(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 107798240 A
(43)申请公布日 2018.03.13

(21)申请号 201610806113.X

(22)申请日 2016.09.07

(71)申请人 武汉安天信息技术有限责任公司
地址 430000 湖北省武汉市东湖新技术开发区软件园东路1号软件产业4-1期B4栋12层01室

(72)发明人 曾祥刚 乔伟

(51)Int.Cl.
G06F 21/56(2013.01)

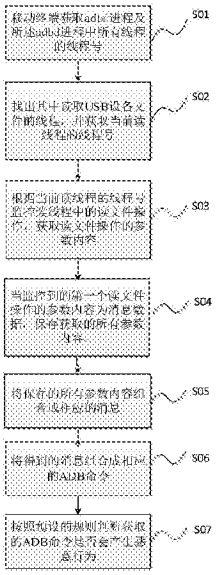
权利要求书2页 说明书8页 附图2页

(54)发明名称

一种用于监控PC端操作移动设备的方法及装置

(57)摘要

本发明公开了一种用于监控PC端操作移动设备的监控方法，利用了PC端访问移动终端时使用ADB命令与Android系统内的adbd进程进行交互的特点，在移动设备端监控adbd进程所有线程中的读USB设备文件的操作，以获取该操作的参数内容，然后将参数内容组合成消息，将消息组合成ADB命令，以获取PC端访问移动终端时发送的命令，从而达到监控的目的。该方法不受PC端环境影响，使用方便，监控效果好。本发明还公开了一种用于监控PC端操作移动设备的监控装置。



CN 107798240 A

1.一种用于监控PC端操作移动设备的监控方法，其特征在于，所述监控方法包括以下步骤：

移动终端获取adbd进程及所述adbd进程中所有线程的线程号；

根据获得的所有线程号分别监控每个线程，找出其中读取USB设备文件的线程并获取当前读线程的线程号；

根据当前读线程的线程号监控该线程中的读文件操作，获取读文件操作的参数内容，所述参数内容包括消息头或消息数据；

当监控到的第一个读文件操作的参数内容为消息数据时，则保存获取的所有参数内容；

将保存的所有参数内容组合成相应的消息，其中，一个消息头与对应的一个消息数据可组合为一条消息；

将得到的消息组合成相应的ADB命令，其中，一条ADB命令包含至少一条消息，属于同一命令的消息拥有相同的ID号，所述ID号位于所述消息头内；

按照预设的规则判断获取的ADB命令是否会产生恶意行为。

2.如权利要求1所述的监控方法，其特征在于，按照预设的时间对每个线程进行监控寻找相应的读文件操作，若在该时间段内没有找到，则立即监控下一个线程。

3.如权利要求1所述的监控方法，其特征在于，找出全部线程号后，逐个对每个线程进行监控，如果当前线程执行读文件操作，并且第一个参数指向的是USB设备文件，则获取当前线程的线程号。

4.如权利要求1所述的监控方法，其特征在于，当终端没有连接USB或者查找读取USB设备文件的线程失败后，会再次获取全部线程号，并查找是否有读取USB设备文件的线程。

5.如权利要求1所述的监控方法，其特征在于，若判断出ADB命令会产生恶意行为，则按照预设的规则修改当前ADB命令，阻止恶意行为发生或/和产生用户告警。

6.一种用于监控PC端操作移动设备的监控装置，其特征在于，所述监控装置包括守护模块、监控模块、检测模块，当PC端访问移动终端时：

所述守护模块用于获取adbd进程及所述adbd进程中所有线程的线程号；根据获得的所有线程号分别监控每个线程，找出其中读取USB设备文件的线程并获取当前读线程的线程号发送给所述监控模块；

所述监控模块用于根据当前读线程的线程号监控读线程中的读文件操作，获取读文件操作的参数内容，所述参数内容包括消息头或消息数据，当判断监控到的第一个读文件操作的参数内容为消息数据时，则保存获取的所有参数内容；

