ICS 35.240
A 90

GA

中华人民共和国公共安全行业标准

GA/T XXXXX—20XX

移动互联网应用程序安全加固产品测评准则

Testing and evaluation criteria for Mobile App security shield products

（试　行）

XXXX－XX－XX 发布　　XXXX－XX－XX 实施

中华人民共和国公安部　　发布

目　　次

前　言

本标准按照GB/T 1.1-2009给出的规则起草。

本标准由公安部网络安全保卫局提出。

本标准由公安部信息系统安全标准化技术委员会归口。

本标准起草单位：国家计算机病毒应急处理中心、公安部网络安全保卫局七处、计算机病毒防治技术国家工程实验室、北京洋浦伟业科技发展有限公司、北京奇虎360公司、百度在线网络技术（北京）有限公司、北京智游网安科技有限公司、北京国信灵通网络科技有限公司

本标准主要起草人：陈建民、祝国邦、陆磊、黄一斌、刘威、曹鹏、杜振华、刘明君、蒋旭宪、包沉浮、郭训平、李江力、王文一、马天成

引　言

随着移动应用的普及和越来越多应用、使用，受限于移动操作系统（Android）平台自身的安全特性，移动应用的安全问题突出，比如被感染恶意代码、破解应用授权机制等二次打包行为，窃取应用图片、代码资源的剽窃行为，对应用运行时本地存储的Token等敏感信息的窃取行为，运行时的界面劫持、动态调试、动态注入截获篡改业务数据的攻击行为，以及由于代码的易读性导致的暴露更多业务风险的问题等等。

基于这些移动应用的诸多风险，市场已出现多家移动应用保护厂商，通过加固移动应用保护移动应用的安全性。各厂商的加固功能不尽相同，加固强度也不同；不同移动应用需要的安全保护能力也不相同。同时也出现月来越多的恶意程序使用安全加固来对抗安全检查，对恶意程序的传播起到推波助澜的作用。为规范、促进移动应用保护行业的发展，特制定本标准。

移动互联网应用程序安全加固产品测评准则

1 范围

本标准规定了移动互联网应用程序安全加固产品的受检要求、功能要求、技术要求、测试方法及评级方法。

本标准适用于移动互联网应用程序安全加固产品的开发和检测。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GA 243-2000 计算机病毒防治产品评级准则

GA 849-2009 移动终端病毒防治产品评级准则

GA/T 757-2008 程序功能检验方法

3 术语和定义

GA 243-2000、GA 849-2009、GA/T 757-2008界定的以及下列术语和定义适用于本标准。

3.1 二次打包 secondary packaging

通过编译、反编译等工具对App进行再次编译打包的方法或技术。

3.2 逆向分析 reverse analysis

是一种产品设计技术再现过程，即对一项目标产品进行逆向分析及研究，从而演绎并得出该产品的处理流程、组织结构、功能特性及技术规格等设计要素，以制作出功能相近，但又不完全一样的产品。

3.3 动态代码注入 dynamic code injection

通过OS提供的技术，可将特征代码写到目标程序空间并让其执行。

3.4 动态调式 dynamic debugging

动态调试是指使用调试工具对目标应用进行动态跟踪、分析的过程，帮助分析人员分析程序逻辑、获取中间数据等。

4 缩略语

下列缩略语适用于本文件。

App：移动应用程序(Mobile Application)

APK：Android安装包文件(Application Package File)

5 移动互联网应用程序安全加固产品描述

移动互联网应用程序安全加固产品是可以对的APP进行安全加固，防逆向、篡改保护，能够有效的对存储数据进行加密保护。

6 受检要求

6.1 检验周期

检验机构对程序版本发生重大升级或名称发生改变的移动互联网应用程序安全加固产品应进行检验；同时，可以根据安全威胁的发展情况对移动互联网应用程序安全加固产品进行专项检验。

