



(12)发明专利申请

(10)申请公布号 CN 109740342 A  
(43)申请公布日 2019.05.10

(21)申请号 201811592730.X

(22)申请日 2018.12.20

(71)申请人 深圳市网心科技有限公司  
地址 518063 广东省深圳市南山区高新南  
九道9号威新软件园5号楼5层

(72)发明人 唐成

(74)专利代理机构 深圳市赛恩倍吉知识产权代  
理有限公司 44334  
代理人 杨毅玲

(51) Int. Cl.  
G06F 21/52(2013.01)  
G06F 21/12(2013.01)

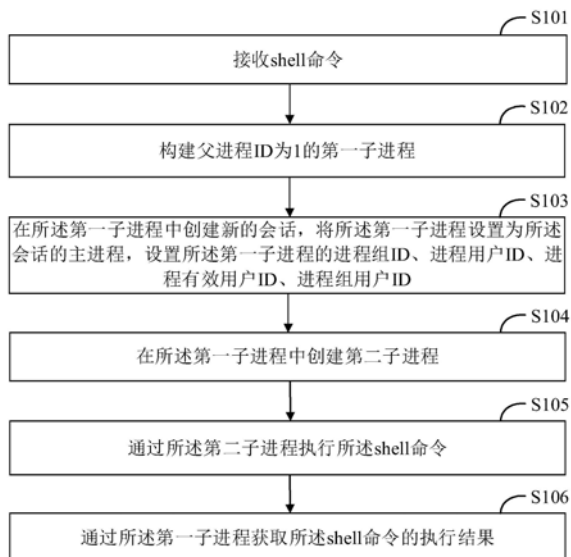
权利要求书2页 说明书12页 附图2页

(54)发明名称

获取shell执行权限的方法、系统、计算机装置及存储介质

(57)摘要

本发明提供一种获取shell执行权限的方法,包括:接收shell命令;构建父进程ID为1的第一子进程;在所述第一子进程中创建新的会话,将所述第一子进程设置为所述会话的主进程,设置所述第一子进程的进程组ID、进程用户ID、进程有效用户ID、进程组用户ID;在所述第一子进程中创建第二子进程;通过所述第二子进程执行所述shell命令;通过所述第一子进程获取所述shell命令的执行结果。本发明还提供一种获取shell执行权限的系统、计算机装置及计算机可读存储介质。本发明可以方便、安全、可靠地获取shell执行权限。



1. 一种获取shell执行权限的方法,其特征在于,所述方法包括:  
接收shell命令;  
构建父进程ID为1的第一子进程;  
在所述第一子进程中创建新的会话,将所述第一子进程设置为所述会话的主进程,设置所述第一子进程的进程组ID、进程用户ID、进程有效用户ID、进程组用户ID;  
在所述第一子进程中创建第二子进程;  
通过所述第二子进程执行所述shell命令;  
通过所述第一子进程获取所述shell命令的执行结果。
2. 如权利要求1所述的方法,其特征在于,所述接收shell命令包括:  
通过套接字建立与socket客户端的数据通讯连接;  
接收所述socket客户端通过所述数据通讯连接发送的所述shell命令。
3. 如权利要求1所述的方法,其特征在于,所述构建父进程ID为1的第一子进程包括:  
通过调用fork函数构建父进程ID为1的第一子进程;  
所述在所述第一子进程中创建第二子进程包括:  
通过调用fork函数在所述第一子进程中创建所述第二子进程。
4. 如权利要求1所述的方法,其特征在于,所述通过所述第二子进程执行所述shell命令包括:  
通过调用execl函数执行所述shell命令。
5. 如权利要求1所述的方法,其特征在于,所述通过所述第一子进程获取所述shell命令的执行结果包括:  
在第一子进程中创建一个管道,通过所述管道获取所述shell命令的执行结果。
6. 如权利要求1所述的方法,所述设置所述第一子进程的进程组ID、进程用户ID、进程有效用户ID、进程组用户ID包括:  
将所述第一子进程的进程组ID、进程用户ID、进程有效用户ID、进程组用户ID均设置为2000。
7. 如权利要求1-6中任一项所述的方法,其特征在于,所述方法还包括:  
通过所述第一子进程处理所述shell命令的执行结果。
8. 一种获取shell执行权限的系统,其特征在于,所述系统包括:  
接收单元,用于接收shell命令;  
提权单元,用于构建父进程ID为1的第一子进程,在所述第一子进程中创建新的会话,将所述第一子进程设置为所述会话的主进程,设置所述第一子进程的进程组ID、进程用户ID、进程有效用户ID、进程组用户ID;  
执行单元,用于在所述第一子进程中创建第二子进程,通过所述第二子进程执行所述shell命令,通过所述第一子进程获取所述shell命令的执行结果。
9. 一种计算机装置,其特征在于,所述计算机装置包括存储器和处理器,所述存储器上存储有可在所述处理器上运行的计算机程序,所述计算机程序被所述处理器执行时实现如权利要求1-7中任一项所述获取shell执行权限的方法。
10. 一种计算机可读存储介质,其特征在于,所述计算机可读存储介质上存储有计算机程序,所述计算机程序被处理器执行时实现如权利要求1-7中任一项所述获取shell执行权

