

|  |  |
| --- | --- |
| 文档编号 | AH-JS-NTNSH-190920 |
| 版本编号 | Ver 1.0 |
| 密 级 | 商业机密 |
| 日 期 | 2019-09-20 |

|  |
| --- |
| 南通农商行金贝  Android客户端安全测试报告 |

|  |
| --- |
| **© 2019安恒信息** |
| **版权所有，未经授权，严禁复制、编辑和传播！** |

* **版权声明**

本文中的所有信息均为南通农商行内部信息，务请妥善保管，未经南通农商行及安恒信息明确作出的书面许可，不得为任何目的、以任何形式或手段（包括电子、机械、复印、录音或其他形式）对本文档的任何部分进行复制、存储、引入检索系统或者传播。

* **版本变更记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本号** | **拟制／修改日期** | **拟制／修改人** | **修改记录** | **批准人** |
| 1.0 | 2019-9-20 | 孙赟 | 创建文档 | 吕翌澍 |

|  |
| --- |
| 目录 |
| [一. 概述 1](#_Toc19885656)  [1.1 项目目标 1](#_Toc19885657)  [二. 测试结论 2](#_Toc19885658)  [三. 测试综述 5](#_Toc19885659)  [3.1 测试对象 5](#_Toc19885660)  [3.2 测试人员 5](#_Toc19885661)  [3.3 风险定级依据 5](#_Toc19885662)  [四. 测试结果 6](#_Toc19885663)  [4.1 客户端静态安全 6](#_Toc19885664)  [4.1.1 安全包签名 6](#_Toc19885665)  [4.1.2 反编译保护 6](#_Toc19885666)  [4.1.3 应用完整性校验 8](#_Toc19885667)  [4.1.4 组件导出安全【中风险】 9](#_Toc19885668)  [4.1.5 AllowBackup备份权限配置安全性【中风险】 9](#_Toc19885669)  [4.1.6 Debuggable调试权限配置安全性 10](#_Toc19885670)  [4.1.7 activity组件拒绝服务漏洞 11](#_Toc19885671)  [4.1.8 activity界面越权访问漏洞 12](#_Toc19885672)  [4.1.9 activity界面劫持 13](#_Toc19885673)  [4.1.10 Fragment注入 13](#_Toc19885674)  [4.1.11 service组件拒绝服务漏洞 14](#_Toc19885675)  [4.1.12 content provider SQL注入漏洞 15](#_Toc19885676)  [4.1.13 content provider 路径遍历漏洞 16](#_Toc19885677)  [4.2 客户端数据安全 17](#_Toc19885678)  [4.2.1 本地文件权限配置【低风险】 17](#_Toc19885679)  [4.2.2 本地文件内容安全 18](#_Toc19885680)  [4.2.3 本地日志内容安全 18](#_Toc19885681)  [4.2.4 本地内存内容安全 19](#_Toc19885682)  [4.3 客户端运行时安全 20](#_Toc19885683)  [4.3.1 输入记录保护 20](#_Toc19885684)  [4.3.2 屏幕录像保护【中风险】 21](#_Toc19885685)  [4.3.3 进程注入保护 21](#_Toc19885686)  [4.3.4 ROOT环境检测【中风险】 22](#_Toc19885687)  [4.4 安全策略 23](#_Toc19885688)  [4.4.1 密码复杂度策略 23](#_Toc19885689)  [4.4.2 账户锁定策略 24](#_Toc19885690)  [4.4.3 单点登录限制策略【中风险】 25](#_Toc19885691)  [4.4.4 会话超时策略 26](#_Toc19885692)  [4.4.5 UI敏感信息安全 27](#_Toc19885693)  [4.4.6 界面切换保护【低风险】 28](#_Toc19885694)  [4.4.7 安全退出 29](#_Toc19885695)  [4.5 通信安全 29](#_Toc19885696)  [4.5.1 通信协议有效性 29](#_Toc19885697)  [4.5.2 通信数据加密 30](#_Toc19885698)  [4.5.3 证书强校验【中风险】 31](#_Toc19885699)  [4.5.4 软件升级缺陷 32](#_Toc19885700)  [4.5.5 数据防重放【中风险】 32](#_Toc19885701)  [五. 安全等级及建议 34](#_Toc19885702) |

1. 概述
   1. 项目目标

根据南通农商行对金贝安卓客户端的安全性需求，授权杭州安恒信息技术股份有限公司（以下简称“安恒信息”）对其进行安全测试；并根据本次安全测试过程中所发现的系统脆弱点提供改进方案，以指导南通农商行开展安全整改、完善安全策略，降低安全风险，使南通农商行金贝安卓客户端的安全保障能力符合国家的相关政策法规和自身业务的安全需求。

1. 测试结论

根据与南通农商行所签订的合同之内容，安恒信息资深安全服务工程师于2019年09月16日至2019年09月20日对金贝安卓客户端开展全面的安全测试，共发现安全漏洞9个，其中中危7个、低危2个。存在的安全隐患主要包括组件导出安全、AllowBackup备份权限配置安全、本地文件权限配置、屏幕录像保护、ROOT环境监测、单点登录策略、界面切换保护、证书强校验、数据防重放等安全漏洞。

**经过整体安全风险分析，可以判断南通农商行金贝安卓版客户端正面临的安全风险等级为：中**。

测试结果综述：

|  |  |  |  |
| --- | --- | --- | --- |
| **测试分类** | **测试项目** | **测试结果** | **漏洞等级** |
| 客户端静态安全 | 安全包签名 | **安全** | 无风险 |
| 反编译保护 | **安全** | 无风险 |
| 应用完整性校验 | **安全** | 无风险 |
| 组件导出安全 | **不安全** | 中风险 |
| AllowBackup备份权限配置安全性 | **不安全** | 中风险 |
| Debuggable调试权限配置安全性 | **安全** | 无风险 |
| activity组件拒绝服务漏洞 | **安全** | 无风险 |
| activity界面越权访问漏洞 | **安全** | 无风险 |
| activity界面劫持 | **安全** | 无风险 |
| Fragment注入 | **安全** | 无风险 |
| service组件拒绝服务漏洞 | **安全** | 无风险 |
| content provider SQL注入漏洞 | **安全** | 无风险 |
| content provider 路径遍历漏洞 | **安全** | 无风险 |
| 客户端数据安全 | 本地文件权限配置 | **不安全** | 低风险 |
| 本地文件内容安全 | **安全** | 无风险 |
| 本地日志内容安全 | **安全** | 无风险 |
| 本地内存内容安全 | **安全** | 无风险 |
| 客户端运行时安全 | 输入记录保护 | **安全** | 无风险 |
| 屏幕录像保护 | **不安全** | 中风险 |
| 进程注入保护 | **安全** | 无风险 |
| ROOT环境检测 | **不安全** | 中风险 |
| 安全策略 | 密码复杂度策略 | **安全** | 无风险 |
| 账户锁定策略 | **安全** | 无风险 |
| 单点登陆限制策略 | **不安全** | 中风险 |
| 会话超时策略 | **安全** | 无风险 |
| UI敏感信息安全 | **安全** | 无风险 |
| 界面切换安全 | **不安全** | 低风险 |
| 安全退出 | **安全** | 无风险 |
| 通信安全 | 通信协议有效性 | **安全** | 无风险 |
| 通信数据加密 | **安全** | 无风险 |
| 证书强校验 | **不安全** | 中风险 |
| 软件升级缺陷 | **安全** | 无风险 |
| 数据防重放 | **不安全** | 中风险 |

