

Android 静态代码扫描流程及工具说明

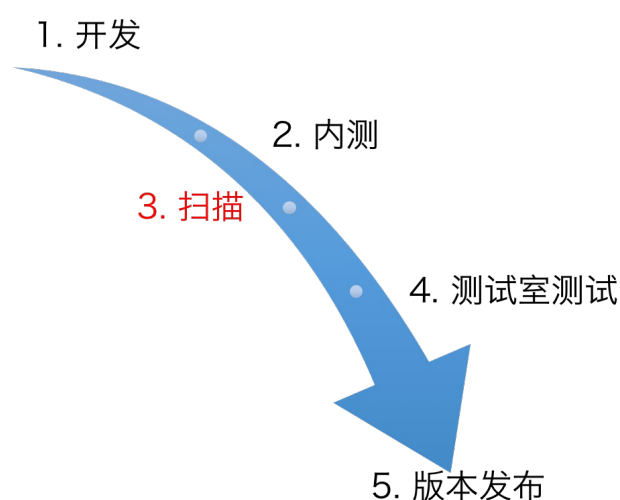
0. 更新说明

撰写人	版本	描述
马季	0.1	初稿
马季	0.2	增加典型案例说明、发布流程能内容

1. 静态扫描流程

1.1 版本发布流程

大致分为5个阶段，静态代码扫描的工作在第3步进行，如图：



1.2 典型案例分析

以下主要针对支付控件PluginBase模块的扫描结果进行案例分析，涵盖内容如下：

- [空指针]空指针引用
- [内存泄露]Stream资源关闭
- [性能]使用indexOf(字符)

- [兼容]系统API兼容性隐患
- [越界]数组下标越界隐患
- [异常] 使用除法或求余没有判断分母长度隐患
- [SQL]注入风险
- [应用安全] AndroidManifest.xml文件中allowBackup设置为true时会导致数据泄露

更多的错误检查示例请查看各检查工具的检查规则说明文档。

1.2.1 [空指针]空指针引用

所在类: `com.unionpay.mobile.android.utils.StringUtil`

方法名: `queryParams`

错误位置: 4

```
public class StringUtil {
    public static final String queryParams(String
param) {
        String ret = "";
        if (param != null || param.length() > 2) {
            ret = param.substring(1, param.length() - 1);
        }
        return ret;
    }
}
```

存在空指针引用,会导致空指针异常。解决方案:

```
public class StringUtil {
    public static final String queryParams(String
param) {
        String ret = "";
        if (param != null && param.length() > 2) {
            ret = param.substring(1, param.length() - 1);
        }
        return ret;
    }
}
```

```
    }  
    return ret;  
}
```

1.2.2 [内存泄露]Stream资源关闭

所在类: `com.unionpay.mobile.android.utils.UPLogUtil`

方法名: `write2logfile`

错误位置: 17

```
private static void write2logfile(String msg) {  
    try {  
  
        File sdCardDir = android.os.Environment  
            .getExternalStorageDirectory();  
  
        File logfile = new  
File(sdCardDir.getAbsolutePath()  
        + File.separator + logfileName);  
  
        if (!logfile.exists()) {  
            logfile.createNewFile();  
        }  
  
        msg += "\n";  
  
        FileOutputStream outputStream = new  
FileOutputStream(logfile, true);  
        outputStream.write(msg.getBytes());  
        outputStream.close();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

资源对象在被关闭或者Return之前可能出现异常，导致无法正常关闭或Return。比如连续关闭多个资源对象时没有进行异常捕获，或者资源对象在Return之前进行了未捕获异常的操作。解决方案：

```
private static void write2logfile(String msg) {
    try {

        File sdCardDir = android.os.Environment
            .getExternalStorageDirectory();

        File logfile = new
File(sdCardDir.getAbsolutePath()
        + File.separator + logfileName);

        if (!logfile.exists()) {
            logfile.createNewFile();
        }

        msg += "\n";

        FileOutputStream outputStream = new
FileOutputStream(logfile, true);
        outputStream.write(msg.getBytes());
    } catch (IOException e) {
        e.printStackTrace();
    } finally{
        if(null != outputStream){
            outputStream.close();
        }
    }
}
```

1.2.3 [性能]使用indexOf(字符)

所在类：[com.unionpay.mobile.android.utils.DeviceInfoUtil](#)

方法名: `getLinuxCoreVer`

错误位置: 340

```
338             int index =
result.indexOf(Keyword);
339             line =
result.substring(index + Keyword.length());
340             index = line.indexOf("
");
341             kernelVersion =
line.substring(0, index);
342         }
343     } catch
(IndexOutOfBoundsException e) {
```

当你检测单个字符的位置时使用String.indexOf(字符), 它执行的很快。解决方案: 不要使用indexOf(字符串)。

```
340             index = line.indexOf('
');
```

1.2.4 [兼容]系统API兼容性隐患

所在类: `com.unionpay.mobile.android.utils.DeviceInfoUtil`

方法名: `getSupportMap`

错误位置: `getDefaultAdapter`方法调用