限的方法。

## 获取shell执行权限的方法、系统、计算机装置及存储介质

### 技术领域

[0001] 本发明涉及计算机装置权限管控技术领域,具体涉及一种获取shell执行权限的方法、系统、计算机装置和计算机可读存储介质。

### 背景技术

[0002] 原有的针对android系统的获取shell执行权限的方法一般通过app调用jni (Java Native Interface)的方式来实现,且均需要对系统进行root,获取系统的root权限,这种方式无疑是最直接的方式,但是却存在着如下问题:

[0003] (1)通常为了获取android手机的root权限,由于ROM具备防root的功能,需要通过刷机的方式来实现,比较耗时耗精力;

[0004] (2)被root的android系统由于开放了root权限,本身也变得不安全,容易被流氓软件攻击;

[0005] (3)被root的android系统无法再继续跟随厂家的版本持续升级,无法及时使用新的功能,修复已有bug,影响使用体验。

### 发明内容

[0006] 鉴于以上内容,有必要提出一种获取shell执行权限的方法、系统、计算机装置、计算机可读存储介质和计算机程序产品,其可以方便、安全、可靠地获取shell执行权限。

[0007] 本申请的第一方面提供一种获取shell执行权限的方法,所述方法包括:

[0008] 接收shell命令;

[0009] 构建父进程ID为1的第一子进程;

[0010] 在所述第一子进程中创建新的会话,将所述第一子进程设置为所述会话的主进程,设置所述第一子进程的进程组ID、进程用户ID、进程有效用户ID、进程组用户ID;

[0011] 在所述第一子进程中创建第二子进程;

[0012] 通过所述第二子进程执行所述shell命令;

[0013] 通过所述第一子进程获取所述shell命令的执行结果。

[0014] 另一种可能的实现方式中,若对所述点播分发系统进行单一文件压力测试,则所述压测文件为单一文件;

[0015] 若对所述点播分发系统进行固定文件列表压力测试,则所述压测文件为固定文件列表中的文件;

[0016] 若对所述点播分发系统进行动态更新文件列表压力测试,则所述压测文件为动态文件列表中的文件。

[0017] 另一种可能的实现方式中,所述接收shell命令包括:

[0018] 通过套接字建立与socket客户端的数据通讯连接;

[0019] 接收所述socket客户端通过所述数据通讯连接发送的所述shell命令。

[0020] 另一种可能的实现方式中,所述构建父进程ID为1的第一子进程包括:

- [0021] 通过调用fork函数构建父进程ID为1的第一子进程；
- [0022] 所述在所述第一子进程中创建第二子进程包括：
- [0023] 通过调用fork函数在所述第一子进程中创建所述第二子进程。
- [0024] 另一种可能的实现方式中，所述通过所述第二子进程执行所述shell命令包括：
- [0025] 通过调用execl函数执行所述shell命令。
- [0026] 另一种可能的实现方式中，所述通过所述第一子进程获取所述shell命令的执行结果包括：
- [0027] 在第一子进程中创建一个管道，通过所述管道获取所述shell命令的执行结果。
- [0028] 另一种可能的实现方式中，所述设置所述第一子进程的进程组ID、进程用户ID、进程有效用户ID、进程组用户ID包括：
- [0029] 将所述第一子进程的进程组ID、进程用户ID、进程有效用户ID、进程组用户ID均设置为2000。
- [0030] 另一种可能的实现方式中，所述方法还包括：
- [0031] 通过所述第一子进程处理所述shell命令的执行结果。
- [0032] 本申请的第二方面提供一种获取shell执行权限的系统，所述系统包括：
- [0033] 接收单元，用于接收shell命令；
- [0034] 提权单元，用于构建父进程ID为1的第一子进程，在所述第一子进程中创建新的会话，将所述第一子进程设置为所述会话的主进程，设置所述第一子进程的进程组ID、进程用户ID、进程有效用户ID、进程组用户ID；
- [0035] 执行单元，用于在所述第一子进程中创建第二子进程，通过所述第二子进程执行所述shell命令，通过所述第一子进程获取所述shell命令的执行结果。
- [0036] 本申请的第三方面提供一种计算机装置，所述计算机装置包括存储器和处理器，所述存储器上存储有可在所述处理器上运行的计算机程序，所述计算机程序被所述处理器执行时实现所述获取shell执行权限的方法。
- [0037] 本申请的第四方面提供一种计算机可读存储介质，所述计算机可读存储介质上存储有计算机程序，所述计算机程序被处理器执行时实现所述获取shell执行权限的方法。
- [0038] 本申请的第五方面提供一种计算机程序产品，包括计算机指令，当其在计算机装置上运行时，使得计算机装置执行所述获取shell执行权限的方法。
- [0039] 本发明通过套接字建立与socket客户端的数据通讯连接；接收所述socket客户端通过所述数据通讯连接发送的shell命令；构建父进程ID为1的第一子进程；在所述第一子进程中创建新的会话，将所述第一子进程设置为所述会话的主进程，设置所述第一子进程的进程组ID、进程用户ID、进程有效用户ID、进程组用户ID；在所述第一子进程中创建第二子进程；通过所述第二子进程执行所述shell命令；通过所述第一子进程获取所述shell命令的执行结果。本发明的以独立运行的本地可执行程序的形式运行提权服务，类似于守护进程，不受其他模块的控制，减少了提权服务运行权限的困扰；并且，本发明的提权过程中不需要获取root权限，也不需要重新刷机，提权过程采用创建具有更高权限的子进程的方式来执行，安全可靠。因此，本发明实现了方便、安全、可靠地获取shell执行权限。