所述检测模块用于将保存的所有参数内容组合成相应的消息，其中，一个消息头与对应的一个消息数据可组合为一条消息；将得到的消息组合成相应的ADB命令，其中，一条ADB命令包含至少一条消息，属于同一命令的消息拥有相同的ID号，所述ID号位于所述消息头内；所述检测模块还用于保存预设的恶意命令语句，将获取的ADB命令与该预设的恶意命令语句进行比对，判断是否产生恶意行为。

7.如权利要求6所述的监控装置，其特征在于，所述监控模块用于按照预设的时间对每个线程进行监控寻找相应的读文件操作，若在该时间段内没有找到，则立即监控下一个线程。

8.如权利要求6所述的监控装置，其特征在于，所述监控模块逐个对每个线程进行监控，如果当前线程执行读文件操作，并且第一个参数指向的是USB设备文件，则获取当前线程的线程号。

9.如权利要求6所述的监控装置，其特征在于，当终端没有连接USB或者查找读取USB设备文件的线程失败后，所述守护模块会再次获取全部线程号，并查找是否有读取USB设备文件的线程。

10.如权利要求6所述的监控装置，其特征在于，若判断出ADB命令会产生恶意行为，所述检测模块按照预设的规则修改当前ADB命令，阻止恶意行为发生或/和产生用户告警。

一种用于监控PC端操作移动设备的方法及装置

技术领域

[0001] 本发明涉及计算机技术领域，尤其涉及一种用于监控PC端操作移动设备的方法及装置。

背景技术

[0002] 目前，用户经常将PC端与移动终端相连以实现PC端对移动终端控制，如应用的安装/卸载，系统root等。如果PC端被感染了恶意代码，则当PC端的恶意代码检测到移动终端被连入PC端时，恶意代码会自动连接手机等移动终端并进行一些恶意操作，如获取root权限、卸载移动终端里的应用、自动安装恶意应用、获取移动终端的文件和信息等。并且，PC端一般是通过USB(通用串行总线)访问移动终端的，对于采用Android系统的移动终端，其本质是使用ADB(Android Debug Bridge，Android调试桥接器)命令行工具的相应命令。ADB命令行工具是由3个部分组成，下面对其做些介绍：

(1) ADB客户端，PC端中运行的命令行工具，安装应用、获取文件等操作均是通过该命令行工具提供。

[0003] (2)ADB服务端，PC端中运行的服务进程，管理PC端与手机直接的连接和数据交互。ADB客户端发起的操作首先发送给ADB服务端，然后由ADB服务端发送到手机。

[0004] (3)ADB手机端进程，在Android系统中是一个名为adbd的进程，接收并执行ADB服务端发送的指令。该进程通过读写Android系统中的USB设备文件来与ADB服务端交互，adbd进程会开启2个线程来操作USB设备文件，分别用于读和写。

[0005] adbd进程在与ADB服务端交互时，有特定的消息格式，Android的源码中有文档进行描述。ADB命令会包含一系列的消息，每条消息又包含了消息头和消息数据2部分，这一系列的消息由同一个ID号进行标识(ID号位于消息头内)，并且通常由OPEN开头，CLOSE结尾。消息头格式见图1，其中每个字段为4字节大小。Command为命令的标识。Arg0，Arg1为消息命令的参数，Arg0即为本文提到的ID号。每个OPEN命令的消息头会有一个新的Arg0值。Data_length为消息头后面消息数据的长度。

[0006] 常用的ADB命令操作有：

(1)安装软件，adb.exe install [options] abc.apk

(2)卸载软件，adb.exe uninstall packageName

(3)文件传输，包含adb.exe push和adb.exe pull命令，push是将本地文件传输到手机中指定路径，pull是将手机里文件拷贝到本地。

[0007] (4)命令执行，adb.exe shell cmd [options]。Android系统是基于Linux系统的，adb通过shell参数可以执行一些Linux命令。如删除文件adb shell rm filepath，卸载软件adb shell pm uninstall packageName，获取系统属性adb shell getprop等。

[0008] 虽然移动终端现有的杀毒软件和安全防护软件可以扫描和分析应用程序，但无法分析ADB命令的操作，所以如果移动终端被连入感染了恶意代码的PC端，则可能导致移动终端被安装恶意应用、信息泄露。本方案可在移动终端上监控ADB命令的操作，若发现恶意安

