**APP 安全测试**

**目录**

**一、安装包测试**

**1.1、关于反编译**

**1.2、关于签名**

**1.3、完整性校验**

**1.4、权限设置检查**

**二、敏感信息测试**

**三、软键盘劫持**

**四、账户安全**

**五、数据通信安全**

**六、组件安全测试**

**七、服务端接口测试**

**一、安装包测试**

**1.1、关于反编译**

目的是为了保护公司的知识产权和安全方面的考虑等，一些程序开发人员会在源码中硬编码一些敏感信息，如密码。而且若程序内部一些设计欠佳的逻辑，也可能隐含漏洞，一旦源码泄漏，安全隐患巨大。

为了避免这些问题，除了代码审核外，通常开发的做法是对代码进行混淆，混淆后源代码通过反软件生成的源代码是很难读懂的，测试中，我们可以直接使用反编译工具（dex2jar和jd-gui工具）查看源代码，判断是否进行了代码混淆，包括显而易见的敏感信息。

**1.2、关于签名**

这点IOS可以不用考虑，因为APP store都会校验。但Android没有此类权威检查，我们要在发布前校验一下签名使用的key是否正确，以防被恶意第三方应用覆盖安装等。可使用下列命令检查：

jarsigner -verify -verbose -certs apk包路径

若结果为“jar 已验证”，说明签名校验成功。

**1.3、完整性校验**

为确保安装包不会在测试完成到最终交付过程中因为知足者趾问题发生文件损坏，需要对安装包进行完整性校验，通常做法是检查文件的md5值，而且一般可以通过自动化做校验。

**1.4、权限设置检查**

一般用户对自己的隐私问题十 分敏感，因此，我们需要对APP申请某些特定权限的必要性进行检查，如访问通讯录等。对于没有必要的权限，一般都建议开发 直接支除。

**Android：**直接检查manifest文件来读取应用所需要的全部权限，并结合需求进行校验此权限是否为必须的。manifest文件的修改也需要关注，在增加新权限前需要进行评估。

**IOS：**没有类似manifest文件来查看，IOS的用户权限只有在用户使用APP到了需要使用的权限时，系统才会弹出提示框，提示用户当前APP需要访问照片、联系人列表等组件。我们可以扫描代码来查看项目工程中有哪些权限设置。通过搜索关键类名，如通讯录一般需要访问ABAddressBookRef，照片是UIImagePickerController等。如果是纯黑盒测试，则必须覆盖到所有代码路径才能保证没有遗漏，也可使用代码覆盖率测试判断是否覆盖。

**二、敏感信息测试**

数据库是否存储敏感信息，某些应用会把cookie类数据保存在数据库中，一旦此数据被他人获取，可能造成用户账户被盗用等严重问题，测试中在跑完一个包含数据库操作的测试用例后，我们可以直接查看数据库里的数据，观察是否有敏感信息存储在内。一般来说这些敏感信息需要用户进行注销操作后删除。如果是cookie类数据，建议设置合理的过期时间。

日志是否存在敏感信息，一般开发在写程序的过程中会加入日志帮助高度，所有可能会写入一些敏感信息，通常APP的发布版不会使用日志，但也不排除特殊情况。

配置文件是否存在敏感信息，与日志类似，我们需要检查配置文件中是否包含敏感信息。

**三、软键盘劫持**

如果用户安装了第三方键盘，可能存在劫持情况，对此，我们在一些特别敏感的输入地方可以做检查，例如金融类APP登录界面的用户名密码输入框等，看是否支持第三方输入法，一般建议使用应用内的软键盘。

**四、账户安全**

4.1、密码是否明文存储在后台数据库，在评审和测试中需要关注密码的存储。

4.2、密码传输是否加密，测试中我们需要查看密码是否被 明文传输，如果是HTTP接口，我们可以使用FIddler等工具直接查看。

4.3、账户锁定策略。对于 用户输入错误密码次数过多的情况，是否会将账户临时锁定，避免被暴力破解，

4.4、同时会话情况。一些应用对同时会话会有通知功能，这样至少可以让用户知识他的账户可能已经被泄漏了。在一定程度上能免提升用户体验。

4.5、注销机制。在客户端注销后，我们需要验证任何的来自该用户的，需要身份验证的接口调用都不能成功。

**五、数据通信安全**

5.1、关键数据是否散列或加密。密码在传输中必须是加密的，其他敏感信息传输前也需要进行散列或者加加密，以免被中间节点获取并恶意利用。

5.2、关键连接是否使用安全通信，例如HTTPS。在获知接口设计后我们需要评估是否其中内容包含敏感信息，如果未使用安全通信，需要知会开发修改。

5.3、是否对数字证书合法性进行验证。即便使用了安全通信，例如HTTPS，我们也需要在客户端代码中对服务端证书进行合法性校验。测试中可以使用Fiddler工具模拟中间人攻击方法。如果客户端对于Fiddler证书没有校验而能正常调用，则存在安全隐患。

