

生死看淡，不服就干！

<http://www.wjdiankong.cn>

[目录视图](#)[摘要视图](#)[RSS 订阅](#)

技术分享公众号

扫一扫关注

关注公众号留言可咨询问题和
技术交流，推送最新技术文章！



微信扫一扫加入安全逆向圈

异步赠书：10月Python畅销书升级 【重磅】Python工程师养成记！ 程序员9月书讯 节后荐书：Python Kotlin（评论送书）

Android中微信抢红包插件原理解析和开发实现

2015-10-04 22:14

70535人阅读

评论(19)

收藏

分享

分类：

Android (166)

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录(?)

[+]

一、前言

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==



移动安全与逆向

微信扫一扫即可进入圈子



友情链接

Github

Website

四哥私人微信

扫一扫加小编微信

添加时请注明：编码美丽！

电锁屏的时候，我们并不知道有人已经开始发红包了，那么这时候也是让我们丧失了一大批红包的原因。那么关于网络的问题，我们开发者可能用相关技术无法解决(当然在Google和Facebook看来的话，他们的理想是能够在任何地方都能连接互联网，当然在偏远的农村也是，不过我们期待他们有一天能够普及开来。到时候才是真正的互联网)。扯得有点远了。我们回归到正题，今天来看看使用技术来解决其他非网络问题。在充电锁屏的时候也可以自动帮我们抢红包。而且你要知道，机器抢红包的准确率是100%的，这个也许就是人和机器的区别。那么保证抢得准确率是100%的话，那就依赖于我们高效准确的算法实现了。下面就来看看原理实现。

当去年我看到抢红包那么火爆的时候，当时作为一个开发者心里是多么渴望开发一个插件出来，

能想到的就是使用：

adb shell monkey

命令去模拟点击屏幕，但是那种方式有一个问题就是是无头绪的盲目点击，所以几乎会出现误点，点击成功率极其低下。所以当时就没有想到其他方法了，因为最近做了有关辅助功能相关的工作的时候，那么功能可以用于抢红包。

其实现在我们可以去各大市场搜索一下看到，有很多抢红包的插件了。当然我们并不是用于商业化，这里只是来解析一下原理。我们会发现那些插件都有一个共同的特点是：**第一步都是引导用户去开启辅助功能。**

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==





天真无谐

我的第一个App:

天真无谐



已经上线了，在各大市场都可以搜索到。希望大家多多支持!!

[点击查看详情](#)

个应用呢？那么辅助功能就是干这些事，他的功能其实就是可以概括两句话：

第一、寻找到我们想要的View节点

第二、然后模拟点击，实现特定功能

我们知道Android中的View体系是一个树形结构，那么每一个View就是一个节点。所以我们可以查找到指定的节点，那么我们该如何查找到我们想要的节点呢？这里我们先看一下辅助功能(AccessibilityService)的用法

第一步、我们需要集成AccessibilityService类

我们需要自定一个Service然后继承AccessibilityService，当然还需要在AndroidManifest.xml中声明

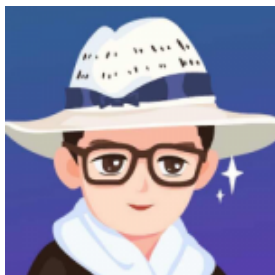
```
<service
    android:name="krelve.demo.rob.RobMoney"
    android:enabled="true"
    android:exported="true"
    android:label="@string/app_name"
    android:permission="android.permission.BIND_ACCESSIBILITY_SERVICE" >
    <intent-filter>
        <action android:name="android.accessibilityservice.AccessibilityService" />
    </intent-filter>

    <meta-data
        android:name="android.accessibilityservice"
        android:resource="@xml/accessibility" />
</service>
```

第二步、声明权限和配置

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==





尼古拉斯_赵四

关注

发私信



访问：5046741次

积分：30842

等级：BLOG > 8

排名：第169名

原创：288篇

转载：0篇

译文：0篇

评论：1605条

当然还要一个meta-data的声明，这个声明是对这个AccessibilityService的配置。我们看一下配置文件内容：

[html]

```
01. <?xml version="1.0" encoding="utf-8"?>
02. <accessibility-service xmlns:android="http://schemas.android.com/apk/res/android"
03.     android:accessibilityEventTypes="typeNotificationStateChanged|typeWindowStateChang
04.     android:accessibilityFeedbackType="feedbackGeneric"
05.     android:accessibilityFlags="flagDefault"
06.     android:canRetrieveWindowContent="true"
07.     android:description="@string/desc"
08.     android:notificationTimeout="100"
09.     android:packageNames="com.tencent.mm" />
```

