

# Android OS 手机平台的安全机制 分析和应用研究

宋 杰,党李成,郭振朝,赵 萌

(安徽大学 计算智能与信号处理教育部重点实验室,安徽 合肥 230039)

**摘 要:**作为现实生活中非常重要的通讯工具,手机的安全性的意义不言而喻。为研究 Android OS 是如何保障手机安全的,文中在深入了 Android 智能手机操作系统平台的结构和特点的基础上,结合手机使用过程中常出现的一些安全问题,细致地诠释了手机安全的涵义,研究了 Android OS 手机安全原则和安全机制。实验以最常见的媒体播放任务为例,具体分析 Android 的安全机制运行原理。实验结果表明,Android 智能平台是以 Linux 安全机制为基础,借助其两个安全元素——UID 和权限,很好地保障了手机中的数据安全和系统安全。

**关键词:**Android OS;手机安全;UID;权限

**中图分类号:**TP309

**文献标识码:**A

**文章编号:**1673-629X(2010)06-0152-04

## The Security Mechanism Analysis and Applied Research of Android OS Mobile Platform

SONG Jie, DANG Li-cheng, GUO Zhen-chao, ZHAO Meng

(Ministry of Education Key Laboratory of Intelligent Computing & Signal Processing,  
Anhui University, Hefei 230039, China)

**Abstract:** Phone is a very important communication tool for human, and the importance of its security also goes without saying. To study how to protect the safety of mobile phones in the Android OS, this in-depth learning the structure and features of the Android smart-phone operating system platform, combined with some security issues emerging frequently during the mobile phones, interprets the meaning of mobile phone safety research of the Android OS handset security principles and security mechanisms in detail. The experiment takes the most common media player missions for example, specifically analyzing of the principle of the security mechanisms in the Android operating OS. Experimental results show that, based on intelligent platform for Linux-based security mechanism, with the help of its two security elements - UID and privileges, Android displays very well in the mobile phone data security and system security.

**Key words:** Android OS; cell phone security; UID; permissions

## 0 引 言

手机作为日常生活和工作的重要通讯工具,其功能越来越强大。但随着智能手机的广泛应用,手机的安全问题日益凸显,也引起人们对于手机的安全性越来越多的关注和担心。

Android 作为 Google 公司推出的开源手机操作系统,由于其安全性上的增强和在 IT 界的巨大影响力,必然会在业界造成巨大影响。

## 1 Android OS 平台

### 1.1 Android 的简介

Android 是 Google 与 OHA(Open Handset Alliance 开放手机联盟)合作开发的基于 Linux2.6 平台的开源智能手机操作系统平台。

### 1.2 Android OS 的架构组成

Android OS 的整体架构分为以下 4 个层次<sup>[1]</sup>:

(1)第一层——Applications。Application 层是 Android OS 的用户应用层,它包括一系列核心应用程序包,例如 email 客户端、SMS 短消息程序、浏览器等。

(2)第二层——Application Framework。该层是 Android 平台专门为应用程序的开发而设计的,允许开发人员完全访问核心应用程序所使用的 API 框架。它由一系列的服务和系统构成,其中包括:视图

收稿日期:2009-08-26;修回日期:2009-12-07

基金项目:安徽省教育科研项目(2008jyxm277);安徽大学教研项目(XJ200705)

作者简介:宋 杰(1966-),男,副教授,硕士生导师,研究方向为嵌入式系统、计算机原理与接口、生物信息学。

(Views)、内容提供者(ContentProviders)等等。

(3)第三层——Libraries(核心库)和 Android Runtime(虚拟机)。这一层主要与进程运行相关,核心库提供了 JAVA 编程语言核心库的大多数功能。另外每一个 Android 程序都有独立的 Dalvik 虚拟机为它提供运行环境。

(4)第四层——Linux Kernel。Android 的内核为 Linux 2.6 内核,它主要用于保障安全性、内存管理、进程管理、网络协议栈和驱动模型等等。

