Computer Science and Application 计算机科学与应用, 2015, 5(11), 421-427
Published Online November 2015 in Hans. http://www.hanspub.org/journal/csa
http://dx.doi.org/10.12677/csa.2015.511053

Hans汉斯

Detection and Prevention of the Phishing Risk of Android Application

Zhenpeng Huang, Shaozhang Niu, Wen Zhang

Beijing University of Posts and Telecommunications, Beijing
Email: huangzhenpeng@gmail.com, szniu@bupt.edu.cn, 14311648@qq.com

Received: Nov. 6th, 2015; accepted: Nov. 20th, 2015; published: Nov. 27th, 2015

Abstract

Android is the most popular mobile device's intelligent operating system. With the rapid development of mobile devices, mobile applications are becoming more and more important in people's life. However, due to the developers' inadequate consideration or attention for mobile device security, a lot of applications have the security vulnerability problems. Using fishing hijacking vulnerabilities can steal the user's important information in the case that users are unaware, which has caused a huge threat to the user's privacy and property security. In this paper, through the study of activity, we design and implement a phishing detection system of Android application. Using the system to detect the 500 common applications from the MM shopping market, the results show that the phishing hijacking vulnerabilities of Android application exist widely. At last, the paper gives the feasible preventive suggestions for the risk of the fishing.

Keywords

Android, Fishing Hijacking, Security

Android应用钓鱼劫持风险的检测与防范

黄振鹏，牛少彰，张　文

北京邮电大学，北京
Email: huangzhenpeng@gmail.com, szniu@bupt.edu.cn, 14311648@qq.com

文章引用: 黄振鹏, 牛少彰, 张文. Android 应用钓鱼劫持风险的检测与防范[J]. 计算机科学与应用, 2015, 5(11): 421-427. http://dx.doi.org/10.12677/csa.2015.511053

收稿日期：2015年11月6日；录用日期：2015年11月20日；发布日期：2015年11月27日

摘　要

Android是当前最流行的移动设备上的智能操作系统。随着移动设备的蓬勃发展，移动端应用在人们生活中也越来越重要。但是由于开发者考虑不足或者对移动端的安全的不够重视，许多的移动应用存在安全漏洞。利用钓鱼劫持漏洞可以在用户未察觉的情况下窃取用户的重要信息，这对用户的信息安全和财产安全造成了巨大的威胁。本文通过对Android Activity组件进行研究，设计并实现了对Android应用的钓鱼劫持漏洞检测系统。利用该系统对从MM商场下载的500个常见应用进行检测，结果表明钓鱼劫持漏洞在Android应用中广泛存在。本文最后给出了针对钓鱼劫持风险可行的防范建议。

关键词

Android，钓鱼劫持，安全

1. 引言

Android 系统是当前最为流行的移动端操作系统。根据 Google 官方数据当前 Android 设备日激活量达到 150 万部，Android 设备的总计出货量在 2014 年已经超过 15 亿部，见图 1。Google Play 的 Android 应用下载量也已突破 500 亿次，可以说 Android 应用已经深入人们生活的方方面面。随着移动应用的普及，对于移动应用的攻击也越来越常见。

当前对于钓鱼劫持大多数开发者和研究者都不是足够重视，相关研究也不多。但是随着移动应用在人们生活中的应用越来普遍，Android 应用涉及的个人信息越来越多也越来也重要，Android 应用的钓鱼劫持所带来的安全威胁也越来越大，所以对 Android 应用的钓鱼劫持漏洞的研究有重要意义[1]。

本篇论文致力于研究 Android 应用中钓鱼劫持漏洞，对导致钓鱼劫持漏洞的原因进行分析，并对钓鱼劫持漏洞的存在情况进行抽样检测，最后为应用开发者提供一套可行的钓鱼劫持防范方案。

