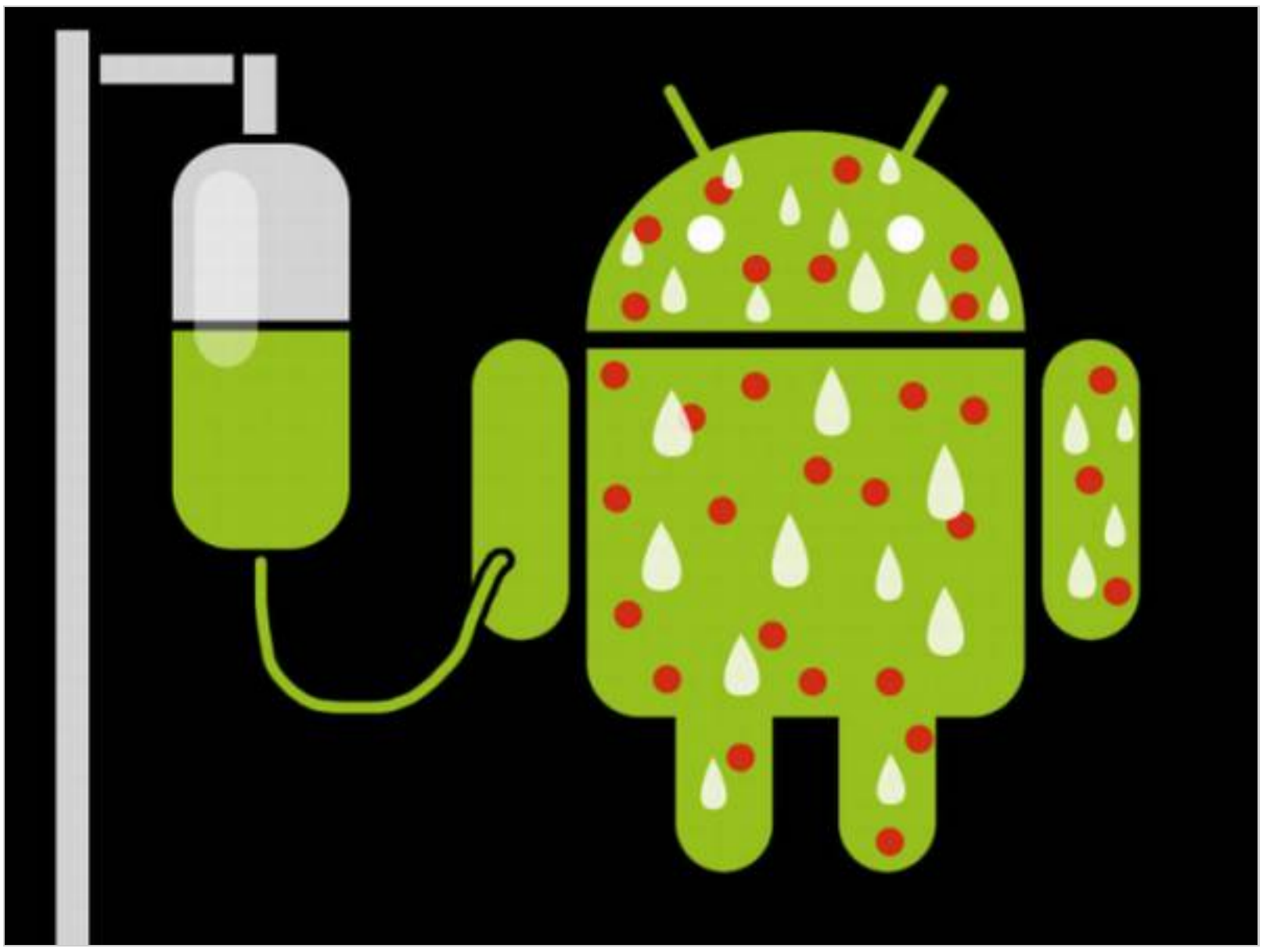
无所不用其极！盘点流氓App驻留后台的奇异手段

脚本之家 2017-09-25

和iOS相比，安卓是一个更加开放的平台，带来了更多自由的同时，也容许了更多“法外之地”。App在安卓系统中的限制比iOS更小，这令安卓孕育出了很多流氓App。尽管Google一直尝试对后台增添更多限制、理清安卓的生态，但流氓App们见招拆招，使出了各种丧心病狂的手段来驻留后台。流氓App驻留后台有哪些技巧？一起来看看吧。



