

人多不足以依赖，要生存只有靠自己。  
深窥自己的心，而后发觉一切的奇迹在你自己。  
凡事皆有终结,因此,耐心是赢得成功的一种手段。

新随笔

联系

订阅

管理



CSDN专栏

昵称：智慧云端日记

园龄：5年1个月

粉丝：379

关注：6

+加关注

## Android 监听短信(同时监听广播和数据库)

暗扣，强烈谴责这种侵害用户利益的行为。。。

下面给大家介绍Android暗扣原理..... Android4.4以下的系统玩游戏就要小心了哈

暗扣方式之一：短信订购，即监听-----拦截-----处理短信。

暗扣方式之二：模拟人为操作（又叫模拟流量），通过后台程序代码模拟人的点击行为，暗自  
营商收取你的费用，当然这其中也需要涉及监听/拦截/处理短信。使用这种方式的原理无非是H  
接入点切换问题，这里就不详细讲解。

回归正题：有的时候，我们的手机程序需要监听手机短信，当满足条件A时，不处理；当满足条件B时，符其议直为已  
读；当满足条件C时，将短信删除。

如果对你有帮助,请任意打赏



[http://blog.csdn.net/zyw\\_java](http://blog.csdn.net/zyw_java)

<	2017年9月						>
日	一	二	三	四	五	六	
27	28	29	30	31	1	2	
3	4	5	6	7	8	9	
10	11	12	13	14	15	16	
17	18	19	20	21	22	23	
24	25	26	27	28	29	30	
1	2	3	4	5	6	7	

搜索

谷歌搜索

我的标签

Android(84)

Java(37)

微信(35)

Linux(32)

CentOS(30)

支付宝(6)

Oracle(5)

SQL(4)

IO(3)

注：Android 4.4以及以后可能由用户来控制程序的权限，如果用户关闭这个程序的权限，意味着你无法监听短信/操作短信内容

目前也有如小米系统在安装时，让用户来控制权限；360来监听优先拦截短信等等（关于谁先安装谁有优先权，动态注册比静态注册优先级别高等问题，在这些情况这里就不讲了）。。。

Android 实现监听短信(同时监听广播和数据库)代码如下：

### 1. 拦截广播



```
package com.javen.sms.receiver;

import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.TimeZone;

import com.javen.util.InterceptKeyKeeper;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.telephony.SmsMessage;
import android.util.Log;
import android.widget.Toast;

public class SmsReceiver extends BroadcastReceiver {
    private Context mContext;
    public static final String SMS_RECEIVED_ACTION =
        "android.provider.Telephony.SMS_RECEIVED";
    public static final String SMS_DELIVER_ACTION =
        "android.provider.Telephony.SMS_DELIVER";

    @Override
    public void onReceive(Context context, Intent intent) {
        this.mContext=context;
        Toast.makeText(context, "接收短信执行了.....", Toast.LENGTH_LONG).show();
        Log.e("SMSReceiver, isOrderedBroadcast()", isOrderedBroadcast()+"");
    }
}
```

如果对你有帮助,请任意打赏

[http://blog.csdn.net/zyw\\_java](http://blog.csdn.net/zyw_java)

加密(3)

更多

## 积分与排名

积分 - 301905

排名 - 543

## 阅读排行榜

1. Java实现Excel导入数据库，数据库中的数据导入到Excel(131264)

2. 使用Jsoup 抓取页面的数据(76295)

3. Android EditText的设置(62045)

4. SSM框架——详细整合教程（Spring+SpringMVC+MyBatis）(61881)

5. Android中的PopupWindow详解(52269)

Android 监听短信(同时监听广播和数据库) - 智慧云端日记 - 博客园

```

Log.e("SmsReceiver onReceive...", "接收短信执行了.....");
String action = intent.getAction();
if (SMS_RECEIVED_ACTION.equals(action) || SMS_DELIVER_ACTION.equals(action)) {
    Toast.makeText(context, "开始接收短信.....", Toast.LENGTH_LONG).show();
    Log.e("SmsReceiver onReceive...", "开始接收短信.....");

    Bundle bundle = intent.getExtras();
    if (bundle != null) {
        Object[] pdus = (Object[])bundle.get("pdus");
        if (pdus != null && pdus.length > 0) {
            SmsMessage[] messages = new SmsMessage[pdus.length];
            for (int i = 0; i < pdus.length; i++) {
                byte[] pdu = (byte[]) pdus[i];
                messages[i] = SmsMessage.createFromPdu(pdu);
            }
            for (SmsMessage message : messages) {
                String content = message.getMessageBody();// 得到短信内容
                String sender = message.getOriginatingAddress();// 得到发信息的号码
                if (content.contains(InterceptKeyKeeper.getInterceptKey(mContext)))
                {
                    Toast.makeText(mContext, "内容为：" + content,
                        Toast.LENGTH_LONG).show();
                    // setResultData(null);
                    this.abortBroadcast();// 中止
                } else if (sender.equals("10010") || sender.equals("10086")) {
                    Toast.makeText(mContext, "内容为：" + content,
                        Toast.LENGTH_LONG).show();
                    this.abortBroadcast();// 中止
                }
                Date date = new Date(message.getTimestampMillis());
                SimpleDateFormat format = new SimpleDateFormat("yy-MM-dd HH:mm:ss");
                format.setTimeZone(TimeZone.getTimeZone("GMT+08:00"));
                String sendContent = format.format(date) + ":" + content;
                Log.e("SmsReceiver onReceive ", sendContent + " ");
            }
        }
    }
}