```
public static String getSupportMap(Context context,
String seInfo) {
    StringBuffer support = new StringBuffer("000");
```

```

        if (!"000".equals(seInfo)) {
            support.setCharAt(2, '1');
        }

        if (VERSION.SDK_INT < 10) {
            return support.toString();
        }

        NfcManager manager = (NfcManager) context
            .getSystemService(Context.NFC_SERVICE);
        NfcAdapter adapter = manager.getDefaultAdapter();
        if (null == adapter) {
            return support.toString();
        } else {
            if (adapter.isEnabled()) {
                support.setCharAt(0, '1');
            } else {
                support.setCharAt(0, '2');
            }
        }

        if (VERSION.SDK_INT >= 19) {
            PackageManager pm =
context.getPackageManager();
            boolean hasNfcHce = pm

.hasSystemFeature(PackageManager.FEATURE_NFC_HOST_CAR
D_EMULATION);
            if (hasNfcHce) {
                support.setCharAt(1, '1');
            }
        }

        return support.toString();
    }
}

```

getDefaultAdapter方法不支持:10(android2.3.3) 以下的版本。 解决方案：加入对版本的系统版本的判别

```
if (Build.VERSION.SDK_INT >=
    Build.VERSION_CODES.GINGERBREAD_MR1) {
    // 包含新API的代码块
} else
{
    // 包含旧的API的代码块
}
```

1.2.5 [越界]数组下标越界隐患

所在类: `com.unionpay.mobile.android.utils.UPPopupPromotion`

方法名: `getSupportMap`

错误位置: `doSelectItem`方法调用

```
private void doSelectItem(int pos) {

    mButtonViews[mSelectedButtonIndex].line.setVisibility
    (View.GONE);

    mButtonViews[mSelectedButtonIndex].buttonText.setText
    Color(Color.BLACK);

    mButtonViews[mSelectedButtonIndex].contentView.setVis
    ibility(View.GONE);

    mButtonViews[pos].line.setVisibility(View.VISIBLE);
    mButtonViews[pos].buttonText

    .setTextColor(KDimens.K_COLOR_PROM_INDICATOR);

    mButtonViews[pos].contentView.setVisibility(View.VISI
    BLE);
    mSelectedButtonIndex = pos;
```

```
}
```

采用下标的方式获取数组元素时，如果下标越界，将产生 `java.lang.ArrayIndexOutOfBoundsException` 的异常，导致app出现Crash。
解决方案：在使用下标的方式获取数组元素时，需判断下标的有效性。

1.2.6 [异常] 使用除法或求余没有判断分母长度隐患

```
public static Drawable zoomDrawable(Context context,
Drawable in, int scaledW, int scaledH) {
    Drawable zoomed = null;
    if (in instanceof BitmapDrawable) {
        Bitmap bm = ((BitmapDrawable) in).getBitmap();
        if (scaledH != -1 && scaledW == -1) {
            scaledW = (int) ((float) (bm.getWidth() /
bm.getHeight())) * scaledH;
        } else if (scaledH == -1 && scaledW != -1) {
            scaledH = (int) ((float) (bm.getHeight() /
bm.getWidth())) * scaledW;
        }

        Bitmap sbm = Bitmap.createScaledBitmap(bm,
scaledW, scaledH, true);
        zoomed = new
BitmapDrawable(context.getResources(), sbm);
    }
    return zoomed;
}
```

使用除法或者求余运算时，如果分母是通过调用函数返回的int，未对返回值进行判断，当返回值为0时，会出现 `java.lang.ArithmeticException: / by zero` 异常。解决方案：调用函数前，对函数的返回值的长度进行判断。

1.2.6 [SQL]注入风险

描述：对Content Provider进行增删改查操作时，程序没有对用户的输入进行过滤，未采用参数化查询的方式，可能导致sql注入攻击。

代码示例：

```
private SQLiteDatabase db;  
db.rawQuery("select * from person", null); //触发规则
```

推荐写法：

1. 服务端充分校验参数
2. 使用参数化查询，比如SQLiteStatement
3. 避免使用rawQuery()方法
4. 对用户输入进行过滤