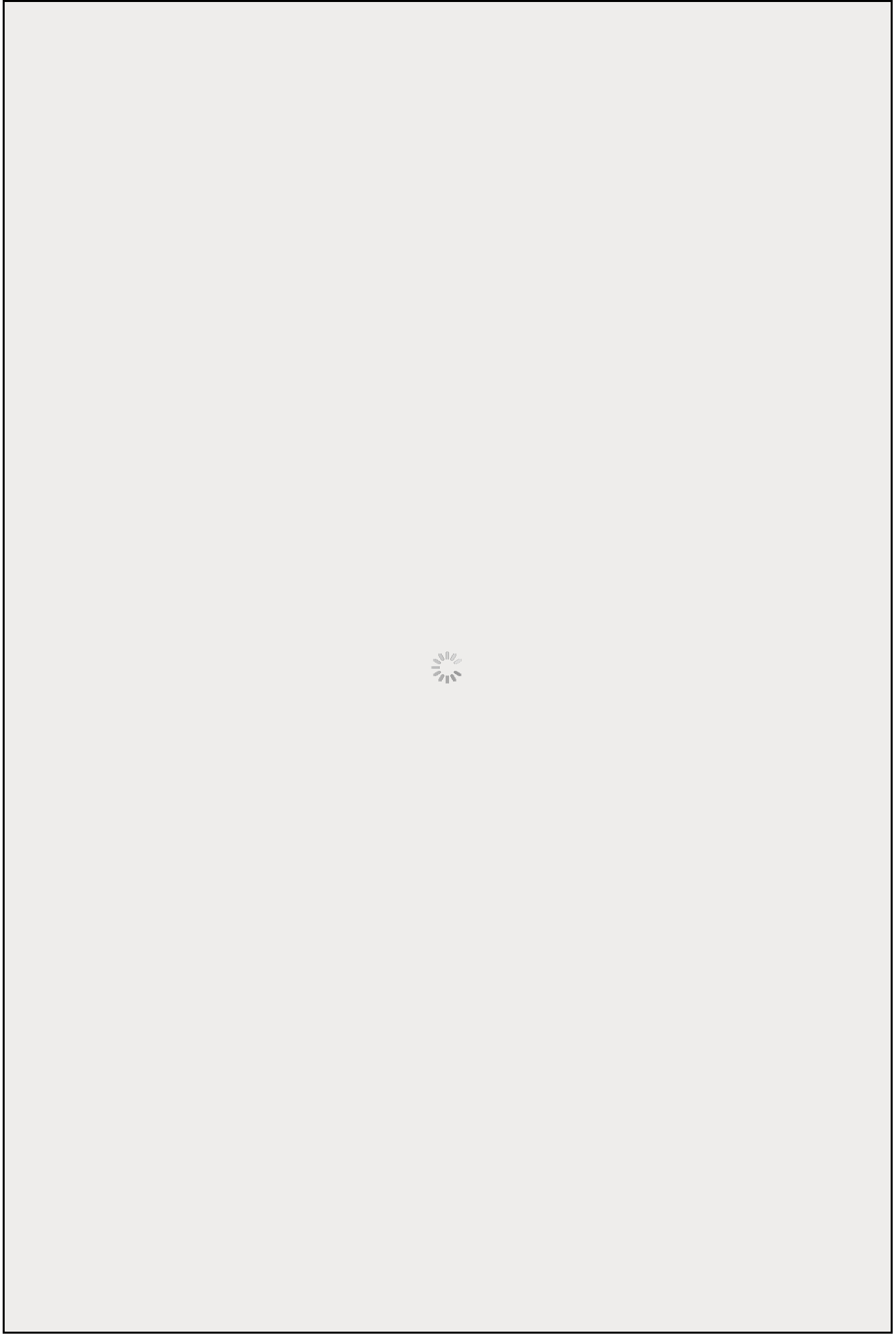
很多App都会注册大量的后台服务，这些后台服务会消耗额外的资源和电量

安卓系统的后台机制

我们先来谈谈安卓的后台机制，这可以让我们更清晰地得知为何流氓App为何容易驻留在后台。安

卓是一个基于Linux的操作系统，因此其后台机制也和Linux类似——一般情况下，返回桌面时程序并不会推出后台，而是在后台持续运行，当系统需要更多资源的时候，相应的程序才会被请出去。

