收输入的数字或字符信息, 以及产生与用户设置以及功能控制有关的键盘、鼠标、操作杆、光学或者轨迹球信号输入。具体地,
25 输入单元 130 可包括触敏表面 131 以及其他输入设备 132。触敏表面 131, 也称为触摸显示屏或者触控板, 可收集用户在其上或附近的触摸操作 (比如用户使用手指、触笔等任何适合的物体或附件在触敏表面 131 上或在触敏表面 131 附近的操作), 并根据预先设定的程式驱动相应的连接装置。可选的, 触敏表面 131 可包括触摸检测装置和触摸控制器两个部分。其中, 触摸检测装置检测
30 用户的触摸方位, 并检测触摸操作带来的信号, 将信号传送给触摸控制器; 触

摸控制器从触摸检测装置上接收触摸信息，并将它转换成触点坐标，再送给处理器 180，并能接收处理器 180 发来的命令并加以执行。此外，可以采用电阻式、电容式、红外线以及表面声波等多种类型实现触敏表面 131。除了触敏表面 131，输入单元 130 还可以包括其他输入设备 132。具体地，其他输入设备 5 132 可以包括但不限于物理键盘、功能键（比如音量控制按键、开关按键等）、轨迹球、鼠标、操作杆等中的一种或多种。

显示单元 140 可用于显示由用户输入的信息或提供给用户的信息以及终端 1200 的各种图形用户接口，这些图形用户接口可以由图形、文本、图标、视频和其任意组合来构成。显示单元 140 可包括显示面板 141，可选的，可以采用 10 LCD(Liquid Crystal Display, 液晶显示器)、OLED(Organic Light-Emitting Diode, 有机发光二极管)等形式来配置显示面板 141。进一步的，触敏表面 131 可覆盖显示面板 141，当触敏表面 131 检测到在其上或附近的触摸操作后，传送给处理器 180 以确定触摸事件的类型，随后处理器 180 根据触摸事件的类型在显示面板 141 上提供相应的视觉输出。虽然在图 11 中，触敏表面 131 与显示面板 15 141 是作为两个独立的部件来实现输入和输入功能，但是在某些实施例中，可以将触敏表面 131 与显示面板 141 集成而实现输入和输出功能。

终端 1200 还可包括至少一种传感器 150，比如光传感器、运动传感器以及其他传感器。具体地，光传感器可包括环境光传感器及接近传感器，其中，环境光传感器可根据环境光线的明暗来调节显示面板 141 的亮度，接近传感器可在终端 1200 移动到耳边时，关闭显示面板 141 和/或背光。作为运动传感器的一种，重力加速度传感器可检测各个方向上（一般为三轴）加速度的大小，静止时可检测出重力的大小及方向，可用于识别手机姿态的应用（比如横竖屏切换、相关游戏、磁力计姿态校准）、振动识别相关功能（比如计步器、敲击）等；至于终端 1200 还可配置的陀螺仪、气压计、湿度计、温度计、红外线传感器等其他传感器，在此不再赘述。 25

音频电路 160、扬声器 161，传声器 162 可提供用户与终端 1200 之间的音频接口。音频电路 160 可将接收到的音频数据转换后的电信号，传输到扬声器 161，由扬声器 161 转换为声音信号输出；另一方面，传声器 162 将收集的声音信号转换为电信号，由音频电路 160 接收后转换为音频数据，再将音频数据 30 输出处理器 180 处理后，经 RF 电路 110 以发送给比如另一终端，或者将音频数据输出至存储器 120 以便进一步处理。音频电路 160 还可能包括耳塞插孔，

以提供外设耳机与终端 1200 的通信。

WiFi 属于短距离无线传输技术,终端 1200 通过 WiFi 模块 170 可以帮助用户收发电子邮件、浏览网页和访问流式媒体等,它为用户提供了无线的宽带互联网访问。虽然图 11 示出了 WiFi 模块 170,但是可以理解的是,其并不属于终端 1200 的必须构成,完全可以根据需要在不改变发明的本质的范围内而省略。

处理器 180 是终端 1200 的控制中心,利用各种接口和线路连接整个手机的各个部分,通过运行或执行存储在存储器 120 内的软件程序和/或模块,以及调用存储在存储器 120 内的数据,执行终端 1200 的各种功能和处理数据,从而对手机进行整体监控。可选的,处理器 180 可包括一个或多个处理核心;优选的,处理器 180 可集成应用处理器和调制解调处理器,其中,应用处理器主要处理操作系统、用户界面和应用程序等,调制解调处理器主要处理无线通信。可以理解的是,上述调制解调处理器也可以不集成到处理器 180 中。