```
SQLiteStatement sqliteStatement =  
db.compileStatement("insert into msgTable(uid, msg)  
values(?, ?)");  
sqliteStatement.bindLong(1, 12);  
sqliteStatement.bindString(3, "text");  
long newRowId = sqliteStatement.executeInsert();
```

1.2.7 [应用安全] AndroidManifest.xml文件中allowBackup设置为true时会导致数据泄露

描述：建议将AndroidManifest.xml文件android:allowBackup属性设置为false。当allowBackup标志值为true时，攻击者可通过adb backup和adb restore来备份和恢复应用程序数据。

推荐写法：

描述：建议将AndroidManifest.xml文件android:allowBackup属性设置为false。当allowBackup标志值为true时，攻击者可通过adb backup和adb

restore来备份和恢复应用程序数据。

推荐写法：

1. minSdkVersion不低于9。
2. android:allowBackup属性显示设置为false。

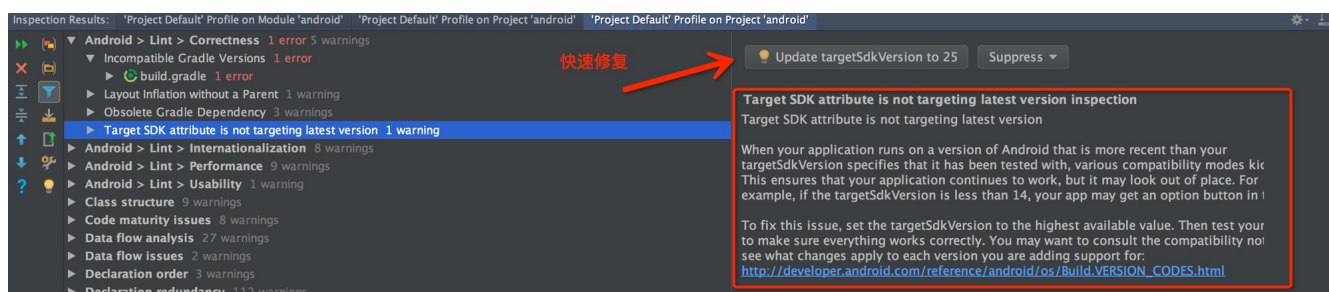
1.3 Android 客户端扫描流程

分为3个阶段：



1.3.1 基础内容扫描

使用 **Android Lint** 对项目进行扫描，该工具已将扫描到的问题进行了分组，同时定义了问题的严重级别：**error** 和 **warning**。在 **Android Studio** 的 **Inspection Results** 视图窗中，点击问题标题即可在右边的详情视图中查看该问题的具体解释等内容，针对部分内容还有直接进行自动修复的按钮，如图：



关注点

由于检查的内容繁多，我们重点关注以下几个**问题组**的相关内容：

- 以 **Android** 开头的组，例如
 - **Android > Lint > Correctness** (可能影响程序正确性)
 - **Android > Lint > Performance** (可能影响程序性能)
 - **Android > Lint > Security** (可能影响程序安全性)
 - 等等
- **Class structure** 组：指出类的设计上可能存在的问题
- **Code style issues** 组：有助于提供代码书写规范
- **Probable bugs** 组：有助于发现隐藏的问题

检查通过标准

上述的列出的问题组别不出现或者出现但只包含**warning**类型的问题

1.3.2 可能引起Crash的问题扫描

使用**360火线**和**Godeyes**对项目进行扫描，下面将分别说明二者扫描时的关注点和检查通过标准：

360火线

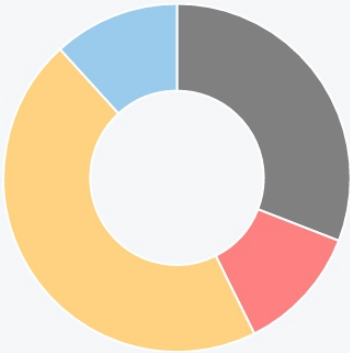
360火线共有61个检查项，按级别分为**Block**、**风险**、**建议**和**优化**，检查报告以html文件输出，在**按规则分类查看**的Tab中，可以查看具体问题位置及示例代码。如图

火线扫描报告

火线汇总报告

按规则分类查看

错误概况



Block

21

风险

8

建议

31

优化

8

项目概况

扫描项目路径	D:\test\TestCase
开始时间	2016-09-23 17:31:20
结束时间	2016-09-23 17:31:21
测试耗时	1秒16毫秒
项目名称	test
提交人	tester
扫描文件数	12个
扫描代码行数	834行
问题咨询联系人	火线用户QQ群: 加入QQ群 问题反馈邮箱: g-qtest-fankui@360.cn

Show 25 entries

Search:

风险类型	规则名称	等级	错误数量	规则描述	查看详情
数据泄露	应用程序备份恢复隐患	风险	1	建议将AndroidManifest.xml文件android:allowBackup属性设置为false。当allowBackup标志值为true时,攻击者可通过adb backup和adb restore来备份和恢复应用程序数据。	
<p>所在位置: D:\branch0224\PluginBase\build\intermediates\manifests\laapt\release\AndroidManifest.xml 元素名: application 错误元素位置: 15</p> <pre>8 android:minSdkVersion="14" 9 android:targetSdkVersion="19" /> 10 11 <application 12 android:allowBackup="true" 13 android:icon="@drawable/ic_launcher" 14 android:label="@string/app_name" 15 android:theme="@style/AppTheme" > 16 </application> 17 18 </manifest></pre>					
空指针异常	存在空指针引用	风险	1	此位置存在空指针引用,会导致空指针异常。	
空指针异常	破坏空判断	风险	1	如果自身抛出空指针异常空检查就会遭到破坏,比如你使用 代替 &&,反之亦然,如 (string!=null !string.equals("")){}。	

关注点

重点关注Block、风险两类标记的问题

检查通过标准

扫描结果中不出现Block、风险两类问题

Godeyes

Godeyes共检查23个错误,扫描结果以html文档的方式输出,文档中包含了检查

问题的描述、示例以及推荐方案，方便理解。值得注意的是在扫描结果的显示上，报告只会给出问题所在的行号。如图

GodEyes 扫描结果

扫描规则	扫描结果
dismiss()方法调用前isShowing未判断的隐患	22
数组下标越界隐患	19
格式化数字异常未捕获的隐患	6
使用String.substring前未判断String长度隐患	4
使用String.split结果未判断长度隐患	2
使用除法或求余没有判断分母长度隐患	2
系统API兼容性隐患	1
使用IO流后没有关闭导致OOM隐患	1
ArrayList使用get方法获取元素未判断下标有效性的隐患	1
复写生命周期函数没有调用super函数隐患	0
主动抛出异常未捕获处理隐患	0
大图片解析导致OOM的隐患	0
通过HashMap获取对象使用未判空隐患	0
Activity未在AndroidManifest.xml中注册隐患	0
销毁Dialog前是否isShowing未判断隐患	0
查询数据库没有关闭游标导致OOM的隐患	0
ArrayList对象使用未判空隐患	0
使用Bundle与使用从Bundle获取到的数据未判空隐患	0
方法中存在return null返回对象直接进行方法调用隐患	0
使用动态载入界面的元素未判断是否属于此界面的隐患	0
使用在intent中获取的数据前未判空隐患	0
添加Fragment前未判断是否IsAdded隐患	0
数据库操作异常未捕获处理隐患	0

1 Android

1.1 【NullPointerException】 dismiss()方法调用前isShowing未判断的隐患

1.1.1 摘要

问题：在调用系统的dismiss()方法前，没有对状态进行判断，导致抛出NullPointerException异常。
解决方案：在调用系统的dismiss()方法时，需要对状态先进行判断。

1.1.2 示例

在下面代码中，popupWindow.dismiss()调用时出现了NullPointerException。