## 附图说明

[0040] 图1是本发明实施例提供的获取shell执行权限的方法较佳实施例的流程图。

[0041] 图2是本发明另一实施例提供的获取shell执行权限的系统较佳实施例的流程图。

[0042] 图3是本发明实施例提供的计算机装置较佳实施例的示意图。

## 具体实施方式

[0043] 为了使本发明的目的、技术方案及优点更加清楚明白,以下结合附图及实施例,对本发明进行进一步详细说明。应当理解,此处所描述的具体实施例仅用以解释本发明,并不用于限定本发明。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0044] 本申请的说明书和权利要求书及上述附图中的术语“第一”、“第二”、“第三”、“第四”等(如果存在)是用于区别类似的对象,而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换,以便这里描述的实施例能够以除了在这里图示或描述的内容以外的顺序实施。此外,术语“包括”和“具有”以及他们的任何变形,意图在于覆盖不排他的包含,例如,包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元,而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

[0045] 需要说明的是,在本发明中涉及“第一”、“第二”等的描述仅用于描述目的,而不能理解为指示或暗示其相对重要性或者隐含指明所指示的技术特征的数量。由此,限定有“第一”、“第二”的特征可以明示或者隐含地包括至少一个该特征。另外,各个实施例之间的技术方案可以相互结合,但是必须是以本领域普通技术人员能够实现为基础,当技术方案的结合出现相互矛盾或无法实现时应当认为这种技术方案的结合不存在,也不在本发明要求的保护范围之内。

[0046] 优选地,本发明的获取shell执行权限的方法应用在一个或者多个计算机装置中。所述计算机装置是一种能够按照事先设定或存储的指令,自动进行数值计算和/或信息处理的设备,其硬件包括但不限于微处理器、专用集成电路(Application Specific Integrated Circuit,ASIC)、可编程门阵列(Field-Programmable Gate Array,FPGA)、数字处理器(Digital Signal Processor,DSP)、嵌入式设备等。

[0047] 所述计算机装置可以是PC(Personal Computer,个人电脑),也可以是智能手机、平板电脑、掌上电脑、便携计算机、智能路由器、矿机、网络存储设备等终端设备。所述计算机装置可以与用户通过键盘、鼠标、遥控器、触摸板或声控设备等方式进行人机交互。

[0048] 实施例一

[0049] 图1是本发明实施例提供的获取shell执行权限的方法的流程图。所述获取shell执行权限的方法应用于计算机装置中,用于获取计算机装置的shell执行权限。

[0050] 参阅1所示,所述获取shell执行权限的方法具体包括以下步骤:

[0051] S101,接收shell命令。

[0052] shell命令用于访问操作系统内核的服务。

[0053] shell命令是字符串的形式。例如,shell命令为“am start -n com.example.hello/.MainActivity”。

[0054] shell命令可以包括文件、目录操作命令,例如ls命令,用于显示文件和目录的信息;又如cd命令,用于改名目录;再如diff命令,用于比较文件内容。

[0055] shell命令还可以包括查看文件内容命令,例如cat命令,用于显示文件的内容;又如tail命令,用于显示文件的最后几行;再如touch命令,用于创建一个空文件。

[0056] shell命令还可以包括基本系统命令,例如man命令,用于查看某个命令的帮助;又如w命令,用于显示登录用户的详细信息;再如who命令,用于显示登录用户。

[0057] shell命令还可以包括监视系统状态命令、磁盘操作命令、用户和组相关命令、压缩命令、网络相关命令等。

[0058] 可以接收用户在所述计算机装置上输入的shell命令。例如,可以在所述计算机装置的屏幕上显示shell命令输入界面,接收用户从所述shell命令输入界面输入的shell命令。

[0059] 在本实施例中,接收单元201可以通过套接字(socket)建立与socket客户端的数据通讯连接,接收所述socket客户端通过所述数据通讯连接发送的shell命令。

[0060] 所述socket客户端可以包含在所述计算机装置中,例如包含在所述计算机装置的其他进程中。所述socket客户端也可以包含在与所述计算机设备连接的外部设备中,例如包含在通过网络或者USB与所述计算机设备连接的外部设备中。

[0061] 所述计算机装置包括socket服务端,socket服务端和socket客户端可以通过下述方法建立数据通讯连接:

[0062] socket服务端创建服务端套接字,为所述服务端套接字绑定IP地址和端口号,监听对所述服务端套接字的连接请求;

[0063] socket客户端创建客户端套接字,根据所述服务端套接字绑定的IP地址和端口号向所述服务端套接字提出连接请求;

[0064] socket服务端接收到所述连接请求后,建立一个新的线程,把服务端套接字的信息发送给socket客户端;

[0065] socket客户端向socket服务端返回连接信息;

[0066] socket服务端接收到所述连接信息后,建立与socket客户端的数据通讯连接。

[0067] 其中,socket服务端可以根据地址类型(IPV4、IPV6)、套接字类型(流式套接字、数据报套接字、原始套接字)、使用的传输层协议(TCP协议、UDP协议等)创建服务端套接字。类似地,socket客户端可以根据地址类型、套接字类型、使用的传输层协议创建客户端套接字。

[0068] 在本实施例中,可以使用AF\_INET域的通讯方式建立与socket客户端的数据通讯连接,这种方式服务端套接字和客户端套接字均需要绑定同样的端口号。

[0069] 在本实施例中,socket客户端将shell命令写入客户端套接字,通过客户端套接字将所述shell命令发送给socket服务端。socket服务端从服务端套接字读取所述shell命令。