*注释：“风险等级”指目标系统的整体安全状况评估结果，等级确定依据本文“3.3风险定级依据”表。*

系统漏洞统计图如下：

1. 测试综述
   1. 测试对象

渗透测试限定于以下对象：

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **对象名称** | **APP的文件名** | **版本** |
| 1 | 南通农商行金贝Android客户端 | jinbei2.5.3Release\_0912\_signed.apk | 2.5.3 |

* 1. 测试人员

安恒信息组成专项安全渗透测试小组，任命（项目经理名称）为项目经理，（填写客户名称）相关工作人员配合。项目组成人员如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **项目角色** | **姓名** | **联系方式** | **项目职责** |
| 1 | 项目经理 | 吕翌澍 | 18552279554 | 项目实施计划、项目实施过程管理 |
| 2 | 质量监控 | 冯丹 | 13588737080 | 项目实施过程管理、质量控制 |
| 3 | 安服工程师 | 孙赟 | 18052128329 | 负责具体渗透实施 |

* 1. 风险定级依据

风险等级参照以下标准：

|  |  |  |
| --- | --- | --- |
| **序号** | **风险等级** | **风险描述** |
| 1 | 高风险 | 可对APP应用造成重大损失，如信息篡改、数据损坏以及远程控制 |
| 2 | 中风险 | 可获取APP应用的敏感信息，并结合其他安全漏洞可能影响目标网站的正常运营 |
| 3 | 低风险 | 可获取APP应用的少量敏感信息，一般不会导致严重的安全事件 |

1. 测试结果
   1. 客户端静态安全
      1. 安全包签名
   * 问题描述

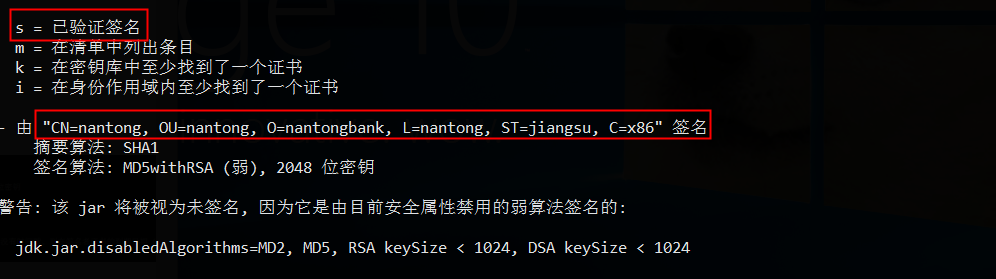
在Android系统中包名相同的两个Android应用会被认为是同一个应用，同时安装时会尝试覆盖安装。而覆盖安装时，签名证书必须一致，否则会被拒绝。如果Android应用没有使用应用所有者的证书进行签名，那么在以后维护过程中应用将存在脱离控制的风险。

* + 风险等级

无

* + 测试过程

校验安装包签名，不存在问题。



* + 安全建议

建议客户端使用APP从属方证书进行签名后进行发布而不是使用第三方开发商的证书进行签名，以防开发商内部监管异常，证书滥用的情况出现。

* + 1. 反编译保护
  + 问题描述

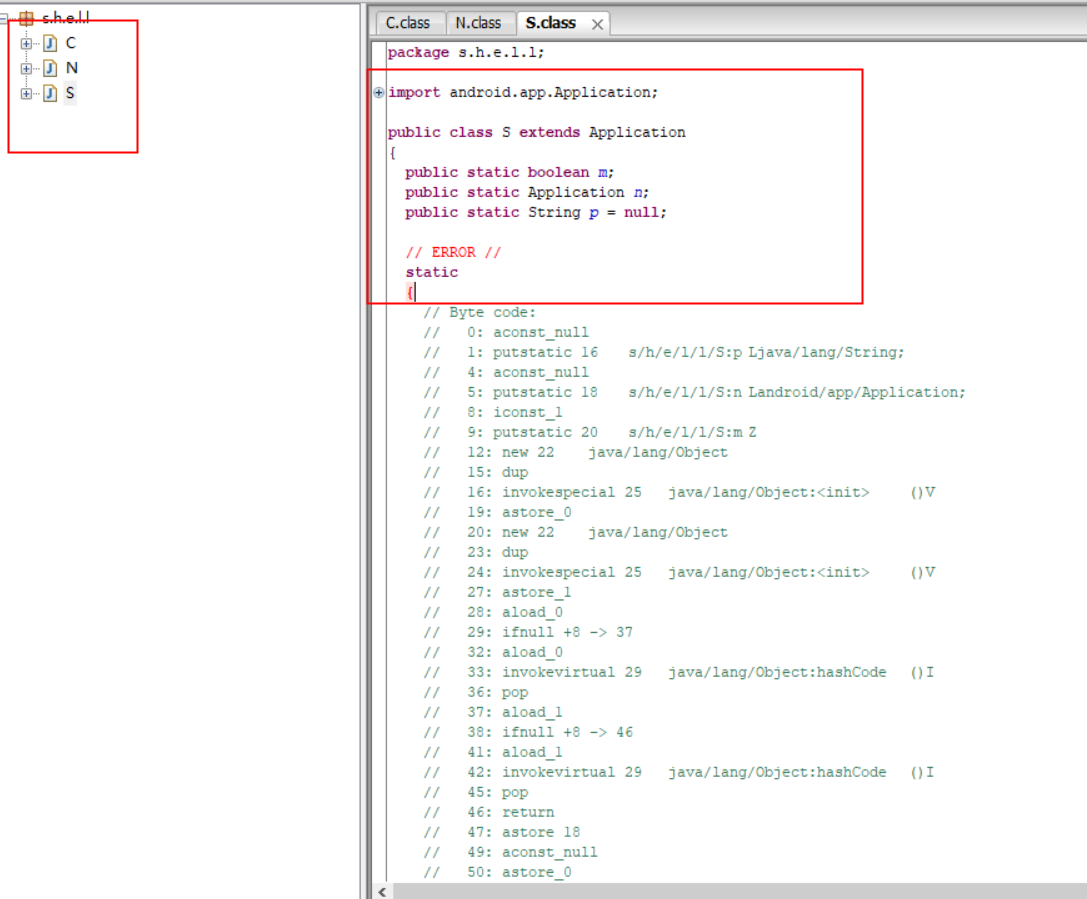
测试客户端安装程序，判断是否能反编译为源代码，java代码和so文件是否存在代码混淆等保护措施。未作保护的java代码，可以轻易分析其运行逻辑，并针对代码中的缺陷对客户端或服务器端进行攻击。

* + 风险等级

无

* + 测试过程

反编译之后的代码无法恢源代码，所以安全：



* + 安全建议

建议对客户端进行加壳，防止攻击者通过工具直接反编译获取到客户端完整源代码，至少要对核心代码进行混淆。

* + 1. 应用完整性校验
  + 问题描述

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，攻击者可能会通过篡改客户端程序窃取手机用户的隐私信息。

* + 风险等级

无

* + 测试过程

修改图标文件后重新打包app，安装之后无法打开应用。



* + 安全建议

建议应用在启动时对客户端自身的完整性进行校验，在验证逻辑中不使用MANIFEST.MF中的数据作为验证凭证，同时需验证是否有不属于该版本的新文件引入。

* + 1. 组件导出安全【中风险】
  + 问题描述

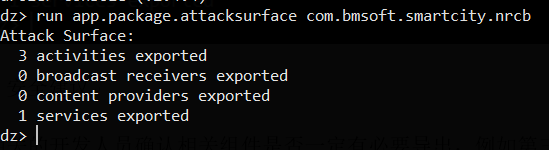
如果没有对导出组件配置进行严格的控制，那么其它的第三方APP就可能通过调用这个导出组件的接口来访问原本没有声明权限的功能，构成本地权限提升，甚至去访问进程中的其他资源造成敏感信息泄露。

