安全评估报告

应用名称：智家宝

应用版本：0.1.1203

检测单位：360加固保

360加固保

目录

第1章 检测摘要

1.1 APP摘要

应用名	智家宝
包名	com.hzspec.zjb
MD5	0bbaec4ede9050c7d9c490ae21a2d704
版本	0.1.1203
加固信息	未加固
签名	签名方式：V1 所有者: CN=yy, OU=bk, O=bk, L=bj, ST=bj, C=cn 发布者: CN=yy, OU=bk, O=bk, L=bj, ST=bj, C=cn 序列号: 432a5e73 有效期为 Tue Oct 09 10:33:06 CST 2018 至 Mon May 03 10:33:06 CST 2117 证书指纹: MD5: AE:13:93:47:AB:A4:02:39:63:84:7F:49:40:D8:5A:0B SHA1: 81:28:C9:67:EC:6B:1A:83:EB:EA:9D:85:BF:5F:25:C3:53:90:CC:35 SHA256: 0D:83:7C:AA:9B:EA:C0:21:1E:7D:BC:9E:23:B3:4D:29:0C:5C:7C:88:94:7E:70:F6:49:40:E9:24:F7:B8:F4:67 签名算法名称: SHA256withRSA 主体公共密钥算法: 2048 位 RSA 密钥 版本: 3
检测单位	360加固保
报告日期	2018-12-06

1.2 测试评分

应用评分	高危漏洞数	中危漏洞数	低危漏洞数
85	1	1	0

1.3 详细信息

序号		测试内容	测试结果	漏洞等级
1	测试项	敏感权限检测	-	-
2		包含APK文件检测	-	-
3		证书安全性检测	-	-
4		APP加固检测	-	-
5	组件安全检测	Activity组件暴露检测	安全	中危
6		Service组件暴露检测	安全	中危
7		BroadcastReceiver组件暴露检测	安全	中危
8		Content Provider组件暴露检测	安全	中危
9		SSL通信服务端检测信任任意证书漏洞检测	安全	高危
10		HTTPS关闭主机名验证漏洞检测	安全	中危
11		隐式意图调用漏洞检测	中危	中危
12		程序数据任意备份漏洞检测	高危	高危
13		程序可被任意调试漏洞检测	安全	中危
14		Webview存在本地Java接口漏洞检测	安全	中危
15		WebView忽略SSL证书错误漏洞检测	安全	高危
16		Intent Scheme URLs攻击漏洞检测	安全	低危
17		全局文件可读漏洞检测	安全	中危
18		全局文件可写漏洞检测	安全	高危
19		配置文件可读漏洞检测	安全	中危

20		配置文件可写漏洞检测	安全	高危
21		DEX文件动态加载漏洞检测	安全	中危
22		AES弱加密漏洞检测	安全	中危
23		Provider文件目录遍历漏洞检测	安全	中危
24	**APP漏洞检测**	activity绑定browserable与自定义协议漏洞检测	安全	低危
25		动态注册广播漏洞检测	安全	中危
26		开放socket端口漏洞检测	安全	低危
27		Fragment注入漏洞检测	安全	低危
28		webview明文存储密码漏洞检测	安全	高危
29		unzip解压缩（ZipperDown）漏洞检测	安全	高危
30		未使用编译器堆栈保护技术漏洞检测	安全	低危
31		未使用地址空间随机化技术漏洞检测	安全	低危
32		动态链接库中包含执行命令函数漏洞检测	安全	低危
33		随机数不安全使用漏洞检测	安全	低危
34		FFmpeg文件读取漏洞检测	安全	低危
35		libupnp栈溢出漏洞漏洞检测	安全	低危
36		Webview组件远程代码执行（调用getClassLoader）漏洞检测	安全	中危
37		AES/DES硬编码密钥漏洞检测	安全	高危
38		Android应用克隆漏洞检测	安全	高危