终端 1200 还包括给各个部件供电的电源 190 (比如电池),优选的,电源可以通过电源管理系统与处理器 180 逻辑相连,从而通过电源管理系统实现管理充电、放电、以及功耗管理等功能。电源 190 还可以包括一个或一个以上的直流或交流电源、再充电系统、电源故障检测电路、电源转换器或者逆变器、电源状态指示器等任意组件。

尽管未示出,终端 1200 还可以包括摄像头、蓝牙模块等,在此不再赘述。具体在本实施例中,终端的显示单元是触摸屏显示器,终端还包括有存储器,以及一个或者一个以上的程序,其中一个或者一个以上程序存储于存储器中,且经配置以由一个或者一个以上处理器执行述一个或者一个以上程序包含用于进行以下操作的指令:

本实施例提供了一种终端,使得在不具有 iptable/netfilter 模块的情况下也可以实现防火墙功能,并且不存在兼容性问题,不受固件版本的限制。

实施例 5

本实施例提供了一种可读存储介质,该可读存储介质可以是上述实施例中的存储器中所包含的可读存储介质;也可以是单独存在,未装配入终端中的计算机可读存储介质。所述计算机可读存储介质存储有一个或者一个以上程序,

所述一个或者一个以上程序被一个或者一个以上的处理器用来执行一个实现安全防护的方法，所述方法包括：

防火墙将包含有第一函数的代码模块注入到目标进程中，所述第一函数用于修改非法应用对应的安全包信息；

5 所述防火墙触发所述第一函数执行修改所述非法应用对应的安全包信息的操作，使所述非法应用因自身对应的安全包信息被修改而执行失败。

假设上述为第一种可能的实施方式，则在第一种可能的实施方式作为基础而提供的第二种可能的实施方式中，所述目标进程为包管理服务 PackageManagerService 的宿主进程，所述安全包信息为所述
10 PackageManagerService 的成员变量。

在第一种可能的实施方式作为基础而提供的第三种可能的实施方式中，所述防火墙将包含有第一函数的代码模块注入到目标进程中，具体包括：

所述防火墙获取自身进程的第二函数地址 A1 和第一动态链接库的基地址 A2；

15 通过查看预设路径的文件获取所述目标进程的第一动态链接库的基地址 A3；

计算得到所述目标进程的第二函数地址 $A4=A3-A2+A1$ ；

将所述目标进程的寄存器地址修改为 A4，并使所述第二函数所调用的动态链接库的地址指向所述第一函数的所属的代码模块的地址；

20 其中，所述第一动态链接库为所述防火墙自身进程和所述目标进程共同调用的动态链接库，所述第二函数位于所述第一动态链接库中，所述寄存器用于记录所述目标进程当前运行的函数的地址。

在第一种可能的实施方式作为基础而提供的第四种可能的实施方式中，所述触发所述第一函数执行修改非法应用对应的安全包信息的操作，具体包括：

25 寻找所述目标进程当前所调用的第二动态链接库的基地址及所述第二动态链接库的重定向表；

在所述重定向表中，获取需要重定向的第三函数的地址；

获取所述第一函数的地址 A5；

将所述重定向表中所述第三函数的地址重定向为 A5；

30 通过调用所述第三函数触发所述第一函数根据用户传递的参数修改非法应用对应的安全包信息的操作；

其中，所述第一函数为对所述第三函数的重写，所述第三函数包含在所述第二动态链接库中。

在第四种可能的实施方式作为基础而提供的第五种可能的实施方式中，所述
5 所述第一函数根据用户传递的参数修改非法应用对应的安全包信息之前，所述方法还包括：

所述防火墙向操作系统注册得到本地服务，所述本地服务用于等待并接收用户传递的参数。

本实施例提供了一种可读存储介质，使得在不具有 iptable/netfilter 模块的
10 情况下也可以实现防火墙功能，并且不存在兼容性问题，不受固件版本的限制。

本领域普通技术人员可以理解实现上述实施例的全部或部分步骤可以通过硬件来完成，也可以通过程序来指令相关的硬件完成，所述的程序可以存储于一种计算机可读存储介质中，上述提到的存储介质可以是只读存储器，磁盘
15 或光盘等。