* + 风险等级

中

* + 测试过程

存在组件导出问题。



* + 安全建议

与相关的开发人员确认相关组件是否一定有必要导出，例如第三方支付接口是必须导出的。若无必要，关闭相应的导出属性。

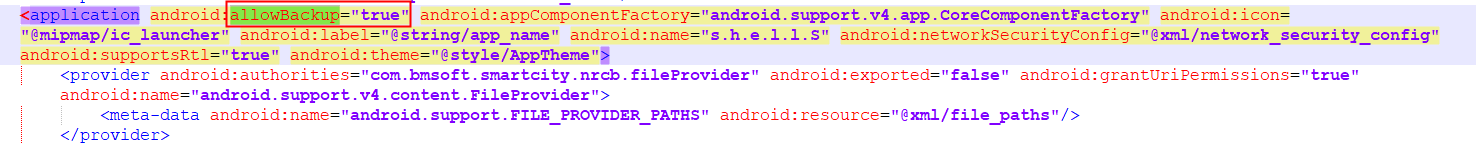
* + 1. AllowBackup备份权限配置安全性【中风险】
  + 问题描述

若配置文件中的allowBackup属性值没有被设置为false，攻击者可通过adb backup对应用数据进行备份，在无root的情况下可以导出应用中存储的所有数据，造成用户数据泄露。

* + 风险等级

中风险

* + 测试过程



* + 安全建议

1. 将AndroidMainfast.xml文件下Allowbackup属性设置为false。
2. 使用良好加密，例如启动应用时计算设备唯一标识，从服务端获取对应密钥后解密本地配置。
   * 1. Debuggable调试权限配置安全性
   * **问题描述**

被测应用的AndroidManifest.xml文件中Debuggable属性值被设置为true，可以设置断点来控制程序的执行流程，在应用程序运行时修改其行为。

* + **风险等级**

无

* + **测试过程**

Debuggable属性在release版本下默认为false



* + **安全建议**

1. 将AndroidMainfast.xml文件下Debuggable属性设置为false。

* + 1. activity组件拒绝服务漏洞
  + **问题描述**

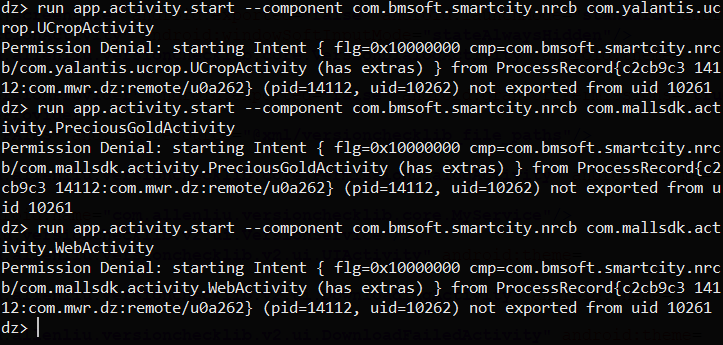
Android应用使用Intent机制在组件之间传递数据，如果应用在使用getIntent()，getAction()，Intent.getXXXExtra()获取到空数据、异常或者畸形数据时没有进行异常捕获，应用就会发生Crash，应用不可使用（本地拒绝服务）。恶意应用可通过向受害者应用发送此类空数据、异常或者畸形数据从而使应用产生本地拒绝服务。

* + **风险等级**

无

* + **测试过程**

经过测试安全。



* + **安全建议**

出于安全考虑，应将不必要的组件导出，防止引起拒绝服务，尤其是杀毒、安全防护、锁屏防盗等安全应用；在AndroidMenifest.xml文件中，将相应组件的 android:exported属性设置为 false。

建议处理通过 Intent.getXXXExtra() 获取的数据时进行以下判断，以及用 try catch 方式进行捕获所有异常，以防止应用出现拒绝服务漏洞：

空指针异常；

类型转换异常；

数组越界访问异常；

类未定义异常；

其他异常；

* + 1. activity界面越权访问漏洞
  + **问题描述**

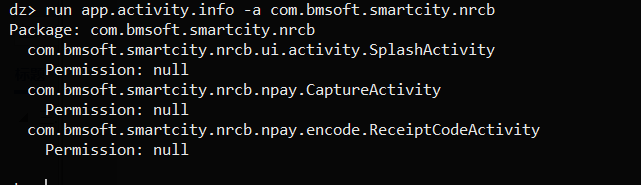
Activity是安卓应用组件，提供与用户进行交互的界面。如果应用对权限控制不当，可以绕过登录界面直接显示该界面。

* + **风险等级**

无

* + **测试过程**

测试过程中未发现这3个activity可以越权访问。



* + **安全建议**

关闭AndroidManifest.xml中的Activity组件导出权限，对于必须导出的组件必须限制于授权用户或者应用组件。

* + 1. activity界面劫持
  + **问题描述**

我们登陆正常的APP界面时，被恶意程序监听到，模仿正常界面启动我们的Activity；例如我们支付界面被劫持后，恶意软件启动自己的Activity来模仿支付界面，诱使你输入账号和密码；从而达到劫持效果。

* + **风险等级**

无

* + **测试过程**

通过工具无法劫持改app的activity的组件。

* + **安全建议**

1. 开启守护进程，当发现应用程序不在栈顶时，在屏幕最上层创建一个悬浮小窗口（提示信息与客户确定），以提醒用户。

2. 使用抢占式，即与劫持程序抢占栈顶。

3. 在应用切到后台时，在通知栏弹出通知提示。

* + 1. Fragment注入
  + **问题描述**

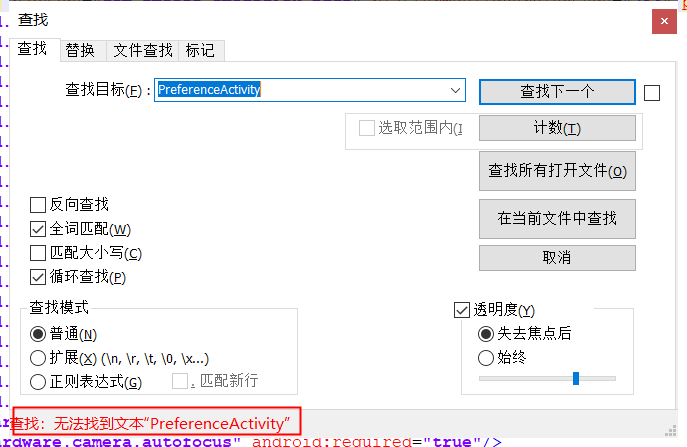
Activity可包含多个Fragment来展示界面，PreferenceActivity是支持Fragment的基类activity，其根据传入的参数EXTRA\_SHOW\_FRAGMENT， (‘:android:show\_fragment’)动态创建fragment实现界面展示。 当PreferenceActivity的activity是属性为export，PreferenceActivity不检查传入的参数直接根据其构建对象时，可以构造intent中的extra数据，调用应用内部的任意fragment。fragment注入攻击可导致应用的敏感信息泄露、远程代码执行或者应用崩溃。

