IOS客户端通用测试

项目范围

适用于IOSAPP安全测试项目基本测试的实施

准备工作

一台电脑、一部苹果越狱机 (本次的测试机:ihone 5sversion:9.0-9.1完美越狱)系统(Mac、windows 7虚拟机)

Ps:

完美越狱简单地说就是越狱很彻底，破解了操作系统的读写权限，完美越狱完成以后可以自由使用，开关机、下载等。

而不完美越狱，则表示iPhone或iPad不能像越狱之前一样随时可以关机

IOS客户端数据安全

1.1 日志信息安全

测试项描述

开发者习惯使用日志辅助程序调试，在日志中打印信息，则可能泄露访问链接、收发的 数据包甚至身份凭证等内容。

0x01使用xcode

Windows->Devices

0x02使用iTools

工具箱->实时日志

风险评级

视泄露数据的情况而定。

高风险——泄露用户身份凭证(如token、cookie)、用户或其他用户隐私信息

中风险——泄露关键函数(加解密、校验函数)调用和逻辑信息或者部分隐私信息

低风险——非上述两种信息之外的内容泄露评定为这个等级

修复方法

将所有的NSLoq调用去掉再重新编译打包程序

IOS客户端数据安全

1.2 程序数据存储安全

测试项描述

检设备中程序对应的数据文件夹是否包含敏感数据。由于设备可以进行越狱，且越狱之后数据不会丢失，则导致本地存储敏感信息将造成较大风险。

SandBox文件存储结构

SubDirectory	Description
AppName.app	存储app执行文件和静态资源文件，该文件夹为只读
Documents	App的配置文件等，该文件夹的内容会被同步到backup文件中
Library	Application support files
Library/Preference	应用程序特定的首选项（plist文件）
Library/Caches	缓存数据(cache file and cookie file)，该文件夹内的内容不会被同步
tmp	在连续启动应用程序时不需要保留的临时文件

测试点有哪些？

测试步骤

1.在设备上安装程序

2.运行程序，进行较大量的操作用来填充数据

3进入设备文件系统，从数据文件夹中获取所有文件:

在越狱设备中，本测试项需要AppSync(hook内核，绕过程序的签名验证)和A FC2(越狱文件系统)

在iOS7中，相关的程序和数据内容在目录:/var/mobile/Application/<UUID>

在iOS8中，相关的程序内容在自录:/var/mobile/Containers/Bundle/Application/<UUID

在iOS8中，相关的数据内容在目录/var/mobile/Containers/Data/Application/<UUID>

4.检查 plist 文件、slite 文件、binarycookie 文件

5.使用keychaindumper 获取 keychaindumper中的信息

6.除了上面的文件之外，还有其他存储的文件(如:pdf 文件)

文件存储目录

1.plist:全名Property List，属性列表文件，它是一种用来存储串行化后的对象的文件。

文件是xml格式的。般都是存储在各种目录下，视情况而定

2.Salite:一般在document下，根据开发者的情况而定

3.Binarycookie:一般在/private/var/mobile/[app directory]/Library/Cookies/

利用BinaryCookieReaderpy工具进行查看

风险评级

视信息泄露的严重性而定。

高风险——大量用户信息或者泄露了登录密码、支付密码等高敏感度信息

中风险—— 未进行类名混淆以及逻辑混淆

低风险——进行了大部分类名混淆，未进行逻辑混淆

修复建议:

针对UserDefault，建议加密存储，而且建议使用用户输入数据作为加密秘钥(但是可能会影响到自动登录):

1.针对salite，使用SQLCipher进行加密，秘钥使用用户输入数据或者使用服务器相关内容;

2.针对binarycookie，只能清除

```
//Delete previous cookies
NSHTTPCookieStorage *cookieStorage = [NSHTTPCookieStorage sharedHTTPCookieStorage];
for (NSHTTPCookie *each in [[[cookieStorage cookiesForURL:YOUR_URL] copy] autorelease]) {
    [cookieStorage deleteCookie:each];
}
```

3.针对keychain，加密存储，密码依然使用用户存输入的数据;另外因为keychain在程序删除仍然存在，且越狱机子上用kevchain 保存完全没有安全可言，不推荐使用keychain 存储高敏感度数据;

4.对于其他文件，进行独立加密，并解密读取即可;

谢谢观赏