## 禁止Activity的启动：

从应用程序的启动过程可以看出，启动应用的第一个Activity经过ActivityThread:performLaunchActivity()，事实上启动任何一个Activity都经过此函数，我们可以在这个函数里添加禁用特定Activity的代码：

可以在大约ActivityThread.java的行1538处的try语句块的开始加上如下类似代码：

………

Log.v("jipengfei:ActivityThread:1541",

"see componet class name:"+component.getClassName(), null);

if(component.getClassName().equals("com.android.accenture.webclip3.Main")){

throw new RuntimeException(

"Unable to instantiate activity " + component

+ ": " + component.getClassName(), null);

}

java.lang.ClassLoader cl = r.packageInfo.getClassLoader();

activity = mInstrumentation.newActivity(

cl, component.getClassName(), r.intent);

………….

这里的component里有需要启动的Activity的详细信息，可以取得此Activity的类全名(包括包名和方法名，可以在此处与我们禁用的Activity进行匹配，如果匹配成功可以采取这种抛出异常的方式来终止启动过程，具体的异常类型可以由我们自己统一定义)

# 禁止应用程序的启动

只要在应用程序的路径上进行设坎，就能禁止应用启动，下面仅列出这两种方法：

1. 在为应用程序建立进程的地方就开始根据packageName进行过滤：

在ActivityManagerService的startProcessLocked函数里 行1739：对processName进行判断(example):

final ProcessRecord startProcessLocked(String processName,

ApplicationInfo info, boolean knownToBeDead, int intentFlags,

String hostingType, ComponentName hostingName, boolean allowWhileBooting) {

Log.v("jipengfei:AMS:startProcessLocked", "processName:"+processName+" info:"+info, null);

//test start

if(info.processName == "com.android.accenture.webclip"){

Log.v("jipengfei:ActivityManagerService", "WebClip interrupt", null);

return null;

}

2.在ActivityManagerService的startProcessLocked函数里 行1881：对processName进行判断(example):

if(app.processName == "com.android.accenture.webclip"){

app.pid = 0;

RuntimeException e = new RuntimeException(

"Failure starting process " + app.processName

+ ": returned pid=" + app.pid);

throw e;

}

//这里开始为应用程序创建进程，从这里也可以看出，每个进程的主线程是ActivityThread

**int pid = Process.start("android.app.ActivityThread",**

**mSimpleProcessManagement ? app.processName : null, uid, uid,**

**gids, debugFlags, null);**

这里可以根据需要直接返回或者选择抛出一个自定义的异常。

注：1. 这里的processName是在AndroidManifest.xml中定义的Android:process的内容，如果此标签未指定，则等同于packageName.

2. 修改后的ActivityManagerService.java文件与此文档在同一目录下，随着调研的深入，该文件可能会被继续修改。