


寸草心(老人端)-安全测试-Android

2017-06-14

北京云测信息技术有限公司

应用信息

	应用名称	寸草心(老人端)	平台	Android
	版本	1.0	得分	45

测试概况



15
通过

2
高危

3
中危

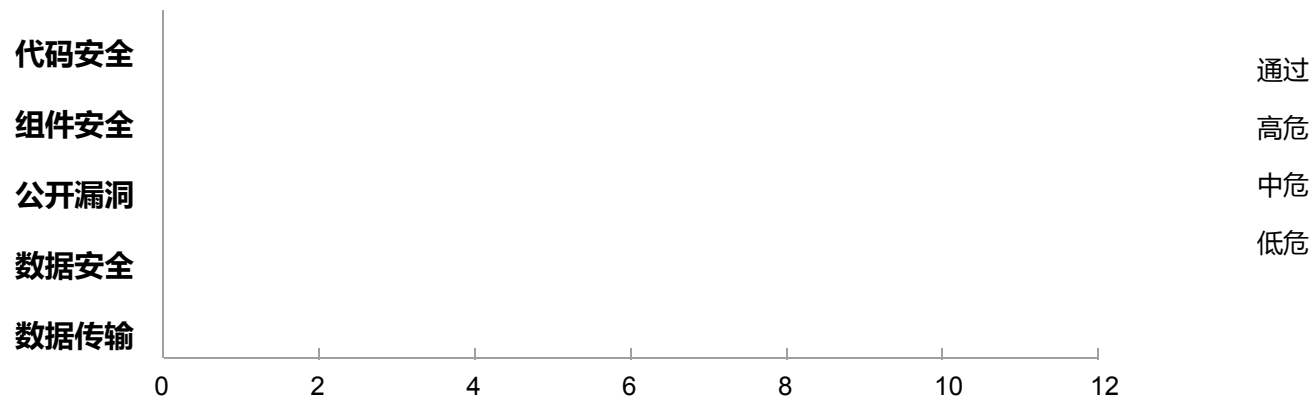
4
低危

测试基本信息

包名	com.example.jasper.ccxapp
MD5	19cdd8b9d66b77a3c9e1d60d9c940dc4
SHA-1	f9288da0d1f8671ba8e38aef01a518d3c330256f
SHA-256	eec138f60fbb87456e7d3c8b7072fc9efe2736c510f81a1565cd811a8d04369a
证书信息	Owner: C=US, O=Android, CN=Android Debug

测试模块概述

配置安全



检测模块	检测项	检测结果
配置安全	权限安全	低危
	防调试	中危
	备份配置	低危
	控制力检测	通过
代码安全	反编译	高危
	代码混淆	通过
	加载DEX	通过
	资源保护	通过
组件安全	Receiver	通过
	Activity	中危
	Service	通过

	Provider	通过
公开漏洞	密码明文	通过
	远程执行	通过
	file绕过	通过
	遗留接口	通过
	不校验https	通过
数据安全	弱加密	低危
	存储安全	通过
	file配置	中危
	Db配置模式	通过
	日志泄露	低危
	异常处理	通过
数据传输	中间人劫持	高危

检测详情

5.1 配置安全 [返回](#)

5.1.1 权限安全

检测说明

申请高危权限可能会造成其他恶意操作。

检测结果

低危

1:android.permission.ACCESS_COARSE_LOCATION ['dangerous', 'coarse (network-based) location', 'Access coarse location sources, such as the mobile network database, to determine an approximate phone location, where available. Malicious applications can use this to determine approximately where you are.'] 2:android.permission.INTERNET ['dangerous', 'full Internet access', 'Allows an application to create network sockets.'] 3:android.permission.ACCESS_FINE_LOCATION ['dangerous', 'fine (GPS) location', 'Access fine location sources, such as the Global Positioning System on the phone, where available. Malicious applications can use this to determine where you are and may consume additional battery power.'] 4:android.permission.GET_TASKS ['dangerous', 'retrieve running applications', 'Allows application to retrieve information about currently and recently running tasks. May allow malicious applications to discover private information about other applications.']

检测详情

5:android.permission.WRITE_EXTERNAL_STORAGE ['dangerous', 'modify/delete SD card contents', 'Allows an application to write to the SD card.'] 6:android.permission.READ_PHONE_STATE ['dangerous', 'read phone state and identity', 'Allows the application to access the phone features of the device. An application with this permission can determine the phone number and serial number of this phone, whether a call is active, the number that call is connected to and so on.'] 7:android.permission.SYSTEM_ALERT_WINDOW ['dangerous', 'display system-level alerts', 'Allows an application to show system-alert windows. Malicious applications can take over the entire screen of the phone.'] 8:android.permission.CAMERA ['dangerous', 'take pictures and videos', 'Allows application to take pictures and videos with the camera. This allows the application to collect images that the camera is seeing at any time.'] 9:android.permission.CHANGE_WIFI_STATE ['dangerous', 'change Wi-Fi status', 'Allows an application to connect to and disconnect from Wi-Fi access points and to make changes to configured Wi-Fi networks.'] 10:android.permission.RECORD_AUDIO ['dangerous', 'record audio', 'Allows application to access the audio record path.']

