Android沙箱机制

### 一、Android沿用Linux权限模型

      沙箱，对使用者来说可以理解为一种安全环境，对恶意访问者来说是一种限制。

      在Android系统中，应用（通常）都在一个独立的沙箱中运行，即每一个Android应用程序都在它自己的进程中运行，都拥有一个独立的Dalvik虚拟机实例。Dalvik经过优化，允许在有限的内存中同时高效地运行多个虚拟机的实例，并且每一个Dalvik应用作为一个独立的Linux进程执行。Android这种基于Linux的进程“沙箱”机制，是整个安全设计的基础之一。

      Android从Linux继承了已经深入人心的类Unix进程隔离机制与最小权限原则，同时结合移动终端的具体应用特点，进行了许多有益的改进与提升。具体而言，进程以隔离的用户环境运行，不能相互干扰，比如发送信号或者访问其他进程的内存空间。因此，Android沙箱的核心机制基于以下几个概念：大多数进程拥有唯一的用户ID（UID），以及严格限制文件系统权限、标准的Linux进程隔离。

      Android系统沿用了Linux的UID/GID权限模型，即用户ID和用户组ID，但并没有使用传统的passwd和group文件来存储用户与用户组的认证凭据，作为替代，Android定义了从名称到独特标识符Android ID（AID）的映射表。初始的AID映射表包含了一些与特权用户及系统关键用户（如system用户/用户组）对应的静态保留条目。

      除了AID，Android还使用了辅助用户组机制，以允许进程访问共享或受保护的资源。例如，sdcard\_rw用户组中的成员允许进程读写/sdcard目录，因为它的加载项规定了哪些用户组可以读写该目录。这与许多Linux发行版中对辅助用户组机制的使用是类似的。除了用来实施文件系统访问，辅助用户组还会被用于向进程授予额外的权限。权限是关于允许或限制应用程序（而不是用户）访问设备资源。

      Android扩展了Linux内核安全模型的用户与权限机制，将多用户操作系统的用户隔离机制巧妙地移植为应用程序隔离。在linux中，一个用户标识（UID）识别一个给定用户；在Android上，一个UID则识别一个应用程序。在安装应用程序时向其分配UID。应用程序在设备上存续期间内，其UID保持不变。仅限用于允许或限制应用程序（而非用户）对设备资源的访问。如此，Android的安全机制与Linux内核的安全模型完美衔接！不同的应用程序分别属于不同的用户，因此，应用程序运行于自己独立的进程空间，与UID不同的应用程序自然形成资源隔离，如此便形成了一个操作系统级别的应用程序“沙箱”。

**二、Android沙箱模型**

**1、应用程序在独立的进程（UID）**

      应用程序进程之间，应用程序与操作系统之间的安全性由Linux操作系统的标准进程级安全机制实现。在默认状态下，应用程序之间无法交互，运行在进程沙箱内的应用程序没有被分配权限，无法访问系统或资源。因此，无论是直接运行于操作系统之上的应用程序，还是运行于Dalvik虚拟机的应用程序都得到同样的安全隔离与保护，被限制在各自“沙箱”内的应用程序互不干扰，对系统与其他应用程序的损害可降至最低。Android应用程序的“沙箱”机制如下图 1，互相不具备信任关系的应用程序相互隔离，独自运行，箭头访问是禁止的：沙箱内的应用程序运行于独立的虚拟机进程中，该进程独享应用程序的资源且无法访问系统和其他应用程序的资源。

总结：（无论是直接运行于操作系统之上的应用程序，还是运行于Dalvik虚拟机的应用程序的）不同的应用程序，一般应用程序都具有不同的UID（系统为每个应用分配一个唯一的用户标识（UID）），运行在不同的进程中，被限制在各自“沙箱”内的应用程序互不干扰，好处：对系统与其他应用程序的损害可降至最低。

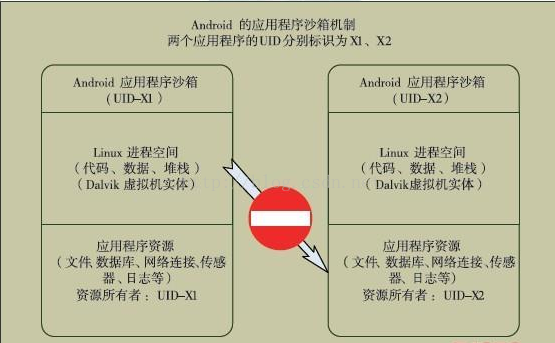


图 1.应用程序在独立的进程

      Android 是一个多用户系统，每个应用是一个独立的用户。系统为每个应用分配一个唯一的用户标识（UID），并为应用中所有文件设置该用户才能访问的权限。每个进程中有一个独立的VM。每个应用在自己的进程中运行，应用的组件需要执行时，系统创建该进程；当系统内存不存时，系统会销毁该进程。

**2、应用程序在同一个进程（共享UID）**

      Android 应用程序运行在它们自己的 Linux 进程上，并被分配一个惟一的用户 ID。默认情况下，运行在基本沙箱进程中的应用程序没有被分配权限，因而防止了此类应用程序访问系统或资源。但是 Android 应用程序可以通过应用程序的 manifest 文件请求权限。

      通过做到以下两点，Android 应用程序可以允许其他应用程序访问它们的资源：

      （1）声明适当的 manifest 权限；

      （2）与其他受信任的应用程序运行在同一进程中，从而共享对其数据和代码的访问。

      在很多情况下，源自同一开发者或同一开发机构的应用程序，相互间存在信任关系。Android系统提供一种所谓共享UID（SharedUserID）机制，使具备信任关系的应用程序可以运行于同一进程空间。通常 ，这种信任关系由应用程序的数字签名确定，并且需要应用程序在manifest文件中使用相同的UID。共享UID的应用程序进程空间如下图：

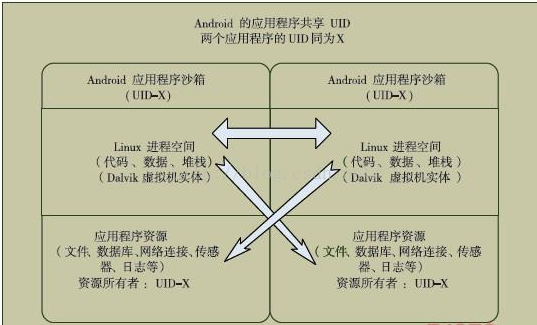


图 2.应用程序共享UID

      不同的应用程序可以运行在相同的进程中。对于此方法，首先必须使用相同的私钥签署这些应用程序，然后必须使用 manifest 文件给它们分配相同的 Linux 用户 ID，这可以通过用相同的值/名定义 manifest 属性 android:sharedUserId 来做到。通过sharedUserId，拥有同一个User id的多个APK安装包可以配置成运行在同一个进程中.所以默认就是可以互相访问任意数据. 也可以配置成运行成不同的进程, 同时可以访问其他APK的数据目录下的数据库和文件.就像访问本程序的数据一样。这样就为同一个机构发开的不同App之间的数据共享，提供了便利。

转载至：https://blog.csdn.net/ljheee/article/details/53191397