```
if (mContext != null && !((Activity)mContext).isFinishing())
{
    popupWindow.dismiss();
    setFocusable(false);
}
```

1.1.3 推荐方案

在调用系统的dismiss()方法前先进入isShowing的判断，修改如下：

```
if (mContext != null && !((Activity)mContext).isFinishing() && isShowing())
{
    popupWindow.dismiss();
    setFocusable(false);
}
```

1.1.4 扫描结果

Java文件名	行号
com.unionpay/mobile/android/widgets/UPWidgetKeyBoard.java	149
com.unionpay/mobile/android/widgets/UPPromotionWidgetHalf.java	46,55,70,86,97
com.unionpay/mobile/android/widgets/UPAreaCodeWidget.java	66,78
com.unionpay/mobile/android/widgets/UPDropDownWidget.java	53,66
com.unionpay/mobile/android/ui/UPStyleWindow.java	132
com.unionpay/mobile/android/widgets/UPWidgetKeyboardSimple.java	157
com.unionpay/mobile/android/widgets/UPPromotionItem.java	68,77,92,103,119
com.unionpay/mobile/android/widgets/UPPromotionWidget.java	46,61,77
com.unionpay/mobile/android/widgets/UPCertTypeWidget.java	62,74

关注点

所有列表检查出的问题。

各扫描项扫描结果为0

1.3.3 空指针和资源泄露扫描

使用Infer工具对可能的空指针和可能的资源泄露进行扫描，Infer工具会在项目的根文件夹下生成infer-out的文件夹，重点关注bugs.txt这个文件，文件中会详细指出可能存在的问题的代码片段及相应的解释，示例如下：

Found 70 issues

PluginBase/src/com/unionpay/mobile/android/resource/ResourceManager.java:219: error: RESOURCE_LEAK

resource of type `java.io.DataInputStream` acquired to `dis` by call to `new()` at line 159 is not released after line 219

****Note****: potential exception at line 164

```
217.                dis.close();
218.                is.close();
219. >                } catch (IOException e) {
220.                    e.printStackTrace();
221.                    dr = null;
```

PluginBase/src/com/unionpay/mobile/android/upviews/UPRuleView.java:402: error: NULL_DEREFERENCE

object returned by `getItemByName("instalment")` could be null and is dereferenced at line 402