6.2 测试用例要求

受检企业应提交其产品的测试用例。

6.3 资料要求

受检企业应提交产品研发人员的个人简历、产品中文使用说明书、核封完整的正式产品。

7 功能要求

7.1 恶意代码特征扫描

通常恶意代码借助各种技术隐藏其特征以躲避杀毒软件测试。要求移动应用在加固前需进行恶意代码测试，对确定是恶意代码的App不予以加固。

7.2 风险行为安全测试

对于新病毒无法通过恶意代码特征识别的风险，需通过风险行为安全测试发现移动应用是否带有恶意行为，恶意行为的定义参考《移动互联网恶意代码描述规范》

7.3 加固前的合法身份认证

加固前需进行开发者身份认证：提供营业执照，或者进行身份证实名认证等；

8 技术要求

8.1 移动应用程序安全加固产品的技术要求分类总体说明

本标准将移动 App 安全加固通用技术要求分为功能要求、性能要求、安全强度要求 3 大类。

其中，功能要求是对 App 安全加固产品应具备的功能提出具体要求，包括防篡改、防 Java 代码逆向、防注入、防脚本文件逆向、防 C 代码逆向、防调试、数据透明加密、设备指纹绑定、防界面劫持、防调试日志泄露、防短信劫持、安全键盘等等。

其中，性能要求对 App 安全加固产品应该达到的性能指标做出推定，例如从 Android2.3 到 Android ART 模式之间各个版本的适配、对第三方 OS 的适配、对 x86 和 ARM 平台的适配、加固后对性能、内存、CPU 的影响等等。

其中，安全强度要求是对 App 安全加固产品自身安全和防护能力提出各种要求，比如加密的颗粒

度、自身加密方法和密钥的保护、安全防护层级、源代码可逆的程度和比例，壳的破解难度等等

8.2 移动应用程序安全加固产品的安全等级说明

根据国内测评认证机构，测评技术和我国 App 安全加固产品开发现状，对移动 App 安全加固产品进行安全等级划分。

对App安全加固产品进行安全等级划分，安全等级分为一级、二级、三级3个逐步提高的级别。功能要求强度、性能要求强度、安全强度要求高低是安全等级划分的具体3个依据。

8.3 移动应用程序安全加固产品的适配和性能要求

1) App Java 代码防逆向

加固产品需提供防逆向反编译保护功能，防止移动应用程序被反编译。避免移动应用的业务逻辑等暴露，划进而导致仿冒、篡改、针对性攻击等攻击。

2) App 防篡改

防篡改保护： 加固产品需提供防篡改保护功能，防止移动应用程序的代码、图片、配置、布局等被增加、修改或删除。避免移动应用程序被二次打包、添加恶意代码等等。

3) App 防基础调试/注入

加固产品需提供反调试保护功能，防止移动应用在运行时被动态调试攻击、动态代码注入攻击。避免移动应用程序在运行时内存被修改、调用方法被 Hook 等，导致使用者信息泄露或经济损失。基础性的防调试/注入是能防止部分调试工具，但对部分主流工具的防预无效。

4) App 防高级调试/注入

加固产品需提供反调试保护功能，防止移动应用在运行时被动态调试攻击、动态代码注入攻击。避免移动应用程序在运行时内存被修改、调用方法被 Hook 等，导致使用者信息泄露或经济损失。

高级防调试/注入是指启动调试/注入工具时，程序即退出。无法搜索和修改内存信息和数据，能预防绝大多数调试攻击。

5) App 脚本文件防反编译

加固产品提供 Html，Html5, JS, Raw 等脚本文件加密保护功能，加密移动应用程序的 ,阻止对脚本实现的逻辑进行破解。

6) App 资源文件防反编译

加固产品提供资源文件加密保护功能，加密保护 assets、raw/res 等安装包目录下的文件，防止被第三方窃取、修改。

7) App 本地数据和内存透明加密

加固产品提供存储数据加密保护功能，加密保护存储在手机上的数据，防止第三方程序窃取、修改。

8) App So 库基础保护

SO 库是采用 C/C++语言开发的动态库。SO 库的逆向要求攻击者需要有一定的汇编语言的基础，相比 Java 的反编译，逆向分析的难度更高。但是 SO 库也是可以被逆向分析，因此需要对 SO 库进行加密的保护。保证 SO 库的汇编代码不被攻击者通过反汇编逆向工具例如 IDA 等进行软件逆向分析，保证被加密算法以及加密的密钥不被攻击者通过反汇编工具逆向分析得到