```

如果对你有帮助,请任意打赏




http://blog.csdn.net/zyw\_java

```

    }
}
}

```



## 2. 开启一个服务开监听数据库



```

package com.javen.service;

import android.app.Service;
import android.content.ContentResolver;
import android.content.Intent;
import android.net.Uri;
import android.os.IBinder;
import android.os.Process;
import android.widget.Toast;

/**
 * @author Javen
 * 开启一个服务开监听数据库
 */
public class SmsService extends Service {

    private SmsObserver mObserver;

    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }

    @Override
    public void onCreate() {
        Toast.makeText(this, "SmsService 服务器启动了...", Toast.LENGTH_LONG).show();
    }
}

```


如果对你有帮助,请任意打赏



[http://blog.csdn.net/zyw\\_java](http://blog.csdn.net/zyw_java)

```
// 在这里启动
ContentResolver resolver = getContentResolver();
mObserver = new SmsObserver(resolver, new SmsHandler(this));
resolver.registerContentObserver(Uri.parse("content://sms"), true, mObserver);
}

@Override
public void onDestroy() {
    super.onDestroy();
    this.getContentResolver().unregisterContentObserver(mObserver);
    Process.killProcess(Process.myPid());
}
}
```



### 3. 数据库观察者



```
package com.javen.service;

import android.content.ContentResolver;
import android.database.ContentObserver;
import android.database.Cursor;
import android.net.Uri;
import android.os.Message;
import android.util.Log;

/**
 * @author Javen
 * 数据库观察者
 */
public class SmsObserver extends ContentObserver {

    private ContentResolver mResolver;
```

如果对你有帮助,请任意打赏



[http://blog.csdn.net/zyw\\_java](http://blog.csdn.net/zyw_java)

```
public SmsHandler smsHandler;

public SmsObserver(ContentResolver mResolver, SmsHandler handler) {
    super(handler);
    this.mResolver = mResolver;
    this.smsHandler = handler;
}

@Override
public void onChange(boolean selfChange) {
    Log.i("SmsObserver onChange ", "SmsObserver 短信有改变");
    Cursor mCursor = mResolver.query(Uri.parse("content://sms/inbox"),
        new String[] { "_id", "address", "read", "body", "thread_id" },
        "read=?", new String[] { "0" }, "date desc");

    if (mCursor == null) {
        return;
    } else {
        while (mCursor.moveToNext()) {
            SmsInfo _smsInfo = new SmsInfo();

            int _inIndex = mCursor.getColumnIndex("_id");
            if (_inIndex != -1) {
                _smsInfo._id = mCursor.getString(_inIndex);
            }

            int thread_idIndex = mCursor.getColumnIndex("thread_id");
            if (thread_idIndex != -1) {
                _smsInfo.thread_id = mCursor.getString(thread_idIndex);
            }

            int addressIndex = mCursor.getColumnIndex("address");
            if (addressIndex != -1) {
                _smsInfo.smsAddress = mCursor.getString(addressIndex);
            }

            int bodyIndex = mCursor.getColumnIndex("body");
            if (bodyIndex != -1) {
                _smsInfo.smsBody = mCursor.getString(bodyIndex);
            }
        }
    }
}
```