第2章 合规风险检测

2.1 敏感权限检测

检测项目	敏感权限	威胁等级	-
风险描述	本项目仅对应用中存在的高级敏感权限进行列举，方便应用厂商自查。权限是android的一种安全机制，主要用于限制应用程序内部某些具有限制性特性的功能使用以及应用程序之间的组件访问。申请过多的高级敏感权限容易被恶意程序利用进行诸如读取、修改通讯录，拨打电话，发送地理位置等高风险操作。从安全性考虑应用应采取最小权限的原则，避免申请冗余权限。		
检测结果	1. android.permission.INTERNET 权限描述：访问网络 2. android.permission.ACCESS_NETWORK_STATE 权限描述：获取网络状态		
修复意见	对于发现的敏感权限，应确定是否必须，对于冗余的敏感权限，应删除。		

2.2 包含APK文件检测

检测项目	包含APK文件	威胁等级	-
风险描述	应用的资源文件中包含的APK文件，由于内部APK文件的未知和不可控性，可能会携带恶意程序或造成其他不可预知的风险。		
检测结果	安全		
修复意见	应确定资源文件包含APK的必要性，如非必要建议用其他方式实现。如必要，应确保APK的安全性，对内部包含的APK文件进行安全核查和检测。		

2.3 证书安全检测

检测项目	证书安全性检测	威胁等级	-
风险描述	本项目主要检测APP是否使用了正式发布证书。部分应用在发布时使用调试证书发布APP，可能导致APP无法上架应用市场。另一方面，调试证书的有效期仅有一年时间，使用调试证书发布APP可能		

	导致各版本签名证书不一致，从而无法对证书签名进行保护，造成二次打包的风险。
检测结果	正式证书
修复意见	应用正式发布时应当采用正式证书而非调试证书进行发布。

2.4 APP加固检测

检测项目	APP加固检测	威胁等级	-
风险描述	检测应用是否经过加固。没有经过加固的应用，在安全对抗性上，尤其是针对外部的逆向、调试、注入和二次打包等攻击手段的对抗上存在较大风险。		
检测结果	未加固		
修复意见	建议采用专业的应用安全厂商对应用进行加固。		

第3章 组件安全检测

3.1 Activity组件暴露检测

漏洞名称	Activity组件暴露检测	威胁等级	中危
漏洞危害	Activity组件的属性exported被设置为true或是未设置exported值但IntentFilter不为空时，activity被认为是导出的，可通过设置相应的Intent唤起activity。黑客可能构造恶意数据针对导出activity组件实施越权攻击。		
检测结果	安全 检测明细： 1. com.hzspec.zjb.MainActivity 导出,无风险		
修复意见	如果组件不需要与其他app共享数据或交互，请将AndroidManifest.xml 配置文件中设置该组件为exported = “False”。如果组件需要与其他app共享数据或交互， 请对组件进行权限控制和参数校验。		
参考链接	https://developer.android.com/guide/components/activities.html		

3.2 Service组件暴露检测

漏洞名称	Service组件暴露检测	威胁等级	中危
漏洞危害	Service组件的属性exported被设置为true或是未设置exported值但IntentFilter不为空时，Service被认为是导出的，可通过设置相应的Intent唤起Service。黑客可能构造恶意数据针对导出Service组件实施越权攻击。		
检测结果	安全		
修复意见	如果组件不需要与其他app共享数据或交互，请将AndroidManifest.xml 配置文件中设置该组件为exported = “False”。如果组件需要与其他app共享数据或交互， 请对组件进行权限控制和参数校验。		
参考链接	https://developer.android.com/guide/components/services.		

	html

3.3 BroadcastReceiver组件暴露检测

漏洞名称	BroadcastReceiver组件暴露检测	威胁等级	中危
漏洞危害	BroadcastReceiver组件的属性exported被设置为true或是未设置exported值但IntentFilter不为空时，BroadcastReceiver被认为是导出的。导出的广播可以导致数据泄漏或者是越权。		
检测结果	安全		
修复意见	如果组件不需要与其他app共享数据或交互，请将AndroidManifest.xml 配置文件中设置该组件为exported = “False”。如果组件需要与其他app共享数据或交互， 请对组件进行权限控制和参数校验。		
参考链接	https://developer.android.com/guide/topics/manifest/receiver-element.html		