```
400.                } else {
401.                    ((UPDropDownWidget)
getItemByName(Rules.TYPE_INSTALMENT))
402. >                    .setmCanShow(true);
403.                    ((UPDropDownWidget)
getItemByName(Rules.TYPE_INSTALMENT))
404.                    .onCheckBoxStatusChanged(true);
```

关注点

所有列表检查出的问题。

检查通过标准

对检查的问题尽量修复或者编写保护语句避免抛出异常。

2. 工具使用

以下内容均以 **Android Studio** 为默认的开发环境

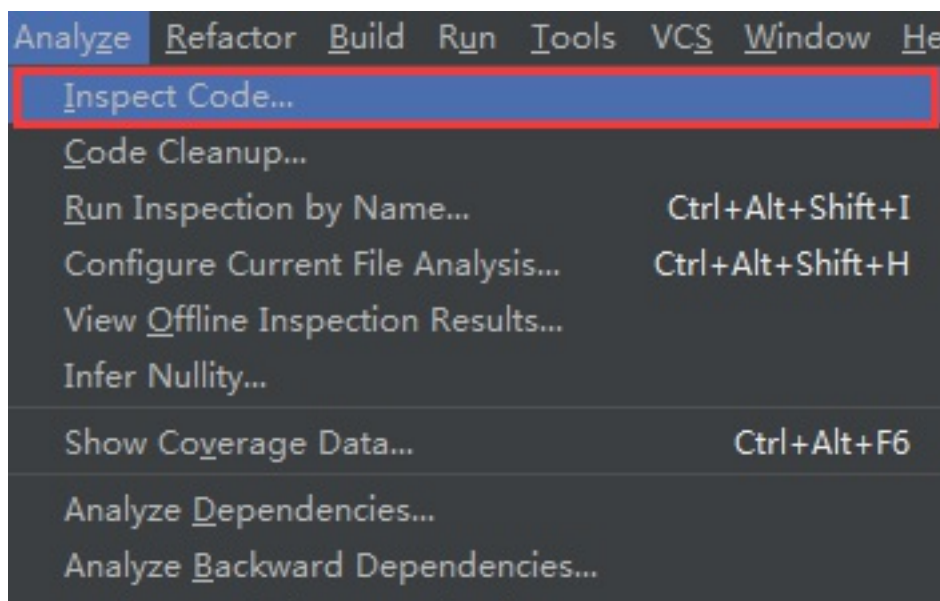
2.1 Android Lint

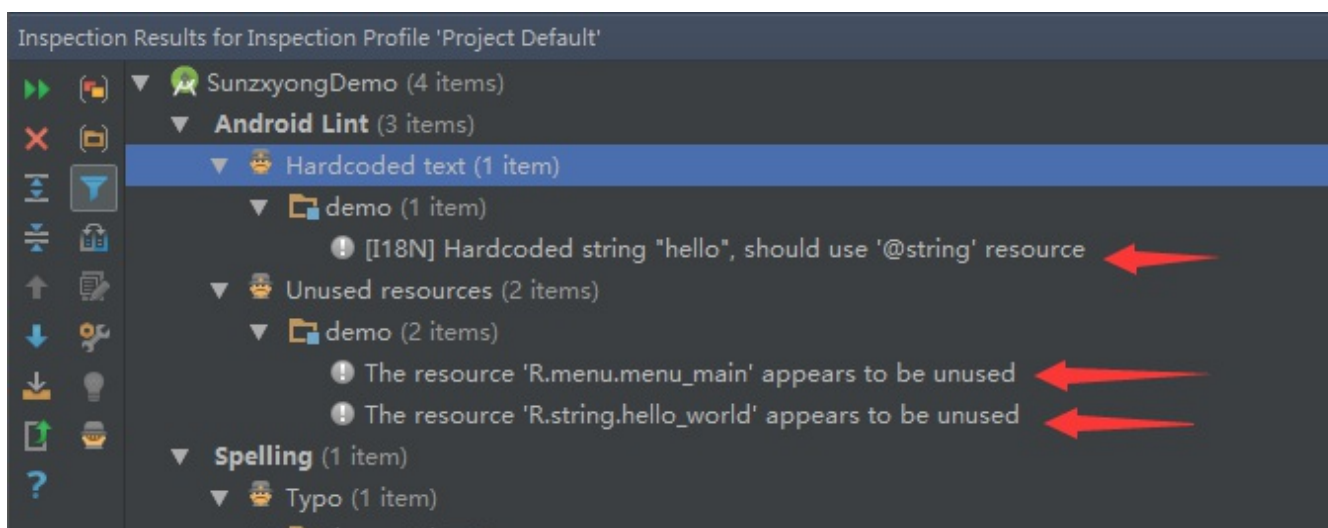
