

安卓云备份模块的代码安全问题分析

梁蛟¹, 刘武¹, 韩伟力¹, 王晓阳¹, 甘似禹², 沈烁³

(1. 上海市数据科学重点实验室(复旦大学), 上海 201203;

2. 上海市信息投资股份有限公司, 上海 200120;

3. 中国科学院计算机网络信息中心, 北京 100190)

摘要: 随着移动端云备份服务的日益普及, 为保障用户隐私数据不被泄露, 研究第三方应用调用云备份软件开发工具包(SDK, software development kit)的安全问题变得尤为重要。通过对目前国内外安卓应用市场中调用云备份服务的普遍性进行调研, 总结出4个当前主流的安卓云备份SDK。分析其SDK实现代码和官方文档, 对比使用情况、协议和接口功能, 总结和发现了第三方应用错误调用SDK以及云备份SDK自身存在的代码安全问题, 同时向第三方开发者提供了相应的解决方案。

关键词: 移动云存储; 代码安全; 备份服务; 软件开发工具包

中图分类号: TP309

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2017.00132

Code security of mobile backup modules on the Android platform

LIANG Jiao¹, LIU Wu¹, HAN Wei-li¹, WANG Xiao-yang¹, GAN Si-yu², SHEN Shuo³

(1. Shanghai Key Laboratory of Data Science, Fudan University, Shanghai 201203, China;

2. Shanghai Information Investment Inc, Shanghai 200120, China;

3. Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: Since more and more third-party Android applications integrate backup services, the security issues of mobile backup modules are critical. By studying how widely these backup services were being used in the Android applications, the differences of code security about four mainstream Android backup SDK were investigated. After analyzing and comparing the usages, protocols and API functions of these SDK. Based on the above findings and three reported security issues of mobile backup services, the countermeasures for third-party application developers were suggested to securely call the SDK of mobile backup services.

Key words: mobile cloud storage, code security, backup services, SDK

1 引言

随着云计算技术和服务的日益普及, 云备份成为目前信息化基础设施的一项基本服务。近年来, 云备份作为一种新型的网络服务为大众所知, 它可以在提供大量存储空间的同时减少本地存储

消耗, 允许用户通过请求远程服务来上传、下载以及分享他们的个人和商业数据。目前研究表明, 越来越多的个人和企业已经倾向于将其数据存储到云备份服务器, 以保证更低的能源消耗和更高的资源利用率^[1]。迄今为止, 已经出现了大量的云备份服务提供商, 如亚马逊、Dropbox、谷歌、

收稿日期: 2016-09-23; 修回日期: 2016-11-15。通信作者: 韩伟力, wlhan@fudan.edu.cn

基金项目: 上海市科技创新行动计划基金资助项目(No.16DZ1100200, No.15511101500)

Foundation Item: The Shanghai Innovation Action Project(No.16DZ1100200, No.15511101500)

微软、百度等,同时也出现了越来越多的商业云备份服务,如 Dropbox^[2]、Google Drive^[3]、OneDrive^[4]、百度云^[5]等。

最近,随着云备份服务和安卓智能手机的普及,一些云备份服务提供商在目前两大主流移动操作系统——安卓和 iOS 系统上也部署了移动备份服务。移动端的用户可以通过备份他们的照片、通信录、聊天记录,甚至商业文件到云备份服务器来减轻智能手机的存储负担,并借此实现数据的跨平台同步。同时,这些云备份服务平台为了扩大服务影响范围和增强用户体验,向第三方应用提供了接入备份服务的接口和功能,这些云备份服务提供商通常将备份服务集成为一个软件开发工具包,即 SDK^[6],以供第三方开发者使用,帮助简化第三方应用的开发。

然而,在调研中发现,云备份服务提供商所开发的 SDK 往往都存在代码安全漏洞。2015 年 3 月,Dropbox 所提供的 1.5.4-1.6.1 版本的安卓系统 SDK 被曝出有严重的漏洞 (CVE-2014-8889),该漏洞可以让攻击者将一个访问令牌插入到 Dropbox SDK 的 AuthActivity 中,实现本地或远程攻击^[6]。此外,第三方应用可能会错误地调用这些 SDK,当这些应用在移动设备上运行时,攻击者可以利用这些漏洞窃取用户的隐私数据。例如,笔者在调研中发现有些应用将等同于客户端令牌的 App_key 和 App_secret 暴露在应用的配置文件 (string.xml 或 AndroidManifest.xml) 中,攻击者通过反编译安卓安装包 (APK, Android package) 文件轻易获取这些敏感信息。因此,第三方应用在调用移动端云备份 SDK 时也可能面临潜在的代码安全问题。

本文对目前四大主流的安卓云备份 SDK (Dropbox、Google Drive、OneDrive 和百度云)进行了全方位的代码安全性分析,这是第一个在代码安全方面分析各大 SDK 使用量、协议和接口功能的研究。同时也总结和发现了一些云备份 SDK 自身的编码漏洞,以及第三方应用错误调用这些 SDK 可能导致的代码安全问题,并对此为安卓第三方应用开发者提供了一系列调用建议。

本文的主要贡献总结如下。

1) 对安卓国外应用市场 Google Play 和国内应用市场酷安网进行了有关移动端云备份 SDK 调用量的实证分析,发现超过一半的第三方应用没有遵照官方开发指南,这可能会引发一系列的代码安全漏洞。

2) 对四大主流的移动端云备份 SDK 进行了协议分析和比较。协议包括 4 个方面:通信协议、加密协议、去重协议和授权协议。结果表明,这四大 SDK 采用的协议有些不同,但均没有采用加密协议,即在数据传输前对用户的文件数据进行加密。此外,百度云的 SDK 采用了基于源的去重协议,这可能会导致严重的安全问题。

3) 调研了 50 个集成这些云备份 SDK 的开源应用,发现了存在于第三方应用和 SDK 自身的几个代码安全问题,如默认下载路径问题,当第三方应用的默认下载路径设置为外部存储路径 (如 SD 卡的根目录) 时,可能会导致用户不易检测到的数据丢失或数据被窃取问题。基于这些问题,本文给开发人员提供了若干建议来减轻调用 SDK 过程中可能造成的数据泄露等安全隐患。

