Computer Science and Application 计算机科学与应用, 2018, 8(8), 1277-1291
Published Online August 2018 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2018.88138

Hans汉斯

Security Analysis and Verification of the Game Library

Ruoying Li, Shuai Shao, Haoliang Cui, Shaozhang Niu

School of Computer, Beijing University of Posts and Telecommunications, Beijing
Email: 18603150177@163.com

Received: Aug. 5th, 2018; accepted: Aug. 20th, 2018; published: Aug. 28th, 2018

Abstract

In recent years, the security issues of Android third-party libraries have appeared in endlessly. In order to study the potential security threats of third-party game libraries, in this paper, 21 mainstream third-party game SDKs are analyzed in depth and three process models are established to reveal the implementation details of the game SDKs. Then it proposes seven security rules and explains five kinds of attack scenarios which game SDK providers, game developers, and users may encounter after the violation of these security guidelines. Lastly, this paper proposes three methods for detecting violations of security guidelines, and uses these three methods to detect the 21 game SDKs and 2000 game applications in Android application markets. The experimental results show that the 21 game SDKs all violated at least one security rule, resulting in hundreds of gaming applications facing security risks.

Keywords

Third-Party Game Libraries, Security Rules, Process Models, Attack Scenarios

游戏库的安全性分析与验证

李若影，邵　帅，崔浩亮，牛少彰

北京邮电大学计算机学院，北京
Email: 18603150177@163.com

收稿日期：2018年8月5日；录用日期：2018年8月20日；发布日期：2018年8月28日

摘　要

近年来，Android第三方库的安全问题层出不穷。为了研究第三方游戏库潜在的安全威胁，该文深入分析了21个主流的第三方游戏SDK，并建立了三种过程模型以揭示游戏SDK的实现细节，然后提出了7条

文章引用: 李若影, 邵帅, 崔浩亮, 牛少彰. 游戏库的安全性分析与验证[J]. 计算机科学与应用, 2018, 8(8): 1277-1291.
DOI: 10.12677/csa.2018.88138

安全准则，并说明了违反这些安全准则后游戏SDK提供方、游戏开发方、用户可能遭遇的5种攻击场景。最后，该文提出了检测违反安全准则的三种方法，并利用这三种方法对这21种游戏SDK及Android应用市场中的2000个游戏应用程序进行了检测，实验结果表明，这21个游戏SDK均违反至少一条安全规则，导致上百个游戏应用程序面临安全风险。

关键词

第三方游戏库，安全准则，过程模型，攻击场景

1. 引言

近几年，移动应用程序市场迅速发展。大量 Android 应用程序使用第三方库，第三方库通过提供已经实现的具体功能，使得应用程序的开发更加便利。有研究表明，许多应用程序使用超过 20 个第三方库[1]，在某些极端情况下，一个应用程序可以引入超过 30 种不同的第三方库[2]。在现存的 Android 第三方 SDK 中，游戏类 SDK 尤为突出，被应用程序开发者广泛使用。iiMedia Research 显示，2016 年中国手游用户规模达 5.23 亿人，市场规模达 783.2 亿元[3]。据统计，每个 Android 游戏 APP 平均使用 SDK 的数量达 17.5 个 0。由此可见，游戏类 SDK 的使用数量不断上升，在 Android 第三方 SDK 中占据重要地位。

然而，尽管第三方 SDK 在开发中被普遍使用，它们却往往存在潜在安全问题，例如，它们泄露用户私人信息，利用其宿主应用程序的权限或跟踪用户。文献[4]指出中国最大的移动广告提供商 Taomike 所提供的广告 SDK 被发现存在秘密监视用户的行为；文献[5]指出 Google Cloud Messaging 提供的推送消息服务 SDK 允许攻击者劫持用户的注册 id 并获取用户隐私信息；文献[6]指出主流应用程序内支付服务提供商提供的支付 SDK 规定商家使用 URL 存储支付命令，这使攻击者很容易获取并篡改支付命令，给用户和商家造成极大经济损失。由此可见，大量第三方 SDK 存在严重安全问题，亟待研究与解决。

本文旨在回答以下问题：1) 什么原因导致了游戏 SDK 存在安全漏洞？2) 攻击者可能对游戏 SDK 执行什么类型的攻击？3) 怎样检测游戏 SDK 中存在的安全漏洞？为了解决这些问题，本文对游戏 SDK 进行了深层次分析，包括以下几个方面：1) 通过逆向工程第三方游戏 SDK 和相关应用程序，分析游戏 SDK 的主要功能及其实现流程，并建立过程模型。2) 总结出一系列规范游戏 SDK 的安全规则。3) 举例说明违反这些安全规则后可能遭受的攻击。4) 使用自动化和手动方法结合来检测在游戏 SDK，游戏 SDK 服务器，游戏应用程序及游戏应用程序服务器上违反安全规则的行为。

2. 相关工作

针对 Android 第三方库的分析研究主要包括同源性分析和漏洞分析。在同源性分析方面，Michael Backes 等人[7]提出一种可以精确检测 Android 应用程序中使用的第三库的技术，它利用原始的第三方库生成特征文件，然后对文件进行特征匹配，此方法可以应对代码混淆技术，不仅可以检测应用程序中使用的不同库，还可以检测出所使用的库版本。Charlie Soh 等人[8]提出一种自动检测第三方库的工具 LibSift，这种检测工具基于应用程序的包依赖图，并采用了模块去耦技术提高识别精度。Ziang Ma 等人[1]首先利用聚类技术收集了大量第三库特征，然后开发了一种叫做 LibRadar 的工具，结合静态分析和特征匹配技