* + **风险等级**

无

* + **测试过程**

通过查看AndroidMainfest文件，未发现暴露的组件有PreferenceActivity。



* + **安全建议**

1．如果应用的Activity组件不必要导出，或者组件配置了intent filter标签，建议显示设置组件的“android:exported”属性为false

2．重写isValidFragment方法，验证fragment来源的正确性

* + 1. service组件拒绝服务漏洞
  + **问题描述**

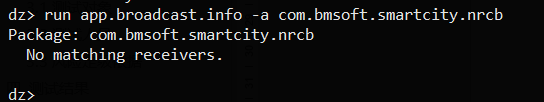
Broadcast Receiver是Android中四大组件用于处理广播事件的组件，若存在配置不当则其他应用可以伪装发送广播从而可造成信息泄露，拒绝服务攻击等。

* + **风险等级**

无

* + **测试过程**

可以看到没有BroadCast Reciever。



* + **安全建议**

被测应用的AndroidManifest.xml文件中，设置receiver的android:exported 属性为false或者通过设置自定义权限来限制对receiver的访问。

* + 1. content provider SQL注入漏洞
  + **问题描述**

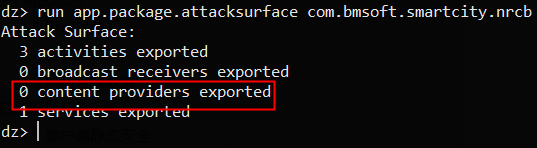
Content Provider组成Apk的四个组件之一，是应用程序之间共享数据的容器，可以将应用程序的指定数据集提供给第三方的App，如果设置了导出权限，可能被系统或者第三方的App直接调出并使用。Content Provider导出可能导致程序内部的敏感信息泄露，SQL注入等风险。

* + **风险等级**

无

* + **测试过程**

不存在可导出的content provider 组件。



* + **安全建议**

在被测应用的AndroidManifest.xml文件中，设置provider的android:exported 属性为false或者通过设置自定义权限来限制对provider的访问。

* + 1. content provider 路径遍历漏洞
  + **问题描述**

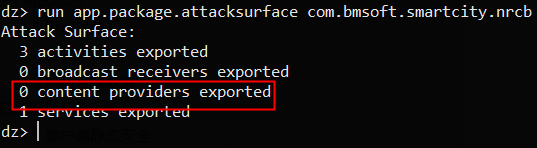
Content Provider组成Apk的四个组件之一，是应用程序之间共享数据的容器，可以将应用程序的指定数据集提供给第三方的App，如果设置了导出权限，可能被系统或者第三方的App直接调出并使用。Content Provider导出可能导致程序内部的敏感信息泄露，路径遍历等风险。

* + **风险等级**

无

* + **测试过程**

不存在可导出的content provider 组件。



* + **安全建议**

在被测应用的AndroidManifest.xml文件中，设置provider的android:exported 属性为false或者通过设置自定义权限来限制对provider的访问。

* 1. 客户端数据安全
     1. 本地文件权限配置【低风险】
  + 问题描述

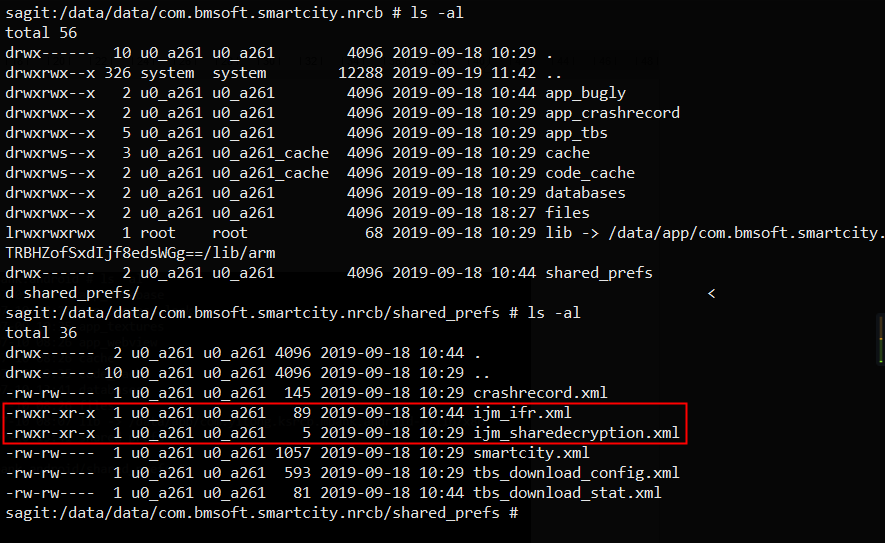
本地文件权限配置不当，可能导致信息泄漏问题。所造成的风险视具体的泄露信息而定，敏感信息泄露可能会方便恶意程序窃取凭据。

* + 风险等级

低风险

* + 测试过程

shared\_pref文件夹下的ijm\_ifr.xml和smartcity.xml的文件权限设置有执行权限。建议与开发人员确认是否必需权限。



* + 安全建议

建议合理设置文件权限，正常的文件权限最后三位应为空（类似“rw-rw----”），目录则允许多一个执行位（类似“rwxrwx—x”）。

* + 1. 本地文件内容安全
  + 问题描述

检查APP数据目录中的文件中是否包含敏感内容。包括文本文件、二进制文件、SharedPreference等XML文件、WebView等数据库文件等等。

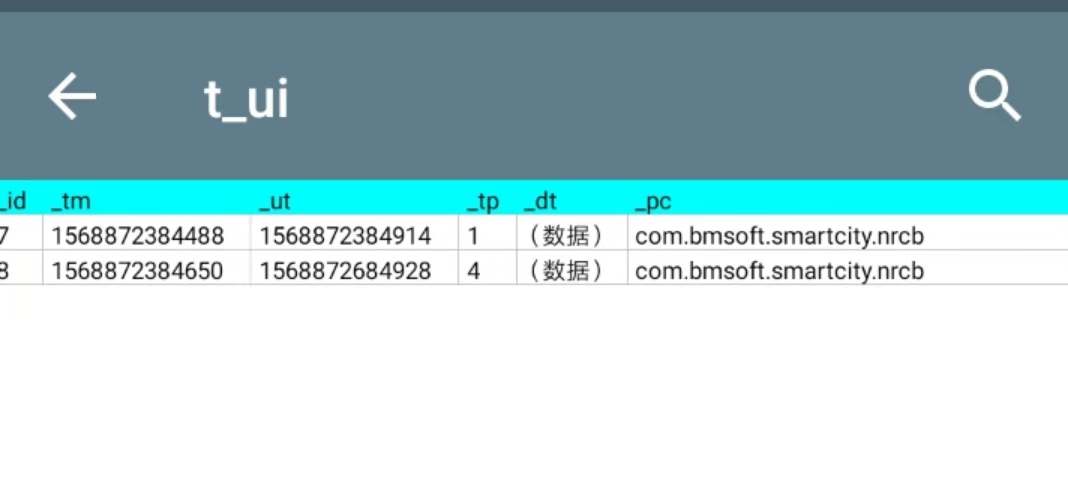
视具体的泄露信息而定，敏感信息泄露可能会方便恶意程序窃取凭据，或者泄露一些原本不希望被用户看到的内容。