2 研究背景与动机

2.1 移动云备份存储

随着云计算技术突飞猛进地发展,在线用户,尤其是移动端用户,越来越倾向于利用远程存储服务对其个人或商业数据进行备份、恢复和共享。这些备份服务提供了比之前更友好的用户体验,尤其由于目前用户的手机存储通常是有限的,如 16 GB,通过移动端云备份服务可以为用户节省大量移动设备的存储空间。此外,用户也可以授权安卓的第三方应用,允许这些应用将其文件备份到远程服务器或将数据下载到本地,如图 1 所示。Dropbox、Google Drive、OneDrive 和百度云是目前最著名的云备份服务。

1) Dropbox

美国 Dropbox 公司于 2007 年成立,公司提供的 Dropbox 服务有文件同步、存储服务等功能,并开发供用户使用的客户端软件^[7]。据相关调查显示,截至 2016 年 3 月,Dropbox 服务已经拥有

来自 200 多个国家的 5 亿多用户, 平均每天有 12 亿的文件上传量。此外, 已经有超过 30 万个第三方应用部署在了 Dropbox 平台上。

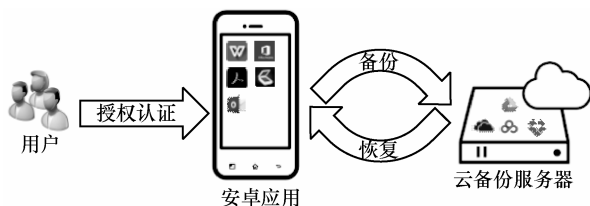


图1 安卓第三方应用调用云备份服务的流程

2) OneDrive

2008 年, 微软开发了 OneDrive (曾用名 SkyDrive, Windows Live SkyDrive, Windows Live Folders) 作为文件备份服务, 允许用户备份文件到服务器, 并通过网页浏览器、桌面客户端或移动设备来访问其存储在远程服务器的数据^[8]。截至 2016 年, OneDrive 已经有超过 2.5 亿的用户使用其云备份服务。

3) Google Drive

Google 公司于 2012 年开发了 Google Drive 作为文件存储同步服务^[9]。据统计数据显示, 到 2014 年 10 月, Google Drive 已经拥有超过 2.4 亿的活跃用户。

4) 百度云

2012 年, 百度网盘正式推出了百度云备份服务。如今, 百度云向用户和第三方开发者提供云存储服务、客户端软件、文件管理、资源共享和第三方集成的服务^[10]。统计数据表明, 截至 2015 年, 百度云的注册用户已超过 3 亿。

这 4 个云备份服务提供商均向第三方开发者提供了移动端平台的云备份 SDK。这些 SDK 通过将云备份服务集成为软件开发工具包, 极大地简化了第三方开发者的开发流程, 更有利于增大服务影响力。开发者可以通过将下载的 SDK 导入到应用工程中对云备份服务进行调用。这些 SDK 的体系结构将云备份服务接口的内部抽象实现进行封装, 仅对外提供有限的功能调用接口, 在节省第三方应用开发者工作量的同时, 也极大地方便了用户的操作和管理。

2.2 动机

本文的研究动机主要基于以下几个方面。

1) 虽然目前安卓系统的云备份模块已经被第三方应用广泛调用, 但至今还没有一个对移动端云备份 SDK 使用量的相关分析。

2) 为了将云备份服务集成到第三方应用中, 所有数据需要通过调用移动端云备份 SDK 来与其远程云备份服务进行交互。目前各项研究已经泄漏了很多云备份服务中身份与数据泄露的漏洞。基于此, 这些备份模块的代码安全问题是保证移动设备数据安全的关键, 但目前针对代码安全的分析还没有出现。

3) 据本文调研显示, 目前移动端云备份 SDK 自身的编码漏洞问题和第三方应用误用这些 SDK 的问题越来越普遍, 通过研究这些 SDK 的代码安全性问题来提醒云备份服务提供商和第三方开发者非常有必要。然而, 这些安全问题和相关措施还没有被总结和提出。

3 部署调研

为了分析上述云备份 SDK 的安全性问题, 本文首先统计了这些安卓云备份 SDK 在第三方应用中被调用的情况。通过从这些不同的云备份服务 (Dropbox、OneDrive、Google Drive 和百度云) 的官方网站中查阅提供给第三方开发者的官方指南和程序示例, 对抓取的所有安卓市场的应用进行反编译, 本文统计了上述 4 个 SDK 在安卓应用中被调用的比率。

如表 1 所示, 通过检测安卓操作系统官方应用市场 Google Play^[11]的前 430 个热门应用, 发现 Google Drive 的 SDK 调用率最高(3.49%), Dropbox SDK 的调用率紧随其后 (2.79%), OneDrive SDK 的调用率是 0.93%。然而, 百度云的 SDK 没有被本次数据集中的应用所调用。此外, 本文还选择了国内最著名的安卓应用市场之一——酷安网^[12]作为另一个调研的市场, 共检测了前 500 个热门应用。统计数据表明, Dropbox SDK 的调用率最高 (9.8%), Google Drive 的 SDK 调用率排第二 (6.00%), OneDrive SDK 的调用率是 2.2%, 百度云 SDK 的调用率最低, 为 0.60%。总体来看, Google Play 应用市场的前 430 个热门应用中, 有 5.11% 的应用调用了这 4 个主流的云备份 SDK; 而酷安网应用市场的前 500 个热门应用中有 12.80%

的应用调用了这些 SDK，可知云备份服务在国内热门应用中普及率更高。在分析过这些调用云备份 SDK 的应用种类后，发现这些应用主要有文件管理器、文档编辑器、文档扫描仪、照片编辑器、笔记和日历表等。

**表 1 Google Play 与酷安网应用市场
云备份 SDK 的使用分布情况**

云备份 服务	Google Play		酷安网	
	应用个数/个	应用比例	应用个数/个	应用比例
Dropbox	12	2.79%	49	9.80%
Google Drive	15	3.49%	30	6.00%
OneDrive	4	0.93%	11	2.20%
百度云	0	0.00%	3	0.60%