术识别应用程序中的第三方库。

基于大量对第三方库同源分析技术的研究，不少研究者分析了 Android 应用程序中各类第三方库存在的安全问题，并提出了相应的检测方法。Yangyi Chen 等人[5]提出了一种工具用于检测推送消息类 SDK 的漏洞，它旨在从 SDK 的示例代码及应用程序的 Manifest 文件中提取指纹，接口等信息，并利用这些信息分析SDK 提供的身份认证方式及客户端 id 的保护方式，它可以检测客户端混淆，服务混淆等漏洞。Wenbo Yang 等人[6]分析了第三方支付类 SDK 可能遭受的攻击，例如秘钥泄露，支付命令被替换等，文章针对每种类型的攻击提出了具体的检测方法。Soteris Demetriou 等人[9]分析了广告库获取用户隐私的四种方式并提出一个移动风险评估框架来检测应用程序是否将个人敏感数据泄漏给广告库。Hui Wang 等人[10]揭示了由于 Android 应用程序中利用第三方 SDK 实现 OAuth 协议而产生的漏洞，并提出一个漏洞评估框架，此框架可以为 Android 应用程序中 OAuth 的实现流程建立五方三阶段模型，从而检测两种类型的漏洞。

上述方法实现了对特定类型的第三方 SDK 的漏洞检测技术，但都不能针对游戏 SDK 的特点进行安全检测，目前也没有研究者对游戏 SDK 进行系统性的安全分析。

3. 过程分析

游戏 SDK 通常包含两种功能，一是用户体系功能，例如注册、登陆等；二是支付功能，通常提供多种支付渠道以实现游戏 APP 中充值服务。虽然游戏 SDK 功能相似，但应用程序如何通过第三方游戏 SDK 完成服务的过程却不完全一致。第一，游戏 SDK 没有统一的规范；其次，游戏 SDK 开发者经常向应用程序开发者提供简单的文档和代码示例，然而这些资料的大部分是模糊的，可能会误导应用程序开发人员。

为了揭示游戏 SDK 服务过程的细节，本文选择了当下流行的游戏 SDK，对它们的示例代码及开发文档进行研究分析，并 SDK 及相关应用程序的源码进行分析。然后，本文总结了游戏 SDK 登陆、支付和找回密码流程，包含基本步骤及传递的参数。

3.1. 登录模型

图 1 所示为一个简单的登陆过程模型，整个过程通常包含以下 8 步：

1) 游戏客户端初始化第三方游戏 SDK，调用 SDK 登陆接口。

2) SDK 客户端向 SDK 服务器发送登起请求，将用户信息 userinfo (例如账号，密码等)及 APP 信息 appinfo (例如客户端唯一标识 APPID)加密并传递给 SDK 服务器。

3) SDK 服务器将接收到的 userinfo 与数据库信息对比，并生成登陆状态码 code，登陆令牌 token 和签名 sign，然后响应 SDK 客户端登陆请求。