* + 风险等级

无风险

* + 测试过程

本地文件内容中未发现敏感数据：



* + 安全建议

避免直接存储明文或者简单编码的敏感信息在本地。

* + 1. 本地日志内容安全
  + 问题描述

应用运行时可能会将一些应用的日志打印出来以方便开发者调试和测试，但是在应用上线后，这些信息可能会导致敏感信息泄漏的问题。

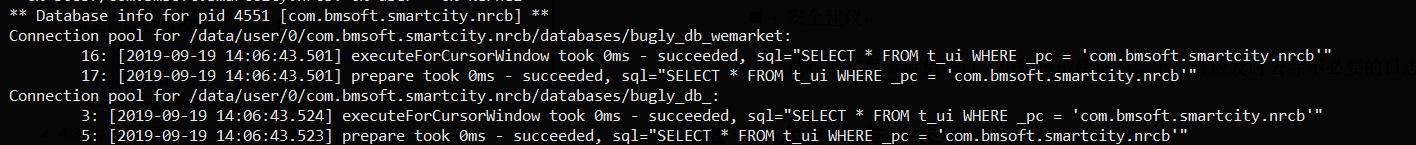
其风险视具体的泄露信息而定，敏感信息泄露可能会方便恶意程序窃取凭据。

* + 风险等级

无

* + 测试过程

不存在账号密码泄漏问题。



* + 安全建议

开发过程中应尽量避免在日志中输出敏感信息，上线前应及时去除不必要的日志输出。

* + 1. 本地内存内容安全
  + 问题描述

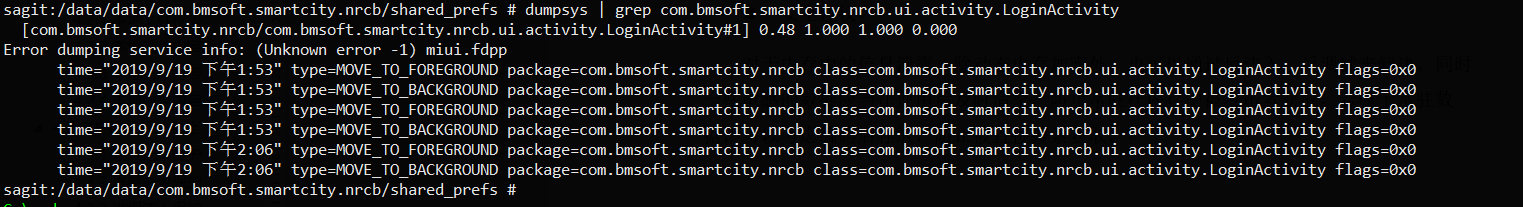
通过对客户端内存的访问，木马将有可能会得到保存在内存中的敏感信息（如登录密码，帐号等）。测试客户端内存中是否存在的敏感信息（卡号、明文密码等）。

* + 风险等级

无

* + 测试过程

未在内存中找到相关用户名密码等信息。



* + 安全建议

对于内存中的信息泄露，除动态内存加密外，也可以通过反注入、反调试来解决。同时一些敏感的数据应当在使用后及时清空，防止常驻在内存中被攻击者获取。若属于常驻数据，可以考虑编码后存放，防止被直接搜索获取。

* 1. 客户端运行时安全
     1. 输入记录保护
  + 问题描述

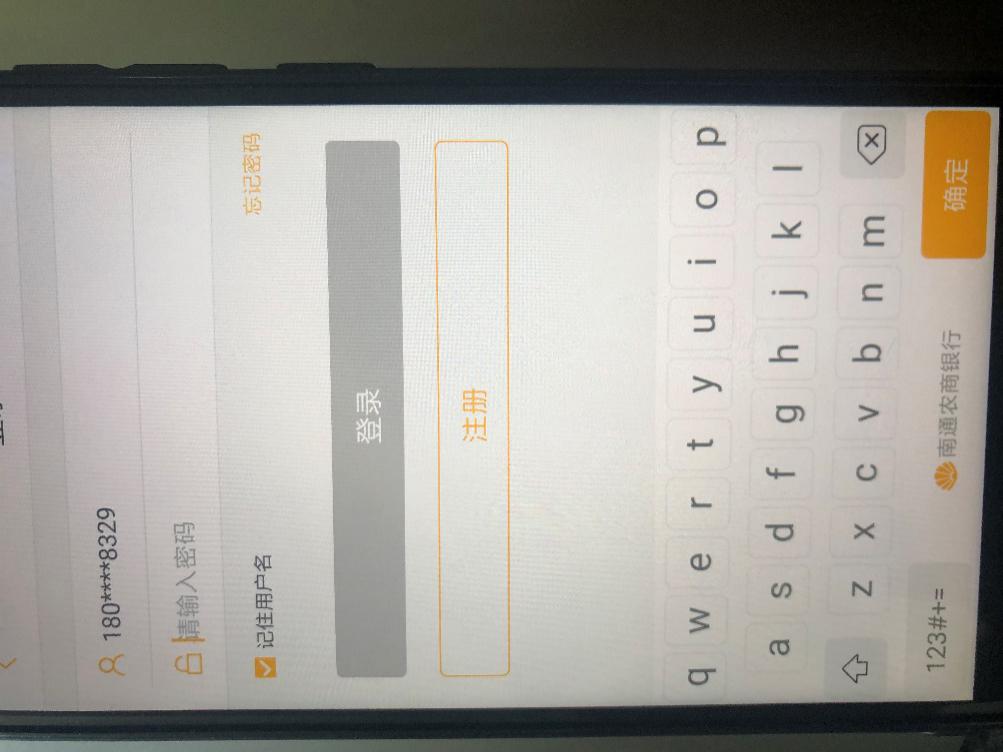
测试客户端程序在密码等输入框是否使用自定义软键盘。安卓应用中的输入框默认使用系统软键盘，手机安装木马后，木马可以通过替换系统软键盘，记录手机银行的密码。

* + 风险等级

无

* + 测试过程

设置了自定义键盘针对输入进行保护，无法监听输出。



* + 安全建议

尽量使用系统自定义的随机软键盘（而非系统输入法）来输入敏感信息。

* + 1. 屏幕录像保护【中风险】
  + 问题描述

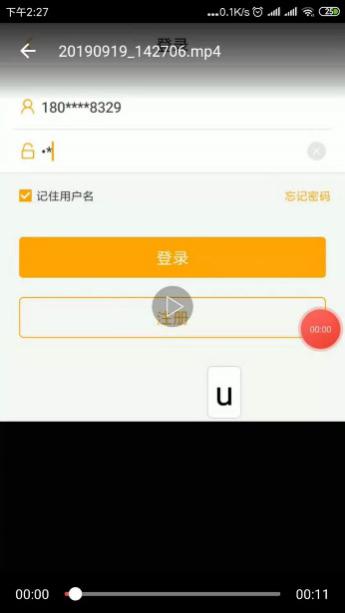
和输入记录类似，恶意程序可以对用户输入的敏感信息（主要是密码）进行窃听。

* + 风险等级

中风险

* + 测试过程

使用录像软件在录像的时候可以看到虽然没办法对键盘进行录制，但是在输入的时候可以看到输入的字符，所以有风险。



* + 安全建议

在敏感信息的输入过程尽量避免视觉反馈。

* + 1. 进程注入保护
  + 问题描述

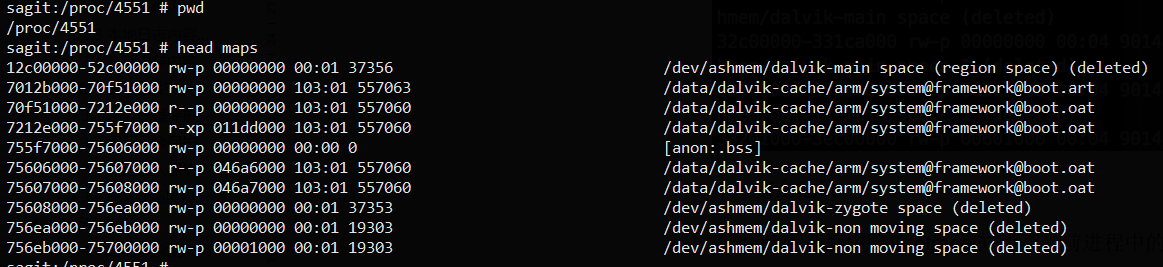
如果Android客户端没有对进程进行有效的保护，攻击者就可以向从Native层面向客户端进程远程加载任意.so链接库，从而侵入客户端进程的进程空间，以搜索、篡改敏感内存或干涉客户端的执行过程。