[0070] S102,构建父进程ID(parent process ID,PPID)为1的第一子进程。

[0071] 每个进程在系统中都会有一个唯一的数字标识符,这个数字被称为进程ID(Process ID)。

[0072] 父进程ID即父进程的进程ID。父进程ID为1即父进程的进程ID为1。进程ID为1的进

程是由内核创建的init进程,该init进程在系统自举过程结束时由内核调用。init进程会一直存在于内存中,直到系统关机才会被销毁。也就是说,init进程具有高权限和高优先级。

[0073] 第一子进程继承父进程ID为1的init进程的权限和优先级,因此,第一子进程同样具有高权限和高优先级。

[0074] 在本实施例中,可以通过调用fork函数构建父进程ID为1的第一子进程。

[0075] S103,在所述第一子进程中创建新的会话(session),将所述第一子进程设置为所述会话的主进程,设置所述第一子进程的进程组ID(GID)、进程用户ID(UID)、进程有效用户ID(EUID)、进程组用户ID(GUID)。

[0076] 会话是一个或多个进程组的集合。会话的主进程就是会话的控制进程,在会话的所有进程中具有高权限和高优先级。

[0077] 进程组ID、进程用户ID、进程有效用户ID、进程组用户ID都是描述进程的参数,设置进程组ID、进程用户ID、进程有效用户ID、进程组用户ID是为了让进程获取shell执行权限。

[0078] 在一具体实施例中,所述第一子进程的进程组ID、进程用户ID、进程有效用户ID、进程组用户ID可以均设置为2000。

[0079] S104,在所述第一子进程中创建第二子进程。

[0080] 在指定进程中创建新的进程就是以指定进程为父进程,创建指定进程的子进程(即新的进程为指定进程的子进程)。因此,第一子进程为第二子进程的父进程,第二子进程为第一子进程的子进程。

[0081] 在本实施例中,可以通过调用fork函数在所述第一子进程中创建所述第二子进程。

[0082] S105,通过所述第二子进程执行所述shell命令。

[0083] 第一子进程具有高权限和高优先级。第二子进程为第一子进程的子进程,继承了第一子进程的权限和优先级。因此,第二子进程同样具有高权限和高优先级,具有shell执行权限。

[0084] 在本实施例中,可以通过调用execl函数执行所述shell命令。

[0085] S106,通过所述第一子进程获取所述shell命令的执行结果。

[0086] 可以在第一子进程中创建一个管道(pipe),第一子进程通过所述管道获取所述shell命令的执行结果。pipe是一种进程间通讯方式,可以实现进程之间的数据交互。

[0087] 实施例一的获取shell执行权限的方法接收shell命令;构建父进程ID为1的第一子进程;在所述第一子进程中创建新的会话,将所述第一子进程设置为所述会话的主进程,设置所述第一子进程的进程组ID、进程用户ID、进程有效用户ID、进程组用户ID;在所述第一子进程中创建第二子进程;通过所述第二子进程执行所述shell命令;通过所述第一子进程获取所述shell命令的执行结果。实施例一的提权服务以独立运行的本地可执行程序的形式运行,类似于守护进程,不受其他模块的控制,减少了提权服务运行权限的困扰;并且,实施例一的提权过程中不需要获取root权限,也不需要重新刷机,提权过程采用创建具有更高权限的子进程的方式来执行,安全可靠。因此,实施例一实现了方便、安全、可靠地获取shell执行权限。



[0088] 在另一实施例中,所述获取shell执行权限的方法还可以包括:通过所述第一子进程处理所述shell命令的执行结果。

[0089] shell命令的执行结果是指shell命令被执行后的输出值。可以根据不同shell命令的不同执行结果进行不同的处理。例如,ps命令的执行结果是当前系统中正在运行的进程信息,可以保存所述进程信息,还可以在所述进程信息中查找需要的信息。又如,ls命令的执行结果是当前文件夹下所有的文件,可以从当前文件夹下所有的文件中查找需要的文件。

[0090] 在本实施例中,可以通过调用回调函数处理所述shell命令的执行结果。例如,调用回调函数从ls命令的执行结果(即当前文件夹下所有的文件)中查找需要的文件。

[0091] 在shell命令来自于socket客户端(即S101中通过套接字建立与socket客户端的数据通讯连接,接收所述socket客户端通过所述数据通讯连接发送的shell命令)的实施例中,所述获取shell执行权限的方法还可以包括:

[0092] 将所述执行结果返回所述socket客户端;和/或

[0093] 将所述执行结果的处理结果返回所述socket客户端;和/或

[0094] 关闭与所述socket客户端的数据通讯连接。

[0095] 实施例二

[0096] 图2为本发明实施例提供的获取shell执行权限的系统较佳实施例的结构图。所述获取shell执行权限的系统应用于计算机装置中,用于获取计算机装置的shell执行权限。

[0097] 如图2所示,所述获取shell执行权限的系统10可以包括:接收单元201、提权单元202、执行单元203。

[0098] 接收单元201,用于接收shell命令。

[0099] shell命令用于访问操作系统内核的服务。

[0100] shell命令是字符串的形式。例如,shell命令为“am start -n com.example.hello/.MainActivity”。

[0101] shell命令可以包括文件、目录操作命令,例如ls命令,用于显示文件和目录的信息;又如cd命令,用于改名目录;再如diff命令,用于比较文件内容。

[0102] shell命令还可以包括查看文件内容命令,例如cat命令,用于显示文件的内容;又如tail命令,用于显示文件的最后几行;再如touch命令,用于创建一个空文件。