安卓并不是随便清理后台的进程的，在安卓系统中，App分为Foreground_App（前台应用）、Visiable_App（可见应用）、Secondary_App（二级应用）、Hidden_App（隐藏应用）、Content_Provider（内容提供器）、Empty_App（空应用）等状态。当内存不足的时候，系统会优先终止Empty_App进程和服务，将内存释放出去；内存再次吃紧，就开始对Content_Provider动手脚了，以此类推。因此，越重要的进程会越得到保留，越无关重要的进程会越被优先清理出内存，这方案乍看之下没啥问题。

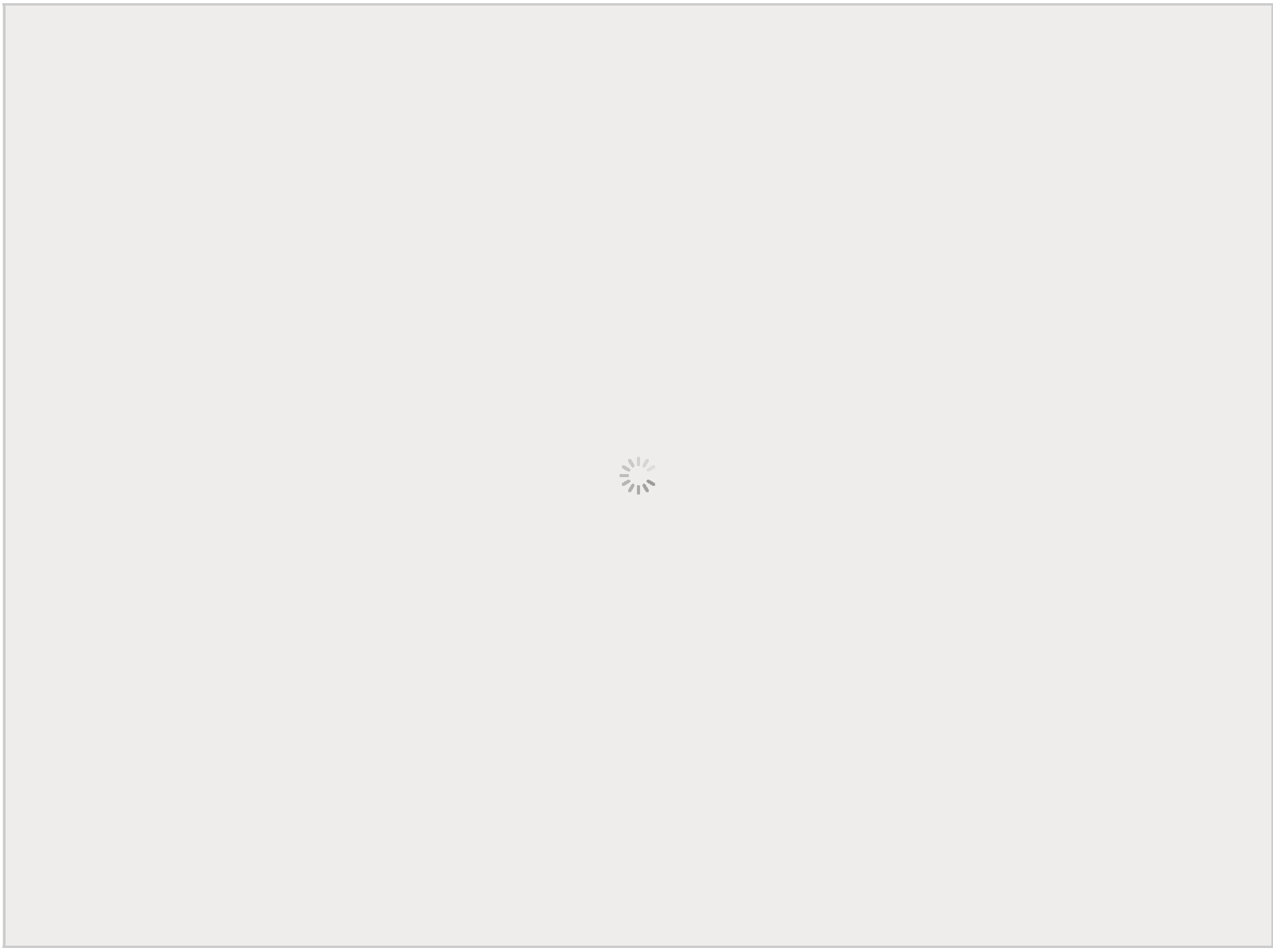


安卓把App状态分成不同类型，自动管理后台是否驻留

然而事情并不尽如人意，流氓App之所以流氓，就是不守规则。安卓系统中留有太多余地让App在后台自由发挥了，就算是Android 6.0引进了Doze打盹机制、Android O倡导后台纯净（Background-free），流氓App们仍能不声不响地在后台龙盘虎踞。如果没有使用特别的杀后台工具或者对后台有特殊限制的ROM，流氓App们甚至能让手机化身续航两小时的暖手宝。