这里我们看到有很多选项,我们看一下常用的几个属性：

1、android:accessibilityEventTypes="typeAllMask"

看属性名也差不多可以明白，这个是用来设置响应事件的类型，typeAllMask当然就是响应所有类型的事件了。

当然还有单击、长按、滑动等。

2、android:accessibilityFeedbackType="feedbackSpoken"

设置回馈给用户的方式，有语音播出和振动。可以配置一些TTS引擎，让它实现发音。

3、android:notificationTimeout="100"

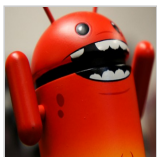
响应时间的设置就不用多说了

4、android:packageNames="com.example.android.apis"

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==



博客专栏



移动安全与逆向

文章：41篇

阅读：692654



算法和数据结构

文章：0篇

阅读：0



Java总结

文章：23篇

阅读：368652



iOS学习总结

文章：27篇

阅读：463964



JavaWeb技术总结

文章：30篇

阅读：438077

注意：

1、我们这些配置信息除了在xml中定义，同样也可以在代码中定义，我们一般都是在onServiceConnected()方法里进行

```
[java]
01.  @Override
02.  protected void onServiceConnected() {
03.      AccessibilityServiceInfo info = getServiceInfo();
04.      info.eventTypes = AccessibilityEvent.TYPES_ALL_MASK;
05.      info.feedbackType = AccessibilityServiceInfo.FEEDBACK_SPOKEN;
06.      info.notificationTimeout = 100;
07.      setServiceInfo(info);
08.      info.packageNames = new String[]{"xxx.xxx.xxx", "yyy.yyy.yyy", "..."};
09.      setServiceInfo(info);
10.      super.onServiceConnected();
11.  }
```

2、这里我们一般都会在这里写上我们需要监听的应用的包名，但是有时候我们需要监听多个应用

我们该怎么办呢？

这时候我们可以这么做：

第一种：我们在代码中注册多个应用的包名，从而可以监听多个应用

Java

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==



文章分类

[Android](#) (166)[Java](#) (36)[JavaWeb](#) (31)[IOS](#) (31)[逆向之旅](#) (38)

文章存档

[2017年09月](#) (2)[2017年08月](#) (3)[2017年07月](#) (5)[2017年06月](#) (2)[2017年05月](#) (2)[展开](#)

阅读排行

[Mac上的抓包工具Charles](#) (300976)[抓包工具Fiddler详解\(主要来抓...\)](#) (133391)[Android中的onActivityResult和...](#) (125755)[Window 通过cmd查看端口占用...](#) (101401)[Android中的Apk的加固\(加壳\)...](#) (88143)

```

07.         super.onServiceConnected();
08.     }

```

第二种：我们在onAccessibilityEvent事件监听的方法中做包名的过滤(这种方式最常用)

```

[java]
01. @Override
02. public void onAccessibilityEvent(AccessibilityEvent event) {
03.     String pkgName = event.getPackageName().toString();
04.     if("xxx.xxx.xxx".equals(pkgName)){
05.
06.     }else if("yyy.yyy.yyy".equals(pkgName)){
07.
08.     }else if("..." .equals(pkgName)){
09.
10.     }
11. }

```

第三步、在onAccessibilityEvent方法中监听指定的事件

比如我们需要监听有通知栏消息的事件：

```

[java]
01. @Override
02. public void onAccessibilityEvent(AccessibilityEvent event) {

```

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==



评论排行

Android工作两年之后的第一个...	(145)
Android中的Apk的加固(加壳)...	(49)
Android逆向之旅---静态方式破...	(41)
Android"挂逼"修炼之行---微信...	(40)
Android中的动态加载机制	(36)
编译Android系统源码和内核源...	(33)
Android中实现静态的默认安装...	(29)
Mac上的抓包工具Charles	(28)
Android中插件开发篇之----动态..	(27)
Android中可以做的两件坏事----	(23)