4 代码分析

为了进一步分析 4 个移动云备份服务的功能特性，本文通过查阅其官方文档和 SDK 源代码分析它们的接口实现。如表 2 所示，文件管理作为云备份服务的基本功能，包括文件的创建、获取、上传、删除等操作，这 4 个 SDK 均支持该功能。此外，还存在一些其他特性以帮助简化用户操作和提高系统性能。配额管理功能允许用户查询其可用存储空间，帮助其合理地规划剩余空间或通过充值来扩大存储空间，4 个云备份服务提供商均支持此项功能。另一个特性是文件的批量处理，允许用户对文件进行批量上传、下载、删除等操作。据本文的调研分析发现，仅百度云在其 SDK 中支持了此项功能，其他 3 个云备份 SDK 没有提供批量处理功能，第三方应用开发者需在其代码逻辑中自行实现。此外，SDK 中还有一些高级特性来提升用户体验，包括获取文件缩略图、转码视频文件、秒速传输大文件、离线下载等，经过调研分析，这些高级特性仅百度云有全面支持，而 Dropbox、OneDrive 和 Google Drive 的 SDK 均注重于基本功能的实现，并未对高级特性和批量处理功能提供支持。

为保证整个传输过程的有效性和安全性，云备份服务提供商制定和采用了一系列协议对第三方应用调用移动端的云备份服务进行规范。本文

通过研究云备份 SDK 源代码和第三方应用的反编译代码，对以上云备份服务所用到的 4 个主要协议（通信协议、加密协议、去重协议和授权协议）进行了全面的分析比较。

表 2 安卓云备份 SDK 的接口功能统计

云备份服务	Dropbox	Google Drive	OneDrive	百度云
文件管理特性	有	有	有	有
配额管理特性	有	有	有	有
批量处理特性	无	无	无	有
高级功能特性	无	无	无	有

1) 通信协议定义了一个通信系统的规则和通信数据格式，以允许 2 个或多个实体通过通信信道和设备实现资源共享和信息互换^[13]。

2) 加密协议提供了将源数据（明文）转换成另一种形式数据（密文）的转换方式，以实现信息隐蔽，使数据不能被除被授权方外其他人轻易解密，一定程度上保证了信息的安全性和可靠性^[14]。

3) 去重协议旨在消除服务器上重复的数据副本。通过建立数据副本与数据的连接，不保存副本数据本身来节约存储空间^[15]。

4) 授权协议通过用户对一个第三方应用进行授权，使该应用可以代表资源拥有者或者其自身来获取有限 HTTP（hypertext transfer protocol）服务的权限，保证用户的数据只能被该应用访问^[16]。

下文讨论这 4 个协议在 Dropbox、Google Drive、OneDrive 和百度云 SDK 的使用情况及其各自的安全问题，这 4 个云备份服务的协议对比情况如表 3 所示。

表 3 安卓云备份 SDK 的协议使用情况统计

云备份服务	通信协议	加密协议	去重协议	授权协议
Dropbox	HTTPS	无	无	OAuth 1.0/2.0
Google Drive	HTTPS	无	无	OAuth 2.0
OneDrive	HTTPS	无	无	OAuth 2.0
百度云	HTTPS	无	基于源的去重	OAuth 2.0

4.1 通信协议分析

表 3 中四大云备份 SDK 均采用了超文本传输安全协议（HTTPS，secure hypertext transfer

protocol) 作为其网络通信协议, 该协议使用安全套接字层 (SSL, secure socket layer) 来保证客户端与服务器安全地完成信息交换。与超文本传输协议 (HTTP) 相比, HTTPS 会导致资源和性能上的消耗^[17], 影响缓存机制, 并会产生一些诸如将 HTTPS 和 HTTP 混合使用导致的浏览器警告问题^[18], 但是据本文调研得知, 这 4 个主流云备份 SDK 在整个网络请求过程中仍然使用的是安全的通信协议 (HTTPS)。在数据通信过程中, 保证数据传输的安全性是一项基本要素, 如果在第三方应用和云备份服务的通信中不使用安全的通信协议将会导致中间人攻击, 虽然采用 HTTPS 会造成一定程度的性能损失, 但对于第三方应用来说这是有必要的。然而, HTTPS 近几年频繁曝出相关漏洞, 如未进行正确边界检查导致的 Heartbleed 漏洞^[19]、证书过期或不合法导致的 SSL 中间人劫持问题^[20]等。HTTPS 目前仍然存在很多亟待解决的安全问题^[21,22], 如 SSL 证书的信用链体系不完备、加密范围也比较有限等问题, 因此, 将 HTTPS 作为保障用户数据安全的唯一措施是不够的。

4.2 加密协议分析

为进一步保证数据传输以及存储过程中的安全性, 有必要采用加密协议, 以在一定程度上减轻数据被攻击者窃取时的安全威胁。如表 3 所示, 本文发现四大云备份 SDK 均没有在客户端提供加密数据的功能。经过分析, 可能有 2 个原因: 一方面, SDK 的开发者认为 HTTPS 提供了端到端的保护机制, 这已经足够减轻截取网络流量产生的安全威胁; 另一方面, 如果要在客户端对用户数据进行加密, 加密密钥需要保存在客户端, 一旦密钥泄露, 加密也形同虚设。基于以上原因, 云备份服务提供商认为在客户端加密是不必要的。

然而, 如 4.1 节所说, 仅使用 HTTPS 进行网络请求对数据进行传输不是绝对安全的。多数情况下, 用户不会主动加密自己的数据。因此, 为保证用户数据的安全, 云备份服务提供商必须考虑将加密工具整合到其 SDK 中, 而用户加密密钥可以保存在服务器端, 这样就可以在保证只产生较低通信成本的基础上, 显著提高用户个人数据

的安全性。

4.3 去重协议分析

随着云备份服务提供给用户越来越多的网络存储空间, 大量数据被存储到远程备份服务器中。为了消除相同数据的多个副本以节约服务器资源, 需要提供一种特有的数据压缩技术。去重协议的出现可以消除冗余数据, 以提高存储和带宽利用率^[23]。基于此, 出于经济效益的考虑, 几乎所有的云备份服务均采用了去重协议。