具体 Android 框架结构如图 1 所示。

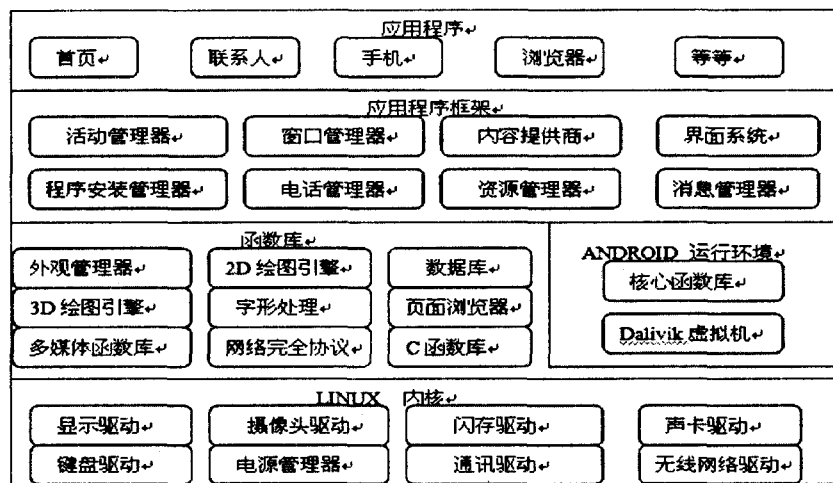


图 1 Android 系统框架结构图

## 2 手机安全的定义

在实际应用中,手机的安全问题主要是由手机病毒引起的,其主要危害可以分为五种情况<sup>[2]</sup>:

- ①导致用户手机里的私人信息丢失;
- ②控制手机进行强行消费,导致手机用户通信费用及信息费用剧增;
- ③通过手机短信的方式传播非法信息;
- ④破坏手机软件或者硬件系统;
- ⑤造成手机通讯瘫痪。

分析这五种安全问题,文中把手机安全进一步归纳为两个部分——数据安全和系统安全。下面从这两部分入手深入分析手机的安全问题。

### 2.1 数据安全

通常手机中数据安全被认为是确保完整性、机密性和可用性三个方面。

#### 2.1.1 完整性

在手机中,数据的完整性是指合法的而且没有经过偶然的或恶意的修改。当进行数据存储或使用时应该考虑其完整性:一方面,物理传输和存储媒介必须是可靠的,以使得数据可以正确地传输而不发生错误。另一方面,必须禁止未经授权的程序不加检测地访问

和修改。

#### 2.1.2 机密性

当受保护的数据只能被授权的程序读取或者修改时,要保持其机密性。这是一个与完整性截然不同的概念:当数据在传输时,它可能是被毫无修改地正确传输,因此确保了其完整性,但是如果被第三方中途截取的话就再也不是机密的了。另一方面在手机中当未被授权的程序可以访问数据传输并从中获取有价值的信息时,只有完整性是不够的。

#### 2.1.3 可用性

即使完整性和机密性都得到了保证,如果数据不能被访问,那么它也是不可用的。可用性措施确保数据永远不会丢失,而且当被请求时,可以以预定义的性能级别被访问。

## 2.2 系统安全

系统安全指的是手机平台本身的安全,是对信息系统的保护,防止未授权的访问及对信息的修改,并防止对授权用户服务的拒绝或对未授权用户服务的允许,包括那些检测、记录和反击此类威胁的措施。

## 3 Android 系统的安全机制分析

针对第 2 节中提到的手机的两类安全问题,本节将结合 Android 的安全设计原则和安全机制,具体分析 Android OS 平台如何保障手机的数据安全和系统安全。为方便讨论,把 Android OS 平台的安全机制也相应地分为系统安全机制和数据安全机制。

### 3.1 Android 的安全设计原则

Android 的安全设计包括以下两个原则:

(1)在默认情况下,在 Android OS 平台下运行的应用程序没有权限执行对其它应用程序、操作系统或用户有害的操作。这些操作包括读/写用户的隐私数据(例如联系方式或 e-mail)、读/写其它应用程序的文件等等。

(2)Android 应用程序的进程是运行在一个安全的“沙箱”环境中。它不能干扰其它应用程序,除非它明确声明权限。这些权限请求能够被不同方式的操作所处理,特别的要基于证书和用户的提示被自动的允许或禁止。而且权限的请求在应用程序中被声明为静态的,所以在此之后在安装时或没有改变时系统会预先知道。

### 3.2 Android 的系统安全机制

由于 Android OS 的内核是根据手机运行环境定

制的 Linux 2.6 内核,因此 Android OS 相应地继承了 Linux 2.6 的安全机制。

同时 Android 的系统安全也主要靠 Linux 2.6 的安全机制来实现。Linux 系统本身有出色的安全性和稳定性,而作为目前比较新的 Linux 2.6 版的内核又加入了安全模块的安全机制来增强 Linux 系统的安全性<sup>[3~5]</sup>。