这个事件类型很多的，我们可以查看AccessibilityEvent类的源码：

```
[java]
01. @Deprecated
02. public static final int MAX_TEXT_LENGTH = 500;
03.
04. /**
05.  * Represents the event of clicking on a {@link android.view.View} like
06.  * {@link android.widget.Button}, {@link android.widget.CompoundButton}, etc.
07.  */
08. public static final int TYPE_VIEW_CLICKED = 0x00000001;
09.
10. /**
11.  * Represents the event of long clicking on a {@link android.view.View} l:
12.  * {@link android.widget.Button}, {@link android.widget.CompoundButton}, (
13.  */
14. public static final int TYPE_VIEW_LONG_CLICKED = 0x00000002;
15.
16. /**
17.  * Represents the event of selecting an item usually in the context of an
18.  * {@link android.widget.AdapterView}.
19.  */
20. public static final int TYPE_VIEW_SELECTED = 0x00000004;
21.
22. /**
23.  * Represents the event of setting input focus of a {@link android.view.View}.
24.  */
25. public static final int TYPE_VIEW_FOCUSED = 0x00000008;
26.
27. /**
28.  * Represents the event of showing the start of a {@link android.widget.EditText}.
```

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==



```
36. public static final int TYPE_WINDOW_STATE_CHANGED = 0x00000020;
37.
38. /**
39.  * Represents the event showing a {@link android.app.Notification}.
40.  */
41. public static final int TYPE_NOTIFICATION_STATE_CHANGED = 0x00000040;
42.
43. /**
44.  * Represents the event of a hover enter over a {@link android.view.View}.
45.  */
46. public static final int TYPE_VIEW_HOVER_ENTER = 0x00000080;
47.
48. /**
49.  * Represents the event of a hover exit over a {@link android.view.View}.
50.  */
51. public static final int TYPE_VIEW_HOVER_EXIT = 0x00000100;
52.
53. /**
54.  * Represents the event of starting a touch exploration gesture.
55.  */
56. public static final int TYPE_TOUCH_EXPLORATION_GESTURE_START = 0x00000200;
57.
58. /**
59.  * Represents the event of ending a touch exploration gesture.
60.  */
61. public static final int TYPE_TOUCH_EXPLORATION_GESTURE_END = 0x00000400;
62.
63. /**
64.  * Represents the event of changing the content of a window and more
65.  * specifically the sub-tree rooted at the event's source.
66.  */
67. public static final int TYPE_WINDOW_CONTENT_CHANGED = 0x00000800;
```

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==




```
75.  * Represents the event of changing the selection in an {@link android.widget.EditText
76.  */
77.  public static final int TYPE_VIEW_TEXT_SELECTION_CHANGED = 0x00002000;
78.
79.  /**
80.   * Represents the event of an application making an announcement.
81.   */
82.  public static final int TYPE_ANNOUNCEMENT = 0x00004000;
83.
84.  /**
85.   * Represents the event of gaining accessibility focus.
86.   */
87.  public static final int TYPE_VIEW_ACCESSIBILITY_FOCUSED = 0x00008000;
88.
89.  /**
90.   * Represents the event of clearing accessibility focus.
91.   */
92.  public static final int TYPE_VIEW_ACCESSIBILITY_FOCUS_CLEARED = 0x00010000;
93.
94.  /**
95.   * Represents the event of traversing the text of a view at a given movement granulari
96.   */
97.  public static final int TYPE_VIEW_TEXT_TRAVERSED_AT_MOVEMENT_GRANULARITY =
98.
99.  /**
100.   * Represents the event of beginning gesture detection.
101.   */
102.  public static final int TYPE_GESTURE_DETECTION_START = 0x00040000;
103.
104.  /**
105.   * Represents the event of ending gesture detection.
106.   */
```

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==



```
114.  /**
115.  * Represents the event of the user ending to touch the screen.
116.  */
117.  public static final int TYPE_TOUCH_INTERACTION_END = 0x00200000;
118.
119.  /**
120.  * Change type for {@link #TYPE_WINDOW_CONTENT_CHANGED} event:
121.  * The type of change is not defined.
122.  */
123.  public static final int CONTENT_CHANGE_TYPE_UNDEFINED = 0x00000000;
124.
125.  /**
126.  * Change type for {@link #TYPE_WINDOW_CONTENT_CHANGED} event:
127.  * A node in the subtree rooted at the source node was added or removed.
128.  */
129.  public static final int CONTENT_CHANGE_TYPE_SUBTREE = 0x00000001;
130.
131.  /**
132.  * Change type for {@link #TYPE_WINDOW_CONTENT_CHANGED} event:
133.  * The node's text changed.
134.  */
135.  public static final int CONTENT_CHANGE_TYPE_TEXT = 0x00000002;
136.
137.  /**
138.  * Change type for {@link #TYPE_WINDOW_CONTENT_CHANGED} event:
139.  * The node's content description changed.
140.  */
141.  public static final int CONTENT_CHANGE_TYPE_CONTENT_DESCRIPTION = 0x00000004;
```