目前存在 2 种去重方法: 基于目标的去重方法 (target-based deduplication) 和基于源的去重方法 (source-based deduplication)^[15]。基于目标的去重方法需要在服务器端进行去重, 即数据文件需要先传输到服务器, 再由服务器根据已存在的数据确定是否为冗余数据, 以决定是否需要保留该数据文件。该去重方法会显著提高服务器端的存储利用率, 但不会节约网络带宽, 对客户端和用户完全透明。基于源的去重方法是指数据文件会在客户端传输前进行判断和去重操作, 即源数据传输前会先将其散列值传到服务器端, 由服务器判断该数据是否已存在, 如果已存在, 则通知客户端不必对该数据再次传输, 实现了数据文件的秒速传输, 在提高服务器存储利用率的同时, 也节约了客户端的网络带宽。

为进一步确认以上四大云备份 SDK 的去重协议, 本文在调研分析中进行了如下实验。

1) 创建一个网络中独有的 550 M 的加密文件 (通过加密文件来保证其在备份服务器上的唯一性)。

2) 将该文件通过第三方应用调用相关云备份 SDK, 分别传输到 Dropbox、Google Drive、OneDrive 和百度云服务器中。

3) 监控各自的网络流量并记录上传时间。

4) 重命名该加密文件, 并再次通过第三方应用将其上传到这 4 个云备份服务器中, 继续监控网络流量和上传时间。

5) 针对 4 个云备份服务分别比较 2 次实验的网络流量和上传时间判断文件是否秒速上传。

基于源的去重方式最显著的特点是当用户上传一个云服务器上已存在的文件时, 会实现秒速上传功能。本文就以上 4 个云备份 SDK 的去重协

议进行了调研,如表3所示,结果表明,Dropbox、Google Drive 和 OneDrive 的 SDK 采用的是基于目标的去重协议,而百度云 SDK 采用的是基于源的去重协议。

此外,在实验过程中,发现有些云备份 SDK 是基于某个特定大小的数据块为单位进行去重,而不是完整的文件数据块。通过监控网络流量,发现 Dropbox 将整个文件分成 8 MB 的数据块进行分片传输和去重,而 OneDrive 以 160 kB 作为数据块大小。此外,百度云 SDK 的数据块大小为 4 MB。但是,Google Drive 并没有提供基于数据块去重的方法,而是以整个文件为单位进行传输与去重。与基于块的去重方式相比,这种方法去重粒度更大,除了会使服务器的存储利用率有明显降低外,还会使大文件的传输失败率增加,导致用户不易察觉的数据丢失等安全问题。

虽然对客户端和服务端而言,云备份服务采用的去重协议有很多优势,但其中仍然有很多潜在的安全威胁,尤其是对于基于源的去重方法。百度云 SDK 所采用的基于源的去重方法是通过服务器在文件传输前检查其信息——摘要算法 5 (MD5, message-digest algorithm 5) 值来确认服务器端是否已存在该副本,如果已经存在,则由服务器端直接生成一个文件链接,而省去了客户端与服务端网络传输该文件的操作。

然而,现有研究已经证明使用 MD5 值来签名一个文件,存在严重的漏洞,攻击者可以事先构建一个文件 B 为文件 A 的 MD5 值碰撞^[24],并将其上传至服务器(需保证文件 B 的首创性,不会被服务器作为冗余数据去重)。由于文件 A 与文件 B 的 MD5 值相同,当用户在随后请求上传文件 A 时,百度云的备份服务器通过判断发现服务器上已存在 MD5 值相同的文件,并直接为文件 A 生成一个 MD5 值相同但文件内容不同的文件 B 的链接,导致文件 A 被判断为冗余数据,并没有通过实际的网络传输保存在备份服务器,最终导致用户的文件 A 数据丢失。此外,基于源的去重可以被攻击者利用来确认文件是否已存在于服务器,或者恶意获取文件的内容等,即侧信道攻击^[15,25]。虽然目前很多研究者已经提出了相关方法来减少基于源的去重方法所引起的安全威胁^[26-28],但是

这些云备份服务提供商仍然没有将这些方法应用到其 SDK 的开发中。因此,为尽可能规避风险,当第三方开发人员将基于源去重方法的云备份 SDK 集成到应用中时,需要提醒用户不要上传敏感数据,或提前通过对数据进行加密来阻止服务器对文件的去重行为。

4.4 授权协议分析

开放授权标准 (OAuth, open authorization) 是一种开放式授权协议,允许用户授权第三方网站或应用访问用户存储在另外服务提供者上的资源信息,而无需提供用户名与密码给第三方网站与应用^[16,29]。在云备份 SDK 中,采用了 OAuth 授权协议使第三方应用在不获取用户云备份用户名与密码的前提下,可以访问其存储在云服务器上的资源。目前,OAuth 协议已经有 2 个版本: OAuth1.0 和 OAuth2.0。为了简化授权流程, OAuth2.0 与 1.0 版本相比做了很多改善,提供了更多的 OAuth 协议流以支持移动端应用程序的用户授权,并且不再需要客户端应用实现签名等复杂的密码学算法^[30]。由表 3 可知,有些云备份服务提供商,如 Google Drive、OneDrive 和百度云在其安卓 SDK 开发中直接废弃了 OAuth1.0 授权方式,而 Dropbox 仍然支持 OAuth1.0 授权,但是将 OAuth2.0 作为默认的授权方式。

目前,OAuth2.0 已经通过各种方法和途径被证明协议本身是安全的^[31-33]。为了避免包括窃取网络流量中的敏感信息、XSS 攻击和跨站点请求伪造等潜在的攻击^[34], OAuth 工作团队给开发者提供了正式的官方指南,即 OAuth 攻击模型^[35]。但,在实际中,开发者很容易因为协议中没有明确的解释和定义而造成对协议的误用和错用^[36,37],尤其是由于移动端与 Web 端系统实现机制的不同所导致的一系列安全问题^[38,39]。例如,使用不安全的方式来传输或保存密码和访问令牌等敏感信息。基于此,本文进行了实验,通过监控网络请求来检测一些敏感信息是否泄露。在实验中发现,一些应用将访问令牌(可作为用户登录和确认用户身份的安全证书)作为一个键值属性以明文方式加入到 HTTP 请求中,攻击者可以通过拦截 HTTP 请求轻易获取用户的访问令牌,进而窃取用户存储在远程备份服务器中的资源数