So 库基本保护是指采取了一些基本的安全机制，比如重定位信息打乱，So 库加入其它文件重新整体加密，对外仍然以 So 文件格式呈现，能部分被反编译。

9) App So 库增强保护

SO 库是采用 C/C++语言开发的动态库。SO 库的逆向要求攻击者需要有一定的汇编语言的基础，相比 Java 的反编译，逆向分析的难度更高。但是 SO 库也是可以被逆向分析，因此需要对 SO 库进行加密的保护。保证 SO 库的汇编代码不被攻击者通过反汇编逆向工具例如 IDA 等进行软件逆向分析，保证被

加密算法以及加密的密钥不被攻击者通过反汇编工具逆向分析得到。

So 增强保护是指，采取了包括但不限于这些安全增强手段：汇编代码压缩及加密保护；SO 库的 ELF 数据信息保护 ；SO 库整体加密 ；解密代码动态清除; So 跟 APK 绑定；防内存 dump 等等组合安全功能。保护的 So 对外不以 So 格式呈现。

10) 日志泄露保护

加固产品提供防日志泄露保护功能，禁用调试日志的输出，防止泄露用户敏感信息、程序逻辑信息。

8.4 移动应用程序安全加固产品的适配和性能要求

Android 系统的各个版本都有大量用户，比如某研究机构报道，截至 2015 年 1 月，Android 2.3 的用户比例大概为 9.1%，Android4.2 的用户比例 7.8%。App 加固产品如果不适配某个版本，意味着安全功能无法在该版本的 Android 系统上启动和发挥效果。

8.4.1 适配性要求

加固后的移动应用，不应有由于加固而导致的可用性问题和兼容性问题。移动应用所有方允许范围内除外。

1) 适配 Android Dalvik 模式(从 Android 2.3 到 Android 4.4.4)；
2) 适配 Android ART 模式；
3) 适配基于原生 Android 进行修改的其他 OS；
4) 适配 ARM, Intle x 86 平台芯片；
5) 适配机型的要求。

8.4.2 性能要求

加固后的移动应用，对性能（内存占用、CPU占用、启动时间）影响应在所有方的允许范围内。

1) 加固前后程序首次启动时间与加固前相比理想值不超过 2 秒，最多不超过 4 秒；
2) 加固前后程序再次启动时间与加固前相比理想值不超过 1 秒，最多不超过 2 秒；
3) 加固前后程序 CPU 占有率变化理想值不超过 5%，最多不超过 15%；
4) 加固前后程序内存占有率变化理想值不超过 10%，最多不超过 20%；
5) Android 代码安全加固后 APK 体积增加理想值不超过 20%，最多不超过 50%。

8.4.3 稳定性要求

加固后的移动应用，不影响移动应用程序的稳定性，应可以连续执行命令1小时不崩溃。

8.5 移动应用程序安全加固产品的安全强度要求

8.5.1 加密单元要求

对加密单元基本要求以 Dex 包体为加密单元；增强要求为以函数/方法，虚拟机指令为加密单元，达到更细的加密颗粒度。

8.5.2 加固壳安全保护要求

加固壳安全保护基本要求以自身实现的安全机制或开源级源代码混淆方案对加固壳进行安全保护；增强要求为以商业级源代码混淆工具对加固壳进行安全保护。

8.5.3 安全功能全面性要求

安全功能全面性至少要求App Java代码防逆向、App 防篡改、App防基础调试/注入和App So库基础保护，包括完整性校验，反调试，本地数据加密，脚本文件保护，So保护等功能；增强安全功能要求安全功能全面性，包括完整性校验，反调试，本地数据加密，脚本文件保护，So保护等功能。