流氓App驻留后台技巧：乱注册状态

前面提到，安卓把App分为好几个状态，不过流氓App可不会遵守这些规矩，让后台按照这些状态来运行。例如，流氓App就可以通过startForeground来把自己注册成为前台应用，让自己的后台成为最高优先级，永远不会被系统干掉。



安卓7.0对很多后台运行的App都在通知栏有公示，后来这些App不得不改变后台驻留的方法

不过，这个方法已经有了Google官方的应对，在安卓4.3以上的系统中，如果有App乱注册这个状态，通知栏就会显示“XX正在后台运行”。尽管流氓App通过一些手段一度绕了过去，但在安卓7.0中Google封堵了该漏洞。如果你升级到了安卓7.0，某App在通知栏持续显示“XX正在后台运行”，不用怀疑，这就是个流氓App。

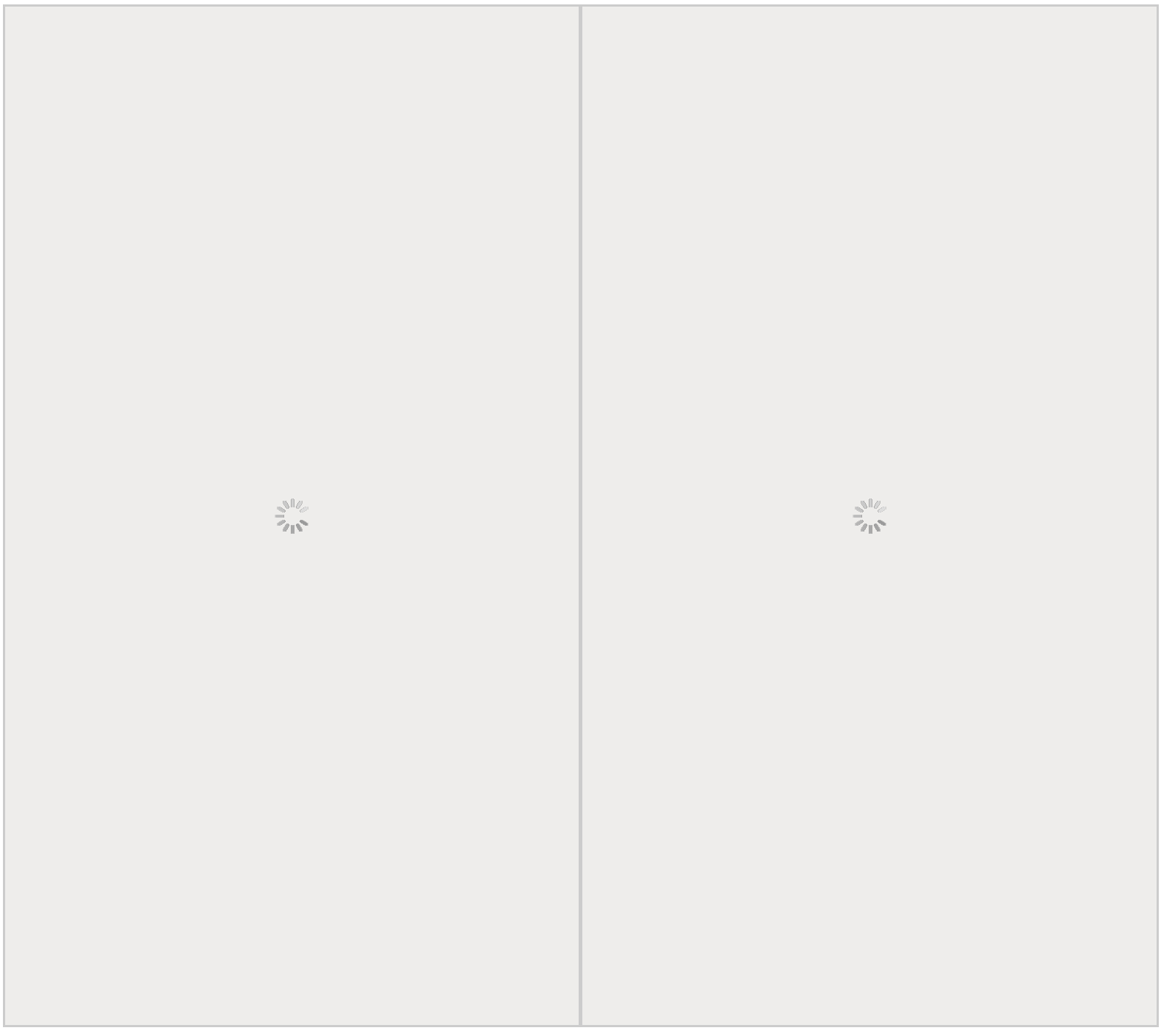
流氓App驻留后台技巧：透明悬浮窗