4) SDK 客户端处理接口的回调结果，将获取的信息传递给 APP 客户端，token 中保存了用户身份信息。

5) APP 客户端将获取的 token 和 sign 作为用户登陆标识向 APP 服务器发送登陆请求。

6) APP 服务器将登录信息发送到 SDK 服务器，请求 SDK 服务器验证此次是否为有效的用户会话。

7) SDK 服务器找到对应的会话，验证登陆是否超市，同时校验签名的正确性，然后返回校验结果(包括登陆状态码，信息描述等)。

8) APP 服务器接收校验结果后，通知 APP 客户端登陆结果，APP 客户端执行登陆操作并将用户登陆态信息存储到本地数据库中。

3.2. 支付模型

图 2 所示为一个简单的支付过程模型，整个过程通常包含以下 10 步：

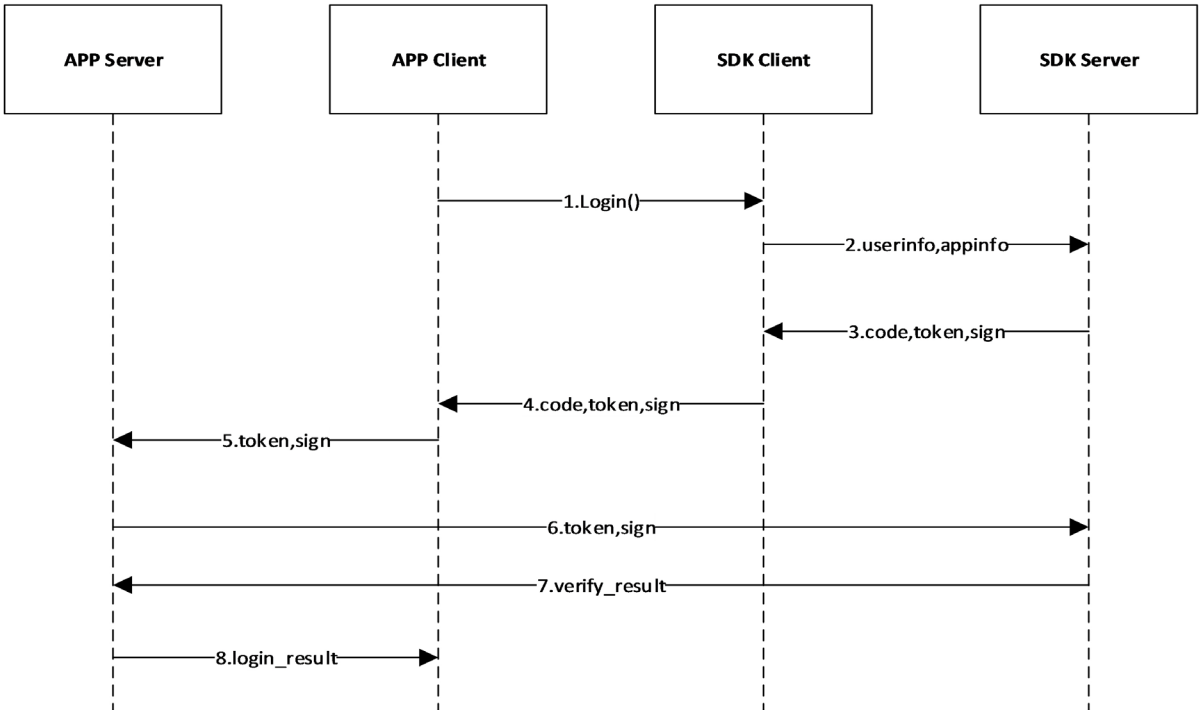

Figure 1. Login process model
图 1. 登陆过程模型

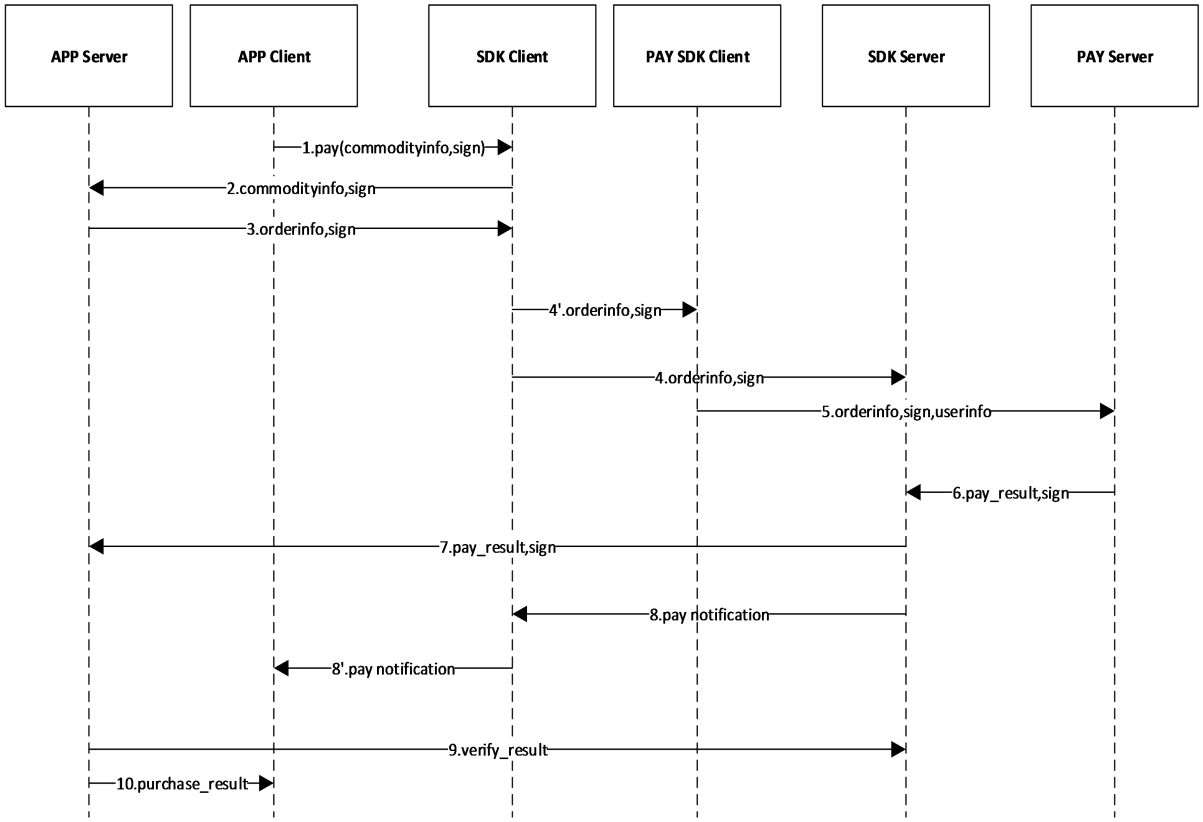

Figure 2. Payment process model
图 2. 支付过程模型

1) APP 客户端调用游戏 SDK 的支付接口，并将商品信息 commodityinfo (例如商品数量等)通过接口传递给 SDK 客户端。

2) SDK 客户端向 APP 服务器发送商品信息和签名以请求充值服务。

3) APP 服务器生成加签订单，并将订单信息 orderinfo (例如订单 ID 等)和签名返回给 SDK 客户端。订单信息和签名将保存在 APP 服务器的数据库中。

4) SDK 客户端根据 orderinfo 中的支付渠道调用对应的支付 SDK，将支付信息发送给 SDK 服务器，SDK 服务器设置订单状态为未完成。

5) 支付 SDK 客户端根据 orderinfo 中的信息向支付服务器发送支付请求，支付服务器回调支付 SDK 方法，弹出支付界面并获取用户输入的支付账户信息 userinfo。

6) 支付服务器对订单进行处理，将充值结果信息 pay_result (包含订单号支付金额，支付失败原因等)和 sign 发送给 SDK 服务器。

7) SDK 服务器更新订单状态并存储充值信息，然后通知 APP 服务器充值结果。

8) SDK 服务器使用回调接口通知 SDK 客户端充值结果，APP 客户端提示用户支付结果。但此时，用户购买商品并未同步。

9) APP 服务器校验签名和订单信息，将校验结果返回给 SDK 服务器。

10) 若第 9 步支付结果校验正确，APP 服务器根据订单信息将用户购买的商品发放给 APP 客户端，并将用户购买商品信息添加到数据库中。

3.3. 找回密码模型

图 3 所示为一个简单的重置密码过程模型，整个过程通常包含以下 9 步：

1) APP 客户端调用 SDK 密码重置接口，用户在密码重置界面输入手机号码，并通过接口参数传入 SDK 客户端进行处理。

2) SDK 客户端向 SDK 服务器发送 userid，手机号，令牌 token，请求 SDK 服务器发送验证码。

3) SDK 服务器生成验证码，请求短信服务商发送短信。

4) SDK 服务器通知 APP 客户端验证码发送成功。

5) 用户在 APP 客户端界面输入接收到的验证码，通过对应接口传递给 SDK 客户端。

6) SDK 客户端发送 userid，手机号，用户输入的验证码到 SDK 服务器，请求校验正确性。

7) SDK 服务器校验验证码是否正确并检查 token 是否在有效期，然后返回给 SDK 客户端校验结果，允许或拒绝用户修改密码。

8) 用户在 APP 客户端输入新密码，并由 SDK 客户端发送给 SDK 服务器，SDK 服务器，SDK 服务器将新密码存储到数据库中。

9) SDK 服务器返回给 SDK 客户端密码重置结果，SDK 客户端通知用户密码重置结果。

4. 安全分析

4.1. 安全准则

大部分游戏 SDK 在发布之前已经通过审核并且被认为是安全的，但由于整个游戏服务过程涉及多方交互，任何一方在多方模式中犯下的任何错误都可能导致整个过程的脆弱性。因此，有必要制定安全规则，对模型中的各方进行规范。