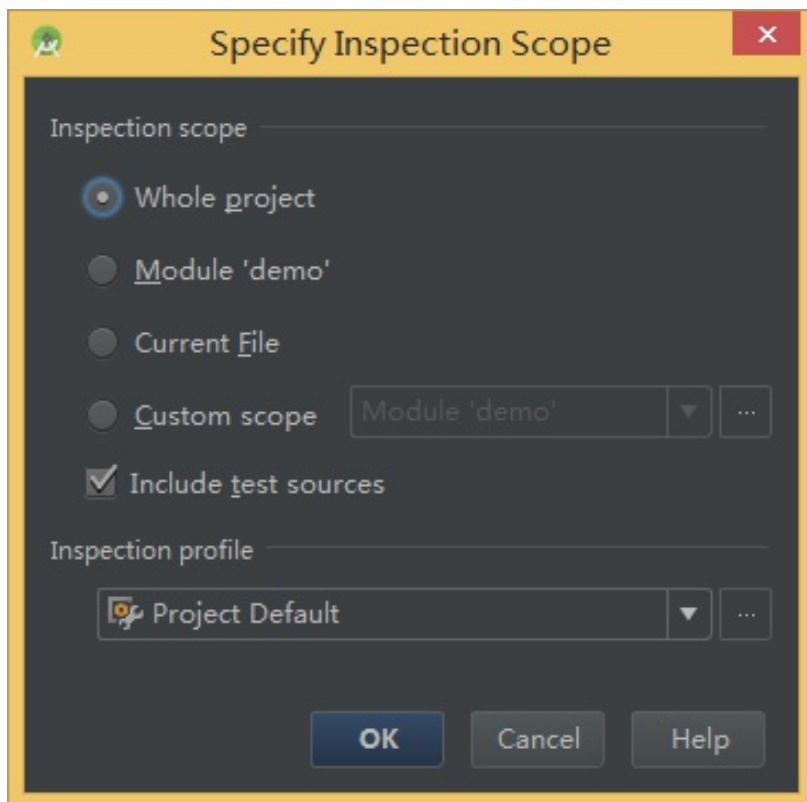
该工具已经默认集成 **Android Studio** 中

使用方法：

Android Studio -> 菜单栏 -> **Analyze** -> **Inspect Code** -> 根据需要选择相应的扫描范围 -> **OK** -> 启动扫描

如图：





参考资料

- <http://tools.android.com/tips/lint/>(需要翻墙)
- <http://www.bubuko.com/infodetail-1055648.html>
- [android-studio-中使用-lint](#)
- [Android APK瘦身之Android Studio Lint \(代码审查\)](#)

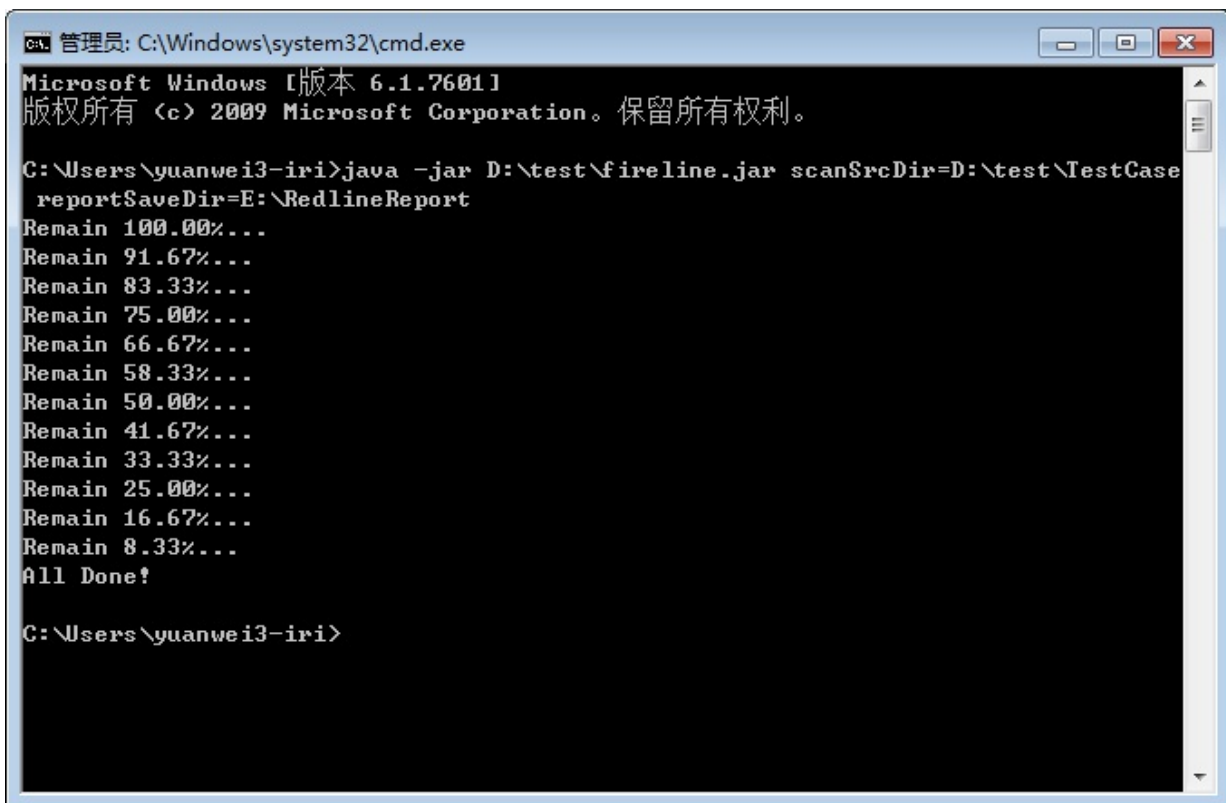
2.2 360 火线

使用方法

详情参考 [官方使用方法](#) 火线插件目前可以在**Android Studio**中进行在线搜索安装。

- jar包版本使用