以上所述仅为本发明的较佳实施例，并不用以限制本发明，凡在本发明的精神和原则之内，所作的任何修改、等同替换、改进等，均应包含在本发明的保护范围之内。

权 利 要 求 书

1、一种实现安全防护的方法，所述方法应用于终端上，其特征在于，所述方法包括：

5 防火墙将包含有第一函数的代码模块注入到目标进程中，所述第一函数用于修改非法应用对应的安全包信息；

所述防火墙触发所述第一函数执行修改所述非法应用对应的安全包信息的操作，使所述非法应用因自身对应的安全包信息被修改而执行失败。

10 2、根据权利要求1所述的方法，其特征在于，所述目标进程为包管理服务 PackageManagerService 的宿主进程，所述安全包信息为所述 PackageManagerService 的成员变量。

15 3、根据权利要求1所述的方法，其特征在于，所述防火墙将包含有第一函数的代码模块注入到目标进程中，具体包括：

所述防火墙获取自身进程的第二函数地址 A1 和第一动态链接库的基地址 A2；

通过查看预设路径的文件获取所述目标进程的第一动态链接库的基地址 A3；

20 计算得到所述目标进程的第二函数地址 $A4=A3-A2+A1$ ；

将所述目标进程的寄存器地址修改为 A4，并使所述第二函数所调用的动态链接库的地址指向所述第一函数所属的代码模块的地址；

其中，所述第一动态链接库为所述防火墙自身进程和所述目标进程共同调用的动态链接库，所述第二函数位于所述第一动态链接库中，所述寄存器用于
25 记录所述目标进程当前运行的函数的地址。

4、根据权利要求1所述的方法，其特征在于，所述触发所述第一函数执行修改非法应用对应的安全包信息的操作，具体包括：

30 寻找所述目标进程当前所调用的第二动态链接库的基地址及所述第二动态链接库的重定向表；

在所述重定向表中，获取需要重定向的第三函数的地址；

获取所述第一函数的地址 A5;

将所述重定向表中所述第三函数的地址重定向为 A5;

通过调用所述第三函数触发所述第一函数根据用户传递的参数修改非法应用对应的安全包信息的操作;

- 5 其中, 所述第一函数为对所述第三函数的重写, 所述第三函数包含在所述第二动态链接库中。

5、根据权利要求 4 所述的方法, 其特征在于, 所述第一函数根据用户传递的参数修改非法应用对应的安全包信息之前, 所述方法还包括:

- 10 所述防火墙向操作系统注册得到本地服务, 所述本地服务用于等待并接收用户传递的参数。

6、一种防火墙, 所述防火墙运行于终端上, 其特征在于, 所述防火墙包括:

- 15 注入模块, 用于将包含有第一函数的代码模块注入到目标进程中, 所述第一函数用于修改非法应用对应的安全包信息;

触发模块, 用于触发所述第一函数执行修改所述非法应用对应的安全包信息的操作, 使所述非法应用因自身对应的安全包信息被修改而执行失败。

- 20 7、根据权利要求 6 所述的防火墙, 其特征在于, 所述目标进程为包管理服务 PackageManagerService 的宿主进程, 所述安全包信息为所述 PackageManagerService 的成员变量。

8、根据权利要求 6 所述的防火墙, 其特征在于, 所述注入模块, 具体包括:

- 25 第一地址获取单元, 用于获取所述防火墙自身进程的第二函数地址 A1 和第一动态链接库的基地址 A2;

第二地址获取单元, 用于通过查看预设路径的文件获取所述目标进程的第一动态链接库的基地址 A3;

地址计算单元, 用于计算得到所述目标进程的第二函数地址 $A4=A3-A2+A1$;

- 30 地址修改单元, 用于将所述目标进程的寄存器地址修改为 A4, 并使所述第二函数所调用的动态链接库的地址指向第一函数所属的代码模块的地址;

其中，所述第一动态链接库为所述防火墙自身进程和所述目标进程共同调用的动态链接库，所述第二函数位于所述第一动态链接库中，所述寄存器用于记录所述目标进程当前运行的函数的地址。

- 5 9、根据权利要求6所述的防火墙，其特征在于，所述触发模块，具体包括：
获取单元，用于寻找所述目标进程当前所调用的第二动态链接库的基地址及所述第二动态链接库的重定向表；

第三地址获取单元，用于在所述重定向表中，获取需要重定向的第三函数的地址；

- 10 第四地址获取单元，用于获取所述第一函数的地址 A5；
重定向单元，用于将所述重定向表中所述第三函数的地址重定向为 A5；
触发单元，用于通过调用所述第三函数触发所述第一函数根据用户传递的参数修改非法应用对应的安全包信息的操作；