* + 风险等级

无

* + 测试过程

不存在进程注入问题。



* + 安全建议

在Native环境中防护ptrace并检查当前进程中的可执行模块等方式。另外部分的加固产品提供了相应的防护功能。

* + 1. ROOT环境检测【中风险】
  + 问题描述

对APP的运行环境进行检测。ROOT的设备是不安全环境，各种恶意程序都容易运行在ROOT环境之中，ROOT环境中运行则难以保证程序使用过程中数据的安全性。

* + 风险等级

中风险

* + 测试过程

应用在root环境下可以正常启动：



* + 安全建议

在检测到运行环境为ROOT环境时，对用户进行风险提示或者应用自动退出。

* 1. 安全策略
     1. 密码复杂度策略
  + 问题描述

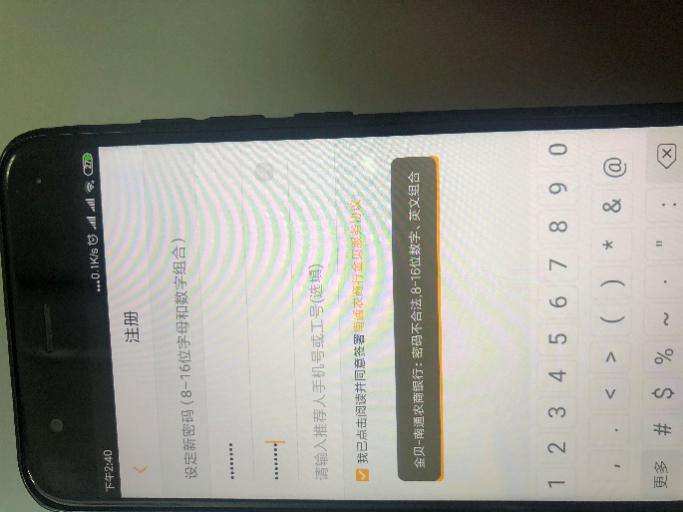
如果系统缺少密码复杂度策略，攻击者将有机会通过暴力猜测、撞库等方式获取一些安全意识淡薄的合法用户的认证凭据。

* + 风险等级

无

* + 测试过程

在注册的时候要求密码长度必须为8位以上密码，并且输入8位纯数字无法注册：



* + 安全建议

在注册账号、修改密码等需要设置密码的过程中，检测用户的密码强度，并在用户尝试设置弱口令时予以提示或阻止。

* + 1. 账户锁定策略
  + 问题描述

如果系统不存在认证失败锁定策略，攻击者将有机会对认证凭据进行暴力猜测。

* + 风险等级

无

* + 测试过程

存在相关锁定策略



* + 安全建议

以用户账号或者一些用户识别技术的识别码为参考记录错误尝试次数，若连续错误达到阈值，则锁定账号，禁止后续的认证。

* + 1. 单点登录限制策略【中风险】
  + 问题描述

如果系统允许同一个用户同时在多个会话中登录，那么用户就很难察觉到自身的账号已经被盗。

* + 风险等级

中风险

* + 测试过程

两个设备登录同一个账号，没有下线：



* + 安全建议

在数据库或服务器缓存中记录每个用户当前登录的SessionID，不允许同一个用户同时在多个Session中登录。

* + 1. 会话超时策略
  + 问题描述

对于认证会话不会超时的APP，如果用户在使用过程中将设备放置一旁并遗忘，攻击者将有可能通过物理接触的方式以用户身份进行操作。

* + 风险等级

无

* + 测试过程

在登录账号10分钟未操作之后提示重新登录。

* + 安全建议

当用户长时间不操作时，应当终止会话。

* + 1. UI敏感信息安全
  + 问题描述

用户的界面上可能泄漏了部分敏感信息，风险根据具体的泄露信息而定。敏感信息泄露可能会方便恶意程序窃取凭据，或者泄露一些原本不希望被用户看到的内容。

* + 风险等级

无风险

* + 测试过程

对一些信息进行了混淆，无法查看原来的信息，安全。



* + 安全建议

敏感信息在界面上显示时，在程序返回信息中使用 \* 替代部分敏感信息，客户端只需负责显示，在登录时提示信息应进行模糊化处理（统一回显“用户名或者密码错误”）。

* + 1. 界面切换保护【低风险】
  + 问题描述

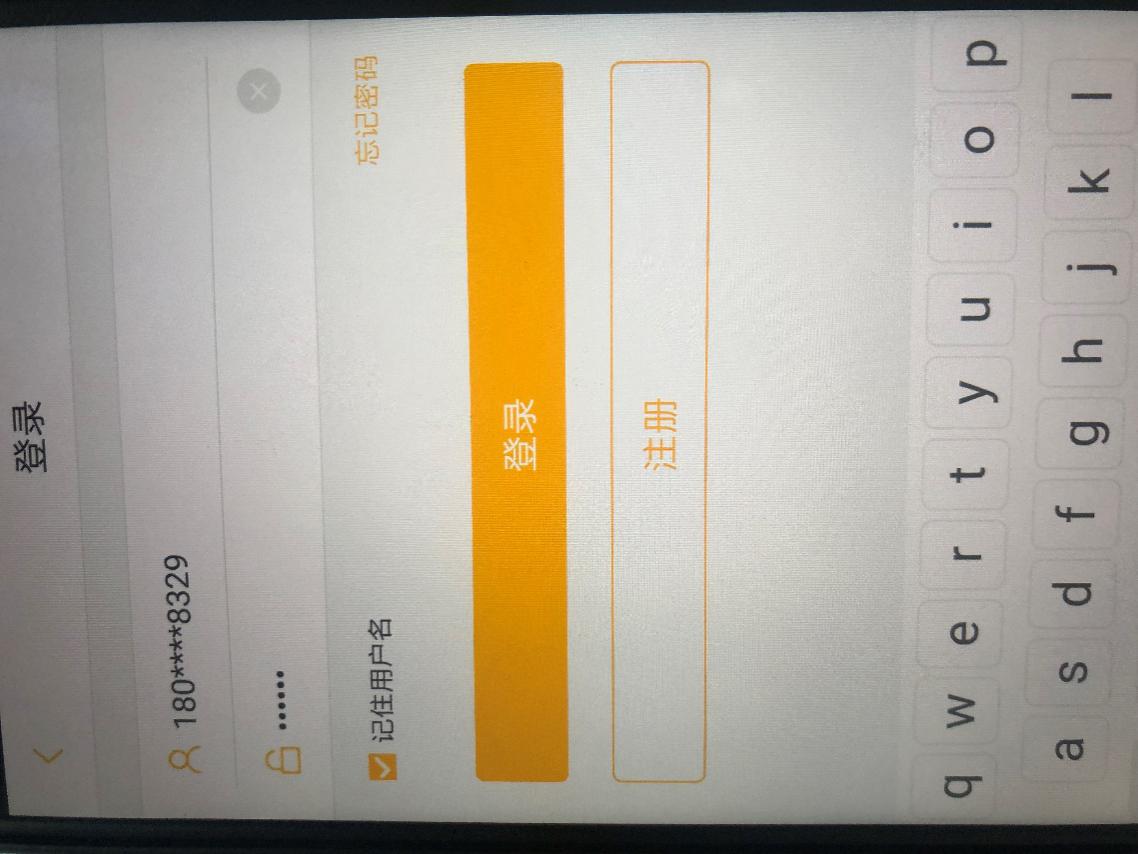
测试客户端程序在切换到其他应用时，已经填写的账号密码等敏感信息是否会清空，防止用户敏感信息泄露。如果切换前处于已登录状态，切换后一定时间内是否会自动退出当前会话。