这里有很多事件，这些事件我们通过名字就可以看出来有很多我们可能都知道，比如当窗口发生变化的时候，

当某个View被点去了，被滚动了等消息都是可以知道的。那么，我们有了这些事件我们就可以做我们的事情了。

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==



这里系统提供了两个方法让我们来进行查找想要的节点View

第一种是通过节点View的Text内容来查找

findAccessibilityNodeInfosByText("查找内容")

这种方式查找，就是像TextView,Button等View有文本内容的，可以使用这种方式快速的找到。

第二种是通过节点View在xml布局中的id名称

findAccessibilityNodeInfosById("@id/xxx")

这个一般很难知道，但是我们在查找系统控件的时候还是可以做的，因为系统的控件的id是可以统一的。

(关于这两个方法我们在写网页爬虫程序的时候可能知道，在html中通过tag/name/id等信息可以找原理都类似)

第五步、模拟点击指定事件

我们找到我们想要的View节点，调用方法模拟事件：

performAction(AccessibilityNodeInfo.ACTION_CLICK)

调用这个方法即可，当然这里的参数就是指定事件的名称，这个和AccessibilityEvent中监听的那些事件是一一对应的，这里是模拟点击事件，我们当然可以模拟View的滚动事件，长按事件等。

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==



首先我们来看一下微信抢红包的流程：

第一步、我们在通知栏会接收到一个微信红包的消息

我们监听通知栏事件：

AccessibilityEvent.TYPE_NOTIFICATION_STATE_CHANGED

然后查看通知栏的消息中是否有：[微信红包] 的文本内容

是的话，就走进入第二步

第二步、我们模拟打开通知栏

打开微信如下图：



我们查找包含有：领取红包 的文本内容的节点View,然后模拟点击，进入第三步：

第三步、我们点击领取红包

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==





下面我们来看一下代码中的具体实现：

[java]

@1 package krelve.demo.red;

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==



```
09. import android.app.PendingIntent;
10. import android.app.PendingIntent.CanceledException;
11. import android.util.Log;
12. import android.view.accessibility.AccessibilityEvent;
13. import android.view.accessibility.AccessibilityNodeInfo;
14.
15. public class RobMoney extends AccessibilityService {
16.
17.     @Override
18.     public void onAccessibilityEvent(AccessibilityEvent event) {
19.         int eventType = event.getEventType();
20.         switch (eventType) {
21.             //第一步：监听通知栏消息
22.             case AccessibilityEvent.TYPE_NOTIFICATION_STATE_CHANGED:
23.                 List<CharSequence> texts = event.getText();
24.                 if (!texts.isEmpty()) {
25.                     for (CharSequence text : texts) {
26.                         String content = text.toString();
27.                         Log.i("demo", "text:"+content);
28.                         if (content.contains("[微信红包]")) {
29.                             //模拟打开通知栏消息
30.                             if (event.getParcelableData() != null
31.                                 &&
32.                                 event.getParcelableData() instanceof Notification) {
33.                                 Notification notification = (Notification) event
34.                                     .getParcelableData();
35.                                 PendingIntent pendingIntent = notification.contentIntent;
36.                                 try {
37.                                     pendingIntent.send();
38.                                 } catch (CanceledException e) {
39.                                     e.printStackTrace();
40.                                 }
41.                             }
42.                         }
43.                     }
44.                 }
45.             default:
46.                 //do nothing
47.         }
48.     }
49. }
```

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==



```
48.         if (className.equals("com.tencent.mm.ui.LauncherUI")) {
49.             //开始抢红包
50.             getPacket();
51.         } else if (className.equals("com.tencent.mm.plugin.luckymoney.ui.LuckyMone
52.             //开始打开红包
53.             openPacket();
54.         }
55.         break;
56.     }
57. }
58.
59. /**
60.  * 查找到
61.  */
62. @SuppressWarnings("NewApi")
63. private void openPacket() {
64.     AccessibilityNodeInfo nodeInfo = getRootInActiveWindow();
65.     if (nodeInfo != null) {
66.         List<AccessibilityNodeInfo> list = nodeInfo
67.             .findAccessibilityNodeInfosByText("抢红包");
68.         for (AccessibilityNodeInfo n : list) {
69.             n.performAction(AccessibilityNodeInfo.ACTION_CLICK);
70.         }
71.     }
72.
73. }
74.
75. @SuppressWarnings("NewApi")
76. private void getPacket() {
77.     AccessibilityNodeInfo rootNode = getRootInActiveWindow();
78.     recycle(rootNode);
79. }
```