8.5.4 攻击工具防范能力要求

1) 被保护代码防常用攻击工具能力的要求：防 Soot、dex2jar、APKtool、baksmali、IDA Pro 等静态工具；
2) 被保护代码防内存 Dump 攻击的要求：防 GDB、dvm、AndBug 加载机制攻击。

8.5.5 篡改工具防范能力要求

1) 被保护代码防常用静态篡改工具能力的要求：防 APKtool、smali 等工具；
2) 被保护代码防常用动态篡改工具能力的要求：防 ptrace 等动态篡改攻击方式。

9 测试方法

9.1 总体说明

测评方法与技术要求一一对应，他给出具体的测评方法来验证移动App加固产品是否达到技术要求中所提出的要求。它由测试环境，测试工具，测试方法和预期结果4个部分组成。

9.2 测试工具

表 1 测试工具

工具名称	介绍
adb	控制和管理 Android 模拟器或者真实 Android 设备
jdb	基于文本和命令行的 java 调试工具
gdb	动态调试工具
APKtool	APK 反编译/打包工具
Dex2jar	APK 反编译工具
Hijack	进程注入测试工具
DDMS	Android 调试监控器
Uedit	文本编辑软件
Beyond Compare	文本比较软件
7-zip	文件压缩/解压缩软件
豌豆荚	可通过电脑管理手机设备

9.3 测试环境

测试环境的示意图如下。测试手机通过 usb 方式与电脑相连，并连接无线热点。测试时，通过在电脑的命令行窗口中输入命令与测试手机进行交互。

图 1 测试网络拓扑

9.4 移动应用程序安全加固产品功能测试

9.4.1 App Java 代码防逆向

加固产品需提供防逆向反编译保护功能，防止移动应用程序被反编译。避免移动应用的业务逻辑等暴露，划进而导致仿冒、篡改、针对性攻击等攻击。

a) **测试方法：**

对 java 代码进行反编译测试，要求如下：

1) Davlik 代码反编译为 smali 文件；
2) Davlik代码反编译为jar文件。

b) **测试结果：**

经过测试，能防止以上的代码防反编译测试。

9.4.2 App 防篡改

防篡改保护：加固产品需提供防篡改保护功能，防止移动应用程序的代码、图片、配置、布局等被增加、修改或删除。避免移动应用程序被二次打包、添加恶意代码等等。

a) **测试方法：**

利用工具对加固后的App进行篡改测试，要求如下：

1) Dalvik代码篡改测试；
2) Android Manifest.xml文件篡改测试；
3) Assets文件篡改测试；
4) Res文件篡改测试；
5) 安装包添加或删除文件；
6) 修改安装包非签名信息的任意文件。

b）**预期结果**:

经过测试，能防止以上的篡改测试。

9.4.3 App 防调试/注入

加固产品需提供反调试保护功能，防止移动应用在运行时被动态调试攻击、动态代码注入攻击。避免移动应用程序在运行时内存被修改、调用方法被 Hook 等，导致使用者信息泄露或经济损失。