装和恶意操作的行为则可进行阻止和报警，保障终端安全。

[0009]

发明内容

[0010] 本发明的目的在于提供一种用于监控PC端操作移动设备的监控方法及装置，其能帮助用户直接在移动终端监控PC端对移动终端的操作，该监控不受PC端环境影响，使用方便，监控效果好。

[0011] 为了实现上述目的，本发明公开了一种用于监控PC端操作移动设备的监控方法，包括以下步骤：

移动终端获取adbd进程及所述adbd进程中所有线程的线程号；

根据获得的所有线程号分别监控每个线程，找出其中读取USB设备文件的线程并获取当前读线程的线程号；

根据当前读线程的线程号监控线程中的读文件操作，获取读文件操作的参数内容，所述参数内容包括消息头或消息数据；

当监控到的第一个读文件操作的参数内容为消息数据时，则保存获取的所有参数内容；

将保存的所有参数内容组合成相应的消息，其中，一个消息头与对应的一个消息数据可组合为一条消息；

将得到的消息组合成相应的ADB命令，其中，一条ADB命令包含至少一条消息，属于同一命令的消息拥有相同的ID号，所述ID号位于所述消息头内；

按照预设的规则判断获取的ADB命令是否会产生恶意行为。

[0012] 进一步的，按照预设的时间对每个线程进行监控寻找相应的读USB设备文件的操作，若在该时间段内没有找到，则立即监控下一个线程。

[0013] 进一步的，找出全部线程号后，逐个对每个线程进行监控，如果当前线程执行读文件操作，并且第一个参数指向的是USB设备文件，则获取当前线程的线程号。

[0014] 进一步的，当终端没有连接USB或者查找读取USB设备文件的线程失败后，会再次获取全部线程号，并查找是否有读取USB设备文件的线程。

[0015] 进一步的，若判断出ADB命令会产生恶意行为，则修改当前ADB命令，阻值恶意行为发生。

[0016] 进一步的，若判断出ADB命令会产生恶意行为，则产生用户告警。

[0017] 为了实现上述目的，本发明还公开了一种用于监控PC端操作移动设备的监控装置，用于移动终端中，所述监控装置包括守护模块、监控模块、检测模块，其中：

当PC端访问移动终端时，所述守护模块用于获取adbd进程及所述adbd进程中所有线程的线程号；根据获得的所有线程号分别监控每个线程，找出其中读取USB设备文件的线程并获取当前读线程的线程号发送给所述监控模块；

所述监控模块用于根据当前读线程的线程号监控读线程中的读文件操作，获取读文件操作的参数内容，所述参数内容包括消息头或消息数据，当判断监控到的第一个读文件操作的参数内容为消息数据时，则保存获取的所有参数内容；

所述检测模块用于将保存的所有参数内容组合成相应的消息，其中，一个消息头与对

应的一个消息数据可组合为一条消息;将得到的消息组合成相应的ADB命令,其中,一条ADB命令包含至少一条消息,属于同一命令的消息拥有相同的ID号,所述ID号位于所述消息头内;所述检测模块还用于保存预设的恶意命令语句,将获取的ADB命令与该预设的恶意命令语句进行比对,判断是否会产生恶意行为。

[0018] 进一步的,所述监控模块用于按照预设的时间对每个线程进行监控寻找相应的读USB设备文件的操作,若在该时间段内没有找到,则立即监控下一个线程。

[0019] 进一步的,当终端没有连接USB或者查找读取USB设备文件的线程失败后,所述守护模块会再次获取全部线程号,并查找是否有读取USB设备文件的线程。

[0020] 进一步的,在找出全部线程号后,所述监控模块逐个对每个线程进行监控,如果当前线程执行读文件操作,并且第一个参数指向的是USB设备文件,则获取当前线程的线程号。若没有找到读USB设备文件的线程号,则再次获取全部线程号,并继续扫描线程。