5.4、是否校验数据合法性。在一些情况下，我们需要有方法来确保服务端下发的明文数据不被篡改。通常开发侧的实现方式是**对数据进行数字签名并在客户端进行校验**。我们可以模拟后台返回进行相关的测试工作。此外，对于其他一些客户端未进行数据校验的接口，我们也需要有意识地思考如果不进行校验是否会产生问题，并通过模拟后台返回验证。

**六、组件安全测试**

**这里主要是指Android平台各个组件是否能被 外部应用恶意调用从而带来一些安全问题。包括Activity、Service、ContentProvider、Broadcast等等。采用的测试方法是通过使用drozer工具结合查看代码的方式，具体使用方法可查看官方文档。**

**七、服务端接口测试**

主要关注服务端接口是否存在以下问题

7.1、SQL注入

7.2、XSS跨站脚本攻击

7.3、CSRF跨站请求伪造

7.4、越权访问

除了上述服务端问题外，我们还需要结合实际的需求，设计和代码，分析是否需求或设计本身就会带来安全问题。

举个例子：如一个购物的应用，下单地的流程包含两个接口，**接口A**返回订单详情，其中包括订单号码和金额总价。调用接口A后用户在客户端看到一个订单页面。用户点击提交后调用**接口B**，客户端传给接口B的参数为接口A返回的订单号码和金额总价，接口B的后台根据传给接口B的金额总价从用户账户中扣款，扣款成功后即根据订单号码发货。

这一设计有什么问题呢？那就是接口B完全信任了客户端传入的金额总价而未做校验。恶意用户可以直接调用接口B，传入伪造的金额和真实订单号，这样就能以便宜的价格购物。

**附录**

**1.软件权限**

1）扣费风险：包括短信、拨打电话、连接网络等。

2）隐私泄露风险：包括访问手机信息、访问联系人信息等。

3）对App的输入有效性校验、认证、授权、数据加密等方面进行检测

4）限制/允许使用手机功能接入互联网

5）限制/允许使用手机发送接收信息功能

6）限制或使用本地连接

7）限制/允许使用手机拍照或录音

8）限制/允许使用手机读取用户数据

9）限制/允许使用手机写入用户数据

10）限制/允许应用程序来注册自动启动应用程序

**2.数据安全性**

1）当将密码或其它的敏感数据输入到应用程序时，其不会被存储在设备中，同时密码也不会被解码。

2）输入的密码将不以明文形式进行显示。

3）密码、信用卡明细或其他的敏感数据将不被存储在它们预输入的位置上。

4）不同的应用程序的个人身份证或密码长度必须至少在4-8个数字长度之间。

5）当应用程序处理信用卡明细或其它的敏感数据时，不以明文形式将数据写到其他单独的文件或者临时文件中。以防止应用程序异常终止而又没有删除它的临时文件，文件可能遭受入侵者的袭击，然后读取这些数据信息。

6）党建敏感数据输入到应用程序时，其不会被存储在设备中。

7）应用程序应考虑或者虚拟机器产生的用户提示信息或安全警告

8）应用程序不能忽略系统或者虚拟机器产生的用户提示信息或安全警告，更不能在安全警告显示前，利用显示误导信息欺骗用户，应用程序不应该模拟进行安全警告误导用户。

9）在数据删除之前，应用程序应当通知用户或者应用程序提供一个“取消”命令的操作。

10）应用程序应当能够处理当不允许应用软件连接到个人信息管理的情况。

11）当进行读或写用户信息操作时，应用程序将会向用户发送一个操作错误的提示信息。

12）在没有用户明确许可的前提下不损坏删除个人信息管理应用程序中的任何内容。

13）如果数据库中重要的数据正要被重写，应及时告知用户。

14）能合理的处理出现的错误。

15）意外情况下应提示用户。

**3.通讯安全性**

1）在运行软件过程中，如果有来电、SMS、蓝牙等通讯或充电时，是否能暂停程序，优先处理通信，并在处理完毕后能正常恢复软件，继续其原来的功能。

2）当创立连接时，应用程序能够处理因为网络连接中断，进而告诉用户连接中断的情况。

3）应能处理通讯延时或中断。

4）应用程序将保持工作到通讯超时，进而给用户一个错误信息指示有链接错误。

5）应能处理网络异常和及时将异常情况通报用户。

6）应用程序关闭网络连接不再使用时应及时关闭，断开。

**4.人机接口安全测试**

1）返回菜单应总保持可用。

2）命令有优先权顺序。

3）声音的设置不影响使用程序的功能。

4）应用程序必须能够处理不可预知的用户操作，例如错误的操作和同时按下多个键。