建议修复

避免开启与无关的权限或者把权限放的太大。

5.1.2 防调试

检测说明	程序开启调试模式会让他们获取应用调试信息并可对应用进行调试，获取应用信息。
检测结果	中危
检测详情	A: android:debuggable(0x0101000f)=(type 0x12)0xffffffff
建议修复	配置文件中 android:debuggable="false"。

5.1.3 备份配置

检测说明	Android属性allowBackup安全风险源于adb backup容许任何一个能够打开USB 调试开关的人从Android手机中复制应用数据到外设，一旦应用数据被备份之后，所有应用数据都可被用户读取；adb restore容许用户指定一个恢复的数据来源（即备份的应用数据）来恢复应用程序数据的创建。因此，当一个应用数据被备份之后，用户即可在其他Android手机或模拟器上安装同一个应用，以及通过恢复该备份的应用数据到该设备上，在该设备上打开该应用即可恢复到被备份的应用程序的状态。
检测结果	低危
检测详情	A: android:allowBackup(0x01010280)=(type 0x12)0xffffffff
建议修复	建议关闭应用备份功能; 在AndroidManifest.xml文件中，将相应组件的“android:allowBackup”属性设置为“false”。

5.1.4 控制力检测

检测说明	申请过高过多组合权限可能会使应用存在安全风险。
检测结果	通过

5.2 代码安全 [返回](#)

5.2.1 反编译

检测说明	Dex为Android应用的核心，保护不当容易被反编译，暴露程序重要信息，面临被植入广告、恶意代码、病毒等风险。
检测结果	高危
检测详情	应用未经过加壳加密保护，可反编译出源程序。
建议修复	1. 使用第三方的加固厂商对客户端进行加密处理，防止静态反编译之后看到程序的主要源代码；2. 插入花指令，防止线性工具对程序的反编译。

5.2.2 代码混淆

检测说明	对代码进行混淆，能够提高黑客阅读和理解代码的门槛，在一定程度上增加了黑客破解的难度。
检测结果	通过

5.2.3 加载DEX

检测说明	Android 系统提供了一种类加载器DexClassLoader，其可以在运行时动态加载并解释执行包含在JAR或APK文件内的DEX文件。外部动态加载DEX文件的安全风险源于：Anroid4.1之前的系统版本容许Android应用将动态加载的DEX文件存储在被其他应用任意读写的目录中(如sdcard)，因此不能够保护应用免遭恶意代码的注入；所加载的DEX易被恶意应用所替换或者代码注入，如果没有对外部所加载的DEX文件做完整性校验，应用将会被恶意代码注入，从而执行的是恶意代码。
检测结果	通过

5.2.4 资源保护

检测说明	解压或和反编译apk相关代码和资源文件，易造成他人窃取或者利用
检测结果	通过

5.3 组件安全 [返回](#)

5.3.1 Receiver

检测说明	可造成信息泄露，拒绝服务攻击等。
检测结果	通过

5.3.2 Activity

检测说明	导出的组件可以被第三方app任意调用，导致敏感信息泄露或者恶意攻击者精心构造攻击载荷达到攻击的目的。
检测结果	中危
检测详情	activity:com.example.jasper.ccxapp.ui.SplashActivity;
建议修复	1. 建议如果组件不需要与其它app共享数据或交互，请在AndroidManifest.xml 配置文件中将该组件设置为exported = "False"。如果组件需要与其它app共享数据或交互，请对组件进行权限控制和参数校验； 2. 在界面切换时，检测下一界面的Activity类，不是程序内的界面，便提示退出。

5.3.3 Service

检测说明	Service存在的安全漏洞包括：权限提升，拒绝服务攻击。没有声明任何权限的应用即可在没有任何提示的情况下启动该服务，完成该服务所作操作，对系统安全性产生极大影响。
检测结果	通过

5.3.4 Provider

检测说明	Content Provider的不安全使用会产生sql 注入、文件遍历等漏洞，导致用户数据泄露。
检测结果	通过

5.4 公开漏洞 [返回](#)

5.4.1 密码明文

检测说明	使用Webview时需要关闭webview的自动保存密码功能,防止用户密码被webview明文存储。
检测结果	通过

5.4.2 远程执行

检测说明	利用android的webView组件中的addJavascriptInterface接口函数，可以实现本地java与js之间交互，攻击者可以利用这一点进行远程任意代码执行。另外，android webview组件包含3个隐藏的系统接口：“accessibility”、“ccessibilityaversal”以及“searchBoxJavaBridge_”，同样会造成远程代码执行。
检测结果	通过

