# Android App 自动化安全审计 需求说明书

## 安全术语

1. ***风险（Risk）***

所谓“风险”是指一个成功的特定攻击中被列为攻击目标，并且通常在特定威胁中暴露的可能性，风险评估是为了确定并尽快解决最重要的潜在的安全漏洞。其中列举了最关键和最有可能的危险，并评估漏洞在费用和可能性相互作用函数中的风险相互关联的程度。

分析风险有助于确定一个在时间和金钱上适当的安全预算，并确定实行安全政策的优先顺序，这样才能最迅速地解决最直接的挑战。

1. ***威胁（Threat）***

所谓“威胁”是指特定类型攻击的来源和手段。

威胁评估是为了确定最佳的办法，确保系统对某一特定威胁，或各类威胁的安全。渗透测试演习基本上是侧重于评估威胁概要，以帮助制定有效的对策来对付特定威胁所代表的各类攻击。如果风险评估更注重分析的潜力和趋势，一个人的资源，以牺牲品各种攻击，威胁评估，更多地侧重于分析攻击者的资源。

　　分析威胁有助于制定具体的安全策略，以利用策略优先权实施在线安全策略，并了解要确保安全的资源的具体执行需求。

1. ***漏洞利用（Vulnerability）***

所谓“漏洞利用”是指的是可以攻击成功的系统安全缺陷。漏洞测试，应在现有基础上由各方负责解决这种漏洞，并有助于向需要加以解决的安全提供用于识别意外危险的数据。这种漏洞不是特别的技术-它们也可以适用于社会因素，如个人身份验证和授权的政策。

漏洞测试有助于保持持续的安全，允许资源的安全负责人在新的危险产生时作出有效响应。这在政策和技术开发上也是宝贵的，并作为技术选择过程的一个组成部分；提早选择合适的技术可以确保节省大量时间、金钱，进一步降低其他经营成本。

理解这些术语的正确用法是重要的，不仅是听起来像你知道你在说些什么，甚至也不只是为了方便交流。此外，它还有助于开发和利用良好的策略。技术行话的特殊性反映了专家们确定专业领域区分的方式，并且甚至能帮助自己澄清应如何应对这些产生的挑战。