据。为了避免不必要的数据丢失和泄露,第三方应用和云备份 SDK 开发者应严格遵守 OAuth 协议官方指南。

5 安全问题与应对策略

随着目前越来越多的安卓开发者将云备份 SDK 集成到其应用中,移动备份模块的安全性问题必需引起广泛的重视。然而,由于云备份 SDK 自身的编码漏洞和第三方开发者不正确的调用,导致目前仍然存在很多安全问题。通过进行广泛的调研分析工作,本文讨论云备份 SDK 的以下几个代码安全问题。为尽可能减轻安全威胁,本文为第三方开发人员提供了相应的对策。

5.1 移动端云备份 SDK 自身的编码漏洞

为简化应用的开发流程,云备份服务向第三方应用提供了 SDK。SDK 是一个函数库,为第三方应用访问云备份服务提供了简便的调用方式。虽然使用这些 SDK 可以显著降低第三方应用的开发周期,并有利于提升用户体验,但目前这些 SDK 自身仍然存在很多缺陷,调用这些 SDK 第三方应用的安全性也会因此受到严重影响。

2015 年 3 月,Dropbox 所提供的安卓平台 SDK 的 1.5.4-1.6.1 版本被曝出认证机制存在严重的安全漏洞^[6] (CVE-2014-8889),该漏洞允许攻击者通过将访问令牌插入到 Dropbox SDK 用于身份认证的 AuthActivity 中,绕过其随机数防护来窃取用户信息。AuthActivity 是一个安卓应用的活动(activity,安卓系统组件之一),可以接受几种不同的意图(Intent,安卓系统用于提供应用间或组件间交互与通信的机制)的附加参数(extra parameter)。由于 AuthActivity 将属性 exported 和 browsable 设置为 true,使其可以被携带有任意 Intent 附加参数的恶意应用和网站启动。研究发现,AuthActivity 中有一个 Intent 附加参数 INTERNAL_WEB_HOST 可以被攻击者控制,最终绕过 Dropbox 的随机数防护机制进行攻击。基于此,攻击者可以利用此漏洞注入自己的访问令牌,将用户手机设备的第三方应用连接至自己的 Dropbox 云备份账户,导致用户将敏感数据备份至攻击者账户,并可能会从攻击者账户下载到恶意数据。

2015 年 11 月,乌云网^[40]曝出了百度 Moplus SDK 漏洞^[41],漏洞被命名为虫洞(wormhole),引起业界广泛关注。由于 Moplus SDK 在实现中保留了后门功能,因此也使攻击者可以借此进行恶意行为,如远程静默安装应用、远程打开任意网页、远程读取写入文件等。百度云应用使用了该 Moplus SDK,由于第三方应用调用百度云 SDK 实现云备份服务的前提是手机设备中已安装了百度云应用,因此,所有调用百度云 SDK 来备份用户数据的第三方应用都会受到 Moplus SDK 漏洞的影响。

一旦 SDK 的漏洞被披露,云备份服务提供商通常会第一时间修复该漏洞,并发行一个新的 SDK 版本。虽然第三方开发者无法避免此类 SDK 自身的编码漏洞,但在调研中发现有些第三方应用不会经常检测其所调用的 SDK 版本,导致这些第三方应用仍然可能会受到已有漏洞的攻击。因此,第三方开发人员需要及时更新云备份 SDK 的版本或设置自动检查 SDK 的版本更新来尽可能保证用户数据的安全性。

5.2 安卓第三方开发者错误调用云备份 SDK 的问题

除了云备份 SDK 自身的编码漏洞所导致的用户存储在远程服务器上的数据泄露外,第三方开发者对云备份 SDK 的不正确调用仍然是一个潜在的威胁。为了帮助第三方开发者简化其集成云备份服务的过程,每个云备份服务提供商均提供了 SDK 及其调用指南,但出于开发过程中的简便性和性能考虑,很多第三方开发者并不会遵守这些使用说明。此外,还有对官方调用指南理解不当导致的误用问题。根据本文调研分析,这可能会导致严重的安全问题。此外,为了保证第三方应用调用云备份服务的安全性,本文为第三方开发人员提供了相应的对策。

1) App_key 和 App_secret 参数的硬编码问题

在第三方应用集成一个移动端云备份 SDK 之前,需要首先在相应的云备份服务中注册该应用,之后会得到一个云备份服务提供商授权的 App_key 和 App_secret。当第三方应用调用 SDK 的身份认证接口时,需要提供 App_key 和 App_secret 让云备份服务提供商确认应用的身份

信息。然而，一旦这2个参数被泄露，攻击者就可以用它们来伪造第三方应用实施恶意行为。虽然这些云备份服务可以在发现异常行为后限制该应用的权限，但这仍然会影响到应用对接口的正常调用。因此，为了使应用正常运行，保证 App_key 和 App_secret 的安全是必需的。

然而，在检测了 Google Play 应用市场的 22 个调用云备份 SDK 的应用和酷安网应用市场的 64 个应用后，本文发现很多开发者仍然没有遵守云备份 SDK 的官方调用指南。结果表明，Google Play 应用市场选取的 22 个应用中，有 2 个应用（9.09%）将其 App_key 和 App_secret 暴露在 string.xml 和 AndroidManifest.xml 等配置文件中，而在酷安网应用市场选取的 64 个被测应用中，有 5 个应用（7.80%）也出现了上述问题。这意味着攻击者通过反编译应用的 APK 文件就可以轻易获取这些敏感信息。为了尽可能避免此类漏洞，第三方应用需要将 App_key 和 App_secret 写入 Java 代码中，并对代码进行混淆。当然，更安全的方法是将 App_key 和 App_secret 保存在应用的服务器端。