高级防调试/注入是指启动调试/注入工具时，程序即退出。无法搜索和修改内存信息和数据。

a) **测试方法：**

利用工具进行注入和调试攻击测试，要求如下：

1) java 层动态调试；
2) C 层动态调试；

3) 动态代码注入；

4) Hook 攻击。

b) **预期结果：**

经过测试，能防止全部测试，为高级防调试/注入保护。防止部分为基础的防调试和注入保护。

9.4.4 App 脚本文件防反编译

加固产品提供 Html、Html5、JS、Raw 等脚本文件加密保护功能，加密移动应用程序的，阻止对脚本实现的逻辑进行破解。

a) **测试方法：**

利用工具，测试脚本文件是否能被反编译，要求如下：

1) HTML/Html5 加密保护；

2) JavaScript 加密保护；

3) 其它脚本文件加密保护。

b) **预期结果：**

经过测试，对 html/Html5、js 等其它脚本均无法反编译。

9.4.5 App 资源文件防反编译

加固产品提供资源文件加密保护功能，加密保护 assets、raw/res 等安装包目录下的文件，防止被第三方窃取、修改。

a) **测试方法：**

利用测试工具，验证资源文件是否能被反编译，技术要求如下：

1) assets 资源加密保护；

2) raw 资源加密保护；

3) 布局文件加密保护。

b) **预期结果：**

经过测试，资源文件无法反编译，受到加密保护。

9.4.6 App 本地数据和内存透明加密

加固产品提供存储数据加密保护功能，加密保护存储在手机上的数据，防止第三方程序窃取、修改。

a) **测试方法：**

利用工具测试加固后的 App，在本地数据和内存上是否收到加密保护，具体要求如下：

1) 本地 SDcard 文件存储加密保护；

2) 本地 SharedPreferences 文件存储加密保护；

3) 本地 SQLite 文件存储加密保护；

4) 本地 File 文件存储加密保护。

b) **预期结果：**

经过测试，本地 SDcard 文件、SharedPreferences 文件、本地 SQLite 文件、本地 File 文件均收到存储加密保护。

9.4.7 App So 库保护

基本 So 保护指，仅仅进行了基本的安全手段，比如符号信息打乱，偏移，加固保护后的 So 对外仍然以 So 文件格式存在。

So 增强保护是指，采取了包括但不限于这些安全增强手段：汇编代码压缩及加密保护、SO 库的

ELF 数据信息保护、SO 库整体加密、解密代码动态清除、So 跟 APK 绑定、防内存 dump 等等组合安全功能。

a) 测试方法：

利用反编译等各种破解工具，对 So 文件进行分析，测试破解难度。检测被保护的客户端 App 中的 So 文件是否提供 So 文件保护功能，被保护的代码是否能被反汇编分析。技术要求如下：

1) So 代码经过初步保护，但能被反汇编；

2) So 代码经过初步保护，但是能找到全部或部分代码逻辑。

b) 预期结果：

根据以下反汇编的程度和还原代码的程度，判定是基础性保护，或者增强保护。

9.4.8 调试日志泄露保护：

加固产品提供防日志泄露保护功能，禁用调试日志的输出，防止泄露用户敏感信息、程序逻辑信息。

a) 测试方法：

检测被保护的客户端 App 是否提供调试日志输出保护功能，方法如下：

1） 运行客户端App；

2） 查看是否有调试日志输出。

b) 预期结果：

没有调试日志输出。

9.5 移动应用程序安全加固产品适配和性能测试

9.5.1 适配 Android Dalvik 模式(从 Android 2.3 到 Android 4.4.4)

确保未加固的包能在 Android2.3/4.0/4.1/4.2/4.3/4.4 系统上运行，测试加固后的包在 Android2.3/4.0/4.1/4.2/4.3/4.4 系统是否能运行。

a) 测试方法：

在 Android2.3/4.0/4.1/4.2/4.3/4.4 的手机上运行加固后的包，观察是否能正常运行。

b) 预期结果：

加固后的包在Android2.3/4.0/4.1/4.2/4.3/4.4系统能运行。

9.5.2 适配 Android 4.4.4, Android 5.0 等 ART 模式测试

确保未加固的包能在 Android 4.4.4、Android 5.0、Android 5.1 等 ART 模式系统上运行，测试加固后的包在 Android 4.4.4、Android 5.0 系统是否能运行。

