1) PackageManagerService 服务.在应用安装时,该服务负责解析应用压缩包.在我们的系统设计中,我们在应用安装解析过程中获得应用的 sharedUserId 属性,并获得活动组件中的 taskaffinity 值.这些属性用来标记应用服务、事件及应用的“亲密性”.应用自身的无障碍事件的处理过程取决于标记.应用间的无障碍事件处理则由“亲密性”决定.
2) View 视图.一个应用启动活动视图、生成视图对象是通过 View 类完成的.在该类中,系统完成了对事件的初始化及分发操作.在我们的系统中,我们设计在初始化无障碍事件时,利用提取的一些属性信息及包名和 UID 值生成标签并对无障碍事件进行标记.
3) AccessibilityManagerService 服务.该无障碍辅助性服务管理服务负责调用哪个无障碍辅助性服务、处理哪些事件.在我们的系统中,为了保证事件得到正当处理、无障碍辅助性服务不被滥用,我们在启动无障碍辅助性服务处理事件时,首先获得服务的“亲密性”属性并进行检查.该步骤的检查包含两个方面:检查无障碍辅助性服务与待处理事件的标签是否匹配或具有亲密性;检查无障碍辅助性服务与将要调用的其他服务的“亲密性”来决定是否有权访问其他服务的处理结果.若满足,则继续处理;反之则拒绝处理.

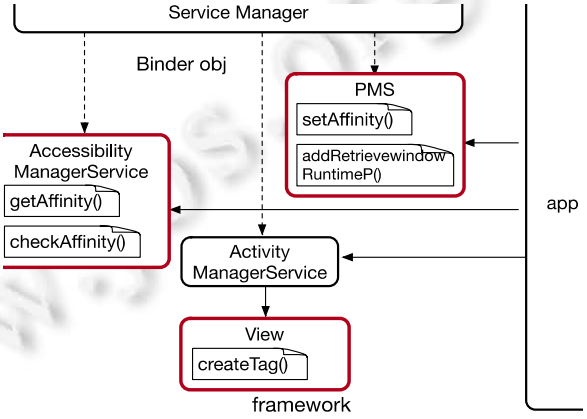


Fig.5 Key modules designed in Tassel system

图 5 Tassel 系统实现中的主要模块

除此之外,为了保证一些隐私数据不被无障碍辅助性服务处理,我们在实现时还进行如下操作.

1) 当一个应用活动启动创建一个视图,而该视图中含有“密码”等隐私信息时,我们设置该视图的 importantForAccessibility 属性为 False.这样,无障碍辅助性服务就不可以对该视图进行操作,无法通过获得视图内容窃取隐私数据.
2) 当一个应用调用无障碍辅助性服务处理事件时,我们首先判断该应用是否在黑名单列表中.目前,应用商店的一些应用已经被报道利用无障碍辅助性服务执行恶意操作,我们将这些应用加入黑名单.若一个应用在黑名单列表或与黑名单列表中的应用有关联关系,则系统拒绝使用无障碍辅助性服务处理事件.

在系统实现过程中,我们还利用安卓系统中接口定义语言 AIDL 的特性,设置接口参数的 in/out 属性来控制信息的流动方向.这样可以保证一些无障碍辅助性服务处理结果不流向非法应用.

3 无障碍辅助性服务安全加固系统的测试

本节对本文实现的系统进行测试评估.

3.1 实验环境

我们的安全加固系统是在安卓 6.0.1AOSP_HammerHead 基础上进行修改实现的,内核版本是 3.4.0-g67595d3.测试使用的设备为 Google Nexus5,处理器是 Qualcomm Snapdragon 800 2.27GHz.

3.2 实验结果及分析

我们主要从系统的可用性、安全性及性能方面对本工作进行评估.

3.2.1 可用性

本文设计的 Tassel 系统主要是在安卓系统基础上添加了一个 Tassel 系统服务来管理无障碍辅助性服务及处理无障碍事件.本节我们主要说明在我们的系统中,安卓整体依然正常运行;系统中的无障碍辅助性服务仍可正常工作,并发挥应有的作用.同时,该服务处理事件的操作受系统制定策略的约束,例如,当活动视图中包含私密信息时,该无障碍辅助性服务的一些特性失效,不能获取视图中的一些信息,并且不能对产生的无障碍事件进行处理.