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==



```
87.         if (info.getChildCount() == 0) {
88.             if(info.getText() != null){
89.                 if("领取红包".equals(info.getText().toString())){
90.                     //这里有一个问题需要注意，就是需要找到一个可以点击的View
91.                     Log.i("demo", "Click"+"isClick:"+info.isClickable());
92.                     info.performAction(AccessibilityNodeInfo.ACTION_CLICK);
93.                     AccessibilityNodeInfo parent = info.getParent();
94.                     while(parent != null){
95.                         Log.i("demo", "parent isClick:"+parent.isClickable());
96.                         if(parent.isClickable()){
97.                             parent.performAction(AccessibilityNodeInfo.ACTION_CLICK);
98.                             break;
99.                         }
100.                        parent = parent.getParent();
101.                    }
102.                }
103.            }
104.        }
105.    } else {
106.        for (int i = 0; i < info.getChildCount(); i++) {
107.            if(info.getChild(i)!=null){
108.                recycle(info.getChild(i));
109.            }
110.        }
111.    }
112. }
113.
114.
115. @Override
116. public void onInterrupt() {
117. }
118.
```

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==




```
[java]
01.  if (content.contains("[微信红包]")) {
02.      //模拟打开通知栏消息
03.      if (event.getParcelableData() != null
04.          &&
05.          event.getParcelableData() instanceof Notification) {
06.          Notification notification = (Notification) event.getParcelableData();
07.          PendingIntent pendingIntent = notification.contentIntent;
08.          try {
09.              pendingIntent.send();
10.          } catch (CanceledException e) {
11.              e.printStackTrace();
12.          }
13.      }
14.  }
```

2、我们在模拟点击通知栏消息之后，还是需要监听：AccessibilityEvent.TYPE_WINDOW_STATE_CHANGED

这个事件，这个事件我们以后会经常用到，这个事件就是在窗口发生改变的时候发出来的事件，例如我们可以通过这个事件来监听TopActivity，然后得到包名，这也是一个实现应用锁的一个原理。

3、我们在查找领取红包的时候，模拟点击的时候做了一个工作，就是从“领取红包”文本的控件View网上查找，查找到一个可以点击的View出来，然后模拟点击

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==



```
07.         while(parent != null){
08.             Log.i("demo", "parent isClick:"+parent.isClickable());
09.             if(parent.isClickable()){
10.                 parent.performAction(AccessibilityNodeInfo.ACTION_CLICK);
11.                 break;
12.             }
13.             parent = parent.getParent();
14.         }
15.     }
16. }
17. }
```