2) 访问令牌的存储问题

访问令牌包含登录会话所需的安全证书和用户的身份认证信息。用户在对第三方应用进行授权后，云备份服务器会颁发访问令牌给该应用，以允许其访问用户存在云备份服务器上受保护的资源。为了避免重复进行认证请求导致的应用性能和网络耗费问题，第三方应用通常需要将访问令牌存在本地以方便保存会话状态。然而，访问令牌存储的位置是否足够安全，需要进一步讨论。为分析此类问题，本文选择了 50 个调用这些云备份 SDK 的开源安卓应用，并分析了相关源代码。结果表明，所有应用均调用一个名为 SharedPreferences 的 Java 类来存储访问令牌，该类可以在应用的数据文件夹下创建一个可扩展标记语言（XML，extensible markup language）文件，文件路径为/data/data/your_packet_name/shared_prefs/your_prefs_name.xml。由于安卓系统的沙箱保护机制，一个外部应用很难访问到其他应用沙箱内的数据。多数情况下，将访问令牌通过调用 SharedPreferences 类来存储是安全的。然而，如果

用户的移动设备已经拥有了 root 权限，且存储在 SharedPreferences 中的访问令牌并未加密，则仍然会导致严重的用户数据泄露问题。

据腾讯调查显示，截至 2014 年 11 月，已经有超过 $\frac{1}{4}$ 的用户（27.44%）对其设备进行了 root^[41]，这会导致很多严重的威胁。因此，如果第三方开发者需要在 SharedPreferences 中安全地存储访问令牌或其他敏感信息，必须考虑此项安全因素。一个可行的解决方案是在访问令牌等敏感信息存入到 SharedPreferences 之前对其进行加密，保证即使攻击者获取了 SharedPreferences 中的数据也无法对其进行解密。此外，另一种有效的方式是将访问令牌等敏感信息存到应用的服务器中，即在调用云备份服务接口之前，第三方应用需要向其服务器发送请求以获取访问令牌。以上 2 种方法均可有效保护访问令牌的安全。

3) 默认保存路径的问题

当用户通过第三方应用从云备份服务器中下载一个文件到本地时，为简化用户操作，第三方应用通常会设置一个默认保存路径。如图 2 所示，通过分析 50 个开源安卓应用的源代码，发现默认下载路径通常是一个外部存储路径，如 SD 卡的根路径。这会导致攻击者也有权限访问和修改这些文件，由于用户通常不会在下载前检查文件内容，因此攻击者可以在用户察觉不到的情况下私自窃取和篡改用户的敏感文件，如收入、身体状况、身份证信息等数据。

为了避免此类数据泄露问题，本文提出第三方开发者需要将文件的默认保存路径设置为应用内部路径，如应用的沙箱路径。此外，如果用户选择将外部存储路径作为下载路径，则需要提醒用户确保文件中不包含任何敏感信息。

6 相关工作

云备份服务的安全问题一直以来都被工业界和学术界广泛讨论与研究^[42]。目前，很多研究主要集中在分析云备份服务提供商是否完整地保存有用户文件的问题，即提供给用户确认文件未被云备份服务器泄露或修改的方法。Subashini 等^[43]总结了由于云计算的服务传递模型导致的安全问题，并提出了几个当前云计算技术所面临挑战的

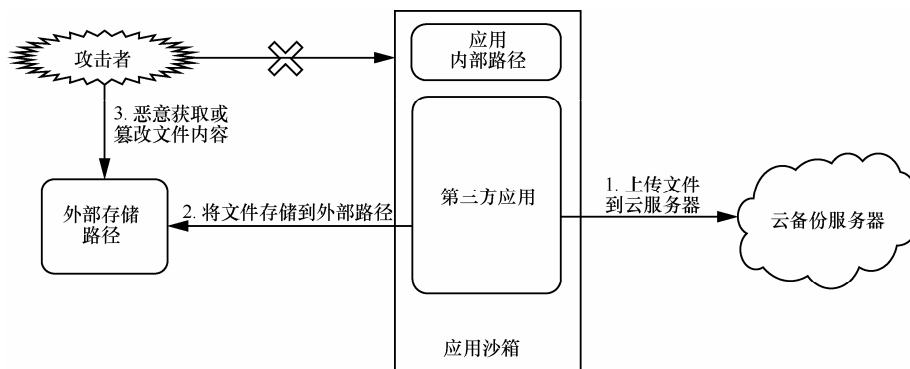


图2 外部下载路径导致的攻击

解决方案。Chow 等^[44]对采用云备份服务会引起的安全问题和严重性进行了分类，并提出第三方应用对数据的控制问题是一个安全隐患。此外，数据持有性验证模型^[45]的出现向用户提供了在不对文件进行逐个检索的前提下，判断云备份服务器上是否存有该数据的验证方法。另外，很多研究提供了更多改进的存储协议^[46-49]。

为了节约云服务器上的存储空间和客户端与服务器间的网络传输带宽，目前很多云备份服务均采用了去重协议^[50]，即确保远程服务器上仅有一份真实存在的文件或文件数据块，去除重复的冗余数据。此外，为了在此基础上保证数据的机密性，Douceur 等^[51]提出了收敛加密（CE，convergent encryption），保证相同的文件加密后仍然相同。之后很多研究主要集中在改进去重和 CE 相结合的云备份系统，以同时保证存储空间利用率和安全性^[26,27,52]。此外，Harnik 等^[15]对基于源的去重方式总结了其侧信道攻击的漏洞^[25]。文中建议用户通过在上传文件前对其数据进行加密来阻止服务器的去重操作，并提出了将去重随机化的解决方案来应对去重所导致的攻击。Mulazzani 等^[53]提出通过客户端的数据所有权证明来避免散列值篡改攻击，他们发现了 Dropbox 的相关漏洞来说明此类问题的严重性。

近几年，由于云备份服务和移动设备的快速崛起，很多研究者将其注意力转移到移动端云备份的安全问题上。基于此，为简化安卓第三方应用的开发，云备份服务提供商通常会开发 SDK 供第三方开发者调用。云备份 SDK 的安全问题直接影响所有调用这些 SDK 的第三方应用的用户数据安全。Hay 等^[6]发现了安卓系统中 Dropbox SDK

的严重缺陷，这个缺陷会导致本地和远程攻击。此外，目前很多安全社区正在致力于检测云备份 SDK 的漏洞。例如，国内的乌云网平台发现了百度 Mopplus SDK 的 Wormhole 漏洞，百度云 SDK 也在此波及范围内，使攻击者可以在用户检测不到的情况下通过一个开放的 HTTP 服务器访问存储在受影响应用中的数据内容^[54]，该漏洞导致 1 亿台安卓设备受到攻击。