2. Activity 及其生命周期

Activity 是 Android 应用的基本组件之一，是应用和用户进行交互的接口，通常以覆盖整个屏幕的窗

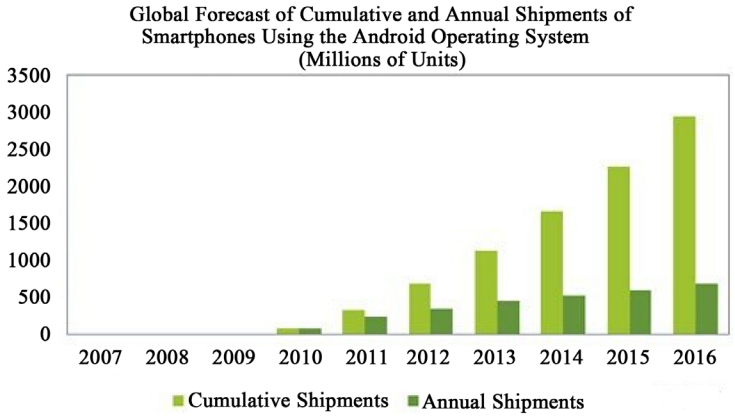

Figure 1. Statistics and forecast of global Android equipment shipments
图 1. 全球 Android 设备出货量的统计和预测

口的形式展示给用户[2]。我们平常对应用进行的操作都是在 Activity 上进行的，比如拨号、翻看照片、发送 Email、收发短信等。大多数应用都由多个 Activity 组成，它们之间是松耦合关系[3]。一般一个应用拥有一个主 Activity，就是该应用启动后展示给用户的第一个 Activity，然后由这个 Activity 自动或由用户选择启动其他的 Activity, 通过一个个的 Activity，应用完成于用户的交互。

每个 Activity 的生命周期都有 4 个状态，分别是活动状态、暂停状态、停止状态、重启状态，见图 2。在活动状态下 Activity 对用户完全可见并且可与用户进行交互[4]。

Android 利用任务栈的形式对运行的 Activity 组件进行管理。当前运行的 Activity 位于栈顶位置，在新的 Activity 启动时，新的 Activity 将被放在栈顶，新的 Activity 进入活动状态。而原运行的 Activity 将在栈中处于其下的位置，此时原 Activity 将进入停止状态。任务栈遵循先进先出的原则，栈中的 Activity 组件的顺序是固定的，不能被重新排序，只能被压栈和弹栈。使用 Back 键退出当前运行的 Activity 时，该 Activity 将被从栈顶位置弹出，位于栈顶之下的 Activity 将处于栈顶，重新处于运行状态。

3. 钓鱼劫持原理

应用可以通过 Activity 管理器获取当前的任务栈信息，取出栈顶元素就可获知当前运行的 Activity 信息[5]。钓鱼应用可以利用这一点，监控当前栈顶的 Activity 信息，从而判断哪个 Activity 将被展示。

若登录界面是一个独立的 Activity，恶意应用通过判断栈顶元素的方式获知应用的登录界面将何时被展示，从而及时弹出伪造的登录界面，替换掉原有的 Activity。且由于手机屏幕的展示空间有限，系统不会主动显示当前实际运行的是那个应用，用户无法察觉当前展示的登录界面是伪造的界面[6]。