如果对你有帮助,请任意打赏



[http://blog.csdn.net/zyw\\_java](http://blog.csdn.net/zyw_java)

```
    }

    int readIndex = mCursor.getColumnIndex("read");
    if (readIndex != -1) {
        _smsInfo.read = mCursor.getString(readIndex);
    }

    // 根据你的拦截策略, 判断是否不对短信进行操作; 将短信设置为已读; 将短信删除
    // TODO
    System.out.println("获取的短信内容为: "+_smsInfo.toString());
    Log.i("SmsObserver ...", "获取的短信内容为: "+_smsInfo.toString());
    Message msg = smsHandler.obtainMessage();
    _smsInfo.action = 2; // 0不对短信进行操作; 1将短信设置为已读; 2将短信删除
    msg.obj = _smsInfo;
    smsHandler.sendMessage(msg);
}

if (mCursor != null) {
    mCursor.close();
    mCursor = null;
}
}
```



#### 4. 短信处理类



```
package com.javen.service;

import android.content.ContentValues;
import android.content.Context;
import android.net.Uri;
```

如果对你有帮助,请任意打赏



[http://blog.csdn.net/zyw\\_java](http://blog.csdn.net/zyw_java)

```
import android.os.Handler;
import android.os.Message;

/**
 * @author Javen
 *
 *      短信的处理
 */
public class SmsHandler extends Handler {
    private Context mContext;

    public SmsHandler(Context context) {
        this.mContext = context;
    }

    @Override
    public void handleMessage(Message msg) {
        SmsInfo smsInfo = (SmsInfo) msg.obj;

        if (smsInfo.action == 1) {
            ContentValues values = new ContentValues();
            values.put("read", "1");
            mContext.getContentResolver().update(
                Uri.parse("content://sms/inbox"), values, "thread_id=?",
                new String[] { smsInfo.thread_id });
        } else if (smsInfo.action == 2) {
            Uri mUri = Uri.parse("content://sms/");
            mContext.getContentResolver().delete(mUri, "_id=?",
                new String[] { smsInfo._id });
        }
    }
}
```



如果对你有帮助,请任意打赏



[http://blog.csdn.net/zyw\\_java](http://blog.csdn.net/zyw_java)



## 5. SmsInfo 数据结构 主要用于短信拦截



```
package com.javen.service;

/**
 * 主要用于短信拦截
 *
 * @author Javen
 *
 */
public class SmsInfo {
    public String _id = "";
    public String thread_id = "";
    public String smsAddress = "";
    public String smsBody = "";
    public String read = "";
    public int action = 0; // 1代表设置为已读, 2表示删除短信
    @Override
    public String toString() {
        return "SmsInfo [_id=" + _id + ", thread_id=" + thread_id
            + ", smsAddress=" + smsAddress + ", smsBody=" + smsBody
            + ", read=" + read + ", action=" + action + "]\n";
    }
}
```



如果对你有帮助,请任意打赏



[http://blog.csdn.net/zyw\\_java](http://blog.csdn.net/zyw_java)

## 6. 需要的权限以及配置信息



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
```

```
package="com.javen.sms"
android:versionCode="1"
android:versionName="1.0" >

<uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="21" />

<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />

<!-- <uses-permission android:name="android.permission.BROADCAST_SMS" />
<uses-permission android:name="android.permission.READ_SMS" />
<uses-permission android:name="android.permission.WRITE_SMS" /> -->
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    >
    <activity
        android:name=".MainActivity"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

    <receiver
        android:name="com.javen.sms.receiver.SmsReceiver"
        android:exported="true"
        android:permission="android.permission.BROADCAST_SMS">
        <intent-filter android:priority="2147483647" >
            <action android:name="android.provider.Telephony.SMS_RECEIVE" />
            <action android:name="android.provider.Telephony.SMS_DELIVER" />
            <category android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </receiver>
```

如果对你有帮助,请任意打赏



[http://blog.csdn.net/zyw\\_java](http://blog.csdn.net/zyw_java)

```
<service android:name="com.javen.service.SmsService"></service>
</application>