根据上面三种过程模型，本文总结了以下准则，要求游戏开发方和游戏 SDK 提供方必须遵守。

1) 永远不要把任何秘密信息存放在客户端 APP 中，本地重要信息应加密存储。

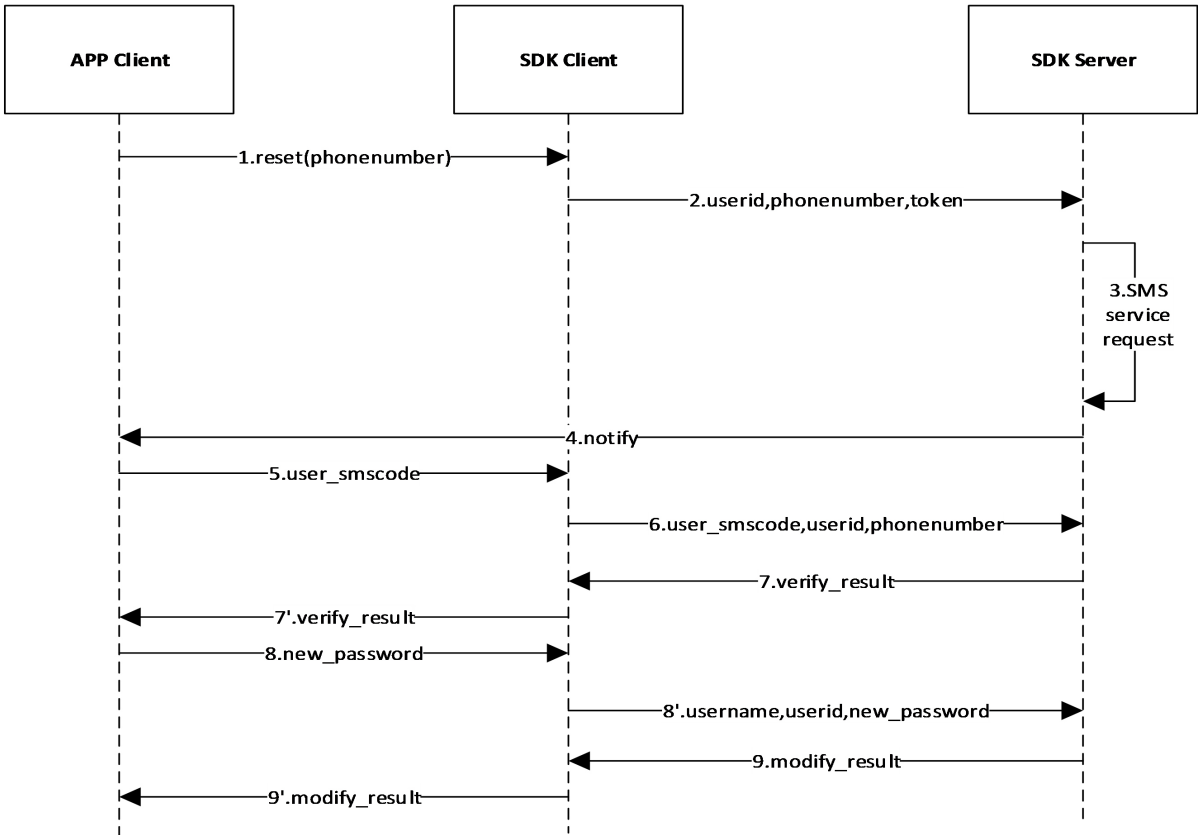

Figure 3. Password retrieval process model
图 3. 找回密码过程模型

2) 校验短信验证码的过程必须在 SDK 服务器端完成。

3) 客户端和服务器之间必须使用安全的网络通信。

4) userid 生成方式应是无规律的。

5) 支付相关的信息必须签名。

6) SDK 必须将付款订单的详细信息通知用户。

7) 通信过程中必须验证接收到的信息的签名。

下面本文将详细描述违反安全规则后会遭遇的 5 种攻击。

4.2. 篡改命令

如果违反了第 1 条，第 5 条或第 7 条中任意一条安全规则，可能会遭受命令被篡改的攻击。在这种情况下，攻击者可以篡改支付订单中的内容，例如篡改支付金额，使得攻击者可以支付较少的费用而获取到商品。

在支付模型中，SDK 客户端将用户购买的商品信息发送给 APP 服务器，APP 服务器生成订单号返回给 APP。在这个过程中，SDK 客户端发送的商品信息和 APP 服务器返回的订单信息都应加签，并且在传递过程中，任何一方都应该验证接收到的签名。订单信息是由 APP 服务器生成并签名，攻击者无法篡改获取的信息。但是，如果订单签名的秘钥在 APP 客户端中，攻击者就可以截获商品或订单信息并篡改，利用获取的秘钥和篡改后的信息生成签名，重新发送给接收端，此时因为签名是合法的，接收端将把篡改后的信息当作正常信息处理。

4.3. 伪造通知

在找回密码过程模型中，SDK 服务器负责校验验证码，若校验成功，SDK 服务器通知 APP 客户端允许用户修改密码。但是，如果违反了第 2 条安全规则，即校验验证码的工作由 APP 客户端完成，攻击者可以修改本地代码逻辑，无论是否收取到验证码，无论修改密码操作是否由获得允许，攻击者都可以使 APP 客户端获取校验成功的通知。在这种情况下，攻击者只要任意输入电话号码，便可以在没有验证码的情况下修改此账户的密码，对用户造成危害。