a) 测试方法：

在 Android 4.4.4、Android 5.0 的手机上运行加固后的包，观察是否能正常运行。

b) 预期结果：

加固后的包在Android 4.4.4、Android 5.0、Android 5.1等ART模式系统上运行。

9.5.3 适配基于原生 Android 进行修改的其他 OS

确保适配基于原生 Android 进行修改的其他 OS 系统上运行，测试加固后的包在适配基于原生 Android 进行修改的其他 OS 是否能运行。

a) 测试方法：

在适配基于原生 Android 进行修改的其他 OS 运行加固后的包，观察是否能正常运行。

b) 预期结果：

加固后的包在适配基于原生 Android 进行修改的其他 OS 系统能运行。

9.5.4 适配 ARM, Intle x 86 平台芯片测试

确保未加固的包能在 ARM, Intle x 86 平台系统上运行，测试加固后的包在 ARM, Intle x 86 平台系统是否能运行。

a) 测试方法：

分别选用两款采用在 ARM, Intle x 86 芯片的手机，测试加固后的包在 ARM, Intle x 86 平台系统是否能运行。

b) 预期结果：

在使用ARM, x86处理器芯片的手机上均可以运行加固包。

9.5.5 适配机型数量级测试

a) 测试方法：

在尽可能数量多的主流设备和系统测试，测试加固后 App 的适配性，要求如下：

1) 0-200 款，适配主要的机型；

2) 200-400 款，适配大多数机型；

3) 400 款以上，适配绝大多数机型。

b) 预期结果：

根据适配机型的数量级，对适配性进行判断。

9.5.6 加固前后程序首次启动时间与加固前相比分别不超过 4/3/2 秒测试

检测加固后的包在运行时，比较程序首次启动时间比原包首次启动的的时间差距，并记录时间差距。

a) 测试方法：

在一款手机上运行加固前后首次启动的 App，记录时间差。

b) 预期结果：

前后启动时间差在 4/3/2 秒的范围内。

9.5.7 加固前后程序再次启动时间与加固前相比分别不超过 2/1.5/1 秒测试

检测加固后的包在运行时，比较程序再次启动时间比原包再次启动的的时间差距，并记录时间差距。

a) 测试方法：

在一款手机上运行加固前后再此启动的 App，记录时间差。

b) 预期结果：

前后启动时间差在 2/1.5/1 秒的范围内。

9.5.8 加固前后程序 CPU 占有率变化不超过 15%/10%/5%测试

检测加固后的包在运行时，平均 CPU 占用与未加固包相比的差距。

a) 测试方法：

在同一个设备上运行加固前后同一个 App，记录 CPU 占用的数据。

b) 预期结果：

加固前后程序CPU占有率变化不超过15%/10%/5%。

9.5.9 加固前后程序内存占有率变化不超过 20%/15%/10%测试

检测加固后的包在运行时，平均内存占用与未加固包相比的差距。

a) 测试方法：

在同一个设备上运行加固前后同一个 App，记录内存占用的数据。

b) 预期结果：

加固前后程序内存占有率变化不超过20%/15%/10%。

9.5.10 Android 代码安全加固后 APK 体积增加不超过 50%/30%/20%测试

比较加固前后 App 的文件大小，Android 代码安全加固后 APK 体积增加分别不超过 50%/30%/20%;

a) 测试方法：

比较加固前后 App 的文件大小，记录 Android 代码安全加固后 APK 体积增加差距数值。

b) 预期结果：

Android代码安全加固后APK体积增加分别不超过50%/30%/20%;

9.5.11 稳定性测试

加固后的 App 持续运行、不崩溃、表现稳定。

a) 测试方法：

使用 monkey 命令运行加固后的 App 一小时，测试是否出现不稳定或崩溃现象。

b) 预期结果：

使用monkey命令运行App一小时，运行稳定。

9.6 移动应用程序安全加固产品安全强度测试

9.6.1 加密单元测试

测试 App 加固的产品的加密单元颗粒度大小

a) 测试方法：

通过解包，测试和确认 App 加固产品的加密单元。

b) 预期结果：

App加固加密的单元为dex包，函数/方法，虚拟机指令。

9.6.2 对自身壳的安全保护能力测试

App 加固产品对自身壳的防护程度，决定着壳的安全程度。

a) 测试方法：

通过解包，测试 App 加固开发者对自身的壳是否有安全措施，比如密钥和密码隐藏，以自身逻辑实现壳的保护，或者是以开源的源代码混淆工具，或者商业级的源代码混淆工具进行保护。