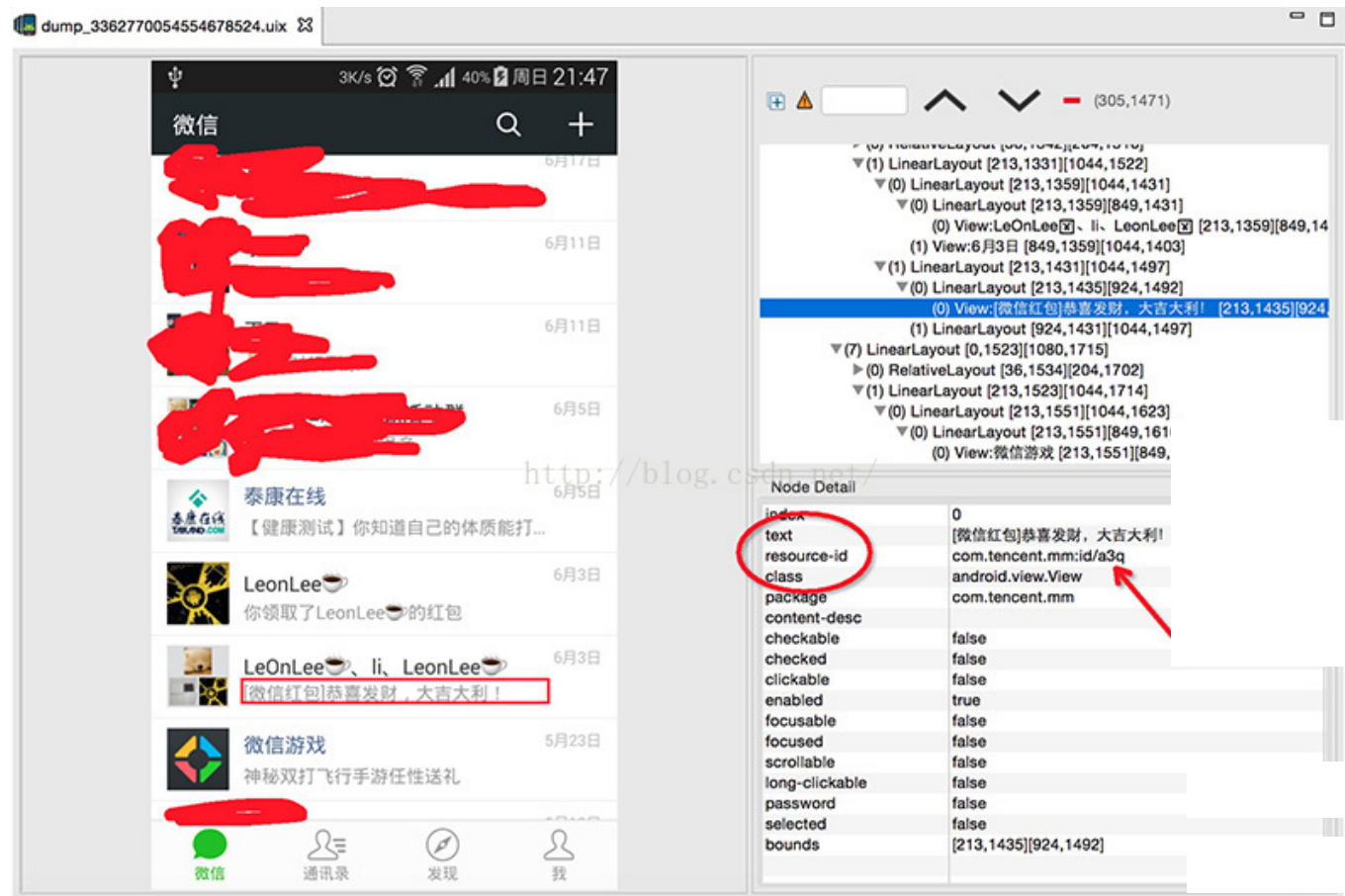
这里为什么这么做，其实原理很简单，因为我们不知道微信他的界面布局，也不知道他对哪个View setOnClickListener。我们可以写一个例子，performAction方法只对调用了setOnClickListener方法点击才有效，其实看View的源码也是可以看出来。这里就不多解释了。所以我们就需要得到一个View，从下往上找，直到找到一个可以click的View为止。

技术延展：

我们其实还可以使用DDMS工具里的Dump View Hierarchy For UI Automator 去分析微信UI结构，这个方法是我后面才发现的，比上面的代码更有效，如下图：