3.4 Content Provider组件暴露检测

漏洞名称	Content Provider组件暴露检测	威胁等级	中危
漏洞危害	Content Provider组件的属性exported被设置为true或是Android API<=16时，Content Provider被认为是导出的。黑客可能访问到应用本身不想共享的数据或文件。		
检测结果	安全		
修复意见	如果组件不需要与其他app共享数据或交互，请将AndroidManifest.xml 配置文件中设置该组件为exported = “False”。如果组件需要与其他app共享数据或交互， 请对组件进行权限控制和参数校验。		
参考链接	https://developer.android.com/guide/topics/providers/content-providers.html		

第4章 APP漏洞检测

4.1 SSL通信服务端检测信任任意证书漏洞检测

漏洞名称	SSL通信服务端检测信任任意证书漏洞检测	威胁等级	高危
漏洞危害	自定义SSL x509 TrustManager，重写checkServerTrusted方法，方法内不做任何服务端的证书校验。黑客可以使用中间人攻击获取加密内容。		
检测结果	安全		
修复意见	严格判断服务端和客户端证书校验，对于异常事件禁止return 空或者null		
参考链接	https://developer.android.com/reference/javax/net/ssl/X509TrustManager.html		

4.2 HTTPS关闭主机名验证漏洞检测

漏洞名称	HTTPS关闭主机名验证漏洞检测	威胁等级	中危
漏洞危害	构造HttpClient时，设置HostnameVerifier时参数使用ALLOW_ALL_HOSTNAME_VERIFIER或空的HostnameVerifier。关闭主机名校验可以导致黑客使用中间人攻击获取加密内容。		
检测结果	安全		
修复意见	APP在使用SSL时没有对证书的主机名进行校验，信任任意主机名下的合法的证书，导致加密通信可被还原成明文通信，加密传输遭到破坏。		
参考链接	https://developer.android.com/reference/javax/net/ssl/HostnameVerifier.html		

4.3 隐式意图调用漏洞检测

漏洞名称	隐式意图调用漏洞检测	威胁等级	中危
	封装Intent时采用隐式设置，只设定action，未限定具体的接收对		

漏洞危害	象，导致Intent可被其他应用获取并读取其中数据。Intent隐式调用发送的意图可能被第三方劫持，可能导致内部隐私数据泄露。
检测结果	1. 类名：Lcom/phonegap/plugins/nativesettings/NativeSettings; 方法名：execute 出现次数：3
修复意见	将隐式调用改为显式调用。
参考链接	https://developer.android.com/guide/components/intents-filters.html

4.4 程序数据任意备份漏洞检测

漏洞名称	程序数据任意备份漏洞检测	威胁等级	高危
漏洞危害	安卓AndroidManifest.xml文件中android: allowBackup为true。app数据可以被备份导出。		
检测结果	1. 类名：None 方法名：None 出现次数：1		
修复意见	AndroidManifest.xml 配置文件中设置为android:allowBackup="false"		
参考链接	https://developer.android.com/guide/topics/manifest/application-element.html#allowbackup		

4.5 程序可被任意调试漏洞检测

漏洞名称	程序可被任意调试漏洞检测	威胁等级	中危
漏洞危害	安卓AndroidManifest.xml文件中android:debuggable为true。app可以被任意调试，攻击者可以发起动态调试攻击。		
检测结果	安全		
修复意见	AndroidManifest.xml 配置文件中设置为android:Debugable="false"。		
参考链接	https://developer.android.com/guide/topics/manifest/application-element.html#debug		