5.4.3 file绕过

检测说明	JavaScript的延时执行能够绕过file协议的同源检查，并能够访问受害应用的所有私有文件，即通过WebView对Javascript的延时执行和将当前Html文件删除掉并软连接指向其他文件就可以读取到被符号链接所指的文件，然后通过JavaScript再次读取HTML文件，即可获取到被符号链接所指的文件。大多数使用WebView的应用都会受到该漏洞的影响，恶意应用通过该漏洞，可在无特殊权限下盗取应用的任意私有文件，尤其是浏览器，可通过利用该漏洞，获取到浏览器所保存的密码、Cookie、收藏夹以及历史记录等敏感信息，从而造成敏感信息泄露。
------	---

检测结果 通过

5.4.4 遗留接口

检测说明 Android webview组件包含3个隐藏的系统接口：searchBoxJavaBridge_，accessibilityTraversal以及accessibility，恶意程序可以利用它们实现远程代码执行。

检测结果 通过

5.4.5 不校验https

检测说明 Android中可以用WebView来访问http和https的网站，但是默认访问https网站时，假如证书不被Android承认，会出现空白页面，且不会有任何提示信息，这时我们必须加多一些配置。

检测结果 通过

5.5 数据安全 [返回](#)

5.5.1 弱加密

检测说明 使用AES/DES加密算法时，应显式指定使用CBC或CFB模式.否则容易受到选择明文攻击(CPA)的风险，造成信息泄露。

检测结果 低危

检测详情
./class1/com/baidu/android/bbalbs/common/security/AESUtil.java:17: paramString2 = new
SecretKeySpec(paramString2.getBytes(), 'AES');
./class1/com/baidu/android/bbalbs/common/security/AESUtil.java:26: paramString2 = new
SecretKeySpec(paramString2.getBytes(), 'AES'); ./class1/com/baidu/tts/tools/AESUtil.java:15:
paramString2 = new SecretKeySpec(paramString2.getBytes(), 'AES');

```
./class1/com/baidu/tts/tools/AESUtil.java:25: paramString2 = new
SecretKeySpec(paramString2.getBytes(), 'AES');
```

建议修复

使用AES/DES加密算法时应使用CBC或CFB模式。或者使用安全组件的安全加密接口SecurityCipher进行加密。

5.5.2 存储安全

检测说明

Shared Preferences存储安全风险源于:1)开发者在创建文件时没有正确的选取合适的创建模式(MODE_PRIVATE、MODE_WORLD_READABLE以及MODE_WORLD_WRITEABLE)进行权限控制;2)开发者过度依赖Android系统内部存储安全机制,将用户信息、密码等敏感重要的信息明文存储在Shared Preferences文件中,导致攻击者可通过root手机来查看敏感信息。

检测结果

通过

5.5.3 file配置

检测说明

android有一套自己的安全模型,当应用程序(.apk)在安装时系统就会分配给他一个userid,当该应用要去访问其他资源比如文件的时候,就需要userid匹配。默认情况下,任何应用创建的文件,sharedpreferences,数据库都应该是私有的(位于/data/data//files),其他程序无法访问。

检测结果

中危

检测详情

```
class1/com/baidu/android/bbalbs/common/util/DeviceId.java:1174: FileOutputStream
localFileOutputStream = this.mContext.openFileOutput('libcuid.so', 1);
class1/com/baidu/tts/tools/DeviceId.java:860: FileOutputStream localFileOutputStream =
this.c.openFileOutput('libcuid.so', 1);
```

建议修复

1. 避免使用MODE_WORLD_WRITEABLE和MODE_WORLD_READABLE模式创建进程间通信的内部存储(Internal Storage)文件;出于安全考虑,建议不要使用全局可读模式和全局可写模式创建进程间通信的文件,此处即为但不限于内部存储(Internal Storage)文件。如果需要与其他进程应用进行数据

共享，请考虑使用content provider。 2. 避免滥用“android:sharedUserId”属性；建议不要在使用“android:sharedUserId”属性的同时，对应用使用测试签名，否则其他应用拥有“android:sharedUserId”属性值和测试签名时，将会访问到内部存储文件数据。 3. 避免将密码等敏感数据信息明文存储在内部存储(Internal Storage)文件中；出于安全考虑，建议不要将密码等敏感信息存储在内部存储(Internal Storage)文件中，即使Android系统内部存储安全机制，使得内部存储文件可不让其他应用读写，但是在Android系统root之后，该安全机制将失效而导致信息泄露。因此应该将敏感信息进行加密存储在内部存储(Internal Storage)文件中。