在支付过程模型中，如果同时违反了第 1 和第 7 条安全规则，也可能会遭到伪造通知的攻击。在完成支付后，支付 SDK 服务器会将支付结果通知发送给 APP 服务器。在这项攻击中，攻击者不会完成支付，而是伪造一个假的支付结果通知发送给 APP 服务器，使得 APP 服务器认为订单已经支付成功。若想完成这个攻击过程，攻击者需要知道支付 SDK 服务器通知 APP 服务器时的 URL。开发人员经常会将此 URL 随意放入 APP 的本地信息中，这违背了第 1 条安全规则。同时，如果 APP 服务器忽略校验通知的签名，这个通知就会被当作正常信息处理。

4.4. 替换命令

这种攻击是由于违反了第 3 条或第 6 条中任意一条安全规则，攻击者通过替换多方之间传输的数据，使用户遭受危害。

当 APP 服务器和 APP 客户端之间使用不安全的网络通信时，攻击者可以执行中间人攻击，截取它们之间传递的信息。在支付过程模型中，攻击者可以用一个伪造的合法订单替换 APP 服务器生成的订单，接下来的过程中，用户将支付攻击者伪造的订单，因为这个订单的格式和信息都是合法的，因此 SDK 会正常接收并处理这个订单。支付订单被替换还有另一个重要原因，即 SDK 没有向用户展示支付订单的细节。在这种情况下，用户没有确认生成的订单就进行了支付。例如，SDK 只向用户展示用户购买的商品时，攻击者可以伪造与原订单具有相同商品信息，金额却不同的订单，此时用户无法辨别订单的不同，攻击者便可以成功完成攻击。

在找回密码过程模型中，用户重置密码后，SDK 客户端将用户名，userid 和用户重置的新密码发送给 SDK 服务器，在这个过程中，攻击者可以监听网络并截取这个过程中的数据包，替换用户重置密码命令，使用户无法成功登陆。这类攻击的根本原因是缺乏安全的通信信道。我们发现多数游戏 SDK 客户端和服务器之间使用 HTTP 通信协议和简单的加密算法，很容易通过中间人攻击工具监听到传输内容。

4.5. 伪造客户端

每一个游戏 APP 用户的账号都有一个 userid，它是唯一确定一个用户的标识，userid 通常由服务器生成，服务器通过网络请求中的 userid 来区分来自不同用户的请求。

根据第 4 条安全规则，userid 的生成方式应该是无规律的，否则攻击者很容易推断出被攻击者的 userid，并以此作为攻击条件。例如一些游戏 APP 请求服务器信息时需要使用 userid，若攻击者获取被攻击者的 userid，就可以伪造用户请求并获取用户信息。

4.6. 非法登录

如果违反了第 1 条或第 3 条安全规则中任意一条，攻击者可能获取身份验证信息伪造登录请求。在登录过程模型中，SDK 服务器会生成唯一的编码 token 并将 token 与用户信息一一对应存入数据库中，之后 token 会作为安全交互的重要信息，开发者经常会将 token 存放在本地 APP 数据库或私有文件中，便于直接读取，甚至有些 SDK 会将用户名和密码存在私有文件。如果这些信息没有加密存储，攻击者便很

容易获取用户的登录信息。另一方面，如果在 token 和账号信息在传递过程中使用不安全网络通信也同样会造成隐私泄露。

5. 检测

由上述章节可知，违反七项安全规则可能会造成可利用的攻击。在本章中，将详细描述如何检测违反这些安全规则的行为，以便发现真实游戏应用程序中存在的漏洞。本文的检测方法主要分为三类，自动化静态检测、中间人检测和手动检测。自动化检测的主要检测对象是 SDK 源码和游戏 APP 源码，中间人检测主要的检测对象是各方之间的网络交互数据，而手动检测主要是检测 SDK 的显示界面、生成文件等或手动篡改 SDK 各方之间的网络通信数据。

5.1. 自动化静态检测

对于URL信息，我们在DEX文件和资源文件中搜索URL特征，然后自动化提取SDK中所有的URL，最后判断来确定这之中是否有涉及到服务器之间通信的 URL 泄露。

对于签名信息，我们可以通过开发文档得知 SDK 采用的签名方式，然后根据不同的签名方法来搜索关键函数内参数或静态变量是否有涉及秘钥泄露；另外，我们发现一些支付秘钥存在明显的特征，例如支付宝秘钥通常是以‘MI’开头且长度超过 300 的字符串。

根据安全规则 3，如果 SDK 使用的是 HTTPS 通信，我们将检查它是否正确验证 SSL 证书。此漏洞分两种情况，第一，SDK 自定义信任所有证书；第二，SDK 没有对证书的主机名进行校验。这两种情况有明显的代码特征，可以通过代码特征匹配完成检测。

5.2. 中间人检测

根据安全规则 2，校验短信验证码的过程应在服务器端完成。我们在中间人攻击工具 Fiddler 基础上实现自动获取短信验证码校验过程数据的功能，通过重建短信验证过程来检测此类漏洞。

根据安全规则 3，游戏服务器，游戏客户端，SDK 服务器之间应该采用安全的通信方式。我们利用MonkeyRunner录制登陆和充值的脚本，在执行脚本的同时利用Fiddler自动获取过程中的全部网络数据。MonkeyRunner 是一种 Android 自动化测试工具，可以通过它提供的 API 来控制 Android 设备和模拟器运行，另外，可以运行 monkeyrecoder.py 脚本录制应用程序测试脚本，并通过 monkeyplayback.py 脚本来解释执行测试脚本。我们收集了一些网络传输中隐私数据参数常用名称，例如密码参数名常用 password，pwd，accessPasswd，passwd 等，我们检查在获取的网络数据中是否有这些隐私数据的关键字，如果 SDK 采用 HTTP 协议且在网络数据中出现账号、密码、订单号、手机号、邮箱、支付金额等参数名，我们认为它是不安全的。

根据安全规则 4，userid 应该是无规律的。为了检测这一漏洞，我们利用 MonkeyRecorder 记录注册框的坐标并随机生成账号和密码，然后利用 MonkeyRunner 在 APP 上自动注册大量用户，同时利用 Fiddler 自动获取注册过程中的 userid 的值，最后比较 userid 的信息熵并与随机数的信息熵，并调用 Matplotlib 工具绘制 userid 数据的散点图，以检测 userid 是否有规律。我们用同样的方法检测违反安全规则 5 的行为，主要检测网络传输数据中是否有例如 sign，Signature，mhtSignature，sig，paySig 等签名参数常用字段。若没有，则认为是有漏洞的。