* + 风险等级

低风险

* + 测试过程

切换之后，密码未清零。



* + 安全建议

增加切换界面登录名和密码清除机制，切出一段时间后自动退出机制。

* + 1. 安全退出
  + 问题描述

测试客户端退出时是否正常终止会话。

* + 风险等级

无

* + 测试过程

当前用户退出后，以前数据包无法正常返回数据。

* + 安全建议

当客户端退出时，向服务器发送终止会话的请求，立即注销当前会话。

* 1. 通信安全
     1. 通信协议有效性
  + 问题描述

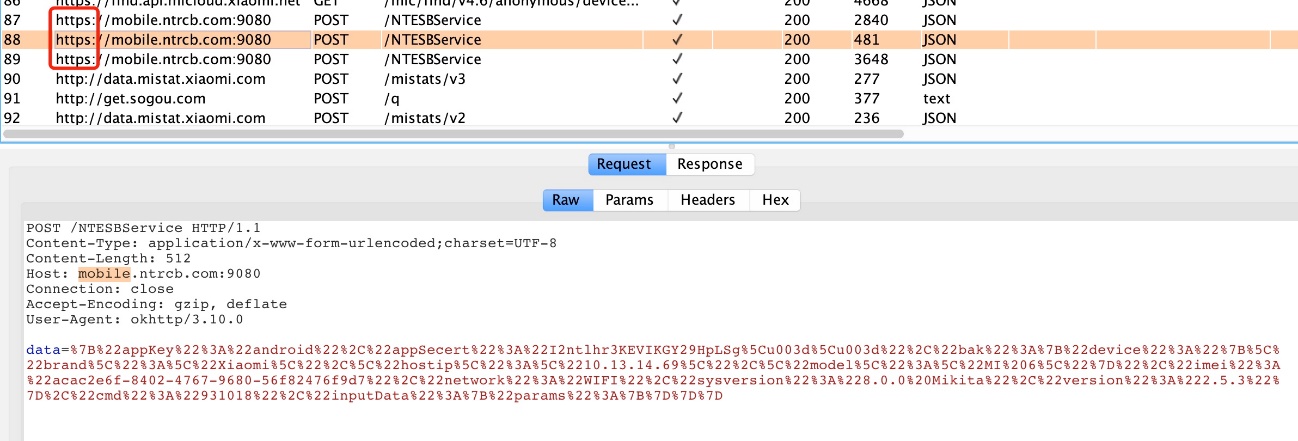
检测客户端与服务器的通信过程能否抵抗被动嗅探攻击、中间人攻击、重放攻击等常见形式的通信攻击。如果客户端与服务器之间的通信加密协议实现不当，攻击者将有机会对当前网络环境中其他合法用户的通信内容进行窃听甚至篡改。

* + 风险等级

无风险

* + 测试过程

本地数据包走HTTPS协议。



* + 安全建议

采用HTTPS或者SSL等证书相关的协议进行加密，且用SSLPinning的方式来绑定证书，避免攻击者或者中间人用其他合法证书进行欺骗 另外可以采用一些其他设计的加密算法。

* + 1. 通信数据加密
  + 问题描述

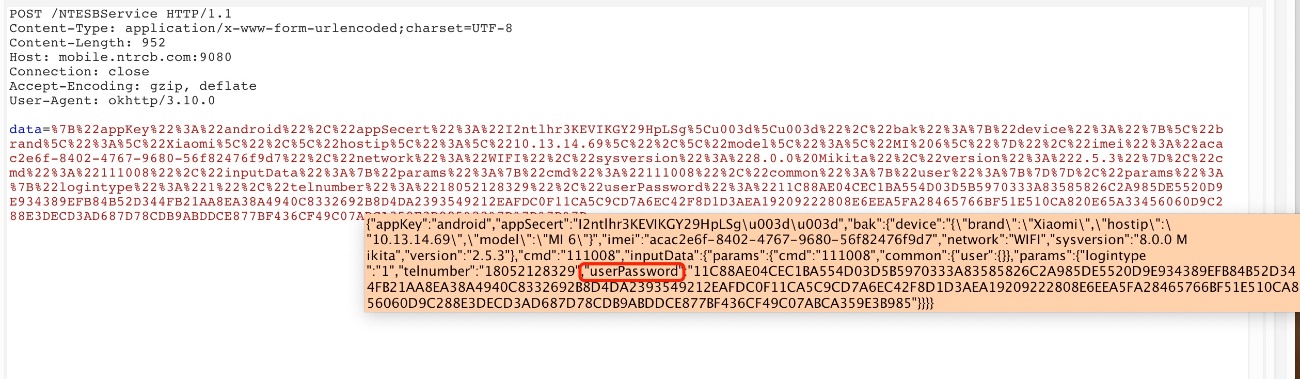
测试客户端程序提交数据给服务端时，密码等关键字段是否进行了加密和校验，防止恶意嗅探和修改用户数据包中的密码等敏感信息。

* + 风险等级

无风险

* + 测试过程

数据信息进行了加密处理。



* + 安全建议

对用户敏感数据（账号、卡号、密码）在报文中进行可靠的二次加密。

* + 1. 证书强校验【中风险】
  + 问题描述

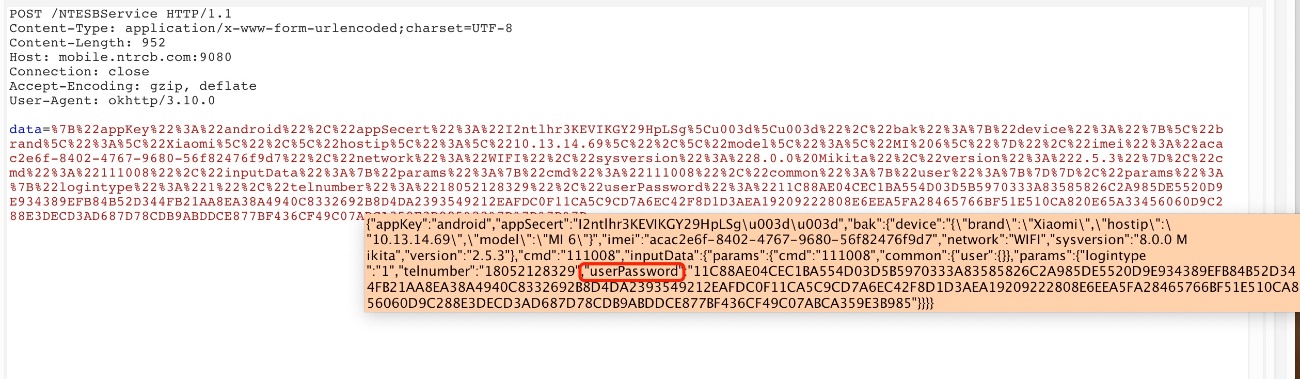
客户端可能存在忽略服务端证书校验的安全漏洞，对服务器没有校验或者没有在校验错误时候进行错误提示等。导致攻击者可通过伪造证书等手法进行攻击获取。