</manifest>
```



## 7. 测试在MainActivity 中启动SmsService 发送短信到10086 或者10010即可测试

```
intentService = new Intent(this, SmsService.class);
startService(intentService);
Toast.makeText(this, "启动service中.....", 1).show();
```

以上就是Android 4.4以下暗扣原理。 Android 4.4以上如何实现大家可以在此博客中讨论交流。。。。。

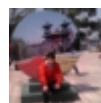
不要让昨天的沮丧 让今天的梦想黯然失色 成功的人总是修改方法而不修改目标 微信公众号：j

标签: [Android](#)

好文要顶

关注我

收藏该文



智慧云端日记

关注 - 6

粉丝 - 379

[+加关注](#)

如果对你有帮助,请任意打赏



[http://blog.csdn.net/zyw\\_java](http://blog.csdn.net/zyw_java)

« 上一篇: [Android 监听屏幕锁屏,用户解锁](#)

» 下一篇: [Eclipse+Tomcat+MAVEN+SVN项目完整环境搭建](#)

posted @ 2015-08-27 18:35 智慧云端日记 阅读(7087) 评论(14) 编辑 收藏

## 评论列表

#1楼 2015-08-27 19:32 路上的脚印

学习了

支持(0) 反对(0)

#2楼[楼主 ] 2015-08-28 10:15 智慧云端日记

@ 路上的脚印  
共同进步.....

支持(0) 反对(0)

#3楼 2015-10-09 17:32 dongweiq

com.javen.util.InterceptKeyKeeper;是什么东东，我这里没有

#4楼[楼主 ] 2015-10-09 18:12 智慧云端日记

@ dongweiq  
就是SharedPreferences

#5楼 2015-10-10 10:15 dongweiq

如果对你有帮助,请任意打赏



[http://blog.csdn.net/zyw\\_java](http://blog.csdn.net/zyw_java)

@ 智慧云端日记

明白了，那段代码的意思应该是要拦截指定关键词。

昨天我也测试成功了，不过经过调试发现了一个有趣的问题

如果在contentprovider里对收到的短信处理（标记成已读或者删除）那么在receiver中就收不到广播了。

但是如果不做任何操作，看log日志的时候，明明是receiver先执行而contentprovider后执行的。

同时在android4.4以下的版本在receiver中使用abortbroadcast，那么contentprovider就不会执行。

现在凌乱的是android系统是怎么执行他们的顺序的，晕晕的，希望您能够指教一下，嘿嘿

支持(0) 反对(0)

---

#6楼[楼主 ] 2015-10-10 10:27 智慧云端日记

@ dongweiq

接收短信 是先有广播的 所以receiver是先执行 如果在receiver中使用abortbroadcast拦截失败 就会保存在收件箱

也就是会执行contentprovider

支持(0) 反对(0)

---

#7楼 2015-10-10 10:40 dongweiq

@ 智慧云端日记

但是为什么android4.4以上版本在处理了短信后（标记已读或者删除）就收不到receiver了行，那么这个解释不通啊

如果对你有帮助,请任意打赏



[http://blog.csdn.net/zyw\\_java](http://blog.csdn.net/zyw_java)

---

#8楼[楼主 ] 2015-10-10 14:07 智慧云端日记

@ dongweiq

我QQ 572839485 加我为好友交流一下

---

#9楼 2015-11-10 16:10 wuball

好像并没有发送短信也

支持(0) 反对(0)

---

#10楼[楼主 ] 2015-11-20 08:56 智慧云端日记

@ wuball

这个是短信的监听 并不是实现发送短信

支持(0) 反对(0)

---

#11楼 2015-11-20 13:53 wuball

@ 智慧云端日记

订购的话需要发送短信吧？

支持(0) 反对(0)

---

#12楼[楼主 ] 2015-11-20 15:02 智慧云端日记

@ wuball

订购？ 不知道你具体的业务逻辑 可以在上面找到我的企鹅聊

支持(0) 反对(0)

---

#13楼 2017-06-03 15:33 Langzi\_beat

这个是我的QQ 907719157 能加上交流下吗？

---

#14楼 2017-06-10 11:58 Langzi\_beat

我真的很想知道5.0以上怎么读取短信数据库

如果对你有帮助,请任意打赏



[http://blog.csdn.net/zyw\\_java](http://blog.csdn.net/zyw_java)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。



#### 最新IT新闻:

- 对话CCtalk CEO陆坚：从盛大到沪江 我喜欢做从0到1
  - 微软全息键盘专利曝光，打通AR/VR输入交互
  - 乐视网宣布：拟更名为新乐视
  - 传滴滴有意投资饿了么狙击美团 饿了么回应：市场传闻
  - 任正非：人工智能要瞄准服务主航道，不做边界外的事情
- » 更多新闻...



#### 最新知识库文章:

- 如何阅读计算机科学类的书
  - Google 及其云智慧
  - 做到这一点，你也可以成为优秀的程序员
  - 写给立志做码农的大学生
  - 架构腐化之谜
- » 更多知识库文章...

#### 历史上的今天:

2012-08-27 处理JAVAWEB中乱码的问题

如果对你有帮助,请任意打赏



[http://blog.csdn.net/zyw\\_java](http://blog.csdn.net/zyw_java)

Copyright ©2017 智慧云端日记

友情链接：[快递查询](#) [快递查询](#)

如果对你有帮助,请任意打赏



[http://blog.csdn.net/zyw\\_java](http://blog.csdn.net/zyw_java)