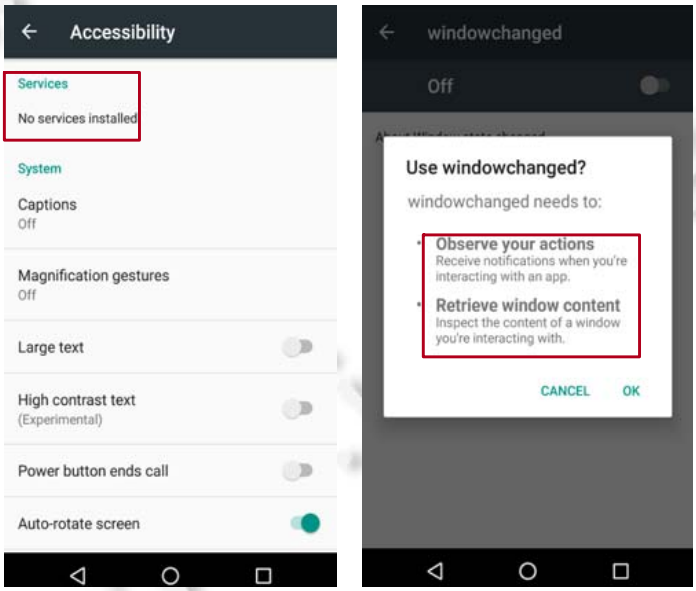
编译修改后的系统并将生成的镜像文件刷入手机设备,我们发现设备系统正常开机运行.分析系统运行日志,本系统中的 Tassel 服务正常开启运行,如图 6 所示.

```
…
SystemServiceManager: Starting com.android.server.tassel.TasselManagerService
TasselManagerService: initialized.
Tassel: start the tasselmanagerservice.
…
```

Fig.6 Piece 1 from log of the Tassel system

图 6 Tassel 系统运行日志片段 1

系统中无障碍辅助性服务运行正常.从图 7(a)可以看到,“No services installed”表示系统应用没有注册无障碍辅助性服务,也没有第三方应用注册无障碍辅助性服务.编写一个样本应用安装在设备中,该应用注册一个无障碍辅助性服务,如图 7(b)所示,可以看到正常注册,并正常获取所需权限.



(a) 系统原始状态 (b) 应用注册无障碍辅助性服务

Fig.7 Normal functioning of accessibility service in Tassel system

图 7 Tassel 系统中无障碍辅助性服务正常运行

在该应用安装注册无障碍辅助性服务过程中,我们系统中的 Tassel 服务被包管理服务调用,并根据当前应用的运行环境初始化,获得应用的一些配置信息对注册的无障碍辅助性服务进行标记:

taTag(for service):com.example.accessibility_apk1000.

当该无障碍辅助性服务开启后,系统中一些窗口界面发生变化时,视图发出无障碍事件,我们的 Tassel 服务在事件初始化时进行标记.由于该简单操作场景下界面变化来源有两种:系统 UI 变化和应用自身窗口变化,下面是对这两个事件的标记:

taTag(for event1): com.android.systemui1001612020;

taTag(for event2):com.example.accessibility_apk100029158.

在对这两个事件分发前,Tassel 服务判断事件 1 不是本应用发出的,所以不予分发处理,而是处理匹配的事件 2.

3.2.2 安全性

我们的系统主要是为了解决无障碍辅助性服务 API 被滥用的问题而设计的.在本系统中,我们规定一个应用注册的无障碍辅助性服务只能对本应用或者有“亲密性”关系的应用产生的无障碍事件处理.一个应用的无障碍事件只能被本应用或者有关联关系的应用注册的无障碍辅助性服务消费处理.

测试时,我们编写并安装两个应用:com.example.accessibility_apk.apk 和 com.example.accessibilityDemo.apk.这两个应用没有被设置 sharedUserId 属性,它们分别注册自己的无障碍辅助性服务并被标记:

taTag(for service1):com.example.accessibility_apk1000;

taTag(for service2):com.example.accessibilityDemo1002.