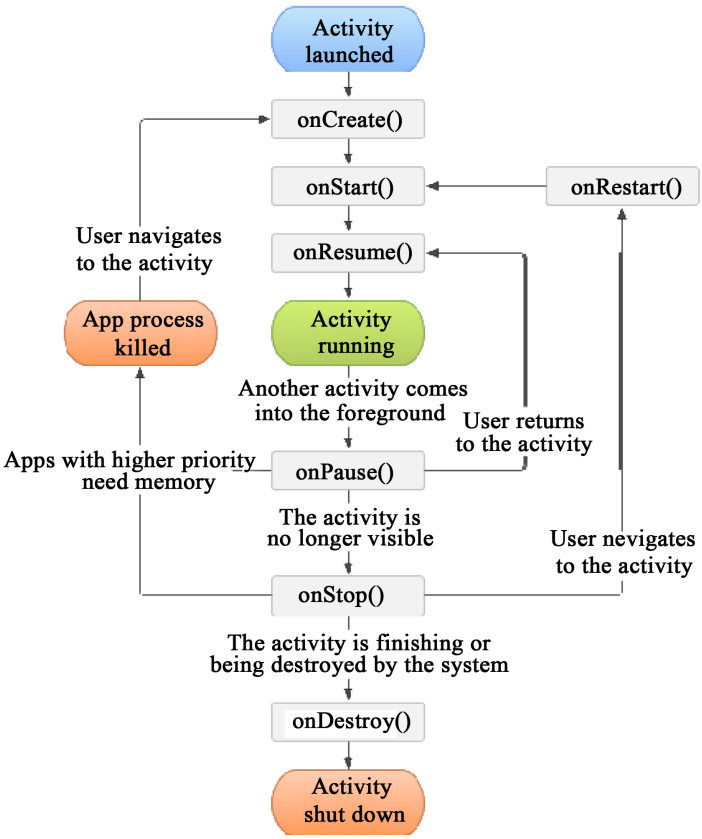

Figure 2. The lifecycle of activity
图 2. Activity 的生命周期

当用户输入了帐号密码等信息后，钓鱼应用就可以通过网络数据或发送短信等形式将获取的数据发送至指定服务器或号码。一些钓鱼应用的伪造界面做的比较逼真，很多用户在账号密码等信息被窃取了都未能察觉，从而给用户的个人隐私和财产安全带来了极大的危害。

4. Android 应用的钓鱼劫持风险检测

4.1. 钓鱼攻击检测系统的设计

恶意应用为了完成钓鱼攻击，需要多个条件，分别是：被测应用需要有敏感信息录入界面；恶意应用可以获知该页面何时被调用；并且需要钓鱼攻击时用户难以发现[7]。因此，对于钓鱼劫持漏洞的检测，需要检测一下内容：

1) 有无需要输入敏感信息的界面

钓鱼程序的攻击目标一般是能有价值的信息，比如用户的账号、密码、手机号码、银行卡号等个人敏感信息，本研究的重点也是关注用户敏感信息的保护，所以首要关注需要用户输入敏感信息的界面。

2) 该界面的运行是否可被其他应用侦测到

钓鱼程序在进行钓鱼攻击前，需要判断钓鱼界面的弹出时机，若是钓鱼程序无法判断需要用户输入敏感数据的界面的弹出时刻，则钓鱼程序无法确定何时攻击或选择错误的时刻进行钓鱼攻击，从而难以钓鱼成功。对应用界面的切换可通过对 Activity 栈进行监控，若是被测应用使用单独 Activity 进行敏感信息的输入，则极易被钓鱼程序检测到。

3) 该应用是在该应用运行时的跳至其他应用的可疑行为对用户进行提示

应用在本该弹出敏感信息录入界面时突然切至其他应用，则有可能被钓鱼劫持。若应用此时即使给用户以提示，则可极大概率降低用户被钓鱼劫持的风险。

所以我们的检测主要包含三部分：1) 查找敏感信息录入页面；2) 判断界面是否为独立的界面，即是否由单独的 activity 组成；3) 判断界面异常切换时是否给用户以提示。钓鱼攻击检测系统的检测流程，见图 3。