[0021] 进一步的,若判断出ADB命令会产生恶意行为,所述检测模块修改当前ADB命令,阻值恶意行为发生。

[0022] 进一步的,若判断出ADB命令会产生恶意行为,所述检测模块产生用户告警。

[0023] 本发明与现有技术相比的有益效果是:本发明利用了PC端访问移动终端时使用ADB命令与Android系统内的adbd进程进行交互的特点,在移动设备端监控adbd进程所有线程中的读USB设备文件的操作,以获取该操作的参数内容,然后将参数内容组合成消息,将消息组合成ADB命令,以获取PC端访问移动终端时发送的命令,从而达到监控的目的。该监控不受PC端环境影响,使用方便,监控效果好。

[0024]

附图说明

[0025] 图1为ADB命令消息头的格式示意图。

[0026] 图2为本发明一种用于监控PC端操作移动设备的监控方法的流程图。

[0027] 图3为本发明一种用于监控PC端操作移动设备的监控装置的结构示意图。

[0028]

具体实施方式

[0029] 为了使本发明的目的、技术方案和优点更加清楚,下面将结合附图对本发明作进一步地详细描述。

[0030] 本发明中的步骤虽然用标号进行了排列,但并不用于限定步骤的先后次序,除非明确说明了步骤的次序或者某步骤的执行需要其他步骤作为基础,否则步骤的相对次序是可以调整的。

[0031] 本发明利用了PC端访问移动终端时使用ADB命令与Android系统内的adbd进程进行交互的特点,在一些实施例中,如图2所示,一种用于监控PC端操作移动设备的监控方法包括以下步骤:

S01,在移动终端中获取adbd进程及所述adbd进程中所有线程的线程号。

[0032] 可直接使用系统提供的ps命令获取adbd进程的全部线程号,也可通过查看/proc文件系统。Linux中每个进程在/proc下有一个对应的目录/proc/[进程号],该目录下的

cmdline文件保存了命令名字，task子目录下记录了进程的全部线程号(后文记为PID号)。

[0033] S02，根据获得的所有线程号分别监控每个线程，找出其中读取USB设备文件的线程，并获取当前读线程的线程号。

[0034] 找出全部线程号后，需要判断其中读取USB设备文件的线程。使用ptrace系统调用(其作用是允许一个进程来跟踪和控制另外一个进程)，逐个对每个线程进行监控。可利用ptrace的PTRACE_SYSCALL来监控线程的系统调用。如果当前线程执行读文件操作，并且第一个参数指向的是USB设备文件，则获取当前线程的PID号。也可以利用strace工具直接查看每个线程进行的系统调用中是否有读取USB设备文件的操作。

[0035] 不同版本Android系统的USB设备文件路径会有差异，具体需要参考源码，例如在Android 4.4版本默认的USB设备文件是/dev/android_adb。

[0036] 优选的，对每个线程的判断可以持续一小段时间，按照预设的时间对每个线程进行监控，寻找相应的读文件操作，若在该时间段内没有找到，则表示当前线程不是需要监控的线程，立即监控下一个线程。

[0037] 可以理解的，由于存在adbd进程的重启可能，因此也需要监控adbd进程的重新启动，每次重新启动后都需要再次获取相应线程的PID号。当终端没有连接USB或者查找读取USB设备文件的线程失败后，会再次获取全部线程号，并查找是否有读取USB设备文件的线程。

[0038] S03，根据当前读线程的线程号监控读线程中的读文件操作，获取读文件操作的参数内容，所述参数内容包括消息头或消息数据。

[0039] 获取当前读线程的PID号后，可使用ptrace中方法来获取读文件系统调用时的参数，读文件方法仅有3个参数，传递时是通过寄存器传递，因此使用ptrace的PTRACE_GETREGS获取寄存器值即可得到参数。其中，数据存储的参数为一个内存地址，读文件操作结束时，读取的内容保存在该内存地址指向的内存中，可以使用ptrace的PTRACE_PEEKTEXT方法将地址内的数据拷贝出来。