在应用运行期间,我们监视系统的运行情况,从运行日志中分析这两个服务对运行过程中产生的各种无障碍事件的处理,处理情况如图 8 所示.

```
…
//开启两个应用的无障碍辅助性服务
create tag for accessibility event. TMS
get the packagename of the event: com.android.settings
get the affinity for event:
Tassel: Mismatch, cannot dispatch the event to the service…
//设置“sharedUserId”属性后
//只开启 com.example.accessibility_apk 应用的无障碍辅助性服务
get the packagename of the event:com..examplei.accessibilityDemo
get the affinity of the event: com.example.accessibility_apk
Tassel: Match! the event can be dispatched to the service
get the packagename of the event: com.android.launcher3
Tassel: Mismatch, cannot dispatch the event to the service…
…
```

Fig.8 Piece 2 from log of the Tassel system

图 8 Tassel 系统运行日志片段 2

通过标签匹配,我们发现这些服务都只对各自的无障碍事件进行处理,而不会去处理其他应用产生的无障碍事件.下一步修改应用并设置 sharedUserId 属性,再次运行.通过分析系统运行日志,我们发现这两个应用产生的无障碍事件可以被互相处理,但依然不能处理系统中其他应用产生的无障碍事件,见表 3.我们修改应用使其中一个应用的活动界面包含“password”或者“密码”字样.当应用启动该活动,视图界面变化触发该应用的无障碍辅助性服务试图获得视图内容时,系统阻止该无障碍辅助性服务的操作,记录如图 9 所示.

```
…
get the packagenameof the event: com.example.accessibility_apk
Tassel: the accessibility service is blocked because this view may have something private
…
```

Fig.9 Piece 3 from log of the Tassel system

图 9 Tassel 系统运行日志片段 3

我们还将被暴露滥用无障碍辅助性服务 API 的应用安装在该系统中运行进行测试,例如游戏类执行静默安装的应用、工具类 WIFI 增强应用等.实验过程中我们发现:一些没有注册无障碍辅助性服务的应用不能调用其他应用的无障碍辅助性服务获得视图内容并执行自动点击而完成静默安装;还有一些应用注册了无障碍辅助性服务,由于系统中设置的约束策略,一些自动点击的行为无法执行,静默安装操作无法完成.

3.2.3 性 能

在保证安全性的同时,本系统标记及检测策略的引入不应该导致整体性能下降.由此,我们检测本系统中 Tassel 服务运行过程中的耗时.我们做如下设计.

a) 在我们的系统中安装应用时,Tassel 服务获得包管理服务解析的一些应用属性,进行服务标记并设置“亲密性”属性.这里,我们获得启动 Tassel 服务消耗的时间以及获得标记进行标记消耗的时间,并与源系统测试对比.

b) 在我们的系统中,我们设计在视图生成、初始化无障碍事件并分发处理该事件时进行标记检查.为了测评该过程的耗时,我们记录在 Tassel 系统及源系统中生成一个无障碍事件的时间和该事件被分发处理的时间:从产生事件到事件被处理得到处理结果的时间差.

通过测试,我们获得以下结果:在第 1 部分中,消耗总时间平均为 1ms;在第 2 部分中,消耗总时间平均为 14ms.相对于安卓原系统,本系统中新策略的引入对性能的影响很小,事件处理延迟不超过 20ms.

4 相关工作

4.1 无障碍辅助性服务问题研究

早先有一些学者针对无障碍辅助性服务进行研究.文献[10]中对安卓 2.3 版本的无障碍辅助性服务进行研究,该工作中研究的无障碍辅助性服务是一个系统内部的服务,该系统版本下的无障碍辅助性服务不允许开发者操作调用.此工作从各种系统,包括 Ubuntu、Window、Android、iOS 中的无障碍辅助性服务入手,主要从 I/O 的角度来考虑攻击,进行分析.CVE-2014-4368[18]利用了 iOS 中的无障碍辅助性服务进行攻击,攻击者通过 AssistiveTouch 事件进行锁屏干扰,该漏洞已经被处理.