## 附录1 安全审计风险列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 风险项目 | 风险等级 | 详细描述 | 审计办法 |
| **安全漏洞与风险评估** | | | | |
| 1 | WebView远程代码执行（CVE-2012-6636） | 10 | JavascriptInterface允许攻击者向页面植入Javascript，通过反射在客户端中执行任意恶意代码。所有应用在4.2以下的应用会受影响，编译API level 小于17的应用在全部系统中都受影响 | 前提：  Android API level 小于17 (即Android 4.2之前的系统版本)才有此漏洞可能。  检测：  反编译apk文件为java源文件，扫描源文件是否调用addJavascriptInterface函数 |
| 2 | Android WebView 远程代码执行漏洞（CVE-2014-1939） |  | 在4.0至4.2的Android系统上，Webview会增加searchBoxJavaBredge\_，导致远程代码执行。攻击者可以向页面植入Javascript，通过反射在客户端中执行任意恶意代码 | 在Webview中调用removeJavascriptInterface("searchBoxJavaBredge\_") |
| 3 | 源代码反编译风险 | 8 | 测试客户端安装程序，判断是否能反编译为源代码，是否存在代码保护措施。未作保护  的 java 代码，可以轻易分析其运行逻辑，并针对代码中的缺陷对客户端或服务器端进行攻击。 | 若应用编译时API level小于17，需要提升SDK版本。如果希望4.2以下的手机不受影响，可以参考替代方案https://github.com/pedant/safe-java-js-webview-bridge |
| 4 | 应用完整性风险 |  | 测试客户端程序是否对自身完整性进行校验。攻击者能够通过反编译的方法在客户端程序中植入自己的木马，客户端程序如果没有自校验机制的话，攻击者可能会利用客户端程序窃取手机用户的隐私信息。 |  |
| 5 | 二次打包风险 |  |  |  |
|  | Manifest不安全属性配置风险 |  |  |  |
|  | app备份风险 |  | 允许程序备份，可能导致用户信息泄露 | 如果不需要备份则添加allowBackup=false，或者实现加密备份  allowBackup和debuggable属性设置存在敏感信息泄露，建议android:debuggable和android:allowBackup设置为false |
| **Drozer自动化安全评估** | | | | |
|  | Activity组件隐私泄露风险 |  | Activity接口可被其它应用调用，存在被用于执行特定的敏感操作或钓鱼欺骗的风险。 | Drozer审计 |
|  | Service组件暴露风险 |  | Service接口可被其它应用调用，这可能导致敏感信息泄露 | Drozer审计 |
|  | BroadcastReceiver组件暴露风险 |  | BroadcastReceiver接口可被外部调用，这可能导致敏感信息泄露 | Drozer审计 |
|  | ContentProvider组件暴露风险 |  | ContentProvider接口可被其它应用调用， 这可能导致敏感信息泄露 | Drozer审计 |
|  | Content Provider组件任意文件读取漏洞 |  | Android Content Provider存在文件目录遍历安全漏洞，该漏洞源于对外暴露Content Provider组件的应用，没有对Content Provider组件的访问进行权限控制和对访问的目标文件的Content Query Uri进行有效判断，攻击者利用该应用暴露的Content Provider的openFile()接口进行文件目录遍历以达到访问任意可读文件的目的。 | Drozer审计 |
|  | Android应用本地拒绝服务漏洞检测 |  | Android系统提供了Activity、Service和Broadcast Receiver等组件，并提供了Intent机制来协助应用间的交互与通讯，Intent负责对应用中一次操作的动作、动作涉及数据、附加数据进行描述，Android系统则根据此Intent的描述，负责找到对应的组件，将Intent传递给调用的组件，并完成组件的调用[1]。Android应用本地拒绝服务漏洞源于程序没有对Intent.getXXXExtra()获取的异常或者畸形数据处理时没有进行异常捕获，从而导致攻击者可通过向受害者应用发送此类空数据、异常或者畸形数据来达到使该应用crash的目的，简单的说就是攻击者通过intent发送空数据、异常或畸形数据给受害者应用，导致其崩溃。 | Drozer审计 |
| **敏感信息与权限** | | | | |
|  | 安装包签名检测 |  | 检测客户端是否经过恰当签名（正常情况下应用都应该是签名的，否则无法安装），签名的CN及其他字段是否正确标识客户端程序的来源和发布者身份。 | 如果发现应用程序的签名被修改，说明该APK应用经过篡改，建议重新打包生成一个新的APK应用，再进行签名。 |
|  | 用户自定义权限滥用风险 |  | 以下权限为normal权限,可能导致敏感信息泄露 | 建议修改为signature或者signatureOrSystem |