[0103] shell命令还可以包括基本系统命令,例如man命令,用于查看某个命令的帮助;又如w命令,用于显示登录用户的详细信息;再如who命令,用于显示登录用户。

[0104] shell命令还可以包括监视系统状态命令、磁盘操作命令、用户和组相关命令、压缩命令、网络相关命令等。

[0105] 可以接收用户在所述计算机装置上输入的shell命令。例如,可以在所述计算机装置的屏幕上显示shell命令输入界面,接收用户从所述shell命令输入界面输入的shell命令。

[0106] 在本实施例中,接收单元201可以通过套接字(socket)建立与socket客户端的数据通讯连接,接收所述socket客户端通过所述数据通讯连接发送的shell命令。

[0107] 所述socket客户端可以包含在所述计算机装置中,例如包含在所述计算机装置的其他进程中。所述socket客户端也可以包含在与所述计算机设备连接的外部设备中,例如

包含在通过网络或者USB与所述计算机设备连接的外部设备中。

[0108] 所述计算机装置包括socket服务端,socket服务端和socket客户端可以通过下述方法建立数据通讯连接:

[0109] socket服务端创建服务端套接字,为所述服务端套接字绑定IP地址和端口号,监听对所述服务端套接字的连接请求;

[0110] socket客户端创建客户端套接字,根据所述服务端套接字绑定的IP地址和端口号向所述服务端套接字提出连接请求;

[0111] socket服务端接收到所述连接请求后,建立一个新的线程,把服务端套接字的信息发送给socket客户端;

[0112] socket客户端向socket服务端返回连接信息;

[0113] socket服务端接收到所述连接信息后,建立与socket客户端的数据通讯连接。

[0114] 其中,socket服务端可以根据地址类型(IPV4、IPV6)、套接字类型(流式套接字、数据报套接字、原始套接字)、使用的传输层协议(TCP协议、UDP协议等)创建服务端套接字。类似地,socket客户端可以根据地址类型、套接字类型、使用的传输层协议创建客户端套接字。

[0115] 在本实施例中,可以使用AF\_INET域的通讯方式建立与socket客户端的数据通讯连接,这种方式服务端套接字和客户端套接字均需要绑定同样的端口号。

[0116] 在本实施例中,socket客户端将shell命令写入客户端套接字,通过客户端套接字将所述shell命令发送给socket服务端.socket服务端从服务端套接字读取所述shell命令。

[0117] 提权单元202,用于构建父进程ID(parent process ID,PPID)为1的第一子进程。

[0118] 每个进程在系统中都会有一个唯一的数字标识符,这个数字被称为进程ID(Process ID)。

[0119] 父进程ID即父进程的进程ID。父进程ID为1即父进程的进程ID为1。进程ID为1的进程是由内核创建的init进程,该init进程在系统自举过程结束时由内核调用。init进程会一直存在于内存中,直到系统关机才会被销毁。也就是说,init进程具有高权限和高优先级。

[0120] 第一子进程继承父进程ID为1的init进程的权限和优先级,因此,第一子进程同样具有高权限和高优先级。

[0121] 在本实施例中,可以通过调用fork函数构建父进程ID为1的第一子进程。

[0122] 提权单元202,还用于所述第一子进程中创建新的会话(session),将所述第一子进程设置为所述会话的主进程,设置所述第一子进程的进程组ID(GID)、进程用户ID(UID)、进程有效用户ID(EUID)、进程组用户ID(GUID)。

[0123] 会话是一个或多个进程组的集合。会话的主进程就是会话的控制进程,在会话的所有进程中具有高权限和高优先级。

[0124] 进程组ID、进程用户ID、进程有效用户ID、进程组用户ID都是描述进程的参数,设置进程组ID、进程用户ID、进程有效用户ID、进程组用户ID是为了让进程获取相应的权限。

[0125] 在一具体实施例中,所述第一子进程的进程组ID、进程用户ID、进程有效用户ID、进程组用户ID可以均设置为2000。

[0126] 执行单元203,用于在所述第一子进程中创建第二子进程。

[0127] 在指定进程中创建新的进程就是以指定进程为父进程,创建指定进程的子进程(即新的进程为指定进程的子进程)。因此,第一子进程为第二子进程的父进程,第二子进程为第一子进程的子进程。

[0128] 在本实施例中,可以通过调用fork函数在所述第一子进程中创建所述第二子进程。

[0129] 执行单元203,还用于通过所述第二子进程执行所述shell命令。

[0130] 第一子进程具有高权限和高优先级。第二子进程为第一子进程的子进程,继承了第一子进程的权限和优先级。因此,第二子进程同样具有高权限和高优先级,具有shell执行权限。

[0131] 在本实施例中,可以通过调用execl函数执行所述shell命令。

[0132] 执行单元203,还用于通过所述第一子进程获取所述shell命令的执行结果。

[0133] 可以在第一子进程中创建一个管道(pipe),第一子进程通过所述管道获取所述shell命令的执行结果。pipe是一种进程间通讯方式,可以实现进程之间的数据交互。

[0134] 实施例二的获取shell执行权限的系统10接收shell命令;构建父进程ID为1的第一子进程;在所述第一子进程中创建新的会话,将所述第一子进程设置为所述会话的主进程,设置所述第一子进程的进程组ID、进程用户ID、进程有效用户ID、进程组用户ID;在所述第一子进程中创建第二子进程;通过所述第二子进程执行所述shell命令;通过所述第一子进程获取所述shell命令的执行结果。实施例一的获取shell执行权限的系统10在不root手机,不重新刷机的情况下获取系统的shell执行权限,能正常执行shell命令。