本文对无障碍辅助性服务的研究是针对高版本系统下优化过的无障碍辅助性服务进行的.安卓 4.0 及以上版本的无障碍辅助性服务相对于安卓 2.3 系统增添了一些特性,并且提供了供应用开发者调用的接口.目前,针对安卓 4.0 以上版本的无障碍辅助性服务的全面研究及有效防范策略还没有.本文主要针对这类系统无障碍辅助性服务进行研究,并设计实现安全加固方案.

4.2 信息流保护机制

目前的计算机系统安全策略有很多,例如访问控制策略[19].这种策略往往考虑的是对象的即时访问特性,而且不考虑相关信息流传播路径.这种模式对于现在的很多场景是不适用的.于是,研究者提出了信息流控制机制[20,21].信息流控制机制是一种用来增强系统安全性的技术,在很多安全类研究中都用到过[22−24].该机制通过使用定义和一些增强手段可以保证数据的私密性,使其在系统中被正确传递,并防止数据被不正当暴露.在信息流系统中,主体和客体被预定义的安全规则标记.安全策略规定了数据流向,该流向遵循有向有限格.该机制有从语言层面来进行标记跟踪的:这种模型往往不能很好地对隐式数据流进行处理;也有从系统层面解决的:这种模型则不能很好地对系统中的应用数据标签进行追踪处理.且该机制暂未广泛地实际应用.

由于传统的安卓权限管理框架不足以用来保护用户的数据,从而导致用户的一些隐私数据很容易遭到泄露.信息流跟踪技术可以解决安卓自身权限系统管理不够的问题.该技术通过一些限定条件来控制系统中的信息流,从而保证一些隐私数据在未授权的情况下不被泄漏出去.然而,该技术仅针对一些已知的对象数据,例如手机设备的 IMEI 进行保护.但在安卓多任务系统中存在着很多无法预知的数据,例如各个应用中的操作数据,传统的信息流跟踪技术将不足以解决多任务应用中的数据泄漏问题.于是,研究者提出了分布式的信息流跟踪技术(DIFC)[25].该技术是信息流跟踪技术的一个拓展,最早由 Myers 和 Liskov 两位研究者提出.该机制可以让每

个应用针对自己的上下文场景做特有的安全标记.这些标记可用来防止特定应用的数据泄漏、防止不安全代码运行在安全系统中带来的安全隐患等.之后,很多系统都采用此技术实现安全策略,例如银行、医务系统等[26].而针对安卓系统的分布式信息流跟踪机制却很少,目前的系统有 Nadkarni 等人提出的 Aquifer 系统[27]、Jia 等人提出的 DIFC 系统[28]、Xu 等人提出的 Maxoid 系统[29]以及 Nadkarni 等人提出的 Weir 系统[30].第 1 个系统通过追踪应用中的数据流,用来保护已知数据的意外泄漏,但该系统禁止了多任务机制,且为了防止标记过多,不对后台组件例如服务、内容提供者等对象进行标记.Jia 等人提出的使用 DIFC 的安全系统中,同样通过在一个应用中限制新方法的调用来禁止多任务机制.在 Maxoid 系统中,为了避免标记爆炸,使用了多实例化方法来分离不同的标记数据,但这个机制仅用在磁盘上的数据,没有考虑内存中的数据.考虑到应用运行时内存中数据的共享,该系统的处理还是不够的.Weir 是 Nadkarni 等人近年来提出的一个安卓上实用的分布式信息流跟踪技术的增强系统,该系统解决了前面的工作中限定多任务、禁止后台组件标记、标记磁盘存储数据的问题,并进一步考虑到安卓系统是网络事件驱动的环境,需要对标记的数据进行“解码”传输,制定了“解码”机制.但该系统依然存在着一些问题,例如,Overlay 文件系统中存储的多实例会带来一定的内存、电池等的消耗.

本文将根据我们的问题场景:安卓系统中无障碍辅助性服务的滥用,基于 DIFC 策略,从设计标记、进行标记、分发及处理解码标记等方面对无障碍辅助性服务进行管理,实现出安全的无障碍辅助性服务安卓系统.本文研究的方案是一种初步模型,考虑到 7.0 版本之后系统的复杂性,例如增添了分屏功能以及系统中网络、文件等元素的处理,策略制定会相对复杂,这将是我们下一步的工作.