4.6 Webview存在本地Java接口漏洞检测

	Webview存在本地Java		

漏洞名称	接口漏洞检测	威胁等级	中危
漏洞危害	android的webView组件有一个非常特殊的接口函数addJavascriptInterface，能实现本地java与js之间交互。在targetSdkVersion小于17时，攻击者利用addJavascriptInterface这个接口添加的函数，可以远程执行任意代码。		
检测结果	安全		
修复意见	建议开发者不要使用addJavascriptInterface，使用注入javascript和第三方协议的替代方案。		
参考链接	https://developer.android.com/reference/android/webkit/WebView.html		

4.7 Webview忽略SSL证书错误漏洞检测

漏洞名称	Webview忽略SSL证书错误漏洞检测	威胁等级	高危
漏洞危害	WebView调用onReceivedSslError方法时，直接执行handler.proceed()来忽略该证书错误。忽略SSL证书错误可能引起中间人攻击。		
检测结果	安全		
修复意见	不要重写onReceivedSslError方法， 或者对于SSL证书错误问题按照业务场景判断，避免造成数据明文传输情况。		
参考链接	https://developer.android.com/reference/android/webkit/WebViewClient.html		

4.8 Intent Scheme URLs攻击漏洞检测

漏洞名称	Intent Scheme URLs攻击漏洞检测	威胁等级	低危
漏洞危害	在AndroidManifast.xml设置Scheme协议之后，可以通过浏览器打开对应的Activity。攻击者通过访问浏览器构造Intent语法唤起app相应组件，轻则引起拒绝服务，重则可能演变为提权漏洞。		
检测结果	安全		
修复意见	配置category filter, 添加android.intent.category.BROWSABLE方式规避风险		

参考链接	https://developer.android.com/guide/components/intents-filters.html

4.9 全局文件可读漏洞检测

漏洞名称	全局文件可读漏洞检测	威胁等级	中危
漏洞危害	APP在创建内部存储文件时，将文件设置了全局的可读权限。攻击者恶意读取文件内容，获取敏感信息。		
检测结果	安全		
修复意见	请开发者确认该文件是否存储敏感数据，如存在相关数据，请去掉文件全局可读属性。		
参考链接	https://developer.android.com/reference/android/content/Context.html#MODE_WORLD_READABLE		

4.10 全局文件可写漏洞检测

漏洞名称	全局文件可写漏洞检测	威胁等级	高危
漏洞危害	APP在创建内部存储文件时，将文件设置了全局的可写权限。攻击者恶意写文件内容，破坏APP的完整性。		
检测结果	安全		
修复意见	请开发者确认该文件是否存储敏感数据，如存在相关数据，请去掉文件全局可写属性。		
参考链接	https://developer.android.com/reference/android/content/Context.html#MODE_WORLD_WRITEABLE		

4.11 配置文件可读漏洞检测

漏洞名称	配置文件可读漏洞检测	威胁等级	中危
漏洞危害	使用getSharedPreferences打开文件时，如果将第二个参数设置为MODE_WORLD_READABLE。当前文件可以被其他应用读取导致信息泄漏。		
检测结果	安全		
修复意见	使用getSharedPreferences时第二个参数设置为MODE_PRIVATE。如果必须设置为全局可读模式供其他程序使用，请保证存储的数		

	据非隐私数据或是加密后存储。
参考链接	https://developer.android.com/reference/android/content/Context.html#getSharedPreferences(java.lang.String,int) https://developer.android.com/reference/android/content/Context.html#MODE_WORLD_READABLE

4.12 配置文件可写漏洞检测

漏洞名称	配置文件可写漏洞检测	威胁等级	高危
漏洞危害	使用getSharedPreferences打开文件时，如果将第二个参数设置为MODE_WORLD_WRITEABLE。当前文件可以被其他应用写入，导致文件内容被篡改，影响应用程序的正常运行或更严重的问题。		
检测结果	安全		
修复意见	使用getSharedPreferences时第二个参数必须设置为MODE_PRIVATE。		
参考链接	https://developer.android.com/reference/android/content/Context.html#getSharedPreferences(java.lang.String,int) https://developer.android.com/reference/android/content/Context.html#MODE_WORLD_WRITEABLE		