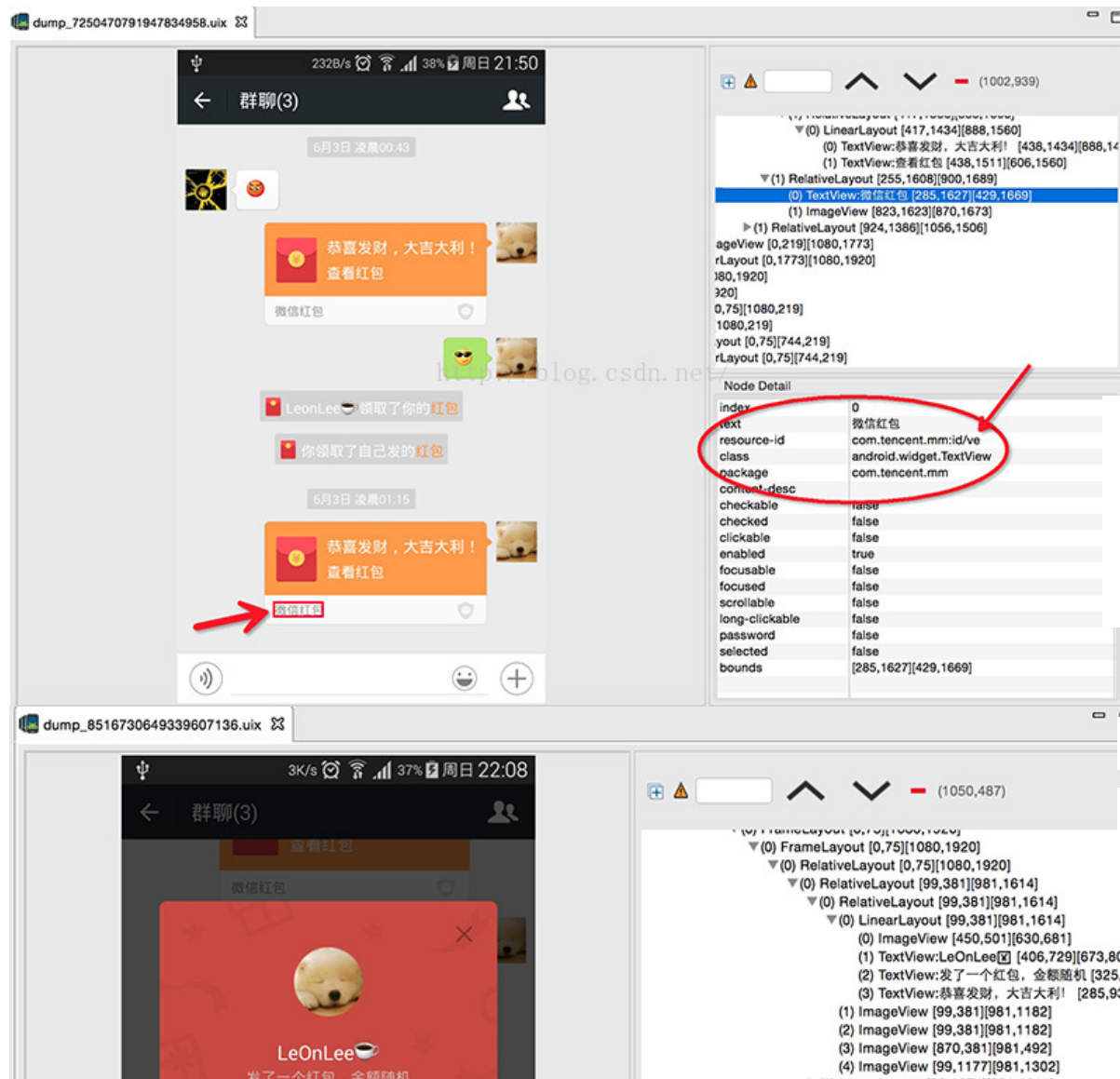
关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==





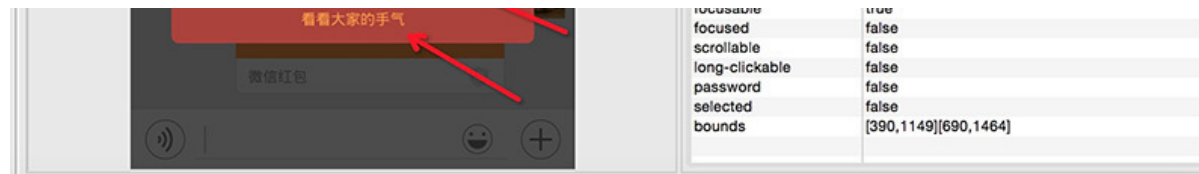
关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==





关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==





这里我们可以看到View的详细布局，还有每个View的属性，还有很重要的信息resource-id，这个就是我们在xml中定义的id，这个id我们也可以使用前面说到的`findAccessibilityNodeInfosById("@id/xxx")`来查找控件了这个也算是学习了，学会使用DDMS来分析View结构。

项目下载：<http://download.csdn.net/detail/jiangwei0910410003/9156347>

四、延展

关于微信抢红包的原理解析上面已经做了分析了，但是要想做到极致，这里还有很多问题的，比如我们还需要过滤一些已经领取过的红包，这样的话效率也是很高的。这个都是算法精确的问题了，我想在这里说的只是我们不仅可以用辅助功能来实现抢红包，还可以实现很多功能，比如

1、静默安装

关于静默安装的实现，之前的做法是，参见这篇文章：

<http://blog.csdn.net/jiangwei0910410003/article/details/36427963>

在这篇文章中我介绍了很多方法来实现静默安装，但是都是有一个限制，那就是root,或者是获取到systemId。但

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==





我们可以监听系统的这个安装界面，然后得到安装节点View,然后模拟点击即可，卸载也是同样的

2、强制停止应用

我们知道Android中停止应用有很多方法，kill进程，stopService,但是这些方法，有一些应用它们都是有对策的，那么我们之前用到的强制停止的方法是获取root权限调用系统的forceStop的api来停止，但是前提还是有root。那么现在如果我们有了辅助功能的话，我们可以这么做：