[0040] S04，当监控到的第一个读文件操作的参数内容为消息数据时，则保存获取的所有参数内容。

[0041] 每一条消息，通常包含消息头和消息数据两部分，因此一次监控读文件操作获取到的可能是消息头也可能是消息数据。首先判断拷贝出来的参数内容是否为消息内容，按照ADB中的消息格式描述，消息内容里有magic字段，通过检查该字段来判断是否为消息头。如果为消息头，则下个读文件操作获取到的就是消息数据，消息数据的长度保存在消息头中。若开始获取到的不是消息头，则丢弃当前消息数据，下一次读文件操作获取到的就必然是消息头。

[0042] S05，将保存的所有参数内容组合成相应的消息。

[0043] 可以理解的，一条消息的消息头和消息数据与连续两次读文件操作的参数相对应，因此把两次读文件操作获取参数的内容合并即可得到一条消息。

[0044] S06，将得到的消息组合成相应的ADB命令，其中，一条ADB命令包含至少一条消息，属于同一命令的消息拥有相同的ID号，所述ID号位于所述消息头内。

[0045] 本领域普通技术人员可以理解的，一条ADB客户端执行的命令是转换成多条消息发送给adbd进程的。例如文件的push操作，在读文件获取的消息中包含了(OPEN, sync) --

> (WRTE, STAT) --> (WRTE, filepath) --> (WRTE, SEND) --> WRTE(filepath + content) --> (WRTE, QUIT) --> (CLOSE),这里用包括2个元素内容的元组来表示一个消息,第一部分为消息头的命令,第二部分为消息数据,如(OPEN, sync)表示要开始进行sync操作,(WRTE, SEND)表示会进行数据发送;因此通过这一系列的消息可以判断出为一个push文件操作,通过filepath可以知道文件存放的地方。其他ADB客户端的命令也是类似的。每条ADB客户端命令所对应的一系列消息拥有同一个ID号,因此可将拥有相同ID号的消息组合成相应的ADB命令。

[0046] S07,按照预设的规则判断获取的ADB命令是否会产生恶意行为。

[0047] 移动终端内保存有预设的恶意命令语句,将组合出的ADB命令与预设的恶意命令语句进行比对,若组合出的ADB命令为预设的恶意命令语句,则判断获取的ADB命令有恶意行为。

[0048] 本发明利用了PC端访问移动终端时使用ADB命令与Android系统内的adbd进程进行交互的特点,在移动设备端监控adbd进程所有线程中的读文件操作,以获取读文件操作的参数内容,然后将参数内容组合成消息,将消息组合成ADB命令,以获取PC端访问移动终端时发送的命令,从而达到监控的目的。该监控不受PC端环境影响,使用方便,监控效果好。

[0049] 优选的,若判断出ADB命令会产生恶意行为,如文件拷贝、应用安装和卸载、端口开启等,则本发明还对ADB命令进行相应处理以避免造成用户损失。

[0050] 下面介绍几种常见的恶意命令语句及针对该命令的较优解决办法:

(1)样本安装命令,在ADB服务端该命令分解为下面两步进行,首先是文件传输(adb push filename.apk /data/local/tmp/filename.apk),然后是执行安装命令(adb shell pm install /data/local/tmp/filename.apk)。可通过扫描文件传输或者是执行安装时指定的文件路径来判断样本安装命令的恶意性。

[0051] 若判断出有样本安装命令,可直接修改命令内容,例如将命令中的包名替换为空字符串,从而使得安装无效。

[0052] (2)文件的传输命令,包括传入移动终端命令(adb push)和传出移动终端命令(adb pull)。push操作可通过对传输文件进行扫描来判断,pull操作需要判断传输文件是否为敏感文件(如联系人数据库)。若传输文件为敏感文件,可通过将文件路径设为空串等方式使命令无效。

[0053] (3)shell命令的执行,即adb shell方式执行的操作。如adb shell getprop获取系统属性,adb shell am命令可以发送广播,adb shell pm命令可以卸载应用。需依据具体命令来判断恶意性,例如卸载的是否为关键应用可以通过pm命令中指定的包名来判断;如果命令为删除文件则可判断文件路径是否为系统文件路径。恶意命令可以通过修改命令内容的方式进行处理。