[0135] 在另一实施例中,所述获取shell执行权限的方法还可以包括:通过所述第一子进程处理所述shell命令的执行结果。

[0136] shell命令的执行结果是指shell命令被执行后的输出值。可以根据不同shell命令的不同执行结果进行不同的处理。例如,ps命令的执行结果是当前系统中正在运行的进程信息,可以保存所述进程信息,还可以在所述进程信息中查找需要的信息。又如,ls命令的执行结果是当前文件夹下所有的文件,可以从当前文件夹下所有的文件中查找需要的文件。

[0137] 在本实施例中,可以通过调用回调函数处理所述shell命令的执行结果。例如,调用回调函数从ls命令的执行结果(即当前文件夹下所有的文件)中查找需要的文件。

[0138] 在shell命令来自于socket客户端(即接收单元201通过套接字建立与socket客户端的数据通讯连接,接收所述socket客户端通过所述数据通讯连接发送的shell命令)的实施例中,所述获取shell执行权限的系统10还可以包括:

[0139] 第一返回单元,用于将所述执行结果返回所述socket客户端;和/或

[0140] 第二返回单元,用于将所述执行结果的处理结果返回所述socket客户端;和/或

[0141] 关闭单元,用于关闭与所述socket客户端的数据通讯连接。

[0142] 实施例三

[0143] 本实施例提供一种计算机可读存储介质,该计算机可读存储介质上存储有计算机程序,该计算机程序被处理器执行时实现上述获取shell执行权限的方法实施例中的步骤,例如图1所示的S101-S106:

- [0144] S101,接收shell命令;
- [0145] S102,构建父进程ID为1的第一子进程;
- [0146] S103,在所述第一子进程中创建新的会话,将所述第一子进程设置为所述会话的主进程,设置所述第一子进程的进程组ID、进程用户ID、进程有效用户ID、进程组用户ID;
- [0147] S104,在所述第一子进程中创建第二子进程;
- [0148] S105,通过所述第二子进程执行所述shell命令;
- [0149] S106,通过所述第一子进程获取所述shell命令的执行结果。
- [0150] 或者,该计算机程序被处理器执行时实现上述系统实施例中各模块/单元的功能,例如图2中的单元201-203:
- [0151] 接收单元201,用于接收shell命令;
- [0152] 提权单元202,用于构建父进程ID为1的第一子进程,在所述第一子进程中创建新的会话,将所述第一子进程设置为所述会话的主进程,设置所述第一子进程的进程组ID、进程用户ID、进程有效用户ID、进程组用户ID;
- [0153] 执行单元203,用于在所述第一子进程中创建第二子进程,通过所述第二子进程执行所述shell命令,通过所述第一子进程获取所述shell命令的执行结果。
- [0154] 实施例四
- [0155] 图3为本发明实施例提供的计算机装置的示意图。所述计算机装置1包括存储器20、处理器30、总线50以及存储在所述存储器20中并可在所述处理器30上运行的计算机程序40,例如shell执行权限获取程序。所述处理器30执行所述计算机程序40时实现上述获取shell执行权限的方法实施例中的步骤,例如图1所示的S101-S106:
- [0156] S101,接收shell命令;
- [0157] S102,构建父进程ID为1的第一子进程;
- [0158] S103,在所述第一子进程中创建新的会话,将所述第一子进程设置为所述会话的主进程,设置所述第一子进程的进程组ID、进程用户ID、进程有效用户ID、进程组用户ID;
- [0159] S104,在所述第一子进程中创建第二子进程;
- [0160] S105,通过所述第二子进程执行所述shell命令;
- [0161] S106,通过所述第一子进程获取所述shell命令的执行结果。
- [0162] 或者,所述处理器30执行所述计算机程序40时实现上述系统实施例中各模块/单元的功能,例如图2中的单元201-203:
- [0163] 接收单元201,用于接收shell命令;
- [0164] 提权单元202,用于构建父进程ID为1的第一子进程,在所述第一子进程中创建新的会话,将所述第一子进程设置为所述会话的主进程,设置所述第一子进程的进程组ID、进程用户ID、进程有效用户ID、进程组用户ID;
- [0165] 执行单元203,用于在所述第一子进程中创建第二子进程,通过所述第二子进程执行所述shell命令,通过所述第一子进程获取所述shell命令的执行结果。
- [0166] 示例性的,所述计算机程序40可以被分割成一个或多个模块/单元,所述一个或者多个模块/单元被存储在所述存储器20中,并由所述处理器30执行,以完成本发明。所述一个或多个模块/单元可以是能够完成特定功能的一系列计算机程序指令段,该指令段用于描述所述计算机程序40在所述计算机装置1中的执行过程。例如,所述计算机程序40可以被

分割成图2中的接收单元201、提权单元202、执行单元203,各单元具体功能参见实施例二。

[0167] 所述计算机装置1可以是PC(Personal Computer,个人电脑),也可以是智能手机、平板电脑、掌上电脑、便携计算机、智能路由器、矿机、网络存储设备等终端设备。