```
java -jar D:\test\fireline.jar -
s=D:\test\TestCase -r=E:\RedlineReport
// 参数解释:
// 【必填项】-s或scanSrcDir为被扫描的项目工程路径
// 【必填项】-r或reportSaveDir为火线报告输出路径
```



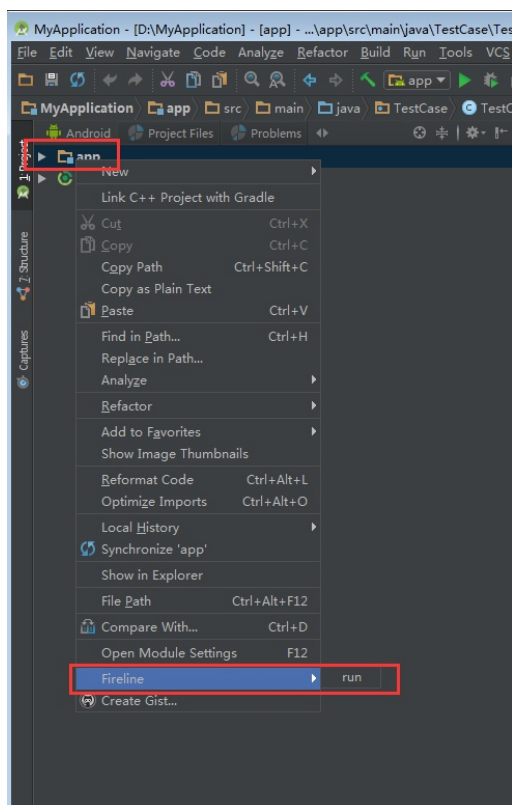
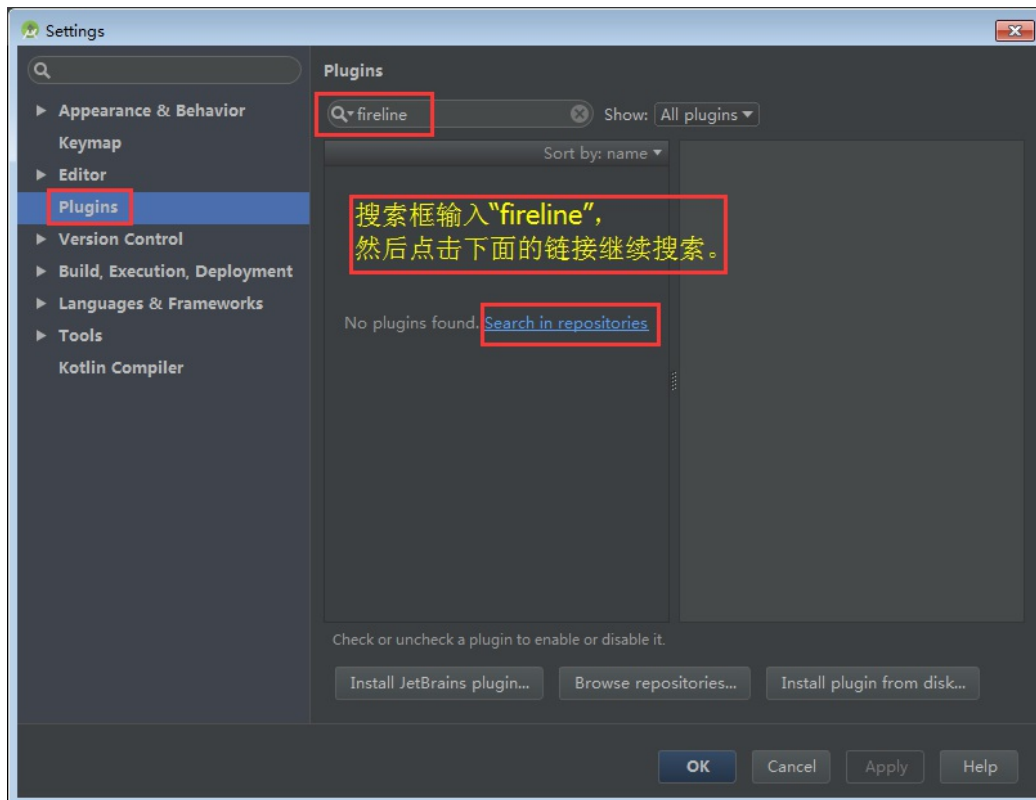
```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\yuanwei3-iri>java -jar D:\test\fireline.jar scanSrcDir=D:\test\TestCase
reportSaveDir=E:\RedlineReport
Remain 100.00%...
Remain 91.67%...
Remain 83.33%...
Remain 75.00%...
Remain 66.67%...
Remain 58.33%...
Remain 50.00%...
Remain 41.67%...
Remain 33.33%...
Remain 25.00%...
Remain 16.67%...
Remain 8.33%...
All Done!