|  | 私有配置文件读写风险 |  | 建议禁用全局写操作 改为MODE\_PRIVATE |  |
|  | 私有配置文件写风险 |  | 建议禁用全局读写操作 改为MODE\_PRIVATE |  |
|  | 全局可读文件 |  | 应用内存在其它任何应用都可以读取的私有文件，可能造成信息泄漏 | 将文件属性改为只有所属用户或同组用户可以读取 |
|  | 全局可写文件 |  | 应用内存在其它任何应用都可以修改的私有文件，可能造成应用行为被修改甚至是代码注入 | 将文件属性改为只有所属用户或同组用户可以修改 |
|  | Uri用户敏感信息泄露 |  | Uri中包含用户敏感信息，导致逆向分析者很容易获得相关信息 | Uri路径做转换 |
|  | 尝试使用root权限 |  | 如果程序具有root权限，且没有对调用做限制的话，可能被恶意利用 | 禁用不必要的高权限，并对关键权限加上校验限制 |
|  | KeyStore风险 |  | Android系统KeyStore密钥存储组件存在敏感信息泄漏漏洞 | 建议禁用android.security.KeyStore |
|  | 检测私有目录下文件权限 |  | 测试APK应用私有目录下的文件权限，确保APP本身应用访问控制的安全，阻止第三方APP应用的非法访问等。 | 如果检测不安全则建议修改相关文件的权限，Linux系统下可以使用chmod命令。 |
|  | 检测客户端是否保存明文敏感信息（SharedPreferences配置文件） |  | 检测客户端SharedPreferences配置文件是否保存明文敏感信息。 | 如果发现shared\_prefs文件夹下存在敏感目录文件的话，则建议对相关敏感的信息进行加密。 |
|  | 检测客户端程序的敏感信息是否进行了加密，加密算法是否安全 |  | 检查手机客户端程序的敏感信息是否进行了加密，加密算法是否安全。 | 建议对客户端的相关敏感的信息进行安全加密，尽量使用强壮的加密算法，而非单次的hash散列算法（如MD5加密）或Base64加密等。 |
|  | 检测客户端储存在手机中的SQLite数据库文件安全 |  | 检查客户端程序存储在手机中的 SQLite数据库文件。 | 建议对客户端SQLite数据库中的相关敏感的信息进行安全加密，尽量使用强壮的加密算法，而非单次的hash散列算法（如MD5加密）或Base64加密等。 |
|  | 检测客户端其他文件存储数据安全 |  | 检查客户端程序的其他文件存储数据。 | 建议客户端尽量不保存相关敏感的信息，如果确实需要则建议采用强壮的加密算法，而非单次的hash散列算法（如MD5加密）或Base64加密等。 |
|  | 检测客户端logcat日志是否包含敏感信息 |  | 检查客户端程序存储在手机中的日志，是否存在敏感信息。 | 如果在logcat日志中检测到敏感信息则建议logcat日志在发行版本中对敏感信息如（用户名、账号等）不进行打印输出，防止敏感信息通过日志泄露。 |
|  | 检查是否具有摄像头、屏幕截屏调用权限 |  | 测试客户端应用的权限，检测app是否具有截屏、摄像头权限 | 通过检查APK应用AndroidManifest.xml文件中uses-permission的权限，发现是否具有截屏（android.permission.ACCESS\_SURFACE\_FLINGER）、摄像头权限（android.permission.CAMERA） |
|  | 检查是否具有SMS相关权限 |  | 测试客户端应用的权限，是否具有调用短信发送的权限。 | 通过检查APK应用AndroidManifest.xml文件中是否具有调用与SMS有关的权限。 |
|  | 检查是否具有防休眠/熄屏权限 |  | 测试客户端应用的权限，是否具有自动进入休眠/熄屏状态的权限 | 通过检查APK应用AndroidManifest.xml文件中是否具有android.permission.WAKE\_LOCK和android.permission.WRITE\_SETTINGS有关的权限 |
|  | 检查是否具有其它敏感权限 |  | 通过检查APK应用中AndroidManifest.xml文件与android通用权限的列表，看APK应用中是否具有非法的其它权限 | 提取APK应用AndroidManifest.xml文件权限列表 |
|  |  |  |  |  |

## 附录2 已有APP审计平台

|  |  |  |
| --- | --- | --- |
| 名称 | URL地址 | 主要功能 |
| 爱加密APP漏洞分析平台 | http://safe.ijiami.cn/ | 内嵌sdk检测；数据存储检测；  源码检测：敏感信息检测 |
| 金刚 - 腾讯安全应急响应中心 | http://service.security.tencent.com/kingkong | APP常规漏洞、组件风险、敏感信息、Manifest安全配置等 |
| 爱内测 | http://www.ineice.com/safetySpec | 提供Android组件检测、权限管理、dex保护、数据安全(传输、存储、输出)检测，以及对危险调试信息等常见的漏洞风险进行检测，检测在开发过程中遗漏、忽略和可被利用的漏洞。 |
|  |  |  |

附图 1 爱加密APP漏洞分析平台报告截图



附图 2 爱加密APP漏洞分析平台报告截图