### 3.3 Android 的数据安全机制

Android 的数据安全机制涉及到两个重要的安全元素:UID(用户标识)、权限,而且 Android OS 平台下的数据安全机制也依赖两者来构造。

#### 3.3.1 两个安全元素的概念

UID,安装在 Android 手机中的每个程序都会被分配一个属于自己的统一的 Linux 用户 ID,并且为它创建一个沙箱以防止影响其它程序(或者其它程序影响它)。用户 ID 在程序安装到手机中被分配,并且在这个设备中保持它的永久性。另外程序创建的任何文件都会被赋予程序的用户标识,并且正常情况下不能被其它包访问。

权限,为 Android OS 中允许用户或者程序执行的操作,包括打开数据文件、发送信息和调用 Android 组件等等。权限是 Android 为保障安全而设定的安全标识,同时也是程序实现某些特殊操作(比如申请系统 Service)的基础。

#### 3.3.2 Android 的数据安全机制

##### (1)数据完整性的实现。

根据 Android 的安全原则知道,Android OS 中的数据在默认情况下(除系统授权外)不会被其它程序破坏、被读取、修改、删除以及丢失。如果一个程序需要对其它程序的数据进行读取等操作,系统(或者手机用户)将会在程序安装阶段审核它有无相应的权限。

##### (2)数据机密性的实现。

在 Android OS 中是结合 UID 和权限来保证数据的机密性。首先由 UID 的定义可以知道在 Android 中,每一个程序所创建的文件都会赋予改程序的 UID,这就相当于给它的文件打上了安全保护“标签”,以后只有通过它才可对这个文件进行访问等操作。也正是由于这种排它性的标识,其它程序在没有系统授予权限的情况下,都不能对这个文件进行访问和修改。

##### (3)数据可用性的实现。

另外,Android OS 在保障数据的完整性和机密性的基础上,采用赋予相应权限的方法来保证数据可以被有效使用,也就是保障了数据的可用性。一般程序要对文件(或者系统服务)进行操作,需要三步走——权限声明、权限审核和权限确认。

## 4 应用研究

给出一个实例来介绍安全机制的应用。下面以调用 MediaPlayer 的应用程序 S 调用系统服务,介绍 Android OS 平台下中授权操作的主要过程。

### 4.1 权限声明

首先必须在该程序中的 AndroidManifest.xml 文件中包含一个 <uses-permission> 标签来声明此权限。需要指定如下内容:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.android.app.myapplication" >
    <uses-permission android:name="android.permission.RECEIVE_MediaPlayer"/>
</manifest>
```

出于安全考虑,系统(或者手机用户)将在程序 S 安装时看到这项声明,并且决定是否批准此权限。如果批准程序将得以安装成功,反之则安装失败。当 S 安装完毕,执行到要进行调用 MediaPlayer 的操作时,需要对系统申请该服务。在 Android 中 S 要进行这种操作是通过 S 的一个 Activity 组件来申请 Service 组件(MediaPlayer 是系统组件)来完成的<sup>[6,7]</sup>。而 Activity 组件与 Service 组件在不同的进程里执行,各有不同的 UID。由于它们各自独自执行,所以 Activity 组件通常依赖 Intent 组件去请求 Android 启动所需要的 Service<sup>[8,9]</sup>。

### 4.2 权限审核

接收到 Activity 组件提出的权限申请之后,系统将在 Service 类里,做权限的审核,其常用指令如图 2 所示。

```
Public class NotifyService extends Service {    @Override
    Public void onCreate() {
        Super.onCreate();
    }

    @Override
    Public IBinder onBind(Intent intent) {
        IBinder ib = new NotifyBinder(this);
        Return ib;
    }

    @Override
    Public void onStart(Intent intent, int startid) {
        Context ctx = super.getApplicationContext();
        Int ck = ctx.getWallpaperDesiredMinimumHeight();
        checkCallingOrSelfPermission(String permission);
        checkCallingOrSelfPermission(Uri uri, int modeFlags);
        checkCallingPermission(String permission);
        checkCallingPermission(Uri uri, int modeFlags);
        checkPermission(String permission, int pid, int uid);
        checkUriPermission(Uri uri, int pid, int uid, int mode);
        checkUriPermission(Uri uri, string readPermission);
    }
}
```

图 2 Service 权限审核源代码

当 Service 确认了对方的这种权限,就将 IBinder 界面的参数(Reference)传给 Activity 物件。Activity 物件就能透过 IBinder 界面去使用 Binder 的服务了<sup>[10]</sup>。如图 3 所示。

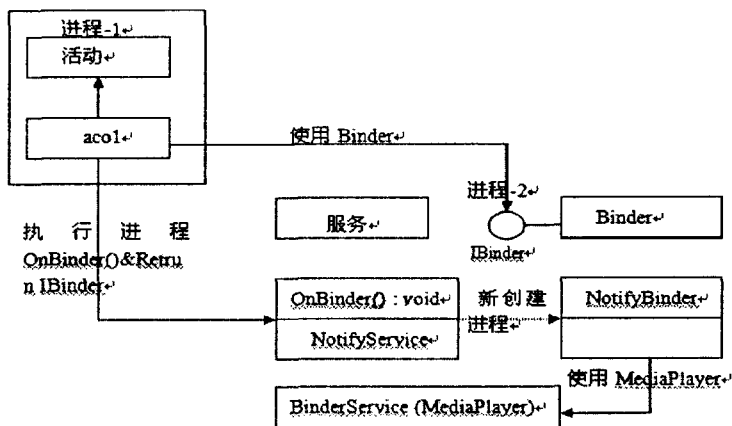


图 3 Activity 调用 MediaPlayer 流程

#### 4.3 权限确认

当 Activity 呼叫 IBinder 的 transact() 等函数时,会反向呼叫 NotifyBinder 子类别的 onTransact() 函数。此时,也可以进行安全检查,例如下图里的指令:

```
int uid = Binder.getCallingUID();
```

就能取得对方 UID 来检查它的身分等。还将进行 checkCallingPermission() 等检查<sup>[11]</sup>。具体源代码如图 4 所示。

```
Public class NotifyService extends Binder {
    Private Context ctx;
    Public NotifyBinder(Context cx) {
        Ctx = cx
    }
    @Override
    Protected boolean on Transact(int c, Parcel da, ) {
        int uid = Binder.getCallingUid();
        int pid = Binder.getCallingPid();
        ctx.
            checkCallingOrSelfPermission(String permission);
            checkCallingOrSelfPermission(Uri uri, int mode);
            checkCallingPermission(String permission): int;
            checkCallingUriPermission(Uri uri, int modeFlags);
            checkPermission(String permission, int pid, int uid);
            checkUriPermission(Uri uri, int pid, int uid, int mode);
            heckUriPermission(Uri uri, string readPermission);
        return;
    }
}
```

图 4 NotifyBinder 类中权限检查代码

最终权限检查通过,系统启动 BinderServer 来

提供 MediaPlayer 播放服务。

以上就是程序 S 申请并实现调用 MediaPlayer 的全过程。通过这个例子可以看出,Android OS 对权限的申请、审核以及确认是相当严格的,同时也正是这样的安全权限机制保障了手机中数据的安全性。

#### 5 结束语

在 Android OS 中采用 Android 安全原则与 Android 安全机制相结合的方法,来解决现实的两类安全问题,很好地保障了手机的安全。同时也应该看到手机的安全问题是一个系统的问题,它需要全面地考虑手机硬件、操作系统和应用程序等方面的问题,手机的长期安全性有待于手机用户在使用过程中增强安全意识,并且做好维护比如及时升级手机系统软件和更新手机病毒补丁等工作。

#### 参考文献:

- [1] Android 中文网(androidcn.net). What is Android[EB/OL]. 2007-12-17. <http://sdk.androidin.com/what-is-android.html>.
- [2] 刘磊,刘克胜.Symbian 操作系统下手机病毒免疫技术研究[J]. 网络安全技术与应用,2006(11):89-91.
- [3] Henricksen M, Caelli P. Securing grid data using mandatory access controls[C]//Fifth Australasian Symposium on Grid Computing and e-Research(AusGrid 2007). [s.l.]:[s.n.], 2007:25-32.
- [4] 杨杰.基于 Linux 的强制访问控制研究[J]. 电脑与电信,2008(11):48-51.
- [5] 李舒亮,习军.基于 Linux 的数据安全传输的研究[J]. 微计算机信息,2008,24(24):18-20.
- [6] Granlich N. Android Programming[EB/OL]. 2008-02-21. <http://andbook.anddev.org>.
- [7] Katysovas T. A first look at Google Android[M]. [s.l.]:Free University of Bolzano,2008:12-26.
- [8] 尹涛,李翔,林祥,等.基于 AOP 的角色访问控制模型设计与实现[J]. 计算机技术与发展,2008,18(10):136-138.
- [9] DiMarzio J. Android A Programmer's Guide[M]. [s.l.]:McGraw-Hill/Osborne Media,2008.
- [10] Meier R. Professional Android Application[M]. [s.l.]:Wiley, John & Sons, Incorporated,2008.
- [11] Pilgrim M. Anatomy & Physiology of an Android[EB/OL]. 2008-06-09. <http://www.youtube.com/watch?v=InK-P-PrGE>.