这是个脑洞大开的一招，我们知道和iOS不同，App可以在系统中显示悬浮窗，为用户提供各种实时信息。而开了悬浮窗的App，会一直运行，进程不会被随便清理掉。于是流氓App就在这方面动歪脑筋了，某流氓App会在设置一个1像素大小的透明悬浮窗，用户是看不到的，但这悬浮窗的确存在。App退到后台后，由于有悬浮窗的存在，进程也得以保留。

这个方法也已经被Google所察觉，在新版的安卓系统以及很多第三方ROM中，App已经不能随便申请到悬浮窗的权限。

流氓App驻留后台技巧：乱请求唤醒

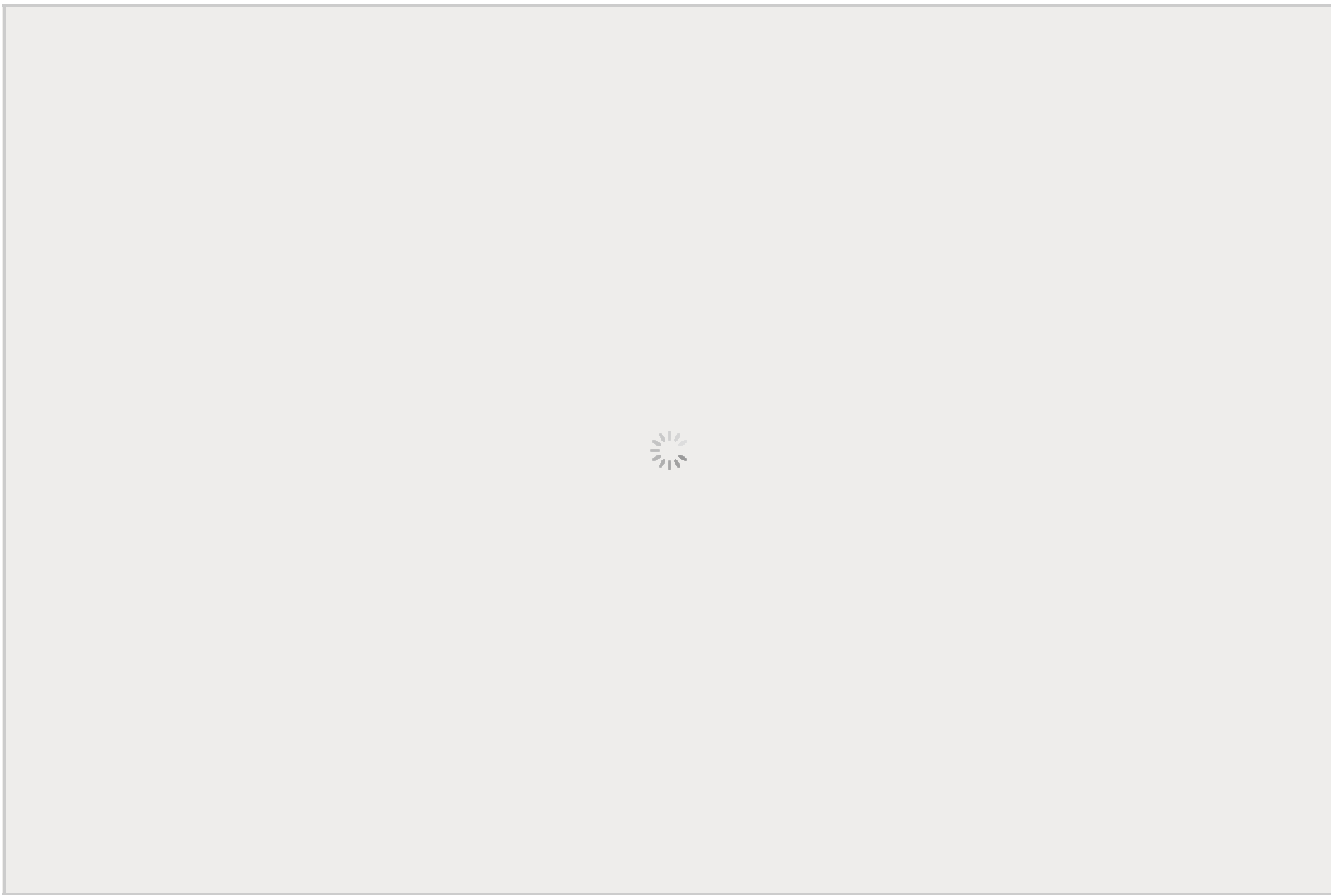
用过安卓的朋友都知道，流氓App最烦人的一点就在于胡乱自行启动，这和流氓App们胡乱请求唤醒是密不可分的。安卓系统拥有唤醒机制，App可以同某些具体事件，触发特定动作。例如到了时间，App可以触发铃声；又例如连上了网络或者间隔一段时间，App可以触发数据同步动作。这些动作都需要唤醒App才能运行，于是流氓App就把频频使用“Alarm”、“Sync Adapter”等周期性任务唤醒自己，让自己不断在后台启动，这也是很多杀后台App无法彻底干掉流氓App后台进程的原因。