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==





我们可以监听系统的应用详情页面，然后找到：结束运行的节点View，然后模拟点击即可

当然上面我就说了两个简单的例子，还有很多辅助功能都是可以做的。他的好处就是不需要root权限。但是他也是需要用户授权的：

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==





如果用户没有授权的话，那么所有的工作都没办法开始了，所以说这个方法也不是万能的。当然有了辅助功能的话，他的危险性比root之后的危险性更大，比如我们上面的抢红包插件，其实我们稍作修改，就可以获取微信通讯录信息，微信支付的密码。这些事都是可以做的，所以说，我们在作为用户的时候，进行授权的时候还是需要三思而后行。

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==



情。当然辅助功能是google对于肢体上有障碍的人开发出来的一个功能，我们开发者或许使用这个功能，可以做一些产品的拓展功能，当然这些是google没有想到的事情，但是这个至少是我们开发者在以后的开发道路上的一个解决问题的一个办法和途径，谨记此功能！

PS: 关注微信，最新Android技术实时推送



关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==



- [上一篇](#) Android中实现Activity的启动拦截之----实现360卫士的安装应用界面
- [下一篇](#) Android逆向之旅---SO(ELF)文件格式详解

相关文章推荐

- 自动抢红包，自动安装原理之AccessibilityService
- Android中微信抢红包插件原理解析和开发
- Presto服务治理与架构优化在京东的实践应用--王哲...
- 10小时深入掌握 Kubernetes
- App 控件定位讲解 (hierarchyviewer 篇)
- Android中微信抢红包插件原理解析和开发
- 【免费直播】Python最佳学习路线--韦玮
- JDK9新特性
- MonkenRunner通过HierarchyViewer定位控件的方法...
- Android进阶——学习AccessibilityService实现微信...
- JS-SDK开发与微信支付
- Android中实现静态的默认安装和卸载应用
- 安卓微信自动抢红包插件优化和实现
- 微信抢红包
- Spring Cloud微服务真实场景实战解析
- Android中微信抢红包插件原理解析和开发

查看评论



yishulin520

17楼 2016-03-29 11:58发表

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==





妞妞妹

15楼 2016-02-04 17:22发表

楼主,下载了一下你的源码,发现不能打开,这是为什么啊?



JayZhaoPLus

14楼 2016-02-02 16:05发表

好像在锁屏状态下不可以拆开红包啊，楼主有什么解决办法吗？求赐教！



维度博客

13楼 2016-01-31 23:

楼主你好！我按照你的方法写，服务已经创建并启动了，响应事件设的是"typeAllMask"，并且没有设android:packageNames，按理来QQ或者微信通知栏状态变时应该能检测到啊，怎么我的onAccessibilityEvent这个回调就是没有被调用呢？想问楼主到底是什么原因？



zuiniao123

12楼 2016-01-28 14:4

大哥我想问下如何监听系统的应用详情页面



smo

11楼 2016-01-26 17:39发表

如何做到锁屏状态下，抢到红包呢？

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==



网上说的透视，避免小包，怎么做到呢



二氧化碳被占用

8楼 2015-11-17 10:02发表

你好，请问一下，像微信的页面路径跟名称“com.tencent.mm.ui.LauncherUI”这种是怎么获取的呢



快乐的kuai小马

7楼 2015-11-04 21:00发表

楼主,感谢您的分享,想请教一个问题: 如何过滤一些已经领取过的红包呢?



zhengfei1992

6楼 2015-11-03 13:00发表

赞个



勤奋的主人公

5楼 2015-10-08 09:14发表

很不错，学习！



yezi19910408

4楼 2015-10-07 09:59发表

多谢分享

关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==





CoolEgos

2楼 2015-10-05 21:20发表

好吧，这样基本的实现很早就有了，过滤已经抢过的，以及如何保证抢得到好像更有意思，哈哈



丁国华

1楼 2015-10-05 08:48发表

谢谢小伙伴的分享，学习了`(*_*)`

该文章已被禁止评论！

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

公司简介 | 招贤纳士 | 广告服务 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服

杂志客服

微博客服

webmaster@csdn.net

400-660-0108

| 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 | 江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved



关注更多技术内容&交流问题，关注微信公众号“编码美丽”，扫描即可关注==