[0054] 上述三类恶意命令的处理方法里均提到可通过修改命令内容进行处理,具体的修改方式描述如下:命令的内容是保存在消息数据读文件操作的参数指向的内存中,由于已经保存读文件操作的参数,因此可以使用ptrace的PTRACE_POKETEXT来修改命令所在内存中的内容,也可以使用ptrace的PTRACE_SETREGS修改读文件操作的参数为空指针,使参数值无效。

[0055] 优选的,若判断出ADB命令会产生恶意行为,则产生用户告警。

[0056] 具体的，可以直接产生用户告警或者将该ADB命令的内容通过进程通信的方式传给常用的社交类或游戏类应用，该类应用可以以弹窗警告的方式进行用户告警。

[0057] 如图3所示，本发明还公开了一种用于监控PC端操作移动设备的监控装置，用于移动终端中，所述监控装置包括守护模块10、监控模块20、检测模块30，其中：

当PC端访问移动终端时，所述守护模块10用于获取adbd进程及所述adbd进程中所有线程的线程号；根据获得的所有线程号分别监控每个线程，找出其中读取USB设备文件的线程并获取当前读线程的线程号发送给所述监控模块20。

[0058] 可直接使用系统提供的ps命令获取adbd进程的全部线程号，也可通过查看/proc文件系统。Linux中每个进程在/proc下有一个对应的目录/proc/[进程号]，该目录下的cmdline文件保存了命令名字，task子目录下记录了进程的全部线程的PID号。

[0059] 找出全部PID号后，使用ptrace系统调用(其作用是允许一个进程来跟踪和控制另外一个进程)，逐个对每个线程进行监控，可利用ptrace的PTRACE_SYSCALL来监控线程的系统调用。如果当前线程调用系统的读文件操作，并且第一个参数指向的是USB设备文件，则获取当前线程的PID号。也可以利用strace工具直接查看每个线程进行的系统调用中是否有读取USB设备文件的操作。

[0060] 优选的，对每个线程的判断可以持续一小段时间，按照预设的时间对每个线程进行监控，寻找相应的读文件操作，若在该时间段内没有找到，则表示当前线程不是需要监控的线程，立即监控下一个线程。

[0061] 可以理解的，由于存在adbd进程的重启可能，因此也需要监控adbd进程的重新启动，每次重新启动后都需要再次获取相应线程的PID号。当终端没有连接USB或者查找读取USB设备文件的线程失败后，会再次获取全部线程号，并查找是否有读取USB设备文件的线程。

[0062] 所述监控模块20用于根据当前读线程的线程号监控读线程中的读文件操作，获取读文件操作的参数内容，并判断监控到的第一个读文件操作的参数内容为消息头或消息数据，若为消息数据则保存获取的所有参数内容，所述参数内容包括消息头或消息数据。

[0063] 获取当前读线程的PID号后，可使用ptrace中方法来获取系统调用时的参数，读文件操作的系统调用方法仅有3个参数，传递时是通过寄存器传递，因此使用ptrace的PTRACE_GETREGS获取寄存器值即可得到参数。用于保存数据存储位置的参数为一个内存地址，可以使用ptrace的PTRACE_PEEKTEXT方法将地址内的数据拷贝出来。

[0064] 每一条消息，通常包含消息头和消息数据两部分，因此一次监控读文件操作获取到的可能是消息头也可能是消息数据。首先判断拷贝出来的参数内容是否为消息内容，按照ADB中的消息格式描述，消息内容里有magic字段，通过检查该字段来判断是否为消息头。如果为消息头，则下个读文件操作获取到的就是消息数据，消息数据的长度保存在消息头中。若开始获取到的不是消息头，则丢弃当前消息数据，下一次读文件操作获取到的就必然是消息头。

[0065] 所述检测模块30用于将保存的所有参数内容组合成相应的消息，其中，一个消息头与对应的一个消息数据可组合为一条消息；将得到的消息组合成相应的ADB命令，其中，一条ADB命令包含至少一条消息，属于同一命令的消息拥有相同的ID号，所述ID号位于所述消息头内；所述检测模块30还用于保存预设的恶意命令语句，将获取的ADB命令与该预设的

