**比较流行是通过自带的webkit浏览器与chrome进行攻击：**

网页攻击结合浏览器漏洞来获取shell

通过一个嵌入了js的php文件，用户访问的时候，会下载编写的js payload，之后这段代码会不提示用户直接执行，获取sd卡上的文件（本来以为因为沙箱的存在也只能是浏览器存储的文件，但是没想到是sd卡上的任意文件）直接上传到服务器。

**不申请权限但是可以执行的行为：**

REBOOT:

    用于重启设别的权限只授予了系统应用程序或者是与之相同签名的app。但是重启完全可以通过DOS，书上介绍是对Toast的使用：

while(true) {

    Toast test =new Toast(getApplicationContext());

    test.setView(new View(getApplicationContext()))

    test.show()    通过这种方式是Toast不可见

}

INTERNET:

    电源管理器API的isScreenOn 检测屏幕的状态。通过默认浏览器，利用它的权限。不过需要申请BROWSABLE 使得浏览器能作为组件用于接受数据。不过对网络的访问都是安装时候的默认权限，算不上什么。

**capability leak：**权限暴露给其他的应用程序，类似于通过浏览器绕过对网络访问权限的申请。也有可能是因为在Androidmainfest.xml中定义了shareaUserId属性，但是签名也要对，它将允许所有使用相同数字证书的程序运行在同一个进程中（也可以配置成不同的进程中），共享用户标识，不仅仅共享了文件资源，从而得到权限。