然而，以上的研究工作均没有对安卓云备份 SDK 的代码安全问题做一个全面分析，即这些工作都将注意力集中到发现一个特定 SDK 的某个缺陷上或提出一个更加安全的系统协议上。本文的研究工作与其他工作不同的是，本文致力于对安卓系统的移动云备份模块进行全面的代码安全分析。本文提供了 4 种不同的安卓云备份 SDK（Dropbox、Google Drive、OneDrive 和百度云）的安全性分析。在总结了其调用情况、协议和接口功能后，本文总结和发现了几个由于 SDK 自身的编码漏洞和第三方开发者错误调用云备份 SDK 所引发的安全问题。最后，本文向第三方开发人员提出了相应对策来尽可能规避这些安全风险。

7 结束语

本文首次对目前四大主流的安卓云备份 SDK（Dropbox、Google Drive、OneDrive 和百度云）进行了代码安全方面的分析。在统计了第三方应用在国内两大安卓应用市场 Google Play 和酷安网上调用云备份 SDK 的比率说明其普及性后，本文对这 4 个 SDK 进行了接口功能的分析，并通过进行源代码分析对 4 类协议（通信协议、加密

协议、去重协议和授权协议)的使用进行了讨论,结果发现这4个SDK均没有在数据传输前对用户文件数据进行加密,同时百度云SDK采用的基于源的去重协议可能会导致严重的安全漏洞。此外,本文总结和发现了云备份SDK自身的代码安全问题和第三方应用开发者错误调用SDK所导致的漏洞。基于此,本文向第三方开发者提供了相应措施以尽可能保证第三方应用的用户数据安全。

接下来,笔者计划将注意力转移到发现更具影响力的安卓平台移动云备份模块的安全隐患上。此外,由于本文总结的代码安全问题仍然基于实例分析,之后计划将这些分析方法集成为一个自动化扫描工具,通过对安卓应用中云备份SDK调用的安全问题进行大规模扫描,提出更具普适性的解决方案。

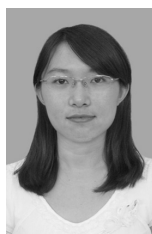
参考文献:

- [1] BUYYA R, YEO C S, VENUGOPAL S, et al. Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility[J]. *Future Generation Computer Systems*, 2009, 25(6): 599-616.
- [2] [EB/OL]. <https://www.dropbox.com/>.
- [3] [EB/OL]. <https://www.google.com/drive/>.
- [4] [EB/OL]. <https://onedrive.live.com/>.
- [5] [EB/OL]. <https://yun.baidu.com/>.
- [6] HAY R, PELES O. Remote exploitation of the dropbox SDK for Android[EB/OL]. <https://dl.packetstormsecurity.net/1503-exploits/exploiting-dropboxsdk-android.pdf>.
- [7] [EB/OL]. [https://en.wikipedia.org/wiki/Dropbox_\(service\)](https://en.wikipedia.org/wiki/Dropbox_(service)).
- [8] [EB/OL]. <https://en.wikipedia.org/wiki/OneDrive>.
- [9] [EB/OL]. https://en.wikipedia.org/wiki/Google_Drive.
- [10] [EB/OL]. https://en.wikipedia.org/wiki/Baidu_Cloud.
- [11] [EB/OL]. <https://play.google.com/store>.
- [12] [EB/OL]. <http://www.coolapk.com/>.
- [13] [EB/OL]. https://en.wikipedia.org/wiki/Communications_protocol.
- [14] VLADIMIROVA T, BANU R, SWEETING M N, et al. On-board security services in small satellites[C]//IETF RFC. 2000.
- [15] HARNIK D, PINKAS B, SHULMAN-PELEG A. Side channels in cloud services: deduplication in cloud storage[J]. *IEEE Security & Privacy*, 2010, 8(6):40-47.
- [16] RECORDON D, HARDT D. The OAuth 2.0 authorization framework[J]. *Polymer*, 2009, 50(24): 5708-5712.
- [17] GOLDBERG A, BUFF R, SCHMITT A. A comparison of HTTP and HTTPS performance[J]. *Computer Measurement Group*, 1998.
- [18] SUN S T, HAWKEY K, BEZNOSOV K. Systematically breaking and fixing OpenID security: formal analysis, semi-automated empirical evaluation, and practical countermeasures[J]. *Computers & Security*, 2012, 31(4):465-483.
- [19] DURUMERIC Z, KASTEN J, ADRIAN D, et al. The Matter of Heartbleed[C]// *Conference on Internet Measurement*. 2014:475-488.
- [20] 魏兴国. HTTP 和 HTTPS 协议安全性分析[J]. *程序员*, 2007(7):53-55.
- [21] WEI X G. Security analysis of HTTP and HTTPS protocol[J]. *The Programmer*, 2007(7):53-55.
- [22] FAHL S, HARBACH M, MUDERS T, et al. Why eve and mallory love Android: an analysis of Android SSL (in)security[C]//ACM Conference on Computer and Communications Security. 2012: 50-61.
- [23] FAHL S, HARBACH M, PERL H, et al. Rethinking SSL development in an appified world[C]//ACM SigSAC Conference on Computer & Communications Security. 2013:49-60.
- [24] PATIL M A V, KALE M N D. Survey on secure authorized de-duplication in hybrid cloud[J]. *International Journal on Recent and Innovation Trends in Computing and Communication*, 2014, 2(11): 3574-3577.
- [25] WANG X, YU H. How to break MD5 and other hash functions[C]//International Conference on Theory and Applications of Cryptographic Techniques. 2005:561-561.
- [26] PULLS T. (More) Side channels in cloud storage[M]//Privacy and Identity Management for Life. Berlin Heidelberg: Springer, 2011: 102-115.
- [27] BELLARE M, KEELVEEDHI S, RISTENPART T. DupLESS: server-aided encryption for deduplicated storage[C]// *Usenix Conference on Security*. 2013:179-194.
- [28] LIU J, ASOKAN N, PINKAS B. Secure deduplication of encrypted data without additional independent servers[C]//The 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM, 2015: 874-885.
- [29] XU J, CHANG E C, ZHOU J. Weak leakage-resilient client-side deduplication of encrypted data in cloud storage[C]//The 8th ACM SIGSAC Symposium on Information, Computer and Communications Security. 2013: 195-206.
- [30] 张天琪. OAuth 协议安全性研究[J]. *信息网络安全*, 2013(3):68-70.
- [31] ZHANG T Q. Study on OAuth protocol security[J]. *Netinfo Security*, 2013(3):68-70.
- [32] HAMMER-LAHAV E. Introducing oauth 2.0[J]. *Hueniverse*, 2010.
- [33] PAI S, SHARMA Y, KUMAR S, et al. Formal verification of OAuth 2.0 using alloy framework[C]//The International Conference on Communication Systems and Network Technologies. 2011: 655-659.
- [34] CHARI S, JUTLA C S, ROY A. Universally composable security analysis of OAuth v2.0[J]. *Iacr Cryptology Eprint Archive*, 2011.
- [35] SLACK Q, FROSTIG R. OAuth 2.0 implicit grant flow analysis using Murphi[J].
- [36] ALESSANDRI A, BESCHI S, CASCIARO R, et al. The devil is in the (implementation) details: an empirical analysis of OAuth SSO systems[C]//ACM Conference on Computer and Communications Security. 2012:378-390.
- [37] MCGLOIN M, HUNT P. OAuth 2.0 Threat model and security considerations[J]. *Internet Engineering Task Force (IETF) RFC*, 2013.
- [38] KIANI K. How to secure your oauth implementation[EB/OL].