b) 预期结果：

加固壳未采取保护措施，以自身逻辑实现保护，或以开源/商业级源代码混淆工具实现保护。

9.6.3 安全功能的完整性测试

a) 测试方法：

通过渗透测试，了解安全功能全面性，测试是否包括完整性校验，反调试，本地数据加密，脚本文件保护，So 保护等功能。

b) 预期结果：

包括或不包括完整性校验，反调试，本地数据加密，脚本文件保护，So 保护等功能。

9.6.4 防基础性的静态反编译攻击测试

被保护代码防常用攻击工具能力的要求：防 Soot、dex2jar、APKtool、baksmali、IDA Pro 等工具 。

a) 测试方法：

使用 Soot、dex2jar、APKtool、baksmali、IDA Pro 等工具进行反编译，检测是否能防止反编译。

b) 预期结果：

防Soot、dex2jar、APKtool、baksmali、IDA Pro等工具的反编译攻击。

9.6.5 防内存 dump 攻击测试

被保护代码防内存 Dump 攻击的要求为防 GDB、dvm 加载机制攻击

a) 测试方法：

进行内存 dump 攻击测试，检测是否能防止内存 dump。

b) 预期结果：

防内存dump攻击。

9.6.6 防基础性的静态篡改攻击测试

被保护代码防常用静态篡改工具能力的要求为防 APKtool、smali 等工具

a) 测试方法：

使用 APKtool、smali 等静态工具进行篡改，测试是否能防止篡改。

b) 预期结果：

防止APKtool、smali等工具的篡改。

9.6.7 防动态篡改攻击测试

被保护代码防常用动态篡改工具能力的要求为防 ptrace 等动态篡改攻击方式

a) 测试方法：

使用 ptrace 等动态篡改攻击，测试是否能防止 ptrace 等动态篡改攻击。

b) 预期结果：

能够防止ptrace等动态篡改攻击。

10 评级方法

根据功能要求、性能要求、保证要求的要求项强度和技术实现难度，将移动互联网应用程序加固产品划分为基本级和增强级，具体划分细则见表2：

表 2 级别划分表

分类	要求项	基本级	增强级
功能要求	App Java 代码防逆向	▲	▲
	App 防篡改	▲	▲
	App 防基础调试/注入	▲	▲
	App 防高级调试/注入		▲
	App 脚本文件防反编译		▲

	App 资源文件防反编译				▲
	App 本地数据和内存透明加密				▲
	App So 库基础保护			▲	▲
	App So 库增强保护				▲
	日志泄露保护				▲
性能要求	适配模式	Android Dalvik 模式(从 Android 2.3 到 Android 4.4.4)		▲	▲
		Android ART 模式			▲
	适配基于原生 Android 进行修改的其他 OS			▲	▲
	适配 ARM, Intle x 86 平台芯片				▲
	适配机型	主要的手机厂家的移动设备		▲	▲
		大多数机型			▲
		绝大多数机型			▲
	加固前后程序首次启动时间与加固前相比延时（秒）		≤4	▲	
			≤2		▲
	加固前后程序再次启动时间与加固前相比延时（秒）		≤2	▲	
			≤1		▲
	加固前后程序 CPU 占有率变化比例（%）		≤15	▲	
			≤5		▲
	加固前后程序内存占有率变化比例（%）		≤20	▲	
			≤10		▲
	Android 代码安全加固后 APK 体积增加比例（%）		≤50	▲	
			≤20		▲
	稳定性方面，加固后的 App 连续执行命令一小时不崩溃			▲	▲
保证要求	加密单元要求	以最基本的 Dex 包体为加密单元		▲	
		以函数/方法，虚拟机指令为加密单元			▲
	加固壳安全保护	采用自身实现的安全机制或开源级源代码混淆方案		▲	
		商业级源代码混淆工具			▲
	安全功能全面性	完整性校验			▲
		反调试			▲
		本地数据加密			▲
		脚本文件保护			▲
		So 保护		▲	▲
	攻击工具防范	防 Soot、dex2jar、APKtool、baksmali、IDA Pro 等静态工具		▲	▲
		防 GDB、dvm、AndBug 加载机制攻击			▲
	篡改工具防范	防 APKtool、smali 等工具		▲	▲
		防 ptrace 等动态篡改攻击			▲

注：“▲”为该等级产品必备要求。