4.13 DEX文件动态加载漏洞检测

漏洞名称	DEX文件动态加载漏洞检测	威胁等级	中危
漏洞危害	使用DexClassLoader加载外部的 apk、jar 或 dex文件，当外部文件的来源无法控制时或是被篡改，此时无法保证加载的文件是否安全。加载恶意的dex文件将会导致任意命令的执行。		
检测结果	安全		
修复意见	加载外部文件前，必须使用校验签名或MD5等方式确认外部文件的安全性。		
参考链接	https://developer.android.com/reference/dalvik/system/DexClassLoader.html		

4.14 AES弱加密漏洞检测

漏洞名称	AES弱加密漏洞检测	威胁等级	中危
漏洞危害	在AES加密时，使用“AES/ECB/NoPadding”或“AES/ECB/PKCS5padding”的模式。ECB是将文件分块后对文件块做同一加密，破解加密只需要针对一个文件块进行解密，降低了破解难度和文件安全性。		
检测结果	安全		
修复意见	禁止使用AES加密的ECB模式，显式指定加密算法为：CBC或CFB模式，可带上PKCS5Padding填充。AES密钥长度最少是128位，推荐使用256位。		
参考链接	https://developer.android.com/reference/javax/crypto/Cipher.html		

4.15 Provider文件目录遍历漏洞检测

漏洞名称	Provider文件目录遍历漏洞检测	威胁等级	中危
漏洞危害	当Provider被导出且覆写了openFile方法时，没有对Content Query Uri进行有效判断或过滤。攻击者可以利用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。		
检测结果	安全		
修复意见	一般情况下无需覆写openFile方法，如果必要，对提交的参数进行“../”目录跳转符或其他安全校验。		
参考链接	https://developer.android.com/reference/android/content/ContentProvider.html#openFile(android.net.Uri,java.lang.String)		

4.16 activity绑定browserable与自定义协议漏洞检测

漏洞名称	activity绑定browserable与自定义协议漏洞检测	威胁等级	低危
漏洞危害	activity设置“android.intent.category.BROWSABLE”属性并同时设置了自定义的协议android:scheme意味着可以通过浏览器使用自定义协议打开此activity。可能通过浏览器对app进行越权调用。		
检测结果	安全		

修复意见	app对外部调用过程和传输数据进行安全检查或检验。
参考链接	https://developer.android.com/reference/android/content/Intent.html#CATEGORY_BROWSABLE

4.17 动态注册广播漏洞检测

漏洞名称	动态注册广播漏洞检测	威胁等级	中危
漏洞危害	使用registerReceiver动态注册的广播在组件的生命周期里是默认导出的。导出的广播可以导致拒绝服务、数据泄漏或是越权调用。		
检测结果	安全		
修复意见	使用带权限检验的registerReceiver API进行动态广播的注册。		
参考链接	https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver,android.content.IntentFilter, java.lang.String, android.os.Handler)		

4.18 开放socket端口漏洞检测

漏洞名称	开放socket端口漏洞检测	威胁等级	低危
漏洞危害	app绑定端口进行监听，建立连接后可接收外部发送的数据。攻击者可构造恶意数据对端口进行测试，对于绑定了IP 0.0.0.0的app可发起远程攻击。		
检测结果	安全		
修复意见	如无必要，只绑定本地ip127.0.0.1，并且对接收的数据进行过滤、验证。		
参考链接	https://developer.android.com/reference/java/net/Socket.html#Socket(java.net.InetAddress,int)		

4.19 Fragment注入漏洞检测

漏洞名称	Fragment注入漏洞检测	威胁等级	低危
漏洞危害	通过导出的PreferenceActivity的子类，没有正确处理Intent的extra值。攻击者可绕过限制访问未授权的界面。		