5.5.4 Db配置模式

检测说明	database配置模式安全风险源于:1)开发者在创建数据库(Database)时没有正确的选取合适的创建模式(MODE_PRIVATE、MODE_WORLD_READABLE以及MODE_WORLD_WRITEABLE)进行权限控制，从而导致数据库(Database)内容被恶意读写，造成账户密码、身份信息、以及其他敏感信息的泄露，甚至攻击者进一步实施恶意攻击。如果在开发中没有使用正确的创建模式数据库(Database)文件，将会导致敏感信息泄露危害，如个人账户密码、身份信息以及金融账户等重要敏感信息。
检测结果	通过

5.5.5 日志泄露

检测说明	使用System.out.print等标准输出打印日志信息或转存日志信息，容易泄漏敏感信息。建议删除所有使用System.out.print等标准输出打印日志或转存日志信息的代码
检测结果	低危
检测详情	./class1/cn/jiguang/api/Utils/ProtocolUtil.java:279: System.out.println(h.a(paramArrayOfString)); ./class1/cn/jiguang/c/a.java:12: Log.v(paramString1, paramString2); ./class1/cn/jiguang/c/a.java:17: Log.v(paramString1, paramString2, paramThrowable); ./class1/cn/jiguang/c/a.java:22: Log.d(paramString1, paramString2); ./class1/cn/jiguang/c/a.java:27: Log.d(paramString1, paramString2, paramThrowable); ./class1/cn/jiguang/c/a.java:32: Log.i(paramString1, paramString2); ./class1/cn/jiguang/c/a.java:37: Log.i(paramString1, paramString2, paramThrowable);

```
./class1/cn/jiguang/c/a.java:42: Log.w(paramString1, paramString2); ./class1/cn/jiguang/c/a.java:47:
Log.w(paramString1, paramString2, paramThrowable); ./class1/cn/bmob/v3/BmobInstallation.java:94:
Log.i('bmob', paramAnonymousThrowable.getMessage()); ./class1/cn/bmob/v3/b/From.java:122:
Log.w('bmob', 'No android.permission.ACCESS_NETWORK_STATE Privileges. ');
./class1/cn/bmob/v3/b/This.java:147: Log.d(paramString1, paramString2);
./class1/cn/bmob/v3/b/This.java:153: Log.i(paramString1, paramString2);
./class1/cn/bmob/v3/b/This.java:155: Log.v(paramString1, paramString2);
./class1/cn/bmob/v3/b/This.java:157: Log.w(paramString1, paramString2);
./class1/cn/bmob/v3/socketio/From.java:118: Log.d('WebSocketClient', 'Error while disconnecting',
localIOException); ./class1/cn/bmob/v3/socketio/From.java:153: Log.d('WebSocketClient', 'WebSocket
EOF!', localEOFException); ./class1/cn/bmob/v3/socketio/From.java:179: Log.d('WebSocketClient',
'Websocket SSL error!', localSSLException); ./class1/cn/bmob/v3/socketio/I.java:50:
Log.d('HybiParser', 'Creating frame for: ' + paramObject + ' op: ' + paramInt1 + ' err: ' + paramInt2);
./class1/cn/bmob/v3/socketio/I.java:369: Log.d('HybiParser', 'Got close op! ' + i + ' ' +
(String) localObject);.....
```

建议修复

建议删除所有使用System.out.print等标准输出打印日志或转存日志信息的代码

5.5.6 异常处理

检测说明

如果有些异常不提示详细信息又会给 用户报告异常带来麻烦，不利于开发人员及时发现并处理异常。会不经处理直接提示给用户则会带来安全隐患。

检测结果

通过

5.6 数据传输 [返回](#)

5.6.1 中间人劫持

检测说明

Android HTTPS中间人攻击漏洞源于：1没有对SSL证书进行校验；2没有对域名进行校验；3. 证书颁发机构(Certification Authority)被攻击导致私钥泄露等。攻击者可通过中间人攻击，盗取账户密码明

文、聊天内容、通讯地址、电话号码以及信用卡支付信息等敏感信息，甚至通过中间人劫持将原有信息替换成恶意链接或恶意代码程序，以达到远程控制、恶意扣费等攻击意图。

检测结果

高危

检测详情

./class1/com/loopj/android/jpush/http/MySSLSocketFactory.java:171:

localMySSLSocketFactory.setHostnameVerifier(org.apache.http.conn.ssl.SSLSocketFactory.ALLOW_ALL_HOSTNAME_VERII

./class1/com/baidu/tts/loopj/MySSLSocketFactory.java:59:

localMySSLSocketFactory.setHostnameVerifier(org.apache.http.conn.ssl.SSLSocketFactory.ALLOW_ALL_HOSTNAME_VERII

建议修复

建议对SSL证书进行强校验（签名CA是否合法、证书是否是自签名、主机域名是否匹配、证书是否过期等）。