恶意命令语句进行比对，判断是否会产生恶意行为。

[0066] 本领域普通技术人员可以理解的，一条消息的消息头和消息数据与连续两次读文件操作的参数相对应，因此把两次读文件操作的参数的内容合并即可得到一条消息。一条ADB客户端执行的命令是转换成多条消息发送给adbd进程的。例如文件的push操作，在读USB设备文件的线程中包含了(OPEN, sync) --> (WRTE, STAT) --> (WRTE, filepath) --> (WRTE, SEND) --> WRTE(filepath + content) --> (WRTE, QUIT) --> (CLOSE)，这里用包括2个元素内容的元组来表示一个消息，第一部分为消息头的命令，第二部分为消息数据，如(OPEN, sync)表示要开始进行sync操作，(WRTE, SEND)表示会进行数据发送；因此通过这一系列的消息可以判断出为一个push文件操作，通过filepath可以知道文件存放的地方。其他ADB客户端的命令也是类似的。每条ADB客户端命令所对应的一系列消息拥有同一个ID号，因此可将拥有相同ID号的消息组合成相应的ADB命令。

[0067] 移动终端内保存有预设的恶意命令语句，将组合出的ADB命令与预设的恶意命令语句进行比对，若组合出的ADB命令为预设的恶意命令语句则判断获取的ADB命令有恶意行为。

[0068] 本发明利用了PC端访问移动终端时使用ADB命令与Android系统内的adbd进程进行交互的特点，在移动设备端监控adbd进程中读取USB设备文件线程中的读文件操作，以获取读文件操作的参数内容，然后将参数内容组合成消息，将消息组合成ADB命令，以获取PC端访问移动终端时发送的命令，从而达到监控的目的。该监控不受PC端环境影响，使用方便，监控效果好。

[0069] 优选的，所述检测模块30还用于当判断出ADB命令有恶意行为时，对该ADB命令进行相应处理以避免造成用户损失。

[0070] 下面介绍几种常见的恶意命令语句及针对该命令的较优解决办法：

(1)样本安装命令，在ADB服务端该命令分解为下面两步进行，首先是文件传输(adb push filename.apk /data/local/tmp/filename.apk)，然后是执行安装命令(adb shell pm install /data/local/tmp/filename.apk)。可通过扫描文件传输或者是执行安装时指定的文件路径来判断样本安装命令的恶意性。

[0071] 若判断出有样本安装命令，可直接修改命令内容，例如将命令中的包名替换为空字符串，从而使得安装无效。

[0072] (2)文件的传输命令，包括传入移动终端命令(adb push)和传出移动终端命令(adb pull)。push操作可通过对传输文件进行扫描来判断，pull操作需要判断传输文件是否为敏感文件(如联系人数据库)。若传输文件为敏感文件，可通过将文件路径设为空串等方式使命令无效。

[0073] (3)shell命令的执行，即adb shell方式执行的操作。如adb shell getprop获取系统属性，adb shell am命令可以发送广播，adb shell pm命令可以卸载应用。需依据具体命令来判断恶意性，例如卸载的是否为关键应用可以通过pm命令中指定的包名来判断；如果命令为删除文件则可判断文件路径是否为系统文件路径。恶意命令可以通过修改命令内容的方式进行处理。

[0074] 上述三类恶意命令的处理方法里均提到可通过修改命令内容进行处理，具体的修改方式描述如下：命令的内容是保存在读文件操作的参数指向的内存中，由于已经保存读

文件操作的参数，因此可以使用ptrace的PTRACE_POKETEXT来修改命令所在内存中的内容，也可以使用ptrace的PTRACE_SETREGS修改读文件操作的参数为空指针，使参数值无效。