[0168] 所称处理器30可以是中央处理单元(Central Processing Unit,CPU),还可以是其他通用处理器、数字信号处理器(Digital Signal Processor,DSP)、专用集成电路(Application Specific Integrated Circuit,ASIC)、现成可编程门阵列(Field-Programmable Gate Array,FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者该处理器30也可以是任何常规的处理器等,所述处理器30是所述计算机装置1的控制中心,利用各种接口和线路连接整个计算机装置1的各个部分。

[0169] 所述存储器20可用于存储所述计算机程序40和/或模块/单元,所述处理器30通过运行或执行存储在所述存储器20内的计算机程序和/或模块/单元,以及调用存储在存储器20内的数据,实现所述计算机装置1的各种功能。所述存储器20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计算机装置1的使用所创建的数据(比如音频数据、电话本等)等。此外,存储器20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,例如硬盘、内存、插接式硬盘,智能存储卡(Smart Media Card,SMC),安全数字(Secure Digital,SD)卡,闪存卡(Flash Card)、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。

[0170] 所述总线50以是外设部件互连标准(peripheral component interconnect,简称PCI)总线或扩展工业标准结构(extended industry standard architecture,简称EISA)总线等。所述总线50可以分为地址总线、数据总线、控制总线等。为便于表示,图3中仅用一条粗线表示,但并不表示仅有一根总线或一种类型的总线。

[0171] 进一步地,计算机装置1还可以包括网络接口,网络接口可选的可以包括有线接口和/或无线接口(如WI-FI接口、蓝牙接口等),通常用于在该计算机装置1与其他电子设备之间建立通信连接。

[0172] 可选地,该计算机装置1还可以包括用户接口,用户接口可以包括显示器(Display)、输入单元比如键盘(Keyboard),可选的用户接口还可以包括标准的有线接口、无线接口。可选地,在一些实施例中,显示器可以是LED显示器、液晶显示器、触控式液晶显示器以及OLED(Organic Light-Emitting Diode,有机发光二极管)触摸器等。其中,显示器也可以适当的称为显示屏或显示单元,用于显示在计算机装置1中处理的信息以及用于显示可视化的用户界面。

[0173] 本领域技术人员可以理解,所述示意图3仅仅是计算机装置1的示例,并不构成对计算机装置1的限定,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。

[0174] 在上述实施例中,可以全部或部分地通过软件、硬件、固件或者其任意组合来实现。当使用软件实现时,可以全部或部分地以计算机程序产品的形式实现。

[0175] 所述计算机程序产品包括一个或多个计算机指令。在计算机装置上加载和执行所述计算机程序指令时,全部或部分地产生按照本发明实施例所述的流程或功能。所述计算

机装置可以是通用计算机、专用计算机、计算机网络、或者其他可编程装置。所述计算机指令可以存储在计算机可读存储介质中,或者从一个计算机可读存储介质向另一计算机可读存储介质传输,例如,所述计算机指令可以从一个网站站点、计算机、服务器或数据中心通过有线(例如同轴电缆、光纤、数字用户线(DSL))或无线(例如红外、无线、微波等)方式向另一个网站站点、计算机、服务器或数据中心进行传输。所述计算机可读存储介质可以是计算机能够存储的任何可用介质或者是包含一个或多个可用介质集成的服务器、数据中心等数据存储设备。所述可用介质可以是磁性介质,(例如,软盘、硬盘、磁带)、光介质(例如,DVD)、或者半导体介质(例如固态硬盘Solid State Disk(SSD))等。

[0176] 所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,上述描述的系统,装置和单元的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0177] 在本申请所提供的几个实施例中,应该理解到,所揭露的系统,装置和方法,可以通过其它的方式实现。例如,以上所描述的装置实施例仅仅是示意性的,例如,所述单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0178] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0179] 另外,在本申请各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0180] 所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读存储介质中。基于这样的理解,本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机装置(可以是个人计算机,服务器,或者网络设备)执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0181] 需要说明的是,上述本发明实施例序号仅仅为了描述,不代表实施例的优劣。并且本文中的术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、装置、物品或者方法不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、装置、物品或者方法所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括该要素的过程、装置、物品或者方法中还存在另外的相同要素。

[0182] 以上仅为本发明的优选实施例,并非因此限制本发明的专利范围,凡是利用本发明说明书及附图内容所作的等效结构或等效流程变换,或直接或间接运用在其他相关的技