5.3. 手动检测

根据安全规则 1，一些重要信息应加密存储。一般情况下，开发者会将 token 存储在数据库或配置文件中，我们在 root 后的手机上检查 APP 的数据库和私有文件，查找 token 信息或用户账号和密码，从而

确定这些信息是否加密存储。

为检测违反安全规则 6 的行为，我们手动检查游戏 SDK 是否在支付过程中向用户显示了支付订单的详细信息，重点检查以下字段：1) 支付订单 ID。2) 用户支付的商品信息。3) 该支付订单所属的用户账号。4) 该支付账单所属的游戏 APP 信息。5) 用户所选用的支付渠道。6) 支付订单的总额。若游戏 SDK 的支付界面未向用户提供上述信息，则认为该游戏 SDK 的支付订单有被篡改的风险。

根据安全规则 7，任何一方都应检测接收到的签名。为了检测这个漏洞，我们拦截所有的支付和登陆过程中的数据包，修改签名后再发送出去，如果接收方没有返回签名错误或者参数错误等信息，则证明它没有正确校验签名。

6. 实验评估

我们从 Android 应用市场中根据下载量从高到低下载了 2000 个游戏 APP，超过 50%的 APP 集成了表 1 中的游戏 SDK，因此我们选取这些游戏 SDK 进行实验，检测它们是否存在第五章中提到的安全缺陷，研究对象包括这些游戏 SDK 的官方文档，SDK 源码和集成这些游戏 SDK 的 APP，实验结果表明这些游戏 SDK 都违反了至少一条安全规则。

Table 1. The name and version of SDKs
表 1. SDK 名称及版本

SDK 名称	SDK 版本
dangle	v4.4.1
SnailGameSDK	v1.6.0
SogouGameSDK	v2.1.0
UmipaySDK	v4.08
BDGameSDK	v3.7.4
AnzhiSDK	v4.2.3
4399SDK	v1.0.4
tencent	v3.3.3
laohuSDK	v3.11.0
37SDK	v3.2.6
lianyun	v4.5
DouyuSDK	v1.1
iSDK	v1.3.2
MiguSDK	v4.1.0
mzw_onlinesdk	v3.3.8.1
Amigo Play SDK	v4.0.6
hmPaySDK	v1.3.1.7
guopan	v4.0.1.2
UCGameSDK	v7.3.4.2
SuperSdk	v3.0.3
49sdk	v7.3.3

6.1. 违反安全规则 1

我们对 500 个集成这些 SDK 的游戏 APP 进行检测，发现近 20%的 APP 存在秘钥泄露的情况，尤其在搜狗游戏 APP 和偶玩游戏 APP 中，我们检测到了支付秘钥泄露。例如，图 4 是在搜狗游戏龙腾传世 APP 中检测到的秘钥泄露，该秘钥用于生成游戏 APP 向 APP 服务器请求订单号时的签名；我们通过分析 APP 的源码得出利用此秘钥生成签名的方式，然后利用中间人攻击获取到签名所需的网络数据(表 2)

```
public static final String ORDER_PRIKEY =
"MIICdgIBADANBgkqhkiG9w0BAQEFAASCAmAwggJcAgEAAoGBAO0m9rBaOFCEj4ncScPeC+6H63X
MHhs4xb08lR2TbthAPKIZV3jZB0cuh91M3XJcpdhlHUGbLhbWlmG5xKgN1Lt8Z+QoebfNEyyKM06I9Ye
DSykwRyEjhhOUgLjeIVV3NI8T/awhl+tb/0yyld+5aoXJKxOx/pzqolzoDRs0omEzAgMBAAECgYBGzwt5P
Hb0E6CIGS4tPW9ymULEuV2D4z+ncR9U5WCDUSrJe6eSfbqellYazYiRTPh31DkYDa2FRC1CoKUHSJnrj
eNR2TMw0WUBFvNcqYe2qOJZg3iOhyUDhIChhQiWWC9VrzAvqSU6tuyKGMy5rAWbfTneEnL7NHsTg
RRDC+0JAQJBAPlRGW6T4TnRBtbOpRcMU+jdCyJAK3zwuRO13alhexDLq105D1osg2uP1d3+XvTQudw
CGo1qRfBSp/W72fynz5kCQQDzgmLyxGzO1rugtJNMLQTqsRGg8ZUoUPmsEVGbmnHwRzd2OGHWbT1
JuIEEb+ivrZV3PfeEObv7fDAT6qIhyiarAkAcd4ka2iG+U0KfpkqtXgf6r7qEt6T/iBDp0js0CuBdY5P2efxpxGl
hD7RQu6ml9Gs0Vr0nZnoD3bw1z7QtKBAJAkBiqBjesqZCxs0NtxtWaYbsbwDta/M6elQtWnbtzA0NhEz8IK
vC7E9AZvgejBiB1JoRzZFSiPGYWiBAcXduqTAxAkEAqG24ePhjesKoF1Us2ViqgJC7zDd96v+LI5eausw3
TfKjO4jj5oMoQiyc+hZFxHYlkyZRfA6XEraF1Rdgngf65w==";
```

Figure 4. The key leaked in the Sogou game APP
图 4. 搜狗游戏 APP 中泄露的秘钥

Table 2. The network communication data of Sogou game APP
表 2. 搜狗游戏 APP 的网络通信数据

Name	Value
model	Nexus 5
payInfo	{appID:”247”,channelID:”27”}
roleName	怀美琪
serverName	传世 141 服
productID	1
sign	cfyzsH+1qf8gVlsk2PasLkvMBnr1sgsopV9iIiyOqukFopZW0G8Zyuyq4GvVRl07gES8qFr7pLd+lHb4Czfjc6FZlAt3ig85IhL8PoXYMtdPF0OKzNmXKR8506TPK9PhDfldQUp+XU2/JjrFamUrw+70A/AL6jnrmm+UfCur2F4=
extension	%7B%22payData%22%3A%5B%221%22%2C%22%E6%80%80%E7%BE%8E%E7%90%AA%22%2C%22110936154%22%2C%2210%22%2C%2210141%22%5D%7D
productDesc	购买 1000 元宝
userID	110936154
ratio	0
money	1000.0
serverID	10141
mnos	4
systemversion	4.4.4
roleID	24659548
productName	1000 元宝