流氓App通常利用广播接收器触发后台自启动

对此，Google官方也尝试使用对齐唤醒机制来解决。在安卓6.0中Google引入Doze机制，让后台进程尽量在统一的周期中同时唤醒，使CPU得到尽可能长的休眠时间。不过，这个机制并不够激进，需要手机无操作静止较长时间才会工作，总体而言效果有限。如果用户使用手机频度较大，Doze机制甚至不

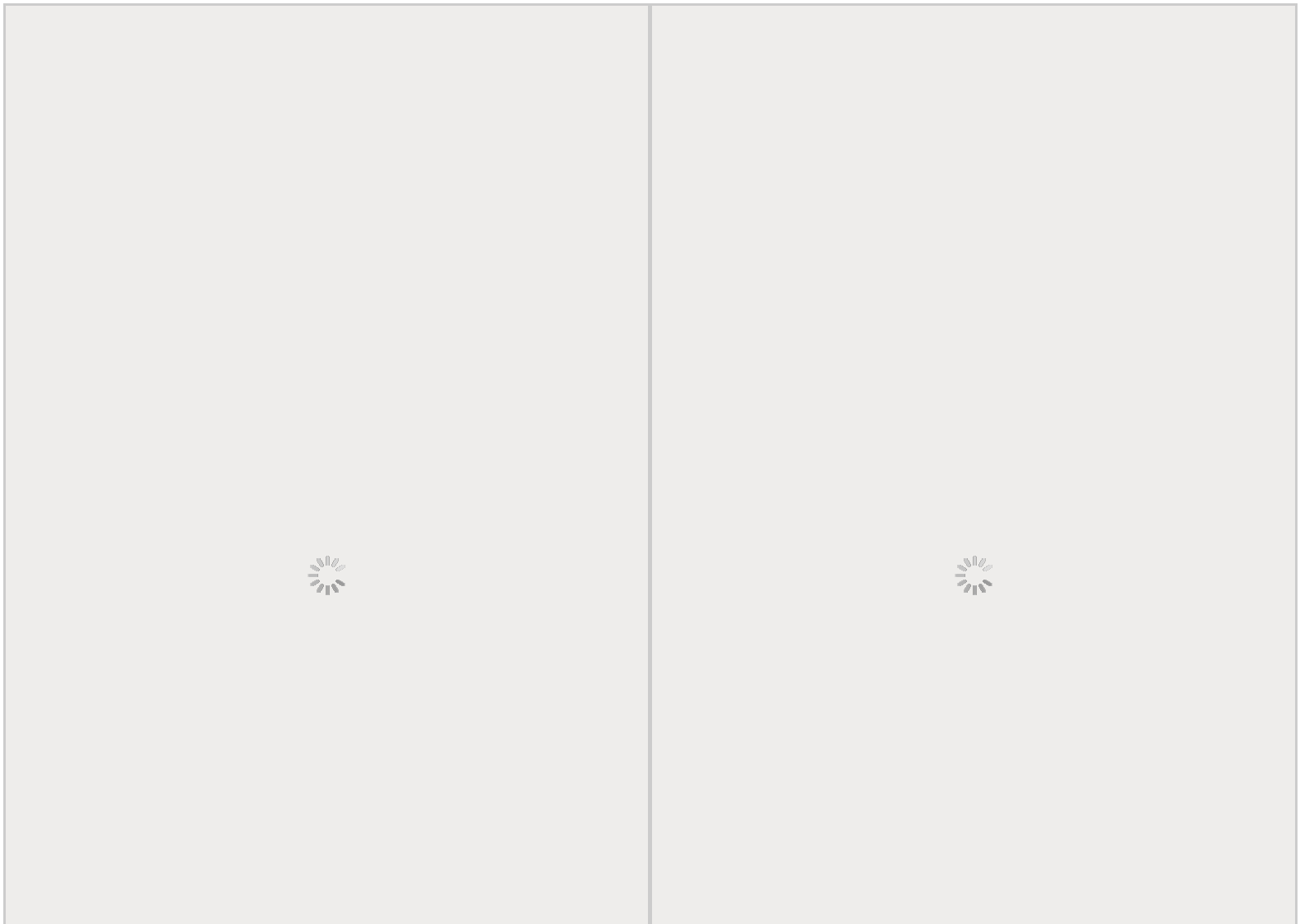
会生效，流氓App们依然会不断吞噬着手机电量。



安卓6.0引入的Doze模式，旨在让App统一唤醒，但并不能彻底解决问题

流氓App驻留后台技巧：进程相互唤醒

除了利用安卓系统的唤醒机制，流氓App们还可以利用进程来相互唤醒。一方面，流氓App可以在后台注册多个进程，就算某个进程被干掉，也可以用其他进程来唤醒——如果你仔细观察，可以发现很多App都在后台注册了不止一个进程，为的就是不断唤醒；另一方面，流氓App们还可以抱团相互唤醒！当你开启了A应用后，它的进程可能会唤醒B应用，B应用的进程接而又会唤醒C应用，这种“链式唤醒”在国内的App中尤为常见，令人防不胜防。



用绿色守护能观察到很多App都是由其他App唤醒的

“链式唤醒”如此常见，是有原因的。由于Google服务的缺失，很多国内App为了实现推送和广告等功能，不得不使用一些第三方的SDK。这些第三方的SDK往往就会让App抱团唤醒，不少App其实不想耍流氓，但用了这些第三方SDK，也不得不变流氓。当然，解决方法还是有的，有心的开发者可以在使用流氓SDK的时候，接入Project Condom这个开源库，避免“链式唤醒”。