<https://www.mendeley.com/research/four-attacks-oauth-secure-oauth-implementation/>.

- [37] WANG R, ZHOU Y, CHEN S, et al. Explicating SDKS: uncovering assumptions underlying secure authentication and authorization[C]//Presented as Part of the 22nd USENIX Security Symposium. 2013: 399-314.
- [38] CHEN E Y, PEI Y, CHEN S, et al. Oauth demystified for mobile application developers[C]//The 2014 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2014: 892-903.
- [39] WANG H, ZHANG Y, LI J, et al. Vulnerability assessment of oauth implementations in android applications[C]//The 31st Annual Computer Security Applications Conference. ACM, 2015: 61-70.
- [40] [EB/OL]. <http://www.wooyun.org/>.
- [41] [EB/OL]. <http://www.androidheadlines.com/2014/11/50-users-root-phones-order-removebuilt-apps-one.html>.
- [42] FERNANDES D A B, SOARES L F B, GOMES J V, et al. Security issues in cloud environments: a survey[J]. International Journal of Information Security, 2013, 13(2):113-170.
- [43] SUBASHINI S, KAVITHA V. A survey on security issues in service delivery models of cloud computing[J]. Journal of Network & Computer Applications, 2011, 35(1):1-11.
- [44] CHOW R, GOLLE P, JAKOBSSON M, et al. Controlling data in the cloud: outsourcing computation without outsourcing control[J]. ACM Workshop on Cloud Computing Security, 2009:85-90.
- [45] ATENIESE G, BURNS R, CURTMOLA R, et al. Provable data possession at untrusted stores[C]//ACM Conference on Computer and Communications Security. ACM, 2007:598-609.
- [46] JUELS A, KALISKI B S. Pors: proofs of retrievability for large files[C]//ACM Conference on Computer and Communications Security. ACM, 2007:584-597.
- [47] BOWERS K D, JUELS A, OPREA A. Proofs of retrievability: theory and implementation[C]//ACM Cloud Computing Security Workshop 2009:43-54.
- [48] ATENIESE G, PIETRO R D, MANCINI L V, et al. Scalable and efficient provable data possession.[C]//The 4th International Conference on Security and Privacy in Communication Networks. 2008:1-10.
- [49] ERWAY C C, PAPAMANTHOU C, TAMASSIA R. Dynamic provable data possession[J]. ACM Transactions on Information & System Security, 2009, 17(4):213-222.
- [50] QUINLAN S, DORWARD S. Venti: a new approach to archival storage[C]// The Conference on File and Storage Technologies. USENIX Association. 2002:89-101.
- [51] DOUCEUR J R, ADYA A, BOLOSKEY W J, et al. Reclaiming space from duplicate files in a serverless distributed file system[C]//The International Conference on Distributed Computing Systems. 2002: 617-624.
- [52] BELLARE M, KEELVEEDHI S. Interactive message-locked encryption and secure deduplication[M]//Public-Key Cryptography-PKC 2015. Berlin Heidelberg: Springer, 2015:516-538.
- [53] MULAZZANI M, SCHRITTWIESER S, LEITHNER M, et al. Dark clouds on the horizon: using cloud storage as attack vector and online slack space[C]//USENIX Security. 2011:5.
- [54] [EB/OL].<http://blog.trendmicro.com/trendlabs-security-intelligence/setting-the-record-straight-on-mopplus-sdk-and-the-wormhole-vulnerability/>.

作者简介:



梁蛟(1993-), 女, 山西忻州人, 复旦大学硕士生, 主要研究方向为云备份系统安全、访问控制。



刘武(1992-), 男, 湖南湘潭人, 复旦大学硕士生, 主要研究方向为云备份系统安全、访问控制。



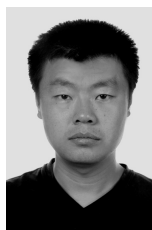
韩伟力(1975-), 男, 浙江绍兴人, 博士, 复旦大学副教授, 主要研究方向为访问控制、数字身份安全、网络与系统安全。



王晓阳(1960-), 男, 上海人, 博士, 复旦大学特聘教授、博士生导师, 主要研究方向为数据库、并行式数据分析和信息安全。



甘似禹(1966-), 男, 江西抚州人, 上海市信息投资股份有限公司大数据研究院高级工程师, 主要研究方向为电子数据交换、数据质量和系统安全。



沈烁(1977-), 男, 辽宁阜新, 博士, 中国科学院计算机网络信息中心物联网中心副主任、副研究员, 主要研究方向为物联网标准体系、信息安全。

网络
信息安全学报

Chinese Journal of
Network and Information Security

读者信箱:
cjni@bjxintong.com.cn
官方微信:



2017年1月15日出版

定价: 30.00元

ISSN 2096-109X