并通过反编译 APP 获取 Manifest 数据(图 5)，最后利用这些信息计算出了正确的签名(图 6)。我们尝试用此方式为篡改的订单生成签名并发送给 APP 服务器，最终 APP 服务器正常处理并返回订单号。

在支付过程中，SDK 客户端负责调起支付，而支付宝要求传入 notify_url 参数，这个参数为支付宝服务器主动通知游戏服务器支付结果的 HTTP/HTTPS 路径。因此，若游戏 SDK 集成了支付宝的支付方式，则这个游戏 SDK 开发的游戏 APP 中会包含此 URL，从而造成 URL 泄露。我们检测了一些 Android 市场中包含支付宝支付方式的游戏 APP，结果这些 APP 均含有 URL 泄露。图 7 是在搜狗游戏 APP 中检测出的 URL 泄露。

对于 token 和账号信息的检测，我们分析了这些集成这些游戏 SDK 的 APP，检测沙箱目录和外存中 APP 生成的文件等，表 3 为有信息泄露的游戏 APP。

6.2. 违反安全规则 2

我们获取集成这些 SDK 的游戏 APP 的密码找回过程的网络数据。结果发现在当乐游戏 APP 存在客户端校验短信验证码的情况，当验证码输入正确时，APP 会直接进入登录流程，而不会接收到验证码校验成功的通知。因此，我们认为这类游戏 APP 使用了错误的校验方式。

6.3. 违反安全规则 3

我们发现这些游戏 SDK 均存在网络通信方面的漏洞，而且它们传输信息的主要方式是 HTTP。例如，

```
<meta-data
        android:name="TANWAN_GAME_ID"
        android:value="247">
</meta-data>
```

Figure 5. Manifest data of Sogou game APP
图 5. 搜狗游戏 APP 的 Manifest 数据

```
cfyzsH+1qf8gVlsk2PasLkvMBnr1sgsopV9iIiyOqukFopZW0G8Zyuyq4GvVRl07gES8qFr7pLd+lHb4Czfjc6
FZlAt3ig85IhL8PoXYMtdPF0OKzNmXKR8506TPK9PhDfldQUp+XU2/JjrFamUrw+70A/AL6jnrmm+UfCu
r2F4=
```

Figure 6. Generated signature
图 6. 生成的签名

```
const-string v1, “&notify_url=http://211.154.151.196.8081/tenpay/payment/notifyCallBackUrl
```

Figure 7. The leaked URL in APP
图 7. APP 中泄露的 URL

Table 3. Privacy disclosure in APP
表 3. APP 中的隐私信息泄露

游戏 APP	是否有 token 泄露	是否有账户信息泄露
dangle	是	否
SogouGameSDK	是	否
laohuSDK	是	否
DouyuSDK	是	否
iSDK	否	是
SnailGameSDK	是	否

搜狗游戏 SDK 利用简单的加密算法甚至是明文传递信息，攻击者可以获取到手机号，账号，密码；蜗牛游戏 SDK 使用简单的加密方式传递信息，解密后可以获取到账号，密码，而且在支付过程中，可以获取支付账号和支付密码等重要信息；4399 游戏 SDK 在找回密码过程中，用户登录账号和登陆密码为明文传递。另外，这些 SDK 中均涉及到 HTTPS 通信，且都存在 SSL 通信客户端检测信任任意证书的漏洞，攻击者可以使用中间人攻击获取加密内容。图 8 是 4399 游戏 SDK 的网络传输数据，密码采用明文传递。

6.4. 违反安全规则 4

我们对游戏 SDK 检测了 userid 相关的漏洞，结果发现，搜狗游戏 SDK 服务器、蜗牛 SDK 服务器、拇指玩游戏 SDK 服务器和综合游戏 SDK 服务器生成的 userid 是顺序的，攻击者很容易推测连续注册用户的 userid。

6.5. 违反安全规则 5

我们对这些游戏 SDK 执行了篡改订单的攻击。结果发现，蜗牛 SDK 发送给 APP 服务器商品信息不携带签名字段(表 4)，订单信息可随意被篡改，例如我们选择充值 100 元，然后将订单中的 amount 字段由 100 修改为 1，最终服务器返回的订单的支付金额为 1 元(图 9)，攻击者可以用较少的金额获取商品；同样，偶玩游戏 SDK 在支付宝支付过程中没有出现任何签名信息，攻击者可以任意重改用户的订单信息。

6.6. 违反安全规则 6

我们手动检查了这些游戏 SDK 支付 Activity，发现大部分 SDK 没有向用户呈现完整的订单信息，容易发生订单被替换的危险。统计结果如表 5 所示。

```
POST https://anquan.4399.com/pwd/?_c=save&_a=pwd&jsoncallback=_HTTP/1.1
Host: anquan.4399.com
Connection: keep-alive
Content-Length: 16
Pragma: no-cache
Acecept: */*
Origin: https://anquan.4399.com
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0(Linux: Android 4.4.4; Nexus 5 Build/KTU84Q) AppleWebKit/537.36(KHTML, like Gecko) Version/4.0 Chrome/33.0.0.0 Mobile Safari/537.36 4399GameCenter/3.9.1.62(android;Nexus 5;4.4.4:1080x1776;WIFI;1236.213;wap4399)
Content-Type: application/x-www-form-urlencoded: charset=UTF-8
Referer: https://anquan.4399.com/pwd/?_a=editPwd
Accept-Encoding: gzip,defalte
Accept-Language: zh-CN,en-US:q=0.8
Cookie:_SA_f94036cd826792b4da49a7c7da0b145d
password=l123456
```