- 15 其中，所述第一函数为对所述第三函数的重写，所述第三函数包含在所述第二动态链接库中。

- 10、根据权利要求9所述的防火墙，其特征在于，所述防火墙还包括：

- 20 通讯模块，用于在所述触发模块触发所述第一函数根据用户传递的参数修改非法应用对应的安全包信息之前，向操作系统注册得到本地服务，所述本地服务用于等待并接收用户传递的参数。

- 11、一种终端，其特征在于，该终端包括：

存储器；

- 25 一个或者一个以上的程序，存储于所述存储器中，且经配置由一个或者一个以上处理器执行，所述一个或者一个以上程序包含用于进行以下操作的指令：

将包含有第一函数的代码模块注入到目标进程中，所述第一函数用于修改非法应用对应的安全包信息；

触发所述第一函数执行修改所述非法应用对应的安全包信息的操作，使所述非法应用因自身对应的安全包信息被修改而执行失败。

- 30 12、根据权利要求11所述的终端，其特征在于，所述将包含有第一函数的

代码模块注入到目标进程中，包括：

获取自身进程的第二函数地址 A1 和第一动态链接库的基地址 A2；

通过查看预设路径的文件获取所述目标进程的第一动态链接库的基地址 A3；

5 计算得到所述目标进程的第二函数地址 $A4=A3-A2+A1$ ；

将所述目标进程的寄存器地址修改为 A4，并使所述第二函数所调用的动态链接库的地址指向所述第一函数所属的代码模块的地址；

其中，所述第一动态链接库为所述防火墙自身进程和所述目标进程共同调用的动态链接库，所述第二函数位于所述第一动态链接库中，所述寄存器用于
10 记录所述目标进程当前运行的函数的地址。

13、根据权利要求 11 所述的终端，其特征在于，所述触发所述第一函数执行修改非法应用对应的安全包信息的操作，包括：

寻找所述目标进程当前所调用的第二动态链接库的基地址及所述第二动态
15 链接库的重定向表；

在所述重定向表中，获取需要重定向的第三函数的地址；

获取所述第一函数的地址 A5；

将所述重定向表中所述第三函数的地址重定向为 A5；

通过调用所述第三函数触发所述第一函数根据用户传递的参数修改非法应
20 用对应的安全包信息的操作；

其中，所述第一函数为对所述第三函数的重写，所述第三函数包含在所述第二动态链接库中。

14、一种可读存储介质，其特征在于，所述可读存储介质存储有一个或者
25 一个以上程序，所述一个或者一个以上程序被一个或者一个以上的处理器用来执行，以实现安全防护的方法，所述方法包括：

将包含有第一函数的代码模块注入到目标进程中，所述第一函数用于修改非法应用对应的安全包信息；

触发所述第一函数执行修改所述非法应用对应的安全包信息的操作，使所
30 述非法应用因自身对应的安全包信息被修改而执行失败。

15、根据权利要求 14 所述的可读存储介质，其特征在于，所述将包含有第一函数的代码模块注入到目标进程中，包括：

获取自身进程的第二函数地址 A1 和第一动态链接库的基地址 A2；

通过查看预设路径的文件获取所述目标进程的第一动态链接库的基地址

5 A3；

计算得到所述目标进程的第二函数地址 $A4=A3-A2+A1$ ；

将所述目标进程的寄存器地址修改为 A4，并使所述第二函数所调用的动态链接库的地址指向所述第一函数所属的代码模块的地址；

其中，所述第一动态链接库为所述防火墙自身进程和所述目标进程共同调用的动态链接库，所述第二函数位于所述第一动态链接库中，所述寄存器用于记录所述目标进程当前运行的函数的地址。

16、根据权利要求 14 所述的可读存储介质，其特征在于，所述触发所述第一函数执行修改非法应用对应的安全包信息的操作，包括：

15 寻找所述目标进程当前所调用的第二动态链接库的基地址及所述第二动态链接库的重定向表；

在所述重定向表中，获取需要重定向的第三函数的地址；

获取所述第一函数的地址 A5；

将所述重定向表中所述第三函数的地址重定向为 A5；

20 通过调用所述第三函数触发所述第一函数根据用户传递的参数修改非法应用对应的安全包信息的操作；

其中，所述第一函数为对所述第三函数的重写，所述第三函数包含在所述第二动态链接库中。