Google也知道进程唤醒的问题所在，于是安卓8.0的开发规范要求App一旦进入后台，需要在短时间内停止所有的后台服务，也不可以随便启动新的后台服务。至于这个效果如何，目前安卓8.0尚未普及，还有待观察。

流氓App驻留后台技巧：沆瀣一气

这应该就是终极的大招了！如果流氓App本身就和ROM有勾结，这App无疑就获得了最高通行权，无论如何也不会被干掉。例如，你何时看到过原生安卓会干掉Google Play（别说Play服务不流氓）？国

内的一堆ROM也不会干掉自家的流氓推送服务，还得靠推送来卖广告呢。

面对这种情况，用户基本上是无能为力的。或许基于AOSP的第三方ROM可以解决问题，但并不是所有设备都有条件刷机。

总结

实际上，安卓的生态已经近似于恶性循环，流氓App们不断找方法驻留后台，而各种ROM为了对付流氓App，对后台限制又日益收紧，这令安卓渐渐缺失了最初的卖点。为此，业界也在想办法解决流氓App的问题，例如就有开发者提出了Android绿色应用公约，国内也打算联合开发者们建立统一的App推送机制，减少App后台驻留的必要性，希望安卓的生态最终还是可以越来越好吧。

- 来自：网络
- 脚本之家整理发布，如涉及作品内容、版权和其它问题，请与我们联系，我们将在第一时间删除内容！