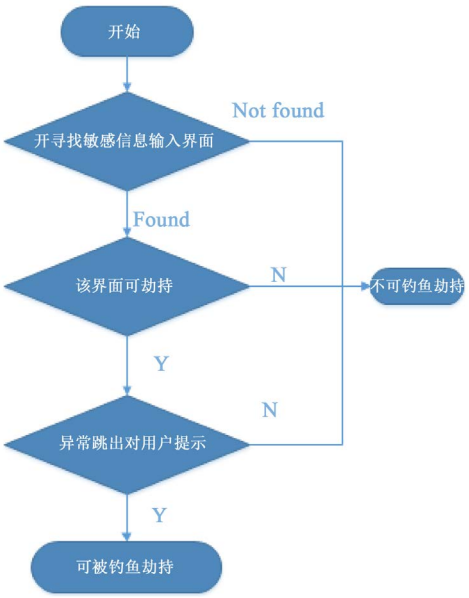

Figure 3. The process of phishing detection
图 3. 钓鱼劫持检测流程图

4.2. 钓鱼攻击检测系统的实现

为了检测应用是否可被钓鱼攻击，首先找到该应用需要输入敏感信息的 Activity，该类 Activity 一般是登录界面、支付界面、信息录入界面等，一般录入信息为用户名、密码、银行卡号、信用卡的安全码、电话号码、地址等信息。

找到相关的界面后，需要获取该 Activity 的相关信息，包括包名、类名等信息，然后使用 Android 自带的 Log 系统，对获取的 Activity 的信息进行输出。通过获取 Activity 栈的信息来获取相关 Activity 的信息，为了更准确的获取相关信息，一般动态运行被测应用并进入敏感信息录入界面，此时获取 Activity 栈的栈元素信息并 Log 输出[8]。该部分代码如下：

```
ActivityManager AM = ((ActivityManager) getSystemService (Context.ACTIVITY_SERVICE));
List<RunningTaskInfo> RTIs = AM.getRunningTasks(1);
ComponentNamecn = RTIs.get(0).topActivity;
Log.e("HijackService_TimerTask", cn.flattenToString());
```

对获取的 Activity 信息进行判断，看起是否为一个单独的可被监测到的 Activity。若是可利用的 Activity，则可以利用上一步获取的 Activity 的信息，构造具体钓鱼攻击程序。利用钓鱼程序对栈顶 Activity 进行监控，当钓鱼攻击程序检测到栈顶 Activity 为上一步获取的 Activity 时，弹出伪造页面[9]。

```
    if (((ActivityManager) getSystemService( Context.
ACTIVITY_SERVICE )).getRunningTasks( 1 ).get( 0 ).topActivity. getClassName().contains( acStr ) ) {
                        handler.post( new Runnable () {
                            @Override
                            public void run( ) {
                            startMyActivity( );/*启动伪造的 Activity*/
                                    }
                        });
    }
```

运行被测应用程序，当再次进入敏感信息输入界面时，查看弹出的界面是原界面还是钓鱼程序伪造的界面。若是弹出了伪造的界面，再查看被测程序是否给出了可能被钓鱼劫持的提示，如果未给出提示，则可认为该应用存在被钓鱼劫持的风险。

4.3. 检测结果

为检测可被钓鱼劫持的软件所占的比例，本研究对从中国移动应用商场随机下载了的 500 个 Android 应用程序进行检测，发现存在钓鱼劫持风险的应用为 431 款。在这 500 款应用中存在不存在输入敏感信息的应用为 442 款，存在敏感信息但不可获取相应 Activity 信息的为 11 款，从输入敏感信息页面跳出应用给出提示的应用为 0 款，具体统计信息见图 4。

通过数据我们可知由于开发者对钓鱼劫持风险缺乏重视，钓鱼劫持在 Android 应用中是广泛存在的，而这对用户的信息安全带来了极大的威胁，故对于钓鱼劫持采取何时的防范措施是必要的。