5 总结与展望

本文对安卓系统中无障碍辅助性服务进行研究,深入分析了该服务的安全性问题及产生原因,并针对此问题引入分布式信息流控制机制,设计实现了一个安全的无障碍辅助性服务安卓系统.本工作将系统部署在真机上进行测试,测试结果表明:本文设计的系统可以在不影响整体性能的基础上,保证无障碍辅助性服务使用的安全性,防止该 API 被滥用.

随着移动设备的广泛使用,安卓系统也被越来越多的用户使用.而安卓系统及安卓应用安全是目前亟待解决的问题.本文设计的系统对安卓系统安全问题有指导意义.对于未来的工作,为了提供给设备使用者更加灵活的选择,我们可以在系统中增添一个本地服务,该服务与 JAVA 层服务共同作用,将设置无障碍辅助性服务标记的选择权交给用户.若用户开启标记设置选项,则系统的无障碍辅助性服务策略起作用;反之则意味着用户不需要这层保护.

References:

[1] Guide for developing Android accessibility service (in Chinese). http://informationaccessibilityassociation.github.io/androidAccessibility/services.htm

[2] AccessibilityService. https://developer.android.com/reference/android/accessibilityservice/AccessibilityService.html

[3] Zhong Y, Weber A, Burkhardt C, Weaver P, Bigham JP. Enhancing Android accessibility for users with hand tremor by reducing fine pointing and steady tapping. In: Proc. of the 12th Web for All Conf. (W4A 2015). New York: ACM Press, 2015. 10. [doi: 10.1145/2745555.2747277]

[4] Android accessibility security research report (in Chinese). 2016. http://blogs.360.cn/360mobile/2016/09/07/research_of_accessibility/

[5] Amit Y. Android clickjacking—Android malware evolution. 2016. https://www.skycure.com/blog/accessibility-clickjacking/

[6] Amit Y. 95.4 percent of all Android devices are susceptible to accessibility clickjacking exploits. 2016. https://www.skycure.com/blog/95-4-android-devices-susceptible-accessibility-clickjacking-exploits/

[7] Ni B. Abusing accessibility service to install android applications automatically (in Chinese). 2015. http://ju.outofmemory.cn/entry/227941

[8] Rout V. Security issues with Android accessibility. 2016. https://android.jlelse.eu/android-accessibility-75fdc5810025

[9] Venkatesan D. Malware may abuse Android's accessibility service to bypass security enhancements. 2016. https://www.symantec.com/connect/blogs/malware-may-abuse-android-s-accessibility-service-bypass-security-enhancements

[10] Jang Y, Song C, Chung SP, Wang T, Lee W. A11y attacks: Exploiting accessibility in operating systems. In: Proc. of the ACM Conf. on Computer and Communications Security (CCS). 2014. 1−13. [doi: 10.1145/2660267.2660295]

[11] Kraunelis J, Chen Y, Ling Z, Fu X, Zhao W. On malware leveraging the Android accessibility framework. In: Stojmenovic I, Cheng Z, Guo S, eds. Proc. of the Mobile and Ubiquitous Systems: Computing, Networking, and Services. MobiQuitous 2013. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Vol 131. Cham: Springer-Verlag, 2014. [doi: 10.1007/978-3-319-11569-6_40]

[12] Learn about global development. 2018. http://www.worldbank.org/en/topic/disability/overview#1

[13] Accessibility. https://developer.android.com/guide/topics/ui/accessibility/index.html

[14] Kompasim. Grabbing red packets automatically in Android system. 2017 (in Chinese). https://github.com/kompasim/android-wechat-tool/blob/master/README.md

[15] Guo L. The implement for silent installation in Android. 2015 (in Chinese). http://blog.csdn.net/guolin_blog/article/details/47803149

[16] Stefanko L. New Android Trojan mimics user clicks to download dangerous malware. 2017. https://www.welivesecurity.com/2017/02/14/new-android-trojan-mimics-user-clicks-download-dangerous-malware/

[17] Xiaoqi in Jianshu. The event delivery mechanism in Android. 2015 (in Chinese). http://www.jianshu.com/p/cf22ea3b09a5