Figure 8. The network communication data of 4399 game SDK
图 8. 4399 游戏 SDK 的网络通信数据

6.7. 违反安全规则 7

我们主要分析了这些游戏 SDK 在登陆和支付过程中，服务器是否每次都验证接收到的数据的签名。结果表明，在当乐游戏 SDK 的支付过程中，点击立即支付之后，我们修改 SDK 发送给 SDK 服务器的信

Figure 9. The tampered order displayed to the user
图 9. APP 向用户展示被篡改的订单

Table 4. The network communication data of the Snail Game SDK
表 4. 蜗牛游戏 SDK 的网络通信数据

Name	Value
gameId	36
paymentId	200
passport	BSNMDBDNMXMS
retypePassport	BSNMDBDNMXMS
areaId	0
imprestDestination	1
captchaValue	{cOrderId:"",nAppId:"23"}
cardTypeId	2580
amount	1
accountid	bc5917af-2fb1-4eae-9492-63d6cf928013
productName	免商店 100 元

Table 5. The payment Activity information of SDKs
表 5. SDK 的支付 Activity 信息

游戏 SDK	订单 id	商品信息	APP 信息	支付渠道	支付金额	支付账户
dangle	无	有	无	有	有	无
SnailGameSDK	有	有	有	有	有	有
SogouGameSDK	无	有	无	有	有	有
UmipaySDK	无	无	无	有	有	无
BDGameSDK	无	有	无	有	有	有
AnzhiSDK	无	无	有	有	有	无
4399SDK	无	有	无	无	有	有
tencent	无	有	有	有	有	无
laohuSDK	无	有	无	有	有	无
37SDK	无	无	无	有	有	有
lianyun	无	有	无	有	有	无
DouyuSDK	无	有	有	无	有	无
iSDK	无	有	有	有	有	有
MiguSDK	无	无	无	有	有	无
mzw_onlinesdk	有	有	有	有	有	无
Amigo Play SDK	无	无	有	无	有	无
hmPaySDK	无	无	无	有	有	无
guopan	无	无	无	有	有	无
UCGameSDK	无	有	无	有	有	有
SuperSdk	无	有	无	无	有	无
49sdk	有	有	有	无	有	无

息中的签名，服务器收到携待错误签名的信息后没有报错，而是返回订单号；同样，37 游戏 SDK 也存在这种情况。表明这些 SDK 的服务器并不是每次都正确地验证了签名。

7. 结束语

本文为了展现游戏 SDK 的实现细节并揭示潜在的安全风险，对 Android 游戏 SDK 进行了全面分析，并总结了七条安全规则，然后揭示了违反安全规则后的严重后果并提出了检测方法，最后对 21 个游戏 SDK 及上千个 Android 应用市场中的游戏 APP 进行了实验。实验结果表明，虽然这些安全规则被开发人员熟知，但多数游戏 SDK 仍然没有遵守规则，导致上百个游戏应用程序易遭受攻击。我们希望我们的研究能够提醒并指导游戏应用程序开发人员和游戏 SDK 开发人员开发更安全的游戏 APP。

基金项目

国家自然科学基金项目(U1536121, 61370195)。

参考文献

[1] Ma, Z., Wang, H., Guo, Y., *et al.* (2016) LibRadar: Fast and Accurate Detection of Third-Party Libraries in Android

Apps. *Proceedings of the 38th International Conference on Software Engineering Companion*, Austin, TX, USA, 14-22 May 2016, 653-656. https://doi.org/10.1145/2889160.2889178

[2] Li, M.H., Wang, W., Wang, P., *et al.* (2017) LibD: Scalable and Precise Third-Party Library Detection in Android Markets. 2017 *IEEE/ACM* 39*th International Conference on Software Engineering (ICSE)*, Buenos Aires, 20-28 May 2017, 335-346. https://doi.org/10.1109/ICSE.2017.38

[3] 环球网. 手机游戏市场规模将破千亿巨头领跑发行市场[EB/OL]. https://m.huanqiu.com/r/MV8wXzExMjAwOTAxXzE4MV8xNTA0MTY4MjAw, 2017-08-31.

[4] Jain, K. (2015) The Hacker News. Warning: 18,000 Android Apps Contains Code That Spy on Your Text Messages.

[5] Chen, Y., Li, T., Wang, X.F., *et al.* (2015) Perplexed Messengers from the Cloud: Automated Security Analysis of Push-Messaging Integrations. *ACM Sigsac Conference on Computer & Communications Security*, Denver, CO, USA, 12-16 October 2015, 1260-1272. https://doi.org/10.1145/2810103.2813652

[6] Yang, W.B., Zhang, Y.Y., Li, J.R., *et al*. (2017) Show Me the Money! Finding Flawed Implementations of Third-Party In-App Payment in Android Apps. *Network & Distributed System Security Symposium*, San Diego, CA, USA, February 26-March 1 2017. https://doi.org/10.14722/ndss.2017.23091

[7] Backes, M., Bugiel, S., Derr, E., *et al.* (2016) Reliable Third-Party Library Detection in Android and Its Security Applications. *Conference on Computer and Communications Security*, Vienna, Austria, 24-28 October 2016, 356-367.

[8] Soh, C., Tan, H.B.K., Arnatovich, Y.L., Narayanan, A. and Wang, L (2017) LibSift: Automated Detection of Third-Party Libraries in Android Applications. *Software Engineering Conference*, Asia-Pacific, Hamilton, New Zealand, 41-48.

[9] Demetriou, S., Merrill, W., Yang, W., *et al*. (2016) Free for All! Assessing User Data Exposure to Advertising Libraries on Android. *Network & Distributed System Security Symposium*, San Diego, CA, USA, 21-24 February 2016, 15 p. https://doi.org/10.14722/ndss.2016.23082

[10] Wang, H., Zhang, Y.Y., Li, J.R., *et al*. (2015) Vulnerability Assessment of Oauth Implementations in Android Applications. *Annual Computer Security Applications Conference*, Los Angeles, CA, USA, 7-11 December 2015, 10 p.