检测结果	安全
修复意见	当targetSdk大于等于19时，强制实现了isValidFragment方法；小于19时，在PreferenceActivity的子类中都要加入isValidFragment ，两种情况下在isValidFragment 方法中进行fragment名的合法性校验。
参考链接	https://developer.android.com/reference/android/preference/PreferenceActivity.html#isValidFragment(java.lang.String)

4.20 webview明文存储密码漏洞检测

漏洞名称	webview明文存储密码漏洞检测	威胁等级	高危
漏洞危害	在使用WebView的过程中忽略了WebView setSavePassword，当用户选择保存在WebView中输入的用户名和密码，则会被明文保存到应用数据目录的databases/webview.db中。如果手机被root就可以获取明文保存的密码，造成用户的个人敏感数据泄露		
检测结果	安全		
修复意见	使用WebView.getSettings().setSavePassword(false)来禁止保存密码		
参考链接	https://developer.android.com/reference/android/webkit/WebSettings.html#setSavePassword(boolean)		

4.21 unzip解压缩（ZipperDown）漏洞检测

漏洞名称	unzip解压缩（ZipperDown）漏洞检测	威胁等级	高危
漏洞危害	解压zip文件，使用getName()获取压缩文件名后未对名称进行校验。攻击者可构造恶意zip文件，被解压的文件将会进行目录跳转被解压到其他目录，覆盖相应文件导致任意代码执行。		
检测结果	安全		
修复意见	解压文件时，判断文件名是否有../特殊字符。		
参考链接	https://developer.android.com/reference/java/util/zip/ZipEntry.html#getName()		

4.22 未使用编译器堆栈保护技术漏洞检测

漏洞名称	未使用编译器堆栈保护技术漏洞检测	威胁等级	低危
漏洞危害	为了检测栈中的溢出，引入了Stack Canaries漏洞缓解技术。在所有函数调用发生时，向栈帧内压入一个额外的被称作canary的随机数，当栈中发生溢出时，canary将被首先覆盖，之后才是EBP和返回地址。在函数返回之前，系统将执行一个额外的安全验证操作，将栈帧中原先存放的canary和.data中副本的值进行比较，如果两者不吻合，说明发生了栈溢出。不使用Stack Canaries栈保护技术，发生栈溢出时系统并不会对程序进行保护。		
检测结果	安全		
修复意见	使用NDK编译so时，在Android.mk文件中添加：LOCAL_CFLAGS := -Wall -O2 -U_FORTIFY_SOURCE -fstack-protector-all		
参考链接	https://en.wikipedia.org/wiki/Stack_buffer_overflow#Stack_canaries		

4.23 未使用地址空间随机化技术漏洞检测

漏洞名称	未使用地址空间随机化技术漏洞检测	威胁等级	低危
漏洞危害	PIE全称Position Independent Executables，是一种地址空间随机化技术。当so被加载时，在内存里的地址是随机分配的。不使用PIE，将会使得shellcode的执行难度降低，攻击成功率增加。		
检测结果	安全		
修复意见	NDK编译so时，加入LOCAL_CFLAGS := -fpie -pie开启对PIE的支持。		
参考链接	https://en.wikipedia.org/wiki/Position-independent_code#Position-independent_executables		

4.24 动态链接库中包含执行命令函数漏洞检测

漏洞名称	动态链接库中包含执行命令函数漏洞检测	威胁等级	低危
	在native程序中，有时需要执行系统命令，在接收外部传入的参数		

漏洞危害	执行命令时没有做过滤或检验。攻击者传入任意命令，导致恶意命令的执行。
检测结果	安全
修复意见	对传入的参数进行严格的过滤
参考链接	http://baike.baidu.com/subview/627587/14965930.htm#2

4.25 随机数不安全使用漏洞检测

漏洞名称	随机数不安全使用漏洞检测	威胁等级	低危
漏洞危害	调用SecureRandom类中的setSeed方法。生成的随机数具有确定性，存在被破解的可能性。		
检测结果	安全		
修复意见	使用/dev/urandom或者/dev/random来初始化伪随机数生成器。		
参考链接	https://developer.android.com/reference/java/security/SecureRandom.html#setSeed(long)		