5. Android 应用钓鱼劫持的防范

为了应对的前述检测到的风险，通过前面的研究，本文认为开发者可以从两个方面防范钓鱼劫持攻击：在遭受疑似攻击时及时给用户提示或使应用的敏感信息录入界面不可被探测。

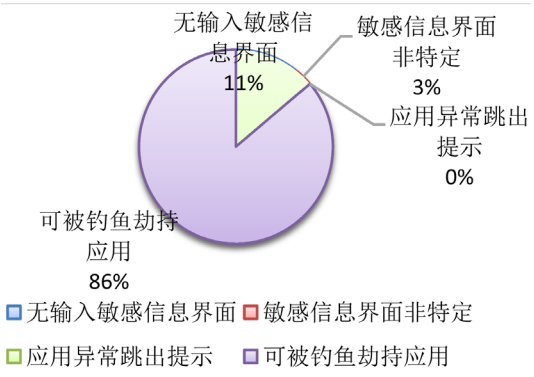

Figure 4. Ratio of fishing detection results
图 4. 钓鱼检测结果比例图示

5.1. 应用异常跳出时给出钓鱼劫持风险提示

应用程序在进入敏感信息输入界面时，在后台启动一个 Service，该 Service 实时监测当前设备的栈顶 Activity 的信息，判断系统当前运行的程序，一旦发现应用切换(可能被劫持)，给予用户提示以防范钓鱼程序的欺诈。获取栈顶 Activity(如下图)，当涉及敏感 Activity(登录、交易等)切换时，判断当前是否仍留在原程序，若不是则通过 Toast 给予用户提示。

5.2. 使用 HTML5 架构或 Android + HTML5 混合开发

将登陆、支付等需要输入敏感信息的页面使用 html5 来实现，而不是将这些界面单独放在某一个 Activity 内，将使钓鱼程序无法通过读取 Activity 栈的方式来判断何时弹出伪造界面，从而降低被钓鱼劫持的风险。

6. 结论

本研究对 Android 应用钓鱼劫持漏洞的原理进行了分析，并设计了针对钓鱼劫持漏洞的检测方法。通过对 500 个应用的检测，发现在常见应用中钓鱼劫持漏洞是普遍存在的。希望本文能够促进移动应用开发者对应用安全的认识。

致　谢

本文的研究得到国家自然科学基金的资助，为研究的顺利进行提供了资金保证，在此对国家自然科学基金表示衷心的感谢。

基金项目

国家自然科学基金项目(61070207, 61370195)。

参考文献 (References)

[1] 符易阳, 周丹平. Android 安全机制分析[C]//中国计算机学会. 第 26 次全国计算机安全学术交流会论文集: 2011 年第 9 期. 北京: 信息网络安全杂志社, 2011: 23-25.

[2] 董晓刚. 浅析 Android 系统的四大基本组件[J]. 中国电子商务, 2013(1): 39.

[3] 傅建明, 李鹏伟, 易乔, 黄诗勇. Android 组件间通信的安全缺陷静态检测方法[J]. 华中科技大学学报(自然科学版), 2013, 41(z2): 259-264.

[4] 洪智勇, 张宁. 移动终端软件开发技术课程教学改革研究与实践[J]. 软件工程师, 2014, 17(6): 46-48.

[5] 李鸥. Android 安全机制分析及双用户安全保护模型的设计[J]. 保密科学技术, 2014(3): 42-46.

[6] 钱宇虹. 多线程环境下如何正确使用 Java 集合类[J]. 软件工程师, 2012(10): 45-48.

[7] 沈才樑, 唐科萍, 俞立峰, 樊甫伟. Android 权限提升漏洞攻击的检测[J]. 电信科学, 2012(5): 115-119.

[8] Shabtai, A., Fledek, Y. and Kanonov, U. (2010) Google Android: A Comprehensive Security Assessment. *IEEE Security and Privacy*, **8**, 35-44. http://dx.doi.org/10.1109/MSP.2010.2

[9] Shabtai, A., Kanonov, U. and Elovici, Y. (2009) Detection, Alert and Response to Malicious Behavior in Mobile Devices: Knowledge-Based Approach. *Proceedings of* 12*th International Symposium on Recent Advances in Intrusion Detection* (*RAID*), Saint-Malo, 23-25 September 2009, 357-358.