[18] CVE-2014-4368. http://www.nsfocus.net/vulndb/27932

[19] QuietHeart. Basic principles of access control permissions in Linux. 2017 (in Chinese). http://www.jianshu.com/p/56d5c68b5363

[20] Denning DE. A lattice model of secure information flow. Communications of the ACM (CACM), 1976,19(5):236−243. [doi: 10.1145/360051.360056]

[21] Wu ZZ, Chen XY, Yang Z, Du XH. Survey on information flow control. Ruan Jian Xue Bao/Journal of Software, 2017,28(1):135−159 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5131.htm [doi: 10.13328/j.cnki.jos.005131]

[22] Krohn M, Yip A, Brodsky M, Kaashoek MF, Kohler E, Morris R. Information flow control for standard OS abstractions. In: Proc. of the ACM SIGOPS Operating Systems Review. New York: ACM Press, 2007. 321−334. [doi: 10.1145/1294261.1294293]

[23] Efstathopoulos P, Krohn M, Van De Bogart S, Frey C, Ziegler D, Kohler E, Morris R. Labels and event processes in the Asbestos operating system. In: Proc. of the SOSP. Brighton: ACM Press, 2005. 17−30. [doi: 10.1145/1095810.1095813]

[24] Zeldovich N, Boyd-Wickizer S, Mazieres D. Securing distributed systems with information flow control. In: Proc. of the NSDI. San Francisco: USENIX, 2008. 293−308.

[25] Myers AC, Liskov B. A decentralized model for information flow control. SIGOPS Operating Systems Review, 1997,31(5):129−142. [doi: 10.1145/269005.266669]

[26] Xu S. The research in information flow control model for distributed system [MS. Thesis]. Shanghai: Shanghai Jiaotong University, 2011 (in Chinese with English abstract).

[27] Nadkarni A, Enck W. Preventing accidental data disclosure in modern operating systems. In: Proc. of the ACM Conf. on Computer and Communications Security (CCS). 2013. 1−13. [doi: 10.1145/2508859.2516677]

[28] Jia L, Aljuraidan J, Fragkaki E, Bauer L, Stroucken M, Fukushima K, Kiyomoto S, Miyake Y. Run-Time enforcement of information-flow prop-erties on Android (extended abstract). In: Proc. of the European Symp. on Research in Computer Security (ES-ORICS). 2013. 1−30.

[29] Xu Y, Witchel E. Maxoid: Transparently confining mobile applications with custom views of state. In: Proc. of the 10th European Conf. on Computer Systems. ACM Press, 2015. 1−14. [doi: 10.1145/2741948.2741966]

[30] Nadkarni A, Andow B, Enck W, Jha S. Practical DIFC enforcement on Android. In: Proc. of the 25th USENIX Security Symp. (USENIX Security 2016). USENIX Association, 2016. 1119−1136.

附中文参考文献:

[1] Android 开发无障碍指南. http://informationaccessibilityassociation.github.io/androidAccessibility/services.htm

[4] Android accessibility 安全性研究报告. http://blogs.360.cn/360mobile/2016/09/07/research_of_accessibility/

[7] 逆巴.滥用 Accessibility Service 自动安装应用.2015. http://ju.outofmemory.cn/entry/227941

[14] Kompasim.Android 实现自动抢红包.2017.

[15] 郭林.Android 静默安装实现方案.2015. http://blog.csdn.net/guolin_blog/article/details/47803149

[17] 小七在简书.Android 事件传递机制.2015. http://www.jianshu.com/p/cf22ea3b09a5

[19] QuietHeart.Linux 访问控制权限基本原理.2017. http://www.jianshu.com/p/56d5c68b5363

[21] 吴泽智,陈性元,杨智,杜学绘.信息流控制研究进展.软件学报,2017,28(1):135−159. http://www.jos.org.cn/1000-9825/5131.htm [doi: 10.13328/j.cnki.jos.005131]

[26] 许帅.分布式系统中的信息流控制模型的研究[硕士学位论文].上海:上海交通大学,2011.

李晓娟(1992－),女,山西忻州人,硕士,主要研究领域为安卓系统与应用安全.

陈海波(1982－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为系统软件,系统结构.