4.26 FFmpeg文件读取漏洞检测

漏洞名称	FFmpeg文件读取漏洞检测	威胁等级	低危
漏洞危害	使用了低版本的FFmpeg库进行视频解码。在FFmpeg的某些版本中可能存在本地文件读取漏洞，可以通过构造恶意文件获取本地文件内容。		
检测结果	安全		
修复意见	升级FFmpeg库到最新版		
参考链接	http://ffmpeg.org/		

4.27 libupnp栈溢出漏洞漏洞检测

漏洞名称	libupnp栈溢出漏洞漏洞检测	威胁等级	低危
漏洞危害	使用了低于1.6.18版本的的libupnp库文件。构造恶意数据包可造成缓冲区溢出，造成代码执行。		

检测结果	安全
修复意见	升级libupnp库到1.6.18版本或以上
参考链接	https://sourceforge.net/projects/pupnp/files/pupnp/libUPnP%201.6.18/

4.28 Webview组件远程代码执行（调用getClassLoader）漏洞检测

漏洞名称	Webview组件远程代码执行（调用getClassLoader）漏洞检测	威胁等级	中危
漏洞危害	使用低于17的targetSDKVersion，并且在Context子类中使用addJavascriptInterface绑定this对象。通过调用getClassLoader可以绕过google底层对getClass方法的限制。		
检测结果	安全		
修复意见	targetSDKVersion使用大于17的版本。		
参考链接	https://developer.android.com/reference/android/content/Context.html#getClassLoader()		

4.29 AES/DES硬编码密钥漏洞检测

漏洞名称	AES/DES硬编码密钥漏洞检测	威胁等级	高危
漏洞危害	使用AES或DES加解密时，采用硬编码在程序中的密钥。通过反编译拿到密钥可以轻易解密APP通信数据。		
检测结果	安全		
修复意见	密钥加密存储或是用过变形后进行加解密运算，切勿硬编码到代码中。		
参考链接	https://developer.android.com/reference/javax/crypto/spec/SecretKeySpec.html#SecretKeySpec(byte[],java.lang.String)		

4.30 Android应用克隆漏洞检测

漏洞名称	Android应用克隆漏洞检测	威胁等级	高危

漏洞危害	该漏洞影响使用WebView控件，开启file域访问并且未按安全策略开发的Android应用APP。检测方法为： 1.检测WebView中是否setAllowFileAccessFromFileURLs 或****setAllowUniversalAccessFromFileURLsAPI配置为true； 2.检测WebView是否可以直接被外部调用，并能够加载外部可控的HTML文件
检测结果	安全
修复意见	1.file域访问为非功能需求时，手动在Activity中配置setAllowFileAccessFromFileURLs或setAllowUniversalAccessFromFileURLs两个API为false。（Android4.1版本之前这两个API默认是true，需要显式设置为false） 2. 若需要开启file域访问，则设置file路径的白名单，严格控制file域的访问范围，具体如下： （1）固定不变的HTML文件可以放在assets或res目录下，file:///android_asset和file:///android_res 在不开启API的情况下也可以访问； （2）可能会更新的HTML文件放在/data/data/(app) 目录下，避免被第三方替换或修改； （3）对file域请求做白名单限制时，需要对“../../”特殊情况进行处理，避免白名单被绕过。 3. 避免App内部的WebView被不信任的第三方调用。排查内置WebView的Activity是否被导出、必须导出的Activity是否会通过参数传递调起内置的WebView等。 4. 建议进一步对APP目录下的敏感数据进行保护。客户端APP应用设备相关信息（如IMEI、IMSI、Android_id等）作为密钥对敏感数据进行加密。使攻击者难以利用相关漏洞获得敏感信息。
参考链接	http://www.cnvd.org.cn/webinfo/show/4365