* + 风险等级

中风险

* + 测试过程

采用https协议，但是未对证书进行强校验，在安装burpsuite的伪证书之后可以抓包查看报文信息：



* + 安全建议

方案一:单向证书校验

方案二:双向校验方案（以tomcat为例）

只是需要在Tomcat配置中加入校验

<!-- root是根证书（二级根证书也可以）、server.p12是服务器证书，两者可以一样 -->

<Connector port="8443" protocol="org.apache.coyote.http11.Http11Protocol" SSLEnabled="true"

maxThreads="150" scheme="https" schemeecure="true"

keystoreType="PKCS12" keystoreFile="server.p12" keystorePass="1234"

truststoreType="JKS" truststoreFile="root.jks" truststorePass="password"

clientAuth="false" sslProtocol="TLS" />

* + 1. 软件升级缺陷
  + 问题描述

在进行版本检查的时候，通常由服务器返回对应的升级地址以方便用户访问升级，可能跳转到某个页面进行下载安装。

* + 风险等级

无

* + 测试过程

功能页面未发现软件升级按钮。

* + 安全建议

在服务端校验升级地址的合法性，禁止在客户端修改升级地址导致升级地址重新定向。

* + 1. 数据防重放【中风险】
  + 问题描述

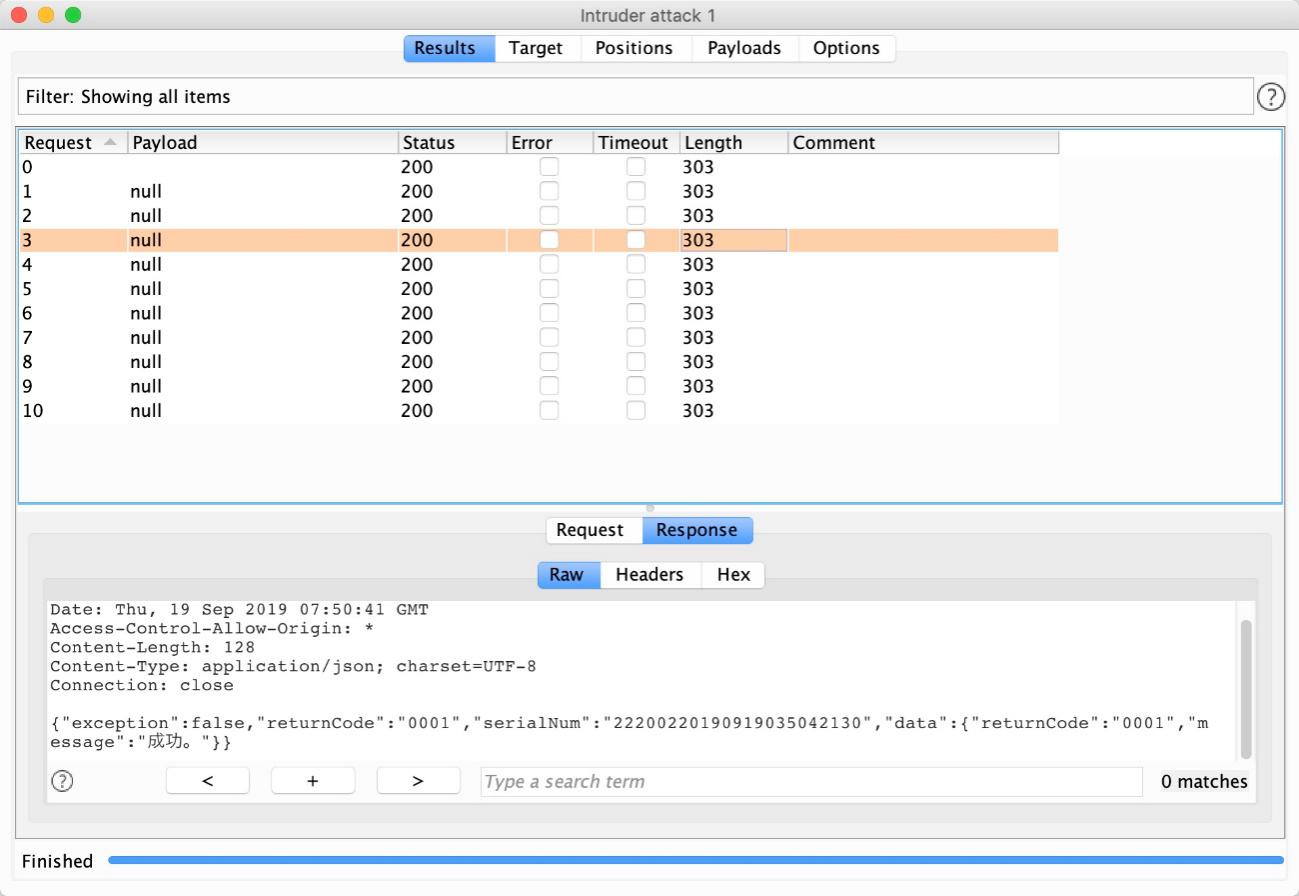
客户端与服务器之间的数据通信应具有防重放机制，如进行随机数校验，防止数据包重复提交攻击，造成业务错误。

* + 风险等级

中

* + 测试过程

数据包无校验手段，可重放



* + 安全建议

建议使用SSL协议，并在客户端对服务端的证书进行验证。如果自行实现加密协议，建议在客户端预先存储服务端公钥，在建立会话时随机生成对称加密密钥，用服务端公钥加密并发送给服务端，随后服务端用私钥解密后，正式开始进行通信。加密过程尽量避免使用CBC模式。

1. 安全等级及建议

通过此次针对南通农商行金贝所开展的安全渗透测试，发现安卓客户端存在的安全隐患主要包括组件导出安全、AllowBackup备份权限配置安全、本地文件权限配置、屏幕录像保护、ROOT环境监测、单点登录策略、界面切换保护、证书强校验、数据防重放等安全漏洞。

**经过整体安全风险分析，南通农商行金贝个人网银安卓版客户端的安全风险等级为：中**。

安恒信息作为一家致力于应用安全的专业产品和服务提供商，建议南通农商行金贝进行以下安全建设，以长期保证安卓客户端的安全。

* 针对安全测试结果协调开发团队或厂商进行有效的安全整改和修复。
* 定期进行专业的安全评估，及时掌握信息系统的安全状况。
* 完善安全管理制度体系，对信息系统的日常维护和使用进行规范。
* 建立一套完善有效的应急响应预案和流程，并定期进行应急演练，一旦发现发生任何异常状况可及时进行处理和恢复，有效避免系统业务中断带来损失。
* 定期对相关管理人员和技术人员进行安全培训，提高安全技术能力和实际操作能力。
* 对于0day等未被发现的问题，可采用APT高级可持续性攻击预警平台和网络安全保险等控制措施；对于不可预知的变化，可采取持续性态势感知监测技术、动态风险评估以及安全可信众测等措施加以防范。