[0075] 所述检测模块30还用于产生用户告警或者将该ADB命令的内容通过进程通信的方式传常用的社交类或游戏类应用，该类应用可以以弹窗警告的方式进行用户告警。

[0076] 上述说明示出并描述了本发明的若干实施例，但如前所述，应当理解本发明并非局限于本文所披露的形式，不应看作是对其他实施例的排除，而可用于各种其他组合、修改和环境，并能够在本文所述发明构想范围内，通过上述教导或相关领域的技术或知识进行改动。而本领域人员所进行的改动和变化不脱离本发明的精神和范围，则都应在本发明所附权利要求的保护范围内。

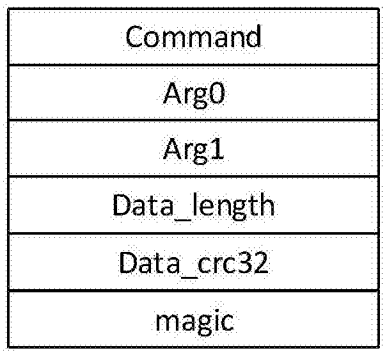

Command
Arg0
Arg1
Data_length
Data_crc32
magic

图1

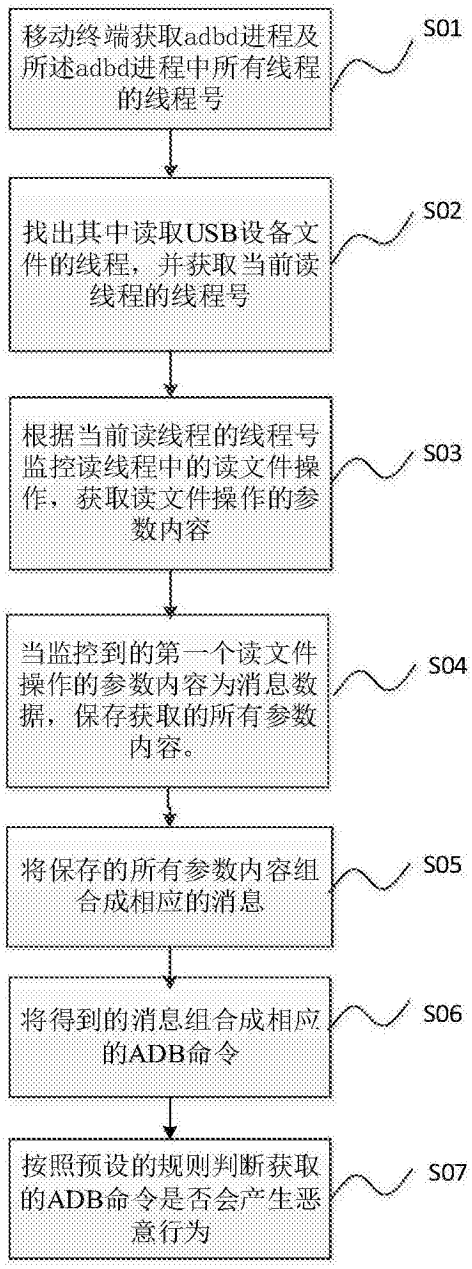

移动终端获取adbd进程及所述adbd进程中所有线程的线程号
S01
找出其中读取USB设备文件的线程，并获取当前读线程的线程号
S02
根据当前读线程的线程号监控读线程中的读文件操作，获取读文件操作的参数内容
S03
当监控到的第一个读文件操作的参数内容为消息数据，保存获取的所有参数内容。
S04
将保存的所有参数内容组合成相应的消息
S05
将得到的消息组合成相应的ADB命令
S06
按照预设的规则判断获取的ADB命令是否会产生恶意行为
S07

图2

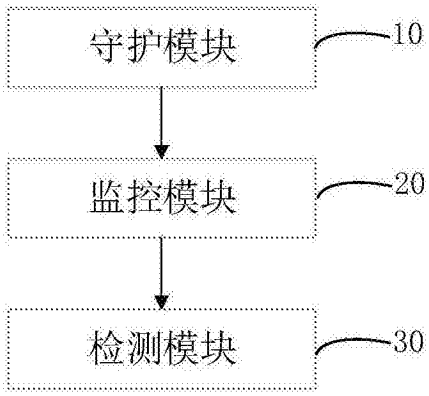


图3