附件A. 安全风险状况等级说明

安全风险状况等级说明	
1	良好状态：应用检测分值大于90分 信息系统处于良好运行状态，没有发现或只存在零星的低风险安全问题，此时只要保持现有安全策略就满足了本系统的安全等级要求。
2	预警状态：应用检测分值在75-90之间 信息系统中存在一些漏洞或安全隐患，此时需根据评估中发现的网络、主机、应用和管理等方面的问题对进行有针对性的加固或改进。
3	严重状态：应用检测分值在60-75之间 信息系统中发现存在严重漏洞或可能严重威胁到系统正常运行的安全问题，此时需要立刻采取措施，例如安装补丁或重新部署安全系统进行防护等等。
4	紧急状态：应用检测分值小于60分 信息系统面临严峻的网络安全态势，对组织的重大经济利益或政治利益可能造成严重损害。此时需要与其他安全部门通力协作采取紧急防御措施。

附件B. 移动安全架构设计基本原则

在应用系统软件开发设计的过程中，对应用系统的总体设计应当满足如下安全原则：

原则	说明
最小权限原则 Least Privilege	应用软件的每个模块如进程、用户只能访问当下所必需的信息或者资源。赋予每一个合法动作最小的权限，以保护数据以及功能避免受到错误或者恶意行为的破坏。
权限分离原则 Separation of Duties	对业务的操作、管理和审计权限应该由软件中的不同角色的用户分别承担；普通用户和管理员用户信息应该存放在不同的数据表中。
深度防御原则 Defense in Depth	在应用程序对业务数据进行处理的每个阶段都要考虑安全性问题，不能仅在某个阶段做安全防御，这样单点防御一旦被突破将造成安全风险。
容错保护原则 Fail Secure	当程序出现故障时或系统异常当系统失败时，可以进入到一个失败保护的状态。如果用户请求失败，系统仍可保障安全。
单点异常终止原则 Single Point of Failure	当用户提交数据超出预期时，应立即终止程序的执行，不要试图加以修正并继续执行下去。
外来代码安全原则 Least Third Party Components	严格控制第三方函数与插件的使用，对外来代码必须进行详细的安全测试。
代码重用原则 Leveraging Existing Components	尽可能的重用软件已有的模块，这样可以降低引入新的漏洞和攻击界面的可能性。
数据保护原则 Data Protection	对用户数据的保护功能应涵盖用户数据存储的完整性、用户数据传输保密性、数据传输的访问控制、剩余信息的保护、数据反转操作等内容；应对系统中关键数据（如用户密码等）的存储和网络传输时应采用加密保护，实用加密算法应该符合国际标准、国家标准和业界标准。
可审计原则 Auditing	在应用系统中设计审计日志记录的功能，并对应用系统产生的日志增加完备的审计功能。

开放设计原则 Open Design	开放设计与“不开放即安全”的原则相对而言，认为设计本身不应具有神秘感。这一原则的具体表现可以参见应用于加密设计的Kerchoff定律，“系统不应单纯依赖私密性，若落入敌人手中则毫无优势可言”；开放设计以提高系统兼容性和可扩展性。
抗抵赖原则 Anti Repudiation	对于涉及支付交易等重要的业务场景，系统设计应有效地防止通信双方抵赖，如采用电子证书签名等方式。
规范性 Standardization	系统设计所采用的安全技术和安全产品应符合国际标准、国家标准和业界标准，为系统的扩展升级、与其他系统的互联提供良好的基础。
可扩展性 Scalability	以当前业务安全需求为基础，充分考虑发展的需要，安全功能模块子系统以插件或接口方式以方便未来的扩展。
实用性 Practicable	安全功能设计需要尽可能的考虑投入产出比，同时尽量控制对用户体验的影响。
符合性 Regulatory Compliance	安全功能的设计尽可能的要符合国家规范、行业规范以及业界的通用标准，如等级保护等规范。