C:\Users\yuanwei3-iri>
```

- Android studio版本

1. **Android Studio** -> 菜单栏 -> **File** -> **Settings...** -> **Plugins**
2. 搜索框 -> 搜索**fireline** -> **install** -> 重启
3. 使用 -> Project视图 -> 鼠标右键 -> **fireline** -> **run** 生成报告



参考资料

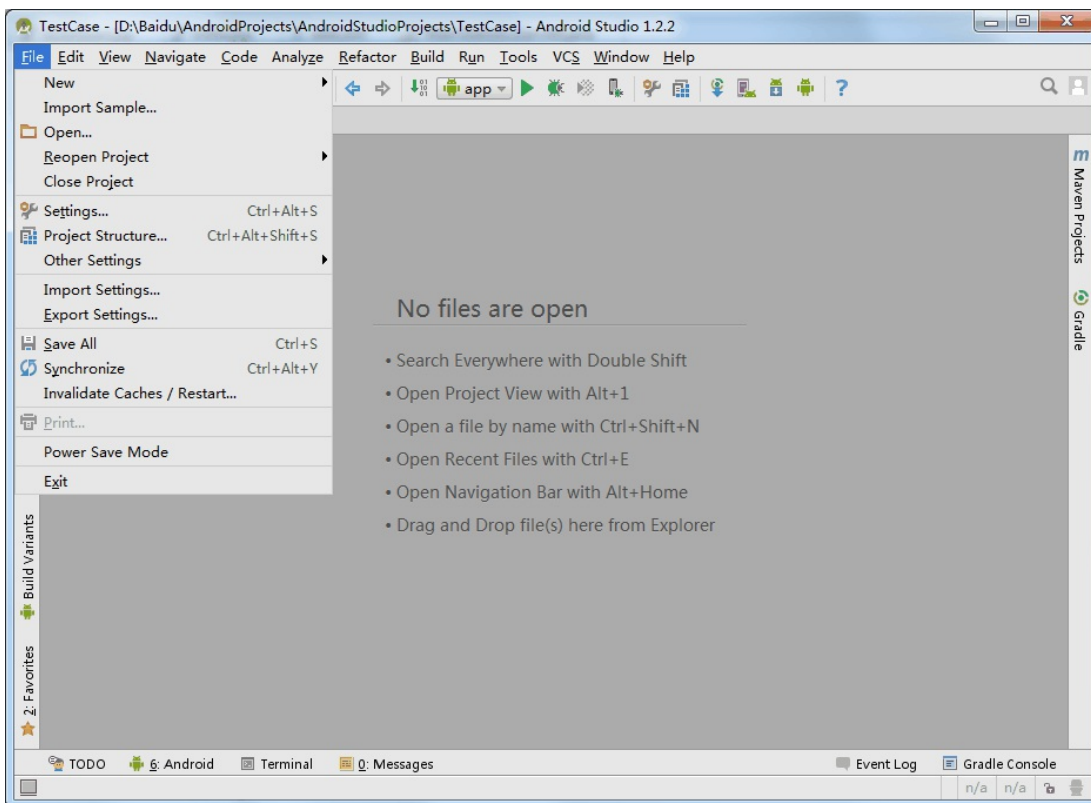
- <http://magic.360.cn/>

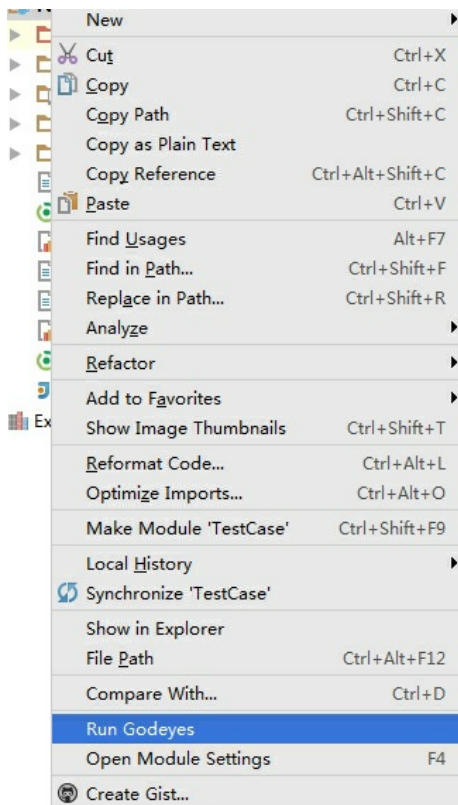
2.3 Godeyes

使用方法

详情参考 [官方使用方法](#)

1. 下载 Android Studio版本插件 [下载地址](#)
2. **Android Studio** -> 菜单栏 -> **File** -> **Settings** -> **Plugins**
3. 选择**Install plugin from disk** -> 选择已下载的
Godeyes_Android_Vx.x_(for_AndroidStudio).zip -> **OK** -> 安装完成
-> 重启
4. **Project**视图 -> 鼠标右键 -> **Run Godeyes** -> 生成报告





参考资料

- <http://godeyes.duapp.com/>
- http://blog.csdn.net/xwh_1230/article/details/51312847

2.4 Infer

注意:

1. 仅支持 Mac 和 Linux 环境
2. 需要Python 且 Python ≥ 2.7

使用方法

- 安装, 请参考 <https://infer.liaohuqiu.net/docs/getting-started.html>
- 使用方法

```
cd {项目的根目录}
./gradlew clean
infer -- ./gradlew build
```

一段时间后会在项目的根目录下生成infer-out这个文件夹，里面的bugs.txt文档里记录的就是扫描出的问题。

参考资料

- <https://infer.liaohuqiu.net/>
- <http://blog.csdn.net/itfootball/article/details/46474235>
- <https://github.com/facebook/infer/blob/master/INSTALL.md>
- <https://github.com/facebook/infer/releases>