术领域,均同理包括在本发明的专利保护范围内。

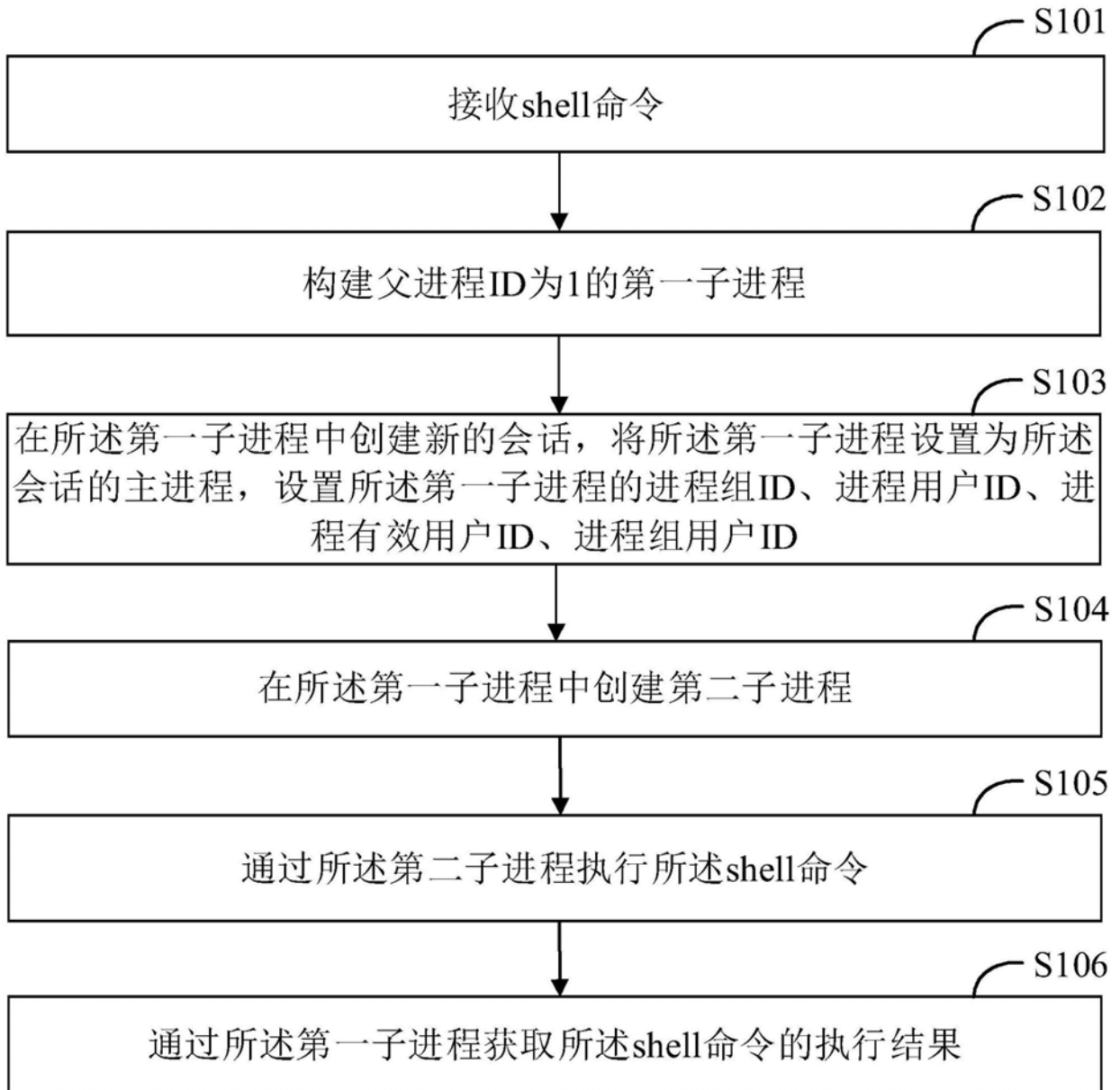


图1



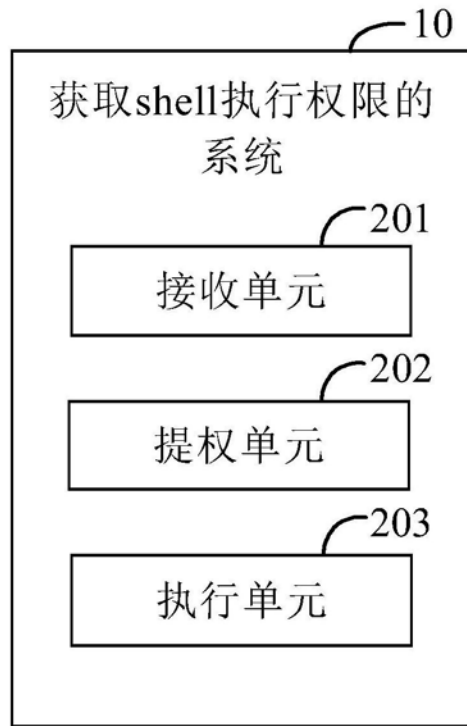


图2

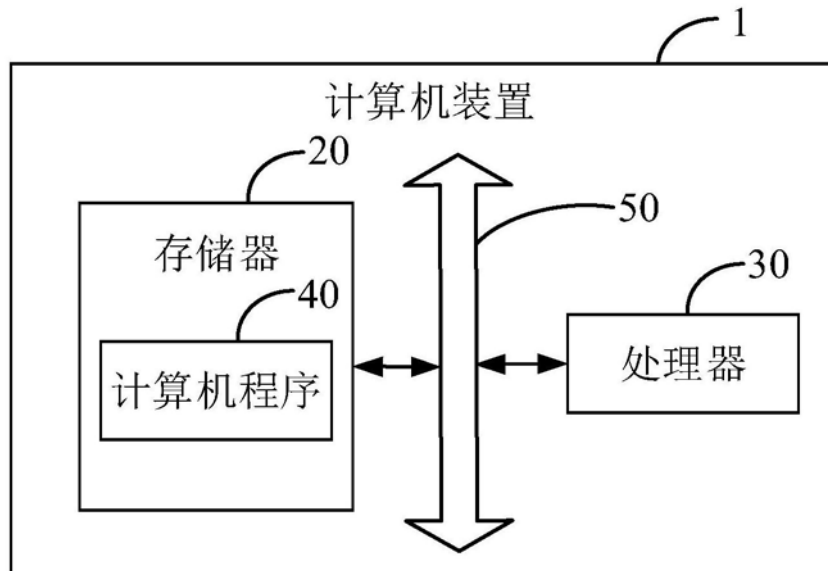


图3